Apprentissage statistique, variétés de formes et applications à la segmentation d'images

Patrick Etyngier

, April 23, 2008 - Get an update. http://certis.enpc.fr/∼etyngier/phd/dissertation

Thesis

Submitted in partial fulfillment of the requirements for the degree of

Docteur de l'École Nationale des Ponts et Chaussées (ENPC). Spécialité *Mathématiques, Informatique*
(Doctor of Philosophy in *Mathematics, Computer Science* at ENPC)

Statistical learning, Shape Manifolds & Applications to Image Segmentation

by

Patrick ETYNGIER

Oral defense: January 21, 2008

PhD Committee:

Nicholas	AYACHE	President
Daniel	CREMERS	Reviewer
Rachid	DERICHE	
Françoise	DIBOS	Reviewer
Renaud	KERIVEN	PhD advisor
Bertrand	THIRION	

within the Odyssée Team

ENPC **INRIA** **ENS**

ÉCOLE NATIONALE DES PONTS ET CHAUSSÉES

,

îledeFrance

Thèse

Présentée pour l'obtention du grade de *Docteur de l'École Nationale des Ponts et Chaussées*

Spécialité: Mathématiques, Informatique

Apprentissage Statistique, Variétés de Formes & Applications à la segmentation d'images

Par

Patrick ETYNGIER

Soutenue le Lundi 21 Janvier 2008

Devant le jury composé de:

Nicolas	AYACHE	Président
Daniel	CREMERS	Rapporteur
Rachid	DERICHE	Examinateur
Françoise	DIBOS	Rapporteur
Renaud	KERIVEN	Directeur de thèse
Bertrand	THIRION	Examinateur

au sein de l'Équipe Odyssée

ENPC **INRIA** **ENS**

ÉCOLE NATIONALE DES PONTS ET CHAUSSÉES

INRIA

LETTRES SCIENCES ÉCOLE NORMALE SUPÉRIEURE

Abstract

Image segmentation with shape priors has received a lot of attention over the past few years. Most existing work focuses on a linearized shape space with small deformation modes around a mean shape, which is only relevant when considering similar shapes. In this thesis, we introduce a new framework that can handle more general shape priors.

We model a category of shapes as a finite dimensional manifold, the shape prior manifold, which we analyze from the shape samples using dimensionality reduction techniques such as diffusion maps. An embedding function is then learned from the manifold. Unfortunately, this model does not provide an explicit projection operator onto the underlying shape manifold, and therefore, our work tackles this problem.

Our solution is threefold.

First, we propose different solutions to the out-of-sample problem and define three attracting forces directed towards the manifold. These forces can be used as projection operators onto the manifold:

- Projection towards the closest point
- Projection with the same embedding
- Projection at constant embedding

Next, we introduce a shape prior term for the active contours/regions framework through a non-linear energy term designed to attract shapes towards the manifold.

Finally, we describe a variational framework for manifold denoising.

Results with real objects such as car silhouettes or anatomical structures show the potential of our method.

Resumé

La segmentation d'image avec a priori de forme a fait l'objet d'une attention particulière ces dernières années. La plupart des travaux existants reposent sur des espaces de formes linéarisés avec de petits modes de déformations autour d'une forme moyenne. Cette approche n'est pertinente que lorsque les formes sont relativement similaires. Dans cette thèse, nous introduisons un nouveau cadre dans lequel il est possible de manipuler des a priori de formes plus généraux.

Nous modélisons une catégorie de formes comme une variété de dimension finie, la variété des formes a priori, que nous analysons à l'aide d'échantillons de formes en utilisant des techniques de réduction de dimension telles que les diffusion maps. Un plongement dans un espace réduit est alors appris à partir des échantillons. Cependant, ce modèle ne fournit pas d'opérateur de projection explicite sur la variété sous-jacente et nous nous attaquons à ce problème.

Les contributions de ce travail se divisent en trois parties.

Tout d'abord, nous proposons différentes solutions au problème des "out-of-sample" et nous définissons trois forces attirantes dirigées vers la variété.

- Projection vers le point le plus proche;
- Projection ayant la même valeur de plongement;
- Projection à valeur de plongement constant.

Ensuite, nous introduisons un terme d'a-priori de formes pour les coutours/régions actifs/ves. Un terme d'énergie non-linéaire est alors construit pour attirer les formes vers la variété.

Enfin, nous décrivons un cadre variationnel pour le debruitage de variété.

Des résultats sur des objets réels tels que des silhouettes de voitures ou des structures anatomiques montrent les possibilités de notre méthode.

Remerciements / Acknowledgment (French)

À mon directeur de thèse, Renaud Keriven

J'adresse toute ma reconnaissance à mon directeur de thèse, Renaud Keriven. Il a su en toutes circonstances, même celles qui me semblaient les plus désespérées, créer un climat de confiance et de sérénité qui m'a été particulièrement favorable. Je mesure la chance d'avoir été son étudiant et l'importance de l'expérience scientifique et humaine qu'il a partagée avec beaucoup de patience.

Aux chercheurs du CERTIS

Je remercie tous les chercheurs du CERTIS pour m'avoir éclairé de leur savoir au long des nombreux échanges et collaborations scientifiques. Je pense en particulier à Florent Ségonne, à Jean-Yves Audibert, à Jean-Philippe Pons, et à Hichem Sahbi.

Je remercie également Nikos Paragios pour l'encadrement de qualité dont j'ai bénéficié lorsqu'il était à l'École des Ponts. Je n'oublie pas Yakup Genc avec qui j'ai collaboré à Siemens Corporate Research à Princeton, NJ aux Etats-Unis.

Aux membres du jury

Je suis conscient de l'honneur que m'ont fait les membres du jury par leur présence le jour de ma soutenance de thèse. En particulier, je remercie Daniel Cremers et Françoise Dibos, rapporteurs de ce manuscrit, Nicholas Ayache, président du jury, ainsi que Rachid Deriche et Bertrand Thirion, examinateurs.

Aux institutions

Je remercie les institutions qui ont organisé la logistique autour de cette thèse et qui ont assuré son bon déroulement. Il s'agit évidemment de l'École des Ponts - ParisTech mais aussi de la région Ile de France pour son soutien financier.
J'en profite pour remercier tous les contribuables qui ont indirectement participé au financement de ces 3 années de recherche.

A tous les membres du certis

Je remercie également tous les membres du CERTIS qui ont tous contribué à une bonne ambiance de travail, propice à la réussite.

A ma famille

Je profite de ces quelques lignes pour exprimer toute ma gratitude à mes parents, Jacques et Myriam, qui m'ont soutenu et encouragé durant toutes mes années d'études quels que soient les choix que j'ai pu faire. Je n'oublie pas mes frères et ma belle soeur, Philippe, Pascal et Jessica, ma grand-mère, Mimi, et mon oncle, Albert qui ont eu la patience d'être à mes côtés même durant mes moments insupportables.

Aux amis

Je garde aussi un peu d'espace pour mes amis qui m'ont encouragé et écouté pendant des heures et des heures et qui m'ont fait l'honneur d'être présent le jour de la soutenance.

Contents

List of Figures

Chapter 1

Introduction

Contents

1.1 Context

1.1.1 On the image side

Image analysis has enjoyed an increasing demand from the medical imaging field for a few years, since computer aided diagnostics rely mainly on *image processing* and *computer vision* techniques. Furthermore, images have now a predominant place in our daily life since camera devices appear everywhere. Cell phones all have one or two cameras embedded and the smallness constraint implies lower quality of optics and sensors, which is offset by an extensive use of image processing (image enhancements such as noise removal, automatic contrast balance etc.). Regular digital cameras have even real time face detectors to optimize the auto focus mode. Surveillance cameras in buildings, cities and highways have been considerably growing in number for the last two decades. In view of the huge

amount of video sequences produced, computer vision techniques can make the surveillance operations [semi-]automatic instead of costly numerous surveillance operators.

Image processing and *computer vision* are the sciences that analyze and extract useful information from digital images and videos. An *image* is basically a n-dimensional array (in general, $n = 2$ or possibly $n = 3$. n is the dimension of the image domain) of which the values are quantified within a given range. Such values may be of dimension 1 (black and white images), 3 (color images) or even higher. *Image processing* is related to low-level vision tasks such as edge and contour extraction, noise removal, deblurring etc; while *computer vision* is related to higher level tasks such as object tracking in video, event detection in video, 3D reconstruction from 2D images, shape recognition etc. The latter relies of course on the former.

Researchers and manufacturers have been developing these techniques since the emergence of computers in the seventies. At that time, the techniques proposed were mainly low level based. Nevertheless, the core idea of image analysis has remained the same since its beginnings. Algorithms mostly attempt to partition images into components or areas of interest in order to do some kind of high level "intelligent process". For example, *edge detection* algorithms partition an image into edge pixels and non edge pixels, *shape recognition* extracts an area of the image domain within a given range of shapes, *object tracking* does the same in video sequences etc. All these processes can be called *image segmentation* in the broad sense of the word. In this thesis, we will be focusing on application on *active contour segmentation* — i.e. *extraction of objects in images by detecting their closed contour* — and unless specified, we will implicitly refer to that kind of segmentation.

1.1.2 On the statistical learning side

As the number of computers over the planet and their storage capacity have considerably increased, they produce more and more data, which enable the development of huge database. Management, organization and understanding of such amounts of data cause many challenging problems to tackle. Statistical approaches are commonly used when dealing with such databases, particularly in the fields of

classification and *dimensionality reduction*.

Classification is the science that classifies data into two or more clusters. First, it extracts features from data with some invariant characteristics. A classification function is then learned from a set of training samples and its capacity to generalize to new samples is assessed using a test set (of course with distinct samples from the training set). An efficient learning algorithm for classification should minimize two different errors called *learning error* and *generalization error*. The former is the error made on the training set while the latter is the error made on the test set. The more complex the function is, the less the learning error is, but the more the generalization error might be. A trade-off between the two errors should then be determined.

Dimensionality reduction techniques deal with high dimensional data and aim to represent them in a lower dimensional space to have a better understanding of their intrinsic characteristics, ideally to recover their intrinsic parametrization. They typically build a mapping from the original space into the lower dimensional space. The premises date back to the beginning of the twentieth century when Kenneth Pearson introduced the Principal Component Analysis technique (PCA) [151] in 1901. PCA assumes that data lie on a linear subspace under a Gaussian distribution. Multi-Dimensional Scaling (MDS) is another linear technique related to PCA, based on pair-wise distance in the dataset [216, 201, 51]. For the last decade, dimensionality reduction has received interest because many problems cannot be modeled using linear models. A non linear version of PCA was proposed in [180] by using kernel methods. In addition, a series of recent techniques relies explicitly on the assumption that data lie on a (non linear) smooth manifold. These approaches, which are particularly successful because of their well-posedness, are also known as *manifold learning* techniques. Among the most popular techniques are Locally Linear Embedding (LLE) [166], Laplacian Eigenmaps [99] and Diffusion maps [117]. These methods are able to output a low dimensional global representation of high dimensional data lying on a non linear manifold by preserving the local neighborhood information, and can be possibly extended to new data.

In both cases, the learning function should be estimated from a training set of limited size and should have good generalization capacities.

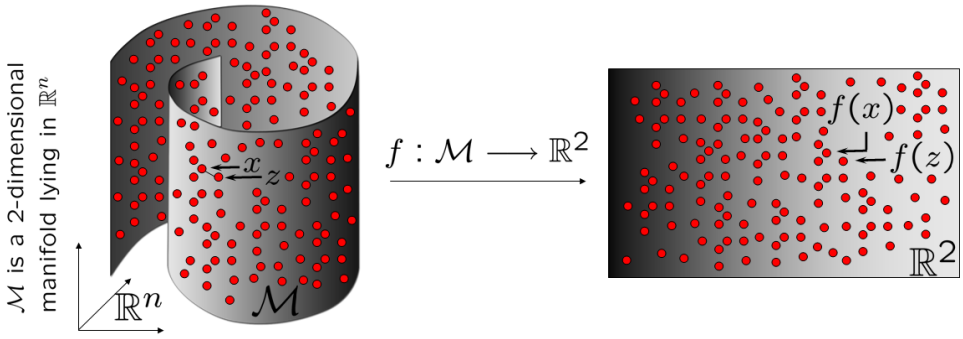

Figure 1.1: **Non linear manifold learning** techniques build an embedding function f that represents the data in a lower dimensional space. Red points sample a non linear manifold. Linear models such as PCA or MDS cannot be applied to such non linear datasets as explained in chapter 5.

1.2 Image segmentation and statistical learning

The goal of image segmentation is basically to partition the image domain into N disjoint regions that usually correspond to part or complete objects in the scene. In earlier approaches, image segmentation methods delineated objects by using only their edges, which are identified by higher gradient values in the target image. Regions of the image domain can also, for instance, be guessed according to given pixel grouping constraints (uniformity, statistical properties etc., within the regions).

In this work, we are limited to two regions separated by a boundary contour. Without loss of generality, many approaches define a curve denoted S which minimizes a given energy functional. For example, *graph cuts* find the curve that has the global minimum of the energy. In this thesis, we focus on gradient descent approaches in order to minimize the target energy functional: the curve S evolves on the image domain according to the minimization of a variational energy E. An energy E_{data} is usually designed to attract the curve S toward the edges of the object to segment. E_{data} can also, for example, be designed regarding the statistical properties inside and outside the curve S. See section (4.3.3) for region-based active contour segmentation.

$$E_1(S) = E_{\text{data}}(S) \tag{1.1}$$

Edges may be badly defined due to blur, noise, or occlusion and a smoothness constraint E_{smooth} along the curve S must be added to equation (2.1) to overcome such difficulties. Nevertheless, it is not sufficient and an *a-priori* knowledge on the possible shapes must also be introduced. Statistical learning is extensively used in computer vision in order to introduce an *a-priori* knowledge, particularly in image segmentation processes. In this work, we focus on *shape priors* learned from a given training set. Segmentation with shape priors is expressed using a model similar to equation (2.1), but new energies $E_{\text{a-priori}}$ and E_{smooth} are added. $E_{\text{a-priori}}$, built from the training set, constrains the curve within a given range of shapes, while E_{smooth} constrains the curve to be smooth.

$$E_2(S) = E_{\text{data}}(S) + \alpha E_{\text{smooth}}(S) + \beta E_{\text{a-priori}}(S) \tag{1.2}$$

where α and β are parameters set to quantify the influence of the *a-priori* knowledge and the smoothness of the curve in the final result. To our best knowledge, the work by Daniel Cremers, Cristoph Schnörr, Joachim Weickert and Christian Schellewald on Diffusion Snakes [58, 57] was the first to combine a generic segmentation method with an added energy.

1.3 Goal and organization of this dissertation

There are many attempts in literature to embed *a-priori* knowledge in segmentation tasks notably by using dimensionality reduction techniques, as in the work of Michael Leventon, Eric Grimson and Olivier Faugeras in [122] but also in [165, 42, 203, 40] to cite a few. These approaches assume that data lie on a linear subspace under a Gaussian distribution.

In this work, we want to depart from such assumptions in order to handle more general non linear shape priors. We model a category of shape — e.g. *the category of fish shapes* — as a smooth non linear finite dimensional manifold that we call the *shape prior manifold*. In this context, our view is that the *shape priors* is a force directed toward the *shape prior manifold*s. An important related work was proposed by Daniel Cremers, Timo Kohlberger and Christoph Schnörr in [53]. The authors introduce non linear shape priors by using a probabilistic version of Kernel PCA (KPCA). Note that we aim to handle more general cases.

Although we are driven by applications for image segmentation with shape priors in this work, we will endeavor to present our contribution in a generic form. The purpose of this dissertation is to extend cutting edge manifold learning techniques such as diffusion maps to design a force that attracts data toward the manifold. The force should be designed to be usable in the context of image segmentation. An overview of the concept is presented in figure (2.2).

This thesis is organized as follow.

Chapter 3: Shapes

This chapter provides the reader with all the necessary background about shapes. In particular, we define the concept of shape and detail some of their possible representations. We also introduce the notion of distance between shapes. Finally, we explain how to interpolate between shapes by using for instance Karcher means [107] and how to model a *shape manifold* from a finite shape set.

Chapter 4: Image segmentation

In this chapter, we briefly review the techniques of image segmentation in the broad sense of the word. Since we are driven by applications in image segmentation, we give the context to show how our approach crosses over a new step in the state of the art.

Chapter 5: Dimensionality reduction and non linear manifold learning

In our work, we aim to learn *shape priors* from a category of shapes lying on a non-linear smooth manifold. Chapter 5 focuses on the statistical learning side and on dimensionality reduction techniques, which define an embedding into the reduced space. This chapter is organized to emphasize the needs and the importance of non linear models. We present and illustrate the most popular methods from basic linear models such as PCA [151] or MDS [201, 202, 51] to cutting edge non linear manifold learning techniques such as Laplacian eigenmaps [15] and diffusion maps [117].

Chapter 6 - Application I: Interactive image retrieval

Since our work relies mainly on manifold learning techniques, we focus on an application of graph Laplacian/diffusion maps to interactive image retrieval based on *relevance feedback* techniques. In these approaches, the system attempts to build a metric and/or a decision rule that reflects the user's intention based on not only the data he labeled but also the others (usually the user does a binary labeling). Indeed, data are supposed to lie on a smooth manifold and diffusion maps are used to learn this manifold and diffuse the labels (this process is known as transductive learning). An interaction between the system and the users is then created in order to refine the response by proposing new unlabeled data to the user. Finally, the system outputs a class of images supposed to reply to a query defined by the user's mind.

Publications related to this chapter

Hichem Sahbi, Patrick Etyngier, and Jean-Yves Audibert and Renaud Keriven. Graph Laplacian for Interactive Image Retrieval. *ICASSP' 08: In proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, Las Vegas, April 2008.

Hichem Sahbi, Patrick Etyngier, and Jean-Yves Audibert and Renaud Keriven. Manifold Learning using Robust Graph Laplacian for Interactive Image Retrieval. *CVPR' 08: In proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, Anchorage, Alaska, June 2008.

Chapter 7: Extensions and *attracting forces*

This chapter contains the main contributions of this dissertation. Our approach relies on a manifold learning techniques called diffusion maps, that builds an embedding function into a low dimensional representation of data. This mapping is defined only on the training samples. Chapter 7 starts by pointing out extensions of the embedding to any new point — i.e. *outside the training set* — that comes into the process. Such extensions avoid recalculating embedding values from scratch.

Our goal is to design a force applied to data, directed toward the learned mani-

fold. A natural idea is to design a projection operator onto the manifold in order to obtain an attracting force.

We propose three different *attracting forces* in this chapter. The first one is based on a projection minimizing the distance to the manifold. Next, the second force relies on a projection having the same embedding value. We finally define a third *attracting force* toward the manifold, which preserves the embedding constant. This latter approach, which relies on diffusion maps [117] and the Nyström extension [145], is well posed and natural.

Again, since the tools developed in this work can be applied to any kind of data lying in a space equipped with a differentiable distance, we endeavor to present this chapter in a generic form.

Publications related to this chapter

Patrick Etyngier, Renaud Keriven, and Jean-Philippe Pons. Towards segmentation based on a shape prior manifold *SSVM' 07: In proceedings of the 1st International Conference on Scale Space and Variational Methods in Computer Vision* Ishia, Italy, May 2007.

Patrick Etyngier, Renaud Keriven and Florent Ségonne. Projection Onto a Shape Manifold for Image Segmentation with Prior. *ICIP' 07: In proceedings of the 14th IEEE International Conference on Image Processing*, San Antonio, Texas, USA, September 2007.

Patrick Etyngier, Renaud Keriven and Florent Ségonne. Shape priors using Manifold Learning Techniques. *ICCV' 07: In proceedings of the 11th IEEE International Conference on Computer Vision*, Rio de Janeiro, Brazil, October, 2007.

