École Nationale des Ponts et Chaussees

P H D T H E S I S

to obtain the title of

PhD of Science

of the École Nationale des Ponts et Chaussées

Specialty : Mathematics,Computer Science

Defended by

Nicolas Thorstensen

Manifold learning and applications to shape and image processing

Thesis Advisor: Renaud Keriven

prepared at École Nationale des Ponts et Chausseés,
CERTIS Team

defended on November 26, 2009

Jury :

Reviewers :	Frank Nielsen	-	LIX
	Nikos Paragios	-	MAS
Advisor :	Renaud Keriven	-	IMAGINE/LIGM
Examinators :	Gabriel Peyré	-	CEREMADE
	Michel Couprie	-	ESIEE

École Nationale des Ponts et Chaussées

Thèse

pour l'obtention du grade de

Docteur

de l'École Nationale des Ponts et Chaussées

Spécialité : Mathématiques, Informatique

Soutenue par

Nicolas Thorstensen

Apprentissage de variétés et applications à l'analyse de formes et images

Directeur de thése: Renaud Keriven

préparée à l'École Nationale des Ponts et Chaussées,
CERTIS Team

soutenue le 26 November, 2009

Jury :

Rapporteurs:	Frank Nielsen	-	LIX
	Nikos Paragios	-	MAS
Directeur de thèse :	Renaud Keriven	-	IMAGINE/LIGM
Examinateurs :	Gabriel Peyré	-	CEREMADE
	Michel Couprie	-	ESIEE

Contents

1 Introduction **5**

1.1 General Context 6
1.2 Manifold Learning 10
1.3 Organisation 13
1.4 Contributions 15

I Background **17**

2 Geometry of metric spaces **19**

2.1 Elementary Metric geometry and Topology 20
2.2 Length spaces 23
2.2.1 An Example of Length Space 25
2.3 Smooth Length Spaces or Riemannian geometry 26
2.3.1 Manifold 26
2.3.2 Tangent Space 29
2.3.3 Riemannian Manifold 31
2.3.4 Normal coordinates 33
2.3.5 Laplace Beltrami operator 33
2.4 Mappings between metric spaces 34

3 Manifold learning **37**

3.1 Linear Methods 39
3.1.1 Principle Component Analysis 39
3.1.2 Multidimensional Scaling 41
3.2 Nonlinear methods 44
3.2.1 Graph based methods 44
3.2.2 Isomap 47
3.2.3 Locally Linear Embedding (LLE) 49
3.2.4 Hessian Eigenmaps 52
3.2.5 Laplacian Eigenmaps 55
3.2.6 Diffusion Maps 57
3.2.7 Summary of metric learning algorithms 60

II Applications **63**

4 Non-linear Manifold learning and Applications to Shape and Image Denoising **65**

4.1 Introduction . . . 67
4.1.1 Contributions . . . 69
4.2 Background . . . 70
4.2.1 Kernel methods . . . 71
4.2.2 Calculus of shapes . . . 76
4.2.3 Mean on statistical manifolds . . . 81
4.2.4 Pre-Image . . . 83
4.2.5 Out-of-sample extension . . . 84
4.3 Normalization in Gaussian Kernel PCA . . . 87
4.3.1 Pre-image in Kernel PCA . . . 89
4.3.2 Application in image denoising . . . 91
4.4 Pre-Image as Karcher means . . . 94
4.4.1 Related work . . . 94
4.4.2 Learning a set of shapes . . . 94
4.4.3 Shape interpolation as Karcher mean . . . 95
4.4.4 Pre-Image and manifold interpolation . . . 96
4.4.5 Implementation issues . . . 97
4.4.6 Results . . . 98
4.5 Discussion . . . 103

5 Non-rigid Shape matching using Geometry and Photometry **105**

5.1 Introduction . . . 107
5.1.1 Our contributions . . . 111
5.2 Background . . . 112
5.2.1 Geometric methods . . . 112
5.2.2 Photometric methods . . . 115
5.2.3 Where geometry meets photometry . . . 117
5.3 Variational correspondence optimization using geometry and photometry . . . 118
5.3.1 Problem formulation . . . 118
5.3.2 Geometry . . . 119
5.3.3 Photometry . . . 119
5.3.4 Optimization . . . 123
5.3.5 Discretization . . . 123
5.3.6 Initialization . . . 124
5.3.7 Gradient descent . . . 124
5.4 Results . . . 127

5.4.1 Validation 127
5.4.2 Real data 127
5.5 Discussion 128

III Conclusion 129

6 Conclusion 131
6.1 Conclusion 131

A Computing $\mathbb{T}_{s_\Theta}\mathcal{X}$ 141
A.1 Interpolation by projecting into the tangent space 141
A.1.1 Symmetric Difference 143
A.1.2 L2 distance 144

B Gradient of the Diffusion maps : $\nabla_s \Psi(s)$ 145
B.1 Density independent kernel 145
B.1.1 Computing $\tilde{w}(.,.)$ and $p(.,.)$ 146
B.1.2 Derivative of the anisotropic kernel $p(.,.)$ 146
B.2 Approximation of Heavyside function 146
B.3 Approximation of Dirac function 147

C Gradient of distance function 149
C.1 Gradient of the distance function on a triangle mesh 149

Bibliography 151

Abstract

The amount of data is continously increasing through online databases such as Flicker[1]. Not only is the amount of stored data increasing constantly but also the data itself is highly complex. The need for smart algorithms is obvious. Recently, manifold learning has made a strong entry into the computer vision community. This method provides a powerful tool for the analysis of high-dimensional complex data. Manifold learning is based on the assumption that the degrees of freedom of the data are much smaller than the dimension of the data space itself. More specifically, these methods try to recover a submanifold embedded in a high-dimensional space which can even be dimensionally infinite as in the case of shapes. The output of such an algorithm is a mapping into a new space (commonly refered to as feature space) where the analysis of data becomes easier. Often this mapping can be thought of as a parametrization of the dataset.
In the first part of this thesis, we introduce the concepts and theory of metric spaces providing the theoretical background to manifold learning. Once the necessary tools are introduced, we will present a survey on linear and non-linear manifold learning algorithms and compare their weak and strong points.
In the second part, we will develop two applications using manifold learning techniques. In both applications manifold learning is applied to recover or approximate the metric on the original space data space. In this way distance between points in the original space can be computed using the metric in the feature space. This allows the formulation of distance based optimization problems.
In this spirit, we tackle a first problem known under the name of Pre-image problem. We will look at this problem in the context of Kernel PCA and diffusion maps. It turns out, that previous Pre-image methods based on Kernel PCA used a wrong normalization in centered feature space. We propose a more subtle normalization improving previously proposed algorithm for the computation of the Pre-image. We then look at the same problem in the context of diffusion maps and shapes. A manifold interpolation algorithm is proposed for solving the Pre-image correctly. We will show very promising results in image and shape denoising.
The second problem is finding a set of correspondences between two non rigid triangle meshes. Our solution builds upon a recent method which allows to embed the first mesh into the second. The method relies on the intrinsic distance function of the surfaces and therefore is purely geometrical.

[1] www.flickr.com/

Unfortunately, when dealing with dynamic multiview stereo the reconstructed surfaces do not satisfy the conditions of the geometric algorithm. Our solution to this problem is an improved energy function taking the geometry of surface and the photometric information from the images into account. This results in a more robust correspondence matching algorithm for non-rigid surfaces.

Key-words : metric geometry, manifold learning, kernel PCA, diffusion maps, multidimensional scaling, pre-image, shape space, non-rigid correspondence problem.

Résumé

Grâce au bases de données en ligne , le volume de données ne cesse d accroitre. Non seulement la quantité de donnes augmente mais aussi la complexité des donnes est hautement complexe. Ce fait nécessite le développement d algorithmes performants. Récemment, une nouvelle classe de méthodes connue sous le nom de: "apprentissage de variétés" a été introuduite. Ces méthodes présentent un formalisme intéressant et performant pour l analyse de données a très haute dimension. Ces méthode assument que les degrés de liberté dans les données sont bien plus petit que la dimension de l espace des données. Le but de ces algorithmes est retrouve une variété plongée dans un espace á haute dimension (voir infinie). La sortie d un tel algorithme es une fonction transformant les données dans un espace (espace de feature) où l'analyse devient plus facile. Souvent cette fonction est considère comme une para métrisation de la variété. Dans la première partie de ce manuscrit, nous allons introduire les idées principales ainsi que la théorie des espaces métriques. Ceci nous fournissera les outils de bases pour les méthodes d'apprentissage de variétés. Par la suite nous présenterons des méthodes linéaires et non- linéaires pour l'apprentissage de variétés et analyserons leurs points forts et faibles. La deuxième partie développera deux applications en utilisant l'apprentissage des variétés. Dans les deux cas l'apprentissage de variétés est appliqué pour approximer le métrique dans l espace initiale. Ainsi la distance entre points dans l'espace originale peut être approximé en utilisant la métrique dans l'espace feature. Ainsi nous pouvant résoudre des problème d optimisation basée sur les distance entre points. Dans cette idée nous regardons le premier problème connu sous le nom "problem de la pre-image". Nous analyserons ce problème dans le contexte de la ACP a noyau and la technique des diffusion maps. Nous remarquerons que les méthodes basées sur l ACP a noyau appliqué une mauvaise normalisation pour les calculs. Nous proposons pour cela une normalisation raffinée améliorant les résultats des méthodes déjà développées. Par la suite nous considérons le même problème dans le cadre des diffusion maps appliqués a l'analyse de formes. Nous proposons une méthodes d interpolation pouvons résoudre le problème de la pré-image. Les deux méthodes sont appliqués au de bruitage d'images et de formes avec des résultats très prometteurs.

Le deuxième problème est celui de retrouver un ensemble de correspondance entre deux maillage non rigide. Notre solution utilise une méthode récente qui permet de plonger un premier maillage dans un deuxième. Cette méthode est basée sur la distance intrinsèque entre les points du maillage et est donc purement géométrique. Malheureusement quand les maillages sont issues de la stéréo photométrique , les surfaces ne sont pas suffisamment précises

pour satisfaire les hypothèses de l'algorithme géométrique. Notre solution a ce problème est une optimisation raffinée, qui prend en compte non seulement la géométrie de la surface mais aussi l'information photométrique des images. Ainsi nous obtenons un algorithme robuste pour retrouver un ensemble de correspondance entre deux maillages non rigides.

Mots-clefs : gémétrie métrique, apprentissage de variétés, ACP à noyau, diffusion maps, multidimensional scaling, pré-image, espace de formes, problème de correspondance non rigides.

Chapter 1

Introduction

Contents

1.1 General Context 6

1.2 Manifold Learning 10

1.3 Organisation 13

1.4 Contributions 15

1.1 General Context

Imaging technology is used extensively in the medical sector where it has become a standard technology for the diagnosis of diseases. These non invasive technologies such as Magnetic Resonance Imaging (MRI) allow the study of morphological variations in the brain. For instance there is an obvious correlation between a decrease of white matter and the Alzheimer diseases. Further, such in-vivo imaging analysis tools provide a way to understand the brain functionality. This is also known as functional brain imaging and allows the medical research to visualize brain activity while a person carries out certain tasks. Although imaging technology is more and more accurate, algorithm engineers still have to deal with imaging problems such as noise in the image which is inherent to the acquisition tools (c.f. MRI). Intelligent preprocessing and postprocessing algorithms are being developed to suppress artefacts from the images while not altering the important content. A common technique to reduce the noise in an image is to convolve the image with a Gaussian function. Unfortunately, this naive approach blurs significant features such as edges and destroys some of the geometric information in the image. A more subtle approach is to design an algorithm blurring homogenous regions but without acting on the frontier between such regions[Tomasi 1998]. The reason for this approach is that the frontiers or edges between homogenous regions are very important for the perception of the geometric content in the image. The development of smart algorithms enhancing image quality is the topic of **Image Processing** [Sonka 1998]. Image enhancement and denoising is not the only concern in image processing. The following list gives an incomplete overview of typical operations in image processing:

- Geometric transformations such as rotation, shearing, scaling
- Color corrections such as brightness and contrast adjustments, quantization, or color translation to a different color space (typical operations in digital cameras)
- Interpolation and demosaicing
- Image registration (aligning two or more images)
- Image morphing
- Image recognition (extract text from images)
- Image segmentation

With the increasing progress of digital consumer technology, digital cameras and digital video cameras are more accessible to a broad public. Due to the widespread usage of digital cameras, the amount of images and videos is increasing constantly. Thanks to the internet, digital content is also diffused easily across the world. Dealing with this huge amount of data calls for intelligent software for automatic image and video recognition, classification and, categorization. The organization is not only challenging on the internet. On the individual side, people are overwhelmed by the digital content they create so easily. Smart algorithms helping to organize or extract high-hevel information are the subject of **Computer Vision** [Forsyth 2002]. Often Computer Vision is seen as the inverse operation of Computer Graphics, thus understanding what was created in this image and segmenting different objects in the image (object segmentation). The next step is to infer what type of objects are depicted in the image. Another intent could be a $3D$ reconstruction of the object previously identified. All of these different stages are done bottom-up. Thus, first a low level treatment is performed where salient points are found which helps in the next step to perform segmentation of the object. Once the object is delimited from the rest of the scene it is interesting to recognize the object which is often done by using Machine Learning. Generally, the community agrees upon the following classification of algorithms into three categories:

- low-level vision
- mid-level vision
- high-level vision

Low-level vision is the sum of operations on the pixel-level of an image. Low-level vision and image processing are often used synonymously. In Computer Vision, low-level vision provides methods to filter images and compute features such as edges. Mid-level vision includes techniques such as segmentation, grouping, perceptual organization and fitting. All these techniques seek to provide a more compact description of the image which might be used for further processing such as high-level vision. High-level computer vision can be divided into geometric methods and probabilistic methods. Geometric methods target to understand the geometric configuration and relation between the observed objects. On the other hand, probabilistic and inferential methods try to recognize and classify the objects observed in order to understand the content of the image.

Another common application of **Computer Vision** is the post processing of movies and images in order to render artificial objects such as explo-

sions into the images. These techniques are also known under the name of special effects. In general, **Computer Vision** is ubiquitous in the movies industry. When it comes to animated movies, it is especially challenging to make the movies as realistic as possible. For instance the motion animation of characters requires a special technique known as **Motion Tracking** [Moeslund 1999, Rosenhahn 2008]. The producer of an animation movie might want to make one of the characters dance like a ballerina. Generating the motion of a ballet dancer manually is very time consuming and the result might appear unrealistic. So the idea of **Motion Tracking** is to record a real ballerina dancing in a camera rig and compute the motion parameters of the recorded scene. The motion parameters are then employed to animate virtual characters in an animation movie resulting in very realistic ballerina moves. **Motion Tracking** can be divided into optical and non-optical methods. Non-optical methods are divided into mechanical, inertial and magnetic methods. We focus in this thesis on optical systems. Optical systems use data captured from cameras to triangulate the 3D position of an object. The three most important methods are :

- Passive markers
- Active markers
- Markerless

Passive optical systems use retroreflective markers attached to the object generating bright spots in the recorded images. In this manner, the markers position in the images are easily pinpointed and triangulated. Contrary to passive markers, active optical systems use LEDs instead of reflecting markers. 3D positions are triangulated by activating the LEDs and identifying them by their relative position to each other using specific software. Lastly, markerless optical systems solely rely on the intensity images recorded by the camera rig.

Many of the above mentioned problems appear in **Computer Vision** and nowadays are being solved using Machine Learning techniques. **Machine Learning**[Hastie 2003] is about the development of algorithms to automatically extract patterns or regularities from data. In the case of **Computer Vision**, the data is usually composed of images as in Figure 1.2. A good algorithm is measured by its capacity to generalize, which means if new data is recognized correctly based on the available samples. Typical tasks of Machine learning are

- Unsupervised: determines how the data is organized

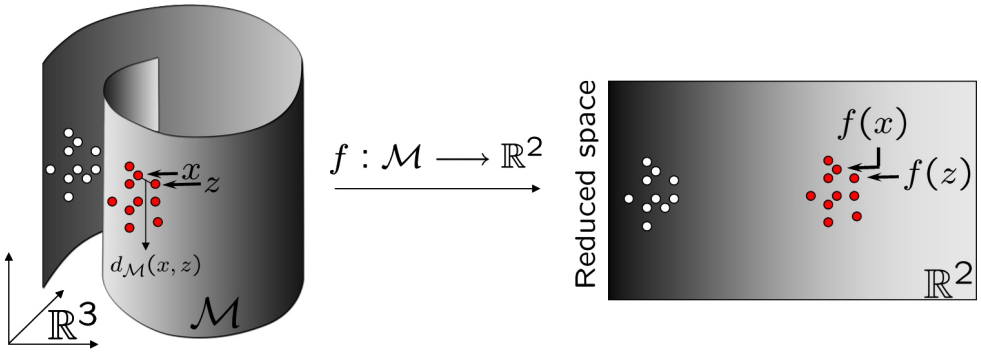


Figure 1.1: A set of points in $\mathbb{R}^3$ lying on a manifold sampled from a 2-dimensional manifold and the corresponding parametrization f of the manifold mapping $\mathcal{M}$ ot $\mathbb{R}^2$.(taken from [Etyngier 2008])

- Semi-supervised: given labelled and unlabelled points, build a labeling function
- Supervised: given labelled points, build a labelling function
- Reinforcement learning: learns how to act given an observation of the world.
- Transduction: the specific test cases are known beforehand and exploited in the training stage. In contrast to the 2-steps inductive inference process, which first learns a mapping from the entire input space to the output space and can then apply it to any novel input, transductive inference only predicts the labels of known test points and uses the unlabeled data distribution in the learning procedure.

Applications for machine learning include machine perception, computer vision, natural language processing, syntactic pattern recognition, search engines, medical diagnosis, bioinformatics, brain-machine interfaces, detecting credit card fraud, stock market analysis, classifying DNA sequences, speech and handwriting recognition, object recognition in computer vision, game playing, software engineering, adaptive websites and robot locomotion.

Over time, several learning tools were designed for the processing of linear data. Unfortunately, the assumption of linear data is invalid most of the time when dealing with real world data. For instance, taking a look at the example in Figure 1.1 the data has a non-linear dependency, which leads to the question of how to treat this data correctly.

1.2 Manifold Learning

Manifold learning deals with the inference of geometrical properties from collections of data (c.f Figure 1.2) represented in vector form and embedded in a Euclidean space or an infinite dimensional space in the case of shapes. One of the major assumptions of manifold learning techniques is that the data lies approximately on a low dimensional surface embedded in a high (possibly infinite) dimensional space (c.f Figure 1.3).



Figure 1.2: Each image is a vector in $\mathbb{R}^{4096}$, but we observe only three degrees of freedom: 2 rotation angles and 1 illumination parameter. Therefore the set of images span a 3-dimensional submanifold in $\mathbb{R}^{4096}$.(taken from http://web.mit.edu/cocosci/isomap/)

Manifold-valued data

In the context of manifold learning, these types of datasets are often called point clouds. Some of the goals of manifold learning are to determine if the data is on (or close to) a manifold and estimating the intrinsic dimensionality of the underlying manifold. In [Tenenbaum 2000, Roweis 2000a, Fan 2009, Chunguang 007, Kégl 2003, Hein 2005b, Hero 2000] the authors propose intrinsic dimension estimators looking at local neighborhoods and combining this information. If indeed the intrinsic dimension is lower then the embedding space, the goal is to find a lower dimensional representation so that the analysis of this data is eased. The idea of intrinsic dimension is equivalent to the number of degrees of freedom in a data set. A good example is the data set of faces depicted in Figure 1.2. The number of degrees of freedom in this data set is three. Indeed we observe two rotation angles of the head and a moving light source.
Linear methods such as MDS and PCA (c.f. Chapter 3) are supposed to transform the data into a new coordinate system but have the drawback of not properly handling nonlinear data. Therefore more sophisticated methods are needed.

Motivation

The problems to be considered in this thesis are denoising and mesh registration. In both cases, we will make use of manifold learning techniques. The first problem to be considered is the denoising of images and shapes in which images are considered as vectors in a finite dimensional space and shapes are represented as implicit functions in an infinite dimensional space. in this case, denoising is achieved through subspace projection onto a low dimensional representation of the data. The idea is similar to Fourier analysis, which assumes that the most important information is in the low frequencies. This point of view can also be adopted for denoising of images of shapes. For instance, given an input signal we assume that it is composed of original data mixed with additive noise. Then the original data corresponds to the low frequency components of the signal and the noise corresponds to the high frequency. Applying manifold learning to uncorrupted category specific data amounts to recovering the modes of variation in the data set. The noise can be eliminated by projecting the noisy datum onto the learned manifold. The projected signal then contains solely low frequency describing the typical frequency for a certain category of signals.

The second problem considers the problem of non-rigid mesh registration

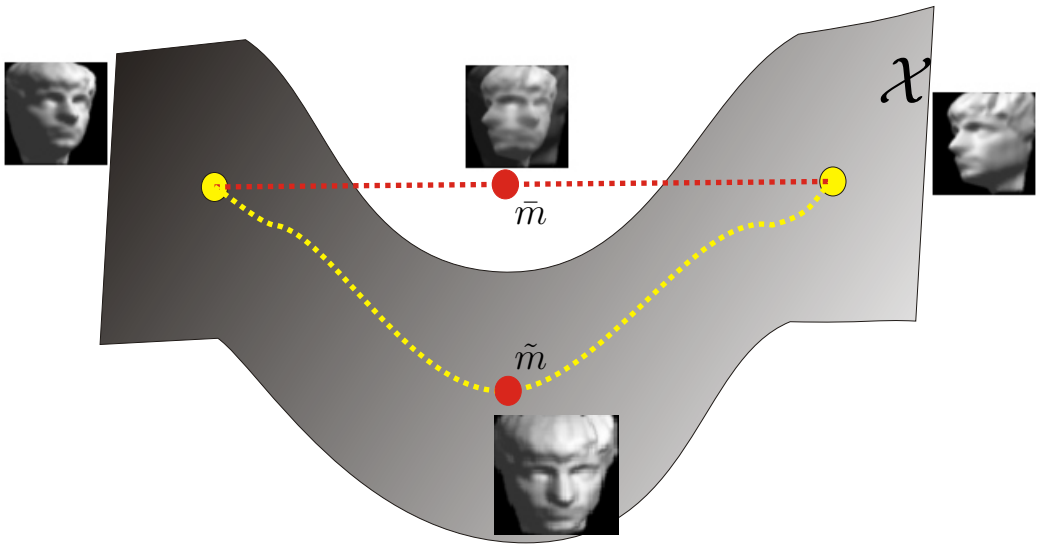


Figure 1.3: The Face data lies on a 3-dimensional curved submanifold $\mathcal{X}$ embedded in $\mathbb{R}^{4096}$. Linear combination $\bar{m}$ of two elements of $\mathcal{X}$ yields an element outside of $\mathcal{X}$. This shows that the dataspace is not a vector space. The mean between two faces $\tilde{m}$ yields an element of $\mathcal{X}$.

in a multiple camera environment. This problem is solved by computing a function which embeds the points of the first mesh onto the second mesh so that the points should correspond to similar locations. This problem was previously solved using a MDS-like approach [Bronstein 2006b]. We show that in the case of reconstructed meshes the isometry assumption is not fully satisfied due to local elastic deformation. The routine proposed in [Bronstein 2006b] does not hold up to such perturbations and is not adapted to our case. To cope with this problem, we modify the optimization functional in [Bronstein 2006b] in order to enforce photometric consistency among image pairs and in time. We then compute the embedding solving the augmented energy. In this way local elastic deformations do not affect the embedding computation too strongly.

1.3 Organisation

One of the objectives of this thesis is to provide an educational and self contained introduction to the theory of differential manifolds and the subject of manifold learning in machine learning. Therefore, the first part starts with the definition of metric spaces. This short introduction on metric spaces provides a justification for the use of metrics in the real world in order to quantify similarities between entities. The beginning of the first chapter introduces rather general mathematical concepts and continues with a geometric view of metric spaces. This leads to the theory of smooth manifolds. The introduction of the terminology of manifolds opens the door to the world of geometric machine learning or commonly known as manifold learning. The second part of this thesis applies the algorithms introduced in the first part to solve two standard problems in Computer Vision.

Chapter 2 is a standard introduction to metric geometry or smooth metric geometry. A set of definitions and concepts are presented. The structure follows the monograph of doCarmo [Do carmo 1992]. This introduction is structered in a slightly different way to better fit the ideas in this thesis. First, several properties of surfaces used for the analysis of their geometry are presented. Second, we will characterize properties of transformations between surfaces forming the second part of this introduction to differential geometry.

Chapter 3 will provide an overview of the major manifold learning methods. We will begin with the presentation of the most prominent linear methods to afterwards move on to the non-linear methods. The intent is offer a nice and easy read without neglecting the formalism necessary for the correct treatment of these methods. Furthermore, several illustrations were added to facilitate the understanding and the geometric intuition behind the scene.

Chapter 4 will consider the problem of pre-image computation. Given a set of training samples, manifold learning techniques embed the training samples into a new space called feature space. As the feature space is more suitable for the analysis of data, operations such as projections and nearest neighbors search are preferably executed in the feature space. Unfortunately, there is no inverse mapping from feature space back to the original space

as the embedding is only known for the training samples. Therefore, for a point with different embedding coordinates than those of the training samples, one has to approximate the pre-image in the data space. As shown by previous work, solving the pre-image problem can be used for denoising or interpolation of datasets. We consider this problem in the context of Kernel PCA and show that special care has to be taken on the issue of centering the data set. The observation made shows significant improvements when applied to the state of the art algorithms for denoising of images. In the second part, we provide a new method for interpolation in the manifold in the context of diffusion maps. The pre-image problem is solved by interpolating between samples. Our methods shows very good results in image and shape denoising.

Chapter 5 will cover the problem of non-rigid registration of 3-D triangle meshes in a multiple view camera environment. This problem is also known as correspondence problem. Unfortunately, most of the existing algorithms make assumptions (almost perfect isometry between the two surfaces for example) which are not true in the case of surfaces reconstructed from images. Such algorithms were mainly designed for the registration of surfaces aquired by range scanners. But image based surface reconstruction often yields imprecise meshes due to occlusions, bad camera calibration or because of the algorithm itself. Therefore, we build upon previous registration methods and exploit the photometric information in the images to render a purely geometric registration method tractable for the case of imprecise meshes issued from 3D reconstruction methods. Firstly, the standard form of the problem is presented for the case of geometric rigid registration. A vast amount of literature was published on this topic. We limit ourselves to the work directly related to our own, with the aim of reviewing the previous work done on mesh registration. Finally, we provide an algorithm which is a combination of geometric properties of the surface and photometric properties of the images. More precisely, we define an energy on a set of correspondences minimizing the photometric disparity in the images. The geometry is taken into account by regularization term which is added to the energy to ease the minimization. The regularization term is a prior on the admissible deformations between the triangle meshes. More precisely, we assume that the two surfaces are intrinsically isometric. At the end we will show some of the results achieved with the algorithm.

Chapter 6 forms the conlcusion of this thesis and will provide a short review of the work presented in this manuscript. At the same time we present

some promising research avenues which lie in the continuation of this work. Including several ideas which could be explored.

1.4 Contributions

The purpose of this thesis is twofold. Firstly, we will provide an introduction to the theory of metric geometry as proposed in the book by Burago et. al. [Burago 2001]. We will mention important theorems of this theory in order to provide a connection to the topic of manifold learning. Several concepts translate to efficient numerical algorithms. This first part provides a concise introduction to the topic of manifold learning and should help to get through the very rich literature and understand weak and strong points of each method and the connections between the methods.
Secondly, we will develop two algorithms using manifold learning techniques. In a first instance we will consider the problem of pre-image computation in the case of an infinite-dimensional shape space. This idea was first suggested by Charpiat [Charpiat 2005] and further developed by Etyngier [Etyngier 2007b]. Based on an idea of Etyngier[Etyngier 2007b], we will propose an efficient way to solve the pre-image of points in a high dimensional space. This requires the projection of the point onto a manifold. We will propose to model the underlying manifold as the set of Karcher means of close sample points. The solution is solved through a non-linear interpolation of close sample points. Results on synthetic 2D shapes and on real 2D images and 3D shapes are presented and demonstrate the superiority of our pre-image method compared to several state-of-the-art techniques in shape and image denoising based on statistical learning techniques. The proposed methods were published in [Thorstensen 2008, Thorstensen 2009c]
As a second problem, we will consider the difficulty of finding correspondences between three-dimensional reconstructions of a deformable surface at different time steps. We suppose that (i) the mechanical underlying model imposes time-constant geodesic distances between points on the surface; and that (ii) images of the real surface are available. This is for instance the case in spatiotemporal shape from videos (e.g. multi-view stereo, visual hulls, etc.) when the surface is supposed to be approximatively unstretchable. These assumptions allow to exploit both geometry and photometry. In particular, we will propose an energy based formulation of the problem, extending the work of Bronstein *et al.* [Bronstein 2006b]. On the one hand, we show that photometry

(i) improves accuracy in case of locally elastic deformations or noisy surfaces and (ii) allows to still find the right solution when [Bronstein 2006b] fails due to ambiguities (e.g. symmetries). On the other hand, the usage of geometry makes it possible to match shapes that have undergone large motion, which is not possible with usual photometric methods. Numerical experiments prove the efficiency of our method on synthetic and real data. The results of this work were presented at the ACCV conference in Xi'An [Thorstensen 2009b].

Part I

Background

Chapter 2

Geometry of metric spaces

Contents

2.1 Elementary Metric geometry and Topology 20

2.2 Length spaces 23

2.2.1 An Example of Length Space 25

2.3 Smooth Length Spaces or Riemannian geometry . . . 26

2.3.1 Manifold 26

2.3.2 Tangent Space 29

2.3.3 Riemannian Manifold 31

2.3.4 Normal coordinates 33

2.3.5 Laplace Beltrami operator 33

2.4 Mappings between metric spaces 34

Overview

In this first Chapter, we will provide an introduction to the geometry of metric spaces and machine learning methods. The intention is to provide some elementary concepts about the mathematical notions necessary for understanding the notations involved in manifold learning. Further, we present the most recent manifold learning algorithms which are used in this thesis to solve specific problems in Computer Vision.

History

The idea of using the distance function to define metrics on sets, dates back to 1905 in a paper of the Romanian mathematician Dimitrie Pompeiu published in the Annales de Toulouse. Felix Hausdorff popularized this approach in his book Grundzüge der Mengenlehre published in 1914 in Leipzig. It is the well known Hausdorff metric he introduced in this text. Through the metric he provides an analytical description of a geometrical object. This was the first work in metric geometry which was strongly popularized by Burago and his coauthors in his excellent monograph on metric geometry[Burago 2001].

The subject of classical Riemannian geometry usually begins with the definition of a C^∞ manifold $\mathcal{X}$ and then introduces positive definite bilinear form G on the tangent bundle of $\mathcal{X}$. The concept of length of curves and geodesic distances on $\mathcal{X}$ is build upon the fact that G varies smoothly over $\mathcal{X}$. This provides a family of continuous norms on the tangent space $\mathbb{T}_x\mathcal{X}$ of $\mathcal{X}$. Then fundamental notions such as the covariant derivative and curvature follow from these basic definitions. Contrary to Riemannian geometry, metric geometry studies the concept of length and geodesic distance as a unique concept and shows how Riemannian geometry and its notions such as curvature and derivatives arise when the metric space exhibits enough smoothness. This chapter starts by introducing metric geometry following the books of Burago et.al [Burago 2001] and the smooth analogue also known as Riemannian geometry as in the book by DoCarmo [Do carmo 1992].

2.1 Elementary Metric geometry and Topology

The concept of "near" and "far" is a very powerful and useful utility in every day life. It classifies the relationship between two "primitives", whether they are close or far apart. An example could be a biologist classifying the resem-

blance between two dogs. The biologist would classify two dogs, let's say a German Sheppard and a sausage dog, as very distant races since they have almost no resemblance at all(far). Of course this kind of classification is not useful for computations. A metric space is the mathematical construction of this vague idea.

Definition 1. Let $\mathbb{X}$ be an arbitrary abstract set. A function $d_{\mathbb{X}} : \mathbb{X}\times\mathbb{X} \mapsto \mathbb{R}\cup\{\infty\}$ is a metric on $\mathbb{X}$ if the following conditions are met for all $x_1, x_2, x_3 \in \mathbb{X}$.

(i) Non-negativity: $d_{\mathbb{X}}(x_1, x_2) \geq 0$ and $d_{\mathbb{X}}(x_1, x_2) = 0 \Leftrightarrow x_1 = x_2$.

(ii) Symmetry: $d_{\mathbb{X}}(x_1, x_2) = d_{\mathbb{X}}(x_2, x_1)$.

(iii) Triangle inequality: $d_{\mathbb{X}}(x_1, x_3) \leq d_{\mathbb{X}}(x_1, x_2) + d_{\mathbb{X}}(x_2, x_3)$.

Then the pair $(\mathbb{X}, d_{\mathbb{X}})$ is a metric space. The elements of $\mathbb{X}$ are called points of the metric space. The function $d_{\mathbb{X}}(x_1, x_2)$ returns the distance between two points. A very well known instance of a metric space is the three dimensional Euclidean space $\mathbb{R}^3$ with the Euclidean metric. In the rest of the document, we will often write a metric space $\mathbb{X}$ instead of a metric space $(\mathbb{X}, d_{\mathbb{X}})$.
In general, any normed vector space V is a metric space with the metric induced by the norm.

Definition 2. Let V be a vector space. A function $\|.\| \mapsto \mathbb{R}$ is a norm on V if the following conditions are met for all $v_1, v_2 \in V$ and $k \in \mathbb{R}$.

(i) Non-negativity: $|v| > 0$ if $v \neq 0$, else $|0| = 0$.

(ii) Linearity: $|kv| = |k|\,|v|$.

(iii) Triangle inequality: $|v_1 + v_2| \leq |v_1| + |v_2|$ with $k \in \mathbb{R}$.

So a normed vector space is a vector space equipped with a norm. For instance, the Euclidean space $\mathbb{R}^n$ is a normed space with norm $\left|(x^1, \ldots, x^d)\right| = \sqrt{(x^1)^2 + \cdots + (x^d)^2}$.
Lastly, we notice that a norm is called Euclidean if it is associated with some scalar product.

Definition 3. Let V be a vector space. A scalar product $\langle .,. \rangle : V \times V \mapsto \mathbb{R}$ on V is a symmetric bilinear form F whose associated quadratic form is positive definite, *i.e.* , $F(v,v) > 0$ for all $v \neq 0$.

The definition of bilinear forms is given as

Definition 4. Let V be a vector space. A bilinear form F on V is a function of two variables $V \times V \mapsto \mathbb{R}$ satisfying the following equations:

$$F(v_1 + v_2, v_3) = F(v_1, v_3) + F(v_2, v_3) \tag{2.1}$$

$$F(\lambda v_1, v_2) = \lambda F(v_1, v_2) \tag{2.2}$$

$$F(v_1, v_2 + v_3) = F(v_1, v_2) + F(v_1, v_3) \tag{2.3}$$

$$F(v_1, \lambda v_2) = \lambda F(v_1, v_2) \tag{2.4}$$

with $\lambda \in \mathbb{R}$.

A norm associated with a scalar product $\langle .,. \rangle$ is given by $|v| = \sqrt{\langle v, v \rangle}$. Further, we mention that two vectors in a Euclidean space are called orthogonal it their scalar product equals zero. An orthonormal base or frame is a collection of unit vectors where the scalar product between any two unit vector equals zero. Given a collection of linearly independent vectors, an orthonormal frame can be constructed using the Gram-Schmidt orthogonalization algorithm. Every finite dimensional Euclidean space provides an orthonormal basis and thus is isomorphic to $\mathbb{R}^n$. This result is particular practical since all geometric results of $\mathbb{R}^n$ can be applied to Euclidean spaces of finite dimension.

With the metric structure in hand, we can quantify the idea of "near" and "far" between elements of metric spaces. We can characterize the connectivity of sets through small neighborhoods of elements which gives rise to a topology of the sets. Therefore we start by defining small neighborhoods as balls.

Definition 5. Let $\mathbb{X}$ be a metric space and an element $x \in \mathbb{X}$ and $r > 0$. The set $\mathcal{B}_r(x) = \{x' \in \mathbb{X} : d_{\mathbb{X}}(x, x') < r\}$ of points with distance less than r from x is called an open metric ball centered at x with radius r. Similarly, $\overline{\mathcal{B}}_r(x) = \{x' \in \mathbb{X} : d_{\mathbb{X}}(x, x') \leq r\}$ is called a closed ball.

With the definition of open and closed balls we can define open sets (sets without boundary) and closed sets (with boundary) which are defined as the

complement of open sets. We can study the topology of $\mathbb{X}$ induce by the metric. The topology of a set allows one to study properties such as connectivity. The space $\mathbb{X}$ is not connected if it is the union of two disjoint open nonempty sets. The converse of this property implies that $\mathbb{X}$ is connected. In a metaphoric way, a set is connected if we can draw a line between every pair of elements in the set without lifting the pen. Further, we are able to investigate convergence of a sequence of points $\{x_i\} \in \mathbb{X}$ with $i = 1, \ldots, \infty$. If there exists a constant K for every $\varepsilon > 0$ such that $x_i \in \mathcal{B}_\varepsilon(x)$ for all $i \geq K$ then the sequence converges to a limit $x \in \mathbb{X}$ as $i \to \infty$. Of course, these properties depend on the chosen metric. Therefore some call it convergence in the metric in order to emphasize the dependency on the metric. For example, a sequence of points might have a limit in a L_1 metric whereas it might not converge when using a L_∞ metric.

2.2 Length spaces

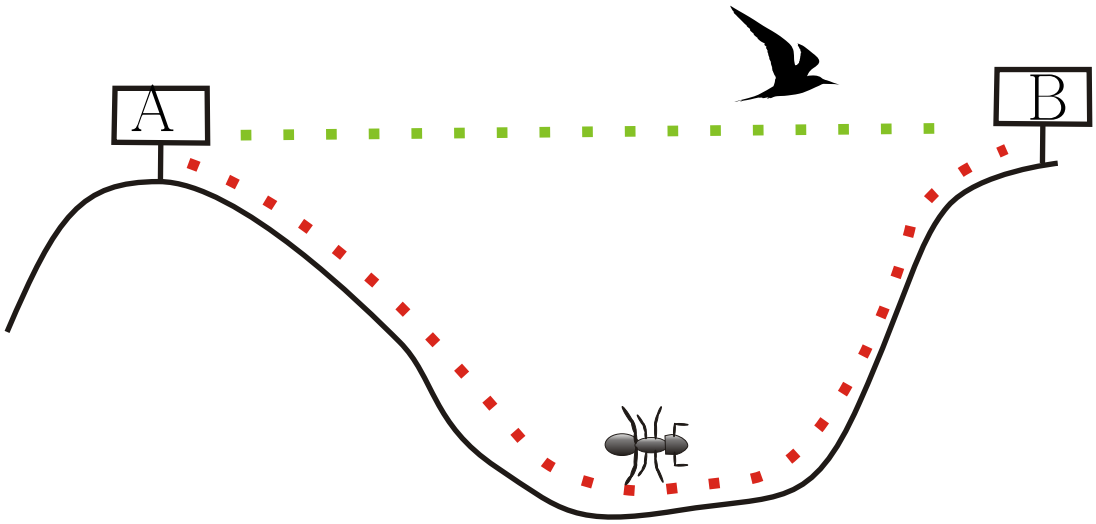


Figure 2.1: Travel length for the bird is shorter as the travel length of the ant. The ant is constrained to move in a different metric space than the bird.

A very intuitive example is provided in a monograph on metric geometry by Burago [Burago 2001] (c.f. Figure 2.1) which we will illustrate here. A bird and an ant meet on the summit of a mountain. After talking about their breakfast they decide to travel to the neighboring mountain from which they have been told the view is much better. So when they met again on the other side the bird asked what took the ant so long. The answer is that they didn't travel the same distance. While the bird could take a straight path, the ant was forced to travel along a path on the ground. Although they both left the same point to reach the same endpoint they traveled differently along different paths with unequal lengths.

Definition 6. A path in $(\mathbb{X}, d_{\mathbb{X}})$ is a continuous map $\Gamma\,[a,b] \subset \mathbb{R} \mapsto \mathbb{X}$. A partition of $[a,b]$ is a subdivision $a = t_0, \ldots, t_k = b$.

We can associate a length to a path Γ

Definition 7. The length of Γ is $L(\Gamma) = \sup \Sigma_{i=1}^{k} d_{\mathbb{X}}(\Gamma(t_{i-1}), \Gamma(t_i)) \in [0, \infty]$, where the supremum is taken over all partitions of $[a,b]$. If $L(\Gamma) < \infty$ then the path is rectifiable.

Thus the smooth path can be approximated by a sequence a small line segments whose length is measured with the Euclidean distance. This idea is well-known from the Riemann integral and which has the form

$$L(\Gamma) = \int_a^b \left\| \dot{\Gamma}(t) \right\|_2 dt, \tag{2.5}$$

where $\dot{\Gamma}(t)$ is the derivative of $\Gamma(t)$ with respect to the parameter t. As we can see, the length structure is induced by the Euclidean metric $\|.\|_2$

By this observation, we see that distances emerge from the length of a path joining two points in the considered space. A length itself is a map assigning each pair of points a nonnegative number. So we call a metric space a length space when the distance is given by the infimum of the length among all admissible paths.

Definition 8. Let $(\mathbb{X}, d_{\mathbb{X}})$ be a metric space and L the length of a path Γ between $x, y \in \mathbb{X}$. Then d_L is a length metric if:

$$d_L(x,y) = \inf_{\Gamma} \{L(\Gamma)\} \tag{2.6}$$

If there is no path connecting x and y then $d_L(x,y) = \infty$

As the reader can see from the definition, length metrics are not necessarily finite. Infinite distances are encountered on disjoint sets. In fact, there exists no smooth path between two distinct components of the set and the distance between any two points of the two infinite components. There are no infinite distances in a length space if for every pair of points in $d_{\mathbb{X}}$ there exists a path between them. In this case, we say the length space is complete. A last remaining question is the uniqueness of shortest paths. In general, it is hard to say whether a geodesic (mathematical name for a shortest path) is unique or not. Consider for example a set with a spherical hole. Let's position two diametrically opposite points on the border of the hole. Then two paths minimizing the length are the solution to definition 8. One path travelling clockwise and the other path travelling counterclockwise around the hole.

2.2.1 An Example of Length Space

A very useful example of a length space is the undirected graph.

Definition 9. A graph is an ordered pair $\mathcal{G} := (\mathcal{V}, \mathcal{E})$ comprising a set $\mathcal{V}$ of vertices or nodes together with a set $\mathcal{E}$ of edges or lines, which are two-element subsets of $\mathcal{V}$

Then a path on a graph is an ordered set of edges $\Gamma = \{(v_1, v_2), \ldots, (v_{n-1}, v_n)\}$ and the associated length is the sum over all edges $L(\Gamma) = \sum_{i=1\ldots n-1} L((v_i, v_i + 1))$. For any two vertices v_1, v_2, the length metric returns the length L of the shortest path joining the two elements.

So far, we did not talk about smoothness of the objects under investigation. When one imposes smoothness on the metric spaces then the theory becomes the topic of Riemanian geometry. Riemanian geometry is the treatment of smooth parametric spaces also known under the name of manifolds. In mathematics, a manifold is a topological space which is locally isomorph to the Euclidean space $\mathbb{R}^d$. But globally the manifold does not need to look like a Euclidean space.

2.3 Smooth Length Spaces or Riemannian geometry

Manifolds are the central objects of differential geometry and play an important role in theoretical physics. Recently, theory of manifolds made an entrance into the world of machine learning. The geometric analysis of data [Hein 2007] and the associated mathematical machinery is explored to detect patterns in high dimensional manifold valued data. The purpose of the next section is to provide the reader with some elementary notions. The exposition follows do Carmo [Do carmo 1992], which is an excellent introduction to this topic.

2.3.1 Manifold

Manifolds are the generalization of vector calculus of $\mathbb{R}^d$ to curved spaces that locally look like $\mathbb{R}^d$. However, the global structure of the abstract space is often much more complicated.

Definition 10. A d-dimensional manifold is a topological space in which points can be separated by neighborhoods and where every point has a neighborhood that is homeomorphically mapped onto an open Euclidean d-dimensional ball.

From this definition, we deduce immediately that $\mathbb{R}^d$ is a manifold since it is trivially homeomorphic (definition c.f. section 2.4) to itself. Similarly, any open set of $\mathbb{R}^d$ is a manifold. This general definition is not yet sufficient to perform calculus on a manifold like the standard vector calculus in $\mathbb{R}^d$. The notion of a differentiable manifold comes hand in hand with the concept of coordinate chart. In the following, whenever we work on manifold metric space we set $\mathcal{X} = \mathbb{X}$ and $d_{\mathcal{X}} = d_{\mathbb{X}}$

Definition 11. A coordinate chart on a d-manifold $\mathcal{X}$ is a pair (U, ϕ) where U is an open subset of $\mathcal{X}$ and $\phi : U \mapsto \widetilde{U}$ is a homeomorphism between U and an open subset $\widetilde{U} = \phi(U) \subset \mathbb{R}^d$.

The chart can be thought of as assigning a set of coordinates to the points in the neighborhood U (c.f. Figure 2.2). Or, any point $x \in U$ has the coordinates $x^1(0), \ldots, x^d(0) = \phi(x)$. The purpose of such local coordinate systems

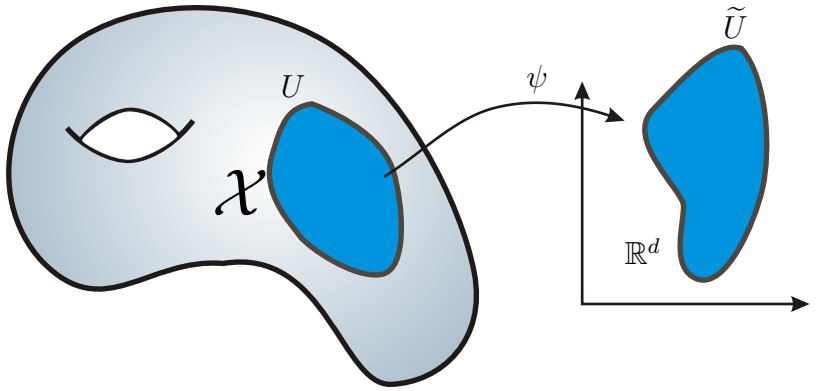


Figure 2.2: A coordinate chart (U, ψ)

become obvious in the coming sections when writing explicit expressions for derivates, tangent vectors and Riemannian metrics. Usually a single chart is

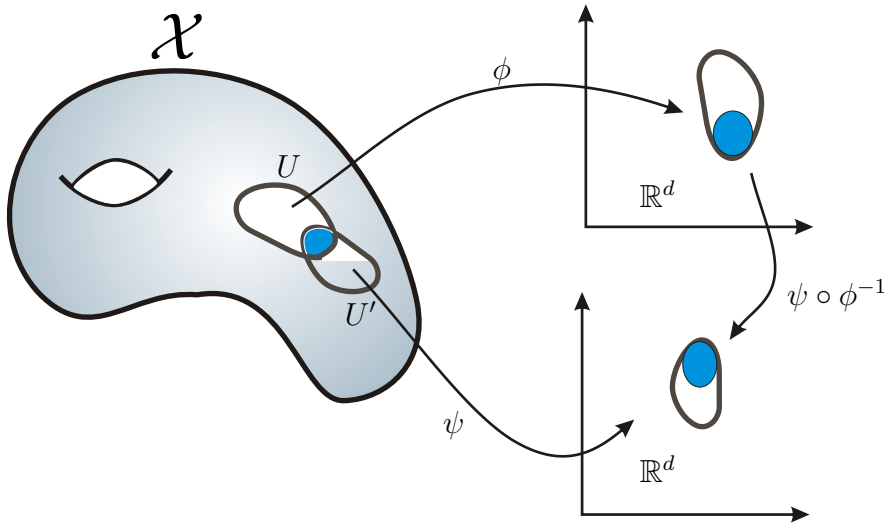


Figure 2.3: Change of coordinates $\psi \circ \phi^{-1}$ or transition map.

not sufficient to provide an entire description (*e.g.*atlas) of the manifold. Then several charts are used to cover the manifold. Of course, we seek smoothness between the charts. And we say two charts (U, ϕ) and (U', ψ) with a non-empty intersection $U \cap U' \neq \emptyset$ are compatible when the map taking a point

from $\phi(U \cap U')$ into $\psi(U \cap U')$ is a homeomorphism. The map is the composition $\psi \circ \phi^{-1}$ and corresponds to a change of coordinates (c.f. Figure 2.3). A family of charts covering the entire manifold $\mathcal{X}$ is called an atlas

Definition 12. A smooth d-dimensional manifold is a set $\mathcal{X}$ and a family of charts (the atlas) (U_α, ϕ_α) with $U_\alpha \subset \mathcal{X}$ and $\phi_\alpha : \mathcal{X} \mapsto \mathbb{R}^d$ such that:

(i) $\bigcup_\alpha U_\alpha = \mathcal{X}$

(ii) for any α, β with $U_\alpha \cap U_\beta = U' \neq \emptyset$, the sets $\phi_\alpha(U')$ and $\phi_\beta(W)$ are open sets in $\mathbb{R}^n$ and the composition $\phi_\beta \circ \phi_\alpha^{-1}$ is differentiable.

We shall look at the example of a circle $\mathbb{S}^1$ as a subset of $\mathbb{R}^2$ (c.f. Figure 2.4)

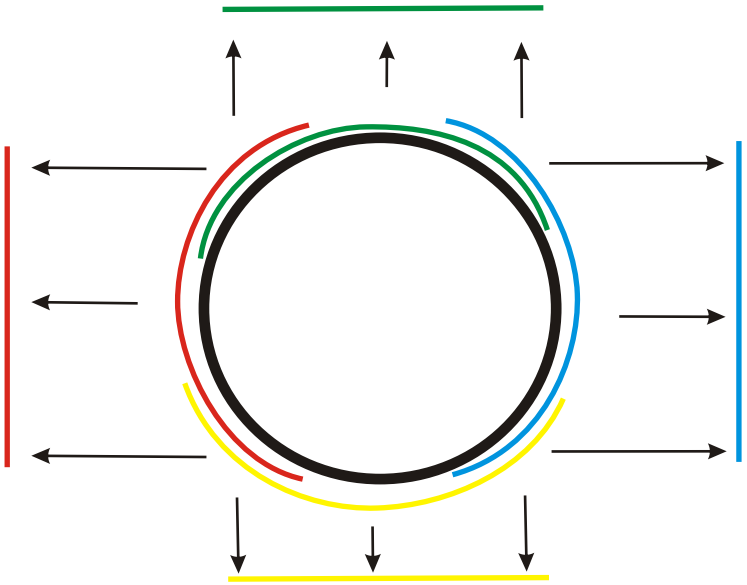


Figure 2.4: Example of an atlas for the circle with four coordinate charts. Each chart has an associated mapping (indicated by arrows) from the circle to an open interval. Assuming that the transition maps are smooth, the four charts cover the whole circle and therefore provide an atlas.

Each of the four charts is mapped to an open interval, and together they cover the whole $\mathbb{S}^1$and therefore is an atlas on $\mathbb{S}^1$. Assuming that composition of the charts are differentiable we have a smooth atlas and thus a smooth manifold. An explicit parametrization of the circle is provided through trigonometric functions as

$$\begin{aligned}
&x_1(t) = a\cos(t) \text{ and } y_1(t) = a\sin(t), t \in (0, \pi)\,,\\
&x_2(t) = a\cos(t) \text{ and } y_2(t) = a\sin(t), t \in (\pi, 2\pi)\,,\\
&x_3(t) = a\cos(t) \text{ and } y_3(t) = a\sin(t), t \in \left(\frac{1}{2}\pi, \frac{2}{3}\pi\right)\,,\\
&x_4(t) = a\cos(t) \text{ and } y_4(t) = a\sin(t), t \in \left(\frac{2}{3}\pi, \frac{1}{2}\pi\right)\,.
\end{aligned}$$

where a is the radius of the circle. This is an example of a 1-dimensional surface embedded in $\mathbb{R}^2$. The embedding space provides the coordinates in order to give every point an exact coordinate.

2.3.2 Tangent Space

The idea of linear approximation of a surface in vector calculus translates into the concept of tangent space in differential geometry. Given a manifold $\mathcal{X}$ embedded in $\mathbb{R}^d$, one can associate a linear subspace of $\mathbb{R}^d$ to each point $x \in \mathcal{X}$. This linear subspace is called the tangent space and is the best linear approximation within a small neighborhood around x. In the literature, tangent vectors are introduced through the directional derivative. Consider a differentiable curve $\Gamma : (-\varepsilon, \varepsilon) \mapsto \mathcal{X} \in \mathbb{R}^d$ centered at x, $\Gamma(0) = x$:

$$\Gamma(t) = (x^1(t), \ldots, x^d(t)), t \in (-\varepsilon, \varepsilon).$$

$(x^1(t), \ldots, x^d(t))$ are the coordinate functions of the curve Γ. Then for all real-valued smooth functions $f : \mathcal{X} \mapsto \mathbb{R}$ in which the composition $f \circ \Gamma$ is smooth the directional derivative writes:

$$\frac{d(f \circ \Gamma)}{dt}|_{t=0} = \Sigma_{i=1}^{d} \frac{\partial f}{\partial x^i}|_{t=0} \frac{dx^i}{dt}|_{t=0} = (\Sigma_{i=1}^{d} \dot{x}^i(0) \frac{\partial}{\partial x^i}) f. \tag{2.7}$$

Geometrically (c.f. Figure 2.5), a tangent vector at x is nothing but the tangent vector to any parametrized curve $\Gamma : (\varepsilon, \varepsilon) \mapsto \mathcal{X}$ at $t = 0$. If, we have a local parametrization (U, ϕ), we can express the tangent vector in the local chart by

$$\dot{\Gamma}(0) = \sum_{i=1}^{d} \dot{x}^i(0) (\frac{\partial}{\partial x^i})_{|0}. \tag{2.8}$$

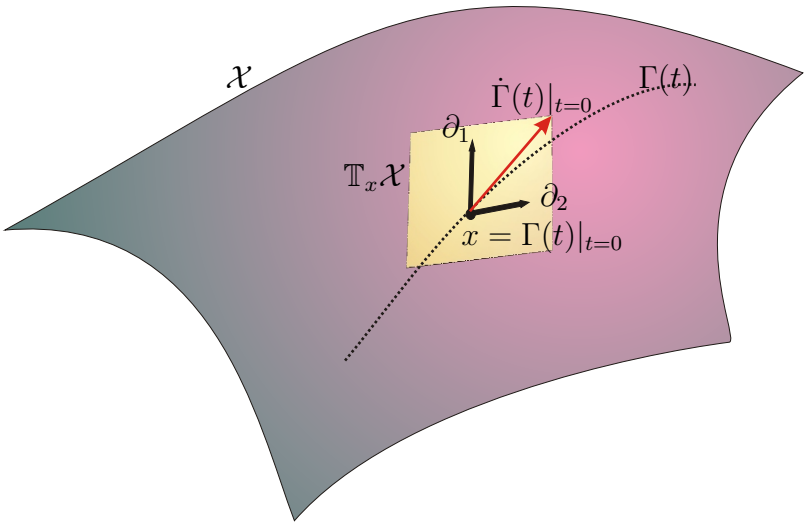


Figure 2.5: The tangent space $\mathbb{T}_x\mathcal{X}$ of $\mathcal{X}$ at x. The basis vectors ∂_1, ∂_2 span the tangent space $\mathbb{T}_x\mathcal{X}$ with $\dot{\Gamma}(t)|_{t=0} \in \mathbb{T}_x\mathcal{X}$.

$(\frac{\partial}{\partial x^i})_{|0}$ is the infinitesimal displacement along the i-th coordinate curve in the local chart. We see that the local parametrization determines a basis $(\frac{\partial}{\partial x^1})_{|0}, \ldots, (\frac{\partial}{\partial x^d})_{|0}$ of the tangent space $\mathbb{T}_x\mathcal{X}$ which is independent with respect to the parametrization. In general, tangent vectors are defined through curves on $\mathcal{X}$. The set of all tangent vectors at x defines the tangent space $\mathbb{T}_x\mathcal{X}$. It is also clear from the definition that there is an equivalence relation between all the curves at x sharing the same tangent vector.

Example 1

Consider the 3 dimensional sphere embedded in $\mathbb{R}^3$ and consider the parametrization $\phi : \left[-\frac{\Pi}{2}, \frac{\Pi}{2}\right] \times [0, 2\Pi)$:

$$\begin{aligned} \phi_1(x^1, x^2) &= \cos x^2 \cos x^1 \\ \phi_2(x^1, x^2) &= \sin x^2 \cos x^1 \\ \phi_3(x^1, x^2) &= \sin x^1. \end{aligned}$$

For a point $x' = (x^1, x^2)$ in the parameter domain the corresponding point on

the sphere is given by $x = \phi(u)$. Then the basis vectors of the tangent plane of the sphere are:

$$\begin{aligned}\partial_1 &= (-\cos x^2 \sin x^1, -\sin x^2 \sin x^1, \cos x^1)\\ \partial_2 &= (-\sin x^2 \cos x^1, \cos x^2 \cos x^1, 0).\end{aligned}$$

To see the effect on the surface of small displacements in the parameter domain, we change the point x' by an infinitesimal quantity dx:

$$\phi(x' + dx') = x' + \partial_1 dx'^1 + \partial_2 dx'^2 = x' + Jdx'$$

J is the Jacobian of the parametrization ϕ. Then the displacement on the surface is given by $dx = Jdx'$. As the infinitesimal quantity dx is an element of the tangent space, we can measure its length using the squared Euclidean norm

$$\|dx\|^2 = \|Jdx'\|^2 = dx'^T J^T J dx' = dx'^T G dx' = \langle dx, dx\rangle_x .$$

G is a symmetric matrix with elements $g_{ij} = \langle \partial_i, \partial_j \rangle$ depending on the chosen parametrization. Each g_{ij} measures the correlation between the basis vectors of the tangent space. As we required smoothness for the map ϕ, we deduce that the determinant of G is strictly bigger than 0 and hence G is a symmetric positive definite matrix. Then the metric matrix G is commonly referred to as Riemannian Metric or First Fundamental Form. Applied to our sphere example, the Riemannian metric is

$$G = \begin{bmatrix}\langle \partial_1, \partial_1\rangle & \langle \partial_1, \partial_2\rangle \\ \langle \partial_2, \partial_2\rangle & \langle \partial_2, \partial_1\rangle\end{bmatrix} = \begin{bmatrix}1 & 0 \\ 0 & \cos^2 x^1\end{bmatrix}$$

So when we take a small step dx^2 along x^2 the image of ϕ induces a displacement by $\cos^2 x^1$ on the sphere.

2.3.3 Riemannian Manifold

In the previous section, we demonstrated the existence of a metric on smooth manifolds. To be more precise, we came to the result that we can define an inner product between two tangent vectors of $\mathbb{T}_x\mathcal{X}$ which smoothly varies over the manifold. Further, we said that when the metric matrix is positive definite the manifold is a Riemannian manifold

Definition 13. A Riemannian metric on a differential manifold $\mathcal{X}$ is a symmetric, bilinear and positive-definite form $\langle ., . \rangle_x$ on $\mathbb{T}_x\mathcal{X}$ varying continuously

over $\mathcal{X}$. The assignment of an inner product to each point $x \in \mathcal{X}$ is called a metric tensor $g_{ij}(x)$. Written in local coordinates on the manifold $\mathcal{X}$ the vectors $\left\{\frac{\partial}{\partial x^1}, \ldots, \frac{\partial}{\partial x^d}\right\}$ define a basis of tangent vectors at each $x \in \mathcal{X}$. In this coordinate system the metric tensor at x writes $g_{ij}(x) := (\frac{\partial}{\partial x^i}, \ldots, \frac{\partial}{\partial x^1})$.

The metric g_{ij} allows to measure lengths, areas or angles. These quantities depend on the geometry of the surface and therefore on the metric tensor. So using the definition of the metric tensor, we can measure the length of a smooth path $\Gamma : [a, b] \mapsto \mathcal{X}$ on $\mathcal{X}$ as

$$L(\Gamma) = \int_a^b \sqrt{\left\langle \dot{\Gamma}(t), \dot{\Gamma}(t) \right\rangle}_{\Gamma(t)}$$

As in mechanics the vector $\dot{\Gamma}(t)$ is often interpreted as the instantaneous velocity of a particle at $\Gamma(t)$. The Riemannian metric $\sqrt{\left\langle \dot{\Gamma}(t), \dot{\Gamma}(t) \right\rangle}_{\Gamma(t)}$ is a local measure of length depending on the geometry of the surface at $\Gamma(t)$. It is nothing than a weighted Euclidean norm. In Riemannain geometry, it is interesting whether Equation 2.6 (c.f. section 2.2) $L(\Gamma)$ has extrema. A *geodesic* is a path Γ which is a local minimizer of L. From a theoretical point of view, an important question is if every pair of points in $\mathcal{X}$ has a connecting geodesic. A metric space with the property of having a connecting geodesic for each pair of points is said to be complete. The Hopf-Rinow theorem tells us that every connected and compact Riemannian manifold is a complete metric space. If we have a complete space, we can introduce the geodesic metric as

$$d_{\mathcal{X}}(x_1, x_2) = \min_{\Gamma} \{L(\Gamma) \text{ such that } \Gamma : [a, b] \mapsto \mathcal{X}, \Gamma(a) = x_1, \gamma(b) = x2\}.$$

Riemann normal coordinates may be thought of as a generalization of Cartesian coordinates from Euclidean space to any manifold (which should be at least twice differentiable) with affine connection. (Including Riemannian manifolds as a special case, of course!)

To define a system of Riemann normal coordinates, one needs to pick a point P on the manifold which will serve as origin and a basis for the tangent space at P . Suppose that the manifold is d dimensional. To any d -tuple of real numbers $(x^1, \ldots x^n)$, we shall assign a point Q of the manifold by the following procedure:

Let v be the vector whose components with respect to the basis chosen for the tangent space at P are $x^1, \ldots x^n$. There exists a unique affinely-parameterized geodesic $C(t)$ such that $C(0) = P$ and $[dC(t)/dt]_{t=0} = v$. Set $Q = C(1)$. Then Q is defined to be the point of which Riemann normal coordinates are $(x^1, \ldots x^n)$.

2.3.4 Normal coordinates

Normal coordinates are introduced as they are used for the convergence proof in section 3.2.4. To each point x on $\mathcal{X}$ we have $\mathbb{T}_x\mathcal{X}$, consisting of a Euclidean space tangential to $\mathcal{X}$ at x. As seen before, a class of curves with minimal length defined on a manifold $\mathcal{X}$ are the geodesics, which define a metric $d_{\mathcal{X}}(x, y)$ on the manifold derived from the length of a geodesic passing through x and y. The Riemannian exponential map, $\exp(v) \in \mathcal{X}, v \in \mathbb{T}_x\mathcal{X}$, is a function which maps points in the tangent space at x, to points on $\mathcal{X}$. If the geodesic is unique, this mapping is one-to-one in a neighborhood of x and its inverse is the log map. The set of points on $\mathcal{X}$ for which there exists more than one shortest path from x is called the cut locus of x. The cut locus of a point on a sphere is for instance its antipodal point. Given a point p and an orthonormal basis ∂_i for the tangent space $\mathbb{T}_p\mathcal{X}$, a Riemannian normal coordinate system is provided by the exponential mapping. A point $x \in \mathcal{X}$ has coordinates $(x^1, \ldots, x^d)$ if $x = \exp(\sum_1^d x^i \partial_i)$. In this simple way, we can compute local coordinates in the tangent plane of any point within a small neighborhood of p.

2.3.5 Laplace Beltrami operator

Let f be a real valued twice differentiable function defined on a Riemannian manifold $\mathcal{X}$. The *Laplace-Beltrami* operator Δ is given by

$$\Delta_{\mathcal{X}} f := div(\nabla_{\mathcal{X}} f)$$

with $\nabla_{\mathcal{X}} f$ the gradient of f and div the divergence of a vector field on a manifold [Do carmo 1992] (the divergence is an operator that measures the magnitude of a vector field's source or sink at a given point). The Laplace-Beltrami operator is a linear differential operator and can be calculated using a local parametrization $(\mathbb{R}^d, \psi)$ with $\psi : \mathbb{R}^d \mapsto \mathbb{R}^D$. Given the metric tensor G of $\mathcal{X}$, the Laplace-Beltrami operator on $\mathcal{X}$ writes as

$$\Delta_{\mathcal{X}} f = \frac{1}{\sqrt{|G|}} \sum_{i,j} \partial_i (G_{ij}^{-1} \sqrt{|G|} \partial_j f).$$

$|G|$ is the determinant of G and ∂_i, ∂_j are the basis vector of the tangent space. In the case where $\mathcal{X} \subset \mathbb{R}^2$, the metric tensor G simplifies to the identity and the Laplace-Beltrami reduces to the Laplace operator in $\mathbb{R}^2$:

$$\Delta f = \frac{\partial^2 f}{(\partial x^1)^2} + \frac{\partial^2 f}{(\partial x^2)^2}.$$

The spectrum of the Laplacian is the set of eigenfunctions and associated eigenvalues solving the Helmholtz equation

$$\Delta_{\mathcal{X}} f = -\lambda f.$$

The solution is an infinite number of eigenvalues λ_i (with $i = 0, \ldots, +\infty$) and eigenfunctions f_i (with $i = 0, \ldots, +\infty$). In the case of a closed surface without boundary, the first eigenvalue λ_0 is always equal to zero and the associated eigenfunction f_0 is a constant function. We will encounter this property in section 3.2.5. Last, let's mention a few properties of the Laplace-Beltrami operator as mentioned in the book of Reuter [Reuter 2006]:

1. The spectrum depends only on the metric tensor and is invariant under isometric mappings(c.f. 2.4)

2. The spectrum of the Laplace-Beltrami operator of d-dimensional manifolds at different scales can be compared by normalizing the eigenvalues appropriately.

3. A change of the surface's shape results in continuous change of the spectrum.

4. The spectrum does not characterize the shape completely, since some non-isometric manifolds with the same spectrum exist. Nevertheless these cases appear to be very rare.

5. A substantial amount of geometrical and topological information is known to be contained in the spectrum. As a consequence of the high dimensionality of the eigenspectrum, cropping the spectrum is unavoidable and consequently induces a loss of information. But nevertheless, the first few eigenvalues contain important information.

2.4 Mappings between metric spaces

In this thesis we will be confronted with mappings between metric spaces. For example, we will consider the problem of matching surfaces in pairs(c.f. chapter 5), which are isometrically embedded into $\mathbb{R}^3$. So the problem is to find a mapping between the two surfaces so that the metric is unchanged. The characterization of different types of mappings between two manifolds $\mathcal{X}$ and $\mathcal{Y}$ will be the purpose of this section.

Definition 14. Homeomorphism. Let $\mathcal{X}$ and $\mathcal{Y}$ be two manifolds. A map $f : \mathcal{X} \mapsto \mathcal{Y}$ is called a homeomorphism if it is one-to-one, continuous and has a continuous inverse f^{-1}. If such a function exists then $\mathcal{X}$ and $\mathcal{Y}$ are homeomorphic.

For example, the unit disc in $\mathbb{R}^2$ is homeomorphic to the unit square in $\mathbb{R}^2$. The two-dimensional torus is homeomorphic to the product space $\mathbb{S}^1 \times \mathbb{S}^1$. Any sphere $\mathbb{S}^2$ in $\mathbb{R}^3$ is not homeomorphic to $\mathbb{R}^2$.

Definition 15. Diffeomorphism. Let $\mathcal{X}$ and $\mathcal{Y}$ be two manifolds. A map $f : \mathcal{X} \mapsto \mathcal{Y}$ is called a diffeomorphism if f and its inverse f^{-1} are differentiable. The set of all diffeomorphisms between $\mathcal{X}$ and $\mathcal{Y}$ is denoted by $\mathrm{Diff}(\mathcal{X}, \mathcal{Y})$. If such a function exists then $\mathcal{X}$ and $\mathcal{Y}$ are diffeomorphic.

From now on we assume all mappings between surfaces to be diffeomorphisms. We now impose even more regularity on the mappings and introduce the group of isometries. Let $\mathcal{X}$ and $\mathcal{Y}$ be two manifolds. A map $f : \mathcal{X} \mapsto \mathcal{Y}$ is called an isometry if the length of any rectifiable curve on $\mathcal{X}$ is the same as that of its image on $\mathcal{Y}$. If such a map exists then $\mathcal{X}$ and $\mathcal{Y}$ are said to be isometric.

Also, we can define an isometric map f as a map between two metric spaces $\mathcal{X} \mapsto \mathcal{Y}$ such that $d_{\mathcal{X}}(x_1, x_2) = d_{\mathcal{Y}}(f(x_1), f(x_2))$. If this property holds for any x_1, x_2 then f is an isometric mapping. We will introduce a further class of mappings which are almost isometric mappings. This is due to the fact that when dealing with real world data such as in chapter 5 the data has some inherent error caused by imperfect measurements or weak processing methods. Therefore we relax the property of exact isometry to the notion of ε-isometry. An ε-isometry is a mapping $f : \mathcal{X} \mapsto \mathcal{Y}$ such that for some $\varepsilon \geq 0$ the error is bounded $|d_{\mathcal{Y}}(f(x_1), f(x_2)) - d_{\mathcal{X}}(x_1, x_2)| \leq \varepsilon$ for all $x_1, x_2 \in \mathcal{X}$. The distortion of an ε-isometry is measured as

$$dis f = \sup_{x_1, x_2 \in \mathcal{X}} |d_{\mathcal{Y}}(f(x_1), f(x_2)) - d_{\mathcal{X}}(x_1, x_2)| .$$

Chapter 3

Manifold learning

Contents

3.1 Linear Methods 39

3.1.1 Principle Component Analysis 39

3.1.2 Multidimensional Scaling 41

3.2 Nonlinear methods 44

3.2.1 Graph based methods 44

3.2.2 Isomap 47

3.2.3 Locally Linear Embedding (LLE) 49

3.2.4 Hessian Eigenmaps 52

3.2.5 Laplacian Eigenmaps 55

3.2.6 Diffusion Maps 57

3.2.7 Summary of metric learning algorithms 60

Overview

Among the most common applications of machine learning we want to mention data mining and computer vision. Mostly these applications involve estimating a function for the purpose of classification or recognition of datasets. Common datasets encountered in computer vision often consist of a large number of samples each of which are themselves of high dimension. Either the datum is treated directly or more complex features are extracted from the datum for further processing. Nevertheless, it might be the entire datum or a set of computed features for which the dimensionality of problems usually remain high slowing down the processing considerably or sometimes making the treatment even intractable. In order to lessen the computational burden an intuitive approach is to reduce the dimensionality of the data. Being able to reduce the dimensionality requires understanding the structure in datasets. This amounts to finding the latent variables in the datasets. In statistics, latent variables (as opposed to observable variables), are variables that are not directly observed but are inferred (through a mathematical model) from other variables (the dataset) which are observed and directly measured. They are also sometimes known as hidden variables, model parameters, hypothetical variables or hypothetical constructs. Once such parameters are identified, expressing the original data in terms of these variables allows to reduce the dimensionality of the data.

There are many approaches to dimensionality reduction based on a variety of assumptions and used in a variety of contexts. We will focus on an approach initiated recently based on the observation that high-dimensional data is often much simpler than the dimensionality would indicate. In particular, a given high-dimensional data set may contain many features that are all measurements of the same underlying cause. The stereotypical problem of such phenomena are for example pictures of the same object with a single moving light source. The images taken as vectors in some high-dimensional space seem very complex. A simplified representation of the data is desirable. And this seems reasonable as the governing parameters are identifiable. The parameters are just the rotation angles of the lightsource. (c.f Figure 1.2). A geometer would say that the data is parametrized and lies on a two-dimensional manifold embedded in a high-dimensional space. This is the geometric point of view in statistical learning. This is also the point of view we will take in this thesis when deriving the learning algorithms and developing applications. Nevertheless, it is to be noted that most of the explanations given in this chapter can be rephrased in a statistical or probabilistic manner which have been studied in a considerable number of

papers.

We are given a set of N data points $\{x_i\} \in \mathbb{R}^D$ (please note that we will change this assumption in chapter 4 and work with more general spaces than $\mathbb{R}^D$) and assume they approximately lie on a smooth manifold $\mathcal{X}$. The intrinsic dimension of $\mathcal{X}$ is $d = dim(\mathcal{X})$ with $d << D$. In other words, we hypothesize that $\mathcal{X}$ is the image of some parameter domain $\mathcal{Y} \subset \mathbb{R}^d$ under a smooth mapping $\Psi : \mathcal{Y} \mapsto \mathbb{R}^D$. The idea of Manifold Learning is to find the low-dimensional coordinates $y_i \in \mathcal{Y}$ of the corresponding points $x_i \in \mathcal{X}$. In the sequel the matrices $\mathbf{X} = [x_1, \ldots, x_N]$ and $\mathbf{Y} = [y_1, \ldots, y_N]$ are matrices of size $D \times N$ and $d \times N$ respectively.
Manifold Learning can be mainly divided into linear and non linear methods. A further distinction within non-linear methods is to divide the methods into purely global methods and methods recovering global structure from local information only. We start with linear methods reviewing the famous Principal Component Analysis and Multidimensional Scaling algorithm. We then move on to the non-linear world to show the extension of Multidimensional Scaling to the non linear case. At last, we examine the most recent local and global methods while taking into account Locally Linear Embedding, Hessian Eigenmaps. Our panoramic view of manifold learning methods ends with Laplacian Eigenmaps and Diffusion Maps. These methods employ Laplacian regularization to the problem providing a connection to harmonic analysis and opening the door for sound theoretical results.

3.1 Linear Methods

3.1.1 Principle Component Analysis

Principal Component Analysis (PCA) is a standard algorithm in multivariate statistics.The input is vectorial data and the output of the algorithm is a new orthogonal coordinate system. The representation of the original data set in the new coordinate system captures most of the variance of the data set. In summary the problem in PCA is to find a d-dimensional subspace of $\mathbb{R}^D$ that best approximates the data in the least square sense. The new coordinates of the original data $\mathbf{X} = \{x_1, \ldots, x_n\}$ are computed through orthogonal projection onto the subspace. Let V' be a d-dimensional subspace of $\mathbb{R}^D$ and let $\mathbf{u}_1, \ldots, \mathbf{u}_D$ be an orthonormal basis of $\mathbb{R}^D$ such that $\mathbf{u}_1, \ldots, \mathbf{u}_d$ is a basis of

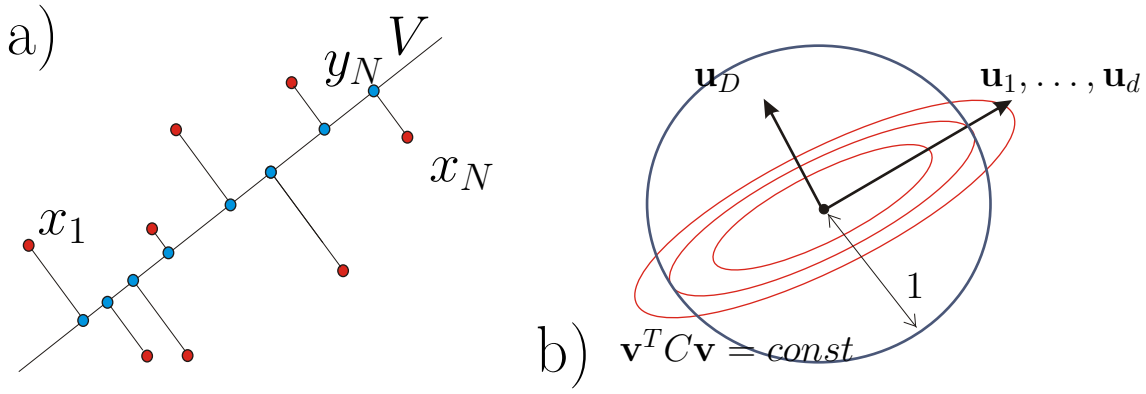


Figure 3.1: a) 1D manifold and the distances between the original points(red) and their projection on the manifold. b) is a visualization of the geometry and constraints of the optimization problem.

V. We then approximate each data point by

$$y_n = \sum_{i=1}^{d} \alpha_{n,i} \mathbf{u}_i + \sum_{i=d+1}^{D} b_i \mathbf{u}_i.$$

In order to recover the vectors $\mathbf{u}_1, \ldots, \mathbf{u}_d$ the following energy function is proposed

$$E = \frac{1}{N} \sum_{n=1}^{N} \|x_n - y_n\|^2 .$$

This is the well known least square estimate. It is the mean over the squared approximation error for each data point. Thus minimizing E with respect to $\alpha_{n,i}$ and b_i leads to minimize the approximation error

$$x_n - y_n = \sum_{i=d+1}^{D} \left\{ (x_n - \overline{x})^T \mathbf{u}_i \right\} \mathbf{u}_i.$$

This equation shows that the best affine subspace given V is passing through $\overline{x}$ and y_n is the orthogonal projection of x_n onto V (c.f. Figure 3.1). As a consequence E only depends on $\mathbf{u}_i$ with $i = d+1, \ldots, D$) and the energy is rewritten as a double sum in the following manner

$$E = \frac{1}{N} \sum_{n=1}^{N} \sum_{i=d+1}^{D} (x_n^T \mathbf{u}_i - \overline{x}^T \mathbf{u}_i)^2 = \sum_{i=d+1}^{D} \mathbf{u}_i^T \mathbf{C} \mathbf{u}_i.$$

Here $\mathbf{C}$ stands for the well-known covariance matrix $\mathbf{C} = \frac{1}{N}\sum_{n=1}^{N}(x_n - \overline{x})(x_n - \overline{x})^T$ and holds the values for mutual projection between points in the data set. So far, we did not propose any solution to the optimization of the energy E. It turns out that the global minimum of E is obtained for the space spanned by the d eigenvectors of $\mathbf{C}$ corresponding to the d largest eigenvalues. In order to see this, we minimize our energy with $d = D - 1$ (for simplicity)

$$E = \mathbf{u}_D^T \mathbf{C} \mathbf{u}_D \text{ under the constraint } \mathbf{u}_D^T \mathbf{u}_D = 1.$$

Such a constrained minimization is done using Lagrange multipliers which then writes as

$$\mathbf{u}_D^T \mathbf{C} \mathbf{u}_D + \lambda(1 - \mathbf{u}_D^T \mathbf{u}_D) = 0.$$

Taking the derivative with respect to $\mathbf{u}_D$: $2\mathbf{C}\mathbf{u}_D - 2\lambda\mathbf{u}_D$ and equating to 0 yields $\mathbf{C}\mathbf{u}_D = \lambda\mathbf{u}_D$ which is the standard form of an eigenvalue problem. The geometry of the minimization problem is illustrated in Figure 3.1b). The level sets of the quadratic form $\mathbf{v}^T\mathbf{C}\mathbf{v}$ are drawn in red. The constraint on the solution $\mathbf{u}_D$ is drawn by a circle of radius 1 reducing the possible solutions. As the reader might expect the solution $\mathbf{u}_1, \ldots, \mathbf{u}_d$ is orthogonal to $\mathbf{u}_D$.
As a final remark, we want to emphasize the two possible interpretations of this problem as mentioned at the beginning of this chapter. The solution computed using PCA is a d-dimensional subspace of $\mathbb{R}^D$ which best approximates $\mathbf{X}$ in a least square sense. The projection of $\mathbf{X}$ onto the subspace minimizes the distances between original and projected data under the L_2 norm(c.f. Figure 3.1a)). We considered the geometry point of view although a more statistical approach is to find a d-dimensional subspace of $\mathbb{R}^D$ onto which the projected data has maximum variance. Therefore this method is also referred to as a maximum variance based method. Lastly, more probabilistic formulations of the problem which lead to the same solution [Etyngier 2008] also exist.

3.1.2 Multidimensional Scaling

Multidimensional Scaling (MDS) corresponds exactly to PCA. The subtle difference to PCA is that MDS is coordinate-free. The only requirement is the squared distance between pairs of points (x_i, x_j) in the data set $\mathbf{X}$. The pairwise distance values are stored in square matrix $\mathbf{D}$ of size $N \times N$. The problem which can be solved with MDS is to find d-dimensional Euclidean coordinates for each data sample so that the pairwise distance of their Euclidean coordinates match the original pairwise distance as closely as possible. The literature proposes several cost functions for this task and a variety of algorithms solving the minimization problem. Nevertheless, we will focus on classical MDS as

we will need this formulation to introduce Isomap in section 3.2.2. The MDS algorithm, shown in Figure 3.2, is based on the duality between the space of Euclidean distance matrices $\mathbf{D}_{ij} = d_{\mathbb{X}}(x_i, x_j)$ of size $N \times N$and the space of Gram matrices $\mathbf{G} = \mathbf{X}\mathbf{X}^T$ (inner-product matrices) of size $N \times N$. The relation between $\mathbf{D}$ and $\mathbf{G}$ is given by

$$\mathbf{G} = -\frac{1}{2}\mathbf{J}\mathbf{D}\mathbf{J} \text{ where } \mathbf{J} = \mathbf{Id}_N - \frac{1}{N}\mathbb{1}\mathbb{1}^T$$

The objective is to find $\mathbf{Y} = \{y_i, \dots, y_N\} \in \mathbb{R}^d$ minimizing the energy

$$E(\mathbf{D}_X, \mathbf{D}_Y) = \|\mathbf{G}_X - \mathbf{G}_Y\|_2^2 = \left\|\frac{1}{2}\mathbf{J}(\mathbf{D}_X - \mathbf{D}_Y)\mathbf{J}\right\|_2^2$$

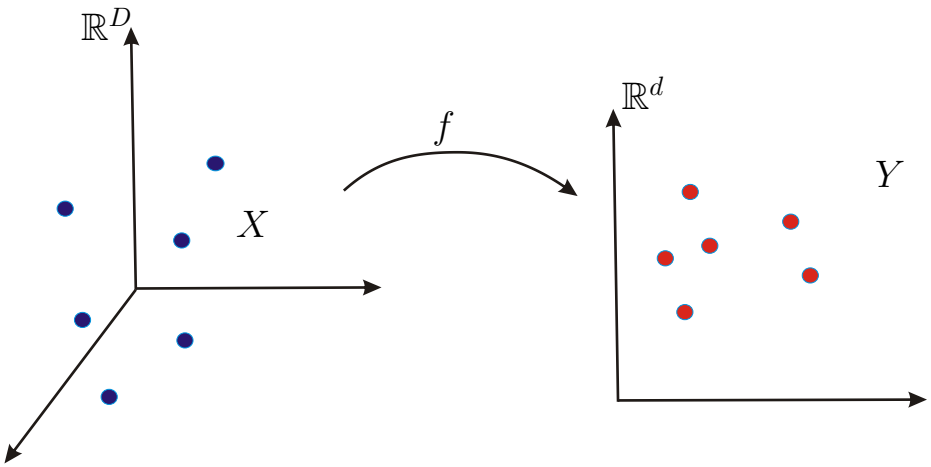


Figure 3.2: The function f maps $\mathbf{X}$ from $\mathbb{R}^D$ to $\mathbf{Y}$ in $\mathbb{R}^d$ so that pairwise distances are as close as possible.