Patrick Etyngier, Renaud Keriven and Florent Ségonne. Active-Contour-Based Image Segmentation using Machine Learning Techniques. *MICCAI' 07: In proceedings of the 10th International Conference on Medical Image Computing and Computer Assisted Intervention*, Brisbane, Australia, October, 2007

Chapter 8 - Application II: Manifold denoising and image segmentation with general non linear shape priors

The tools developed in chapters 5 and 7 of this dissertation are successfully applied to manifold denoising and image segmentation with non linear shape priors. In particular, chapter 8 illustrates results obtained on various 2D and 3D examples corresponding to different *shape manifolds*: fishes, cars and anatomical structures.

Publications related to this chapter

Patrick Etyngier, Renaud Keriven, and Jean-Philippe Pons. Towards segmentation based on a shape prior manifold *SSVM' 07: In proceedings of the 1st International Conference on Scale Space and Variational Methods in Computer Vision* Ishia, Italy, May 2007.

Patrick Etyngier, Renaud Keriven and Florent Ségonne. Projection Onto a Shape Manifold for Image Segmentation with Prior. *ICIP' 07: In proceedings of the 14th IEEE International Conference on Image Processing*, San Antonio, Texas, USA, September 2007

Patrick Etyngier, Renaud Keriven and Florent Ségonne. Shape priors using Manifold Learning Techniques. *ICCV' 07: In proceedings of the 11th IEEE International Conference on Computer Vision*, Rio de Janeiro, Brazil, October, 2007

Patrick Etyngier, Renaud Keriven and Florent Ségonne. Active-Contour-Based Image Segmentation using Machine Learning Techniques. *MICCAI' 07: In proceedings of the 10th International Conference on Medical Image Computing and Computer Assisted Intervention*, Brisbane, Australia, October, 2007.

Appendix A: Radon/Hough space for pose estimation

In this appendix, we present a learning based approach to one shot camera calibration. First, features based on the Radon/Hough transform are learned while 3D reconstruction of a given indoor scene is achieved. An inference step then determines the position of a one-shot image from its features and the 3D reconstruction.

Publication related to this chapter

Patrick Etyngier, Nikos Paragios, Renaud Keriven, Yakup Genc and Jean-Yves Audibert. Radon space and Adaboost for Pose Estimation. *ICPR' 06: In proceedings of the 18th International Conference on Pattern Recognition* Hong-Kong, August 2006.

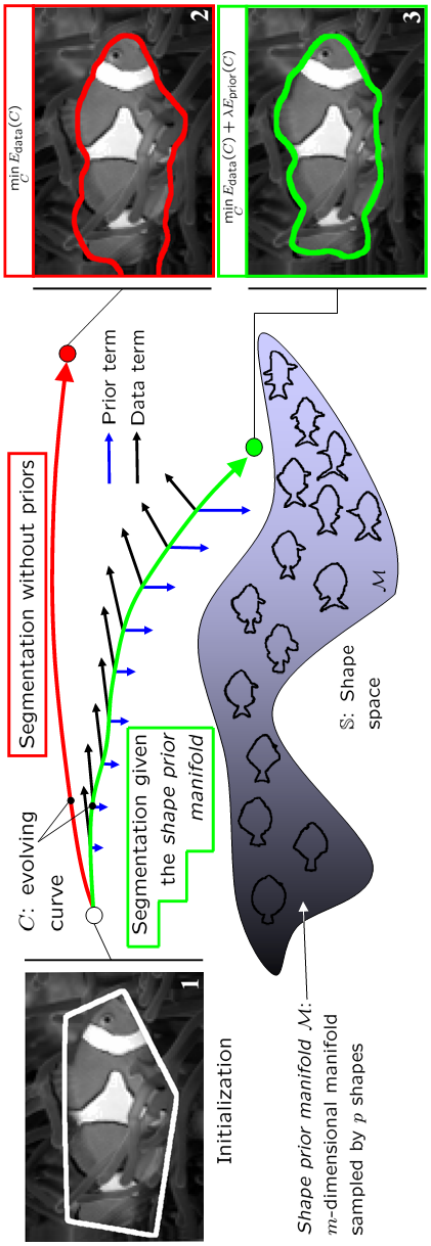

Figure 1.2: **Overview and goal of this dissertation**: a category of shapes — e.g. *the fish shapes* — is represented as a finite dimensional manifold. The *shape prior* term is a force directed toward the manifold and combined with a data term force

Chapitre 2

Introduction (Version française)

Contents

2.1 Contexte

2.1.1 L'image comme point de départ

Depuis quelques années, l'analyse d'image est de plus en plus sollicitée par les applications d'imagerie médicale, puisque les diagnostics médicaux assistés par ordinateur reposent principalement sur les techniques de traitement d'image et de vision par ordinateur.
Par ailleurs, les images numériques ont pris une place prédominante dans notre vie quotidienne où les caméras vidéo et appareils photos deviennent omniprésents. Les téléphones portables intègrent tous une voire deux capteurs. La contrainte de taille entraîne une qualité amoindrie des optiques et des capteurs, compensée par l'utilisation intensive de traitement d'image (amélioration de l'image tel que la réduction du bruit, réglage automatique du contraste etc.). Désormais, même les

appareils photos non professionnels ont des détecteurs de visage en temps réel pour optimiser l'*autofocus*.

Aussi, le nombre de caméras de surveillance a explosé depuis les vingt dernières années dans les bâtiments publics et privés, dans les villes, sur les autoroutes etc... Face à la quantité de séquences vidéo produite quotidiennement, les techniques de vision par ordinateurs peuvent rendre les opérations de surveillance beaucoup moins coûteuses, en réduisant par exemple le nombre d'opérateurs nécessaires grâce à l'automatisation des processus.

Le *traitement d'image* et la *vision par ordinateur* sont les sciences qui analysent et extraient des informations utiles des images et vidéo numériques. Une *image* est un tableau de dimension n (en général, $n = 2$ ou parfois $n = 3$. n est la dimension du domaine de l'image), dont les valeurs sont quantifiées dans un intervalle borné. Ces valeurs peuvent être de dimension 1 (images en "noir et blanc"), 3 (image en "couleur") ou parfois de dimension supérieure.

Le *traitement d'image* est un ensemble de techniques qui réalisent des tâches de vision bas niveau tels que l'extraction de bords et de contours, la réduction de bruit, le défloutage etc, tandis que la *vision par ordinateur* est un ensemble de techniques qui réalisent des tâches de vision de niveau plus élevé, telles que le suivi d'objet et la détection d'événements dans les vidéo, la reconstruction 3D à partir d'images 2D, la reconnaissance de formes etc.... La *vision par ordinateur* repose naturellement sur les techniques de *traitement d'images*.

Les chercheurs et industriels ont développé ces techniques depuis l'émergence des ordinateurs dans les années 70. A cette époque, les techniques étaient principalement de bas niveau. Cependant, l'idée de base de l'analyse d'image est restée inchangée depuis ses débuts. Les algorithmes tentent, pour la plupart, de morceler le domaine de l'image en composantes ou régions d'intérêt, afin de réaliser des "processus intelligents" de haut niveau. Par exemple, les algorithmes de détection des bords séparent les pixels appartenant aux bords et les autres, les algorithmes de reconnaissance de forme extraient une région de l'image contrainte dans un ensemble donné de formes, le suivi d'objet a les mêmes objectifs dans les séquences vidéo etc. Tous ces processus sont appelés *segmentation d'image* au sens large du terme. Dans cette thèse, nous nous concentrerons sur la segmentation par contours actifs — i.e. *l'extraction d'objets dans les images en utilisant leur contour* — et sauf mention contraire, nous ferons implicitement référence à ce type de segmen-

tation.

2.1.2 L'apprentissage statistique comme point de départ

Alors que le nombre d'ordinateurs sur la planète et leur capacité de stockage ont considérablement augmenté, de plus en plus de données nourrissent des *bases de données* énormes. La gestion, l'organisation et la compréhension de telles quantités de données suscitent des nouveaux défis à relever, particulièrement dans les domaines de la *classification* et de la *réduction de dimension*.

La *classification* est la science qui classe les données en deux (ou plus) ensembles. D'abord, des caractéristiques invariantes sont extraites des données. Une fonction de classification est alors apprise à partir d'un ensemble d'apprentissage et sa capacité à généraliser à de nouveaux échantillons est évaluée au moyen d'un ensemble test (dont les échantillons sont évidemment distincts de ceux contenus dans l'ensemble d'apprentissage). Un algorithme d'apprentissage efficace pour la classification doit minimiser deux erreurs appelées *erreur d'apprentissage* et *erreur de généralisation*. L'*erreur d'apprentissage* est l'erreur faite sur l'ensemble d'apprentissage tandis que l'*erreur de généralisation* est l'erreur faite sur l'ensemble de test. Plus la fonction de classification est complexe, plus l'erreur d'apprentissage est petite, mais plus l'erreur de généralisation est importante. Un compromis entre les deux erreurs doit donc être trouvé.

Les techniques de *réduction de dimension* ont pour objectif de représenter des données de grande dimension dans un espace de plus petite dimension, afin de d'en extraire des caractéristiques intrinsèques, idéalement pour retrouver leur paramétrisation intrinsèque. Pour cela, il faut construire une application de l'espace original des données vers un espace de plus petite dimension. Les prémisses remontent au début du vingtième siècle avec l'analyse en composante principales (ACP) introduite par Kenneth Pearson en 1901 [151]. L'utilisation de l'ACP suppose que les données analysées sont sur un sous-espace linéaire selon une distribution gaussienne. Il existe également une autre approche linéaire, sous le nom de Multi-Dimensional Scaling (MDS), proche de l'analyse en composante principale mais qui construit une application dans un espace de plus petite dimension à partir des distances entre les points uniquement [216, 201, 51]. Depuis une dizaine d'années, la *réduction de dimension* a connu un regain d'intêret car beaucoup de

problèmes ne peuvent être résolus à l'aide des méthodes linéaires. Une version non linéaire de l'analyse en composante principale a été proposée en utilisant les méthodes à noyaux : l'analyse en composante principale à noyau [180]. Par ailleurs, une série de techniques récentes reposent explicitement sur l'hypothèse que les données échantillonnent une variété différentiable (non linéaire). Parmi ces méthodes, on trouve : Linear Embedding (LLE) [166], Laplacian Eigenmaps [99] et Diffusion maps [117]. Ces méthodes sont capables de représenter globalement des données de grande dimension échantillonnant une variété différentielle, en préservant uniquement l'information locale de voisinage. L'extension aux nouveaux points est possible.

Dans les deux cas, la fonction d'apprentissage doit être estimée à partir d'un ensemble de points (ensemble d'apprentissage) de taille limitée et doit avoir de bonnes capacités de généralisation.

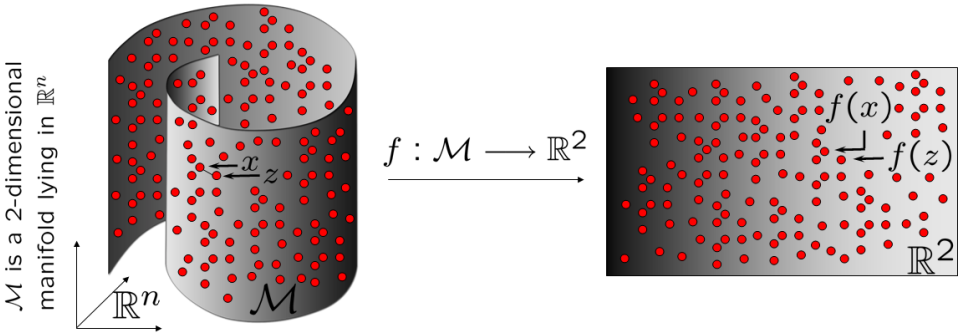

FIG. 2.1 – Les techniques d'**apprentissage non linéaire de variété** construisent un plongement f qui représente les données dans un espace de plus petite dimension. Sur la figure ci-dessus, les points rouges échantillonnent une variété (non linéaire). Les modèles linéaires tels que ACP ou MDS ne peuvent pas être utilisés pour analyser de manière fiable les ensembles non linéaires (Cf chapitre 5).

2.2 Segmentation d'image et apprentissage statistique

Le but de la segmentation est de diviser le domaine d'une image en N régions disjointes qui correspondent, en général, à des objets partiels ou complets.

Historiquement, la segmentation d'image délimitait les objets en utilisant uniquement leurs bords, où les valeurs du gradient de l'image sont maximales. Les régions du domaines de l'image peuvent aussi, par exemple, être détectées selon des contraintes de regroupements de pixels (uniformité, propriétés statistiques etc ... à l'intérieur de la région).
Dans ce travail, nous nous limitons à deux régions séparées par un contour. Beaucoup d'approches définissent une courbe, notée S, qui minimise une énergie. Par exemple, les *graph cuts* trouvent la courbe qui a le minimum global de l'énergie. Dans cette thèse, nous nous utiliserons principalement des approches variationnelles par descente de gradient afin de minimiser une fonctionnelle d'énergie : on fait évoluer la courbe S sur le domaine de l'image de manière à minimiser une énergie variationelle E. En général, on écrit une énergie E_{data} qui attire la courbe S vers les bords de l'objet à segmenter. Les propriétés statistiques à l'intérieur ou à l'extérieur de la courbe peuvent également être utilisées pour décrire l'énergie E_{data}. Voir la section (4.3.3) pour la segmentation par contour actifs basée sur les régions.

$$E_1(S) = E_{\text{data}}(S) \tag{2.1}$$

Les bords dans images sont parfois mal définis car flous, bruités ou recouverts par d'autres objets. Une contrainte de lissage E_{smooth} s'impose alors le long de la courbe. Cependant, une telle contrainte n'est en général pas suffisante et un *a priori* sur les formes possible doit être introduit. L'apprentissage statistique est utilisé de manière intensive en vision par ordinateur pour introduire des *a priori*, particulièrement dans les processus de segmentation d'image.

Dans ce travail, nous nous concentrons sur l'apprentissage d'*a priori de formes* à partir d'un ensemble de formes. La segmentation avec a priori de formes s'écrit mathématiquement à l'aide d'un modèle similaire, mais avec l'ajout d'une nouvelle énergie $E_{\text{a priori}}$. $E_{\text{a priori}}$, construite à partir d'un ensemble d'apprentissage, contraint la courbe dans un ensemble de formes possibles.

$$E_2(C) = E_{\text{data}}(C) + \lambda E_{\text{a priori}}(C) \tag{2.2}$$

où λ est un paramètre qui quantifie l'influence ou la contribution de l'*a priori* dans le résultat final de segmentation. A notre connaissance, les travaux de Daniel Cremers, Cristoph Schnörr, Joachim Weickert et Christian Schellewald sur les Diffusion Snakes [58, 57] sont les premiers à combiner une segmentation avec un

terme d'énergie supplémentaire.

2.3 But et organisation de ce manuscrit

De nombreux articles en segmentation d'image utilisent des *a priori* de forme à l'aide de techniques de réduction de dimensions. On trouve par exemple les travaux de Michael Leventon, Eric Grimson et Olivier Faugeras in [122], mais aussi [165, 42, 203, 40]. Cependant, ces approches supposent que les données sont sur un sous-espace linéaire selon une distribution gaussienne.

Dans ce travail, nous voulons sortir de l'hypothèse gaussienne afin d'obtenir des *a priori* de formes plus généraux, non linéaires. En particulier, nous modélisons une catégorie de formes — e.g. *la catégorie des formes de poissons* — comme une variété différentielle de dimension finie, non linéaire, que nous appelons la *variété des formes a priori*. Dans ce contexte, nous considérons que l'*a priori de formes* est une force dirigée vers la *variété des formes a priori*.

Bien que nous souhaitions manipuler des cas plus généraux, nous attirons l'attention du lecteur sur des travaux connexes proposés par Daniel Cremers, Timo Kohlberger et Christoph Schnörr [53]. Les auteurs introduisent des *a priori* de forme non linéaires en utilisant une version probabilistique de l'ACP à noyaux (KPCA).

L'objectif de cette thèse est d'étendre les techniques d'apprentissage de variété tels que les *diffusion maps*, et de créer une force qui attire des points vers la variété. Cette force doit être conçue pour être utilisable dans le contexte de la segmentation d'image. La contribution de cette thèse sera présentée dans une forme générique alors que les applications gardent une place importante. Notre approche est illustrée sous la forme d'un schéma, en figure 2.2.

Dans la suite, nous décrivons l'organisation de cette thèse.

Chapitre 3

Ce chapitre donne au lecteur les connaissances de base pour travailler avec des formes. En particulier, nous définissons le concept de forme et nous détaillons

quelques représentations possibles. Nous introduisons également la notion de distance entre formes. Enfin, nous expliquons comment interpoler entre les formes en utilisant des moyennes pondérées de forme.

Chapitre 4

Dans ce chapitre, nous passons en revue les techniques de segmentation d'image, au sens large du terme. Puisque les contributions de cette thèse ont été guidées par des applications en segmentation d'image, nous présentons le contexte et nous montrons comment notre approche franchit une nouvelle étape dans l'état de l'art.

Chapitre 5

Dans ce travail, l'objectif est d'apprendre des *a priori* de forme non linéaires à partir d'une catégorie de formes situées sur une variété différentielle. Le chapitre 5 se concentre sur l'apprentissage statistique, en particulier les *méthodes de réduction de dimensions*. L'organisation de chapitre est conçue pour souligner les besoins et l'importance de modèles non linéaires. Nous présentons et illustrons les méthodes les plus populaires depuis l'analyse en composante principales [151] ou le Multi-dimensional Scaling [201, 202, 51] jusqu'aux dernières techniques d'apprentissage non-linéaire de variétés, telle que les *Laplacian maps* [15] et *diffusion maps* [117].

Chapter 6 - Application I : Extraction intéractif d'image

Cette thèse repose principalement sur les techniques d'apprentissage de variété. Nous proposons donc une application des Laplaciens de graphe / diffusion maps, l'extraction intéractive d'images dans le cadre des techniques de *relevance feedback*. Dans ces approches, le système construit une métrique / règle de décision qui reflète les intentions de l'utilisateur, à partir de données étiquetées ou non (en général, l'utilisateur étiquette quelques données de manière binaire). En effet, nous supposons que les données échantillonnent une variété différentielle et les *diffusion maps* sont utilisés pour apprendre cette variété et diffuser les étiquettes (ce processus porte le nom d'apprentissage transductif). Une intéraction entre le système et l'utilisateur est alors créée afin de raffiner la réponse en proposant de nouveaux points non étiquetées à l'utilisateur. Finalement, le système renvoie une classe d'image censée répondre à la requête formulée dans l'esprit de l'utilisateur.

Publications liées à ce chapitre

Hichem Sahbi, Patrick Etyngier, and Jean-Yves Audibert and Renaud Keriven. Graph Laplacian for Interactive Image Retrieval. *ICASSP' 08 : In proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, Las Vegas, April 2008.

Hichem Sahbi, Patrick Etyngier, and Jean-Yves Audibert and Renaud Keriven. Manifold Learning using Robust Graph Laplacian for Interactive Image Retrieval. *CVPR' 08 : In proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, Anchorage, Alaska, June 2008.

Chapitre 7 :

Ce chapitre contient nos principales contributions. Notre approche repose sur une technique d'apprentissage de variété, les *diffusion maps*, qui construisent un plongement (i.e. une application) vers une représentation des données dans un espace de plus petite dimension. Le plongement, est défini seulement sur les échantillons de l'ensemble d'apprentissage. Le chapitre 5 présente d'abord les possibilités d'extension du plongement aux nouveaux points —i.e. *en dehors de l'ensemble d'apprentissage*. Ces extensions évitent de recalculer toute l'application plongement.

Notre but est de construire une force appliquée aux données, dirigée vers la variété apprise. L'idée naturelle est de construire un opérateur de projection pour obtenir une cible.

Nous proposons trois *forces attractives* dans ce chapitre. La première repose sur une projection minimisant la distance à la variété. Ensuite, la deuxième force repose sur une projection ayant la même valeur de plongement. Nous définissons enfin une troisième *force attractive* dirigée vers la variété qui préserve le plongement constant. Cette dernière approche, qui repose sur les *diffusion maps* [117] et l'extention de Nyström [145], est bien posée et naturelle.