Like PCA the solution is given by the spectrum of the matrix $\mathbf{G}_X$ with eigenvalues $\lambda_1 \geq \cdots \geq \lambda_N \geq 0$ and orthogonal eigenvectors $\{\mathbf{u}_1, \dots, \mathbf{u}_N\} \in \mathbb{R}^n$. Then $\mathbf{Y} \in \mathbb{R}^d$ minimizing E is given by the columns of the $d \times N$ matrix

$$\mathbf{Y} = \begin{pmatrix} \sqrt{\lambda_1}\mathbf{u}_1^T \\ \vdots \\ \sqrt{\lambda_d}\mathbf{u}_d^T \end{pmatrix}$$

Again we want to show how to obtain the solution of this minimization prob-

lem. Therefore we rewrite the minimization problem as

$$\begin{aligned}
\min_{\mathbf{Y}} \|\mathbf{G}_X - \mathbf{G}_Y\|_2^2 &= \min_{Y} \left\|\mathbf{X}\mathbf{X}^T - \mathbf{Y}\mathbf{Y}^T\right\|_2^2 && (3.1)\\
&= \min_{Y} \sum_{i=1}^{N}\sum_{j=1}^{N}(x_i^T x_j - y_i^T y_j)^2 && (3.2)\\
&= \min_{\mathbf{Y}} Tr((\mathbf{X}\mathbf{X}^T - \mathbf{Y}\mathbf{Y}^T)^2). && (3.3)
\end{aligned}$$

As $\mathbf{X}\mathbf{X}^T$ and $\mathbf{Y}\mathbf{Y}^T$ are positive semidefinite, we have the following spectral property $\mathbf{X}\mathbf{X}^T = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T$ and equally $\mathbf{Y}\mathbf{Y}^T = \mathbf{W}\mathbf{\Lambda}'\mathbf{W}^T$. $\mathbf{V}$ and $\mathbf{W}$ being the corresponding eigenvectors having the property $\mathbf{V}\mathbf{V}^T = \mathbf{W}\mathbf{W}^T = \mathbf{Id}_N$ and their associated eigenvalues $\mathbf{\Lambda} = Diag(\lambda_1, \ldots, \lambda_N)$ and $\mathbf{\Lambda}' = Diag(\lambda_1', \ldots, \lambda_d', 0, \ldots, 0)$ in descending order. Since the solution $\mathbf{Y}$ is d-dimensional the remaining eigenvalues in $\mathbf{\Lambda}'$ equal 0. Then we rephrase the optimization problems in terms of $\mathbf{V}$ and $\mathbf{W}$

$$\begin{aligned}
\min_{Y} Tr((\mathbf{X}\mathbf{X}^T - \mathbf{Y}\mathbf{Y}^T)^2) &= \min_{\mathbf{W},\mathbf{\Lambda}'} Tr(\mathbf{\Lambda} - \mathbf{V}^T\mathbf{W}\mathbf{\Lambda}'\mathbf{W}^T\mathbf{V})^2 \text{ (using } Tr(\mathbf{W}\mathbf{V}) = Tr(\mathbf{V}\mathbf{W}))\\
&= \min_{\mathbf{Q},\mathbf{\Lambda}'} Tr(\mathbf{\Lambda} - \mathbf{Q}\mathbf{\Lambda}'\mathbf{Q}^T) \text{ with } = \mathbf{V}^T\mathbf{W}\\
&= \min_{\mathbf{Q},\mathbf{\Lambda}'} Tr(\mathbf{\Lambda}^2) + Tr(\mathbf{Q}\mathbf{\Lambda}'\mathbf{Q}^T\mathbf{Q}\mathbf{\Lambda}'\mathbf{Q}^T) - 2Tr(\mathbf{\Lambda}\mathbf{Q}\mathbf{\Lambda}'\mathbf{Q}^T)\\
&= \min_{\mathbf{\Lambda}} Tr(\mathbf{\Lambda}^2 + \mathbf{\Lambda}'^2 - 2\mathbf{\Lambda}\mathbf{\Lambda}')\\
&= \min_{Y} Tr(\mathbf{\Lambda} - \mathbf{\Lambda}').
\end{aligned}$$

The minimum is obtained for $\mathbf{\Lambda}' = Diag(\lambda_1', \ldots, \lambda_d', 0, \ldots, 0)$ and therefore we can set $\mathbf{Q} = \mathbf{V}^T\mathbf{W} = \mathbf{Id}_N$.

As mentioned in the introduction MDS requires only the pairwise squared distances between the elements in $\mathbf{X}$ to compute the coordinates of the points $\mathbf{Y}$. In order to apply MDS on the distance matrix $\mathbf{D}$ one has to perform double centering on $\mathbf{D}$: $\mathbf{G} = -\frac{1}{2}\mathbf{J}\mathbf{D}\mathbf{J}$. Nevertheless, some issues remain when $\mathbf{D}$ is not computed from a data set $\in \mathbb{R}^D$. In this case $\mathbf{G}$ might have negative eigenvalues which reflects the fact that $\mathbf{D}$ is non Euclidean. Eckert and Young [Eckart 1936] show that the d-dimensional embedding computed by MDS has a Gram matrix which is the best approximation to $\mathbf{G} = -\frac{1}{2}\mathbf{J}\mathbf{D}\mathbf{J}$ and provide the following bound:

$$\left\|\mathbf{Y}_{MDS}\mathbf{Y}_{MDS}^T - \mathbf{G}\right\| \leq \left\|\mathbf{Y}\mathbf{Y}^T - \mathbf{G}\right\|$$

for any $\mathbf{Y}$ in the space of positive definite matrices and $\mathbf{Y}_{MDS}$ being the embedding obtained with MDS.
One can generalize MDS to geodesic distances between point samples from

a non-linear manifold. This yields a non-convex energy function which is tedious to optimize. In the next section we will leave the linear world to discuss non-linear problems and provide an overview of non-linear manifold methods. Recently, MDS was extended to Bregman divergence[Lai 2009].

3.2 Nonlinear methods

Linear methods such as PCA and MDS produce bad results when the data is sampled from non linear manifolds. In Figure 1.1 (page 7) for example, the data is sampled from a two-dimensional nonlinear manifold embedded in the three-dimensional Euclidean space. All methods we review in this section assume that the data is distributed along a d-dimensional submanifold $\mathcal{X}$ embedded in $\mathbb{R}^D$.

3.2.1 Graph based methods

Non-linear dimensionality reduction often relies on adjacency graphs. These graphs are build on the input data according to certain rules which should reflect the structure in the neighborhood of some points. We will review the two most prominent methods. Once the graphs are computed, methods such as MDS need some pairwise distances between the nodes of the graph. Therefore we also quickly review the computation of distance between the nodes in the graph and analyse their approximation power. In fact, we can bound the approximation error to the true geodesics distance on the manifold $\mathcal{X}$ under certain assumption on the sampling density.

3.2.1.1 K-nearest neighbor graph

The construction of this graph is straightforward. Every data point $x_i \in \mathbf{X}$ is connected by edges to the $K \in \mathbb{Z}^+$ nearest neighbors for a fixed K. This always yields a graph with a single component for some K. The weak point of this construction is the lack of symmetry. As a simple example consider isolated far away point. The K-nearest neighbors of the isolated point probably might themselves not be connected to the isolated points because of the large distance. To circumvent this problem, the alternative symmetric K-nearest neighbor graph might be used instead. The construction goes as follows: two vertices x_i, x_j with $i \neq j$ are connected if x_i is among the K-nearest neighbors of x_j **or** vice versa. Finally, the mutual K-nearest neighbor graph is built by connecting two vertices (x_i, x_j) with $i \neq j$ if x_i is among the

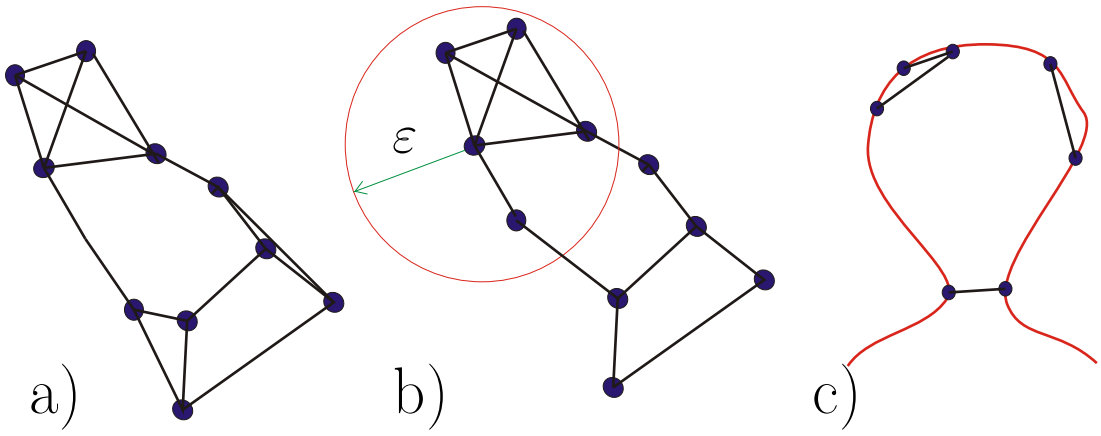


Figure 3.3: a) shows the result of the nearest K-neighbor rule for $K = 3$ and b) depicts the graph obtained by applying the ε distance rule. c) illustrates the problem when either the geometry is strongly curved or the sampling is not uniform. This can lead to cycles and or several components in the graph. In this example we used an ε neighborhood.

K-nearest neighbors of x_j **and** vice versa. Similarly, we use a weighted mutual K-nearest neighbor graph, in which all edges are weighted by the similarity of the adjacent points. This last rule is also depicted in Figure 3.3a).

3.2.1.2 ε neighborhood graph

An alternative to the K-nearest neighbor construction is the use of ε-neighborhood graphs. In this case, two vertices (x_i, x_j) with $i \neq j$ are connected by an edge if and only if the distance is equal or smaller than ε, $d_{\mathcal{X}}(x_i, x_j) \leq \varepsilon$. In order to assure a completely connected graph one uses a ε-neighborhood graph ε set to its maximal value.
In both cases, we have to set a parameter K or ε which is not trivial to choose. In [Maier 2009] Meier et.al. explored the influence of graph construction on graph-based clustering measures. They analyze the convergence of a graph clustering criteria as the sample size tends to infinity. The study reveals that a K-nearest neighbor graph yields different clusters than a r-nearest neighbor graph with $r \neq K$ and thereby showing the strong dependency on the neighborhood graph.
Lastly, we look at the problems which might occur when using the ε-nearest

neighbor graph (Figure 3.3c)). A small ε might create multiple components in the graph and most of the latter methods will not compute meaningful results. On the other hand, when the data is sparsely sampled from a highly curved manifold the ε-nearest neighbor graph does not recover the geometry properly and connects nodes which should remain disconnected.

Nevertheless we will see in the following section that the ε-nearest neighbor graph is more suited for the analysis of the approximation of geodesics in graphs (Bernstein et.al. [Bernstein 2000]).

3.2.1.3 Geodesic distance approximation

The idea of building graphs from local neighborhoods aims at recovering the geometry of the low dimensional curved submanifold embedded in $\mathbb{R}^D$. There are two possibilities one can adopt to measure the distance between two points $x_i, x_j \in \mathcal{X}$ with $i \neq j$. The common approach is to measure the distance between the embedding space coordinates. Unfortunately, this method does not take into account the geometry of the data. Whereas, if one considers a graph constructed with one of the previous methods as a length space(c.f. section 2.2) or a submanifold, the distance between $(x_i, x_j) \in \mathbf{X}$ with $i \neq j$ with respect to the metric on the graph, is more meaningful. Recently, the idea of approximating the data's geometry using a graph has emerged. Therefore, a nearest neighbor graph is built on the dataset. Then each edge is weighted using a similarity function (e.g. Euclidean distance). This construction induces a length metric on the graph. The distance between two nodes x_i and x_j is computed as the sum over the edges contained in the path joining the nodes. Using the length structure on the graph, we approximate geodesics on $\mathcal{X}$ by computing the shortest paths in the graph. The problem of shortest paths in graphs can be efficiently solved using dynamic programming as proposed by Dijskstra [Dijkstra 1959].
An important issue for the latter algorithm is the quality of approximation of the true geodesic distances computed using Dijskstra algorithm. MDS strongly depends on the approximation quality of the true geodesics. Bernstein et.al. [Bernstein 2000] provide some upper and lower bound on the approximation error:

Theorem 16. *For $\lambda > 0$ and some small enough $\delta, \varepsilon > 0$ and $\delta < \varepsilon$*

$$(1-\lambda)d_{\mathcal{X}}(x_i, x_j) < d_{\mathcal{G}}(x_i, x_j) < (1+\lambda)d_{\mathcal{X}}(x_i, x_j)$$

holds for all x_i, x_j if X is a δ-sample of $\mathcal{X}$ and if $\mathcal{G}$ is an ε nearest neighbor graph.

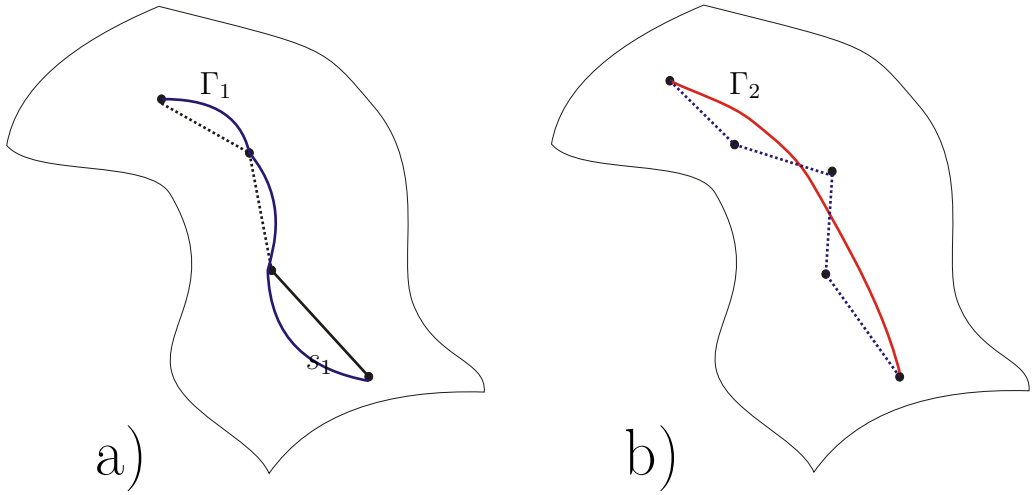


Figure 3.4: a) The sampling is not sufficiently dense to approximate the arc segment s_1 of Γ_1. b) depicts a sufficiently dense sampling to properly approximate the curve Γ_2 using Euclidean distance between sample points.

The complete proof of this theorem can be found in [Bernstein 2000]. The proof has two important arguments. The first argument is the δ sampling density condition. This condition assumes that for every point $x \in \mathcal{X}$ there is a data point x_i so that $d_{\mathcal{X}}(x, x_i) < \delta$, where $d_{\mathcal{X}}$ is the geodesic distance in $\mathcal{X}$. Figure 3.4a) does not satisfy this condition. The second argument is that the geodesic distance in $\mathcal{X}$ can be approximated by short Euclidean distance hops as depicted in Figure 3.4b). This condition is intuitively coupled with the local curvature of $\mathcal{X}$. So assuming that the $d_{\mathcal{X}}(x_i, x_j) \approx \|x_i - x_j\|$ assumes some regularity on the curvature of $\mathcal{X}$.

3.2.2 Isomap

Isomap [Tenenbaum 2000] is a generalization of classical MDS in which the pairwise distance matrix is replaced by the matrix of pairwise geodesic distances approximated by distances in graphs. The algorithm consists of three steps:

1. Approximate the manifold $\mathcal{X}$ by building a graph on the data set as introduced in section 3.2.1.

2. Compute pairwise distance matrix $\mathbf{D}$ with the Dijkstra algorithm.

3. Apply MDS on $\mathbf{D}$.

In Figure 3.5, we illustrate the behavior and limits of Isomap. Figure 3.5a) shows that as isometric as possible mapping produces no inconsistency. Indeed, in the case where $\mathcal{X}$ is developable and convex the geodesic distances in the graph are almost equal to the Euclidean distances in the reduced space $\mathbb{R}^d$. Nevertheless, if $\mathcal{X}$ is not convex we see that Isomap does not work properly (c.f. Figure 3.5b)). Finally, in Figure 3.5c), we note that manifolds without boundary cannot be unfolded into $\mathbb{R}^d$ using Isomap.

Although Isomap is very efficient due to certain theoretical guarantees such as the convergence, if the geodesic approximation and its organizational functionality on data are good enough, it has some negative points. First of all, we note that Isomap is a global method which involves the decomposition of a very large and dense Gram matrix. In the case of huge datasets Isomap is not tractable anymore. A very intuitive idea is to sparsify the data by using landmarks. Voting for representatives of small neighborhoods reduces the amount of data tremendously. Furthermore, the constraint on $\mathcal{X}$ to be isometric to a convex open set of $\mathbb{R}^d$ is rarely satisfied and a way to circumvent this restriction is to look at conformal maps.

Landmark Isomap

The main motivation to use Landmark Isomap (L-Isomap) [Silva 2003a] is to circumvent the scalability issue. Two major bottlenecks are observed when using Isomap. The first is the computation of all pairs of shortest paths which has complexity $\mathcal{O}(kN^2 \log N)$. Secondly, the factorization of a dense $N \times N$ matrix has cubic complexity. In a multiresolution fashion, L-Isomap uses $n << N$ landmark points from the original dataset $\mathbf{X}$ and computes a $n \times N$ pairwise geodesic distance matrix from each point to the landmark points only. The computational savings achieved are obvious since the distance calculations are now $\mathcal{O}(knN \log N)$. Instead of $\mathcal{O}(N^3)$ the factorization complexity reduces to $\mathcal{O}(nN^2)$. Several heuristics exist in the literature to choose the landmark points for L-Isomap. The d-dimensional embedding is actually obtained by applying classical MDS to the $n \times n$ matrix of squared distances between landmarks. The coordinates in the reduced space are given by the eigenvalues λ_i and eigenvectors $\mathbf{u}_i$ of the Gram matrix :

$$\begin{bmatrix} \sqrt{\lambda_1}\mathbf{u}_1^T \\ \vdots \\ \sqrt{\lambda_n}\mathbf{u}_n^T \end{bmatrix}$$

In order to find the coordinates of the non-landmark point x_i a distance based triangulation approach is used. Essentially, the procedure computes a weighted linear combination of the eigenvectors $\mathbf{u}_i$ in which the weights are given by the squared distance of x_i to the landmark points. This is also equivalent to project x_i onto the first d principal components of the landmark embedding. The second interpretation works only if the coordinates of the input points are known.

Conformal Isomap

In mathematics, a conformal mapping is a function preserving oriented angles between curves. The idea of conformal Isomap [de Silva 002] is to add angular information into the optimization. If we assume $\mathcal{X} = f(\Omega)$ where $f : \Omega \mapsto \mathbb{R}^D$ is a parametrization and Ω the support of f in $\mathbb{R}^d$. f is said to be a conformal embedding if for any $y \in \Omega$ there exists a scaling factor $s(y) > 0$ s.t. for any $\mathbf{x} \in \mathbb{R}^d$ the following equality holds $\|D_y f(\mathbf{x})\| = s(y)x$. ($D_y$ derivative w.r.t. y). To integrate this idea into Isomap the weight between edges is normalized by an approximation $\sqrt{M(i)M(j)}$ of the conformal factor $s(y)$. The weight between two nodes (x_i, x_j) is then calculated as $\frac{d_{\mathbb{X}}(x_i,x_j)}{\sqrt{M(i)M(j)}}$ with $M(i)$ being the mean over the distances between x_i and it's nearest neighbors.

Under the assumption of uniformly sampled points from a bounded convex region de Silva et.al.[Silva 2003b] provide a probabilistic lower and upper bound on the difference between the distance in the embedding $\mathcal{X}$ and the distance in the parameter domain Ω. The observed effect of the conformal factor is that regions of high density are inflated whereas regions with low density tend to shrink.

3.2.3 Locally Linear Embedding (LLE)

Locally Linear Embedding(LLE) which was introduced by Roweis [Roweis 2000b] aims at recovering the low dimensional geometry of the data by looking at the local interaction between data points. This is a new class of non-linear dimensionality reduction algorithms because the global geometry is inferred only from local interactions between points. Local interactions are recovered through the construction of a nearest neighbor graph which then

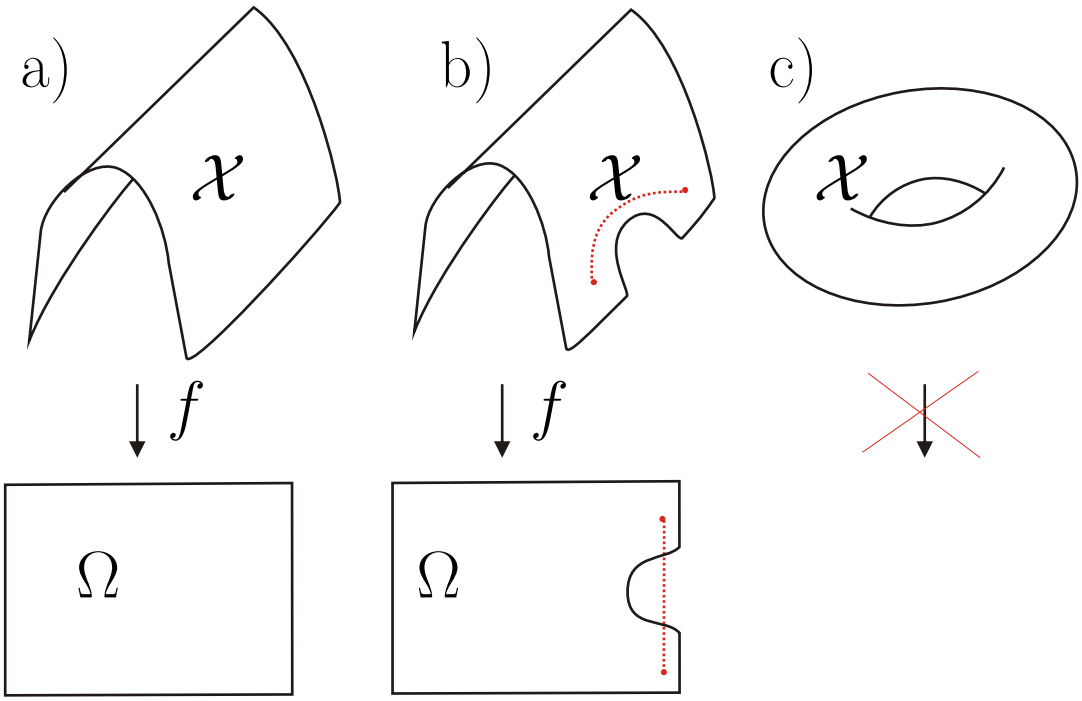


Figure 3.5: a) $\mathcal{X}$ is isometric to a convex open set b) $\mathcal{X}$ is not isometric to the parameter domain and therefore geodesics are not mapped correctly c) MDS fails because $\mathcal{X}$ is neither convex nor open

is further processed to reduce the dimensionality of the data and provide a parametrization in a d-dimensional hyperplane. The strength of local interaction is usually measured by a decreasing function which is used as weight between nodes of the graph. The algorithm has three steps:

1. Build a neighborhood graph $\mathcal{G}$ (c.f. section 3.2.1) on the data set $\mathbf{X}$.

2. For each connected pair of nodes (x_i, x_j) compute weights w_{ij} that best reconstruct each point x_i from its neighbors in the sense of the following energy: $E(W) = \sum_i \left\| x_i - \sum_j w_{ij} x_j \right\|^2$ such that $w_{ij} = 0$ if x_i and x_j are not connected in $\mathcal{G}$ and $\sum_j w_{ij} = 1$. (c.f. Figure 3.6a).

3. Finally minimize the quadratic error function $\sum_i \left\| y_i - \sum_j w_{ij} y_j \right\|$ with respect to y_i.

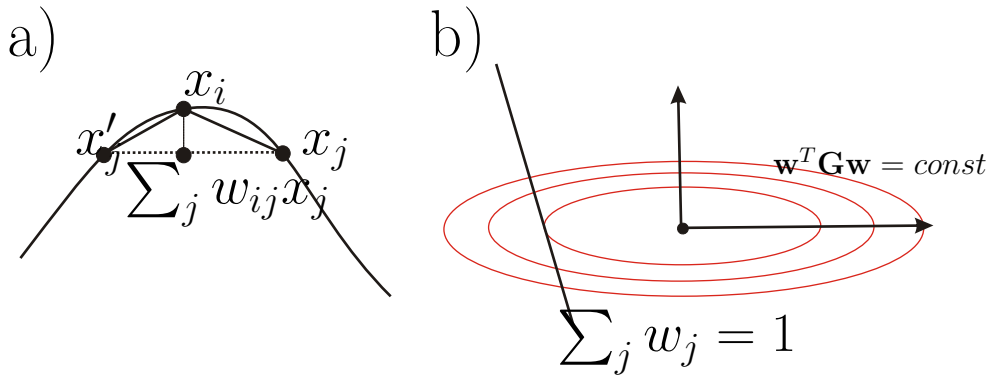


Figure 3.6: a) Approximation of the vertex x_i by it's neighbors (x'_j, x_j). b) geometry of the optimization problem. $\mathbf{w}^T\mathbf{G}\mathbf{w}$ and the level sets are drawn in red. The constraint is visualized by the black line touching the level set at a single point.

Step 1) is standard and was explained previously (c.f. section 3.2.1). Step 2) involves finding a set of optimal weights w_{ij} for each edge connecting x_i and its neighbors. The minimization of this problem requires N quadratic minimizations under constraints. For any $x \in \mathbf{X}$ and its neighbors x_j we note $w_j = w_{ij}$ the corresponding edge weight, we write

$$E = \left\| x - \sum_j w_j x_j \right\|^2 = \left\| \sum_j w_j (x - x_j) \right\|^2 = \sum_{j,k} w_j w_k \mathbf{G}_{jk} = \mathbf{w}^t \mathbf{G} \mathbf{w}.$$

$\mathbf{G}_{ik} = (x - x_j)^T (x - x_i)$ is the Gram matrix and therefore positive-semidefinite. The solution $\mathbf{w}^*$ minimizing E has the following closed form expression $\mathbf{G}\mathbf{w} = (1, \ldots, 1)^T$. Figure 3.6b) depicts the geometry the optimization problem. Note that the solution $\mathbf{w}^*$ must be rescaled to meet the condition $\sum_j w_j^* = 1$ (c.f. the black line in Figure 3.6b). The only remaining issue is when the number of neighbors is higher then the dimension of the data space $\mathbb{R}^D$. Next, we have to compute the embedding coordinates. In step 2), we have computed a set of weights $\mathbf{W}$ for each vertex which implicitly carries the information about the local connectivity between the x_i's in the original space $\mathbf{X}$. Hence, we do not need the original points x_i to compute the embedding coordinates y_i. The embedding $\mathbf{Y}$ is obtained by minimising a quadratic energy. Expanding and

rearranging the cost function we find:

$$E(\mathbf{Y}) = \sum_i \left\| y_i - \sum_j w_{ij} y_j \right\| = \sum_{ij} \mathbf{M}_{ij} y_i^T y_j$$

where $\mathbf{M}_{ij} = \delta_{ij} - w_{ij} - w_{ji} + \sum_k w_{ki} w_{kj}$ or in matrix notation $\mathbf{M} = (\mathbf{Id} - \mathbf{W})^T(\mathbf{Id} - \mathbf{W})$. An additional translational $\bar{y} = \sum_i y_i = 0$ and rotational constraint $\frac{1}{N} = \sum_i y_i y_i^T = \mathbf{Id}_d$ are added to the optimization problem in order to remove two degrees of freedom. The embedding $\mathbf{Y}$ is then given by the eigenvectors $\mathbf{u}_0, \dots \mathbf{u}_d$ of $\mathbf{M}$ with associated eigenvalues $(\lambda_0 \leq \dots \leq \lambda_d)$. Note that we keep $d+1$ eigenvectors although we have a d-dimensional embedding. This is due to the fact that the first eigenvector $\mathbf{u}_0$ is the constant vector and is left out. The coordinates y_i in the d-dimensional space $\mathbf{Y}$ is the i-th row of eigenvector matrix $\mathbf{U}$.

3.2.4 Hessian Eigenmaps

This section starts by setting up the notations and provides a theoretical result which proves the convergence of this method. We then briefly summarize the discrete implementation of this algorithm by an LLE like algorithm. The strength of the Hessian eigenmaps [Donoho 2003] is its proven asymptotic behavior. In fact, Hessian Eigenmaps are guaranteed to recover the true manifold under relatively general assumptions. This comes at the cost of estimating second order differential properties as the name of the method already implies. A major advantage of Hessian eigenmaps is that it can handle non convex parameter domains. Let's establish some ideas which are illustrated in Figure 3.7. $\mathcal{Y} \subset \mathbb{R}^d$ is an open connected set embedded in some high dimensional space $\Psi : \mathcal{Y} \mapsto \mathcal{X}$. We assume that Ψ is a smooth and locally isometric embedding. For any $x \in \mathcal{X}$ we have a local orthonormal coordinates system $(x^1, \dots, x^d)$ on the tangent space $\mathbb{T}_x\mathcal{X}$. Within a small neighborhood $\mathcal{B}(x)$ of x the projection $Pr_m(\mathcal{X})$(c.f. Figure 3.7) of any $m \in \mathcal{B}(x)$ is meaningful. Then for any twice differentiable function $f : \mathcal{X} \mapsto \mathbb{R}$, the Hessian H_f^{tan} of f at x in tangent coordinates writes :

$$H_f^{tan}(x)_{ij} = \frac{\partial}{\partial x_i} \frac{\partial}{\partial x_j} f(Pr^{-1}_{\mathbb{T}_x\mathcal{X}}(x))|_{x=0}.$$

Donoho et.al. [Donoho 2003] propose the following theorem (c.f Figure 3.8):

Theorem 17. *Let $\mathcal{X} = \Psi(\Omega)$ be an open connected set and Ψ a locally isometric embedding of Ω into $\mathbb{R}^D$. Then the nullspace of the following quadratic*

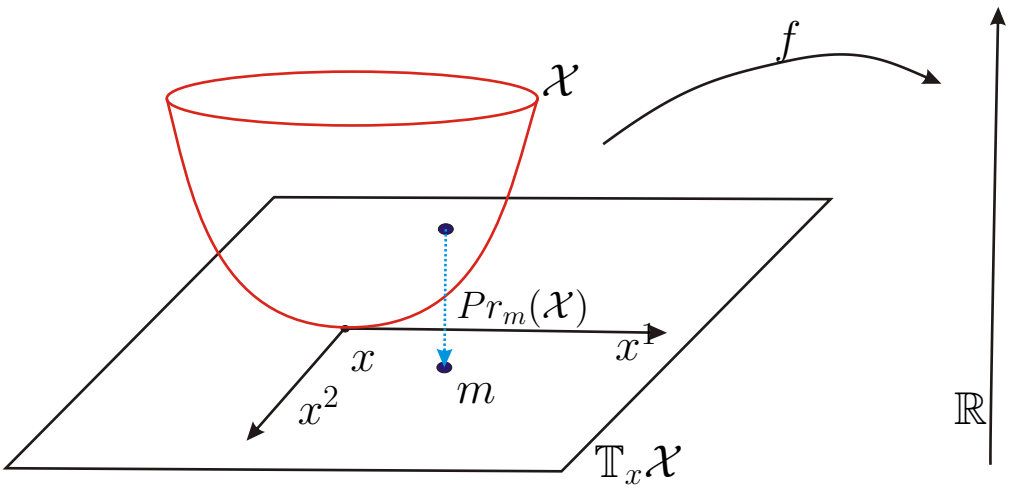


Figure 3.7: Approximation of $\mathcal{X}$ in $\mathbb{T}_x\mathcal{X}$ using a local coordinate system $(x^1, \ldots, x^d)$ in $\mathbb{T}_x\mathcal{X}$.

form

$$\mathcal{H}(f) = \int_{\mathcal{X}} \left\| H_f^{tan}(x) \right\|^2 dx.$$

is $d+1$*-dimensional and induced by the constant functions and the* d *original isometric coordinates* $pr_i \circ \Psi^{-1}$*.* $pr_i : \mathbb{R}^d \mapsto \mathbb{R}$ *is the linear projection onto the* i*-th coordinate axis in* $\mathbb{R}^d$*.*

In this theorem $\|.\|^2$ stands for the squared Frobenius norm (sum over squared entries).
In order to see the meaning of this theorem, we look at the problem in the coordinate space $\mathcal{Y} \cup \mathbb{R}^d$. Then for any twice differentiable function $g : \mathcal{Y} \mapsto \mathbb{R}$,

$$H_g^{euc}(y)_{ij} = \frac{\partial}{\partial y_i} \frac{\partial}{\partial y_j}$$

is the corresponding Hessian matrix at $y \in \mathcal{Y}$ in Euclidean coordinates. As before we define the functional

$$\mathcal{H}^{euc}(g) = \int_{\mathcal{X}} \left\| H_g^{euc}(y) \right\|^2 dy.$$

The kernel of H^{euc} is the $(d+1)$-dimensional space of affine functions on $\mathbb{R}^d$ with everywhere vanishing Hessian. Now the key ingredient to the theorem postulated before is that $\mathcal{H}$ and $\mathcal{H}^{euc}$ have the same $d+1$ dimensional nullspace under the correspondence provided by the local isometry Ψ. Or a function $f : \mathcal{X} \mapsto \mathbb{R}$ is in the nullspace of $\mathcal{H}$ if and only if $f \circ \Psi : \mathcal{Y} \mapsto \mathbb{R}$ is in the null space of $\mathcal{H}^{euc}$. So we see that finding a parametrization of $\mathcal{X}$ amounts to estimate a basis in the nullspace of the Hessian functional $\mathcal{H}(f)$.

We now describe a Hessian LLE algorithm as proposed by Donoho et.al. in [Donoho 2003]. It is an adaption of the LLE algorithm but estimating the nullspace of the Hessian of f. The algorithm can be summarized in three steps:

1. Build a K-nearest neighbor graph, where nonsymmetric neighborhoods are allowed.

2. At each data point x_i estimate a d-dimensional tangent coordinate system $\mathbb{T}_{x_i}\mathcal{X}$ by computing a singular value decomposition on the nearest neighbors.

3. Estimate the Hessian H^i at x_i in tangent coodinates in the least square sense.

4. Build a symmetric matrix $\tilde{\mathbf{H}}$ from the Hessian estimators $\mathbf{H}_i$ in the following manner:

$$\tilde{\mathbf{H}}_{ij} = \sum_l \sum_r ((H^l)_{ri}(H^l)_{rj}).$$

   H^l is the Hessian estimator

5. Performan eigenanalysis of $\tilde{\mathbf{H}}$, and identify the $d+1$ dimensional subspace corresponding to the $d+1$ smallest eigenvalues. The $d+1$ eigenvactors of $\tilde{\mathbf{H}}$ are the embedding coordinates.

So the general idea is to estimate H from the local Hessian estimates $\mathbf{H}_i$ of the data set $\mathbf{X}$. Construct a sparse $N \times N$ matrix $\tilde{\mathbf{H}}$ which approximates the continuous operator H. Note that H has the same sparsity structure as the matrix used in LLE. As with LLE, we then choose $\mathbf{Y}$ to be the eigenvectors corresponding to the d smallest eigenvalues by omitting the constant vector $\mathbf{u}_0$. However, it differs in that higher order derivatives are used to estimate the nonzero entries of $\tilde{\mathbf{H}}$.

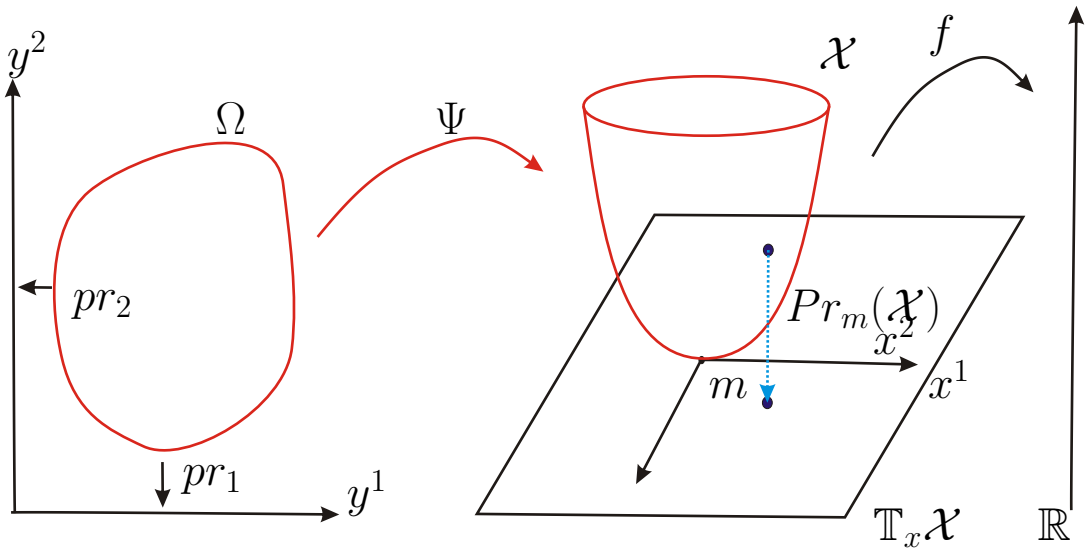


Figure 3.8: Schematic visualisation of the proof of Theorem 16.

3.2.5 Laplacian Eigenmaps

Laplacian Eigenmaps were introduced by Belkin [Belkin 2003]. Its intent is to embed the input data $\mathbf{X}$ in a d-dimensional space such that small neighborhood relations are preserved. The name Laplacian Eigenmaps is due to the Laplacian regularization in the optimization. The use of Laplacian regularization opens the door to the mathematics of harmonic analysis as we will see later. First we summarize the main steps of the algorithm:

1. Build a neighborhood graph using the K or ε nearest neighbor rule

2. Each edge connecting a pair (x_i, x_j) of points is weighted using the heat kernel $\mathbf{W}_{ij} = w_{ij} = e^{-\frac{\|x_i - x_j\|^2}{\sigma}}$ and $w_{ij} = 0$ otherwise. σ is the variance of the data set $\mathbf{X}$.

3. The solution is given by the eigenvectors of the following generalized eigenvector problem
$$\mathbf{L}f = \lambda \mathbf{D}f. \tag{3.4}$$
$\mathbf{D}$ is a diagonal matrix $\mathbf{D}_{ii} = \sum_j w_{ij}$ and $\mathbf{L} = \mathbf{D} - \mathbf{W}$ is the approximation of the Laplacian operator on the neighborhood graph.

The operator $\mathbf{L}$ is acting on a function defined on the nodes of the graph. The embedding of an initial point x_i is given by $y_i = (\Psi_1(x_i), \ldots, \Psi_d(x_i))$ initial data is given by y_i. $[\Psi_1, \ldots, \Psi_d]$ the eigenvectors to the associated generalized eigenproblem ordered in ascending order. Also note that we switched notation for the eigenvectors to emphasize the connection to eigenfunctions in the smooth case. It is obvious that when computing the eigenfunctions we actually compute eigenvectors, but we may switch between these terms in this and the next section. The first eigenvector Ψ_0 is left out because it is constant and carries no information. The optimization problem in equation 3.4 is obtained by minimizing the energy

$$E = \sum_{i,j} (y_i - y_j)^2 w_{ij} \tag{3.5}$$

with respect to the embedding $\mathbf{y}^T = (y_1, \ldots, y_N)$ of $\mathbf{X}$ in $\mathbb{R}$. This energy penalizes heavily if close points in the data space are mapped far apart in the embedding space. By expanding and rearranging the energy in equation 3.5 we obtain

$$E = \sum_{i,j} (y_i^2 + y_j^2 - 2y_i y_j)^2 w_{ij} = \sum_i y_i^2 \mathbf{D}_{ii} + \sum_j y_j \mathbf{D}_{jj} - 2 \sum_{i,j} y_i y_j w_{ij} = 2\mathbf{y}^T L \mathbf{y}$$

In order to avoid trivial solutions and scaling issues, the constraint $\mathbf{y}^T \mathbf{D} \mathbf{y} = 1$ is imposed on the solution. The Lagrangian function of the minimization yields the generalized eigenvalue problem in equation 3.4.
We now turn to the same problem in which a function is defined on a Riemannian manifold as done in the exposition by Belkin et.al [Belkin 2008]. They consider the problem of finding the best map $f : \mathcal{X} \mapsto \mathbb{R}$ so that close points on $\mathcal{X}$ remain close under the map f. Assuming the map f to be smooth the distortion induced on the neighborhood is measured by $\|\nabla f\|$. The function $f \in L^2$ minimizing the distortion is found by minimizing

$$\arg\min_{\|f\|_{L^2}} \int_{\mathcal{X}} \|\nabla f(m)\|^2 \, dm.$$

Using Stoke's identity, we have for any vector field Z on $\mathcal{X}$, $\int_{\mathcal{X}} \langle Z, \nabla f \rangle = -\int_{\mathcal{X}} div(Z) f$. From this it directly follows that

$$\int_{\mathcal{X}} \|\nabla f(m)\|^2 = \int_{\mathcal{X}} \mathcal{L}(f) f$$

where $\mathcal{L}$ is the well known Laplace-Beltrami operator defined on $\mathcal{X}$: $\mathcal{L}f = -div\nabla(f)$. The solution is obtained by the first eigenfunction f_i with associated eigenvalue λ_i different from 0. In [Belkin 2008], the authors prove convergence of the discrete Laplace operator on graphs to the continous operator as the number of samples tend to infinity.

3.2.6 Diffusion Maps

We will follow the construction of Diffusion Maps as described in [Coifman 2005]. The first step in computing Diffusion Maps is the construction of a symmetric graph in which each node x_i corresponds to a data point of $\mathbf{X} = \{x_i\}$ i=0 . . . N-1 . The connectivity between nodes is computed based on some user-defined and application dependent scale parameter σ. The adjacency graph $\mathbf{W}$ is built as the ε-neighborhood graph or the K-nearest neighbor graph.