Puisque les outils développés dans ce travail peuvent être appliqués à toute sorte de données équipées d'une distance différentiable, nous nous efforcerons de présenter ce chapitre dans une forme générique.

Publications liées à ce chapitre

Patrick Etyngier, Renaud Keriven, and Jean-Philippe Pons. Towards segmentation based on a shape prior manifold *SSVM' 07 : In proceedings of the 1st International Conference on Scale Space and Variational Methods in Computer Vision* Ishia, Italy, May 2007.

Patrick Etyngier, Renaud Keriven and Florent Ségonne. Projection Onto a Shape Manifold for Image Segmentation with Prior. *ICIP' 07 : In proceedings of the 14th IEEE International Conference on Image Processing*, San Antonio, Texas, USA, September 2007.

Patrick Etyngier, Renaud Keriven and Florent Ségonne. Shape priors using Manifold Learning Techniques. *ICCV' 07 : In proceedings of the 11th IEEE International Conference on Computer Vision*, Rio de Janeiro, Brazil, October, 2007.

Patrick Etyngier, Renaud Keriven and Florent Ségonne. Active-Contour-Based Image Segmentation using Machine Learning Techniques. *MICCAI' 07 : In proceedings of the 10th International Conference on Medical Image Computing and Computer Assisted Intervention*, Brisbane, Australia, October, 2007

Chapitre 8

Les outils développés dans cette thèse sont appliqués avec succès à la segmentation d'image avec a priori de formes non linéaires, ainsi qu'au débruitage de variétés. En particulier, le chapitre 8 illustre les résultats obtenus sur des exemples 2D et 3D de *variétés de formes a priori* dans différents contextes : poissons, voitures et même dans le domaine de l'imagerie médicale avec des ventricules.

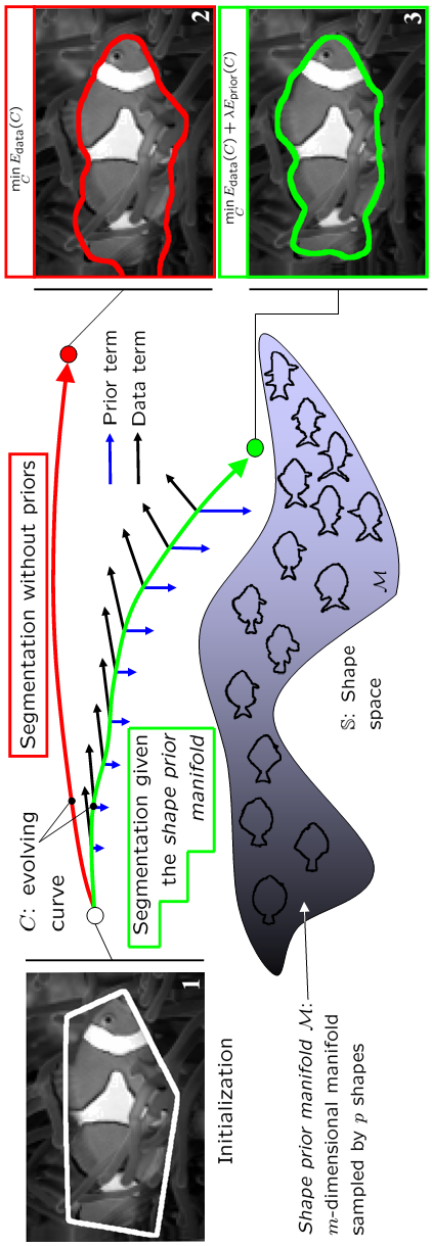

FIG. 2.2 – **Vue d'ensemble de notre approche et but cette thèse** : une catégorie de formes — e.g. *les formes de poissons* — est représentée comme une variété différentielle de dimension finie. L'*a priori de forme* est une force dirigée vers la variété et est combiné avec une force d'attache aux données

Chapter 3

Shapes

Abstract

In this chapter, we give the reader the very basic knowledge about shapes. We begin defining shapes and the shape space $\mathbb{S}$ following definitions given in Guillaume Charpiat's work [39] *and in a book written by Michel Delfour and Jean-Paul Zolesio* [62].
Then, we discuss shape representations in practice and outline some of them, which are commonly used in computer vision literature. We also tackle the question of comparing shapes and specifying a topology in the shape space.
Finally, we provide the reader with the notion of shape gradient and Karcher mean shapes, which we use to interpolate the shape space between shape samples.

Contents

Key points & Original contributions

We provide the reader with the necessary background about shapes.

We propose weighted Karcher means of shapes as a way to interpolate locally *shape manifolds*.

Related publications: SSVM'07 [75], ICIP'07 [76], ICCV'07 [81], MICCAI'07 [80].

3.1 Basic definitions for shapes

In this first section, we define the basics, in particular we introduce *shape spaces* by using definitions taken from [62].

Definition 1 (Shape) *Let $D \subset \mathbb{R}^n$ be a domain of interest, also named the image domain. A* shape *S is a bounded closed subset of D and we denote $\mathfrak{S}$ the space of such shapes.*

The set of shape $\mathfrak{S}$ is too large for our purpose and we need to restrict it. We construct a new shape space following Guillaume Charpiat's work [39] based on two subsets of the shape space $\mathfrak{S}$; the first being limited to shapes of which the boundary is smooth, the second to shapes that have limited S-curving along the boundary, below a given scale. We need some more definitions to specify properly these subsets.

Definition 2 (Distance function and oriented distance function) *The* distance function *$d_S(x)$ to a shape is denoted $d_S(x)$*

$$d_S(x) = \inf_{y \in S} d(x, y) = \inf_{y \in S} |x - y|$$

We also denote $\complement_S$ the complementary of shape S in D and define the oriented distance function *to a shape S (considered as a bounded region, subset of $\mathbb{R}^n$)*

$$b_S(x) = d_S(x) - d_{\complement_S}(x) \tag{3.1}$$

Definition 3 (Projection) *Given $S \subset D$, $S \neq \emptyset$, the set of projection of $x \in D$ on S is given by:*

$$\Pi_S(x) \stackrel{def}{=} \left\{ p \in \overline{S} : |p - x| = d_S(x) \right\}$$

Definition 4 (h-tubular neighborhood) *Given $S \subset D$, $S \neq \emptyset$, and a real number $h > 0$, the h-tubular neighborhood of S is defined as*

$$U_h(S) \stackrel{def}{=} \{ y \in D : d_S(y) < h \}$$

We can now define the two following sets used to build the set of shapes of interest.

Definition 5 (Set of smooth shapes) *The set $\mathcal{C}_0$ (resp $\mathcal{C}_1, \mathcal{C}_2$) of smooth shapes is the set of subset of D whose boundary is non-empty and can be locally represented as the graph of a $\mathcal{C}_0$ (resp. $\mathcal{C}_1, \mathcal{C}_2$) function.*

Definition 6 (Set $\mathcal{F}$ of shapes of positive reach - h_0-Federer's sets) *A non empty subset S of D is said to have positive reach if there exists $h > 0$ such that $\Pi_S(x)$ and $\Pi_{\mathtt{C}_S}(x)$ are a singleton for every $x \in U_h(S)$. The maximum h for which the property holds is called the reach of S and is denoted* $\mathrm{reach}(S)$
The h_0-Federer's set *[86] $\mathcal{F}_{h_0}$ is the subset of $\mathcal{F}$ such that* $\mathrm{reach}(S) > h_0$ *for all $S \in \mathcal{F}_{h_0}$.*

Definition 7 (Set of shapes $\mathbb{S}$) *The set, denoted $\mathbb{S}$, of shapes of interest is the subset of $\mathcal{C}_2$ whose elements are also in h_0-Federer's set for a given and fixed $h_0 > 0$.*

$$\mathbb{S} \stackrel{def}{=} \mathcal{C}_2 \cap \mathcal{F}_{h_0}$$

Note that the notation $\mathbb{S}$ should indicate the dimension of the domain D, but we skip it in the text for the sake of clarity since it can be easily deduced from the context.

Shape spaces such as $\mathbb{S}$ are of particular interest due to their inherent properties and complexities. They are indeed neither vector spaces since 2 shapes cannot be added, nor finite dimensional spaces since infinite degrees of freedom in such shape spaces exist. Nevertheless, they can be assumed to be differentiable manifolds equipped with a Riemannian metric [138].

3.2 Shape representations in practice

3.2.1 Introduction

In practice, there are many ways to tackle the problem of representing *shapes*. A first approach is to consider a shape as a simple (i.e. non-intersecting) closed curve, surface or hypersurface lying in a larger dimensional space, in other words by the contour of the object of interest. We denote such representations as *boundary shapes*. Alternatively, a second aspect is to comprehend a shape as a closed bounded region of the domain space; the shape is the region itself and *full shape* stands for such representations. *Full shapes* are found for instance in Marcin

Iwanowski's [105] or Jan Erik Solem's works [191, 190]. Note that the links between both representations is beyond the scope of this thesis and we will equally switch from one representation to the other.

Geometrical representations of shapes by their boundaries have received a lot of attention in the computer vision and graphics literatures, and many models of deformable surfaces have emerged. For a complete review, the reader is referred to [141]. Among the continuous models, we can roughly divide these representations into two classes, and we outline a very limited number of them. On the one hand, the first strategy consists in coding explicitly the boundary shape by means of control points and splines, parameterized hypersurfaces or meshes. On the other hand, the second representation codes the boundary implicitly as the isolevel of a scalar function of higher dimension.

In variational approaches, the curve S deforms and evolves by applying deformation fields, which lie on the tangent bundle of the shape space $\mathcal{S}$. In addition, an objective functional E is minimized and a gradient descent is used. Without loss of generality, the gradient of energy $E(S)$, denoted $\nabla_S E$, is a deformation field and the scheme is given as follow (See section (3.4.2) and Guillaume Charpiat's PhD dissertation for the definition of $\nabla_S E(S)$ [39]).

$$\left\{ \begin{array}{rcl} S(t) & = & S_0 \\ \dfrac{\partial S}{\partial t} & = & -\nabla_S E \end{array} \right. \tag{3.2}$$

where S_0 is a given initial curve and t is the time variable. This will be detailed in the sequel.

3.2.2 Explicit form

Parametrized representations

Most explicit methods require a parametrization of the hypersurface (B-splines [174] , Fourrier harmonics [193], parametric equations such as superquadrics in [197] to cite a few).

We now focus on planar curves. Let $S(q) : [0, 1] \longrightarrow \mathbb{R}^2$ be a parameterized curve. Michael Kass, Andrew Witkin and Dimitri Terzopoulos [108] firstly pro-

posed in a seminal work an *active contour model* called the *snakes* model. It is based on a parameterized curve $S(q)$ associated with an energy designed for image segmentation. This energy will be described in the following chapter.

B-splines are used, for instance, by Daniel Cremers, Timo Kohlberger and Christoph Schnörr in the context of image segmentation with shape priors [53]. B-splines (B stands for basis) form a basis $B_1(q), \ldots, B_N(q), q \in [0,1]$ of which the elements are minimal support functions with respect to a given domain partition, degree and smoothness. Shapes are then approximated by piecewise polynomial parametric curves, and represented by a linear combination of B-splines using a set of control points $z = x_1, y_1, \ldots, x_N, y_N$.

$$\begin{array}{ccccc} S_z(q) & [0,1] & \rightarrow & & \Omega \subset \mathbb{R}^2 \\ & s & \mapsto & S_z(q) = \sum_{n=1}^{N} \begin{pmatrix} x_n \\ y_n \end{pmatrix} B_n(q) \end{array} \tag{3.3}$$

The main drawback of parameterized curves is that the energy minimization depends on the parametrization. Indeed, let $E_a(s)$ denote an arbitrary energy of a parameterized curve $S(q)$ of the following form.

$$E_a(S) = \int_0^1 f\left(S(q)\right) dq \tag{3.4}$$

where f is a function $\mathbb{R}^2 \longrightarrow \mathbb{R}$. We now define a new parametrization of the curve, via $q = \eta(r)$, $\eta : [c,d] \longrightarrow [0,1], \eta' > 0$, which leads to a new energy $E_b(S)$.

$$E_b(S) = \int_c^d f\left(S(q \circ \eta(r))\right) \frac{\partial \eta}{\partial r}(r) dr \tag{3.5}$$

Thus, the energy can arbitrarily be changed by choosing a new parametrization not intrinsic to the curve. We will detail in the sequel how to evolve a curve by minimizing this kind of energy in the variational framework. Accordingly, it means that this shape representation should not be employed because their evolution might depend on the parametrization q.

Furthermore, explicit methods do not allow topology change during the evolution. Note that one can, however, find some attempts to design explicit methods with topology changes. For example, François Leitner and Philippe Cinquin [121] build B-splines with basic topology changes.

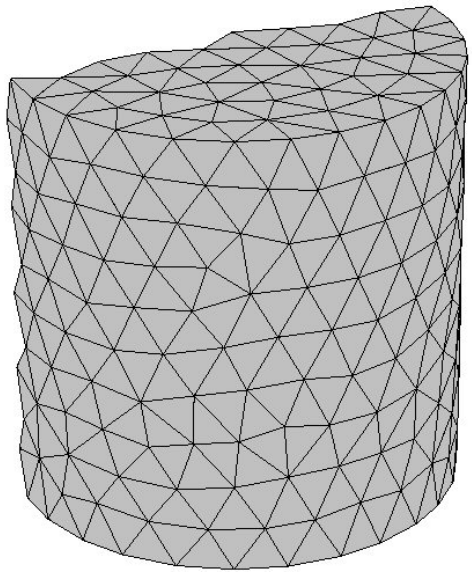

Figure 3.1: **Example of mesh for a half cylinder surface**

Non-parameterized representations

Meshes are explicit but not parameterized representations of surfaces. A *mesh* in computer graphics is a collection of vertex points sampling a hypersurface, and connected by edges, faces or hyperfaces (depending on the dimension). For an introduction to algorithmic geometry, please refer to Jean-Daniel Boissonnat's and Mariette Yvinec's textbook [25].

Deformable surfaces represented by meshes cannot basically change their topology during the evolution, unlike the *Level Set* method. The authors in [129, 130] propose however topology adaptive meshes, named *T-snakes* and *T-surface*, which consists in resampling periodically the curve or the surface. The method has serious drawbacks such as fixing a uniform spatial distribution, and the curve has limited admissible motions.
Note that some methods have been proposed to handle topological changes in meshes [116, 63, 64] but they are computationally expensive. Nevertheless, a very recent work by Jean-Philippe Pons and Jean-Daniel Boissonnat [157] introduces a new efficient method to deal with general topological changes during surface

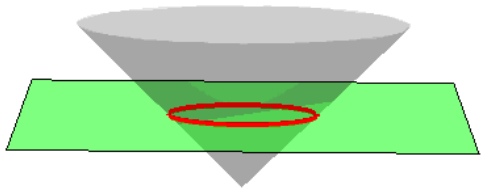

Figure 3.2: **Representation of the unit circle as the isolevel (in red) of a scalar function (in grey)**

evolutions. Although the *Level Set* technique (see below) is well established in computer vision, new advances in algorithmic geometry might give serious competitive advantages to meshes.

3.2.3 Implicit functions and distance functions

Implicit methods represent shape boundaries as the isolevel of a scalar function of higher dimension. We illustrate this idea with a very basic example, the unit circle in the plane $\mathbb{R}^2$. Consider for instance the function f defined by:

$$\begin{array}{rccl} f: & \mathbb{R}^2 & \rightarrow & \mathbb{R} \\ & (x,y) & \mapsto & f(x,y) = x^2 + y^2 - 1 \end{array} \tag{3.6}$$

The implicit representation of the unit circle is then given by the 0-isolevel of f i.e. the set of points $\{(x,y), f(x,y) = 0\}$. In other words, the shape of the unit circle in the plane $\mathbb{R}^2$ is implicitly represented by the codimension 1 isolevel of a 3 dimensional surface defined on a domain $D \subset \mathbb{R}^2$ (Fig. 3.2). Generalization to higher dimension is straight; indeed a p dimensional shape is seen as the isolevel of a n dimensional surface, with $n \geq p+1$. Different forms of parametric functions f are found in literature (algebraic surfaces [92], superquadrics [13], hyperquadrics [46]) but applications are limited to a small family of shapes.

A very popular and well established technique, called *Level Set*, uses implicit representations. It was sketched by Alain Dervieux and François Thomasset [67] and developed by Stanley Osher and James Sethian [147, 185, 146]. The *Level Set* technique is of particular importance since it handles complex geometries, useful in many applications (it will be detailed in the image segmentation framework in section (4.3.3)). In this method, the 0-level of the signed distance function $b_S(x)$

to the boundary ∂S is often used to represent the shape S. Thus, the function f is not explicitly expressed (Eq. 3.7).

$$f(x) = b_S(x)s \tag{3.7}$$

Many methods exist to calculate the distance function $b_{\partial S}(x)$ over a domain D. Among the basic methods in literature, we find Per-Erik Danielson's algorithm [61], chamfer distances [198, 88]. See also [59, 26].
A more elaborate means of obtaining the distance function $b_{\partial S}(x)$, is to solve the following partial differential equation (PDE) [183] on an implicit function (the reader is also referred to [11])

$$\begin{cases} \dfrac{\partial f}{\partial t} + \text{sign}\left(f\left(x, t_0\right)\right)\left(|\nabla f| - 1\right) = 0 \\ f\left(x, 0\right) = g\left(x, t_0\right) \end{cases} \tag{3.8}$$

where g is an initial estimate of which the zero level set represents the shape. At convergence, f is a signed distance function.
Finally, *Fast Marching* methods by James Sethian [184] [194], are the most efficient way to compute distance functions in speed.

3.3 Shape space, topology and distances

3.3.1 Distances between shapes

The notion of regularity involved by the manifold viewpoint requires a definition which shapes are close and which shapes are far apart. However, there is no agreement in computer vision literature on the right way of measuring shape similarity. Many different definitions of the distance between two shapes have been proposed, depending on the shape representation.

Symmetric difference

One classic choice is the area of the symmetric difference between the regions bounded by two shapes S_1 and S_2:

$$d_{SD}(S_1, S_2) = \frac{1}{2}\int \left|\chi_{\text{Int}(S_1)} - \chi_{\text{Int}(S_2)}\right| \tag{3.9}$$

where χ_Ω is the characteristic function over the domain Ω. This distance was proposed by Daniel Cremers, Stanley Osher and Stefano Soatto in [54].

Hausdorff distance

Another classic definition of distance between shapes is the Hausdorff distance, appearing in the context of shape analysis in image processing in the works of Jean Serra [182], in the context of mathematical morphology.

$$d_H(S_1, S_2) = \max \left\{ \sup_{x \in S_1} \inf_{y \in S_2} \|x - y\| \ , \ \sup_{y \in S_2} \inf_{x \in S_1} \|x - y\| \right\} \ . \tag{3.10}$$

Guillaume Charpiat, Olivier Faugeras and Renaud Keriven propose also in [40] a boundary-based Hausdorff distance $d_H(\partial S_1, \partial S_2)$. A comparison between both approaches can be found in Guillaume Charpiat's PhD dissertation [39]. Note that the Hausdorff distance is not continuous for the Hausdorff topology, while it is essential when using variational methods. To overcome such a limitation, the authors in [39] define a smooth differentiable approximation of the boundary-based Hausdorff distance.

Level-set based distance

Another definition, based on the *Level Set* representation, has been proposed in [122, 165, 40]. 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:

$$d_{L^2}(S_1, S_2)^2 = ||\bar{D}_{S_1} - \bar{D}_{S_2}||^2_{L^2(\Omega, \mathbb{R})} \ , \tag{3.11}$$

$$d_{W^{1,2}}(S_1, S_2)^2 = ||\bar{D}_{S_1} - \bar{D}_{S_2}||^2_{L^2(\Omega, \mathbb{R})} + ||\nabla \bar{D}_{S_1} - \nabla \bar{D}_{S_2}||^2_{L^2(\Omega, \mathbb{R}^n)} \ , \tag{3.12}$$

where $\bar{D}_{S_i}$ denotes the signed distance function of shape S_i ($i = 1, 2$), and $\nabla \bar{D}_{S_i}$ its gradient.

3.4 Smooth deformation, shape [sub-]manifolds, & interpolation in the shape space

3.4.1 Introduction

The question of how to "deform smoothly" a shape into another is central in many applications of computer vision or medical imaging, but is very complex due to the particular properties of the space $\mathbb{S}$. In the previous section, we supposed that the shape space $\mathbb{S}$ can be seen as a differentiable manifold. Once this assumption is established, one can construct geodesics in the shape space based on suitably chosen metrics [132, 138].

In this work, we also model a category of shapes as a finite dimensional submanifold lying in $\mathbb{S}$, that we name *shape manifold* in the sequel. Note that we will not be parameterizing *shape manifolds*. Instead, we assume that they can be learnt from a set of sample shapes by using manifold learning techniques, as carried out in chapter 5. This view leads to the problem of extending shape manifolds out of the samples by using interpolation. In our context, the underlying idea of building geodesics is indeed to interpolate between shapes in the space $\mathbb{S}$. Further in the text, we also propose a "simple interpolation" between several shapes based on Karcher means [107].

3.4.2 Shape gradient & Gâteaux derivatives

In order to deform shapes, we need to define the notions of *normal deformation field* and *functional derivative*, in particular in the shape space. Indeed, we are driven by variational approaches and energy minimization methods based on gradient descent. Smooth shape deformation is achieved by applying successively small normal deformation fields to the evolving shape. *Shape gradients* are *deformation fields* calculated from the gradient of a given energy with regards to the evolving shape. In this work, shape gradient will be *normal deformation fields* (See extended gradients in [39] for details)

Applying a *normal deformation field* to a shape

Since the shape space $\mathbb{S}$ is considered as a smooth Riemannian manifold, any shape $S \in \mathbb{S}$ has a tangent space denoted $\mathbb{T}_S$. $\mathbb{T}_S$ is seen as a vector space, and its

elements define *normal deformation fields* $\mathbf{U}_S \in \mathbb{T}_S$. Let S be parameterized as $S : [0,1] \longrightarrow \mathbb{R}^2$ with a corresponding normal unit at p denoted $\mathbf{n}(p)$. We denote then *normal deformation fields* by $\forall p \in [0,1]$, $\mathbf{U}_S = U_S(p)\mathbf{n}(p)$. For the sake of clarity, we will not be specifying the parametrization p in the sequel.
$S + \mathbf{U}_S$ is not in the shape space $\mathbb{S}$ in general. Nevertheless, if $\mathbf{U}_S$ is smooth enough (i.e. it is C^2), and $\varepsilon \in \mathbb{R}$ is small enough too, then $S + \varepsilon\mathbf{U}_S$ remains in $\mathbb{S}$ [39].

Functional derivative: the Gâteaux derivative

The *shape gradient* relies on the Gâteaux derivative, which is a generalization of the usual derivative to more general spaces such as some functional spaces. Let $\mathcal{V}$ and $\mathcal{W}$ be locally convex topological vector spaces (Banach spaces for example). We also define the mapping $E : \mathcal{V} \longrightarrow \mathcal{W}$. In our context, $\mathcal{V} = \mathbb{T}_S$, $\mathcal{W} = \mathbb{R}$ and the above hypotheses are assumed to be verified [39]. Now, let $\mathbf{U}_S \neq 0$ be an element of $\mathbb{T}_S$. The Gâteaux derivative $dE(S, \mathbf{U}_S)$ of S in the "direction" of the deformation field $\mathbf{U}_S$ is defined as follow.

$$dE(S, \mathbf{U}_S) = \lim_{\varepsilon \to 0} \frac{E(S + \varepsilon\mathbf{U}_S) - E(S)}{\varepsilon} \tag{3.13}$$

$$= \frac{d}{d\varepsilon} E(S + \varepsilon\mathbf{U}_S)\Big|_{\varepsilon=0} \tag{3.14}$$

Note that we have $E(S, +\varepsilon\mathbf{U}_S) \approx E(S) + \varepsilon dE(S, \mathbf{U}_S)$.

Variational approach and energy minimization for smooth shape deformation

The mapping $\mathbf{U}_S \longrightarrow dE(S, \mathbf{U}_S)$ defines a linear continuous form in $\mathbb{T}_S$ and, as such, the Riez representation theorem is employed: a deformation field denoted $\nabla_S E(S)$ and named the *shape gradient* of the energy E exists such that

$$\forall \mathbf{U}_S, \quad dE(S, \mathbf{U}_S) = < \nabla_S E(S), \mathbf{U}_S > \tag{3.15}$$

where the scalar product $< \cdot, \cdot >$, is implicitly supposed to be the standard Hilbert scalar product $< f, g >= \int f(x) \cdot g(x)dx$. The direction of steepest descent is given by $\mathbf{U} = -\nabla_S E(S)$. Generalized gradients have been proposed in [40], in order to overcome local minimums in the energy landscape and favor rigid motion by modifying the standard product to more generalized ones. The question of the

action of group transformations on shapes arises and the reader is referred for example to [139, 215, 195].

Given an initial shape $S(0)$, we want to establish a family of shapes $S(t)$, $t > 0$ such that $E(S(t+1)) \leq E(S(t))$. The shapes $S(0), S(1), \ldots$ actually sample a "path" of smooth deformation until a minimum of energy E is reached, and is obtained using a gradient descent approach. We finally obtain the following partial differential equation (PDE).

$$\begin{aligned} \frac{\partial S}{\partial t} &= -\nabla_S E(S) \\ S(0) &= S_0 \end{aligned} \tag{3.16}$$

Example of shape deformation within the *Level Set* framework

We briefly outline evolutions of curves within the *Level Set* framework. First, the gradient of the distance function gives the direction of the normal at any point of the shape. Let $\mathbf{N}$ be the outward unit normal and ϕ be a distance function. Within the *Level Set* framework, $\mathbf{N}$ is expressed in equation (3.17).

$$\mathbf{N} = \frac{\nabla \phi}{|\nabla \phi|} \tag{3.17}$$

Given a deformation field $\mathbf{v}$, the evolution is implemented with the following Partial Differential Equation (PDE).

$$\frac{\partial \phi}{\partial t} = -\mathbf{v} \cdot \nabla \phi \tag{3.18}$$

Note that only the normal component of $\mathbf{v}$ is considered.
One can easily show that (Eq. 3.18) is equivalent to $\partial S/\partial t = \mathbf{v}$ for the 0-isolevel. Yet, shape S corresponds to the 0-isolevel of the distance function ϕ and we have a straight correspondence between $\partial S/\partial t$ and $\partial \phi/\partial t$. We now link (Eq. 3.18) with (Eq. 3.16). Equation (3.18) says that each level of ϕ evolves according to the deformation field $\mathbf{v}$, usually a *shape gradient*, in the normal direction of the curve. Equation (3.16) says similarly that shape S evolves according to a *shape gradient*.

ϕ_t should be regularly reinitialized, for instance to a distance function, to make sure that the gradient descent does not become unstable. Note that specific implementations may impose ϕ_t to remain a distance function while iterating over the

time index t [97].

Mean Curvature Motion [82] is an example of shape deformation that minimizes the length of a closed curve (or the area of a closed surface) by moving its points in the normal direction with a velocity proportional to its curvature. For the sake of completeness, we start by defining a parameterized curve $S(q) : [0,1] \longrightarrow \mathbb{R}^2$. Let us also define the energy $E_l(S)$ which calculates the length of S.

$$E_l(S) = \int_0^1 \left| \frac{\partial S(q)}{\partial q} \right| dq \tag{3.19}$$

As previously mentioned, the energy $E_l(S)$ might be changed by choosing a new parametrization not intrinsic to the curve. To overcome such limitation, we choose the curvilinear abscissa as intrinsic parametrization of the curve S. Note that when the curvilinear abscissa parametrization is chosen, we can write $ds = |\partial S(q)/\partial q| dq$ and express Eq. (3.19) depending on the length $L(S)$ of the curve.

$$E_l(S) = \int_0^{L(S)} ds \tag{3.20}$$

where ds is the Euclidian metric. We solve the Euler-Lagrange equation by using the Gâteaux derivative of energy $E_l(S)$, which gives the heat equation. This can be easily generalized to higher dimensions (surfaces, hypersurfaces).

$$\frac{\partial S}{\partial t} = \nabla_S E_l(S) = -\kappa \mathbf{N} \tag{3.21}$$

where $\kappa \mathbf{N} = \partial^2 S / \partial q^2$ is the curvature.
Even when the parametrization is intrinsic (by means of curvilinear abscissa), the evolution is not guaranteed to be causal and the curve should be reparameterized after each step to have a stable evolution. The *Level Set* implementation of equation (3.21) avoids such a process. The curvature κ within the *Level Set* framework is given by equation (3.22).

$$\kappa = \operatorname{div}\left(\frac{\nabla \phi}{|\nabla \phi|} \right) \tag{3.22}$$

The PDE is then given as follows, and figure (3.3) illustrates the evolution.

$$\frac{\partial \phi(x,y)}{\partial t} = -\kappa \left(\phi \left(x, y \right) \right) |\nabla \phi(x,y)| \tag{3.23}$$

During the evolution, the shape becomes convex and then disappears into a “circle point” in a finite time.

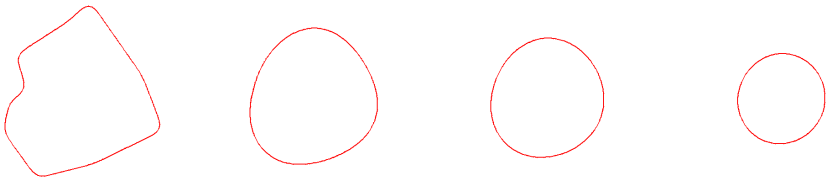

Figure 3.3: **Mean Curvature Motion: Evolution of curves in the Level Set framework** following equation (3.23): the shape becomes convex and then disappears into a "circle point" in a finite time

3.4.3 *Shape manifold* interpolation

Introduction & assumptions

In this work, *shape manifolds* are learnt from a finite set of training samples using dimensionality reduction techniques, as previously mentioned. Basically, these techniques build a mapping to a low dimensional space that preserves the local neighborhood information, based on a neighborhood graph (see chapter 5). In-between shapes on *shape manifolds* remain however unknown. In this section, we tackle such a problem, provide simple approaches to interpolate in the shape space and propose a solution to the interpolation of *shape manifolds*.

How to interpolate in the shape space ?

First, the minimization scheme presented in the previous section is used to interpolate between two shapes. Indeed, any differentiable distance such as the Sobolev distance or the smooth differentiable approximation of the Hausdorff distance can be employed as energy E. Given two shapes S_1 and S_2, we build a smooth deformation path $S_1 \rightarrow S_2$ by minimizing $d(S, S_2)$ with S_1 as initial shape. Nevertheless, the paths $S_1 \rightarrow S_2$ and $S_2 \rightarrow S_1$ are not symmetrical in general, and cannot be used to parameterize locally the shape space. Furthermore, it cannot be easily extended to more shapes when $n > 2$.

Inspired by H. Karcher [107], Guillaume Charpiat, Olivier Faugeras and Renaud Keriven [40] propose to use a mean of n shapes $S_1, \cdots, S_n$, the shape $\bar{S}$

given by:

$$\bar{S} = \arg\min_{S} \sum_{i=1}^{n} d\left(S_i, S\right)^2$$

Following the same path, we propose to build *weighted Karcher means* to interpolate between n sample shapes $S_1, \ldots, S_n$.

Definition 8 (Weighted means between n shapes)
Let $S_1, \cdots, S_n \in \mathbb{S}$ be n shapes and $\Lambda = \{\lambda_1, \cdots \lambda_n\}$ be n respective weights such that $\begin{cases} \sum_i \lambda_i = 1 \\ \forall i \in 1, \cdots n \quad \lambda_i \geq 0 \end{cases}$. *The weighted mean shape is given as follows.*

$$\bar{S}(\Lambda) = \arg\min_{S} \sum_{i=1}^{n} \lambda_i d\left(S_i, S\right)^2 \tag{3.24}$$

As in [40], the weighted mean $\bar{S}(\Lambda)$ is obtained by a gradient descent, a shape S evolving according to a gradient flow:

$$-\sum_{i} \lambda_i d\left(S_i, S\right) \nabla d\left(S_i, S\right) \tag{3.25}$$

We now briefly detail the case $n = 2$, where $\Lambda = \{\lambda_1, \lambda_2\}$ with $\lambda_1 = 1 - \lambda_2 = \lambda$ and build an interpolation path of which the shapes correspond to the values of λ in the unit segment $[0, 1]$.
Let $\lambda(1), \ldots, \lambda(p)$ be a uniform discretization of the unit segment $[0, 1]$ such that $\lambda(1) = 0$ and $\lambda(p) = 1$. The Karcher mean shapes $\bar{S}(\Lambda(1)), \ldots, \bar{S}(\Lambda(p))$ that correspond to $\lambda(1), \ldots, \lambda(p)$ represent a discretization of the interpolation path we aim to build ($\forall i \in \{1, \ldots p\}$, $\Lambda(i) = \{\lambda_1(i) = \lambda(i), \lambda_2(i) = 1 - \lambda(i)\}$). Note that $\bar{S}(\Lambda(1)) = S_1$ and $\bar{S}(\Lambda(p)) = S_2$. In order to avoid local minima, we proceed iteratively: $\forall i = 2, \ldots, p$, the shape $\bar{S}(\Lambda(i))$ is calculated by solving equation (3.24) with the initial shape $\bar{S}(\Lambda(i-1))$.
Figure (3.4) shows an example of interpolation path between a bird and rabbit, by using Karcher means. Although it involves two shapes only, it should be noted that: (i) the number of shapes is not limited to $n = 2$, and (ii) even when $n = 2$, the path defined by the weighted means is neither a geodesic for some distance, nor a straight gradient descent from S_1 to S_2.
Nevertheless, the uniqueness of the means is not proved and we still cannot ensure that the path is completely symmetrical.

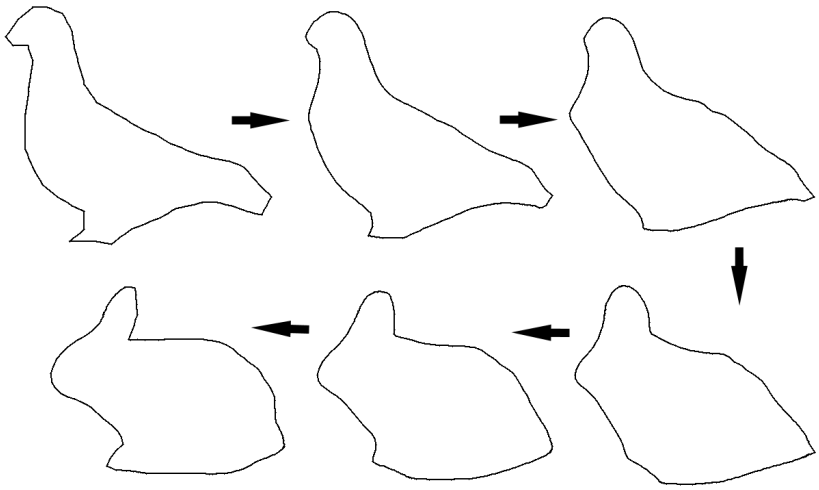

Figure 3.4: **Interpolation in the shape space** between two points (a bird and a rabbit) using six weighted means.$\lambda = 0, 0.2, 0.4, 0.6, 0.8$ and 1

To cope with this non uniqueness issue, we also tried to develop another idea, based on the works of Fernand Meyer [136], Jean Serra [182] and Serge Beucher [18]. The authors all build shape geodesics in the context of mathematical morphology.
Given two shapes S_1, S_2 and an evolving intermediate shape S, the sum $d(S, S_1) + d(S, S_2)$ should remain constant equal to $d(S_1, S_2)$. Such a condition is necessary but not sufficient to build geodesic path (see the proof in [39]). We design the following energy based on this idea and mean shapes, the parameter λ_i being used to parameterize intermediate shapes.

$$\min_S \sum_{i=1}^{n} (v_i(S) - \lambda_i v)^2 \quad \text{with} \quad \sum_{i=1}^{n} \lambda_i = 1 \tag{3.26}$$

where v is the hypervolume formed by the $n+1$ shapes, $v_i(S)$ is the i^{th} hypervolume formed by n shapes and the shape S. When $n = 2$, it has the simple following form.

$$\min_S \left[d\left(A, S\right) - \lambda d\left(A, B\right)\right]^2 + \left[d\left(B, S\right) - (1-\lambda)\, d\left(A, B\right)\right]^2$$

The implementation is however not straight when $n > 2$ and experiments with $n = 2$ and different distances have not shown to be better in general.

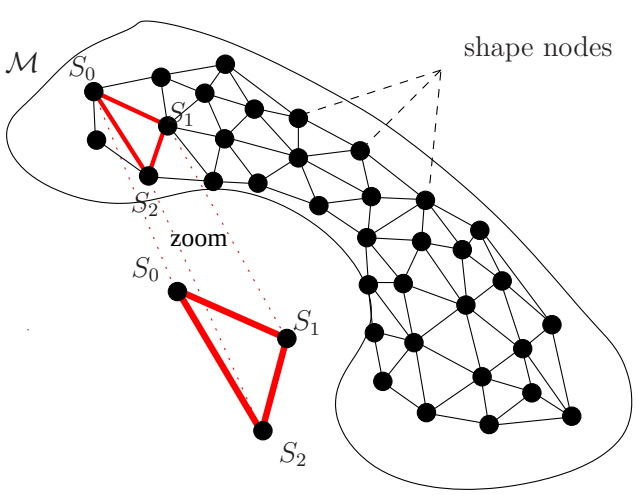

Figure 3.5: **Local interpolation of the *shape manifold* $\mathcal{M}$ of dimension n:** *weighted mean shapes* are used to locally interpolate $\mathcal{M}$ within a given n dimensional simplex (in red), that linearly and locally approximates $\mathcal{M}$

Local interpolation of *shape manifolds*

Let $\mathcal{M}$ be a n dimensional *shape manifold* sampled by p shapes. We assume that the p samples are the nodes of a n-dimensional mesh lying in the shape space and forming n dimensional simplexes (and so convexes). The construction of such meshes will be detailed in the sequel using the Delaunay triangulation and the diffusion maps technique [117] (see chapter 5). Since a category of shape is seen as a smooth manifold (a *shape manifold*), we also suppose that simplexes are good local linear approximations of the shape manifold. Now, the question is *how to build intermediate shape within a given simplex made of* $n + 1$ *shape nodes ?*

We suppose that *Karcher mean shapes* can roughly locally interpolate in $\mathbb{S}$ a n dimensional *shape manifold* between $n + 1$ shape node $S_0, \cdots, S_n$ forming a simplex $\mathbb{N}$. Λ can be viewed as a local parametrization of the shape manifold $\mathcal{M}$ within the simplex $\mathbb{N}$. The set covered by $\bar{S}_{\mathbb{N}}(\Lambda)$ for all the possible values of Λ

provides a continuous approximation of the manifold within simplex $\mathbb{N}$. This is illustrated in figure (3.5).

Chapter 4

Image segmentation

Abstract

Although the contributions of this thesis are primarily presented in a generic form (see chapter 7), we are driven by applications in image segmentation. We then carefully pay attention to highlight how our work fits in with the context of prior advances in this field of computer vision.

Contents

Key points

We give a short review of the state-of-the-art techniques in image segmentation.