In a second step we need to weight each edge in the graph where the weight reflects the similarity between two connected nodes. A common weight function is the Gaussian or heat kernel $w(x_i, x_j) = exp(-|||x_i - x_j|||^2/(2 * \sigma^2))$ which is also used in [Belkin 2003, Coifman 2005]. From the construction we see that this method is intimately related to the Laplacian Eigenmaps of Belkin et. al. [Belkin 2003]. It is obvious that the distance measured between nodes in the graph depends on the application and must be chosen by the user. But it also shows the generality of this method. Once a norm between data points is provided the method becomes applicable if the weight function $w(.,.)$ meets the two conditions, for all x_j and x_i in $\mathbf{X}$:

- symmetry $w(x_i, x_j) = w(x_j, x_i)$ and
- non negativity $w(x_i, x_j) \geq 0$

The weight function provides a notion of similarity and defines the amount of local interaction between nodes. We will explain this idea in more detail later.. The common construction of Graph-Laplacian's starts with the definition of the degree function $deg : \mathbf{X} \mapsto \mathbb{R}^+$:

$$deg(s_i) = \sum_{x_j \sim x_i} w(x_i, x_j)$$

where $\sim$ stands for: x_j adjacent to x_i. Then the unnormalized Laplacian, also referred to as the combinatorial Laplacian, $\mathbf{L}_u$ writes as

$$\mathbf{L}_u(x_i, x_j) = \begin{cases} deg(x_i), & \text{if } x_i = x_j \\ -w(x_i, x_j), & \text{if } x_i \sim x_j \\ 0, \text{ otherwise.} \end{cases}$$

In matrix form $\mathbf{L}_u$ writes as

$$\mathbf{L}_u = \mathbf{D} - \mathbf{W}$$

where $\mathbf{D}$ is a diagonal matrix so that $\mathbf{D}_{ii} = deg(x_i)$ and $\mathbf{W}$ is the adjacency matrix containing the weights $\mathbf{W}_{ij} = w(x_i, x_j)$. The normalized graph Laplacian $\mathbf{L}_n$ is given as

$$\mathbf{L}_n(x_i, x_j) = \begin{cases} 1 - \frac{w(x_i,x_j)}{deg(x_j)}, & \text{if } x_i = x_j \\ -\frac{w(x_i,x_j)}{\sqrt{d(x_i)d(x_j)}}, & \text{if } x_i \sim x_j \\ 0, \text{ otherwise.} \end{cases}$$

Again, we can write this in matrix form as $\mathbf{L}_n = \mathbf{Id} - \mathbf{D}^{-\frac{1}{2}}\mathbf{W}\mathbf{D}^{-\frac{1}{2}}$. In order to clarify the relation between the two Laplacian's we express the normalized Laplacian in terms of the unnormalized which then writes

$$\mathbf{L}_n = \mathbf{D}^{-\frac{1}{2}}\mathbf{L}_u\mathbf{D}^{-\frac{1}{2}}$$

Diffusion Maps [Coifman 2005] uses a diffusion kernel $w(x_i, x_j)$ to build the normalized Laplacian matrix

$$\mathbf{P}_{ij} = \frac{w(x_i, x_j)}{deg(x_i)}. \tag{3.6}$$

The diffusion kernel $w(x_i, x_j)$ encodes the probability of transition between x_i and x_j and $deg(x_i)$ normalizes the quantity in equation (3.6) such that $\sum_{x_j \sim x_i} p(x_i, x_j) = 1$. Therefore, the quantity $p(x_i, x_j)$ can be seen as the probability of a random walker to jump from x_i to x_j and $\mathbf{P}$ becomes a Markov chain on the adjacency graph. If we introduce a time index t on the random walk matrix $\mathbf{P}^t$, where t corresponds to the t-th power of $\mathbf{P}$ then the kernel $\mathbf{P}^t_{ij}$ corresponds to the probabilities of transition between x_i and x_j in t time steps. For a connected graph when $t = \infty$ the random walk converges to a unique stationary distribution ϕ_0. In [Belkin 2003], the author states that the Gaussian kernel approximates the Laplace Beltrami operator if the data lies approximately on a Riemannian submanifold with uniform distribution. This idea goes hand in hand with the asymptotic behavior of the diffusion kernel which converges to a constant function and illustrates the well-known averaging behavior of the Laplace Beltrami operator. Therefore ϕ_0 is an eigenvector of $\mathbf{P}$ such that $\phi_0^T\mathbf{P} = \phi_0^T$. Using a well known fact from spectral graph theory, Coifman [Coifman 2005] shows the following eigendecomposition of the kernel p^t :

$$p^t(x_i, x_j) = \sum_{l \geq 0}^{N-1} \lambda_l^t \Psi_l^t(x_i)\phi_l^t(x_j). \tag{3.7}$$

$\{\lambda_i^t\}$ is the decreasing eigenspectrum of $\mathbf{P}^t$ and $\{\phi_l^t(x_j)\}$ respectively $\{\Psi_l^t(x_i)\}$ being the corresponding biorthogonal left and right eigenvectors such that

$$\phi_0(x)\Psi_l(x) = \phi_l(x). \tag{3.8}$$

Then the diffusion distance $\mathcal{D}_t(x_i, x_j)$ between two points x_i and x_i can be written as

$$\mathcal{D}_t^2(x_i, x_j) = \sum_{x_l \in X} \frac{(\mathbf{P}_{il}^t - \mathbf{P}_{jl}^t)^2}{\phi_0(x_l)}. \tag{3.9}$$

This simple L^2 weighted distance between the conditional probabilities $\mathbf{P}_{i.}^t$, $\mathbf{P}_{j.}^t$ defines a metric on the data that measures the amount of connectivity of the points x_i and x_j along paths of length t. Therefore if the graph has a large number of paths connecting x_i and x_j then the diffusion distance will be small. This behavior of the diffusion distance is used by Lafon et al. to propose a clustering algorithm using Diffusion Maps [Lafon 2006a]. In order to relate the diffusion distance and the eigenvectors of $\mathbf{P}^t$ we insert equation (3.7) into equation (3.9) and use the biorthogonality between right- and left eigenvector to find:

$$\mathcal{D}_t^2(x_i, x_j) = \sum_{i \geq 0}^{m-1} \lambda_t^2 (\Psi_i^t(x_i) - \Psi_i^t(x_j))^2. \tag{3.10}$$

Ψ_i^t are the right eigenvectors associated to $\mathbf{P}^t$ and since Ψ_0^t is a constant vector it is left out of the sum. Equation (3.10) shows that the right eigenvectors of $\mathbf{P}^t$ can be used to express the diffusion distance in terms of the eigenvectors of $\mathbf{P}^t$. To this end, we introduce the family of Diffusion Maps indexed by a time parameter t

$$\Psi_t(x) = \begin{pmatrix} \lambda_0^t \Psi_0^t(x) \\ \lambda_1^t \Psi_1^t(x) \\ \vdots \\ \lambda_{N-1}^t \Psi_{N-1}^t(x) \end{pmatrix}. \tag{3.11}$$

In the sequel we will omit the parameter t and assume it is set to a fixed value [Lafon 2006b]. From equation (3.10), we can see that Diffusion Maps generate a quasi-isometric mapping since the diffusion distance is approximately equal to the L^2 metric in the new coordinate system when retaining the first d eigenvectors. The parameter d is intimately related with the intrinsic dimension present in the data. In general, its automatic estimation is very difficult. Usually one relies on the eigenspectrum to infer the parameter d. See [Hein 2005a] for an algorithm to estimate the intrinsic dimension of submanifolds in $\mathbb{R}^D$. Here, d is a parameter to be set manually. Also note that methods like LLE or Laplacian Eigenmaps do not provide an explicit metric which is crucial for our method. Let us finally mention that a completely density invariant Markov chain can be built by re-weighting the kernel as:

$$\tilde{w}(x_i, x_j) = \frac{w(x_i, x_j)}{q(x_i) q(x_j)}, \tag{3.12}$$

with $q(x_i) = \sum_{x_j \in \mathcal{X}} w(x_i, x_j)$. Then the new transition probability becomes

$$p(x_i, x_j) = \frac{\tilde{w}(x_i, x_j)}{\tilde{q}(x_i)} \tag{3.13}$$

with $\tilde{q}(x_i) = \sum_{x_j \in \mathcal{X}} \tilde{w}(x_i, x_j)$. In this way one can recover the geometry of the manifold independent of the data's density.

At end, the embedding for every data point $x_i \in \mathbf{X}$ in $\mathbb{R}^d$ is given by

$$\mathbf{y}_i = \Psi(x_i) = \begin{pmatrix} \lambda_0 \Psi_0(x_i) \\ \lambda_1 \Psi_1(x_i) \\ \vdots \\ \lambda_{d-1} \Psi_{N-1}(x_i) \end{pmatrix}. \tag{3.14}$$

3.2.7 Summary of metric learning algorithms

Depending on the availability of the training examples, most distance metric learning techniques can be classified into two categories: Supervised Distance Metric Learning and Unsupervised Distance Metric Learning. We reviewed unsupervised methods in this chapter. While supervised metric learning makes use of label information to identify, unsupervised distance metric aims to reduce the dimensionality of the dataset to the number of degrees of freedom. The resulting low-dimensional embedding preserves geometric relationships between data points with respect to some user defined distance. In table 3.1 the general properties for distance metric learning methods in this chapter are listed. For instance the following properties were considered: linear vs. nonlinear and global vs. local. These properties describe the major strategies for unsupervised learning algorithms.

Method	Locality	Linearity
PCA	global structure preserved	linear
MDS	global structure preserved	linear
ISOMAP	global structure preserved	nonlinear
LLE	local structure preserved	nonlinear
HLLE	local structure preserved	nonlinear
Laplacian Eigenmaps	local structure preserved	nonlinear
Diffusion maps	local structure preserved	nonlinear
Method	Strategy	special feature
PCA	preserves variance of data	based on inner product
MDS	preserve inter-point distance	based on distance function
ISOMAP	preserve the geodesics	graph based
LLE	preserve local neighbor	linear neighborhood approximation
HLLE	preserve local neighbor	theoretical guarantees
Laplacian Eigenmaps	preserve local neighbor	laplacian regularization
Diffusion maps	preserve local neighbor	wavelet approach and density independence

Table 3.1: Overview of unsupervised distance based learning methods and their properties.

Part II

Applications

Chapter 4

Non-linear Manifold learning and Applications to Shape and Image Denoising

Contents

4.1 Introduction 67

4.1.1 Contributions 69

4.2 Background 70

4.2.1 Kernel methods 71

4.2.2 Calculus of shapes 76

4.2.3 Mean on statistical manifolds 81

4.2.4 Pre-Image 83

4.2.5 Out-of-sample extension 84

4.3 Normalization in Gaussian Kernel PCA 87

4.3.1 Pre-image in Kernel PCA 89

4.3.2 Application in image denoising 91

4.4 Pre-Image as Karcher means 94

4.4.1 Related work 94

4.4.2 Learning a set of shapes 94

4.4.3 Shape interpolation as Karcher mean 95

4.4.4 Pre-Image and manifold interpolation 96

4.4.5 Implementation issues 97

4.4.6 Results 98

4.5 Discussion 103

Abstract

We address the problem of finding the pre-image of a feature vector in the feature space induced by diffusion maps. We offer a manifold interpretation and define the pre-image as a Karcher mean interpolating between neighboring samples for the diffusion distance. The close connections between manifold learning techniques and kernel methods are clearly exposed, providing some insights on the superiority of diffusion maps over kernel methods for learning the structure and intrinsic geometry of a manifold. In addition, a careful analysis of kernel PCA in centered feature space reveals how to improve existing pre-image methods. Results on 3D shapes and 2D images are presented and demonstrate the superiority of our approach.

4.1 Introduction

The notion of shape is important for several fields in Computer Vision and is used for object tracking in image sequences or for object recognition in scene understanding. Tracking for instance, is usually solved by segmenting the image into two regions. One region being the object itself and the background being the rest of the image. In order to regularize the segmentation, the notion of shape is used to account for low contrast or partially hidden objects. Especially when the targeted object is known before the segmentation, then such knowledge is used as a prior or shape prior. Nevertheless, shape is a complex notion which cannot be easily modeled, unlike images, as a vector in $\mathbb{R}^D$. An important question is how to model space of shapes. For instance, we want to be able to compute distances to make the models computable. Distances require a metric on the space of shape. Finding a metric on the space of curves and surfaces is the topic of the work by Michor [Michor 2006]

In general the questions arising in shape analysis are the following:

- Definition of a shape
- Structural properties of the shape space
- Metrification of the space of shapes (comparing shapes)
- Characterization of a family of shapes and their modes of variation

There is no consensus found so far on the definition of the shape. And the defition is strongly dependent of the application targeted. A common choice of representation of shapes are landmarks distributed on the object

features. This is the representation chosen in [Cootes 1995]. Another very common representation nowadays is a boundary representation of the object. The object is defined through a closed surface separating interior and exterior of the shape. This representation is done using an implicit function.

This leads to the points of view as in Grenader's theory of patterns [Grenander 1993]. Like many authors before [Charpiat 2005, Michor 2006, Beg 2005, Charpiat 2006b] we share this point of view in this thesis. It considers a shape as a point on an infinite-dimensional manifold.

But contrary to previous work, we do not model the variations of shapes by infinite-dimensional groups, such as diffeomorphisms[Michor 2006, Beg 2005]. These authors have built a Riemannian structure on the set of shapes, i.e., to go from an infinitesimal metric structure to a global one. The infinitesimal structure is defined by an inner product in the tangent space (the set of normal deformation fields) and has to vary continuously from point to point, i.e., from shape to shape. The Riemannian metric is then used to compute geodesic curves between two shapes: these geodesics define a way of warping either shape onto the other. For instance this is the work done by Trouve and Younes [Trouvé 2005]. The major drawback is that these methods have to deal with parametrization of the shape.

Since we focus on the problem of image and shape denoising we give a short overview of scientific research done on these topics:

Image denoising

Image denoising is very popular in the mathematical imaging community as this kind of problem is very suitable to make numerical experiments for mathematical models. The standard model is : given an original image u altered by additive noise v and eventually an operator R. Such an operator is usually modeled as a convolution and not necessarily invertible. Then given an observed image $I = Ru + v$, the goal is to recover the original image u. Under the simplistic assumption of Gaussian noise v, the method of maximum likelihood leads to a minimization problem

$$\inf_u \|f - Ru\|_2^2$$

where $\|.\|_2^2$ is the L^2 norm. This problem is an inverse problem and ill-posed. In order to resolve it numerically, a regularization term is added to the original problem leading to

$$\inf_u \underbrace{\|f - Ru\|_2^2}_{\text{data attachement}} + \underbrace{L(u)}_{\text{regularization}} .$$

A great amount of literature is available to help resolve this problem. We compiled a short list with the most important methods ignoring hybrid approaches.

- Gaussian smoothing model [Lindenbaum 1994]
- Anisotropic filtering Perona-Malik model [Perona 1990]
- Rudin-Osher-Fatemi total variation model [Guichard 2001]
- Wavelet tresholding [Donoho 1995]
- Non-local mean denoising [Buades 2005]

Shape denoising

In the case of shape denoising where a shape is taken in the sense of as specified before, very little work has been done. The first to propose shape denoising based on manifold learning techniques was Rathi et.al. [Dambreville 2006]. In this work, authors use KPCA to compute an embedding in the feature space and then perform a projection onto the first leading eigenvectors. The corresponding data space of the projected datum is obtained by solving a fixed point iteration. Further work on manifold learning applied to shapes was introduced in [Etyngier 2007b] by Etyngier et.al. Their approach rely on diffusion maps. Their objective was the design of a prior energy for image segmentation. Nevertheless, the proposed algorithm can also be applied to solve the preimage problem in the context of diffusion maps.
A different approach based on level-sets was introduced in by Dong et.al.[Dong 2008]. In their paper they extend the non-local smoothing technique for images to the case of a surface represented by surfaces. They show promising results on synthetic data as well as city terrain and cortical surfaces. Nevertheless, the noise is very low in their example. This approach cannot cope with strong noise.

4.1.1 Contributions

We propose a new method to solve the pre-image (see section 4.4) problem in the context of diffusion maps for shape and image denoising. We suggest a manifold interpretation and learn the intrinsic structure of a given training set. We compute a global metric using diffusion maps on the set of shapes and compute smooth approximations of these metrics based on similarity functions. We then minimize these functions using techniques of the

calculus of variation by computing their gradient and performing infinitesimal gradient descent to interpolate between shapes. Our method relies on a geometric interpretation of the problem which naturally leads to the definition of the pre-image as a Karcher-mean [Karcher 1977] that interpolates between neighboring samples according to the diffusion distance. Previous pre-image methods were designed for Kernel PCA. Our motivation for using diffusion maps comes from the fact that the computed mapping captures the *intrinsic* geometry of the underlying manifold independently of the sampling. Therefore, the resulting Nyström extension (see section 4.2.5) proves to be more "meaningful" far from the manifold and leads to quantitatively better pre-image estimations, even for very noisy input data. Further, this method has several significant advantages:

- In the case of Kernel PCA, our new normalization criteria performs better on very noisy images.
- We provide a variational formulation of the pre-image problem in the case of diffusion maps.
- Diffusion maps are more stable even when dealing with very noisy data
- We provide a way to interpolate between shape respecting the underlying geometry of the shape space.

Last but not least, the method proposed has the advantage of being relatively general. The main ingredient in order for this to work, is the use of a differentiable distance function. One could expect the method to work on splines and nurbs and several other representations in Computer Vision.
Further, we show some results on denoising of shapes. We compare our results to the work proposed by Dambreville [Dambreville 2006] and for image denoising, to several denoising algorithms using Kernel PCA: [Kwok 2004], [Arias 2007], [Carreira-Perpiñan 2007]. Results on 3D shapes and 2D images are presented and demonstrate the superiority of our method. The shapes considered are medical MRI 3D volumes. For computations, the volumes are converted into a boundary representation using an implicit function. The gradient descent to solve the pre-image yields a hyperbolic equation which is solved using the Level-Set approach [Sethian 1999b].

4.2 Background

For the sake of clarity, we review and further detail some of the methods introduced in chapter 2 and 3.

4.2.1 Kernel methods

Kernel methods constitute a class of algorithms for the analysis of various kind of data (vectors, character chains, graphs, images) using a similarity measure $w(.,.)$. The similarity measure is commonly named **kernel** and associates each pair of input data (x_i, x_j) a real number $w(x_i, x_j)$ measuring the similarity between the data.
By making certain assumptions on the kernel $w(.,.)$, the kernel implicitly maps the data into a highdimensional Hilbert space through $\Psi : \mathcal{X} \mapsto \mathcal{Y}$ such that the following relation holds : $w(x_i, x_j) = \langle \Psi(x_i), \Psi(x_j) \rangle_{\mathcal{Y}}$. $\langle ., . \rangle_{\mathcal{Y}}$ is the inner product in the feature space. In the literature[Smola 1998, Schölkopf 2002], the space $\mathbb{H}$ is commonly called **feature space** and the space where the data lies $\mathcal{X}$, **input space**.
This yields a very general and flexible framework for data analysis. But its behavior will depend on the choice of kernel, which is equivalent to say on the choice of the mapping. Once the kernel is fixed several linear algorithm can be applied such as PCA and Support Vector Machines (SVM).
In the section, we review the main definitions related to kernel methods as well as some results and applications useful and in relation to this work. As mentioned in [Bengio 2003, Ham 2003], there exists a deep connection between kernel methods and manifold learning.

Kernel Mapping

Definition 18 (Positive definite kernel)**.** A positive definite kernel with input space $\mathcal{X}$ is a symmetric mapping $k : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$

$$w(x, x') = w(x', x) \text{ for every } (x, x') \in \mathcal{X}^2,$$

such that for every $n \in \mathbb{N}$, $(x_1, \ldots, x_N) \in \mathcal{X}^n$, $(a_1, \ldots, a_n) \in \mathbb{R}^n$

$$\sum_{i=1}^{N} \sum_{i=j}^{N} a_i a_j w(x_i, x_j) \geq 0$$

In matrix notation, the similarity matrix $\mathbf{W}_{ij}$ is positive semi-definite for every n-tuple of input points. The most trivial case is the linear kernel defined on $\mathcal{X} = \mathbb{R}^D$. Then, $k : \mathcal{X} \mapsto \mathbb{R}$ is given by

$$\forall (x, x') \in \mathcal{X}^2, \; w(x, x') = \langle x, x' \rangle_{\mathbb{R}^D}$$

It is easy to show that the linear kernel is positive definite. In fact, symmetry is obvious and positivity is deduced by linearity as

$$\sum_{i=1}^{N}\sum_{i=j}^{N} a_i a_j \langle x_i, x_j \rangle = \left\| \sum_{i=1}^{M} a_i x_i \right\|_{\mathbb{R}^D}^2 \geq 0.$$

The linear kernel can be generalized in the following way. Let $\mathcal{X}$ be a metric space, $\mathcal{Y}$ a vector space of finite dimension and $\Psi : \mathcal{X} \mapsto \mathcal{Y}$. Then the positive definite function $k : \mathcal{X}^2 \mapsto \mathbb{R}$ is defined as :

$$\forall (x, x') \in \mathcal{X}^2,\ w(x, x') = \langle \Psi(x), \Psi(x') \rangle_{\mathcal{Y}}\,.$$

In this way, we are able to define a kernel function defined over general metric space through the mapping Ψ mapping the space $\mathcal{X}$ into a finite dimensional vector space.

Kernel function as inner-product

We now seek a representation of some inner product in a space $\mathcal{X}$ through a positive definite kernel $w(.,.)$.

Theorem 19. *$w(.,.)$ is positive definite if and only if there exists a mapping $\Psi : \mathcal{X} \mapsto \mathcal{Y}$ such that for all $x, x' \in \mathcal{X}$*

$$w(x, x') = \langle \Psi(x), \Psi(x') \rangle_{\mathcal{Y}}$$

The idea of this theorem is illustrated in Figure 4.2.1. The kernel $w(.,.)$ implicitly maps the data into a high-dimensional Hilbert space $\mathcal{Y}$ through the unknown mapping Ψ and evaluates the inner product between the two points. With the property we can measure distances in the feature space $\mathcal{Y}$. A space $\mathcal{Y}$ having this property is called a kernel reproducing Hilbert space(RKHS). Since computing the matrix $\mathbf{W}_{ij} = w(x_i, x_j)$ amounts to computing all inner products between all pairs of points of $\mathcal{X}$, the kernel defines a metric on $\mathcal{X}$ and $\mathbf{G} = \mathbf{W}$ is the associated Gram matrix. Since $w(.,.)$ is differentiable and defines an inner product on $\mathcal{X}$, [Amari 1999] propose an explicit expression for the metric tensor on $\mathcal{X}$ in terms of the second order derivatives of the kernel:

$$g_{ij}(x) = \langle \nabla\Psi(x), \nabla\Psi(x') \rangle\,|_{x=x'} = \frac{\partial}{\partial x_i}\frac{\partial}{\partial x'_j} w(x, x')|_{x=x'}.$$

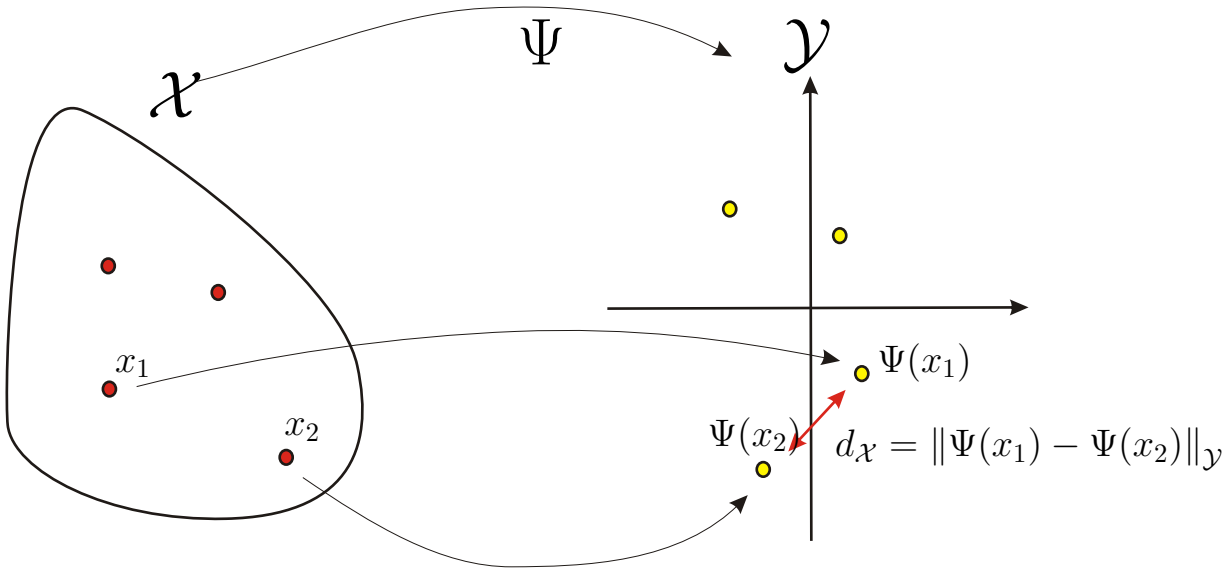


Figure 4.1: Embedding of $\mathcal{X}$ into $\mathcal{Y}$. $d_{\mathcal{X}}$ is a distance defined on $\mathcal{X}$ through a positive definite kernel.

Kernel trick and distance computation

In practice, the mapping does not have to be computed explicitly as most techniques only require the computation of dot products that can be evaluated directly using the kernel $w(.,.)$. This is called the kernel trick. Therefore evaluating the kernel on the points amounts to mapping the points into a high dimensional possibly infinite dimensional space $\mathcal{Y}$ and compute their inner product.

$$w(x, x') = \langle \Psi(x), \Psi(x') \rangle_{\mathcal{Y}} .$$

With this trick, points in feature space are processed implicitly through the kernel. For instance, we can compute the distance between two points in the feature space. Note that,

$$\begin{aligned} d_{\mathcal{X}}(x, x')^2 &= \|\Psi(x) - \Psi(x')\|_{\mathcal{Y}}^2 \\ &= \langle \Psi(x) - \Psi(x'), \Psi(x) - \Psi(x') \rangle_{\mathcal{Y}} \\ &= \langle \Psi(x), \Psi(x) \rangle_{\mathcal{Y}} + \langle \Psi(x'), \Psi(x') \rangle_{\mathcal{Y}} - 2 \langle \Psi(x), \Psi(x') \rangle_{\mathcal{Y}} \\ &= w(x, x) + w(x', x') - 2w(x, x'). \end{aligned}$$

This defines a distance on $\mathcal{X}$ which is calculated through a kernel $w(.,.)$(c.f. Figure 4.2.1). More precisely, $d_{\mathcal{X}}$ is a pseudo-distance because Ψ is not injective and therefore the property $d(x,x') = 0 \Leftrightarrow x = x'$ is not verified. In view of the development of a kernel version of PCA, we show how to remove the mean $\text{Mean} = \frac{1}{N}\sum_{i=1}^{N}\Psi(x_i)$ from the Gram matrix $\mathbf{G}$. Remember the definition of the Gram matrix $\mathbf{G}_{ij} = \langle x_i, x_j\rangle$ then the centered Gram matrix $\mathbf{G}^C_{ij} = \langle x_i - \text{Mean}, x_j - \text{Mean}\rangle$. When using a kernel to evaluate the dot product between two points, the mean can not computed explicitly as the mapping Ψ is not known. Nevertheless, by using the kernel trick we can write a centered version of the Gram matrix when using a kernel

$$\begin{aligned}
\mathbf{G}^C_{ij} &= \langle \Psi(x_i) - \text{Mean}, \Psi(x_j) - \text{Mean}\rangle_{\mathcal{Y}} \\
&= \langle \Psi(x_i), \Psi(x_j)\rangle_{\mathcal{Y}} - \langle \text{Mean}, \Psi(x_i) + \Psi(x_j)\rangle_{\mathcal{Y}} + \langle \text{Mean}, \text{Mean}\rangle_{\mathcal{Y}} \\
&= \langle \Psi(x_i), \Psi(x_j)\rangle_{\mathcal{Y}} - \frac{1}{N}\sum_{k=1}^{N}\langle \Psi(x_k), \Psi(x_i) + \Psi(x_j)\rangle_{\mathcal{Y}} + \frac{1}{N^2}\sum_{k=1}^{N}\sum_{l=1}^{N}\langle \Psi(x_k), \Psi(x_l)\rangle_{\mathcal{Y}} \\
&= \mathbf{G}_{ij} - \frac{1}{N}\sum_{k=1}^{N}(\mathbf{G}_{i,k} + \mathbf{G}_{j,k}) + \frac{1}{N^2}\sum_{k=1}^{N}\sum_{l=1}^{N}\mathbf{G}_{kl}
\end{aligned}$$

which writes in matrix notation as

$$\mathbf{G}^C = \mathbf{HGH}.$$

$\mathbf{H}$ is called the centering matrix $\mathbf{H} = \mathbf{Id}\mathbb{1}\mathbb{1}^T$. It might be interesting to visualize the pre-image of the mean in feature space. Therefore we must compute $x_{\text{Mean}} \in \mathcal{X}$ such that $\Psi(x_{\text{Mean}}) = \text{Mean}$. This problem is called the pre-image problem and will be discussed in more details in section 4.2.4.

Empirical Kernel Mapping

The previous section introduced very general ideas on reproducing kernel Hilbert spaces. Unfortunately, the mappings involved are usually infinite dimensional and do not allow any computations except for the kernel trick. To make computations in the RKHS, we introduce a finite dimensional approximation to the original possibly infinite dimensional space.
For instance, we take a finite sampling X of the input space $\mathcal{X}$ as this is the standard situation encountered in applications where data is gathered from several measurements. If we consider only linear operations between the embedded points $\Psi(x_i)$ with $i = 1, \ldots, N$, we can restrict the infinite dimensional case to a finite dimensional subspace spanned by the N mappings $\Psi(x_i)$ as

$$\mathcal{Y}^N = \left\{ \mathbf{u} = \sum_{i=1}^{N}\alpha_i\Psi(x_i),\ \alpha_i \in \mathbb{R},\ x_i \in X \right\}$$

The space $\mathcal{Y}^N$ is the empirical feature space which is spanned by a weighted linear combination of the embedded points $\Psi(x_i)$ with $i = 1, \ldots, N$. A representation of the basis in $\mathcal{Y}^N$ can be done by expanding on the basis given by the mappings $\Psi(x_i)$ with $i = 1, \ldots, N$. If $\boldsymbol{\alpha}_\mathbf{u}$ be the coefficient vector of u

$$\mathbf{u} = \sum_{i=1}^{N} (\boldsymbol{\alpha}_\mathbf{u})_i \Psi(x_i).$$

Then the inner product between two elements $\mathbf{u}$ and $\mathbf{u}' \in \mathcal{Y}^N$ with corresponding coefficient vectors $\boldsymbol{\alpha}_\mathbf{u}$ and $\boldsymbol{\alpha}_{\mathbf{u}'}$ is given by

$$\langle \mathbf{u}, \mathbf{u}' \rangle_\mathcal{Y} = \sum_{i=1}^{N} \sum_{j=1}^{N} (\alpha_\mathbf{u})_i (\alpha_{\mathbf{u}'})_j w(x_i, x_j^{'}) = \boldsymbol{\alpha}_\mathbf{u}^T \mathbf{W} \boldsymbol{\alpha}_{\mathbf{u}'}.$$

$\mathbf{W}_{ij} = w(x_i, x_j)$ is the usual kernel matrix and $\boldsymbol{\alpha}_\mathbf{u} = [\alpha_1, \ldots, \alpha_N]$ and analogously $\boldsymbol{\alpha}_{\mathbf{u}'}$. In order to compute the $\mathbf{u}$, we need the coefficient vectors $\boldsymbol{\alpha}$ which are computed by minimizing an energy functional [Smola 1998].

Kernel Principal Component Analysis (KPCA)

A different basis of $\mathcal{Y}^N$ can be computed by the principal component approach presented in section 3.1.1.
Using the kernel trick presented in section 4.2.1 and assuming we have a RKHS, we know from the previous section that the eigenvectors are expressed in the finite dimensional subspace spanned by the embbeded points $\Psi(x_1), \ldots, \Psi(x_N)$. We can rewrite the eigenvalue problem of section 3.1.1 in terms of the inner product using the kernel trick $w(x_i, x_j) = \langle \Psi(x_1), \Psi(x_N) \rangle$ which gives us the gram matrix $\mathbf{G} = \mathbf{W}$. This yields a dual generalized eigenvalue problem

$$N\lambda\boldsymbol{\alpha} = \mathbf{W}\boldsymbol{\alpha},$$

with α being a coefficient vector satisfying the normalisation condition $\boldsymbol{\alpha}^T \mathbf{W} \boldsymbol{\alpha} = 1$. The associated solution vectors of the eigenvalue problem are

$$\boldsymbol{\alpha}_i = \frac{1}{\sqrt{\lambda_i} \mathbf{e}_i}$$

where $(\mathbf{e}_i, \lambda_i)$ are the associated eigenvectors and eigenvalues of $\mathbf{W}$ and the projection of the input data $\mathbf{x}_1, \ldots, \mathbf{x}_N$ on the i-th component are computed as $\mathbf{W}\boldsymbol{\alpha}_i$.

4.2.2 Calculus of shapes

A considerable amount of work is published on the definition and representation of shapes. And yet there is no best representation and one has to make a choice depending on the targeted application. In this chapter, we choose an implicit representation of the shapes.

Shape representation

The definition of shape we will need in this chapter is adopted from the work of Charpiat [Charpiat 2006a]. It is a purely geometrical approach and independent of the pixels intensity. Therefore the shape is only defined by the idea of interior and exterior. In order to achieve parametrization independency, we define the characteristic function of a shape s as

$$\chi(s) = \begin{cases} 1 & \text{if } x \in s, \\ 0 & \text{if } x \notin s. \end{cases}$$

or of it's contour $\gamma = \partial s$. Then we define the signed distance function from the contour of the shape s

$$\mathbb{D}_\gamma = \varepsilon(x) \inf_{y \in \partial s} d(x, y) \text{ with } \varepsilon(x) \begin{cases} +1 & \text{if } x \in s, \\ -1 & \text{if } x \notin s \end{cases}. \tag{4.1}$$

Further, we assume that locally the shape is the graph of a twice differentiable function and sufficiently apart of it's skeleton in the sense of Charpiat [Charpiat 2006a]. We call the set $\mathcal{S}$ of all such shapes a shape space with nice topological and metric properties.

Level Sets

The level set method is very popular for the numerical simulation dynamic interfaces. First proposed by Alain Dervieux and François Thomasset [**?**] and then further developed by Osher and Sethian [Osher 1988], its success is mainly due to the automatic handling of complex geometries and topological changes, (c.f. Figure 4.2b)) as well as efficient numerical schemes and almost trivial computations of intrinsic quantities. Further, this method easily generalizes to high dimensions. Nevertheless, level sets bear some drawbacks such as loss of mass and big memory consumption. Numerous domains ranging from fluid dynamics [Osher 2001] and image segmentation [Osher 2003] to 3D reconstruction [Keriven 1998] and shape optimization [Charpiat 2006a] still employ level sets. For example, Nikos Paragios and Rachid Deriche propose

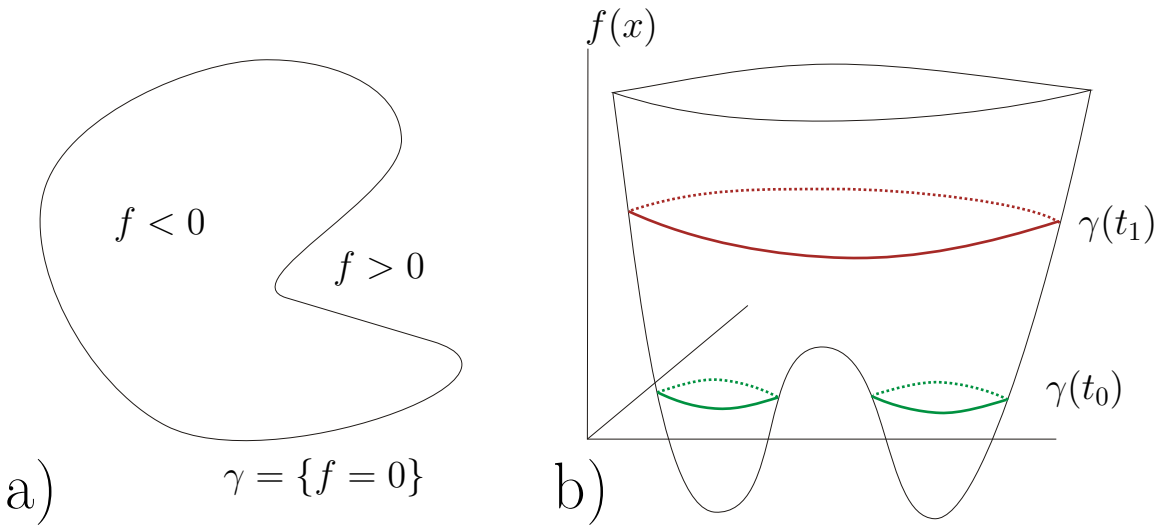


Figure 4.2: a) shows a 2D curve and a normal deformation field. b) shows the embedding of the curves in a 2+1 dimensional space a the cross section at time t =0 and different shapes at different times (t_0, t_1).

geodesic active regions [Paragios 2000, Paragios 2002] still implemented in the Level Set framework. The curve evolves according to a statistical analysis based on the maximum likelihood principle for the observed density functions (e.g. image histogram).

Level sets are represented implicitely as the 0-level set of some higher-dimensional function f defined on the whole domain. Given a hypersurface $\gamma \cup \mathbb{R}^n$, we define a function $f : \mathbb{R}^n \mapsto \mathbb{R}$ such that:

$$\gamma = \{x \in \mathbb{R}^n | f(x) = 0\}.$$

Further, we assume that f takes negative values inside the curve and positive values outside. One such function is the signed distance equation 4.1

In the context of deformable models, the evolution of a curve $\gamma(t)$ with $t \geq 0$ under a deformation field $\mathbf{v}$ is written as

$$\frac{\partial \gamma}{\partial t} = \mathbf{v}.$$

Applied to the level set equation, the deformation of γ under $\mathbf{v}$ becomes

$$\frac{\partial f}{\partial t} + \mathbf{v}\nabla f = 0$$

when $f(\gamma(t), t)$ is differentiated with respect to t at $t = 0$. Furthermore, we easily see that solely the normal component of the deformation field $\mathbf{v}$ influences the shape geometry if we know that

$$\mathbf{n} = \frac{\nabla f}{|\nabla f|}$$

and rewrite the level set equation as

$$\frac{\partial f}{\partial t} + |\nabla f|\, \mathbf{v} \cdot \mathbf{n} = 0.$$

Hence any tangential motion corresponds just to a reparametrization. Also note that the computation of several intrinsic quantities is easily computed as explained in [Osher 2001, Sethian 1999c].

Shape distances

The standard approach to shape optimization in computer vision is to minimize the distance between shapes. Therefore, a suitable distance function measuring the similarity of two shapes is required. Two ways are commonly used: The first is to equip the shape space with a metric on the tangent space at each point. This implies that the shape space is an infinite dimensional Riemannian manifold. The distance between two shapes is defined as the length of a geodesic path joining the two shapes. This framework was established by Trouve et.al.[Trouvé 2005].
A different approach is to consider a distance defined on the global geometry of the shape. Contrarily to the previous idea, this approach does not necessarily require differentiability of the underlying structure and further does not expect any parametrization of the shape space. This is the case for the following distance functions.

Symmetric difference

A commonly used similarity measure between two shapes s_1, s_2 defined over two measurable sets Ω_1, Ω_2 is the symmetric difference function:

$$d_{SD}(s_2, s_2) = \|\chi\Omega_1 - \chi\Omega_2\|_{L^2} = \left(\frac{1}{2}\int (\chi\Omega_1(x) - \chi\Omega_2(x))^2 dx\right)^{\frac{1}{2}}.$$

Geometrically (c.f. Figure 4.3b)), this distance measures the area of the symmetric difference between the two shapes. This distance was first proposed by Cremers et.al.[Cremers 2004].

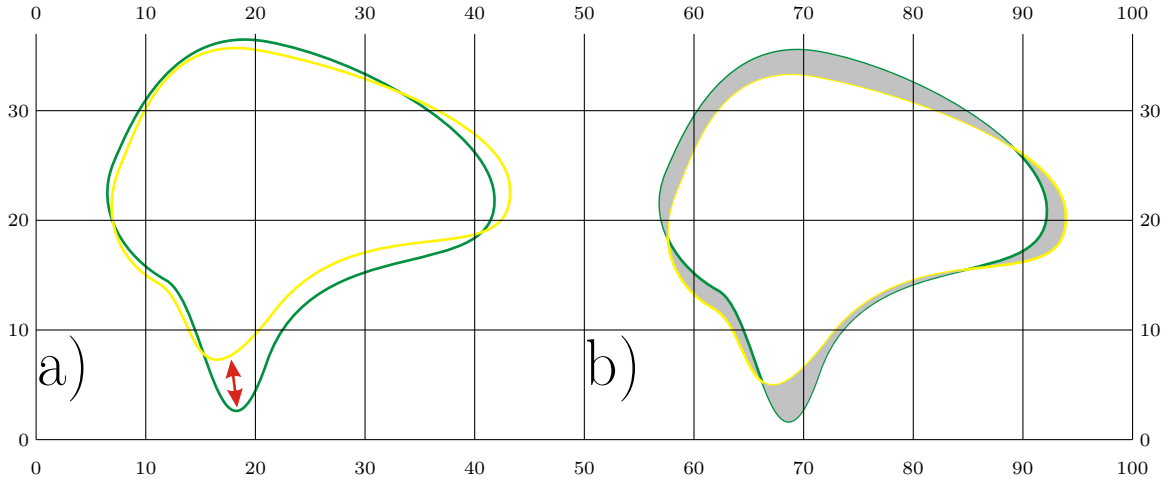


Figure 4.3: a) Haussdorf distance between two curves. b) Area of symmetric distance.

Hausdorff distance

Another well known distance is the Hausdorff distance which was first used by Serra [Serra 1998] for image processing using mathematical morphology. Therefore we consider the distance between distance functions of shapes $\mathbb{D}_{\Omega_1}, \mathbb{D}_{\Omega_2}$

$$d_H(\Omega_1, \Omega_2) = \|\mathbb{D}_{\Omega_1} - \mathbb{D}_{\Omega_1}\|_{L^\infty} = \sup |\mathbb{D}_{\Omega_1} - \mathbb{D}_{\Omega_1}|$$

which in fact turns out to be the Hausdorff distance:

$$d(\Omega_1, \Omega_2) = \max \left\{ \sup_{x \in \Omega_1} \mathbb{D}_{\Omega_2}, \sup_{x \in \Omega_2} \mathbb{D}_{\Omega_1} \right\}$$

This distance as it is, is not smooth and therefore not applicable for variational methods. Charpiat et.al. [Charpiat 2005] propose a smooth approximation to overcome this problem.