4.1 Introduction

Image segmentation is a manual or automated task that achieves a partition of an image domain into disjoint sub-regions of interest. Basically, its goal is to distinguish objects from background in two or three dimensional images as illustrated in figure (4.4). Segmentation is a central problem in computer vision and has many application fields such as medical imaging or [semi-]automatic surveillance, for recognition, identification and detection. Nevertheless, it still remains an ill-posed problem due to various perturbing factors such as noise, occlusions, missing parts, cluttered data, etc.

The image segmentation problem can be approached from two opposite sides, by considering discontinuities or regions in images. On the one hand, discontinuities are detected to highlight edges in the image, pixels are classified into two groups: edge pixels and non-edge pixels. These methods date back to the beginnings of image processing [161, 158, 35, 65] to cite earlier works. On the other hand, regions with a given statistical property (often uniform regions) are extracted. Theses techniques were introduced latterly in literature as in [33, 142, 20, 152, 220, 125, 187] to cite a few. Most image segmentation techniques rely on these two views and some more recent methods use advantages of both. Roughly, edge-based methods have better detection precision and are easier to implement while region-based are more robust but less precise on edges.

The rest of this chapter is organized as follows. In section 4.2, we introduce fundamentals of image segmentation based on low level vision operators and in the broad sense. We then focus on active contour segmentation in section 4.3 and finally review image segmentation with shape priors in section 4.4.

4.2 Image segmentation

4.2.1 Edge detection

At the lowest level of image analysis, segmentation attempts to extract edge pixels which consist of high variations of image intensity that delineate uniform sub-regions. Let I be an image defined on a domain D as $I : D \subset \mathbb{R}^2 \longrightarrow \mathbb{R}$ and ∇I

its gradient.

$$\begin{aligned} D \subset \mathbb{R}^2 : &\longrightarrow \mathbb{R}^2 \\ (x, y) &\longmapsto \nabla I = \begin{bmatrix} \partial I/\partial x \\ \partial I/\partial y \end{bmatrix} \end{aligned} \tag{4.1}$$

Basics of edge detection rely on local derivative operators such as the gradient of which the norm $||\nabla I||$ is used to measure the variability of the image around a given pixel. Among the most popular edge detectors based on local operators are the *Canny filter* [35], the *Sobel filter* and the *Deriche filter*[65], but also the *Robert filter* [161], the *Prewitt filter* [158] etc. Robert's, Prewitt's and Sobel's operator are a discrete approximation of the gradient (Eq. 4.1). Pixels are then classified as edges or non edges by means of a single threshold.

Canny's operator seeks to extract edges with reliable detection, accurate localization and unique response. By using calculus of variation, John Canny found an "optimal" linear operator, which approximates the first derivative of the Gaussian operator applied to the image. The Gaussian operator smooths the image to remove noise. For one dimensional signals, the filter is given by the following function $f(x)$, and the output is then thresholded.

$$f_{\text{Canny}}(x) = Axe^{-\frac{x^2}{2\sigma^2}} \tag{4.2}$$

Deriche's edge detector is another "optimal" filter close to the Canny's operator. It can however be implemented very efficiently by using a recursive filter.

$$f_{\text{Deriche}}(x) = Axe^{-\frac{|x|}{\sigma}} \tag{4.3}$$

Hysteresis thresholding is employed right after applying the Canny or Deriche filter to the image. It consists in removing non maxima in the direction of the gradient from the filter response by setting two thresholds, t_l (low) and t_h (high) such that $t_l < t_h$. Pixels with value higher than t_h are classified as edge and lower than t_l as non-edge. Both Canny and Deriche filters involve directional informations, which is used to trace edges and determine the state of pixels with intermediate values, higher than t_l and lower that t_h. Then, in-between pixels can still be identified as edges if they are neighbors of edge pixels. This process produces a binary image and allows us to mark out strong edges while weak edges are maintained and noise is reduced.

In order to extract closed edge paths, second order derivative operators have been

developed in literature (detection is achieved by finding the zero-crossing of the second derivative). Among the most common filters are Laplacian operator, Laplacian of Gaussian operator and difference of Gaussian. For a more complete review and recent developments of edge detection, the reader is referred to [125].
We illustrate the results of four common operators (Sobel, Laplacian of Gaussian Canny and Deriche) in figure (4.1).

4.2.2 Region-based / Pixel-grouping methods

Introduction

Another point of view to image segmentation is to search for uniform sub-regions. Formally, region based methods divide the image domain D into N regions $\{D_1, \ldots, D_N\}$ such that $D = \bigcup_{i=1}^{N} D_i$ and $\forall i \neq j \quad D_i \cap D_j = \emptyset$, regions D_i being "uniform". In a very early work by Claude Brice and Claude Fennema in 1970 [33] region-based segmentation is introduced by using a *region growing* algorithm, which aggregates new pixels to a given region. The authors even use simple grammar rules for basic object recognition. *Region growing* has been also developed latterly in [1].

The *Mumford-Shah* energy

Smoothing images while preserving edges is the core idea of region-based segmentation. *Edges preserving smoothing* can be achieved using different techniques such as Markov Random Field [14] (commonly denominated MRF, seminal work by Donald Geman and Stuart Geman in [96]). Global optimization of a variational energy based on piece-wise representations is proposed by David Mumford and Jayant Shah in [142, 143]. The *Mumford-Shah* functional energy is a very popular model and extensively used in computer vision. It is described as follows: Given an image g and a resulting image f, the objective function should:

- **smooth the image** f: It involves a term of the following form $\iint_D |\nabla f|^2$
- **keep the image** f **closed to** g: The square difference should be simply minimized $\iint_D (f-g)^2$
- **allow discontinuities along edges and regularize the contours** The contours are grouped into a set $\mathcal{C}$, and the above smoothing term is modified

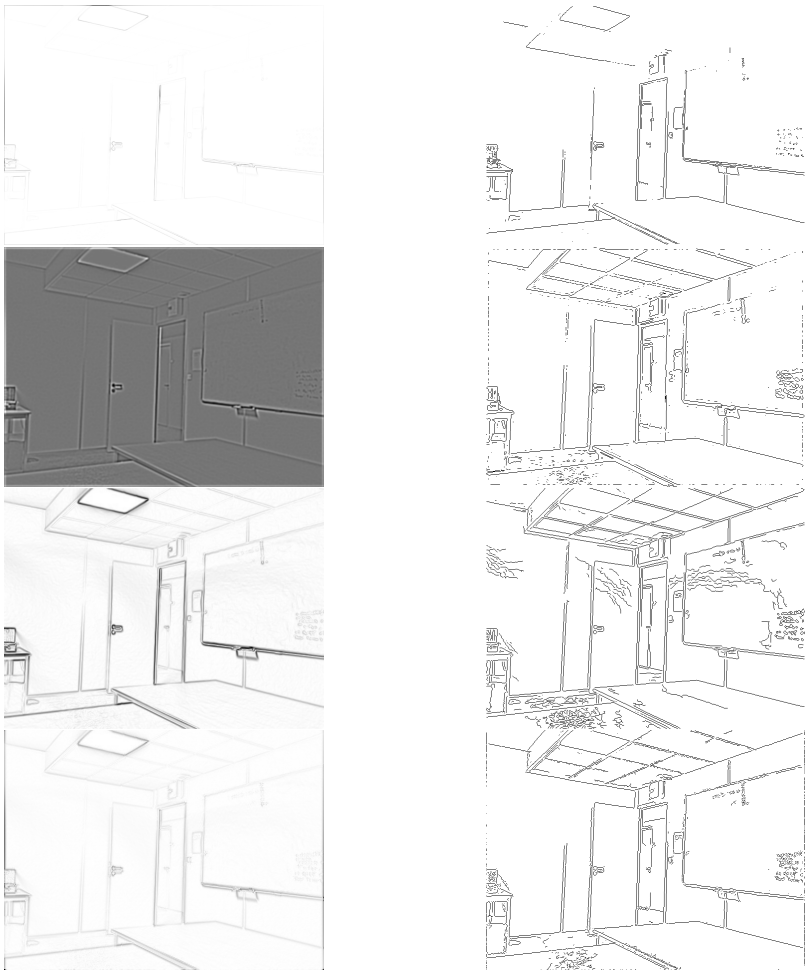

Figure 4.1: **Edge detectors - First Column:** filter response **Second Column:** Thresholded filter response (thresholds automatically estimated) **First row:** Sobel operator, which estimates the gradient of the image. Single threshold: 11.11 **Second row:** Laplacian of Gaussian operator, detection of zero crossing. Single threshold: 0.43 **Third row:** Canny operator. Hysteresis thresholding, low threshold: 0.018, high threshold: 0.046 **Fourth row:** Deriche operator. Hysteresis thresholding, low threshold: 0.012 , high threshold: 0.031

into $\iint_{D\setminus\mathcal{C}} |\nabla f|^2$, and the length $|\mathcal{C}|$ of the contours should be minimized.

This leads to the following resulting energy functional.

$$E(f,\mathcal{C}) = \iint_D (f-g)\,dxdy + \lambda^2 \iint_{D\setminus\mathcal{C}} \|\nabla f\|^2\,dxdy + \alpha\,|\mathcal{C}| \tag{4.4}$$

where λ and α are parameters that balance the influence of different terms. Several techniques have been proposed in literature to minimize the so-called Mumford-Shah energy. Andrew Blake and Andrew Zisserman describe an algorithm in [23, 24] that searches for a minimum by means of a discrete form of (Eq. 4.4) as follows.

$$E_d(u,C) = \sum_{i,j=1}^{N} (d_{ij}-u_{ij})^2 + \lambda \sum_{i,j} G_{ij}\,(1-l_{ij}) + \alpha \sum_{i,j} l_{ij} \tag{4.5}$$

where C is the discrete set of contour edges, d is the discrete image, u is the resulting image, G is the image gradient, $\cdot_{ij}$ is the index of the image pixel at coordinates i and j and l_{ij} is given by

$$l_{ij} = \begin{cases} 1 & \text{if} \quad (i,j) \in C \\ 0 & \text{if} \quad (i,j) \notin C \end{cases} \tag{4.6}$$

In order to implement this energy such that it depends only on u, the set C is clarified and a discontinuity is made explicit when the gradient is beyond a given value, leading to a new formulation.

$$E_{nd}(u) = \sum_{i,j=1}^{N} (d_{ij}-u_{ij})^2 + \sum_{i,j=1}^{N} g_{\alpha,\lambda}\,(G_{ij}) \tag{4.7}$$

where

$$g_{\alpha,\lambda}\,(G) = \begin{cases} \lambda^2 G^2 & \text{if} \quad \lambda^2 G^2 \leq \alpha \\ \alpha & \text{else} \end{cases} \tag{4.8}$$

The energy is then minimized by gradient descent with $u = d$ as initialization. The Blake-Zisserman algorithm outlined above, is actually more developed in order to overcome local minimum. Basically, they approximate $g_{\alpha,\lambda}$ to insure $E_{nd}(u)$ to be convex, and then build a series of functions $g^p_{\alpha,\lambda}$ that approach the function $g_{\alpha,\lambda}$. This is known as the *Graduated non convexity* (GNC) algorithm [23, 24].
Many other methods exist to minimize the Mumford-Shah functional energy such as Mean Field Analysis (MFA) [22], which is a faster approximation of the simulated annealing technique [112].

The *region merging* [20] and *region competition* [220] algorithms merge regions from smaller ones, given a criterion involving the uniformity of region and/or sharpness of boundaries. Note that these approaches can also be employed to minimize the Mumford-Shah energy.

Diffusion

A similar concept but with a different approach based on the well known heat diffusion is suggested by Pietro Perona and Jitendra Malik in [152]. The heat diffusion is implemented by the following partial differential equation (PDE).

$$\begin{cases} I(0) &= I_0 \\ \dfrac{\partial I}{\partial t} &= \nabla \cdot \nabla I \end{cases} \tag{4.9}$$

Perona-Malik diffusion is a non linear process build to avoid diffusing over discontinuities and is given by the following PDE.

$$\begin{cases} I(0) &= I_0 \\ \dfrac{\partial I}{\partial t} &= \operatorname{div}\left(g(\nabla I)\nabla I\right) \end{cases} \tag{4.10}$$

where $g(x)$ is a decreasing function of the gradient intensity such as g_1 or g_2: the diffusion is stopped where the gradient intensity is high.

$$g_1(x) = \frac{1}{1+\left(\dfrac{x}{K}\right)^2} \tag{4.11}$$

$$g_2(x) = \exp\left(-\left(\frac{x}{K}\right)^2\right) \tag{4.12}$$

Perona-Malik diffusion is illustrated by figure (4.2). Although the process tends to partition the image into uniform regions, diffusion over weak edges may happen.

Other approaches

Recent developments of graph-based methods include spectral methods by Jianbo Shi and Jitendra Malik [187], maximum network flow [30], semidefinite programming techniques by Jens Keuchel et al. [110], Semi supervised *graph cut* Segmentation by Yuri Boykov and Marie-Pierre Joly [29] (see also [28]), random walk

Figure 4.2: **Perona-Malik diffusion:** from left to right, original image, after 50 iterations and after 200 iterations

/ potential theory by Leo Grady [98], Minimum Ratio Cycles (MRC) on product graphs [177] by Thomas Schoenemann and Daniel Cremers. Note that shape priors are also incorporated in graph-based methods. For instance, Thomas Schoenemann and Daniel Cremers employ MRC to find a global optimal image segmentation with elastic shape prior [176], or Ali Kemal Sinop and Leo Grady use ratios to infer basic shape priors [188]

The watershed algorithm [19, 134, 135] comes from a topographical notion and was proposed in the context of image segmentation by Serge Beucher and Christian Lantuejoul in 1979 [19]. The image is seen as a topographical landscape flooded by water. The watershed is the set of point such that a drop of water can end up in two different regional minimum. It can be theoretically related to graph-cut techniques [5].

Recently, some techniques based on perception (colors, textures, Gestalt cues such as continuity etc.) have been developed to generate perceptual grouping or visually pleasing segmentation results. In [205], Zhuowen Tu and Song-Chun Zhu combine bottom up and data driven techniques with high level prior knowledge such as colors, texture and boundary continuity hypotheses. Gestalt cues, texture, brightness and good continuation are used by Xiofeng Ren and Jitendra Malik in [160]. In [109], John Kaufhold and Anthony Hoogs learn boundary edges. First, they densely segment the images and then each edge is classified into boundary / non boundary using a classifier trained on a ground true. The authors use perception hypothesis and perceptual rules during the process. In [211] segmentation is directly used for object recognition. The image is also over-segmented and a bottom up MCMC (Markov Chain Monte Carlo) mechanism is used to group regions by image information and shape similarity constraints (shape priors).

4.3 Segmentation with active contours

4.3.1 Introduction

Segmentation algorithms cited previously all have in common the fact that they are pixel-wise at local or global scale. In this section, a different approach to image segmentation is introduced. Basically, a *curve* evolves in the image domain by expansions and shrinkages according to constraints defined by the image. In the final stage of the segmentation process, the curve is supposed to delimit a region defined by the object of interest. Note that the region is not necessarily connected. These methods mostly make a curve evolve, according to a partial differential equation (PDE) as previously described in chapter 3. In the sequel, we will refer only to that kind of segmentation.

4.3.2 Active contours: *Snakes*

In [108], Michael Kass, Andrew Witkin and Demetri Terzopoulos published a seminal paper on a new approach to image segmentation called *snakes*, based on an active contour model. Let $S(q) : q \longrightarrow \mathbb{R}^2$ be a parametrized curve.

$$E_s(S) = \nu_1 \int |\frac{\partial S}{\partial q}|^2 dq + \nu_2 \int |\frac{\partial^2 S}{\partial q^2}|^2 dq - \nu_3 \int |\nabla I(S)|^2 dq \qquad (4.13)$$

The first two terms are called internal energies, i.e. they depends only on the curve. $\partial S/\partial q$ controls the elasticity or tension of the curve and make the curve behave as membrane while $\partial^2 S/\partial s^2$ controls its stiffness and make the curve behave as a thin plate. The last term is called external energy i.e. it depends on a potential scalar energy such as the gradient in (Eq. 4.13). Note that a prior term can be added to E_s. In practice, the *snakes* model consists of a set of point sampling the curve associated to a discretized version of the energy E_s. Snakes are illustrated in figure (4.3) where the internal energy enforces the snake to segment a large circle.
This method, which results in closed contour segmentation with high precision edge detection, has however important drawbacks as mentioned in chapter 3. Indeed, we recall that the shape representation

1. requires a reparametrization during the evolution, since it is based on an explicit representation (in order to have a stable evolution)

2. is difficult to extend to higher dimensions

3. cannot be easily extended to other criteria and
4. does not have an intrinsic parametrization
5. cannot *a-priori* deal with topological changes.

Concerning the last point, note that some topology adaptive meshes (*T-snakes* and *T-surface*) have been introduced in [129, 130]. *Level set* implementation mostly overcomes these limitations.

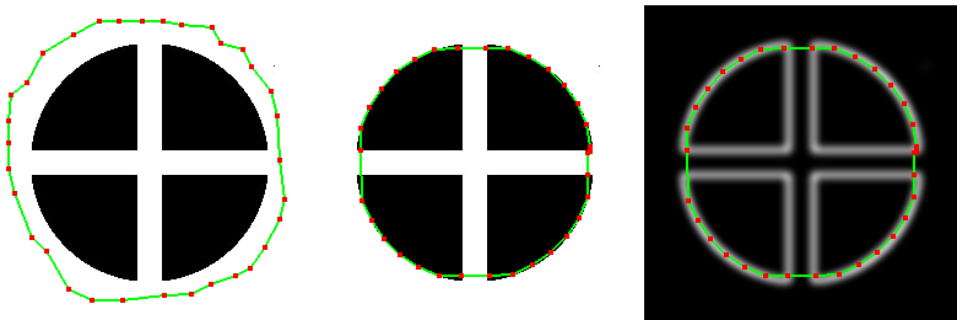

Figure 4.3: **Active snakes contours:** from left to right, initialization of a snake, after convergence (original image), after convergence (norm of intensity gradient image)

4.3.3 Within the *Level Set* framework

Introduction

The *Level set* representation has been commonly accepted and employed in the computer vision community because it has many advantages (see chapter 3) such as allowing topological changes. The methods presented in this section are all implemented within the *Level Set* framework, though other techniques could be used to minimize the proposed objective functions. *Level set* based segmentation can be classified into two categories, edges-based [43, 36, 128, 37, 111] and region-based [38, 149, 204, 214, 148, 150].

Edge-based methods

The methods in the first category rely on local filtering techniques and gradient-based operators used by edge operators. Edge-based methods have low computational costs compared to region based approaches (see further in the text). Shape

variation are also naturally handled. Finally, they are also insensitive to global illumination changes, since edge detection is based on relative illumination changes.

Geometric active contours [36], introduced by Vincent Caselles, Francine Catté, Tomeu Coll and Françoise Dibos, are described by the evolution equation (4.16) [147]. Basically, it relies on the *Mean Curvature Motion* previously mentioned (Eq. 3.21, p. 54). Contrary to the *snake* model, it is based on a curvilinear abscissa parametrization and can then be advantageously implemented within the *Level Set* framework. The model is built from the evolution equation (3.21) and is completed by a stopping function g and a constant velocity ν in order to detect objects.

$$\frac{\partial S}{\partial t} = -g(\nabla \hat{I})\,(\kappa + \nu)\,\mathbf{N} \tag{4.14}$$

where κ is the curvature and $\hat{I}$ is a smoothed version of image I. The term κ enforces the curve to shrink accordingly to the curvature while ν ensures the curve to evolve into the normal direction at a minimum constant speed and allows us to detect non-convex objects. Both terms, κ and ν, are weighted by the stopping function g which blocks the evolution along edges. g is a decreasing function of the gradient that slows or stops the evolution as the gradient become higher. Equation (4.15) is an example of stopping function.

$$g(x) = \frac{1}{1 + ||x||^n}, \quad n = 1, \ldots \tag{4.15}$$

Note the similarity with non linear diffusion presented above. See (Eq. 4.10). The *Level Set* evolution equation is then given by

$$\frac{\partial \phi(x,y)}{\partial t} = -g\left(\nabla \hat{I}(x,y)\right)\,(\kappa(\phi(x,y)) + \nu)\,|\nabla \phi(x,y)| \tag{4.16}$$

The reader is also referred to similar works [128, 43].