L^2 distance between Level-Set

As in [Charpiat 2005], we define the L^2 norm between two Level-Set or equivalent distance functions $\mathbb{D}_{s_1}, \mathbb{D}_{s_2}$ as:

$$d_{L^2}(s_1, s_2) = \|\mathbb{D}_{s_1} - \mathbb{D}_{s_2}\|_{L^2} .$$

Sobolev norm between Level-Set

If more regularity of the shapes is required, Charpiat et.al [Charpiat 2005]

and Sundaramoorthi et.al[Sundaramoorthi 2008] proposed to restrict the function to the Sobolev space $W^{1,2}$. This imposes regularity on the gradient of the distance function of shapes. The Sobolev norm between two shapes then writes as:

$$d_W^{1,2}(s_1, s_2) = \|\mathbb{D}_{s_1} - \mathbb{D}_{s_2}\|_{L^2} + \|\nabla\mathbb{D}_{s_1} - \nabla\mathbb{D}_{s_2}\|_{L^2}\,.$$

Gradient descent

In most cases the global minimization of an energy measuring similarity between shapes is not possible. One has to resort to a local optimization scheme. The calculus of variation provides suitable tools for the local minimization and optimality conditions to reach a local minima. By assuming a manifold structure on the data space $\mathcal{X}$, we can apply a gradient descent on the manifold like in standard variational calculus. By considering a small variation of the shape we have a meaningful interpretation of the derivative of the energy with respect to the shape and can perform a gradient descent:

Gradient descent on a manifold:

Let $\mathcal{X}$ be some Riemannian manifold and $f : \mathcal{X} \mapsto \mathbb{R}$ a smooth real valued function on $\mathcal{X}$, $x \in \mathcal{X}$ and $v \in \mathbb{T}_x\mathcal{X}$ a tangent vector on $\mathcal{X}$ at x. Then the directional derivative of f in x in the direction of v is given by

$$\frac{\partial f}{\partial v}(x) = df(x)v = \frac{d}{dt} f(\Gamma(t))|_{t=0}$$

for every differentiable curve Γ on $\mathcal{X}$ such that $\Gamma(0) = x$ and $\dot{\Gamma}(0) = v$. Then using the Riesz theorem, the unique vector $\nabla_{\mathcal{X}} f(x) \in \mathbb{T}_x\mathcal{X}$ corresponding to the gradient of f in x is

$$df(x)v = \langle \nabla_{\mathcal{X}} f(x), v\rangle_x\,.$$

Once the gradient can be computed, we can perform a gradient descent which is an iterative procedure to minimize a function. Since the gradient always points in the direction of biggest change of f, the idea is take a step along the opposite direction of the gradient. Thus the opposite gradient always decreases the function f

$$df(x)v = \langle \nabla_{\mathcal{X}} f(x), -\nabla_{\mathcal{X}} f(x)\rangle_x = -\|\nabla_{\mathcal{X}} f(x)\|_x^2 \leq 0.$$

Gâteaux derivative and shape gradient

In the case of curves or surfaces, the directional derivative is not a single vector v but rather a vector field β. The vector field β is a deformation normal to γ such that for any $x \in \gamma$ x transforms as $x+\beta(x)\mathbf{n}(x)$ where $\mathbf{n}$ is the normal to γ at x. Under the assumption of sufficient regularity and bounded energy, the deformation defines a new curve (or surface) as $\gamma + \beta\mathbf{n}$. In particular if $\beta\mathbf{n}$ is $\mathcal{C}^2$ then $\gamma + t\beta\mathbf{n}$ is well defined for small t and corresponds to the tangent space of γ. Given a certain energy functional $E(\gamma)$, the derivative of E with respect to γ in the direction of β writes as

$$dE(\gamma)\beta = \frac{d}{dt}E(\gamma + t\beta\mathbf{n})_{|_{t=0}} = \lim_{t\to 0}\frac{E(\gamma + t\beta\mathbf{n}) - E(\gamma)}{t}$$

This type of derivative is a functional derivative. And it is a Gâteaux derivative if the mapping $\beta \mapsto dE(\gamma)\beta$ is defined and continuous on the vector space of normal deformation fields β.
Defining the gradient of E in γ in the direction of β requires an inner product on the space of normal deformation fields. Usually, the space of normal deformations is defined as the sub-space of the Hilbert space $L^2(\gamma)$ with inner product

$$\langle \beta_1, \beta_2 \rangle = \frac{1}{|\gamma|}\int_{x\in\gamma} \beta_1(x)\beta_2(x)dx.$$

The gradient $\nabla_\beta E(\gamma)$ is then defined as the unique deformation field so that

$$dE(\gamma)\beta = \langle \nabla_\beta E(\gamma), \beta \rangle_\gamma .$$

The choice of a specific inner product such as the canonical L^2 which we will use in this chapter and which is systematically used throughout the literature defines the space of admissible deformation fields. As pointed out by Charpiat *et al.* [Charpiat 2007], the gradient depends on the particular choice of the inner product. This opens the door on the design of specific metrics through the choice of inner products [Charpiat 2007],[Sundaramoorthi 2008].

4.2.3 Mean on statistical manifolds

The linear definition $\frac{1}{N}\sum_i x_i$ is not so easily generalized in the case of manifolds with probability distributions. In this section, we develop the notion of a mean in the sense of Karcher and Frećhet which provides the theoretical framework for the correct definition of a mean for shapes.

Fréchet expectation

Let x be a random element of $\mathbb{R}^n$. Fréchet observed in [Fréchet 1944, Fréchet 1948] that the variance $\sigma_x^2(y) = \mathbb{E}\left[d(\mathbf{x}, y)^2\right]$ is minimized by the mean $\bar{x} = \mathbb{E}\left[\mathbf{x}\right]$. The main assumption in order to generalize this observation is that the expectation of a real measurable function is always well-defined. We consider now a distance on a manifold $\mathcal{X}$. In our case we will choose the intrinsic distance with respect to the surface metric. Let $\mathbf{x}$ be a random primitive of density p_x. The expectation of the squared distance between the random primitive and some fixed primitive y is defined as:

$$\sigma_x^2(y) = \mathbb{E}\left[d(x, y)^2\right] = \int_{\mathcal{X}} d(y, z)^2 . p_{\mathbf{x}} . d\mathcal{X}(z) \tag{4.2}$$

If the variance $\sigma_x^2(y)$ is defined for every y (this is true if the density has compact support), we call mean or expected primitive $\bar{x}$ minimizing this variance. We then have the set of mean primitives as

$$\mathbb{E}\left[x\right] = \arg\min_{y \in \mathcal{X}} \mathbb{E}\left[d(x, y)^2\right] = \int_{\mathcal{X}} d(y, z)^2 . p_x . d\mathcal{X}(z) \tag{4.3}$$

If there exists at least one mean $\bar{x}$, the variance is the minimum value $\sigma_x^2 = \sigma_x^2(\bar{x})$ and the standard deviation is the root of this value. In the same spirit, we define the empirical mean of a set of measurements $x_1, \ldots, x_n$ by a discrete version of Eq.(4.3):

$$\mathbb{E}\left[\{x_i\}\right] = \arg\min_{y \in \mathcal{X}} \mathbb{E}\left[\left\{d(x_i, y)^2\right\}\right] = \arg\min_{y \in \mathcal{X}} \left(\frac{1}{n} \Sigma_i d(y, x_i)^2\right) \tag{4.4}$$

Again if there exists at least one mean primitive $\bar{x}$, we call an empirical standard deviation or root mean square the value $\sqrt{\frac{1}{n}\Sigma_i d(\bar{x}, x_i)^2}$. We can go further and define different types of central primitives (moments). Let's call mean deviation of order α the value

$$\sigma_{x,\alpha}(y) = \left(\mathbb{E}\left[d(x, y)^\alpha\right]\right)^{\frac{1}{\alpha}} = \left(\int_{\mathcal{X}} d(y, z)^\alpha . p_x . d\mathcal{X}(z)\right)^{\frac{1}{\alpha}} \tag{4.5}$$

Under reasonable smoothness assumption every primitive $\bar{x}_\alpha$ minimizing Eq.4.5 is a central primitive of order α. Then setting $\alpha = 0$ amounts to finding the modes of the density and setting $\alpha = \infty$ amounts to find the barycentre of the support of the density. Let's also emphasize that the Fréchet mean is defined in every metric space and therefore in particular on every Riemannian manifold. The Fréchet mean is an important concept and used throughout the literature on the topic of shape statistics.

Existence and uniqueness : Karcher expectation

It is obvious that the existence of the mean in the sense of Fréchet is not guaranteed as it is the solution of a minimization and the global minima might not be reached. Generally, the result of the minimization is not a single element but rather a set of elements since we might encounter several minima. This is typically the case when the random element has several central values. Equivalently, this can be seen as multi-modal distribution functions representing a variable centered around several values. With the definition of the Fréchet expectation we cannot define all modes because we keep only the minimal value. In order to cope with this constraint, Karcher [Karcher 1977] proposed to consider local minima of variance $\sigma_x^2(y)$ instead of solely global minima. As there are usually more local minima than global, the set of means in the sense of Fréchet is a subset of the means in the sense of Karcher. Also, remember that we only need second order derivatives to characterize the solutions at the potential points. Using this extended definition, Karcher [Karcher 1977] and Kendall [Kendall 1990] stated certain conditions about the manifold and the distribution to guarantee existence and uniqueness of the mean.

Theorem 20 (Existence and uniqueness of Karcher mean)**.** *Let x be random variable(primitive) with density p_x*

(Karcher, 1977) If the support of p_x is included in the geodesic ball $\mathcal{B}_r(y)$ and the ball $\mathcal{B}_{2r}(y)$ with two times the radius is still geodesic and regular then the variance $\sigma_x^2(z)$ is a convex function of z. Therefore we have a unique critical point in $\mathcal{B}_r(y)$ which is necessarily the Kracher mean.

(Kendall, 1990) If the support of p_x is included in the geodesic ball $\mathcal{B}_r(y)$ then there exists one and only one Karcher mean of x in $\mathcal{B}_r(y)$.

These conditions are quite restricting but assure a good behavior of the mean.

4.2.4 Pre-Image

Given a point Υ in the feature space, the pre-image problem consists of finding a point $\hat{x} \in \mathcal{X}$ in the input space such that $\Psi(\hat{x}) = \Upsilon$,

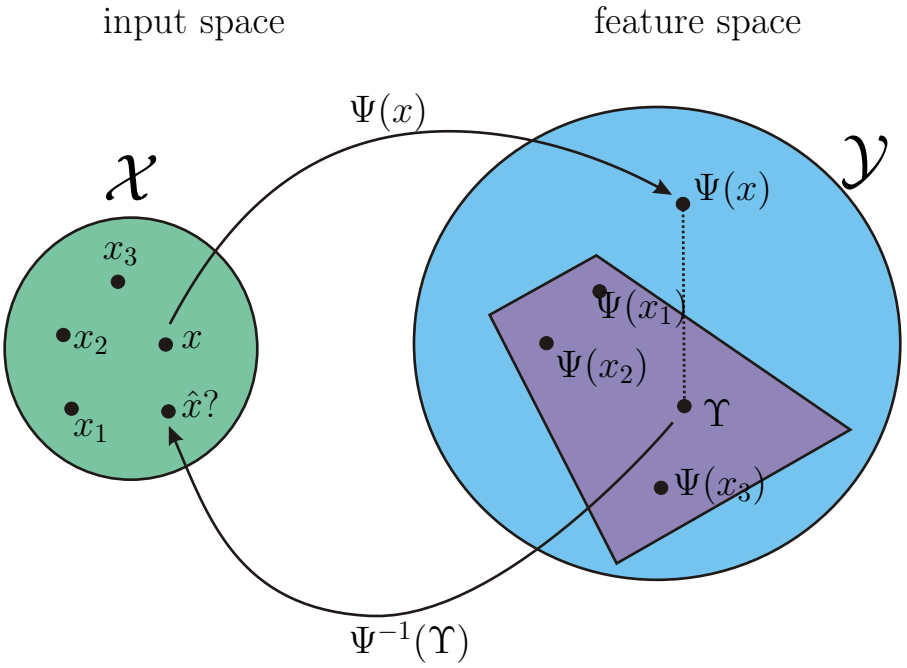


Figure 4.4: Problem configuration: We are looking for the pre-image $\hat{x}$ of Υ in the input space $\mathcal{X}$

i.e. the pre-image of Υ. The exact pre-image might not exist (when it exists, it might also not be unique) and the pre-image problem is ill-posed [Arias 2007, Dambreville 2006, Kwok 2004, Mika 1999]. To circumvent this problem, one usually settles for an approximate solution and searches for a pre-image that optimizes a given optimality criterion in the feature space. The pre-image problem has received a lot of attention in kernel methods [Arias 2007, Dambreville 2006, Kwok 2004, Mika 1999] and different optimality criteria have been proposed, e.g.:

$$\hat{x} = \arg\min_{z \in \mathcal{X}} \|\Psi(z) - \Upsilon\|^2 \tag{4.6}$$

This problem, known as the pre-image problem, is the problem to be addressed in this chapter.

4.2.5 Out-of-sample extension

In general, the mapping Ψ (c.f section 4.2.1 and section 3.2.6), also referred to as an embedding, is only known over the training set. The

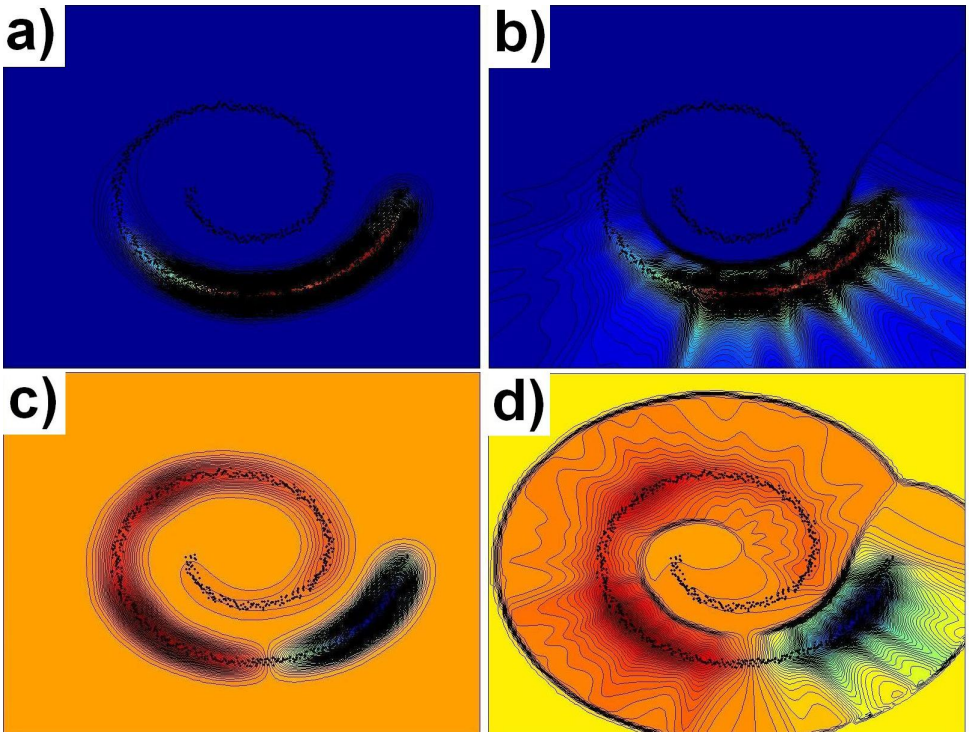


Figure 4.5: Nyström extension of the first embedding coordinate (from blue corresponding to smaller coordinates to red corresponding to larger ones) provided by the leading eigenvector in different kernel methods : Kernel PCA without (a) and with normalization [Arias 2007] (b), Kernel PCA in centered feature space without (c) and with normalization (d)

extension of the mapping to new input points is of primary importance for kernel based methods. These methods success significantly depend on the "accuracy" of the extension. This problem, referred to as the *out-of-sample* problem, is often solved using the popular Nyström extension method [Arias 2007, Bengio 2004, Lafon 2006a]. Instead of recomputing the whole embedding, which can be costly for very large datasets because it involves a spectral decomposition, the problem is solved through a method borrowed from numerical analysis [Baker 1996]. It is obvious that the extension depends on the data and recomputing the whole embedding with the new datum would yield a different embedding. But in general the approximation works well and is used throughout the literature.

Nyström approximation

As we have seen in the previous section computing diffusion maps involves the computation of eigenvectors of a symmetric matrix of size $N \times N$. This takes a considerable amount of time. Thus the question of how to extend an empirical function outside the range of samples becomes important in order to avoid recomputing the empirical function when a new datum is input. In the machine learning community this problem is often solved with the Nyström method which numerically approximates eigenfunctions of integral equation of the form

$$\int_a^b w(x,y)\Psi(y)dy = \lambda\Psi(x),$$

with $w(.,.)$ being a suitable kernel function. This eigenfunction problem can be approximated by evaluating the equation at evenly spaced points $\eta_1, \eta_2, \ldots, \eta_n$ on the interval $[a,b]$ with the following quadrature rule

$$\frac{(b-a)}{n}\sum_{j=1}^{n} w(x,\eta_j)\hat{\Psi}(\eta_j) = \lambda\hat{\Psi}(x). \tag{4.7}$$

Here $\hat{\boldsymbol{\psi}}(x)$ is an approximation to the true $\Psi(x)$. In order to solve equation (4.7), we set $x = \eta_i$

$$\frac{(b-a)}{n}\sum_{j=1}^{n} w(\eta_i,\eta_j)\hat{\Psi}(\eta_j) = \lambda\hat{\Psi}(\eta_i).$$

Let $[a,b]$ be$[0,1]$, we rewrite the system of equations in matrix form as

$$\mathbf{W}\hat{\mathbf{\Psi}} = n\hat{\mathbf{\Psi}}\Lambda.$$

Where $\mathbf{W}_{ij} = w(\eta_i,\eta_j)$ and $\hat{\mathbf{\Psi}} = \left[\hat{\Psi}_1,\ldots,\hat{\Psi}_n\right]$ are the n eigenvectors of $\mathbf{W}$ with associated eigenvalues $\{\lambda_1,\ldots,\lambda_n\}$. Then substituting back into Equation (4.7) yields the desired Nyström extension of x for each coordinate

$$\hat{\Psi}_i(x) = \frac{1}{n\lambda_i}\sum_{j=1}^{n} w(x,\eta_j).\hat{\Psi}(\eta_j) \tag{4.8}$$

This expression allows us to extend an eigenvector computed for a set of sample points to an arbitrary point x in terms of a linear combination of the eigenvectors weighted by the kernel $w(.,\eta_j)$.

In addition, the reverse mapping, from the feature space back to the input space, is often required. After operations are performed in feature space (these operations necessitate the extension of the mapping), corresponding data points in input space often need to be estimated. This problem, known as the *pre-image* problem, is the problem to be addressed in this chapter.

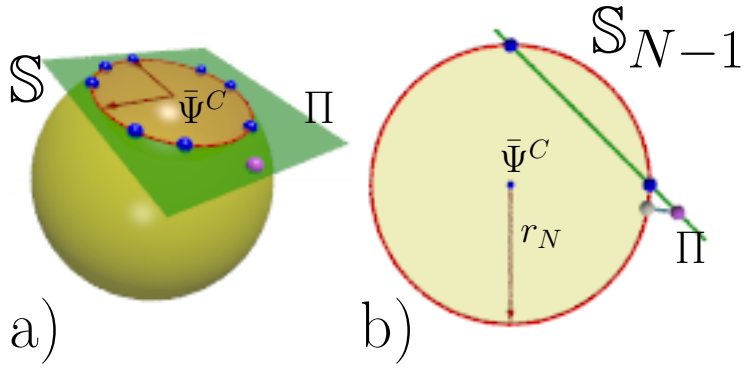


Figure 4.6: a) Visualization of the feature points(blue) geometry in $\mathcal{Y}$ and the affine subspace(red circle); b) Affine subspace $\mathbb{S}_{p-1}$

4.3 Normalization in Gaussian Kernel PCA

Kernel PCA is achieved by applying a principal component analysis on the mapped training samples. PCA computes an eigen-decomposition of a kernel matrix deduced from the adjacency matrix $\mathbf{W}$. Before applying PCA, the data is centered at the origin. In Kernel PCA the mean of the mapped input points is not known. Therefore, to simplify, one often assumes that the mapped training points $\Psi(\mathbf{x}_i)$ are already centered in the feature space $\mathcal{Y}$ and "incorrectly" diagonalize the adjacency matrix $\mathbf{W}$ [Arias 2007, Mika 1999]. Although simpler to understand, the resulting presentation of kernel methods misses some important points.

Our analysis of the kernel PCA methods studies in detail the centering of the data and underlines some important properties of the geometry of the mapped data induced by the kernel. We focus on the Gaussian kernel $w(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-d^2_{\mathcal{X}}(\mathbf{x}_i, \mathbf{x}_j)/2\sigma^2\right)$, with σ estimated as the median of all the distances between all training points [Arias 2007, Lafon 2006b]. In accordance with the geometry induced by the Gaussian kernel, we highlight some non-trivial elements and rephrase some pre-image methods in a centered feature space[Arias 2007]. A comparison based on numerical experiments demonstrates the superiority of our pre-image methods using a careful normalization in a centered feature space.

Let $\{\mathbf{x}_1, \cdots, \mathbf{x}_N\} = \mathbf{X}$ be a set of training data in the input space $\mathcal{X}$. Kernel PCA computes the principal components of mapped features in the feature space $\mathcal{Y}$. The mapping can be explicitly computed by the eigen-decomposition of a kernel matrix deduced from the adjacency matrix $\mathbf{W}$. The

coefficients of the adjacency matrix $\mathbf{W}$ are a measure of similarity between samples. Typically, the kernel function $w(.,.)$ is a decreasing function of the distance $d_{\mathcal{X}}$ between training points $\mathbf{x}_i$ and $\mathbf{x}_j$.

Here, we focus on the Gaussian kernel. The Gaussian kernel has the important property of implicitly mapping the training points onto the unit sphere of $\mathcal{Y}$, since $\|\Psi(\mathbf{x}_i)\|^2 = \langle \Psi(\mathbf{x}_i), \Psi(\mathbf{x}_i) \rangle_{\mathbb{H}} = \mathbf{W}_{ii} = 1$. This important normalization property has been extensively used by Arias and coworkers [Arias 2007] to improve the "accuracy" of previous pre-image methods [Dambreville 2006, Kwok 2004, Mika 1999]. Furthermore, we state the Kernel PCA methodology in centered space and shows that a finer degree of normalization can be achieved by considering the geometry of the mapped features.

Let $\bar{\Psi} = \frac{1}{N} \sum_{\mathbf{x}_k \in \Gamma} \Psi(\mathbf{x}_k)$ and Ψ^C denote the centered mapping, i.e. $\Psi^C(\mathbf{x}_i) = \Psi(\mathbf{x}_i) - \bar{\Psi}$. The mapping $\mathbf{\Psi}^C$ can be computed by the eigen-decomposition of a centered kernel $\mathbf{W}^C$ [Schölkopf 1999]:

$$\mathbf{W}^C = \mathbf{HWH} = \mathbf{\Psi}^C \Lambda^C (\mathbf{\Psi}^C)^T = \mathbf{\Psi}^C \sqrt{\Lambda^C} (\mathbf{\Psi}^C \sqrt{\Lambda^C})^T,$$

where $\mathbf{H}$ is the centering matrix $\mathbf{H} = \mathbf{Id}_N - \frac{1}{N} \mathbb{1}_N \mathbb{1}_N^T$ and $\mathbf{\Lambda}^C = \text{diag}\{\lambda_1^C, \cdots, \lambda_N^C\}$ with $\lambda_1^C \geq \cdots \geq \lambda_{N-1}^C > \lambda_N^C = 0$. We denote $\hat{\mathbf{\Lambda}} = \text{diag}\{\lambda_1^C, \cdots, \lambda_{N-1}^C\}$ and $\hat{\mathbf{\Psi}} = (\Psi_1^C, \cdots, \Psi_{N-1}^C)$, the mapping is obtained as:

$$\Psi^C : \mathcal{X} \to \mathbb{R}^{N-1}, \ \mathbf{x}_i \mapsto \sqrt{\hat{\mathbf{\Lambda}}} \hat{\mathbf{\Psi}}^T \mathbf{e}_i^C. \tag{4.9}$$

The canonical basis $\{\mathbf{e_1^C}, \cdots, \mathbf{e_{N-1}^C}\}$ of $\mathbb{R}^{N-1}$, defined formally by $\mathbf{e}_k^C = \frac{1}{\sqrt{\lambda_k^C}} \sum_{\mathbf{x}_i \in \Gamma} \Psi_k^C(\mathbf{x}_i) \Psi^C(\mathbf{x}_i)$, captures the variability of the point cloud of training samples. Projection of a new test point $\mathbf{x} \in \mathcal{X}$ onto the k^{th}-canonical vector $\mathbf{e}_k^C$ in the feature space can be shown to be:

$$\beta_k(\mathbf{x}) = \left\langle \mathbf{e}_k^C, \Psi^C(\mathbf{x}) \right\rangle = (e_k^C)^T \hat{\mathbf{\Lambda}}^{-\frac{1}{2}} \hat{\mathbf{\Psi}}^T p_{\mathbf{x}}^C, \tag{4.10}$$

$$\text{where } p_{\mathbf{x}}^C(\mathbf{x}_j) = \mathbf{H}(p_{\mathbf{x}} - \tfrac{1}{N} \mathbf{W} \mathbb{1}_N)(\mathbf{x}, \mathbf{x}_j). \tag{4.11}$$

$p_{\mathbf{x}}^C(\mathbf{x}_j)$ is the extended mapping in centered feature space computed by centering the kernel vector $p_{\mathbf{x}} = [w(\mathbf{x}, \mathbf{x}_1), \ldots, w(\mathbf{x}, \mathbf{x}_N)]^T$. This way of extending embedding coordinates to new test points has been used implicitly[Dambreville 2006, Kwok 2004, Mika 1999] or explicitly[Arias 2007] in kernel methods[Ham 2003]. Projecting a new test point $\mathbf{x} \in \mathcal{X}$ onto the subspace spanned by the first d^C vectors $\{\mathbf{e}_1^C, \cdots, \mathbf{e}_{d^C}^C\}$ (i.e. $P_{d^C}(\Psi^C(s)) = \sum_{1 \leq k \leq d^C} \beta_k(\mathbf{x}) \mathbf{e}_k^C$) does not require the explicit computation of the mapping $\bar{\Psi}^C(\mathbf{x})$ since equation 4.10 can only be written in terms of the kernel.

Working in a centered feature space, some important (often mistakenly ignored) comments follow. We show that the fundamental property of the mapped input points $\|\Psi(\mathbf{x}_i)\|^2 = w(\mathbf{x}_i, \mathbf{x}_i) = 1$ can be greatly improved in a centered feature space. To do so, we define the mean in feature space $\bar{\Psi}^C(\in \mathbb{R}^{N-1}) = \frac{1}{N}(\mathbf{\Lambda}^C)^{(\frac{1}{2})}(\mathbf{\Psi}^C)^T\mathbf{H}\mathbf{W}\mathbb{1}_N$ and consider some properties of the feature points mapped under:

$$\tilde{\Psi}^C : \mathcal{X} \to \mathbb{R}^{N-1}, \mathbf{x} \mapsto \bar{\Psi}^C + \Psi^C(\mathbf{x}). \tag{4.12}$$

Under this mapping, the training samples verify: $\left\langle \tilde{\Psi}^C(\mathbf{x}_i), \tilde{\Psi}^C(\mathbf{x}_j) \right\rangle = w(\mathbf{x}_i, \mathbf{x}_j) - \bar{\Psi}^C_N$, with $0 \leq \bar{\Psi}^C_N \leq 1$. The adjacency matrix $\mathbf{W}$ therefore gives (up to an additional factor $(\bar{\Psi}^C_N)^2$ the inner product between two points in the feature space under the mapping $\tilde{\Psi}^C$. The constant $\bar{\Psi}^C_N$ has a simple geometric interpretation. In the feature space, the N non-centered training points, which belong to the unit sphere, define an affine space that is isomorphic to $\mathbb{R}^{N-1}$. This affine space, spanned by the vectors $\{\mathbf{e}^C_1, \cdots, \mathbf{e}^C_{N-1}\}$, is at distance $\bar{\Psi}^C_N$ from the origin $\mathbf{0}$. Consequently, feature points mapped under $\tilde{\Psi}^C : s \mapsto \bar{\Psi}^C + \Psi^C(s)$ all belong to a hypersphere of $\mathbb{R}^{N-1}$ of radius $r_N = \sqrt{1 - \bar{\Psi}^{C2}_N}$, i.e. $\mathbb{S}_{N-1}(\mathbf{0}, r_N)$. This implies that, for all training sample $\mathbf{x}_i \in \mathbf{X}$, we have $\|\tilde{\Psi}^C(\mathbf{x}_i)\| = r_N$. This normalization property of training samples is stronger than the usual property $\|\Psi(\mathbf{x}_i)\| = 1$ and will prove important in the next section[1]. In particular, this allows us to rephrase some pre-image methods, such as[Arias 2007], in a centered feature space, leading to better results (section 5.4). Finally, we note that the mapping Ψ can be deduced from Ψ^C by $\Psi : \mathcal{X} \to \mathbb{R}^N, \mathbf{x} \mapsto (\tilde{\Psi}^C(\mathbf{x})^T, \bar{\Psi}^C_N)^T$.

4.3.1 Pre-image in Kernel PCA

Given a point in the feature space Υ, the pre-image problem consists in finding a point $\mathbf{x} \in \mathcal{X}$ in the input space such that $\Psi(\mathbf{x}) = \Upsilon$, i.e. the pre-image of Υ. Previous methods [Arias 2007, Dambreville 2006, Kwok 2004, Mika 1999] propose certain optimality criteria to solve this problem. Although most of those are based on the property $\|\Psi(\mathbf{x}_i)\|^2 = 1$, significant improvement can be attained by considering that the mapped feature points $\tilde{\Psi}^C(\mathbf{x}_i)$ belong to the hypersphere $\mathbb{S}_{N-1}(\mathbf{0}, r_N)$ (or equivalently stated that $\|\tilde{\Psi}^C(\mathbf{x}_i)\| = r_N$). In particular, we insist on the fact that the popular normalization $\frac{\Psi(\mathbf{x})}{\|\Psi(\mathbf{x})\|}$ is not equivalent to the normalization $\frac{\tilde{\Psi}^C(\mathbf{x})}{\|\tilde{\Psi}^C(\mathbf{x})\|}$. In more detail, note that after

[1]Note that to compute the radius value r_N (or, equivalently, the distance $\bar{\Psi}^C_N$), it is sufficient to compute $\|\tilde{\Psi}^C(x_i)\|$ for only one of the training samples $\mathbf{x}_i \in \Gamma$.

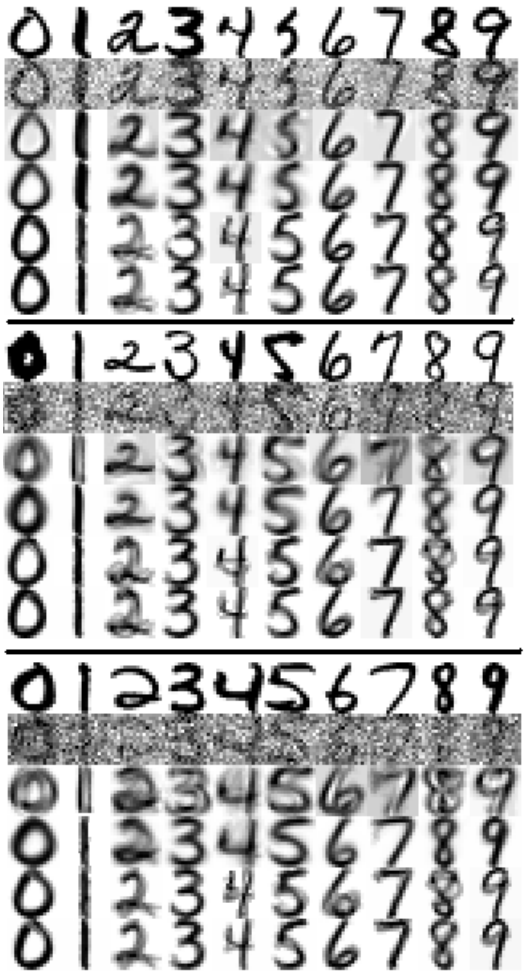


Figure 4.7: Digit images corrupted by additive Gaussian noise (from top to bottom, $\sigma = 0.25, 0.45, 0.65$). The different rows respectively represent: the original digits and corrupted digits; different reconstruction methods: [Dambreville 2006] ; [Dambreville 2006] with normalization ; [Kwok 2004] ; [Kwok 2004] with normalization.

normalization by the former criterion, a feature point does not any longer belong to the affine space defined by the N-training points. This behavior can also be seen in Figure 4.6b), which is the two dimensional visualization of the affine subspace(red circle) in Figure 4.6a). Figure 4.6a) shows the sphere $\mathbb{S}$ and the layout of feature points on $\mathbb{S}$. The extended mapping of a new input point does not lie on the sphere(visualized as a purple point). As can be clearly seen the normalization as proposed in [Arias 2007] projects the feature point(purple) onto the sphere(white). But the projected point does not lie in the span. This is clearly problematic as the principal modes of variations span only this affine space. The later normalization is the correct one and should be advantageously used. Therefore, we capitalize on our careful analysis of KPCA and define the different optimality criteria in centered feature space:

$$\text{Distance:}\hat{\mathbf{x}} = \arg\min_{z\in\mathcal{X}} \|\tilde{\Psi}^C(z) - \tilde{\Upsilon}^C\|^2, \tag{4.13}$$

$$\text{Collinearity:}\hat{\mathbf{x}} = \arg\max_{z\in\mathcal{X}} \left\langle \frac{\tilde{\Psi}^C(z)}{\|\tilde{\Psi}^C(z)\|}, \frac{\tilde{\Upsilon}^C}{\|\tilde{\Upsilon}^C\|} \right\rangle, \tag{4.14}$$

where $\tilde{\Upsilon}^C = \bar{\Psi}^C + \Upsilon^C$. Recently, Arias and coworkers[Arias 2007] have shown the connections between the out-of-sample and the pre-image problems and proposed a normalized optimality criterion addressing the important lack of normalization in kernel methods:

$$s = \arg\min_{z\in\chi} \|\tilde{\Psi}^C(z) - \bar{\Upsilon}\|^2 \text{ with } \bar{\Upsilon} = r_N \frac{\tilde{\Upsilon}^C}{\|\tilde{\Upsilon}^C\|}. \tag{4.15}$$

Instead of directly solving the pre-image in equation 4.15, they first estimate the optimal kernel vector as a standard least-squares problem $p_\Upsilon^C = \hat{\mathbf{\Psi}}\sqrt{\hat{\mathbf{\Lambda}}}(\bar{\Upsilon} - \bar{\Psi}^C)$ and then use previous methods[Dambreville 2006, Kwok 2004] to estimate the optimal pre-image.

4.3.2 Application in image denoising

In order to validate the proposed algorithm, we run experiments on real world data. We test our pre-image algorithm on the denoising of noisy images and compare our approach to previous methods. The computation of Kernel PCA is done using the Gaussian kernel $\exp\left(-d^2_{\mathcal{X}}(\mathbf{x}_i, \mathbf{x}_j)/2\sigma^2\right)$ where σ is the median over all distances between points[Arias 2007].

To test the performance of our approach on the task of image denoising, we apply the algorithm on the USPS dataset of handwritten digits[2]. We show that our normalization method improves two recent state-of-the-art algorithms [Dambreville 2006], [Kwok 2004]. Therefore, we form two training

[2]The USPS dataset is available from http://www.kernel-machines.org.

sets composed of randomly selected samples (60 and 200 respectively) for each of the ten digits. The test set is composed of 60 images randomly selected and corrupted by some additive Gaussian noise at different noise levels. The process of denoising simply amounts to estimating the pre-images of the feature vectors given by the Nyström extension of the noisy samples. In the case of Kernel PCA, we use the first $d^C = 8$ eigenvectors $\{\mathbf{e}_1^C, \cdots, \mathbf{e}_{d^C}^C\}$ to compute projections in feature space.

σ^2	[Dambreville 2006]	[Dambreville 2006] improved	[Kwok 2004]	[Kwok 2004] improved
0.25	10.39	11.71	15.88	16.18
0.45	10.22	12.54	15.80	16.35
0.65	9.95	12.72	15.54	16.32
0.85	9.52	12.58	15.31	16.28
0.25	12.11	12.14	15.83	15.89
0.45	10.22	12.54	15.80	16.35
0.65	9.95	12.72	15.54	16.32
0.85	9,24	12.59	15.31	16.28

Table 4.1: Average PSNR (in dB) of the denoised images corrupted by different noise level. Training set is composed of 60 samples (first 4 rows) and 200 samples (last 4 rows). The first and third column show the denoising results without and the second and last columns with the normalization we proposed.

Figure 4.7 displays some of the computed pre-images using different methods. Table 4.3 shows a quantitative comparison between different methods based on the pixel-signal-to-noise ratio(PSNR). Our normalisation method visually and quantitatively improves both pre-image methods. The results confirm that the new normalisation criterion in centered features space (second and fourth column) yields better results than previous pre-image methods (first and third column).

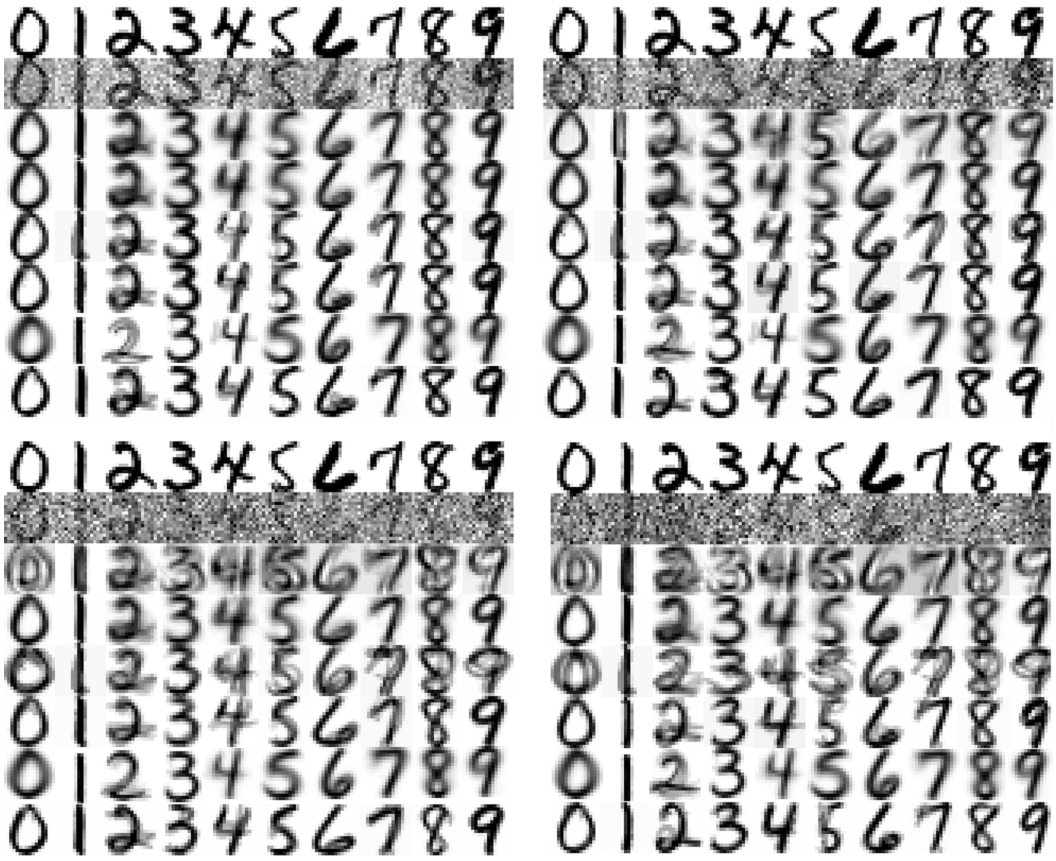


Figure 4.8: Digit images corrupted by additive Gaussian noise (from left to right, $\sigma^2 = 0.25, 0.45, 0.65, 0.85$). The different rows respectively represent, from top to bottom: the original digits; the corrupted digits; denoising with [Dambreville 2006] ; with [Dambreville 2006]+[Arias 2007] ; with [Kwok 2004] ; with [Kwok 2004]+[Arias 2007] ; with [Carreira-Perpiñan 2007]; with our Karcher mean based method. See table 4.3 for quantified results

4.4 Pre-Image as Karcher means

4.4.1 Related work

Statistical methods for shape processing are very common in computer vision. A seminal work in this direction was published by Leventon et.al. [Leventon 2000] adding statistical knowledge into energy based segmentation methods. Their method captures the main modes of variation by performing a PCA on the set of shapes. With the apparition of non-linear methods, Dambreville et.al [Dambreville 2006] developed a method for shape denoising based on Kernel PCA. So did Kwok et.al. [Kwok 2004] in the context of image denoising. Both methods compute a projection of the noisy datum onto a low dimensional space. In [Lu 2008, Carreira-Perpiñan 2007] the authors propose another kernel method for data denoising, the so called *Laplacian Eigenmaps Latent Variable Model* (LELVM), a probabilistic method. This model provides a dimensionality reduction and reconstruction mapping based on linear combinations of input samples. LELVM performs well on motion capture data but fails on complex shapes (see Fig. 4.8). Additionally, we would like to mention the work of Pennec [Pennec 2006] and Fletcher [Davis 2007] modeling the manifold of shapes as a Riemannian manifold and the mean of such shapes as a Karcher mean [Karcher 1977]. Their methodology is used in the context of computational anatomy to solve the average template matching problem. Closer to our work is the algorithm proposed by Etyngier et. al. [Etyngier 2007b]. They use diffusion maps as a statistical framework for non linear shape priors in segmentation. They augment an energy functional by a shape prior term. Contrary to us, they do not compute a denoised shape but propose an additional force toward a rough estimate of it.

4.4.2 Learning a set of shapes

Let $\Gamma = \{s_1 \cdots s_N\}$ be N independent random points of a d-dimensional manifold $\mathcal{X}$ locally sampled under a certain density $q_{\mathcal{X}}(s)$ ($d << N$). The manifold $\mathcal{M}$ is assumed to be a smooth finite-dimensional sub-manifold embedded in a (potentially infinite-dimensional) space $\mathcal{S}$. The density $q_{\mathcal{M}}(s)$ is unknown and might not be uniform. In this work, we consider more general spaces than the traditional Euclidean space $\mathbb{R}^n$ and only assume that the input space $\mathcal{S}$ is equipped with a distance $d_{\mathcal{S}}$.

Also note that methods like LLE or Laplacian Eigenmaps do not provide an explicit metric which is crucial for the contribution in this thesis. We now tackle the problem of pre-image computation using diffusion maps.

We push the manifold interpretation and define the pre-image of $\Upsilon \in \mathbb{R}^N$ as the point $s = \Psi_{|\mathcal{X}}^{-1}(\Upsilon)$ in the manifold $\mathcal{X}$ such that $\Psi(s) = \Upsilon$. Although diffusion maps extract the global geometry of the training set and define a robust notion of proximity, they cannot permit the estimation of the manifold between training samples, i.e. the local geometry of the manifold is not provided. Following [Etyngier 2007a], we propose to approximate the manifold as the set of Karcher means [Karcher 1977] interpolating between correctly chosen subsets of $d+1$ sample points, d being the fixed dimension reduction parameter. Usually it is chosen by observing the eigenvalues of the eigenvectors. As mentioned in section 3.2.6 only a few eigenvectors are needed to properly approximate the diffusion distance. And the parameter d is exactly the number of eigenvectors retained. From a dimensionality reduction point of view, this parameter corresponds to the degree of freedom in the data set but cannot be computed automatically and therefore must be guessed. In [Etyngier 2007a], these subsets are the Delaunay simplices of a d-dimensional Delaunay triangulation of the sample points. In practice, this limits d to small integer values. Here, we simply exploit the Euclidean nature of the feature space: for a given Υ, we choose the interpolating subset as its $d+1$ nearest neighbors with respect to the diffusion distance $\mathcal{D}$. We then define the pre-image $s = \Psi_{|\mathcal{X}}^{-1}(\Upsilon)$ as a Karcher mean that minimizes the mean-squared criterion:

$$s = \arg\min_{z \in \mathcal{S}} \|\Psi(z) - \Upsilon\|^2 \tag{4.16}$$

4.4.3 Shape interpolation as Karcher mean

Given a set of neighboring points $\mathbb{N} = \{s_1, \cdots, s_{d+1}\}$ (*i.e.* neighboring for the diffusion distance $\mathcal{D}$), we assume that the manifold $\mathcal{X}$ can be locally described (*i.e.* between neighboring samples) by a set of weighted-mean samples $\{s_\Theta\}$ that verifies:

$$s_\Theta = \arg\min_{z \in \mathcal{S}} \sum_{1 \le i \le d+1} \theta_i d_{\mathcal{S}}(z, s_i)^2, \tag{4.17}$$

where $d_{\mathcal{S}}$ is the distance in the input space and $\left(\theta_i \ge 0,\ \sum_{i=1}^{d+1} \theta_i = 1\right)$. The coefficients $\Theta = \{\theta_1, \ldots, \theta_{d+1}\}$ are the barycentric coefficients of the point s_Θ with respect to its neighbors $\mathbb{N}$ in $\mathcal{S}$. Proposed by Charpiat el al.[Charpiat 2005], this model has proven to give natural shape interpolations, compared to linear approximations. One classical choice is the area of the

symmetric difference between the regions bounded by the two shapes:

$$d_{SD}(s_1, s_2) = \left(\frac{1}{2} \int (\chi\Omega_1(x) - \chi\Omega_2(x))^2 dx \right)^{\frac{1}{2}}. \tag{4.18}$$

This distance was recently advocated by Solem in [Solem 2006] to build geodesic paths between shapes. But the drawback is that this distance yields no unique geodesics [Serra 1998](Appendix A). Another definition has been proposed [Leventon 2000, Rousson 2002, Charpiat 2005], based on the representation of a curve in the plane, of a surface in 3D space, by its signed distance function.In this context, the distance between two shapes can be defined as the L^2-norm or the Sobolev $W^{1,2}$-norm of the difference between their signed distance functions. Let us recall that $W^{1,2}(\Omega)$ is the space of square integrable functions over Ω with square integrable derivatives:

$$d_{L^2}(s_1, s_2) = \|\mathbb{D}_{s_1} - \mathbb{D}_{s_2}\|_{L^2} \ ,$$

$$d_W^{1,2}(s_1, s_2) = \|\mathbb{D}_{s_1} - \mathbb{D}_{s_2}\|_{L^2} + \|\nabla\mathbb{D}_{s_1} - \nabla\mathbb{D}_{s_2}\|_{L^2}$$

where $\mathbb{D}_{s_i}$ denotes the signed distance function of shape s_i $(i = 1, 2)$, and $\nabla\mathbb{D}_{s_i}$ its gradient(c.f. section).

4.4.4 Pre-Image and manifold interpolation

We propose to define the pre-image of a target point Υ in the feature space, as the point s_Θ that minimizes the energy $E_\Psi(s_\Theta) = \|\Psi(s_\Theta) - \Upsilon\|^2$, s_Θ being expressed as a Karcher mean for the neighborhood $\mathbb{N}$ made of the $d+1$ samples of $\mathbf{X}$ which embedding are the $d+1$-closest neighbors of Υ in the feature space equipped with $\mathcal{D}$:

$$\Psi^{-1}_{|\mathcal{M}}(\Upsilon) = \arg\min_\Theta \|\Psi(s_\Theta) - \Upsilon\|^2, \tag{4.19}$$

$$\text{with } \mathbb{N} \subset \Gamma,\ \mathbb{N} = \{d+1 \text{ closest neighbors of } \Upsilon \text{ in the feature space}\} \tag{4.20}$$

When the input space is some Euclidean space $\mathbb{R}^n$ with its traditional L_2-norm, this indeed amounts to assuming that the manifold $\mathcal{X}$ is piecewise-linear (*i.e.* linearly interpolated between neighboring training samples). For shapes, we will see that this yields natural pre-images. By simple extension, we define the projection of any new test sample s on the manifold $\mathcal{X}$ by $\Pi_\mathcal{X}(s) = \Psi^{-1}_{|\mathcal{X}}(\Psi(s))$.

4.4.5 Implementation issues

The pre-image $\Psi^{-1}_{|\mathcal{M}}(\Upsilon)$ is computed by gradient descent. Instead of optimizing over Θ, we use a descent over s_Θ itself (equation 4.19), constraining it to remain a Karcher mean (equation 4.17). This boils down to projecting the deformation field $\nabla_s E_\psi$ onto the tangent space $\mathbb{T}_{s_\Theta}\mathcal{X}$ of $\mathcal{X}$ at point s_Θ. Note that to compute this tangent space, we are implicitly assuming that the space $\mathcal{X}$ has a manifold structure, in particular that the tangent space $\mathbb{T}_{s_\Theta}\mathcal{X}$ of $\mathcal{X}$ at location s_Θ (*i.e.* the space of local deformations around s_Θ) is equipped with an inner product that we denote $\langle .|. \rangle_{\mathcal{X}}$.

The optimality condition of equation 4.17 is:

$$\forall \vec{\beta} \in \mathbb{T}_{s_\Theta}\mathcal{X},\ \sum_{i=1}^{d+1} \theta_i d_i \langle \nabla_s d_i | \vec{\beta} \rangle_{\mathcal{X}} = 0,$$

where we denote $\mathbb{N} = \{s_1, ..., s_d + 1\}$ and $d_i = d_{\mathcal{S}}(s_\Theta, s_i)$. In order to recover the tangent space $\mathbb{T}_{s_\Theta}\mathcal{M}$ at s_Θ, one needs to relate the d-independent modes of variations of the coefficient Θ (remember that $\sum_{i=1}^{d+1} \theta_i = 1$) with local deformation fields $\vec{ds_\Theta} \in \mathbb{T}_{s_\Theta}\mathcal{S}$. To a small variation of the barycentric coefficients $\Theta \to \Theta + \vec{d\Theta}$, corresponds a small deformation of the sample $s_\Theta \to s_\Theta + \vec{ds_\Theta}$. Differentiating the optimality condition with respect to Θ and s_Θ provides the relation between $\vec{d\Theta}$ and $\vec{ds_\Theta}$ (see Appendix A for more detail). For example, when the input space is taken to be the Euclidean space, *i.e.* $\mathcal{S} = \mathbb{R}^n$, we obviously obtain $\vec{ds_\Theta} = \sum_1^{d+1} d\theta_i s_i$. Remembering $\sum_1^{d+1} d\theta_i = 0$ and fixing the $d\theta_i$ appropriately, we can recover $\mathbb{T}_{s_\Theta}\mathcal{X}$. Therefore we optimize for s_Θ without explicitly computing Θ. The gradient descent generates a family of samples $s : \tau \in \mathbb{R}^+ \mapsto s(\tau) \in \mathcal{X}$ such that

$$s(0) = s_0,\ \frac{ds}{d\tau} = -\vec{v}_{\mathcal{X}}(s_\tau),$$

with $s_0 \in \mathbb{N}$ (in practice, the nearest neighbor of Υ). The velocity field $\vec{v}_{\mathcal{X}}(s_\tau)$ is the orthogonal projection of the deformation field $\nabla_{s_\tau} E_\Psi = (\Psi(s_\tau) - \Upsilon)^T \Lambda \Psi^T \nabla_{s_\tau} p_{s_\tau}$ onto the tangent space $\mathbb{T}_{s_\tau}\mathcal{X}$. Here Λ is a diagonal matrix of eigenvalues and Ψ are the corresponding eigenvectors. Note that before projecting onto $\mathbb{T}_{s_\tau}\mathcal{X}$ we first orthogonalize the tangent space by using Gram-Schmidt. In the case of the L_2-norm the Θ's can be easily recovered. When using a different distance function such as the symmetric difference or the Sobolev $W^{1,2}$-norm, one needs to additionally solve a system of linear equations in each step of the gradient descent (Appendix A).

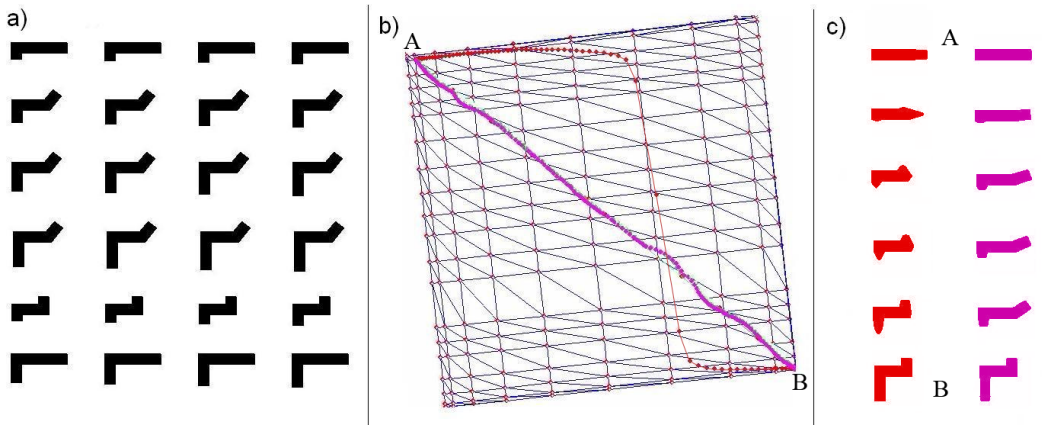


Figure 4.9: Synthetic sample of 200 articulated and elongated shapes. From left to right: (a) a subset of the sample. (b) triangulated two-dimensional embedding computed using diffusion maps and a gradient descent from an initial shape to a target one, without (red dots) and with (purple dots) remaining on the interpolated manifold. (c) Some shapes of the resulting evolution (left column: without projection, right column: with projection.

4.4.6 Results

In order to validate the proposed method, we run several experiments on real and synthetic data. First, we test the Karcher mean interpolation with the reconstruction problem of occluded 3D medical shapes [Dambreville 2006]. In a second experiment we validate the purpose of the projection of the gradient onto the tangent space. Finally, a third experiment demonstrates the superiority of our method for a standard denoising problem on images.

Remaining on the manifold

To validate both the Karcher mean modeling of the manifold and our projecting constraint (section 4.4.5), we generate a set of 200 synthetic shapes parameterized by an articulation angle and a scaling parameter (Fig. 4.9a). The corresponding embeddings are shown Fig. 4.9b. Choosing two distant shapes A and B, we compute a path $s(\tau)$ from A to B by means of a gradient descent starting from $s(0) = A$ and minimizing $d_{\mathcal{S}}(s(\tau), B)$. Fig. 4.9c and 4.9b show in red the intermediate shapes and the corresponding embeddings.

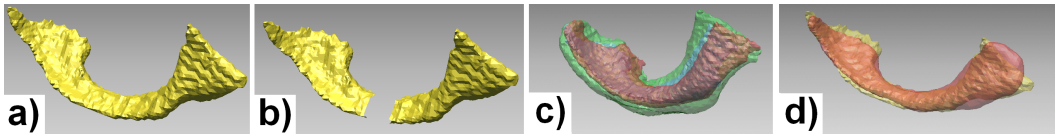


Figure 4.10: Interpolation using Karcher means for 39 three-dimensional sample shapes. From left to right: a) a new shape not in the given sample b) the same shape with an occlusion c) the 3 nearest neighbors of the corrupted shape according to the diffusion distance (in red, green and blue) d) the original shape (in yellow) and our interpolation (in red). See text for quantitative results.

In purple the same path is shown when projecting the gradient in order to remain on the manifold. Observe in that case, how the intermediate shapes look more like the original samples. Note also that when remaining on $\mathcal{M}$, the interpolating path is almost a straight line with respect to the diffusion distance.

Projection and manifold as Karcher means

Here we test the validity of using Karcher means as a manifold interpolation model. We consider the space of two-dimensional surfaces embedded in $\mathbb{R}^3$. For such a general space, many different definitions of the distance between two shapes have been proposed in the computer vision literature but there is no agreement on the correct way to measure shape similarity. In this work, we represent a surface s_i in the Euclidean embedding space $\mathbb{R}^3$ by its signed distance function $\mathbb{D}_{s_i}$. In this context, we define the distance between two shapes to be the L_2-norm of the difference between their signed distance functions [Leventon 2000]:

$$d^2_{\mathcal{S}}(s_1, s_2) \quad = \quad ||\mathbb{D}_{s_1} - \mathbb{D}_{s_2}||^2_{L^2}$$

Note that, in order to define a distance between shapes, which is invariant to rigid displacements (*e.g.*rotations and translations), we first align the shapes using their principal moments before computing distances. Note also that the proposed method is obviously not limited to a specific choice of distance [Charpiat 2005, Etyngier 2007b]. We use a dataset of 39 ventricles nuclei extracted from Magnetic Resonance Image (MRI). We learn a random

subset of 30 shapes and corrupt the nine remaining shapes by an occlusion (Fig. 4.10a,b). In order to recover the original shapes we project, with our method, the shapes onto the shape manifold. We then compare the reconstruction results with the nearest neighbor, the mean of the $d+1$ nearest neighbors and the method of Dambreville [Dambreville 2006]. The parameters of this experiment are $d = 2$. In Figure 4.10-d one example of a reconstructed shape (red) is obtained from the $d+1$ nearest neighbors of $s_\bullet$ (Fig. 4.10c). In order to quantitatively evaluate the projection, we define the reconstruction error as $e(s) = d_{\mathcal{S}}(s_\circ, s)/\sigma$, where $s_\circ$ is the original shape and s is the reconstructed shape. The occluded shape has an error of $e(s_\bullet) = 4.35$, while the nearest-neighbor has an error of 1.81. In Table 4.2 we see that our method is superior to the one proposed by Dambreville [Dambreville 2006].

Avg. err occluded shapes	Nearest neighbors(NN)	Mean of NN
4.67	1.81	1.96
[Dambreville 2006]	Our method	
1.1	0.58	

Table 4.2: Average reconstruction error for a set of 9 noisy shapes

Application: denoising of Digits

To test the performance of our approach on the task of image denoising, we apply the algorithm on the USPS dataset of handwritten digits[3]. In a first experiment, we compare our method to five state-of-the-art algorithms [Dambreville 2006], [Dambreville 2006]+[Arias 2007], [Kwok 2004], [Kwok 2004]+[Arias 2007] and [Carreira-Perpiñan 2007]. For each of the ten digits, we form two training sets composed of randomly selected samples (60 and 200 respectively). The test set is composed of 40 images randomly selected and corrupted by some additive Gaussian noise at different noise levels. The process of denoising simply amounts to estimating the pre-images of the feature vectors given by the Nyström extension of the noisy samples. For all the methods, we take $d = 8$ for the reduced dimension (number of eigenvectors for the kernel-PCA based methods). Table 4.3 shows a quantitative comparison based on the pixel-signal-to-noise ratio (PSNR). Our method visually (Fig. 4.8) and quantitatively outperforms other approaches. Interestingly, it

[3]The USPS dataset is available from http://www.kernel-machines.org.

is less sensitive to noise than other ones and yields good results even under heavy noise.

σ^2	[Dambreville 2006]	[Kwok 2004]	[Arias 2007] +[Dambreville 2006]	[Arias 2007] +[Kwok 2004]	[Carreira-Perpiñan 2007]	Our method
σ^2						
0.25	8.50	15.71	10.17	16.18	14,01	17.71
0.45	9.05	13.87	9.98	15.42	13,91	17.52
0.65	9,78	13.10	9.58	13.60	13.89	17.38
0.85	9.06	12.58	8.61	13.91	13.87	17.32
0.25	9.35	16.08	11.97	16.21	15,27	17.95
0.45	9.64	15.70	10.18	15.98	14.85	17.85
0.65	9.41	13.97	10.26	15.85	14.13	17.79
0.85	9.24	13.06	10.25	15.07	14.07	17.75

Table 4.3: Average PSNR (in dB) of the denoised images corrupted by different noise levels σ. Training sets consist in 60 samples (first 4 rows) and 200 samples (last 4 rows).

4.5 Discussion

Kernal PCA

First, we focused on the pre-image problem in kernel methods such as Kernel PCA thus elucidating the issue of correctly normalizing in centered feature space. A geometric interpretation eased the understanding of operations involved when working with centered data in feature space. As a consequence, we deduced a new normalization criterion for previously proposed pre-image methods. The theoretical results could be nicely verified at hand of computed examples. Despite the strong results, we have to point out a weak point that could be improved. In fact, we don't compute the true mean of the $N-1$-dimensional hypersphere but rather the centroid of the points in feature space. Therefore the computed centroid strongly depends on the distribution of the points on the $N-1$-dimensional hypersphere. For instance, one might consider the case when all the points are mapped onto the upper part of the sphere. In this case the empirical mean is a very bad approximation of the true mean. Some exciting developments in the area of high dimensional geometry were the introduction of the idea of core sets [Agarwal 2005]. In this context, Bâdoiu [Bâdoiu 2003] proposed an algorithm for the computation of balls for core-sets. The idea is to approximate a set P of N points by selecting a set of points S (not necessarily contained in P) of size $O(\frac{1}{\varepsilon})$. S is called the *core-set*. A property of the core-set is that the value of the optimal solution on S is close (i.e within a $1+\varepsilon$ factor) to the set P (in our case the true hypersphere). Applying the idea of core-sets to the estimation or approximation of the true hypersphere is excpected to yield a better approximation to the mean in feature space.

Diffusion maps

Second, we focused on the pre-image problem. We provided a solution to the pre-image problem using diffusion maps. Following a manifold interpretation of the training set, we define the pre-image as a Karcher mean interpolation between neighboring samples with respect to the diffusion distance. Results on real world data, such as 3D shapes and noisy 2D images, demonstrate the superiority of our approach. In the continuation of this work several ideas may be exploited. In the perspective of working on complex shape spaces, our projection operator, defined from a manifold point-of-view, could be used in different tasks, such as segmentation with shape priors [C. Florin 2007, Rousson 2008, A. Besbes 2009, Essafi 2009], interpolation and reconstruction

of shapes, and manifold denoising. Interestingly, our approach is able to deal with manifolds of complex topology. In the context of manifold denoising this property can be useful. So far, none of the pre-image problems were tested when the training data itself contains heavy noise. We are currently investigating these directions.

Chapter 5

Non-rigid Shape matching using Geometry and Photometry

Contents

5.1 Introduction 107

5.1.1 Our contributions 111

5.2 Background 112

5.2.1 Geometric methods 112

5.2.2 Photometric methods 115

5.2.3 Where geometry meets photometry 117

5.3 Variational correspondence optimization using geometry and photometry 118

5.3.1 Problem formulation 118

5.3.2 Geometry 119

5.3.3 Photometry 119

5.3.4 Optimization 123

5.3.5 Discretization 123

5.3.6 Initialization 124

5.3.7 Gradient descent 124

5.4 Results 127

5.4.1 Validation 127

5.4.2 Real data 127

5.5 Discussion 128

5.1 Introduction

Matching of objects is an essential task in Computer Vision. Matching occurs at several stages starting with matching of features or regions in images for the purpose of camera calibration, panoramic image stitching or image warping to name just a few. While this kind of matching is considered as low-level image processing, the matching of shapes (for instance curves and surfaces) is important for high-level Computer Vision such as recognition and semantization of scenes. The term matching or also commonly known as the **correspondence problem** often appears in connection with the term registration. And in fact they are closely related as matching, e.g. finding a set of correspondences on different objects, is the preliminary step for the registration, e.g. aligning two shapes into a common reference frame. As a very important application of matching between shapes we want to mention shape morphing, e.g. the interpolation between two shapes [Kilian 2007, Bronstein 2007]. These three tasks allow further processing such as recognition and classification of shapes [Shilane 2004]. One possibility to measure similarity between shapes is to measure the energy (or distortion) introduced by morphing one shape onto another [Younes 2000, Younes 1998, Trouvé 2005, Bronstein 2006b]. Therefore, if the interpolation between two shapes introduces a strong distortion the shapes are likely to be dissimilar and vice versa. This measure is then often used as distance which allows to use a distance based clustering algorithm such as MDS (Section 3.1.2) and perform recognition of shapes [Elad 2001]. Another application of registration is to solve the average template matching [Hirani 2001] which is used to compute the mean shape of a dataset of input shapes. Once the template or mean shape is computed shapes from the original data set are morphed onto the mean shape. In this way shapes can be characterized by the deformations around the mean [Charpiat 2006a, Pennec 2006]. It is obvious that correspondence matching and registration is an essential task for many Computer Vision applications.
In this work, we focus on the problem of **correspondence matching** between two surfaces which is a subproblem of the more general problem : the **registration problem**.

Registration problem

The registration problem takes two or more surfaces as input. Each surface provides a set of m primitives $\{P_1^1, \ldots, P_m^1\} \in S^1$ and a set of n primitives $\{P_1^2, \ldots, P_n^2\} \in S^2$ corresponding to the second surface. We now seek a transformation (aligning) of the two objects.
In pattern recognition an object is classified if there exists a transformation

$\boldsymbol{f}$ aligning a sufficient large number of primitives between the two objects. Therefore one seeks to maximize the number of correspondences between the objects and the quality of these matches at the same time. In other words, the computed transformation maps the corresponding primitives onto the same coordinates in a common metric space.
We call correspondence function the map π associating to every index i of a primitive P_i^1 of the model S^1 the index $j = \pi(i)$ of a primitive P_j^2 of the model S^2. We define the similarity measure $\alpha(x, x') = 1$ if $d_{\mathbb{X}}(x, x') \leq \varepsilon$. The optimal correspondence map $\hat{\pi}$ maximizes the following matching score:

$$\hat{\pi} = \arg\max_{\pi} \left(\Sigma_i \alpha(f(P_i^1), P_{\pi(i)}^2) \right).$$

Simultaneously, we want to optimize the quality of the correspondences and find the transformation f aligning the correspondences in a common metric space. The map $\hat{f}$ is usually estimated in the least square sense with the convention that $d_{\mathbb{X}}(P_i^1, .) = 0$ if P_i^1 has no correspondence. The map $\hat{f}$ is the solution to the following optimization problem

$$\hat{f} = \arg\min_{\boldsymbol{f}} \left(\Sigma_i d_{\mathbb{X}}(f \circ x_i, y_{\pi(i)}) \right),$$

which minimizes the mean error between the transformation of a primitive and it's correspondence. This is the most general setting for the registration problem. And one can easily see that several issues have to be addressed seperatly.
As one can see that whole problem of shape matching consists of simultaneously seeking in the space of correspondences and in the space of transformations. In this work, we focus on the problem of correspondence matching, fix some of the variables (c.f. section 5.3.1) and leave out the estimation of the transformation.

In many cases, the task of correspondence matching and alignment (or morphing, interpolation) are solved simultaneously using one algorithm. Such algorithms are modeled most of the time to solve a specific problem and therefore are not easily adaptable to different problems. Most methods can devided into classes based on four criteria:

- shape representation
- type of transformation
- similarity measure
- algorithmic approach

These are the most important characteristics of shape matching methods but there are several other aspects such as partial [Gal 2006] and multiple shape matching [Mitra 2004] which will not be considered here.
Due the vast literature produced over the last two decades, we will limit the literature review to directly related work. Nevertheless, we want to mention some interesting surveys on shape matching among which Lazarus [Lazarus 1998] extensively reviews 3D matching with an emphasis on the transformation between shapes. On the other hand, Alt [Alt 1996] reviews geometric techniques based on the Hausdorf distance whereas Veltkamp [Veltkamp] gives an overview of several similarity measures for shape matching. In the context of medical imaging, Audette et.al. [Audette 1999] provides a quite general overview of matching methods. Finally, we want to mention the paper of Planitz [Planitz 2005] providing a classification of popular methods in shape matching.

Shape Representations:

In general a surface S can be represented parametrically as the image of a certain parameter domain or implicitly as the 0 level-set of a smooth function(c.f. section 4.2.2). The advantages and disadvantages were discussed previously in section 4.2.2. We focus here on two-dimensional Riemannian manifolds. Together with the geodesic distance function $d_S : S \times S \mapsto \mathbb{R}^+$ induced by the Riemannian metric, we define the metric space (d_S, S). Here S is embedded in $\mathbb{R}^3$. In extrinsic coordinates, a point on the surface $P \in S$ is written as $P = [x, y, z]^T$. In practice, the surface represented by a finite dimensional set of sample consisting of N points. In addition, we assume that the point set is triangulated so that it is manifold. This kind of representation is called a triangle mesh. The triangle mesh is defined by a set of N points $\{P_1, \ldots, P_N\}$ and a list of K triangles $\{(T_{11}, T_{12}, T_{13}), \ldots, (T_{K1}, T_{K2}, T_{K3})\}$, where each triangle is defined by a triplet of point indices $T_{ik} \in \{1, \ldots, N\}$. At the end, we have a piecewise linear approximation of the original surface. For the sake of completeness, we mention some further representations. For instance, there are several smooth representations based on splines and nurbs. These representations are very common in computer aided engineering. Other parametric representations include : spherical harmonics, medial axis and point based representation. Several more representation exist but are beyond the scope of this work.

Transformations:

The mapping between two surfaces $f : S^1 \mapsto S^2$ cannot always be com-

puted directly. Therefore it is common to compute a map $f : \mathbb{X} \mapsto \mathbb{Y}$ between the embedding spaces of the shapes. This amounts to deform the embedding space and therefore deform the shape so that the two shapes align. In this context, the map is usually non-linear and solves the registration problem directly. The non-linear map is modeled either parametric or non-parametric. Non parametric methods were sucessfuly applied in [Modersitzki 2004]. Parametric methods, for instance thin-plate splines [Bookstein 1989] and Bézier splines [Khamene 2007, Rueckert 1999] have been applied in several registrations tasks.
On the other hand, methods operating directly on the surfaces and not the ambient space can be seperated into two classes: i) Inspired by differential geometry (c.f section 2.3.1), a common intermediate parametrization of S^1 and S^2 is computed and the problem is formulated in the parameter domain. For more details the reader is referred to [Zöckler 2000, Gu 2004, Litke 2005]. ii) The second type of approach computes a map directly between S^1 and S^2. This methodology was applied in [Schreiner 2004, Bronstein 2006b].

Similarity Measures:

In order to optimize a function, we need a similarity measure. Several geometric similarity measures were proposed for the task of registration and correspondence matching. Mostly, such measures evaluate the surfaces S^1, S^2 and their derivatives. Among these methods the most prominent is the ICP algorithm [Zhang 1994, Besl 1992] evaluating only the measure between the surfaces. Unfortunately, this measure is not invariant under transformation. However, measures based on first and higher order geometric features are invariant. Therefore commonly used measures quantify metric distortions or stretching. In [Litke 2005, Bronstein 2006b] the author computes the distortion using geodesics on the surface. Higher order derivatives such as curvature measures bending of a surface. Terzopoulus [Terzopoulos 1987] introduces an energy measuring curvature deviation through the thin-shell energy. Basically the idea is to look at the difference between first and second fundamental forms at corresponding points. Closely related energies are the ones proposed by Gu [Gu 2004] and Litke [Litke 2005].

Those basic properties are the ingredients for a whole family of geometric registration algorithms. The common approach then, is to optimize this similarity measure composed of a geometric measure and a regularization term. The geometric term is supposed to align the two surfaces consistently whereas the regularization term adds a certain regularity to the problem in order to achieve proper convergence of the method. Beyond the purely geometric approach, let us mention a few that combine geometry and

statistical information.

Pennec et.al. [Pennec 2006] use statistics on the manifold of diffeomorphisms. The deformation between a set of shapes is driven by an elastic energy with the statistical information on the diffeomorphisms. The major drawback of this method is the requirement of a carefully selected training set which reflects the geometric properties of the shape to match.

Lipman [Lipman 2009] proposes a Tensor voting scheme in the complex parametrization plane of the surfaces. The votes output a fuzzy correspondence matrix which is used to compute a permutation matrix which outputs a set of correspondences with confidence values. The main advantage of this algorithm is that it can find intrinsic point correspondences in the spirit of [Bronstein 2006b] even under extreme deformation.

5.1.1 Our contributions

The goal of this chapter is to contribute a new model for finding a set of correspondences between two non-rigid surfaces. The strategy proposed is to combine two methods to find the best set of correspondence optimizing, on one side a geometric energy and, on the other side forcing photometric consistency between correspondences.
We propose to take the best of both worlds and design an extension of the work of [Bronstein 2006b] by adding a photometric term to their energy. We obtain a robust multi-resolution 3D surface mapping procedure that combines photometric and geometric information(c.f. Figure 5.3). We experiment with it for non-rigid surface correspondence between two surfaces and observed a different time steps in a multiple cameras environment thus demonstrating its superiority.
The main contributions are:

- registration non-rigid shapes
- handles large displacements
- handles locally elastic deformation
- handles symmetry ambiguities arising when working with distance function

The result of this work was published at the ACCV conference in Xi'AN, China [Thorstensen 2009b].

5.2 Background

The correspondence problem is one of the fundamental challenges in computer vision. It might be in the context of optical flow, calibration or surface registration. For rigid surface and point cloud registration, Iterative Closest Point (ICP) and its variants [Gelfand 2005] are the standard algorithms. Operating on a purely geometric level, they rely on approximated differential quantities, e.g. curvature, or more robust surface descriptors [Belongie 2002]. Recently, several algorithms also address the problem of non rigid surface registration. They can be mainly divided into two categories: geometric and photometric. Whereas geometric methods assume the known geometry, photometric methods estimate structure and motion.

5.2.1 Geometric methods

We give a short description of the most important and directly related methods.

Rigid Iterated Closest Point (ICP)

The algorithm "Iterated Closest Point" was first introduced by Besl and McKay [Besl 1992] and Zhang [Zhang 1994]. The idea is to alternate the optimization of the correspondences and the transformation. Starting with an initial transformation f_0 the following two steps are alternatively executed:

Correspondence matching:

Each primitive x_i of the transformed model is matched against the closest primitive y_j such that:

$$\pi_t(j) = \arg\min_j \mathrm{d}(f_t \circ x_i, y_i) \text{ or } y_{\pi_t(j)} = \mathrm{NN}(f_t \circ x_i)$$

where NN stands for the nearest neighbor.

Registration:

The transformation is usually computed using a least square estimation, especially when working with point clouds. The explicit solution has the advantage of being fast which is a desirable property when working with iterative minimization techniques.

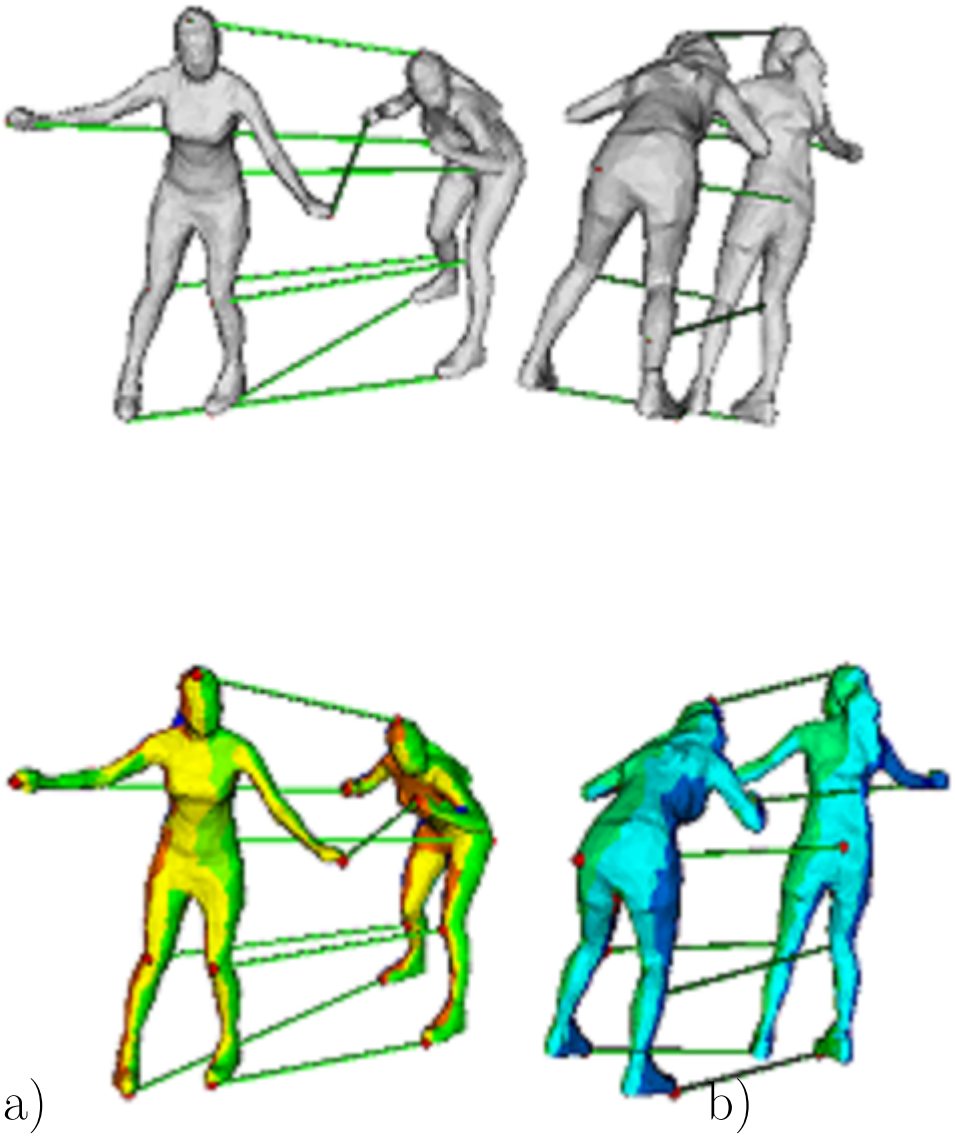


Figure 5.1: a) Initialization using solely geometric information is sensible to symmetry in shapes. b) Color coded labeling of the meshes (see text) and our photometric plus geometric initialization, in which ambiguities have been solved.

A major problem encountered when using a distance based matching criteria is the lack of symmetry of the solution. Generally, one can encounter the situation where $y_i = \mathrm{NN}(x_i)$ and $x_k = \mathrm{NN}(y_i)$ with $i \neq k$. To circumvent this mathematical issue, Pennec and Ayache[Pennec 1998] proposed a symmetrized version where the points x_i and $y_i = \mathrm{NN}(x_i)$ are matched against, if and only if, $\mathrm{NN}(\mathrm{NN}(x_i)) = x_i$.
Under certain conditions, convergence of the algorithm to a local minimum can be proven. The idea is to consider the two steps presented before as two distinct minimizations of the same criteria with respect to two different parameter sets: the matching function (NN) and the transformation (f_t). In this setting Cohen [Cohen 1996] shows the convergence of the procedure. An analysis of convergence under several minimization algorithms is provided in [Pottmann 2006]. Further, he shows how to extend the concepts to the simultaneous registration of an arbitrary number of views.
This type of iterative minimization algorithm requires a termination criteria. Contrary to the continuous case, we usually have properly separated primitives (*e.g.*points) in computer vision and graphics. Therefore the set of transformations is discrete and each transformation is unique. Once the algorithm has converged the correspondences don't change from one iteration to the next and the computed transformation remains the same. This behavior allows to define a stopping criteria terminating the algorithm once the transformations remain unchanged from one iteration to the next. Otherwise, if the algorithm oscillates around a local minimum we can force the algorithm to step by limiting the maximum number of iterations. Lastly we notice the sensitivity of ICP with respect to the initial conditions. This behavior especially applies when the two models only partially overlap. In this case, it is necessary to estimate either a transformation or a set of correspondences very close to the expected solution. Although ICP is widely used for rigid registration it applies often as a verification step once the scenes are already registered and refine an earlier registration method such as feature based registration or geometric hashing.

Non-rigid ICP

A non-rigid counterpart to ICP was introduced in [Amberg 2007]. Further, in [Brown 2007], the authors propose a non rigid registration method by piecewise rigid registration and local thin plate spline alignment of range scan images.

Generalized Multidimensional Scaling

Elad *et al.* [Elad 2003] take an intrinsic point of view of the matching problem. By isometrically embedding two surfaces into a common Euclidean space the problem is reduced to the simpler of rigid matching between surfaces. In this common Euclidean space, moment descriptors of the two surfaces are computed. Based on the descriptors, a simple descriptor matching is applied to find correspondences. Following this idea, Bronstein *et al.* [Bronstein 2006a] take it one step further and solve the correspondence problem by embedding the surface directly into another. The major drawback of this approach is the difficulty to handle ambiguities like shape symmetries. As the method solely relies on the distance function of the surface, the algorithm tends to wrongly match pairs of symmetric points (see Figure 5.1). Also their approach is more targeted towards classification problems.

5.2.2 Photometric methods

On the photometric side, non rigid registration is commonly known under the name of scene flow. Vedula *et al.* [Vedula 2005] were the first to introduce the concept of scene flow which is the 3D extension of 2D optical flow. It is the simultaneous reconstruction of structure and motion between time frames. Further work on scene flow was undertaken by other teams [Furukawa 2008, Huguet 2007, Pons 2007a]. The first complete static dense stereo was proposed by Keriven based on the local minimization of a functional in the third dimension which is independent of the surface normals. Pons et.al. [Pons 2007a] extend the work in [Keriven 1998] to solve the scene flow problem which implicitly solves the correspondence problem between non rigid surfaces. We therefore present the work of Keriven and Pons.
The model of Keriven considers a set of n cameras which project the observed scene onto images I_i through the projections $\Pi_i : \mathbb{R}^3 \mapsto \mathbb{R}^2$. Furthermore, each camera has an associated indicator function χ_k computing the visible points of S in the corresponding camera. Their model assumes that the visibility does not change with respect to small perturbations which is of course not valid but works well in practice. Then the objective function proposed in [Keriven 1998] measures the photometric disparity of each projected point P of the surface among all cameras it is visible from. This error is integrated over the whole surface which yields a global scoring functional of the following form:

$$E(S) = \int_S \rho^S ds \text{ with } \rho(P)^S = \|I_i \circ \Pi_i(P) - I_i \circ \Pi_i(P)\|_1 .$$

It is obvious that this model is too simplistic to be of use for real world 3d multi-view reconstruction. Pixel by pixel comparison of intensity values bears too little information and the risk of local minima is unavoidable. Therefore Keriven suggests a better functional ρ based on normalized cross correlation. The normalized cross correlation is differentiable and robust to illumination changes and measures photometric similarity in a neighborhood around the points p_1, p_2 in the images I_1, I_2:

$$E(S) = \int_S \rho^S ds \text{ with } \rho(P)^S = (1 - NCC^\sigma(I_i \circ \Pi_i(P), I_i \circ \Pi_i(P)),$$

with:

$$\begin{aligned} NCC^\sigma(p_1, P_2 &= \nu^\sigma(p_1, p_2)/\sqrt{\nu^\sigma(p_1)\nu^\sigma(p_2)}, \\ \nu^\sigma(p_1, p_2) &= G^\sigma * (p_1 - \mu^\sigma(p_1))(p_2 - \mu^\sigma(p_2))/\omega^\sigma, \\ \nu^\sigma(p_1) &= G^\sigma * (p_1 - \mu^\sigma(p_1))^2/\omega^\sigma + \tau^\sigma, \\ \mu^\sigma(p_1) &= G^\sigma * p_1/\omega^\sigma, \end{aligned}$$

where NCC^σ is the normalized cross correlation over a Gaussian window of scale σ. $\omega_\sigma(x) \int_\Omega G(x - y)dy$ normalizes the quantities with respect to the shape of the correlation window and τ is a small constant to avoid numerical issues. Minimizing this energy not only modifies the position but also the surface normal such that the projection of the surface best matches the images. The extension of the previous work to 3D scene flow was proposed by Pons [Pons 2007a]. Pons proposes a variational model to estimate shape and motion of a moving object by minimizing a global similarity measure. In order to recover motion between time he not solely considers the reprojection errors between images but also between images at different times. Therefore, contrary to the work of Keriven, this methods minimizes the photoconsistency measure with respect to the back-projected images (shape) and the predicted images (motion) induced by a time dependent motion field. In order to make the computation tractable, the energy is minimized in the images. Images are back-projected on the surface and reprojected on another camera. Using projective texture maps from OpenGL, projective distortion is handled automatically and allows a parallel implementation of the image based similarity measure. Courchay [Courchay 2009] propose a novel method to simultaneously and accurately estimate the 3D shape and 3D motion of a dynamic scene from multiple-viewpoint calibrated videos. The motivation is to formulate an energy exploiting spatio-temporal redundancy. The shape is represented as an animated mesh with fixed connectivity capturing the motion of the object in scene. Like in [Pons 2007a], the energy is minimized in the

image which can be efficiently computed and fully handles projective distortion and partial occlusions. In order to properly initialize variational methods, authors often use a conservative first estimate [Vu 2009, Goldlücke 2004, Esteban 2008, Vlasic 2009] such as the visual hull. Computing the visual hull of an object based on the images needs a foreground-background segmentation of the images. I refer the interested reader to the rich literature on this topic [Laurentini 1994, Matusik 2000, Lazebnik 2001, Franco 2003].

5.2.3 Where geometry meets photometry

Lastly, both approaches have their advantages. Several recent papers address this problem. In [Starck 2005], Starck *et al.* provide a method to register shapes from silhouettes. The method embeds the two surfaces into a common spherical domain. Using the analytic expression of geodesic distances on the sphere allows one to minimize the geometric distance between correspondences and the distances between the associated color feature. This work is the most related to ours in the sense that they perform an isometric mapping into a common metric space and use geodesic distances as regularization for the appearance minimization. In order to be robust with respect to topological changes, they then developed a matching algorithm based on a Markov random field optimization [Starck 2007a]. A much more heuristic approach is used by Ahmed *et al.* [Ahmed 2008]. A set of SIFT features is computed which are then used for initialization of a refinement model. Lastly, the work of Kiran *et al.* [Varanasi 2008] also makes use of SIFT. The 3D position of the interest points are used to compute a sparse estimate of the motion field. By applying a diffusion operator on the sparse motion field, they recover a dense motion field.