Latterly, Vincent Caselles, Ron Kimmel and Guillermo Sapiro present the *geodesic active contours* [37], which is similar in spirit to the *geometric active contour*. Contrary to *geometric active contour* model, the energy minimized is explicitly known and is expressed in a closed form (from the *snake* contour model). Indeed, it can be advantageously interpreted as the length of a contour in a Riemannian space with a metric induced by image intensity. Yet, when the curvilinear abscissa parametrization is chosen, the energy is expressed depending on the length $L(S)$ of the curve.

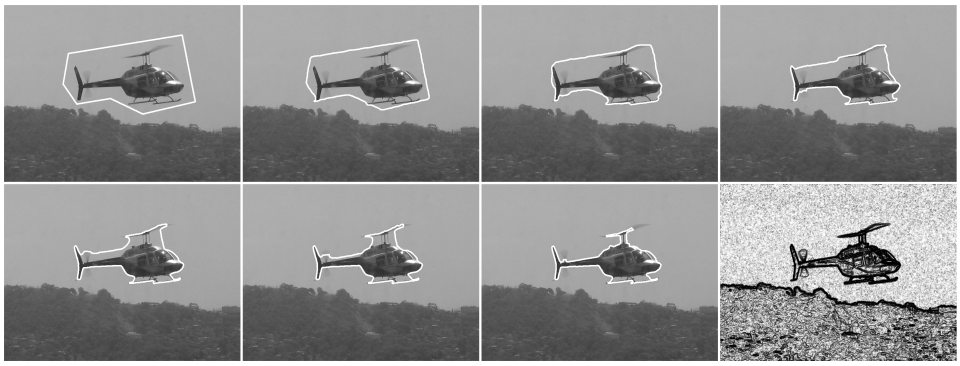

Figure 4.4: **Geometric active contours:** Evolution in the level set framework following (Eq. 4.16). Bottom right image represents the function g. Here $g(x) = 1/1 + ||x||^2$

$$E_g(S) = \int_0^{L(S)} g(\nabla \hat{I}) ds \tag{4.17}$$

where $ds = \left|\frac{\partial S(q)}{\partial q}\right| dq$ is the Euclidean metric. Equation (4.17) is compared to the length of a curve (Eq. 3.20, p. 54). The new length of the curve S is obtained by weighting the Euclidean element of length ds by $g(\nabla \hat{I})$, which contains edge information (hence the name *geodesic active contour*).
The Euler-Lagrange equation is solved to obtain the evolution equation.

$$\frac{\partial S}{\partial t} = -\Big(g(\nabla \hat{I})\kappa \mathbf{N} - (\nabla g \cdot \mathbf{N})\mathbf{N}\Big) \tag{4.18}$$

We also compare the new evolution equation (4.18) with (Eq. 4.14). In the old model (Eq. 4.14), the curve stops only when $g = 0$ which occurs only on ideal edges. In real images the gradient intensity varies along the boundaries notably when edges are badly defined and ideal edges are rare. We notice that this situation is naturally handled by the term $\nabla g \cdot \mathbf{N}$ embedded in the new model (Eq. 4.18). This term still attracts the curve toward the boundaries of the object. A second advantage of *geodesic active contours* is that the velocity term ν is removed, since extra parameters are usually unwanted properties.
Equation (4.18) is easily implemented in the Level Set framework.

$$\frac{\partial \phi(x,y)}{\partial t} = g(\nabla \hat{I}(x,y))|\nabla \phi(x,y)|\kappa(\phi(x,y)) + \nabla g(\nabla \hat{I}(x,y)) \cdot \nabla \phi(x,y) \tag{4.19}$$

The reader is also referred to [111].

In [7], Ben Appleton and Hugues Talbot introduce a graph based technique called *Globally Optimal Geodesic Active Contour* (GOGAC), that searches for minimal cycles in weighted graphs.

Region-based methods

In spite of their advantages, edges-based Level Set techniques remain very sensitive to noise and robustness to noise is hardly satisfied. To cope with such limitation, *region-based* approaches have been developed.

For example, Nikos Paragios and Rachid Deriche propose *geodesic active regions* [148, 149, 150] 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. an image histogram, within the regions partitioning the image domain. A boundary term is modeled in this framework as well.

$$\begin{aligned} E_{\text{GAR}}(S) &= \alpha \int_0^{L(S)} g\left(\mathrm{p}_S(I(S(q)))\right) ds \\ &+ (1-\alpha) \int_{\mathcal{R}_A} g\left(\mathrm{p}_A(I(x,y))\right) dxdy \\ &+ (1-\alpha) \int_{\mathcal{R}_B} g\left(\mathrm{p}_B(I(x,y))\right) dxdy \end{aligned} \tag{4.20}$$

where $\mathcal{R}_A$, $\mathcal{R}_B$ are the regions defined by the partitioning curve. $\mathrm{p}_A(I(S(q))$, $\mathrm{p}_B(I(S(q))$ are conditional intensity density functions with respect to segmentation hypotheses.The first term (boundary term) imposes regularity, minimal length of the curve and attracts toward edges. The others are region terms. The reader is also referred to [214, 204]

In [38], the Chan-Vese's model solves the minimization problem by deriving the Mumford-Shah energy in the Level Set framework and leads to the following

evolution equation.

$$\begin{aligned}\frac{\partial\phi(x,y)}{\partial t} &= \delta_\epsilon\left(\phi(x,y)\right)\Big\{\alpha\kappa(\phi(x,y)) - \nu \\ &\quad - \Big[(I(x,y)-\mu_o)^2 - (I(x,y)-\mu_b)^2\Big]\Big\}\end{aligned} \tag{4.21}$$

where δ_ϵ is a smooth approximation of the *Dirac* function.
The image is supposed to be segmented into two regions, the object and the background. Each region can be approximated as a piecewise constant intensity function with an associated intensity value: μ_o and μ_b one for each region. In other words, the object can be approximated by the mean value μ_o and the background by the mean value μ_b. The evolution seeks to minimize the difference between the mean values inside/outside the curve and the given values μ_o and μ_b.
Note that a discussion on the accuracy and limitations of edge-based approaches compared to region-based approaches can be found in Daniel Cremers' PhD dissertation [52].

A recent review on image segmentation by Daniel Cremers, Michael Rousson and Rachid Deriche is available in [56]. The reader is also referred to [119] for completeness.

4.4 Incorporating shape priors in active contours

4.4.1 Introduction

When dealing with complex images, some prior shape knowledge may be necessary to remove any ambiguity in the segmentation process. The use of such prior information in the deformable models framework has long been limited to a smoothness assumption or to simple parametric families of shapes.

4.4.2 Learning linear shape priors

A recent and important trend in this domain is the development of deformable models integrating more elaborate prior shape information. An important work in this direction is the *active shape model* by Timothy Cootes, Christopher Taylor and David Cooper [48]. This approach performs a principal component analysis

(PCA) on the position of some landmark points placed in a coherent way on all the training contours. The number of degrees of freedom of the model is reduced by considering only the principal modes of variation. The active shape model is quite general and has been successfully applied to various types of shapes (hands, faces, organs). However, the reliance on a parameterized representation and the manual positioning of the landmarks, particularly tedious in 3D images, somehow limits it applicability.

Michael Leventon, Eric Grimson and Olivier Faugeras [122] circumvent these limitations by computing parameterization-independent shape statistics within the Level Set representation [147, 185, 146]. Basically, they perform a PCA on the signed distance functions of the training shapes, and the resulting statistical model is integrated into a geodesic active contours framework. The evolution equation contains a term which attracts the model toward an optimal prior shape. The latter is a combination of the mean shape and of the principal modes of variation. The coefficients of the different modes and the pose parameters are updated by a secondary optimization process. The work by Daniel Cremers, Stanley Osher and Stefano Soatto [55] is is the first, to our knowledge, to introduce a Bayesian formulation on the space of level set functions and to model an arbitrary a priori probability distribution by kernel density estimates. Note that several improvements to such approaches have been proposed [165, 42, 203], and in particular an elegant integration of the statistical shape model into a unique MAP Bayesian optimization. The authors in [95], Muriel Gastaud and Michel Barlaud, define the shape prior as a functional of the distance between the active contour and a contour of reference.

Performing PCA on distance functions might be problematic since they do not define a vector space. To cope with this limitation, Guillaume Charpiat, Olivier Faugeras and Renaud Keriven [40] proposed shape statistics based on differentiable approximations of the Hausdorff distance. However, their work is limited to a linearized shape space with small deformation modes around a mean shape. Such an approach is relevant only when the learning set is composed of very similar shapes.

4.4.3 Non linear shape priors

In order to deal with non linear shape priors, *kernel* approaches are used to map data in higher dimensional linear *feature space*. (See chapter 5). Daniel Cremers, Tomi Kohlberger and Christoph Schnörr proposed in [53] another neat Bayesian prior shape formulation based on a B-spline representation and a probabilistic version of KPCA (Kernel Principal Component Analysis, see chapter 5). In this model, shapes are represented by a linear combination of B-splines using a set of control points z. A shape prior energy, which is calculated in the feature space from z by using KPCA-like approach (Eq. 3.3), minimizes both the *distance in feature space* and the *distance from the feature space* (that is equivalent to minimizing a reconstruction error).

Let us also mention [162, 206] which are other KPCA-based approaches in the statistical shape framework.

Chapter 5

Dimensionality reduction & manifold learning techniques

Abstract

Many problems of computer vision involve challenging data, due to their high — sometime infinite — dimensionality. In this chapter, we review common methods used to represent data in a space of reduced dimension from the well-established principal component analysis *technique (PCA) to more recent ones such as* diffusion maps *(DFM).*

In this work, we divide dimensionality reduction methods into three categories. First, we begin with a presentation of the popular PCA and its extension KPCA (Kernel PCA) to non linear data. Basically, these techniques attempt to recover orthogonal directions of largest variance, which requires a large sample set. Next, we outline some methods based on the distance between data points only. Usually, such methods are employed when the data dimension is very high compared to the number of samples in the dataset. Finally, the most recent techniques assume that data have a structure of low dimensional manifold embedded in a higher dimensional space. These methods are theoretically well-founded, and for these reasons, there is a clear trend toward their use in computer vision. Those are Laplacian-based methods *since their core relies on discrete approximations of the Laplace-Beltrami operator. Last but not least, we highlight links between the methods presented in this chapter.*

Contents

Key points & Original contributions

We present the most popular dimensionality reduction techniques: Principal Component Analysis (PCA), Kernel Principal Component Analysis (KPCA), Multi-Dimensional Scaling (MDS), Isomap, Laplacian eigenmaps (LEM) and diffusion maps (DFM).

We emphasize the links between these methods.

5.1 Maximum variance based methods

5.1.1 PCA - Principal Component Analysis

Principal component analysis (PCA) is a technique introduced by Kenneth Pearson [151] to explain the dispersion of a point cloud by projecting data onto a carefully chosen linear subspace. If, for example, the linear subspace is of dimension 1, it is chosen such that the data projection should have the maximum variance. Note that one can capture the dispersion in linear subspaces of higher dimension, by building an orthonormal basis such that the projection along each axis has a decreasing maximum variance. In other words, PCA reduces the dimensionality of the point cloud while keeping the maximum of variance information. Schematically, PCA projects data onto a lower dimensional space in order to uncorrelated them. PCA has many applications, in visualization of high dimensional data, in denoising and in classification.

We outline the PCA algorithm in $\mathbb{R}^n$ using a prediction linear model. We consider p points $\Gamma = \{\mathbf{x}_1, \ldots, \mathbf{x}_p\}$ of dimension n drawn from a random variable $\vec{X}$ and arrange these samples in a $(n \times p)$ matrix, denoted $\mathbf{X}$. We also assume that the p samples are centered and their number is far superior to the dimension n ($p >> n$). Let $\mathbf{u} = \{\mathbf{u}_1, \ldots, \mathbf{u}_m\}$ be a m dimensional orthonormal basis. Let also $\mathcal{P}(\vec{X}, \mathbf{u})$ denotes the projection of $\vec{X}$:

$$\mathcal{P}\left(\vec{X}, \mathbf{u}\right) = \sum_{k=1}^{m} \mathbf{u}_k \langle \mathbf{u}_k, \vec{X} \rangle \tag{5.1}$$

We are looking for $\mathbf{u}^*$ such that

$$\vec{X} = \mathcal{P}\left(\vec{X}, \mathbf{u}^*\right) + \vec{W} \tag{5.2}$$

where $\vec{W}$ is the residual noise. The choice of the basis $\mathbf{u}$ should minimize $\vec{W}$. This prediction model (5.2) is supported by a theorem of Konstantinos Diamantaras and Sun Yan Kung, which states that *minimizing the reconstruction* error is equivalent to *maximizing the variance* of the projection along the axis $\mathbf{u}_1, \ldots, \mathbf{u}_m$ [69]. We now detail the reconstruction error $[U_{\vec{X}}(\mathbf{u})]$ and the maximum variance $[I_{\vec{X}}(\mathbf{u})]$ below.

$$U_{\vec{X}}(\mathbf{u}) = \mathbb{E}\left\{\left\|\vec{X} - \mathcal{P}\left(\vec{X}, \mathbf{u}\right)\right\|^2\right\} \tag{5.3}$$

$$I_{\vec{X}}(\mathbf{u}) = \mathbb{E}\left\{\left[\mathcal{P}\left(\vec{X}, \mathbf{u}\right) - \mathbb{E}\left\{\mathcal{P}\left(\vec{X}, \mathbf{u}\right)\right\}\right]^2\right\} = \text{var}\left\{\mathcal{P}\left(\vec{X}, \mathbf{u}\right)\right\} \tag{5.4}$$

Yet again, PCA maximizes an objective function $I_{\vec{X}}(\mathbf{u})$ (Eq. 5.4) that describes the inertia of the point cloud projected in the subspace spanned by $\mathbf{u}$, which is equivalent to minimizing the objective function $U_{\vec{X}}(\mathbf{u})$ (Eq. 5.3):

$$\hat{\mathbf{u}} = \arg\min_{\mathbf{u}} U_{\vec{X}}(\mathbf{u}) = \arg\max_{\mathbf{u}} I_{\vec{X}}(\mathbf{u}) \tag{5.5}$$

The dataset is supposed to be centered, i.e. $\sum_{i=1}^{p} \mathbf{x}_i = 0$. Let $\mathbf{C}^{\text{PCA}} = \frac{1}{p}\mathbf{X}\mathbf{X}^T$ be the covariance matrix. To simplify the notations, we will be using matrix $\mathbf{C} = \mathbf{X}\mathbf{X}^T$ in the sequel. Since PCA searches for a basis $\mathbf{u}$ that uncorrelated the variables, we diagonalize the covariance matrix $\mathbf{C}$ (Eq. 5.6 by solving the eigenproblem (Eq. 5.7)) [69]: the basis $\mathbf{u}$ is formed by the m first eigenvectors of $\mathbf{C}$, also named the *principal axes*. Note that we will be skipping the notation $\hat{\cdot}$ in the sequel.

$$\mathbf{C} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T = \left(\mathbf{U}\sqrt{\mathbf{\Lambda}}\right)\left(\mathbf{U}\sqrt{\mathbf{\Lambda}}\right)^T \tag{5.6}$$

$$\mathbf{C}\mathbf{U} = \mathbf{U}\mathbf{\Lambda} \tag{5.7}$$

where $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_n]$ and $\mathbf{\Lambda} = \text{diag}(\lambda_1, \ldots, \lambda_n)$ are the decreasing eigenvalues corresponding to the right eigenvectors $\mathbf{u}_1, \ldots, \mathbf{u}_n$. The so-called *principal components* $\mathbf{X}^T\mathbf{U}$ correspond to the projection of the dataset onto the principal axes. We also define matrix $\mathbf{V}$ such that

$$\mathbf{V} = \mathbf{X}^T\mathbf{U}\left(\sqrt{\mathbf{\Lambda}}\right)^{-1} \tag{5.8}$$

where in detail $\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_n]$ and $\forall k = 1, \ldots, n \quad v_k = \lambda_k \mathbf{X}^T \mathbf{u}_k$. This choice will be becoming clear in the sequel since it will be highlighting the relationship between the different method presented in this chapter. Note that the principal components are calculated with a change basis through matrix $\mathbf{U}$ and, as such, PCA is a simple linear transformation. Then, we denote the mapping Φ^{PCA} that projects the p points $\mathbf{x}_1, \ldots, \mathbf{x}_p$ into a m dimensional subspace and then defines the m first *principal components* of the point cloud.

$$\begin{array}{rccl} \Phi^{\text{PCA}}: & \Gamma \subset \mathbb{R}^n & \longrightarrow & \mathbb{R}^m \\ & \mathbf{x}_i & \longmapsto & \left\{\sqrt{\lambda_1}\,(\mathbf{v}_1)_i, \ldots, \sqrt{\lambda_m}\,(\mathbf{v}_m)_i\right\} \end{array} \tag{5.9}$$

where $(\mathbf{a})_i$ denotes the i^{th} component of vector $\mathbf{a}$.

We now focus on the extension of the analysis to a given new point $\mathbf{x}$. Since PCA is a linear transformation, point $\mathbf{x}$ is straightforwardly projected onto the k^{th} principal axis $\mathbf{u}_k$ through the scalar operation $\langle \mathbf{x}, \mathbf{u}_k \rangle$, which allows us to define the mapping $\tilde{\Phi}^{\text{PCA}}$

$$\begin{array}{rccc} \tilde{\Phi}^{\text{PCA}}: & \mathbb{R}^n & \longrightarrow & \mathbb{R}^m \\ & \mathbf{x} & \longmapsto & \left(\tilde{\Phi}_1^{\text{PCA}}, \ldots, \tilde{\Phi}_m^{\text{PCA}}\right) \end{array} \tag{5.10}$$

where

$$\forall k = 1, \ldots, p \quad \tilde{\Phi}_k^{\text{PCA}}(\mathbf{x}) = \langle \mathbf{u}_k, \mathbf{x} \rangle = \left(\mathbf{U}^T \mathbf{x}\right)_k \tag{5.11}$$

We go further in the analysis because we aim to link PCA with further methods presented in this chapter. By using equations (5.7) and (5.8), we deduce that $\mathbf{XV} = \mathbf{U\Lambda}$ and

$$\mathbf{U} = \mathbf{XV}\left(\sqrt{\mathbf{\Lambda}}\right)^{-1} \tag{5.12}$$

We merge (Eq. 5.10) with (Eq. 5.12) and obtain

$$\begin{array}{rccc} \forall k = 1, \ldots, m & \tilde{\Phi}_k^{\text{PCA}}(\mathbf{x}) & = & \left(\mathbf{\Lambda}^{-1/2}\, \mathbf{V}^T \mathbf{X}^T \mathbf{x}\right)_k \\ & & = & \left(\mathbf{\Lambda}^{-1/2}\, \mathbf{V}^T \langle \mathbf{X}, \mathbf{x} \rangle\right)_k \end{array} \tag{5.13}$$

5.1.2 KPCA - Kernel Principal Component Analysis

In the previous section, PCA's goal was simply to express the data in a new orthogonal basis by means of a linear transformation. This detail is of particular importance since PCA is efficient only if the data lie on a linear subspace up to residual noise, more precisely, if the dataset is "homogeneously" distributed around a given mean. Nevertheless, data sets used in real problems mostly have non-linear features and PCA cannot be applied (as, for instance, exemplified in figure (5.2)). To cope with this situation, kernel methods first map data into a higher dimensional space –possibly an infinite dimensional space– and then force them to lie into a linear subspace. This technique relies on the well known *kernel trick*.

The kernel trick

The kernel trick is a technique which was primarily used in classification of clustered data separated with non linear boundaries. Since details of KPCA is beyond

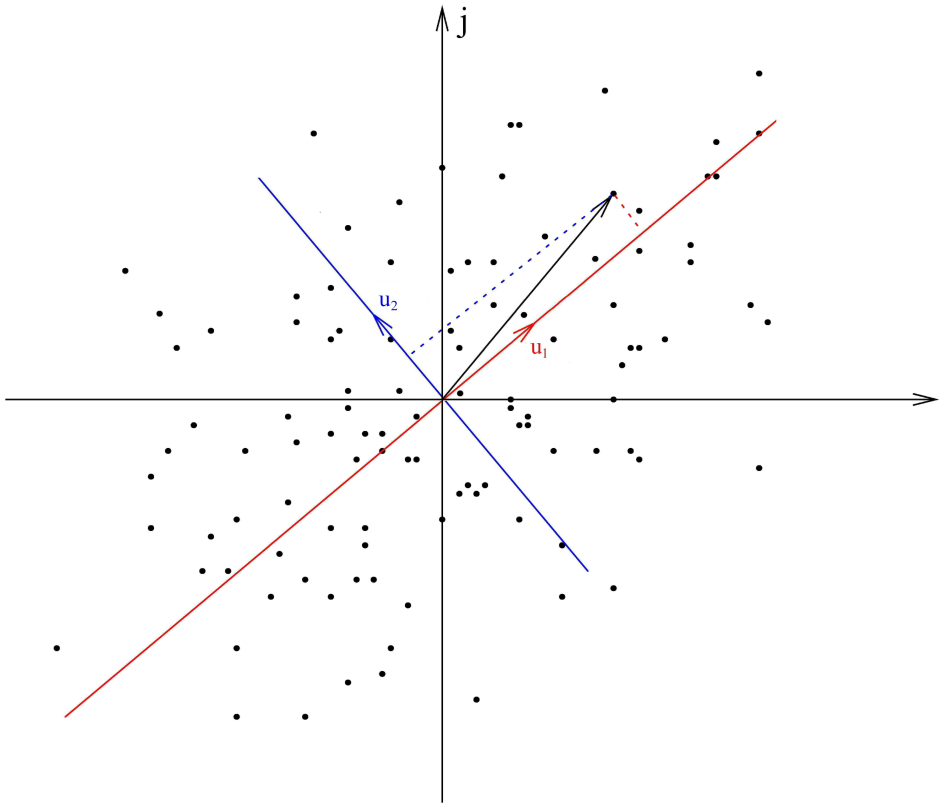

Figure 5.1: **Example of PCA in 2 dimensions**

the scope this thesis, we only outline the kernel trick using a toy example. Assume that a dataset in an input space $\mathcal{X} = \mathbb{R}^2$ is compounded of two labeled clusters and separated by a circle boundary as represented in figure 5.3. Common classification methods, such as PCA, are linear and cannot deal with this situation. However, a mapping φ into a higher dimensional Hilbert space $\mathcal{Y}$, called *feature space* in our context, can be constructed so that the data become linearly separable (see figure 5.3)

$$\begin{array}{rccc} \varphi: & \mathbb{R}^2 & \to & \mathbb{R}^3 \\ & \mathbf{x} = (a, b) & \mapsto & \left(a^2, \sqrt{2}ab, b^2\right) \end{array} \tag{5.14}$$

We denote by $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, the inner product in the feature space $\mathcal{Y}$. By doing a simple calculation, we find that it can be expressed by means of the inner product

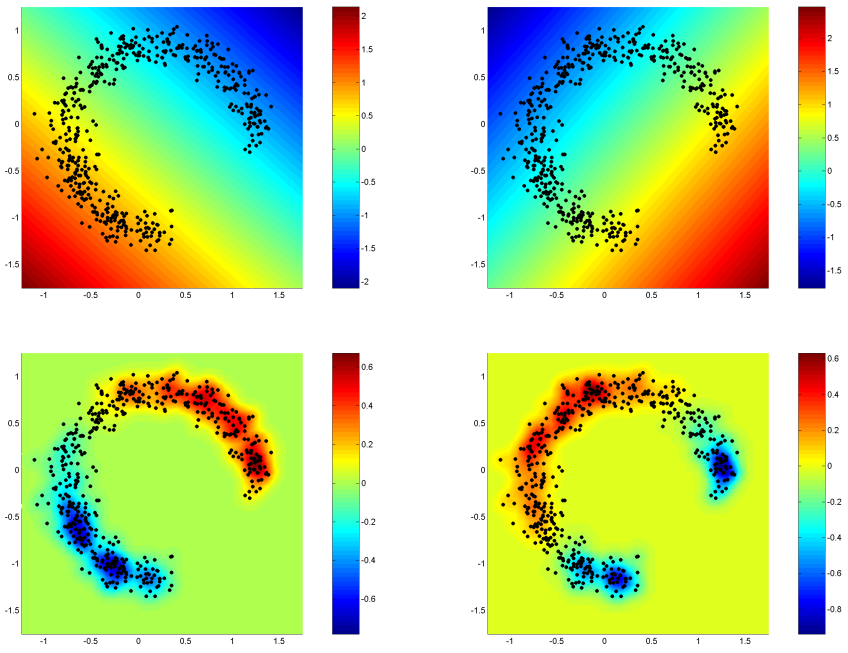

Figure 5.2: **PCA fails with "non linear data"**: Black points are the data to be analyzed. The color code represents the values of the point projection onto the principal axes. **Top:** PCA Values on the first (left) and second (right) principal axis. **Bottom:** KPCA values on the first (left) and second (right) principal axis

in the input space $\mathcal{X}$.

$$K(\mathbf{x}_i, \mathbf{x}_j) = \langle \varphi(\mathbf{x}_i), \varphi(\mathbf{x}_j) \rangle_{\mathcal{Y}} \quad (5.15)$$

$$= (\langle \mathbf{x}_i, \mathbf{x}_j \rangle)^2 \quad (5.16)$$

where $\langle \cdot, \cdot \rangle_{\mathcal{Y}}$ is the scalar product in the feature space $\mathcal{Y}$. K is called the Mercer kernel and matrix $K = (K_{i,j})_{1 \leq i,j \leq p}$ ($K_{i,j} = K(\mathbf{x}_i, \mathbf{x}_j)$) is called the Graam matrix.

The mapping φ cannot always be explicitly expressed. Nevertheless, the Mercer's theorem [133] states that: *Any symmetric positive semi-definite continuous kernel function K can be expressed as a dot product in a higher dimensional space* $K(\cdot, \cdot)$ is often the Gaussian kernel
In kernel methods, $\mathcal{Y}$ is a reproducing kernel Hilbert space (RKHS) [9] and we

have $\forall f : \mathcal{X} \longrightarrow \mathcal{Y}$

$$f(x) = \langle K(x, \cdot), f \rangle_{\mathcal{Y}} \tag{5.17}$$

and the Mercer's kernel defines an integral operator

$$(Kf)(x) = \int K(x, y) f(y) dy \tag{5.18}$$

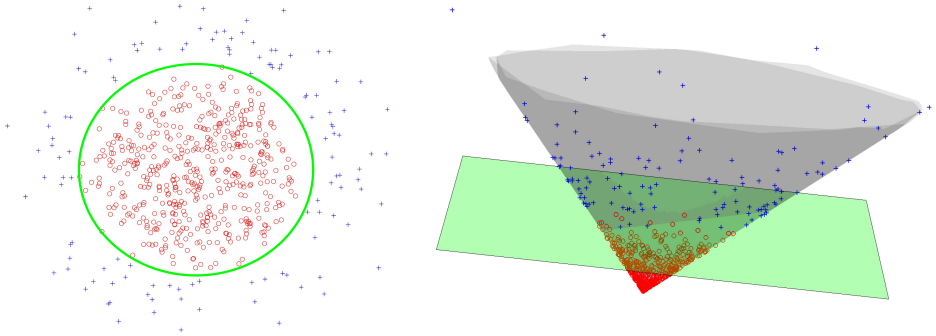

Figure 5.3: The *kernel trick*: data mapping (from left to right) following $\varphi(a, b) = \left(a^2, \sqrt{2}ab, b^2\right)$

Non-linear PCA in a higher dimensional space

The Kernel Principal Component Analysis (KPCA, or Kernel PCA) was introduced by Bernhard Schölkopf, Alexander Smola and Klaus-Robert Müller in [180] to compute a "PCA" for data lying in non-linear subspaces. It uses the kernel trick to map in a higher linear space. KPCA assumes implicitly that data have a Gaussian distribution (or at least "close to") in the feature space.

We denote $\mathbf{\Phi} = [\varphi(\mathbf{x}_1), \ldots, \varphi(\mathbf{x}_p)]$ and assume temporarily that data are centered in the feature space. Note that we can form the kernel matrix $\mathbf{K} = K(\mathbf{x}_i, \mathbf{x}_j)_{1 \leq i,j \leq p}$ such that $\mathbf{K} = \mathbf{\Phi}^T \mathbf{\Phi}$. Similarly to the linear case, the KPCA aims at computing the eigenvectors of the covariance matrix $\mathbf{C}_{\mathbf{fs}}^{\mathbf{KPCA}}$ in the feature space.

$$\mathbf{C}_{\mathbf{fs}}^{\mathbf{KPCA}} = \frac{1}{p} \sum_{i=1}^{p} \phi(\mathbf{x_i}) \phi(\mathbf{x_i})^T = \frac{1}{p} \mathbf{\Phi} \mathbf{\Phi}^T \tag{5.19}$$

in the feature space. Again, to simplify the notation, we will be using $\mathbf{C_{fs}} = \mathbf{\Phi}\mathbf{\Phi}^T$ in the sequel.
Roughly, the following derivations are similar to those of PCA, $\mathbf{X}$ being replaced by $\mathbf{\Phi}$ (we abuse somehow the matrix notation)

$$\mathbf{C_{fs}}\mathbf{U} = \mathbf{U}\mathbf{\Lambda} \tag{5.20}$$

where $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_p]$ are the eigenvectors or principal axes, and $\mathbf{\Lambda} = [\lambda_1, \ldots, \lambda_p]$ the corresponding eigenvalues. The principal components in the feature space are given by $\mathbf{\Phi}^T\mathbf{U}$. Let denote $\mathbf{V} = \mathbf{\Phi}^T\mathbf{U}\left(\sqrt{\mathbf{\Lambda}}\right)^{-1} = [\mathbf{v}_1, \ldots, \mathbf{v}_p]$.
Now, from (Eq. 5.20) we deduce the following derivation.

$$\begin{aligned}
\mathbf{\Phi}^T\mathbf{\Phi}\mathbf{\Phi}^T U &= \mathbf{\Phi}^T\mathbf{U}\mathbf{\Lambda} && (5.21)\\
\mathbf{K}\mathbf{\Phi}^T\mathbf{U}\left(\sqrt{\mathbf{\Lambda}}\right)^{-1} &= \mathbf{\Phi}^T\mathbf{U}\sqrt{\mathbf{\Lambda}} && (5.22)\\
\mathbf{K}\mathbf{V} &= \mathbf{V}\mathbf{\Lambda} && (5.23)\\
\mathbf{K} &= \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T = \left(\mathbf{V}\sqrt{\Lambda}\right)\left(\mathbf{V}\sqrt{\Lambda}\right)^T && (5.24)
\end{aligned}$$

As in PCA, we denote the mapping Φ^{KPCA} that projects the points in a m dimensional subspace and then defines the m first *principal components* in the feature space ($m \leq p$).

$$\begin{array}{rccl}
\Phi^{\text{KPCA}}: & \Gamma \subset \mathbb{R}^n & \longrightarrow & \mathbb{R}^m \\
& \mathbf{x}_i & \longmapsto & \left\{\sqrt{\lambda_1}\left(\mathbf{v}_1\right)_i, \ldots, \sqrt{\lambda_m}\left(\mathbf{v}_m\right)_i\right\}
\end{array} \tag{5.25}$$

Again, we focus on the extension of the analysis to a new given point $\mathbf{x}$ and define the mapping $\tilde{\Phi}_k^{\text{KPCA}}$ that projects point $\mathbf{x}$ onto the k^{th} principal axis $\mathbf{u}_k$ in the feature space. Note that it relies on the Mercer theorem [133] and the RKHS theory [9].

$$\begin{array}{rccl}
\tilde{\Phi}^{\text{KPCA}}: & \mathbb{R}^n & \longrightarrow & \mathbb{R}^m \\
& \mathbf{x} & \longmapsto & \left(\tilde{\Phi}_1^{\text{KPCA}}\left(\mathbf{x}\right), \ldots, \tilde{\Phi}_m^{\text{KPCA}}\left(\mathbf{x}\right)\right)
\end{array} \tag{5.26}$$

where:

$$\forall k = 1, \ldots, p \quad \tilde{\Phi}_k^{\text{KPCA}}\left(\mathbf{x}\right) = \langle \mathbf{u}_k, \varphi(\mathbf{x})\rangle_{\mathcal{Y}} = \mathbf{U}^T\varphi(\mathbf{x}) \tag{5.27}$$

Following the same path as PCA and using the above derivations we easily find that

$$\mathbf{U} = \mathbf{\Phi}\mathbf{V}\left(\sqrt{\mathbf{\Lambda}}\right)^{-1} \tag{5.28}$$

Again, by merging equations (5.29) and (5.28) we obtain

$$\tilde{\Phi}_k^{\text{KPCA}}(\mathbf{x}) = \left(\mathbf{\Lambda}^{-1/2}\,\mathbf{V}^T\mathbf{K}(\mathbf{x},\cdot)\right)_k = \left(\mathbf{\Lambda}^{-1/2}\,\mathbf{V}^T\langle\mathbf{\Phi},\mathbf{x}\rangle_{\mathcal{Y}}\right)_k \tag{5.29}$$

where $\mathbf{K}(\mathbf{x},\cdot) = \langle\mathbf{\Phi},\mathbf{x}\rangle_{\mathcal{Y}} = [K(\mathbf{x},\mathbf{x}_1),\ldots,K(\mathbf{x},\mathbf{x}_p)]^T$

Note that in the previous derivations, we assume that data were centered in the feature space. Centering data in the feature space can be achieved by using a matrix $\mathbf{H} = \mathbf{I} - p^{-1}\mathbb{1}\mathbb{1}^T$ [180] and consider a new matrix $\tilde{\mathbf{K}}$ defined as follows.

$$\tilde{\mathbf{K}} = \mathbf{HKH} = \tilde{\mathbf{U}}\tilde{\mathbf{\Lambda}}\tilde{\mathbf{U}}^{\mathbf{T}} \tag{5.30}$$

5.2 Distance-based methods

In this section, we focus on two popular methods to achieve dimensionality reduction, based on the distance between points within a dataset. The first method, the Multi-Dimensional Scaling, is a PCA-like technique used when PCA itself cannot be applied. Such a situation occurs if the covariance matrix cannot be calculated from the dataset, when data are, for instance, qualitative or infinite dimensional, when only a point-wise distance is known or also when the size p of the sample set is lower than the dimension of data. Note that we choose to present Isomap in this section as an extension of the Multi-Dimensional Scaling method.

5.2.1 MDS - Multi-Dimensional Scaling

Introduction

The Multi-Dimensional Scaling (MDS) method has roots in psychometrics; a first step toward MDS appeared in the journal *Psychometrika* in 1941 [216]. Torgeson, W.S. also proposed in [201] a MDS method to explain psychometric experiments on people's perception of the similarities inside classes of objects. MDS [51] is a technique to represent sets of general objects, equipped with a notion of comparative distances, into a low dimensional Euclidean space that best approximates the dissimilarities between the objects.

Algorithm outline

For the sake of clarity, we detail the MDS algorithm with objects represented by points in $\mathbb{R}^n$. Yet, it can be used with more general data by using an adapted similarity measure between objects. Let $\mathbf{o}_1, \ldots, \mathbf{o}_p$ be a set of p objects and $\boldsymbol{\mathcal{D}} = (\mathcal{D}_{ij})_{1 \leq i,j \leq p}$ be a matrix of squared distances: $\mathcal{D}_{ij} = d(\mathbf{o}_i, \mathbf{o}_j)^2$ is the squared distance or similarity between objects $\mathbf{o}_i$ and $\mathbf{o}_j$. A dissimilarity matrix $\mathbf{S}$ is easily obtained from the distance matrix by means of a centering matrix $\mathbf{H} = \mathbf{I} - p^{-1}\mathbb{1}\mathbb{1}^T$:

$$\mathbf{S} = \mathbf{H}\boldsymbol{\mathcal{D}}\mathbf{H} \tag{5.31}$$

The matrix $\mathbf{S}$ is proved to be an inner product matrix [201], whether the distance d is Euclidean or not. $\mathbf{S}$ is also symmetric semi-positive definite and of rank p. In other words, there exists a $(p \times p)$matrix X such that $\mathbf{S} = \mathbf{X}^T\mathbf{X} = (S_{ij})_{1 \leq i,j \leq p}$, with $S_{ij} = \mathbf{x}_i^T\mathbf{x}_j$ (Torgerson equalities, see [202]). Note that X is a centered matrix. Thus, we can write the spectral decomposition of $\mathbf{S}$ and infer the matrix $\mathbf{X}$:

$$\mathbf{S} = \mathbf{X}^T\mathbf{X} = \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^T \tag{5.32}$$

$$\mathbf{X} = \sqrt{\boldsymbol{\Lambda}}\mathbf{V}^T \tag{5.33}$$

where $\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1, \ldots, \lambda_p)$ is the diagonal matrix of eigenvalues $\lambda_1, \ldots, \lambda_p$ and $\boldsymbol{\Lambda}^{\frac{1}{2}} = \mathrm{diag}(\sqrt{\lambda_1}, \ldots, \sqrt{\lambda_p})$. As previously, we can define a mapping $\Phi^{\mathbf{MDS}}$ from the data in the original space into the reduced space of dimension m:

$$\begin{array}{rccl} \Phi^{\mathbf{MDS}}: & \Gamma \subset \mathbb{R}^n & \longrightarrow & \mathbb{R}^m \\ & \mathbf{x}_i & \longmapsto & \left\{\sqrt{\lambda_1}\,(\mathbf{v}_1)_i, \ldots, \sqrt{\lambda_m}\,(\mathbf{v}_m)_i\right\} \end{array} \tag{5.34}$$

Link with PCA

MDS is closely related to PCA. Let $\mathbf{X} = [\mathbf{x_1}, \ldots, \mathbf{x_p}]$ be p centered points in $\mathbb{R}^n$ (PCA advantageously requires a centered dataset) such that $p > n$.
$\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_p]$ are the eigenvectors (principal axes) of the covariance matrix $\mathbf{X}\mathbf{X}^T$, and $\lambda_1, \ldots, \lambda_p$ are the corresponding decreasing eigenvalues.

$$\mathbf{X}\mathbf{X}^T\mathbf{U} = \mathbf{U}\boldsymbol{\Lambda} \tag{5.35}$$

where $\boldsymbol{\Lambda} = \mathrm{diag}([\lambda_1, \ldots, \lambda_p])$.
We define $\mathbf{V} = \mathbf{X}^T\mathbf{U}\left(\sqrt{\boldsymbol{\Lambda}}\right)^{-1}$ and multiply equation (5.35) by $\mathbf{X}^T$. Thus, we

find the following derivation.

$$\begin{aligned}
\mathbf{X}^T\mathbf{X}\left(\mathbf{X}^T\mathbf{U}\right) &= \left(\mathbf{X}^T\mathbf{U}\right)\mathbf{\Lambda} && (5.36)\\
\mathbf{S}\left(\mathbf{X}^T\mathbf{U}\right) &= \left(\mathbf{X}^T\mathbf{U}\right)\mathbf{\Lambda} && (5.37)\\
\mathbf{S}\mathbf{V}\left(\sqrt{\mathbf{\Lambda}}\right)^{-1} &= \mathbf{V}\sqrt{\mathbf{\Lambda}} && (5.38)\\
\mathbf{S} &= \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T = \left(\mathbf{V}\sqrt{\mathbf{\Lambda}}\right)\left(\mathbf{V}\sqrt{\mathbf{\Lambda}}\right)^T && (5.39)
\end{aligned}$$

The relation between (Eq. 5.38) and (Eq. 5.32) is clearly highlighted.
$\mathbf{V}$ appears to be the eigenvector of the inner product matrix $\mathbf{S}$. We can then deduce straightforwardly the mapping $\tilde{\Phi}^{\text{MDS}}(\mathbf{x})$

$$\begin{array}{rccc}
\tilde{\Phi}^{\text{MDS}}: & \mathbb{R}^n & \longrightarrow & \mathbb{R}^m \\
& \mathbf{x} & \longmapsto & \left(\tilde{\Phi}_1^{\text{MDS}}(\mathbf{x}), \ldots, \tilde{\Phi}_m^{\text{MDS}}(\mathbf{x})\right)
\end{array} \qquad (5.40)$$

$\forall k = 1, \ldots, m \quad \tilde{\Phi}_k^{\text{MDS}}(\mathbf{x})$ projects a new point $\mathbf{x}$ onto the k^{th} principal axis $\mathbf{u}_k$.

$$\forall k = 1, \ldots, m \quad \tilde{\Phi}_k^{\text{MDS}} = \tilde{\Phi}_k^{\text{PCA}}(\mathbf{x}) = \left(\mathbf{\Lambda}^{-1/2}\mathbf{V}^T\langle\mathbf{X}, \mathbf{x}\rangle\right)_k \qquad (5.41)$$

Link with KPCA: MDS as a particular case of KPCA

By comparing the derivations (Eq. 5.21), (Eq. 5.22), (Eq. 5.23), (Eq. 5.24) and (Eq. 5.36), (Eq. 5.37), (Eq. 5.38), (Eq. 5.39), MDS can be straighforwardly seen as the special case of KPCA such that the mapping φ is identity.

$$\varphi(x) = x \qquad (5.42)$$

5.2.2 Isomap

MDS is limited to data lying on a linear subspace and for instance, cannot be applied to the swiss roll example as explained by figure 5.4. Joshua Tenenbaum, Vin de Silva and John Langford recently proposed in *Science* [196] an extended version of MDS, named Isomap, for data lying on smooth non-linear manifold. Note that most of recent dimensionality reduction methods (see section 5.3) rely on such assumptions. This was a major advance in the state of the art at that time.