Most of the algorithms presented so far use heuristics to recover correspondences between two meshes. In this work we want to consider a more rigorous model and propose a variational refinement method using ideas from variational stereo [Keriven 1998, Courchay 2009, Pons 2007a] and geometric embedding methods [Elad 2003, Bronstein 2006a]. In this way, we recover temporal correspondences between meshes at different resolutions and independently of the combinatorial configuration. Further, our method can handle local elastic deformations so that the correspondence is consistent with the observed images. Although the geometric embedding methods work well on very smooth meshes, we cannot expect to work on meshes issued from photometric stereo. For instance, we want to find correspondences between meshes issued from an approximate surface reconstruction methods such as in [Aganj 2009]. This is challenging since the reconstructed surfaces do not

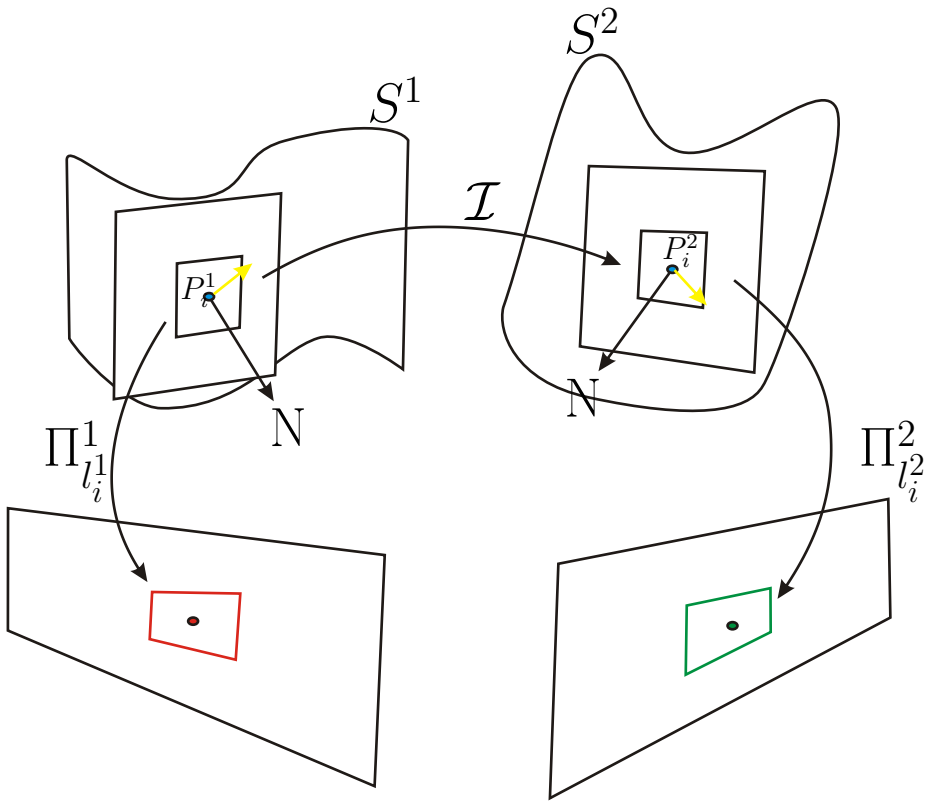


Figure 5.2: A visualization describing the problem configuration as in section 5.3.1.

have the properties required to work on a purely geometric level. To alleviate this limitation, we exploit photometric information and improve the method in [Bronstein 2006a]. The recovered set of correspondences are then used to improve variational methods as in [Courchay 2009].

5.3 Variational correspondence optimization using geometry and photometry

5.3.1 Problem formulation

Let S^1 and S^2 be the two surfaces to match. Each of them is observed by a certain number of cameras. Although not required, we simplify notations and

suppose that the number and positions of the cameras are constant, so that both surfaces are observed by n cameras defined by their respective projections $\Pi_1, \dots \Pi_n$. We denote by I_i^k the i^{th} image of surface k.

Following [Bronstein 2006a], we use a Lagrangian point of view where a set of m correspondences are constraint to move on the surface such that they minimize a given energy. We denote (P_i^1, P_i^2) such a correspondence where $P_i^k \in S^k$ and $\Theta = \{(P_i^1, P_i^2), 1 \leq i \leq m\}$ the set of all correspondences. m is a fixed integer that can be estimated during initialization (see Section 5.3.6). Our energy is a function of parameter Θ that writes:

$$E_{tot}(\Theta) = \alpha E_{geom}(\Theta) + \beta E_{photo}(\Theta) \tag{5.1}$$

The first term E_{geom} is the geometric part, taken from the work of Bronstein *et al.* [Bronstein 2006b] while E_{photo} is our photometric attachment. As usual, α and β are positive constants that control the relative weights among these terms. Minimizing energy (5.1) with respect to Θ will position the correspondences on the mesh such that their projections in the images minimize a photometric dissimilarity measure while geodesic distances on the surfaces are respected.

5.3.2 Geometry

Bronstein *et al.* [Bronstein 2006a] propose to embed near-isometric surfaces one into another by minimizing the following energy:

$$E_{geom}(\Theta) = \sum_{i>j} (d_{S^1}(P_i^1, P_j^1) - d_{S^2}(P_i^2, P_j^2))^2. \tag{5.2}$$

where d_{S^k} is the geodesic distance on surface S^k. Again, this energy suffers mainly from two weaknesses: (i) symmetries yield ambiguities and (ii) if the object undergoes locally elastic deformations between shape S^1 and shape S^2, geodesic distances are not preserved exactly. Note that this is also the case when the surface are 3D reconstructions, since they are unavoidably noisy.

5.3.3 Photometry

For our image matching term E_{photo}, we chose the normalized cross-correlation to measure similarity between corresponding points. Its simplicity, robustness in the presence of varying lighting conditions and differentiability make it a common choice in variational methods. Each surface point is generally seen from several cameras and one might be tempted to correlate multiple pairs

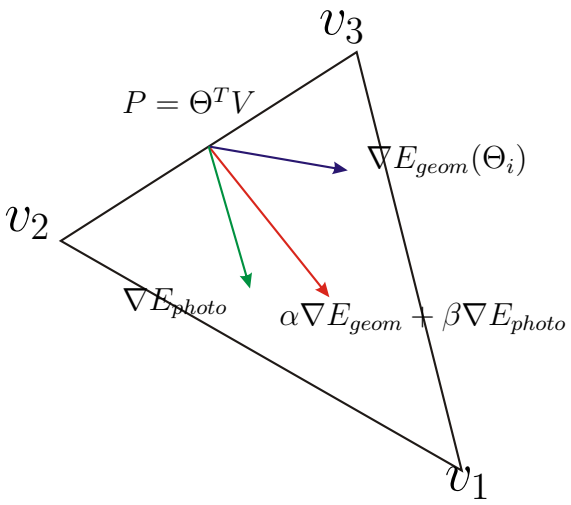


Figure 5.3: This visualization shows the interaction between the geometric and the photometric gradient in a triangle composed by the vertices (v_1, v_2, v_2). The point $P = \Theta^T V$ is given as a weighted combination of the vertices. P travels along the red arrow which is calculated as a linear combination of the geometric term ∇E_{geom} and the photometric term ∇E_{photo}.

of images. However, in our experiments, the number of cameras is relatively small. Thus, using information from only one pair of cameras for each surface point reveals to be enough.

As a first step, for each surface S^k, we associate to each point $M \in S^k$ an optimal image $I^k_{l^k(M)}$. Choice of labels l^k might be guided by different criteria. Here, we compute partitions of the surfaces following [Allène 2008]. This method assigns each point smoothly to a label corresponding to the camera from which it is best viewed. Using graph-cut optimization, the labeling is obtained by minimizing a weighted sum of two terms which represents a good trade off between resolution and color continuity, while respecting occlusions (c.f. Figure 5.1b).

Let $l^k(i)$ be a short notation for $l^k(P^k_i)$, our photometric energy then writes:

$$E_{photo}(\Theta) = \sum_{i=1}^{m} g[NCC(I^1_{l^1(i)} \circ \Pi_{l^1(i)}, I^2_{l^2(i)} \circ \Pi_{l^2(i)})(P^1_i, P^2_i)] \tag{5.3}$$

where g is a positive decreasing function and $NCC(f^1, f^2)(M^1, M^2)$ denotes the normalized cross-correlation of functions f^1 and f^2 between two related neighborhoods of points M^1 and M^2.

Following the stereovision work of Keriven *et al.* [Keriven 1998], we approximate locally the surfaces by their tangent planes at points M^k. In their case, only one surface is considered and M_1 and M_2 are the same point, with the same tangent plane, thus the same neighborhood. Their correlation boils down to correlating image regions related by a homography. In our case, we suppose that the tangent plane to S^1 at point M^1 and the tangent plane to S^2 at M^2 are related by a given two-dimensional isometry $\mathcal{I}_{M^1,M^2}$ sending M^1 to M^2. Under this assumption, neighborhoods on the respective tangent planes are related and the correlation $NCC(f^1, f^2)(M^1, M^2)$ is correctly defined. Moreover, it (and its derivatives) remains easy to compute since corresponding image regions are still related by a homography. Introducing the isometry $\mathcal{I}_{M^1,M^2}$ at each point pair (M^1, M^2) might be thought as problematic since one would be required to match the surfaces to know it, thus yielding a chicken and egg problem. Practically, this is not the case. As usual, we will minimize the energy by means of a gradient descent starting from a coarse initialization (see Section 5.3.4). This approximate solution reveals to be sufficient to obtain a robust $\mathcal{I}_{M^1,M^2}$. We proceed in the following manner: (i) each correspondence point P^k_i define a geodesic distance map $d_{S^k}(P^k_i, .)$ on S^k; (ii) the gradients of these distance maps at a given point M^k define local directions that should correspond from one surface to the other if M^1 corresponds to M^2; (iii) as a consequence, the best[1] isometry from the tangent

[1] in the least squares sense

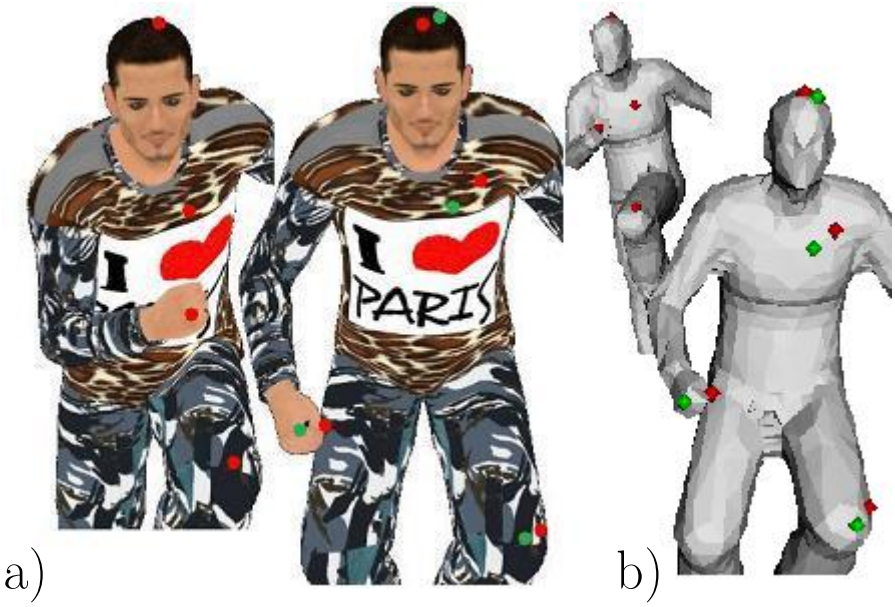


Figure 5.4: a) On the left, the front image of the first mesh, red dots being the projections of some key points. On the right, the front image of the second mesh, showing the reprojections of the corresponding points, in red obtained with the method in [Bronstein 2006a] (initialized with our method), in green obtained with our method. b) Shows the same correspondences directly on the meshes with the same color code. The inaccuracies of [Bronstein 2006a], due to local elastic deformations, are corrected by our photometric + geometric criterion.

plane at M^1 to the one at M^2 which sends both M^1 to M^2, and the distance gradients directions at M^1 to the ones at M^2, is a good estimate of $\mathcal{I}_{M^1,M^2}$. Please note that computing distance maps represents no extra cost since it is part of E_{geom}. Note also that the isometries $\mathcal{I}$ are actually needed only for the pairs (P_i^1, P_i^2) and that they will be refined during the gradient descent iterations as the pairs move.

5.3.4 Optimization

5.3.5 Discretization

We suppose that the surfaces are both discretized as collections of triangles. Following [Bronstein 2006a], points P_i^k are taken as barycenters of triangle vertices. Θ consists in a choice of triangles and corresponding barycentric coordinates. From now on, we consider the notations $\Theta_{S^k}^i$ and $P_{S^k}^i$ synonymous. The geodesic distances between all vertices of the mesh are computed using the Fast Marching algorithm for triangular surfaces [Sethian 1999a, Kimmel 1998]. Geodesic distance is then interpolated like in [Bronstein 2006a] (note that the problem is not only to interpolate the distance at a certain barycenter, but also the distance **to** some barycenter).

For the photometric part of the energy, discretization is not a particular issue: the labeling method [Allène 2008] is designed for triangle meshes, and we use a standard normal interpolation method to estimate the tangent planes.

Minimizing the energy with respect to both the P_i^1's and the P_i^2's is obviously not well posed. Although different cases might apply, in our experiments we have no further constraint on the choice of the points to be matched. Thus, we fix points P_i^1 to their initial position (see Section 5.3.6) and minimize the energy with respect to the positions of points P_i^2.

As written above, we use a classical gradient descent. Properly minimizing it is not trivial because the problem is non-convex. In order to cope with local minima, we apply a multi resolution strategy, considering the problem at several scales. Once a solution is found at a coarse scale, it is used to initialize the problem at a finer scale. Our problem has two scalable dimensions. The first one is the number of correspondences and the second is the scale of the images. This leads to a two step multi resolution scheme. Starting with a small number of correspondences, we iteratively increase the number of points by interpolating the solution from the coarser level to the next finer level. This scheme is adapted from [Bronstein 2006c]. Then, at each level, we perform a gradient descent in a multi scale manner using a Gaussian pyramid of the images.

5.3.6 Initialization

We first have to initialize the correspondences. Copying [Bronstein 2006c], we take advantage of the geodesic distance maps and use the farthest point sampling (FPS) strategy [Moenning 2003] to get geometry-based feature points on the surfaces. For near isometric surfaces we can expect the sampling to be almost identically distributed on both surfaces S^1 and S^2 [Bronstein 2006a]. Taking photometry into account to avoid geometric ambiguities, we then reject points that have an autocorrelation score below a given threshold, thus corresponding to non textured regions. As in [Bronstein 2006c], points are then associated using branch and bound optimization [Dasgupta 2007], yielding m initial pairs. Here, to the initial geodesic distance based criterion, we add a photometric one in order to cancel out geometric ambiguities. Because no correlation is possible (at this stage, tangent planes cannot yet be related by isometries), we use SIFT descriptor based similarity, being thus invariant to scale and orientation. The results of the initialization of the correspondences can be viewed in Figure 5.1. Note how geometric ambiguities are solved. The parameteres (α, β) are set manually.

5.3.7 Gradient descent

Optimization is performed at all scales until convergence is reached, i.e. the norm of the gradient goes below a given threshold. The expressions of the gradients of the geometric and the photometric parts of our energy can be found in [Bronstein 2006c] and [Keriven 1998] respectively. The interaction between the photometric part and the geometric term is illustrated in Figure 5.3.

Remember that Θ consists of these coordinate but also of the choice of the triangles to which the barycenters are related. As in [Bronstein 2006a], the gradients are computed for a fixed choice. However, when a point P_i^k gets out of its related triangle, we force it to stop at the reached edge and assign it to the triangle at the "other side" of this edge. Doing it this way, points travel gently from one triangle to another if needed.

Gradient of geometric term

The geometric energy is taken to be the same as in [Bronstein 2006c] and writes as $E_{geom}(\Theta_1, \ldots, \Theta_m) = \sum_{i>j}(d_{S^1}(P_i^1, P_j^1) - d_{S^2}(P_i^2, P_j^2))^2$. For one fix Θ_i using the three point geodesic distance interpolation scheme [Bronstein 2006c], the energy writes

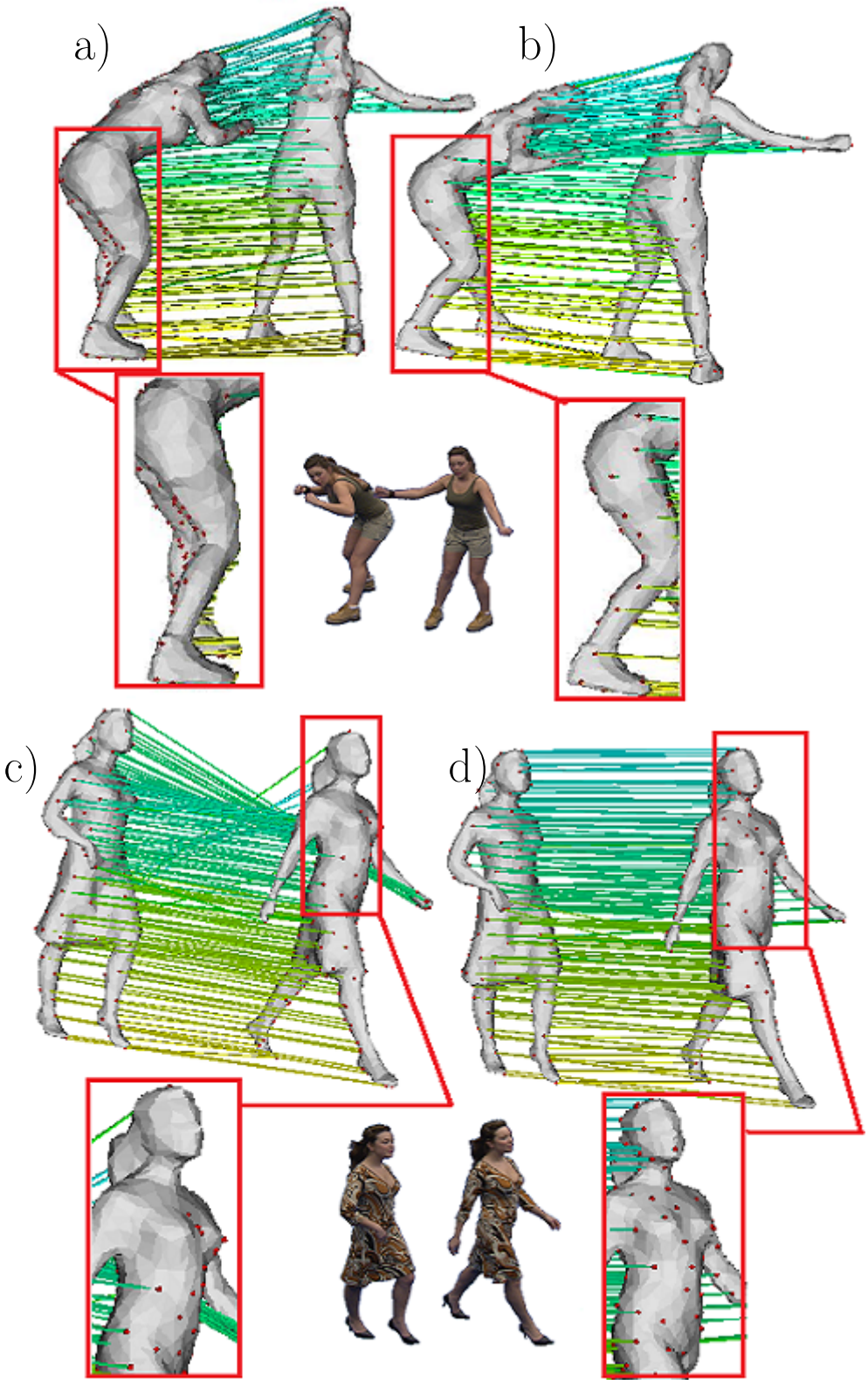


Figure 5.5: Results on a real-world dataset[Starck 2007b]. a) and c) show the results obtained using the method in [Bronstein 2006a]. b) and d) are obtained using our method.

$$E_{geom}(\Theta_i) = (d_{S^1}(P_i^1, P_j^1) - \Theta_i^T \boldsymbol{D}_{S^2}(t_i, t_j)\Theta_j)^2$$
$$= \Theta_i^T \boldsymbol{A}_i \Theta_i + 2b_i^T \Theta_i + c_i$$

where $\boldsymbol{D}_{S^2}(t_i, t_j)$ is a matrix containing all distances from the vertices of triangle t_i to all vertices of triangle t_j and

$$\boldsymbol{A}_i = \Sigma_{j \neq i} \boldsymbol{D}_{S^2}(t_i, t_j)\Theta_j \Theta_j^T \boldsymbol{D}_{S^2}(t_i, t_j)^T$$
$$b_i = -\Sigma_{j \neq i} d_{S^1} \boldsymbol{D}_{S^2}(t_i, t_j)\Theta_i$$
$$c_i = \Sigma_{j \neq i} d_{S^1}^2(P_i^1, P_j^1).$$

Taking the gradient with respect to Θ_i yields

$$\nabla E_{geom}(\Theta_i) = \boldsymbol{A}_i \Theta_i + \boldsymbol{b}_i.$$

Gradient of photometric term

The photometric term is taken to be the same as in the work of Keriven et.al. [Keriven 1998, Pons 2007b]. We assume the tangent planes aligned as explained in section 5.3.3. Then we fix all variables except Θ_i and compute the derivative with respect to Θ_i :

$$\frac{E_{photo}(\Theta_i)}{\partial \Theta_i} = \underbrace{\partial NCC}_{1\times d} \underbrace{DI}_{d\times 2} \underbrace{D\Pi}_{2\times 2}(\Theta_i)$$

d can be either 1 or 3 depending on whether one works with color or grey images. Then the gradient first involves the derivative of NCC with respect to the second argument since we keep fix the first argument. The exact computation of the expression can be found in [Pons 2007b]. DI involves a gradient in the image and is approximated using numerical central finite differences in the image. Finally, the linearization of the projection matrix is straight forward. The coordinates (u, v) in image I_i are obtained by projection

$$u(\Theta) = \frac{a_{11}\Theta X^T + a_{11}\Theta Y^T + a_{13}\Theta Z^T + q_1}{a_{31}\Theta X^T + a_{32}\Theta Y^T + a_{33}\Theta Z^T + q_3}$$
$$v(\Theta) = \frac{a_{21}\Theta X^T + a_{21}\Theta Y^T + a_{23}\Theta Z^T + q_1}{a_{31}\Theta X^T + a_{32}\Theta Y^T + a_{33}\Theta Z^T + q_3}.$$

$q = [q_1, q_2, q_3]^T$ and $Q = [Q_1, Q_2, Q_3]^T$ with $Q_{.,i} = [a_{.,1}, a_{.,2}, a_{.,3}]$ is the corresponding coordinate transformation $P^{2'} = QP^2 + q$ of the ith camera. (X, Y, Z) are the x,y and z coordinates of the triangle vertices such that $X = [x_i, x_2, x_3]^T$. The jacobian of Π_i is then obtained by differentiating u and v with respect to Θ.

5.4 Results

In order to validate the proposed method, we run several experiments on real and synthetic data. First, we test it on a synthetic dataset. In a second experiment, we validate our algorithm on real images.

5.4.1 Validation

Our first experiment focuses on the validation of our energy by testing the algorithm on a synthetic dataset (8 cameras) and comparing it to the result of [Bronstein 2006a]. This experiment aims at justifying the photometric part of the energy. The parameter α is set to 1.0 whereas β equals 8.0. We take 12 pairs of correspondences and a 3 level Gaussian pyramid. The advantages of our initialization having already been demonstrated in Figure 5.1, we prefer to launch the original method proposed by Bronstein *et al.* [Bronstein 2006a] with our initialization. The red dots in the left image of Figure 5.4a are some of the P_i^1 projected on the front image of S^1. The red dots in the right image of Figure 5.4a are the projections of the corresponding P_i^2 obtained after running the optimization of [Bronstein 2006a]. The green dots correspond to the result obtained with our combined photometric-geometric optimization. One can clearly see, the green dots are consistent with the initial sampling in the left image although the zone around the knee and shoulder exhibit elastic deformation. Whereas the red dots in the right image ignore the image signal and are pushed away by the local elastic deformations. Figure 5.4b shows the same points on the meshes.

5.4.2 Real data

In order to see how our method performs, we ran several experiments on image data courtesy of J. Starck[2] [Starck 2007b], again with 8 cameras. In this experiment the number of correspondences is 150 and the number of image levels is set to 3. α and β are set to 0.9 and 1.5 respectively. The results are depicted in Figure 5.5. Notice how the method of [Bronstein 2006a] fails to solve the matching problem. Local elastic deformations are observed in both cases (Figure 5.5a and 5.5c) and wrong matches occur because of symmetry. Nevertheless, our method can handle the symmetries and local elastic deformations as can be observed around the hair and the back in Figure 5.5b and in the zones located on the skirt and the hair in Figure 5.5d.

[2] `http://personal.ee.surrey.ac.uk/Personal/J.Starck/`

5.5 Discussion

We have described a variational framework for the correspondence problem of non-rigid surfaces coming from multi-view reconstruction or from any setup that provides images of the scene. By using photometric and geometric information, our method improves the one by Bronstein *et al.* [Bronstein 2006a] and allows to cope with elastic stretches and symmetries of the shape. Like in the initial work, and contrary to usual photometric methods, we are not limited to small deformations. Numerical experiments show the success of our method. Our future work includes extending our framework to surfaces with varying topology. A first step in this direction could be the use of more robust embedding such as the one proposed in [Bronstein]. Another improvement is the use of second order information in the gradient descent which allows to perform Newton steps. Finally, we notice that regularization is important in low-textured image regions. Therefore we currently investigate the effect of automatically setting the control weights as proposed in [Vu 2009].

Part III

Conclusion

Chapter 6

Conclusion

6.1 Conclusion

In this thesis, we exploited recent machine learning algorithms designed to compute isometric embeddings. Two cases were considered.

First, based on the diffusion maps methodology, we have proposed a deformable model framework for shape denoising where shapes are modeled as points in an infinite dimensional space. The major assumption of this method was that our data approximately lies on a manifold. Using diffusion maps, the dimensionality of the problem is reduced to the intrinsic dimension of the data. We approximate the manifold by linear complexes and capture the intrinsic geometry of the data. This provides a metric in the feature space allowing us to perform a gradient descent. We demonstrated the strength of our approach by applying these ideas in different experiments either with synthetic or real data. We stress the fact that the proposed method is general and is not necessarily restricted to specific shape representations. In particular, the only requirement is a differentiable distance (and differentiable kernel). We expect our approach to work on splines and other representations.

In the second part, we focused on the task of finding a set of correspondences satisfying geometric and photometric constraints. This kind of correspondence algorithm is especially well suited to the case of meshes output by multi-view reconstruction methods. Our method is again based on an energy measuring geometric (deviation from isometric deformation) and photometric consistency (normalized cross-correlation) and an energy minimization process. We tested the model on synthetic data as proof for the concept and on real-world dynamic stereo data. Currently, we are assembling a complete pipeline in the spirit of the standard 3D reconstruction pipeline for dynamic multiview stereo reconstruction. The pipeline consists of three stages. First, a global 4D reconstruction based on a quasi-dense point cloud is computed

using the method in [Aganj 2009]. Using our method this is put into a 4D mesh with fixed connectivity. Lastly, a global dense variational refinement is applied on the 4D mesh using the method in [Courchay 2009]. The results will be presented in a forthcoming paper.

Future Work

Of course, there is still room for improvement in many parts of this thesis. In the first part, we are convinced that our interpolation methods can be used in a number of computer vision problems, such as segmentation with a priori knowledge [Etyngier 2007b], shape tracking, and also shape registration. But a thorough experimental evaluation and modelization should be carried out in order to see if such implementations are possible. In the second part of this thesis our first results are promising and motivate further development of the method. Especially scalability should be investigated. Further it would be interesting to test the algorithm on several other datasets. This is currently under investigation.

Some avenues for future research are pointed out in the following lists

Diffusion Maps

In the case of shape analysis using diffusion maps, we could

- Apply our framework to shape segmentation with shape priors.
- Experiment on manifold denoising.
- Experiment with different data such as 2D and 3D Splines for character animation.

Isometric shape matching

- Compute the derivative with respect to the metric for isometric deformation prior like in [Devir 2009].
- Investigate the extension of GMDS proposed in [Bronstein] to cope with topological changes.
- Achieve scale independency of the GMDS method.

In the case of shape optimization with deformation prior, no work has ever before considered exploiting intrinsic shape prior such as pairwise geodesic distance preservation. Such a prior could play an essential role when segmenting tubular structures or calibrating cameras. Therefore it would be interesting to define an intrinsic energy based on geodesics acting as a regularizer for segmentation and other shape optimization problems (3D reconstruction).

List of Figures

1.1 A set of points in $\mathbb{R}^3$ lying on a manifold sampled from a 2-dimensional manifold and the corresponding parametrization f of the manifold mapping $\mathcal{M}$ ot $\mathbb{R}^2$.(taken from [Etyngier 2008]) 9

1.2 Each image is a vector in $\mathbb{R}^{4096}$, but we observe only three degrees of freedom: 2 rotation angles and 1 illumination parameter. Therefore the set of images span a 3-dimensional submanifold in $\mathbb{R}^{4096}$.(taken from `http://web.mit.edu/cocosci/isomap/`) 10

1.3 The Face data lies on a 3-dimensional curved submanifold $\mathcal{X}$ embedded in $\mathbb{R}^{4096}$. Linear combination $\bar{m}$ of two elements of $\mathcal{X}$ yields an element outside of $\mathcal{X}$. This shows that the dataspace is not a vector space. The mean between two faces $\tilde{m}$ yields an element of $\mathcal{X}$. 12

2.1 Travel length for the bird is shorter as the travel length of the ant. The ant is constrained to move in a different metric space than the bird. 23

2.2 A coordinate chart (U, ψ) 27

2.3 Change of coordinates $\psi \circ \phi^{-1}$ or transition map. 27

2.4 Example of an atlas for the circle with four coordinate charts. Each chart has an associated mapping (indicated by arrows) from the circle to an open interval. Assuming that the transition maps are smooth, the four charts cover the whole circle and therefore provide an atlas. 28

2.5 The tangent space $\mathbb{T}_x\mathcal{X}$ of $\mathcal{X}$ at x. The basis vectors ∂_1, ∂_2 span the tangent space $\mathbb{T}_x\mathcal{X}$ with $\dot{\Gamma}(t)|_{t=0} \in \mathbb{T}_x\mathcal{X}$. 30

3.1 a) 1D manifold and the distances between the original points(red) and their projection on the manifold. b) is a visualization of the geometry and constraints of the optimization problem. 40

3.2 The function f maps $\mathbf{X}$ from $\mathbb{R}^D$ to $\mathbf{Y}$ in $\mathbb{R}^d$ so that pairwise distances are as close as possible. 42

3.3 a) shows the result of the nearest K-neighbor rule for $K = 3$ and b) depicts the graph obtained by applying the ε distance rule. c) illustrates the problem when either the geometry is strongly curved or the sampling is not uniform. This can lead to cycles and or several components in the graph. In this example we used an ε neighborhood. 45

3.4 a) The sampling is not sufficiently dense to approximate the arc segment s_1 of Γ_1. b) depicts a sufficiently dense sampling to properly approximate the curve Γ_2 using Euclidean distance between sample points. 47

3.5 a) $\mathcal{X}$ is isometric to a convex open set b) $\mathcal{X}$ is not isometric to the parameter domain and therefore geodesics are not mapped correctly c) MDS fails because $\mathcal{X}$ is neither convex nor open . 50

3.6 a) Approximation of the vertex x_i by it's neighbors (x'_j, x_j). b) geometry of the optimization problem. $\mathbf{w}^T\mathbf{G}\mathbf{w}$ and the level sets are drawn in red. The constraint is visualized by the black line touching the level set at a single point. 51

3.7 Approximation of $\mathcal{X}$ in $\mathbb{T}_x\mathcal{X}$ using a local coordinate system $(x^1, \ldots, x^d)$ in $\mathbb{T}_x\mathcal{X}$. 53

3.8 Schematic visualisation of the proof of Theorem 16. 55

4.1 Embedding of $\mathcal{X}$ into $\mathcal{Y}$. $d_{\mathcal{X}}$ is a distance defined on $\mathcal{X}$ through a positive definite kernel. 73

4.2 a) shows a 2D curve and a normal deformation field. b) shows the embedding of the curves in a 2+1 dimensional space a the cross section at time t =0 and different shapes at different times (t_0, t_1). 77

4.3 a) Haussdorf distance between two curves. b) Area of symmetric distance. 79

4.4 Problem configuration: We are looking for the pre-image $\hat{x}$ of Υ in the input space $\mathcal{X}$ 84

4.5 Nyström extension of the first embedding coordinate (from blue corresponding to smaller coordinates to red corresponding to larger ones) provided by the leading eigenvector in different kernel methods : Kernel PCA without (a) and with normalization [Arias 2007] (b), Kernel PCA in centered feature space without (c) and with normalization (d) 85

4.6 a) Visualization of the feature points(blue) geometry in $\mathcal{Y}$ and the affine subspace(red circle); b) Affine subspace $\mathbb{S}_{p-1}$ 87

4.7 Digit images corrupted by additive Gaussian noise (from top to bottom, $\sigma = 0.25, 0.45, 0.65$). The different rows respectively represent: the original digits and corrupted digits; different reconstruction methods: [Dambreville 2006] ; [Dambreville 2006] with normalization ; [Kwok 2004] ; [Kwok 2004] with normalization. 90

4.8 Digit images corrupted by additive Gaussian noise (from left to right, $\sigma^2 = 0.25, 0.45, 0.65, 0.85$). The different rows respectively represent, from top to bottom: the original digits; the corrupted digits; denoising with [Dambreville 2006] ; with [Dambreville 2006]+[Arias 2007] ; with [Kwok 2004] ; with [Kwok 2004]+[Arias 2007] ; with [Carreira-Perpiñan 2007]; with our Karcher mean based method. See table 4.3 for quantified results 93

4.9 Synthetic sample of 200 articulated and elongated shapes. From left to right: (a) a subset of the sample. (b) triangulated two-dimensional embedding computed using diffusion maps and a gradient descent from an initial shape to a target one, without (red dots) and with (purple dots) remaining on the interpolated manifold. (c) Some shapes of the resulting evolution (left column: without projection, right column: with projection. . . . 98

4.10 Interpolation using Karcher means for 39 three-dimensional sample shapes. From left to right: a) a new shape not in the given sample b) the same shape with an occlusion c) the 3 nearest neighbors of the corrupted shape according to the diffusion distance (in red, green and blue) d) the original shape (in yellow) and our interpolation (in red). See text for quantitative results. 99

5.1 a) Initialization using solely geometric information is sensible to symmetry in shapes. b) Color coded labeling of the meshes (see text) and our photometric plus geometric initialization, in which ambiguities have been solved. 113

5.2 A visualization describing the problem configuration as in section 5.3.1. 118

5.3 This visualization shows the interaction between the geometric and the photometric gradient in a triangle composed by the vertices (v_1, v_2, v_2). The point $P = \Theta^T V$ is given as a weighted combination of the vertices. P travels along the red arrow which is calculated as a linear combination of the geometric term ∇E_{geom} and the photometric term ∇E_{photo}. 120

5.4 a) On the left, the front image of the first mesh, red dots being the projections of some key points. On the right, the front image of the second mesh, showing the reprojections of the corresponding points, in red obtained with the method in [Bronstein 2006a] (initialized with our method), in green obtained with our method. b) Shows the same correspondences directly on the meshes with the same color code. The inaccuracies of [Bronstein 2006a], due to local elastic deformations, are corrected by our photometric + geometric criterion. 122
5.5 Results on a real-world dataset[Starck 2007b]. a) and c) show the results obtained using the method in [Bronstein 2006a]. b) and d) are obtained using our method. 125

List of Tables

3.1 Overview of unsupervised distance based learning methods and their properties. 61

4.1 Average PSNR (in dB) of the denoised images corrupted by different noise level. Training set is composed of 60 samples (first 4 rows) and 200 samples (last 4 rows). The first and third column show the denoising results without and the second and last columns with the normalization we proposed. 92
4.2 Average reconstruction error for a set of 9 noisy shapes 100
4.3 Average PSNR (in dB) of the denoised images corrupted by different noise levels σ. Training sets consist in 60 samples (first 4 rows) and 200 samples (last 4 rows). 102

Appendix A

Computing $\mathbb{T}_{s_\Theta}\mathcal{X}$

A.1 Interpolation by projecting into the tangent space

As a first step we compute the eigenvectors $\{\Psi_i\}$ using Diffusion Maps. For a given sampe s_1 we compute the iso-embedding Υ of s_1 using the Nyström extension. We intialise the shape z as the nearest neighbor of Υ in the reduced space. Now we want to warp shape z such that is has the same embedding as Υ. This can be done by minimizing the diffusion distance

$$\hat{s} = \arg\min_{s\in\mathbb{S}} \|\Psi(s) - \underbrace{\Upsilon}_{const}\|^2. \tag{A.1}$$

Minimizing this functional can be done by taking its gradient and perform a gradient descent. Since the only requirement to perform a gradient descent w.r.t to the diffusion distance is the use of a differentiable kernel[Etyngier 2007b]. It is obvious that a simple gradient descent doesn't generate proper intermediate shapes lying on the manifold. This observation is due to the fact that the gradient does not respect the curved geometry of the data. Our goal is to define an interpolation between shapes constrained such that intermediate shapes lie on the manifold. Therefore, we introduce a constraint which is expressed as Karcher mean:

$$E = \arg\min_{s,\theta} = \sum_{1}^{d+1} \theta_i d(s, s_i)^2$$
$$\text{such that } \sum \theta_i = 1 \text{ and } \theta_i \geq 1$$

The first order optimality condition of the Karcher mean constraint must satisfy the following equation

$$\nabla|_s E = 2\sum_1^{d+1} \theta_i d(s, s_i) \frac{\partial}{\partial s} d(s, s_i) = 0.$$

Hence any variation in s at s_0 yields a deformation field which increases our energy functional.
We are now looking for the differential equation which describes the motion constrained to the manifold $\mathcal{X}$ in terms of the θ_i and the shape gradient. More precisely, we want to estimate the tangent space $\mathbb{T}\mathcal{X}$ at every point on the manifold so that we can constrain the gradient flow to the manifold by projection onto tangent space. We assume $s \in \mathcal{X}$, then any variation of the coefficient $\theta_i \mapsto \theta_i + d\theta_i$ is related to a deformation field of the curve $s \mapsto s + d\beta$. Using the Calculus of Variation, we want to minimize the functional $E(s_\theta) = \int_\Omega F(s, \theta) dx$ depending on s and θ and Ω is a certain integration domain. We compute the derivative of E

$$\frac{dE}{d\varepsilon}|_{\varepsilon=0} = \frac{E(s_\theta + \varepsilon\psi) - E(s_\theta)}{\varepsilon} = \langle \nabla E, \psi \rangle = 0$$

for all test functions $\psi \in \mathcal{D}$ (space $\mathcal{D}$ of test functions). Remembering the coupling of θ and s, we have $\nabla E = \frac{\partial E}{\partial \theta} + \frac{\partial E}{\partial s} d\beta$.

Summary :

ϕ is the levelset representation of a shape s. Then the optimality condition for any choice of distance $d_i = d(\phi, \phi_i) =< \phi, \phi_i >$ is:

$$\forall \psi \ \sum_1^{d+1} \theta_i d_i D_\phi d_i(\psi) = 0.$$

For the dot product induced by the distance d, we have:

$$D_\phi d_i(\psi) =< \nabla_\phi d_i, \psi >,$$

which gives the following optimality condition:

$$\sum_1^{d+1} \theta_i d_i \nabla_\phi d_i = 0.$$

Differentiating this condition leads to:

$$0 = \sum_1^{d+1} \left\{ \begin{array}{c} d\theta_i d_i \nabla_\phi d_i \\ +\theta_i < \nabla_\phi d_i, \beta > \nabla_\phi d_i \\ +\theta_i d_i H_i \beta \end{array} \right\},$$

which we can write as:

$$0 = \sum_{1}^{d+1} d\theta_i d_i \nabla_\phi d_i + \sum_{1}^{d+1} \theta_i (\nabla_\phi d_i . \nabla_\phi^T d_i + d_i H_i)\beta.$$

In this formulation H_i is the Hessian matrix of the distance d_i: $H_i = \nabla_\phi \nabla_\phi d_i$. We can find a basis for the tangent space at location $\{\theta_1, \ldots, \theta_{d+1}\}$ as long as

$$\beta \mapsto \sum_{1}^{d+1} \theta_i (\nabla_\phi d_i . \nabla_\phi^T d_i + d_i H_i)\beta$$

is invertible.

A.1.1 Symmetric Difference

In this section, we consider the symmmetric difference function

$$d(s_1, s_2) = \frac{1}{2} \int (\chi_{\Omega_1} - \chi_{\Omega_2})^2 dx$$

as given in the paper of Solem. Its gradient writes as

$$\nabla f(x) = (\frac{1}{2} - \chi\Omega_0).$$

Proof: using the unit step function $H : x \in \mathbf{R} \mapsto \{0, 1\}$ with $H(x) = 1$ iff $x \geq 0$, we have:

$$\begin{aligned} d(S_1, S_2) &= \frac{1}{2} \int (\chi_{\Omega_1} - \chi_{\Omega_2})^2 dx, \\ &= \frac{1}{2} \int (H(\phi_1(x)) - H(\phi_2(x)))^2 dx, \\ &= d(\phi_1, \phi_2) \end{aligned}$$

We can express this distance as $d(\phi_1, \phi_2) = \frac{1}{2} \int H(\phi_1(x))(1 - 2H(\phi_2(x)))dx + f(\phi_2)$ ($f(\phi_2)$ is not relevant for differentiation with respect to ϕ_1), which implies that:

$$\begin{aligned} D_{\phi_1} d(\phi_1, \phi_2)(\psi) &= lim_{s \mapsto 0} \frac{d(\phi_1 + s\psi, \phi_2) - d(\phi_1, \phi_2)}{s} \\ &= \frac{1}{2} \int \psi(x)(1 - 2H(\phi_2(x)))\delta(\phi_1(x))dx \\ &= < \psi(x), \underbrace{\frac{1}{2}(1 - 2H(\phi_2(x)))\delta(\phi_1(x))}_{\nabla_{\phi_1} d(\phi_1, \phi_2)} > . \end{aligned}$$

This shows directly that the gradient (with respect to the image domaine ϕ_1) is $(\frac{1}{2} - H(\phi_2))\delta(\phi_1)$, or equivalently with respect to the normal field to the curve Γ_1: $(\frac{1}{2} - \chi\Omega_2)$ since:

$$\begin{aligned} D_{\phi_1} d(\phi_1, \phi_2)(\psi) &= \frac{1}{2}\int_{\mathbf{R}^n} \underbrace{\frac{\psi(x)}{|\psi(x)|}}_{\beta}(1 - 2H(\phi_2(x)))\underbrace{\delta(\phi_1(x))|\psi(x)|dx}_{d\sigma} \\ &= \int_\Gamma \beta(\frac{1}{2} - \chi_{\Omega_2})d\sigma, \\ &= < \beta, (\frac{1}{2} - \chi_{\Omega_2}) >_\Gamma \end{aligned}$$

with $\beta = \frac{\psi}{|\psi|}$ the normal deformation to the curve. It is a result from geometric integration theory that the Euclidean measure on Γ can be expressed as $d\sigma = \delta(\phi_1(x))|\psi(x)|dx$. $\delta \in \mathcal{D}(\mathbb{R})$ is the Driac distribution on the real line. Precomposition of δ by ϕ is nothing then the pullback of δ by ϕ and indicates a change of variable. So it follows that

$$|\Gamma| = \int_\Gamma |\nabla\phi|\delta(\phi)dx = \int_{\Re^{d+1}} |\nabla H(\phi)|dx$$

The complete proof can be found in [Aubert 2006].

A.1.2 L2 distance

In this section we consider the L2 distance between two distance functions ϕ_1, ϕ_2

$$d(\phi_1, \phi_2) = \int (\phi_1 - \phi_2)^2 dx$$

The gradient of $d(\phi_1, \phi_2)$ computes as

$$\begin{aligned} \lim_{\varepsilon\to 0} &= \frac{\int(\phi_1 + \varepsilon\psi\delta(\phi_1) - \phi(x)_i dx)^2 - \int(\phi_1(x) - \phi_2(x))^2 dx}{\varepsilon} \\ &= \frac{\int(\phi_1 - \phi_2)^2 - 2\varepsilon\psi\delta(\phi_1(x))(\phi_1(x) - \phi_2(x)) + \varepsilon^2\psi\delta(\phi_1) - (\phi_1(x) - \phi_2(x))}{\varepsilon} \\ &= \frac{2\varepsilon\psi\delta(\phi_1)(\phi_1 - \phi_2) + (\varepsilon\psi\delta(\phi_1))^2}{\varepsilon} \\ &= 2\psi\delta(\phi_1)(\phi_1 - \phi_i) \text{ valid for test functions } \psi \\ &= 2\delta(\phi_1)(\phi_1 - \phi_2). \end{aligned}$$

Therefore we have

$$D_{\phi_1} d(\phi_1, \phi_2) = 2\delta(\phi_1)(\phi_1 - \phi_2).$$

Appendix B

Gradient of the Diffusion maps : $\nabla_s \Psi(s)$

B.1 Density independent kernel

Using the diffusion maps of Coifman[Coifman 2005], similarity between shapes are defined via the diffusion distance of the Laplace-Beltrami Operator on the manifold of shapes. Therefore minimizing the distance between shapes can be done via the gradient of the diffusion distance. In the following, we review the diffusion distance following the work of Coifman and show how to compute the gradient. Let $\Gamma = \{s_1, \cdots, s_N\} \in \mathcal{X}$ be N sample points of the d-dimensional manifold $\mathcal{X}$ sampled under unknown density. From this sample points, we build the adjacency matrix $(\boldsymbol{W}_{ij})$ between points, where connecting edges are weighted with a positive decraesing function of the distances between shapes. In our case we use

$$w(s_i, s_j) = \exp(-d^2(s_i, s_j)/2\sigma^2)$$

where σ is the median distance between all shapes. By normalizing the adjacency matrix

$$\tilde{w}(s_i, s_j) = \frac{w(s_i, s_j)}{q(s_i)q(s_j)} \text{with } q(s) = \sum w(s, y)$$

we overcome the problem of unknown density and contruct an ansisotropic kernel $(P_{i,j})$ such that

$$p(s_i, s_j) = \frac{\tilde{w}(s_i, s_j)}{\tilde{q}(s_i)} \text{with } q(\tilde{s}) = \sum \tilde{w}(s, y).$$

Then the kernel $\boldsymbol{Id} - \boldsymbol{P}$ is an density independent approxiamtion of the Laplace Beltrami operator on $\mathcal{X}$.

B.1.1 Computing $\tilde{w}(.,.)$ and $p(.,.)$

$$\tilde{w}(s_i, s_j) = \frac{w(s_i, s_j)}{q(s_i)q(s_j)} \text{with } q(s) = \sum w(s, y)$$

then $p(.,.)$ computes as

$$\begin{aligned} p(s_i, s_j) &= \frac{\tilde{w}(s_i, s_j)}{\tilde{q}(s_i)} \\ &= \frac{\frac{w(s_i,s_j)}{\cancel{q(s_i)}q(s_j)}}{\sum_b \frac{w(s_i,s_j)}{\cancel{(q(s_i)}q(b))}} \\ &= \frac{w(s_i, s_j)}{\sum_b \frac{q(s_j)}{q(s_b)} w(s_i, s_j)} \\ &= \frac{w(s_i, s_j)}{\sum_b K_{jb} w(s_i, s_b)} \text{with } K_{jb} = \frac{\sum_a w(s_a, s_b)}{\sum_d w(s_d, s_b)}. \end{aligned}$$

B.1.2 Derivative of the anisotropic kernel $p(.,.)$

$$\begin{aligned} p(s, s_j) &= \frac{w(s, s_j)}{\sum_b K_{jb} w(s, s_b)} \\ \frac{\partial}{\partial s} p(s, s_j) &= \frac{1}{\left(\sum_b K_{jb} w(s, s_b)\right)^2} \left(\underbrace{w'(s, s_j)}_{*1} \sum_b K_{jb} w(s, s_b) - w(s, s_j) \sum_b K_{jb} \underbrace{w'(s, s_b)}_{*} \right) \\ &= \frac{1}{\left(\sum_b K_{jb} w(s, s_b)\right)^2} \sum_{b \neq j} w'(s, s_j) w(s, s_b) - K_{jb} w(s, s_b) w'(s, s_j) \end{aligned}$$

We finally obtain the gradient of the embedding $\hat{\Psi}$ with respect to s become

$$\begin{aligned} \hat{\Psi}(s) &= \lambda_k^{-1} \sum_j p(s, s_j) \Psi_k(s_j) \\ \nabla_s \hat{\Psi}(s) &= \lambda_k^{-1} \sum_j \nabla_s p(s, s_j) \Psi_k(s_j). \end{aligned}$$

B.2 Approximation of Heavyside function

$$H_\alpha(x) \begin{cases} \frac{1}{2}(1 + \frac{x}{\alpha} + \frac{1}{\pi} sin(\frac{\pi x}{\alpha})) & \text{if } |x| \leq \alpha, \\ 1 & \text{if } x > \alpha, \\ 0 & \text{if } x < -\alpha, \end{cases}$$

B.3 Approximation of Dirac function

$$\delta_\alpha(x) \begin{cases} \frac{1}{2\alpha}(1 + cos(\frac{\pi x}{\alpha})) & \text{if } |x| \leq \alpha, \\ 0 & \text{if } |x| \geq \alpha. \end{cases}$$

Appendix C

Gradient of distance function

C.1 Gradient of the distance function on a triangle mesh

Given a triange defined through three vertices v_0, v_1, v_2. d_0, d_1, d_2 are the corresponding distances defined on each vertex computed using the Fast Marching method for triangle meshes. Let e_0, e_1, e_2 be the edges between the vertices such that $e_0 = (v_0 - v_2)/\|v_0 - v_2\|$ and $e_1 = (v_1 - v_2)/\|v_1 - v_2\|$.. Then we compute the gradient at a point in a local frame in [Peyré 2003]. The gradient $(g_u, g_v)^T$of the distance function U in directions (e_0, e_1) is

$$\begin{pmatrix} \langle \nabla U, e_0 \rangle \\ \langle \nabla U, e_1 \rangle \end{pmatrix} = \begin{pmatrix} (d_0 - d_2)/\|e_0\| \\ (d_1 - d_2)/\|e_1\| \end{pmatrix} = \begin{pmatrix} g_u \\ g_v \end{pmatrix}.$$

We are searching for the gradient of the distance function which is defined as $\nabla U = dxe_0 + dye_1$. This gives rise to the system

$$\begin{pmatrix} \langle e_0, e_0 \rangle & \langle e_0, e_1 \rangle \\ \langle e_0, e_1 \rangle & \langle e_1, e_1 \rangle \end{pmatrix} \begin{pmatrix} dx \\ dy \end{pmatrix} = \begin{pmatrix} g_u \\ g_v \end{pmatrix}.$$

The solution $(dx, dy)^T$ is obtained by solving the previous system.

Bibliography

[A. Besbes 2009] N. Paragios A. Besbes N. Komodakis and G. Langs. *Shape Priors and Discrete MRFs for Knowledge-based Segmentation.* IEEE Conference in Computer Vision and Pattern Recognition, 2009. 103

[Aganj 2009] E. Aganj, J.-P. Pons and R. Keriven. *Globally optimal spatio-temporal reconstruction from cluttered videos.* In Asian Conference on Computer Vision, Xi an, China, Sep 2009. 117, 132

[Agarwal 2005] P.K. Agarwal, S. Har-Peled and K.R. Varadarajan. *Geometric approximation via coresets*, 2005. 103

[Ahmed 2008] N. Ahmed, C. Theobalt, C. Roessl, S. Thrun and Seidel. *Dense Correspondence Finding for Parametrization-free Animation Reconstruction from Video.* In IEEE International Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, June 2008. 117

[Allène 2008] C. Allène, J-P. Pons and R. Keriven. *Seamless image-based texture atlases using multi-band blending.* In 19th International Conference on Pattern Recognition, Tampa, US, Dec 2008. 121, 123

[Alt 1996] Helmut Alt and Leonidas J. Guibas. *Discrete Geometric Shapes: Matching, Interpolation, and Approximation A Survey.* Rapport technique B 96-11, Freie Universit"at Berlin, 1996. 109

[Amari 1999] S. Amari and S. Wu. *Improving support vector machine classifiers by modifying kernel functions.* In Neural Networks 12, pages 783–789, 1999. 72

[Amberg 2007] B. Amberg, S. Romdhani and T. Vetter. *Optimal Step Nonrigid ICP Algorithms for Surface Registration.* IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2007. 114

[Arias 2007] Pablo Arias, Gregory Randall and Guillermo Sapiro. *Connecting the out-of-sample and pre-image problems in kernel methods.* In IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 18-23 jun 2007. 70, 84, 85, 87, 88, 89, 91, 93, 100, 102, 136, 137

[Aubert 2006] G. Aubert and P. Kornprobst. Mathematical problems in image processing: Partial differential equations and the calculus of variations (second edition), volume 147 of *Applied Mathematical Sciences.* Springer-Verlag, 2006. 144

[Audette 1999] Michel A. Audette, Frank P. Ferrie and Terry M. Peters. *An Algorithmic Overview of Surface Registration Techniques for Medical Imaging.* Medical Image Analysis, vol. 4, pages 201–217, 1999. 109

[Bâdoiu 2003] Mihai Bâdoiu and Kenneth L. Clarkson. *Smaller core-sets for balls.* In SODA '03: Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms, pages 801–802, Philadelphia, PA, USA, 2003. Society for Industrial and Applied Mathematics. 103

[Baker 1996] Christopher T. H. Baker and Christopher T. H. Baker. *Numerical analysis of Volterra functional and integral equations.* In In I.S. Duff G.A. Watson (eds), The state of the art in numerical analysis, p193–222, pages 193–222. University Press, 1996. 85

[Beg 2005] M. Faisal Beg, Michael I. Miller, Alain Trouvé; and Laurent Younes. *Computing Large Deformation Metric Mappings via Geodesic Flows of Diffeomorphisms.* Int. J. Comput. Vision, vol. 61, no. 2, pages 139–157, 2005. 68

[Belkin 2003] M. Belkin and P. Niyogi. *Laplacian Eigenmaps for Dimensionality Reduction and Data Representation.* Neural Computation, vol. 15, no. 6, pages 1373–1396, 2003. 55, 57, 58

[Belkin 2008] Mikhail Belkin and Partha Niyogi. *Towards a theoretical foundation for Laplacian-based manifold methods.* J. Comput. Syst. Sci., vol. 74, no. 8, pages 1289–1308, 2008. 56

[Belongie 2002] Serge Belongie, Jitendra Malik and Jan Puzicha. *Shape matching and object recognition using shape contexts.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, pages 509–522, 2002. 112

[Bengio 2003] Y. Bengio, P. Vincent and et al. *Spectral Clustering and Kernel PCA are Learning Eigenfunctions.* Rapport technique 1239, Département d'informatique et recherche opérationnelle, Université de Montréal, 2003. 71

[Bengio 2004] Yoshua Bengio, Jean-Francois Paiement, Pascal Vincent, Olivier Delalleau, Nicolas Le Roux and Marie Ouimet. *Out-of-Sample Extensions for LLE, Isomap, MDS, Eigenmaps, and Spectral Clustering.* In Sebastian Thrun, Lawrence K. Saul and Bernhard Schölkopf, editeurs, Advances in Neural Information Processing Systems 16. MIT Press, Cambridge, MA, 2004. 85

[Bernstein 2000] M. Bernstein, V. de Silva, J.C. Langford and J.B. Tenenbaum. *Graph approximations to geodesics on embedded manifolds.* Technical Report, 2000. 46, 47

[Besl 1992] P. Besl and N. McKay. *A Method for Registration of 3-D Shapes.* PAMI, vol. 14, no. 2, pages 239–256, 1992. 110, 112

[Bookstein 1989] F. L. Bookstein. *Principal warps: thin-plate splines and the decomposition of deformations.* Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 11, no. 6, pages 567–585, 1989. 110

[Bronstein] A. Bronstein, R. Kimmel, M. Mahmoudi and G. Sapiro. *A Gromov-Hausdorff framework with diffusion geometry for topologically-robust non-rigid shape matching.* International Journal of Computer Vision (IJCV). 128, 132

[Bronstein 2006a] A. Bronstein, M. Bronstein and R. Kimmel. *Efficient computation of isometry-invariant distances between surfaces.* In SIAM J. Sci. Comput., volume 28, 2006. 115, 117, 118, 119, 122, 123, 124, 125, 127, 128, 138

[Bronstein 2006b] A. Bronstein, M. Bronstein and R. Kimmel. *Generalized multidimensional scaling: a framework for isometry-invariant partial surface matching.* Proc. National Academy of Sciences (PNAS), vol. 103, no. 5, pages 1168–1172, January 2006. 12, 15, 16, 107, 110, 111, 119

[Bronstein 2006c] M. Bronstein, A. Bronstein and R. Kimmel. *Efficient Computation of Isometry-Invariant Distances Between Surfaces.* In Techn. Report CIS-2006-02, Dept. of Computer Science, Technion, Israel, 2006. 123, 124

[Bronstein 2007] A. Bronstein, M. Bronstein and R. Kimmel. *Calculus of non-rigid surfaces for geometry and texture manipulation.* IEEE Trans. Visualization and Computer Graphics, 2007. 107

[Brown 2007] B Brown and S. Rusinkiewicz. *Global Non-Rigid Alignment of 3-D Scans.* ACM Transactions on Graphics (Proc. SIGGRAPH), vol. 26, no. 3, August 2007. 114

[Buades 2005] Antoni Buades and Bartomeu Coll. *A non-local algorithm for image denoising.* In In CVPR, pages 60–65, 2005. 69

[Burago 2001] Dmitri Burago, Yuri Burago and Sergei Ivanov. A course in metric geometry. AMS Graduate Studies in Mathematics, 2001. 15, 20, 24

[C. Florin 2007] G. Funka-Lea J. Williams. C. Florin N. Paragios. *Liver Segmentation Using Sparse 3D Prior Models with Optimal Data Support.* Information Processing in Medical Imaging, 2007. 103

[Carreira-Perpiñan 2007] M. Carreira-Perpiñan and Z. Lu. *The Laplacian Eigenmaps Latent Variable Model.* Journal of Machine Learning Research, vol. 2, pages 59–66, 2007. 70, 93, 94, 100, 102, 137

[Charpiat 2005] G. Charpiat, O. Faugeras and R. Keriven. *Approximations of Shape Metrics and Application to Shape Warping and Empirical Shape Statistics.* Foundations of Computational Mathematics, vol. 5, no. 1, pages 1–58, 2005. 15, 68, 79, 95, 96, 99

[Charpiat 2006a] G. Charpiat. *Distance-based shape statistics for image segmentation with priors.* PhD thesis, Ecole Polytechnique, 2006. 76, 107

[Charpiat 2006b] G. Charpiat, O. Faugeras, R. Keriven and P. Maurel. *Distance-Based Shape Statistics.* In IEEE International Conference on Acoustics, Speech and Signal Processing, volume 5, pages 925–928, 2006. 68

[Charpiat 2007] G. Charpiat, P. Maurel, J.-P. Pons, R. Keriven and O. Faugeras. *Generalized Gradients: Priors on Minimization Flows.* International Journal of Computer Vision, 2007. 81

[Chunguang 007] Li Chunguang, Guo Jun and Nie Xiangfei. *Intrinsic Dimensionality Estimation with Neighborhood Convex Hull.* International Conference on Computational Intelligence and Security, 2007. 11

[Cohen 1996] L. Cohen. *Auxiliary variables and two-step iterative algorithms in computer vision problem.* volume 6, pages 59–83, 1996. 114

[Coifman 2005] R. Coifman, S. Lafon, A. Lee, M. Maggioni, B. Nadler, F. Warner and S. Zucker. *Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps.* PNAS, vol. 102, no. 21, pages 7426–7431, 2005. 57, 58, 145

[Cootes 1995] T. Cootes, C. Taylor, D. Cooper and J. Graham. *Active Shape Models: their Training and Application.* Computer Vision and Image Understanding, vol. 61, no. 1, pages 38–59, 1995. 68

[Courchay 2009] Jérôme Courchay, Jean-Philippe Pons, Pascal Monasse and Renaud Keriven. *Dense and Accurate Spatio-Temporal Multi-View Stereovision.* In Asian Conference on Computer Vision, Xi an, China, September 2009. 116, 117, 118, 132

[Cremers 2004] Daniel Cremers, Stanley Osher and Stefano Soatto. *Kernel density estimation and intrinsic alignment for knowledge-driven segmentation: Teaching level sets to walk.* LNCS, vol. 3175, no. 2, page 3644, 2004. 78

[Dambreville 2006] S. Dambreville, Y. Rathi and A. Tannenbaum. *Statistical shape analysis using kernel PCA.* IST SPIE Symposium on Electronic Imaging, 2006. 69, 70, 84, 88, 89, 90, 91, 92, 93, 94, 98, 100, 102, 137

[Dasgupta 2007] S. Dasgupta, C.H. Papadimitriou and U.V. Vazirani. Algorithms. McGrawHill, 2007. 124

[Davis 2007] B. Davis, P. Fletcher, E. Bullitt and S. Joshi. *Population Shape Regression From Random Design Data.* Iccv, vol. 1, 2007. 94

[de Silva 002] V. de Silva and J. B. Tenenbaum. *Unsupervised Learning of Curved Manifolds.* Nonlinear Estimation and Classification, 2002. 49

[Devir 2009] Y. Devir, G. Rosman, A. M. Bronstein, M. M. Bronstein and R. Kimmel. *On reconstruction of non-rigid shapes with intrinsic regularization.* Proc. Workshop on Nonrigid Shape Analysis and Deformable Image Alignment (NORDIA), 2009. 132

[Dijkstra 1959] E. W. Dijkstra. *A note on two problems in connexion with graphs.* Numerische Mathematik, vol. 1, pages 269–271, 1959. 46

[Do carmo 1992] Manfredo Perdigão Do carmo. Riemannian geometry. Birkhauser, 1992. 13, 20, 26, 33

[Dong 2008] Bin Dong, Jian Ye, Stanley Osher and Ivo Dinov. *Level Set Based Nonlocal Surface Restoration.* Multiscale Model. Simul., vol. 7, no. 2, pages 589–598, 2008. 69

[Donoho 1995] D. Donoho. *De-noising by soft-thresholding.* IEEE Transactions on Information Theory, vol. 41, pages 613–627, 1995. 69

[Donoho 2003] D. L. Donoho and C. Grimes. *Hessian eigenmaps: New locally linear embedding techniques for high dimensional data.* Technical Report 2003-08, Stanford Statistics Department, 2003. 52, 54

[Eckart 1936] C. Eckart and G. Young. *The approximation of one matrix by another of lower rank.* In Psychometrika, volume 1, pages 211–218, 1936. 43

[Elad 2001] Asi Elad and Ron Kimmel. *Bending Invariant Representations for Surfaces*, 2001. 107

[Elad 2003] A. Elad and R. Kimmel. *On bending invariant signatures for surfaces.* In IEEE Trans. on Pattern Analysis and Machine Intell., volume 25 of *LNCS*, page 12851295, October 2003. 115, 117

[Essafi 2009] S. Essafi, G. Langs and N. Paragios. *Left Ventricle Segmentation Using Diffusion Wavelets Prior and Boosting.* International Conference on Medical Image Computing and Computer Assisted Intervention., 2009. 103

[Esteban 2008] Carlos Hernandez Esteban, George Vogiatzis and Roberto Cipolla. *Multiview Photometric Stereo.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 30, no. 3, pages 548–554, 2008. 117

[Etyngier 2007a] P. Etyngier, R. Keriven and F Ségonne. *Projection Onto a Shape Manifold for Image Segmentation with Prior.* In 14th IEEE International Conference on Image Processing, San Antonio, Texas, US, Sep 2007. 95

[Etyngier 2007b] P. Etyngier, F. Ségonne and R. Keriven. *Shape priors using Manifold Learning Techniques.* In 11th IEEE International Conference on Computer Vision, Rio de Janeiro, Brazil, Oct 2007. 15, 69, 94, 99, 132, 141

[Etyngier 2008] Patrick Etyngier. *Statistical learning, Shape Manifolds and Applications to Image Segmentation.* PhD thesis, Ecole Nationale des Ponts et Chausees, 2008. 9, 41, 135

[Fan 2009] *Intrinsic dimension estimation of manifolds by incising balls.* Pattern Recognition, vol. 42, no. 5, pages 780 – 787, 2009. 11

[Forsyth 2002] David A. Forsyth and Jean Ponce. Computer vision: A modern approach. Prentice Hall, 2002. 7

[Franco 2003] J.-S. Franco and E. Boyer. *Exact Polyhedral Visual Hulls.* In British Machine Vision Conference, volume 1, pages 329–338, 2003. 117

[Fréchet 1944] M. Fréchet. *L'intégrale abstraite d'une fonction abstraite d'une variable abstraite et son application á la moyenne d'un élement aléatoire de nature quelconque.* Revue Scientifique, pages 438–512, 1944. 82

[Fréchet 1948] M. Fréchet. *Les éléments aléatoires de nature quelconque dans un espace distancié.* pages 215–310, 1948. 82

[Furukawa 2008] Y. Furukawa and J. Ponce. *Dense 3D Motion Capture from Synchronized Video Streams.* In IEEE International Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, June 2008. 115

[Gal 2006] Ran Gal and Daniel Cohen-Or. *Salient geometric features for partial shape matching and similarity.* ACM Trans. Graph., vol. 25, no. 1, pages 130–150, 2006. 109

[Gelfand 2005] N. Gelfand, N. J. Mitra, L. J. Guibas and H. Pottmann. *Robust global registration.* In In M. Desbrun and H. Pottmann, editors, Eurographics Association, ISBN 3-905673-24-X., pages 197–206, 2005. 112

[Goldlücke 2004] B. Goldlücke and M. Magnor. *Space-time Isosurface Evolution for Temporally Coherent 3D Reconstruction.* In IEEE Conference on Computer Vision and Pattern Recognition, volume 1, pages 350–355, 2004. 117

[Grenander 1993] U. Grenander. General pattern theory. Oxford University Press, 1993. 68

[Gu 2004] Xianfeng Gu and Baba C. Vemuri. *Matching 3d shapes using 2d conformal representations.* In In MICCAI, pages 771–780, 2004. 110

[Guichard 2001] Frederic Guichard and Jean-Michel Morel. *Image Analysis and P.D.E.s.* IPAM GBM Tutorial, 2001. 69

[Ham 2003] J. Ham, D. D. Lee, S. Mika and B. Schölkopf. *A Kernel View of the Dimensionality Reduction of Manifolds.* Rapport technique 110, Max-Planck-Institut für Biologische Kybernetik, Tübingen, Germany, 2003. 71, 88

[Hastie 2003] Trevor Hastie, Robert Tibshirani and Jerome Friedman. The elements of statistical learning: Data mining, inference, and prediction. Springer, 2003. 8

[Hein 2005a] M. Hein, J.-Y. Audibert and U. von Luxburg. *From Graphs to Manifolds - Weak and Strong Pointwise Consistency of Graph Laplacians.* In Peter Auer and Ron Meir, editeurs, COLT, volume 3559 of *Lecture Notes in Computer Science*, pages 470–485. Springer, 2005. 59

[Hein 2005b] Matthias Hein and Jean-Yves Audibert. *Intrinsic dimensionality estimation of submanifolds in Rd.* In ICML '05: Proceedings of the 22nd international conference on Machine learning, pages 289–296, New York, NY, USA, 2005. ACM. 11

[Hein 2007] Mathias Hein. *Geometrical aspects of statistical learning theory.* PhD thesis, Technischen Universitaet Darmstadt, 2007. 26

[Hero 2000] A. Hero and O. Michel. *Estimation of Renyi Information Divergence via Pruned Minimal Spanning Trees.* Proceedings of the IEEE International Symposium on Information Theory, 2000. 11

[Hirani 2001] Anil N. Hirani, Jerrold E. Marsden and James Arvo. *Averaged Template Matching Equations.* In Proceedings of Energy Minimisation in Computer Vision and Pattern Recognition (EMMCVPR), volume 2134, pages 528 – 543. Springer-Verlag, 2001. 107

[Huguet 2007] Frederik Huguet and Frederic Devernay. *A Variational Method for Scene Flow Estimation from Stereo Sequences.* In ICCV, pages 1–7, 2007. 115

[Karcher 1977] H. Karcher. *Riemannian center of mass and mollifier smoothing.* Comm. Pure Appl. Math, no. 30, pages 509–541, 1977. 70, 83, 94, 95

[Kendall 1990] W. Kendall. *Probability, convexity, and harmonic maps with small image I: uniquness and fine existence.* Proc. London Math. Soc., no. 61, pages 371–406, 1990. 83

[Keriven 1998] R. Keriven and O. Faugeras. *Complete dense stereovision using level set methods.* In 5th European Conference on Computer Vision, 1998. 76, 115, 117, 121, 124, 126

[Kégl 2003] Balázs Kégl. *Intrinsic Dimension Estimation Using Packing Numbers*, 2003. 11

[Khamene 2007] Ali Khamene, Fred Azar, Loren Schwarz, Darko Zikic, Nassir Navab and Eike Rietzel. *A Unified and Efficient Approach for Free-form Deformable Registration.* IEEE International Conference on Computer Vision,, pages 1–8, 2007. 110

[Kilian 2007] Martin Kilian, Niloy J. Mitra and Helmut Pottmann. *Geometric Modeling in Shape Space.* ACM Trans. Graphics, vol. 26, no. 3, 2007. Proc. SIGGRAPH. 107

[Kimmel 1998] R. Kimmel and J. Sethian. *Computing geodesic paths on manifolds*, 1998. 123

[Kwok 2004] James T. Kwok and Ivor W. Tsang. *The Pre-Image Problem in Kernel Methods.* IEEE Transaction in Neural Network, vol. 15, no. 6, pages 1517–1525, 2004. 70, 84, 88, 89, 90, 91, 92, 93, 94, 100, 102, 137

[Lafon 2006a] S. Lafon and A. B. Lee. *Diffusion maps and coarse-graining: a unified framework for dimensionality reduction, graph partitioning, and data set parameterization.* Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 28, no. 9, pages 1393–1403, 2006. 59, 85

[Lafon 2006b] Stephane Lafon, Yosi Keller and Ronald R. Coifman. *Data Fusion and Multicue Data Matching by Diffusion Maps.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, no. 11, pages 1784–1797, 2006. 59, 87

[Lai 2009] Pei Ling Lai and Colin Fyfe. *Bregman Divergences and Multi-dimensional Scaling.* pages 935–942, 2009. 44

[Laurentini 1994] A. Laurentini. *The Visual Hull Concept for Silhouette-Based Image Understanding.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 16, no. 2, pages 150–162, 1994. 117

[Lazarus 1998] Francis Lazarus and Anne Verroust. *3D Metamorphosis: a Survey.* The Visual Computer, vol. 14, pages 8–9, 1998. 109

[Lazebnik 2001] S. Lazebnik, E. Boyer and J. Ponce. *On Computing Exact Visual Hulls of Solids Bounded by Smooth Surfaces.* In IEEE Conference on Computer Vision and Pattern Recognition, volume 1, pages 156–161, 2001. 117

[Leventon 2000] M. Leventon, E. Grimson and O. Faugeras. *Statistical Shape Influence in Geodesic Active Contours.* In IEEE Conference on Computer Vision and Pattern Recognition, pages 316–323, 2000. 94, 96, 99

[Lindenbaum 1994] M. Lindenbaum, M. Fischer and A. M. Bruckstein. *On Gabor Contribution To Image Enhancement.* Pattern Recognition, vol. 12, pages 1–8, 1994. 69

[Lipman 2009] Yaron Lipman and Thomas Funkhouser. *Mobius Voting for Surface Correspondence.* ACM Transactions on Graphics (Proc. SIGGRAPH), vol. 28, no. 3, August 2009. 111

[Litke 2005] Nathan Litke, Marc Droske, Martin Rumpf and Peter Schröder. *An image processing approach to surface matching.* In SGP '05: Proceedings of the third Eurographics symposium on Geometry processing, page 207, Aire-la-Ville, Switzerland, Switzerland, 2005. Eurographics Association. 110

[Lu 2008] Z. Lu, M. Carreira-Perpiñan and C. Sminchisescu. *People Tracking with the Laplacian Eigenmaps Latent Variable Model.* pages 1705–1712, 2008. 94

[Maier 2009] M. Maier, U. von Luxburg and M. Hein. *Influence of Graph Construction on Graph-based Clustering Measures.* Advances in Neural Information Processing Systems, vol. 22, pages 1025–1032, 2009. 45

[Matusik 2000] W. Matusik, C. Buehler, R. Raskar, S.J. Gortler and L. McMillan. *Image-Based Visual Hulls.* In ACM SIGGRAPH, pages 369–374, 2000. 117

[Michor 2006] Peter W. Michor and David Mumford. *Riemannian geometries on spaces of plane curves.* J. Eur. Math. Soc., vol. 8, pages 1–48, 2006. 67, 68

[Mika 1999] S. Mika, B. Schölkopf, A. J. Smola, K.-R. Müller, M. Scholz and G. Rätsch. *Kernel PCA and De–Noising in Feature Spaces.* In M. S. Kearns, S. A. Solla and D. A. Cohn, editeurs, Advances in Neural Information Processing Systems 11. MIT Press, 1999. 84, 87, 88, 89

[Mitra 2004] N. J. Mitra, N. Gelfand, H. Pottmann and L. Guibas. *Registration of Point Cloud Data from a Geometric Optimization Perspective.* In R. Scopigno and D. Zorin, editeurs, Eurographics Symposium on Geometry Processing, pages 23–32, 2004. 109

[Modersitzki 2004] Jan Modersitzki. Numerical methods for image registration. Oxford University Press, 2004. 110

[Moenning 2003] C. Moenning and N. A. Dodgson. *Fast marching farthest point sampling.* In EuroGraphics, Sept. 2003. 124

[Moeslund 1999] Thomas B. Moeslund and Erik Granum. *A Survey of Computer Vision-Based Human Motion Capture*, 1999. 8

[Osher 1988] S. Osher and J.A. Sethian. *Fronts Propagating with Curvature-Dependent Speed: Algorithms Based on Hamilton–Jacobi Formulations.* Journal of Computational Physics, vol. 79, no. 1, pages 12–49, 1988. 76

[Osher 2001] S. Osher and R.P. Fedkiw. *Level Set Methods: an Overview and Some Recent Results.* Journal of Computational Physics, vol. 169, no. 2, pages 463–502, 2001. 76, 78

[Osher 2003] Stanley Osher and Nikos Paragios, editeurs. Geometric level set methods in imaging, vision, and graphics. Springer, 2003. 76

[Paragios 2000] Nikos Paragios and Rachid Deriche. *Coupled geodesic active regions for image segmentation: A level set approach.* In Sixth European Conference on Computer Vision, volume 2, pages 224–240, 2000. 77

[Paragios 2002] Nikos Paragios and Rachid Deriche. *Geodesic active regions: A new framework to deal with frame partition problems in computer vision.* In Journal of Visual Communication and Image Representation (JVCIR), volume 13, pages 249–268, 2002. 77

[Pennec 1998] X. Pennec and N. Ayache. *A geometric algorithm to find scmall but highly similar 3D substructures in proteins.* vol. 14, pages 516–522, 1998. 114

[Pennec 2006] Xavier Pennec. *Intrinsic Statistics on Riemannian Manifolds: Basic Tools for Geometric Measurements.* Journal of Mathematical Imaging and Vision, vol. 25, no. 1, pages 127–154, July 2006. A preliminary appeared as INRIA RR-5093, January 2004. 94, 107, 111

[Perona 1990] P. Perona and J. Malik. *Scale-space and Edge Detection Using Anisotropic Diffusion.* IEEE Transaction on PAMI, vol. 12, no. 7, pages 629–639, 1990. 69

[Peyré 2003] Gabriel Peyré and Laurent Cohen. *Geodesic remeshing using front propagation.* In Int. J. Comput. Vision, pages 33–40, 2003. 149

[Planitz 2005] B. M. Planitz, A. J. Maeder and J. A. Williams. *The correspondence framework for 3D surface matching algorithms.* Comput. Vis. Image Underst., vol. 97, no. 3, pages 347–383, 2005. 109

[Pons 2007a] J.-P. Pons, R. Keriven and O. Faugeras. *Multi-view stereo reconstruction and scene flow estimation with a global image-based matching*

score. The International Journal of Computer Vision, vol. 72, no. 2, pages 179–193, Apr 2007. 115, 116, 117

[Pons 2007b] J.-P. Pons, R. Keriven and O. Faugeras. *Multi-view stereo reconstruction and scene flow estimation with a global image-based matching score*. The International Journal of Computer Vision, vol. 72, no. 2, pages 179–193, 2007. 126

[Pottmann 2006] H. Pottmann, Q. Huang, Y.-L. Yang and S.-M. Hu. *Geometry and convergence analysis of algorithms for registration of 3D shapes*. Int. J. Computer Vision, vol. 67, no. 3, pages 277–296, 2006. 114

[Reuter 2006] Martin Reuter. Laplace spectra for shape recognition. 2006. 34

[Rosenhahn 2008] Bodo Rosenhahn, Reinhard Klette and Dimitris Metaxas, editeurs. Human motion - understanding, modeling, capture and animation, volume 36 of *Computational Imaging and Vision*. Springer, Dordrecht, The Netherlands, 2008. 8

[Rousson 2002] M. Rousson and N. Paragios. *Shape Priors for Level Set Representations*. In European Conference on Computer Vision, volume 2, pages 78–92, 2002. 96

[Rousson 2008] M. Rousson and N. Paragios. *Prior Knowledge, Level Set Representations and Visual Grouping*. International Journal of Computer Vision, vol. 76, 2008. 103

[Roweis 2000a] S. Roweis and L. Saul. *Nonlinear Dimensionality Reduction by Locally Linear Embedding*. Science, vol. 290, pages 2323–2326, 2000. 11

[Roweis 2000b] S. T. Roweis and L. K. Saul. *Nonlinear dimensionality reduction by locally linear embedding*. Science, vol. 290, pages 2323–2326, 2000. 49

[Rueckert 1999] D. Rueckert, L. I. Sonoda, C. Hayes, D. L. G. Hill, M. O. Leach and D. J. Hawkes. *Nonrigid registration using free-form deformations: Application to breast MR images*. IEEE Transactions on Medical Imaging, vol. 18, pages 712–721, 1999. 110

[Schölkopf 1999] B. Schölkopf, A.-J. Smola and K.-R. Müller. *Kernel principal component analysis*. Advances in kernel methods: support vector learning, pages 327–352, 1999. 88

[Schölkopf 2002] Bernhard Schölkopf and Alexander J. Smola. Learning with kernels : support vector machines, regularization, optimization, and beyond. Adaptive computation and machine learning. MIT Press, 2002. 71

[Schreiner 2004] John Schreiner, Arul Asirvatham, Emil Praun and Hugues Hoppe. *Inter-surface mapping.* ACM Trans. Graph., vol. 23, no. 3, pages 870–877, 2004. 110

[Serra 1998] Jean Serra. *Hausdorff distances and interpolations.* In In International Symposiumon Mathematical Morphology and its Applications to Image and Signal Processing, pages 107–114, 1998. 79, 96

[Sethian 1999a] J. A. Sethian. *Fast marching methods.* SIAM Review, vol. 41, pages 199–235, 1999. 123

[Sethian 1999b] J. A. Sethian. Level set methods and fast marching methods. Cambridge University Press, 1999. 70

[Sethian 1999c] J.A. Sethian. Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Sciences. Cambridge Monograph on Applied and Computational Mathematics. Cambridge University Press, 1999. 78

[Shilane 2004] Philip Shilane, Patrick Min, Michael Kazhdan and Thomas Funkhouser. *The Princeton Shape Benchmark.* In Shape Modeling and Applications, pages 167–178, 2004. 107

[Silva 2003a] Vin De Silva and Joshua B. Tenenbaum. *Global Versus Local Methods in Nonlinear Dimensionality Reduction.* In Advances in Neural Information Processing Systems 15, pages 705–712. MIT Press, 2003. 48

[Silva 2003b] Vin De Silva and Joshua B. Tenenbaum. *Improving support vector machine classiers by modifying kernel functions.* In Advances in Neural Information Processing Systems 15, pages 705–712. MIT Press, 2003. 49

[Smola 1998] A. J. Smola. *Learning with Kernels.* PhD thesis, Technische Universität Berlin, 1998. 71, 75

[Solem 2006] J.E. Solem. *Geodesic Curves for Analysis of Continuous Implicit Shapes.* In International Conference on Pattern Recognition, volume 1, pages 43–46, 2006. 96

[Sonka 1998] M. Sonka, V. Hlavac and R. Boyle. Image processing, analysis, and machine vision. Chapman & Hall, 2 édition, 1998. 6

[Starck 2005] J. Starck and A. Hilton. *Spherical Matching for Temporal Correspondence of Non-Rigid Surfaces.* In ICCV, pages 1387–1394, 2005. 117

[Starck 2007a] J. Starck and A. Hilton. *Correspondence labelling for wide-timeframe free-form surface matching.* IEEE International Conference on Computer Vision (ICCV), 2007. 117

[Starck 2007b] J. Starck and A. Hilton. *Surface Capture for Performance Based Animation.* IEEE Computer Graphics and Applications, vol. 27(3), pages 21–31, 2007. 125, 127, 138

[Sundaramoorthi 2008] Ganesh Sundaramoorthi, Anthony J. Yezzi, Andrea Mennucci and Guillermo Sapiro. *New Possibilities With Sobolev Active Contours.* International Journal of Computer Vision, 2008. 80, 81

[Tenenbaum 2000] J. B. Tenenbaum, V. de Silva and J. C. Langford. *A global geometric framework for nonlinear dimensionality reduction.* Science, vol. 290, no. 5500, pages 2319–2323, December 2000. 11, 47

[Terzopoulos 1987] Demetri Terzopoulos, John Platt, Alan Barr and Kurt Fleischer. *Elastically deformable models.* In SIGGRAPH '87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques, pages 205–214, New York, NY, USA, 1987. ACM. 110

[Thorstensen 2008] Nicolas Thorstensen, Florent Segonne and Renaud Keriven. *Normalization and Preimage Problem in Gaussian Kernel PCA.* International Conference on Image processing (ICIP), 2008. 15, 167

[Thorstensen 2009a] Nicolas Thorstensen, Patrick Etyngier, Florent Segonne and Renaud Keriven. *Diffusion Maps as a framework for shape modeling.* Computer Vision and Image Understanding, vol. submitted, 2009. 167

[Thorstensen 2009b] Nicolas Thorstensen and Renaud Keriven. *Non-rigid shape matching using Geometry and Photometry.* Asian Conference on Computer Vision (ACCV)., 2009. 16, 111, 167

[Thorstensen 2009c] Nicolas Thorstensen, Florent Segonne and Renaud Keriven. *Pre-Image as Karcher Mean using Diffusion Maps: Application to shape and image denoising.* Scale Space and Variational Methods (SSVM)., 2009. 15, 167

[Tomasi 1998] C. Tomasi and R. Manduchi. *Bilateral Filtering for Gray and Color Images.* In ICCV '98: Proceedings of the Sixth International Conference on Computer Vision. IEEE Computer Society, 1998. 6

[Trouvé 2005] Alain Trouvé and Laurent Younes. *Metamorphoses through lie group action.* In Foundations of Computational Mathematics, volume 5, pages 173–198, 2005. 68, 78, 107

[Varanasi 2008] K. Varanasi, A. Zaharescu, E. Boyer and R. Horaud. *Temporal Surface Tracking Using Mesh Evolution.* In Proceedings of the Tenth European Conference on Computer Vision, volume Part II of *LNCS*, pages 30–43, October 2008. 117

[Vedula 2005] S. Vedula S.and Baker, P. Rander, R. Collins and T. Kanade. *Three-Dimensional Scene Flow*, 2005. 115

[Veltkamp] Remco C. Veltkamp and Michiel Hagedoorn. *Shape Similarity Measures, Properties, and Constructions.* 109

[Vlasic 2009] Daniel Vlasic, Pieter Peers, Ilya Baran, Paul Debevec, Jovan Popović, Szymon Rusinkiewicz and Wojciech Matusik. *Dynamic Shape Capture using Multi-View Photometric Stereo.* ACM Trans. Graphics (Proc. SIGGRAPH Asia), vol. 28, no. 5, December 2009. 117

[Vu 2009] H. Vu, R. Keriven, P. Labatut and J.P. Pons. *Towards high-resolution large-scale multi-view stereo.* Conference on Computer Vision and Pattern Recognition(CVPR), Jun 2009. 117, 128

[Younes 1998] Laurent Younes. *Computable Elastic Distances Between Shapes.* SIAM Journal of Applied Mathematics, vol. 58, pages 565 – 586, 1998. 107

[Younes 2000] Laurent Younes. *Deformations, Warping and Object Comparison: A Tutorial*, 2000. 107

[Zhang 1994] Z. Zhang. *Iterative Point Matching for Registration of Free-Form Curves and Surfaces.* Int. Joun. Comp. Vis., vol. 13, no. 2, pages 119–152, 1994. 110, 112

[Zöckler 2000] Malte Zöckler, Detlev Stalling and Hans-Christian Hege. *Fast and Intuitive Generation of Geometric Shape Transitions*. The Visual Computer, vol. 16, no. 5, pages 241–253, 2000. 110

Publications of the author

- Normalization and Preimage Problem in Gaussian Kernel PCA N. Thorstensen, F. Ségonne and R. Keriven ICIP(2008), Conf. ICIP 2008[Thorstensen 2008]

- Pre-Image as Karcher Mean using Diffusion Maps: Application to shape and image denoising N. Thorstensen, F. Ségonne and R. Keriven SSVM(2009), Proc. SSVM 09 Voss , Norway[Thorstensen 2009c]

- Non-rigid shape matching using Geometry and Photometry N. Thorstensen and R. Keriven ACCV(2009), Proc. ACCV 09, Xi'An, China[Thorstensen 2009b]

- Diffusion Maps as a framework for Shape Modeling. Thorstensen Nicolas and Etyngier Patrick and Ségonne Florent and Keriven Renaud in Computer Vision and Image Understanding submitted [Thorstensen 2009a]