The difference between MDS and Isomap lies in the calculation of the distance matrix. Indeed, only the distances between closed points on the manifold can be

approximated by the distance in the data space. Far apart points on the manifold can be very close in the data space and these distances should be carefully calculated (see figure 5.4). Isomap calculate approximations of geodesic path along the manifold between all pairs of points. The Isomap method can be summed up in three steps. First, we consider a distance $d(\mathbf{x}, \mathbf{y})$ in the data space, between points $\mathbf{x}$ and $\mathbf{y}$. The algorithm builds a *neighborhood graph* based on a ϵ-neighborhood or k nearest neighbor points whose weights are set to $\mathcal{D}_{ij} = d(\mathbf{x}_i, \mathbf{x}_j)$ if an edge link exists between points $\mathbf{x}_i$ and $\mathbf{x}_j$, otherwise $\mathcal{D}_{ij} = \infty$. The distance between all pair of points is then calculated using a Dijkstra-like algorithm in the graph, and thus the distance matrix $\boldsymbol{\mathcal{D}}$ can be updated. Note that if the graph is fully connected, no infinite distance should remain in matrix $\boldsymbol{\mathcal{D}}$. Finally, the MDS can be applied as previously detailed.

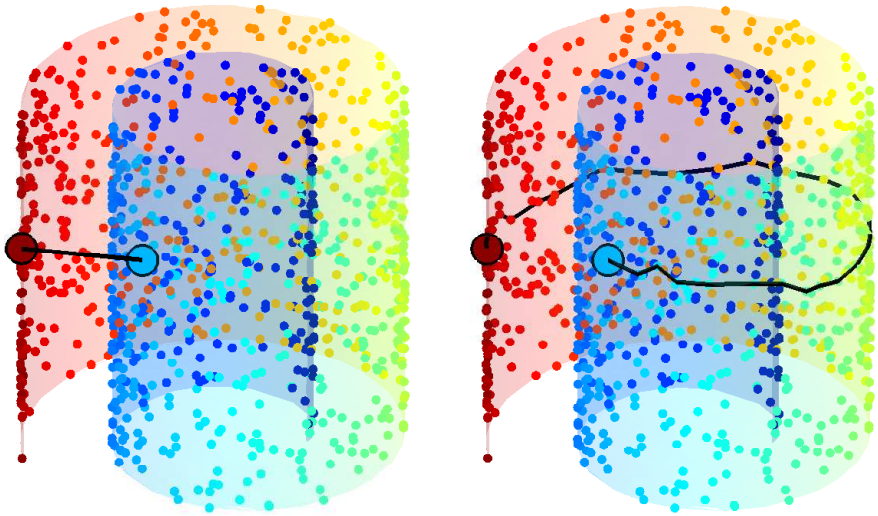

Figure 5.4: **Multi-Dimensional Scaling** - **On the left**: Distance between two points using a distance in the data space. Applying MDS with such a distance would produce unexpected results. **On the right**: Distance between two points using an approximated geodesic path.

5.3 Laplacian-based methods: Data space as a smooth manifold

Although some of the previous methods such as KPCA can recover the non-linear structure of data, they mostly do not consider explicitly that the data may lie on a low dimensional manifold (except Isomap). On the contrary, recent techniques for dimensionality reduction assume that data have a structure of low dimensional manifold embedded in a higher dimensional space. They construct maps into a low dimensional space preserving the local neighborhood topology. In the beginning of this section, we explain how this idea is related to the Laplace-Beltrami operator in the continous context. Among the most recent and popular Laplacian-based techniques are the Locally Linear Embedding (LLE) [166], Laplacian eigenmaps (LEM) [15] Locally Preserving Projections (LPP) [99], diffusion maps (DFM) [117] and maximum variance unfolding (MVU) [212]. Dimensionality reduction with minimal local distortion is achieved using spectral methods, through an analysis of the eigen-structure of some matrices derived from the adjacency graph. We give outlines of these latest techniques named *Laplacian-based methods*.

5.3.1 Dimensionality reduction and Laplace-Beltrami operator on manifolds

We denote $\mathcal{M}$ a manifold of dimension m lying in $\mathbb{R}^n$ with $n >> m$. Basically, dimensionality reduction techniques attempt to construct a mapping $f : \mathcal{M} \longrightarrow \mathbb{R}^m$ called an embedding, such that if two points x and z are *close* in $\mathcal{M}$, then also are $f(x)$ and $f(z)$. See figure 5.5. This idea can be easily expressed [15] for $m = 1$ (further generalization to $m > 1$ is actually straightforward):

$$|f(z) - f(x)| \leq \text{dist}_{\mathcal{M}}(x, z)\, ||\nabla f(x)|| + o(\text{dist}_{\mathcal{M}}(x, z)) \tag{5.43}$$

where $\text{dist}_{\mathcal{M}}(x, z)$ is the geodesic distance on the manifold between points x and z. In order to obtain a map that preserves the locality on average, $\int_{\mathcal{M}} ||\nabla f(x)||^2$ should be minimized under the constraint $||f||_{L^2(\mathcal{M})} = 1$ that avoid the trivial solution $f = 0$. We denote $\langle f, g \rangle_{\mathcal{M}} = \int_{\mathcal{M}} f \cdot g$ for any function f,g defined on $\mathcal{M}$. By using the Stoke's theorem [1]and a Lagrange multiplier μ, we rewrite the

[1]The Stoke's theorem on Riemannian manifolds results in

$$\int_{\mathcal{M}} \langle X, \nabla f \rangle = \int_{\mathcal{M}} \text{div}(X) \tag{5.44}$$

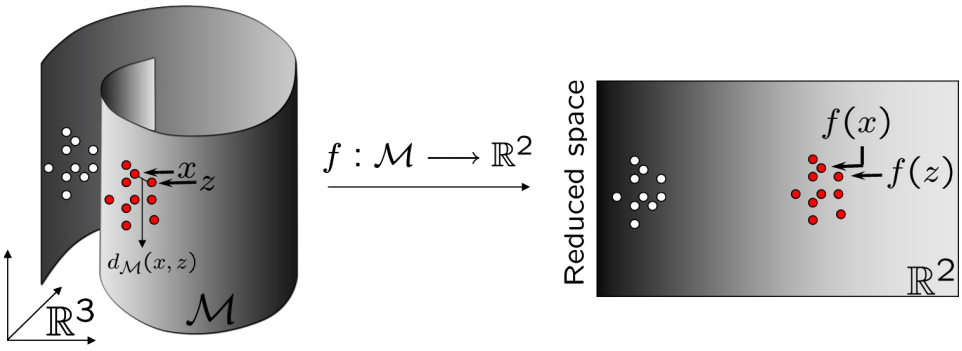

Figure 5.5: **Dimensionality reduction** - the embedding f preserves the local information of the manifold $\mathcal{M}$

unconstrained functional energy $\mathcal{G}(f)$ to be minimized:

$$\mathcal{G}(f) = \langle \mathcal{L}(f), f \rangle_{\mathcal{M}} + \mu \left(1 - \langle f, f \rangle_{\mathcal{M}}\right) \tag{5.45}$$

where $\mathcal{L}f = -\text{div}\left(\nabla f\right)$ is the Laplace-Beltrami operator. The minimization of $\mathcal{G}(f)$ is achieved when $\mathcal{L}(f) = \mu f$. Therefore, the optimal mapping f^* is given by the eigenfunction of the Laplace-Beltrami operator corresponding to the smallest non-zero eigenvalue. Note that a m dimensional embedding is usually built from the eigenfunctions corresponding to the m non-zero smallest eigenvalues

In practice, a discrete counterpart to this continuous formulation must be used since we only have access to a discrete and finite set of example data points. We will assume that this set constitutes a "good" sampling of the manifold, where "good" stands for "exhaustive" and "sufficiently dense" in a sense that will be clarified below [101].

5.3.2 Discrete Laplace-Beltrami Operator

In the previous section, we emphasized the importance of the Laplace-Beltrami operator in dimensionality reduction techniques. Its discrete counterpart is the Laplacian operator of a graph ([45]) built from p sample points $\Gamma = \{\mathbf{x}_1 \cdots \mathbf{x}_p \in \mathbb{R}^n\}$ of the m dimensional manifold $\mathcal{M}$.

In the discrete framework, we construct a neighborhood graph using the sample set $\mathbf{\Gamma} = \{\mathbf{x}_1, \ldots, \mathbf{x}_p\}$ and a decreasing function $w(\mathbf{x}_i, \mathbf{x}_j)$ of the distance between

with f a function and X a vector field

data points $\mathbf{x}_i$ and $\mathbf{x}_j$. In this work, we use the Gaussian kernel

$$w(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(\frac{-d^2(\mathbf{x}_i, \mathbf{x}_j)}{2\sigma^2}\right) \tag{5.46}$$

Let $\mathrm{NN}_{\mathbf{\Gamma}}(\mathbf{x})$ denote the nearest neighbor of point $\mathbf{x}$ in the set $\mathbf{\Gamma}$. In practice, σ is, for instance, chosen by (Eq. 5.47) or (Eq. 5.48)

$$\sigma_1 = \underset{\mathbf{x}_i \in \mathbf{\Gamma}}{\mathrm{mean}} \left\{ d\left(\mathbf{x}_i, \mathrm{NN}_{\mathbf{\Gamma}}(\mathbf{x}_i)\right)\right\} = \frac{1}{p}\sum_{i=0}^{p} d\left(\mathbf{x}_i, \mathrm{NN}_{\mathbf{\Gamma}}(\mathbf{x}_i)\right) \tag{5.47}$$

$$\sigma_2 = \underset{\mathbf{x}_i \in \mathbf{\Gamma}}{\mathrm{median}} \left\{ d\left(\mathbf{x}_i, \mathrm{NN}_{\mathbf{\Gamma}}(\mathbf{x}_i)\right)\right\} \tag{5.48}$$

The discrete formulation implies that a distance $d(\mathbf{x}_i, \mathbf{x}_j)$ is indeed defined between two data points in the ambient space.

Two slightly different approaches are to be considered to build the neighborhood graph:

ε-neighborhoods: Two nodes $\mathbf{x}_i$ and $\mathbf{x}_j$ ($i \neq j$) are connected in the graph if $d\left(\mathbf{x}_i, \mathbf{x}_j\right) < \varepsilon$, for some well-chosen constant $\varepsilon > 0$.

k nearest neighbors: Two nodes $\mathbf{x}_i$ and $\mathbf{x}_j$ are connected in the graph if node $\mathbf{x}_j$ is among the k nearest neighbors of $\mathbf{x}_i$, or conversely, for some constant integer k.

The study of advantages and disadvantages of both approaches is beyond the scope of this work. An adjacency matrix $\mathbf{W} = (W_{i,j})_{i,j\in 1,\ldots,p}$ is then constructed, the coefficients of which measure the strength of the different edges in the adjacency graph and the k nearest neighbor graph is used.

$$\begin{array}{rcll} W_{i,j} & = & w(\mathbf{x}_i, \mathbf{x}_j) & \text{if } \mathbf{x}_j \text{ is among the } k \text{ nearest neighbors of } \mathbf{x}_i \\ & = & 0 & \text{otherwise} \end{array} \tag{5.49}$$

A Laplacian matrix, a discrete form of the Laplace-Beltrami operator, is finally built from the adjacency matrix. There are however at least three ways to define the Laplacian matrix in the literature [45, 101]. Following [101], we denote the three Laplacian matrices as follows. $\mathbf{L_u}$ stands for *unnormalized*, $\mathbf{L_n}$, for *normalized*,

$\mathbf{L_r}$ for *random walk* Laplacian matrix. Finally, we detail the three forms below

$$\begin{aligned}
\mathbf{L_u} &= \mathbf{D} - \mathbf{W}, & (\mathbf{L_u} f)(i) &= d_i f(i) - \tfrac{1}{n}\textstyle\sum_{j=1}^{n} w(\mathbf{x}_i, \mathbf{x}_j) f(j) \\
\mathbf{L_n} &= \mathbf{D}^{-\frac{1}{2}}(\mathbf{D} - \mathbf{W})\mathbf{D}^{-\frac{1}{2}}, & (\mathbf{L_n} f)(i) &= f(i) - \tfrac{1}{n}\textstyle\sum_{j=1}^{n} \frac{w(\mathbf{x}_i, \mathbf{x}_j)}{\sqrt{d_i}\sqrt{d_j}} f(j) \\
\mathbf{L_r} &= \mathbf{I} - \mathbf{D}^{-1}\mathbf{W}, & (\mathbf{L_r} f)(j) &= f(i) - \tfrac{1}{d_i}\tfrac{1}{n}\textstyle\sum_{j=1}^{n} w(\mathbf{x}_i, \mathbf{x}_j) f(j)
\end{aligned}$$

where $(\mathbf{L}f)(i) = (\mathbf{L}f)(\mathbf{x}_i)$ is the value of the Laplacian at point $\mathbf{x}_i$, and $\mathbf{D} = (D_{i,j})_{1 \leq i,j \leq p}$ is a degree matrix given by equation (5.50)

$$\mathbf{D} = \mathrm{diag}(d_1, \ldots, d_p) \text{ with } d_i = D_{i,i} = \sum_{k=1}^{p} W_{ii,k} \tag{5.50}$$

$d_i = D_{ii}$ is the degree of node i and $\mathbf{D}^{-1}\mathbf{W}$ is a probability transition matrix between sample points $\mathbf{x}_1, \ldots, \mathbf{x}_p$.

5.3.3 Normalization & convergence

Nevertheless, none of these three graph Laplacian converge to the Laplace-Beltrami operator while the number of sample p increases and the size of the kernel diameter σ decreases at the same time. Instead, following Matthias Hein and Jean-Yves Audibert and Ulrike von Luxburg [101], we build three new Laplacian matrices that converge to the weighed Laplace-Beltrami operator, which is in other words "the generalization of the Laplace-Beltrami operator Δ_s[2] for a Riemannian manifold equipped with a non-uniform probability measure $\mathrm{p}(\cdot)$" [101]. In practice $\mathrm{p}(\cdot)$ represents the density of the manifold sampling ($\mathrm{p}(\cdot)$ should not be confused with the number of samples p).

$$\Delta_s = \frac{1}{\mathrm{p}^s} \mathrm{div}\,(\mathrm{p}^s\, \mathrm{grad}) \tag{5.52}$$

Now, consider that the p sample points $\mathbf{\Gamma} = \{\mathbf{x}_1 \cdots \mathbf{x}_p \in \mathbb{R}^n\}$ of the m dimensional manifold $\mathcal{M}$ are sampled under an unknown density $\mathrm{p}(\cdot)$ ($m \leq p$).

[2]Note that the following equality still applies similarly to equation (5.44):

$$\int_{\mathcal{M}} g\,((\Delta_s f)\,\mathrm{p}^s = -\int_{\mathcal{M}} \langle \nabla g, \nabla f \rangle \mathrm{p}^s \tag{5.51}$$

In order to construct an approximation of the Laplace-Beltrami operator that is independent of the unknown density $\mathrm{p}(\cdot)$, we re-normalize the adjacency matrix $(W_{i,j})$. Briefly, we form the new adjacency matrix $\widetilde{\mathbf{W}} = \left(\widetilde{W}_{i,j}\right)_{1\leq i,j\leq p}$ with $\widetilde{W}_{i,j} = \tilde{w}_\beta(\mathbf{x}_i, \mathbf{x}_j)$

$$\tilde{w}_\beta(\mathbf{x}_i, \mathbf{x}_j) \quad = \quad \frac{w(\mathbf{x}_i, \mathbf{x}_j)}{(q(\mathbf{x}_i)q(\mathbf{x}_j))^\beta} \tag{5.53}$$

where $q(\mathbf{x}) = \sum_{y\in\Gamma} w(x, y)$ is the Nadaraya-Watson estimate of the density $\mathrm{p}(\cdot)$ at location x (up to a normalization factor) [117, 101]. β and s are linked: $\beta = 1 - s/2$. The density independent Laplacian matrices $\widetilde{\mathbf{L_u}}$, $\widetilde{\mathbf{L_r}}$ and $\widetilde{\mathbf{L_r}}$ are built following (Eq. 5.50), by using the renormalized adjacency matrix. Convergence results found in [101] are reproduced below.

$$\begin{aligned}
\left(\widetilde{\mathbf{L_r}} f\right)(x) \quad &\rightsquigarrow \quad -\left(\Delta_{2(1-\beta)} f\right)(x) \\
\left(\widetilde{\mathbf{L_u}} f\right)(x) \quad &\rightsquigarrow \quad -\mathrm{p}(x)^{1-2\beta}\left(\Delta_{2(1-\beta)} f\right)(x) \\
\left(\widetilde{\mathbf{L_n}} f\right)(x) \quad &\rightsquigarrow \quad -\mathrm{p}(x)^{\frac{1}{2}-\beta}\Delta_{2(1-\beta)}\left(\frac{f}{p^{\frac{1}{2}-\beta}}\right)(x)
\end{aligned}$$

where $(\widetilde{\mathbf{L_r}} f)(x)$, $(\widetilde{\mathbf{L_u}} f)(x)$ and $(\widetilde{\mathbf{L_n}} f)(x)$ are extensions reproducing respectively $(\widetilde{\mathbf{L_r}} f)(i)$, $(\widetilde{\mathbf{L_u}} f)(i)$ and $(\widetilde{\mathbf{L_n}} f)(i)$ and $\rightsquigarrow$ stands for "*converge almost surely when the number of samples p increase and the size of the kernel decreases to* 0". The convergence results show that the parameter β plays an important role in estimating the Laplace-Beltrami operator. It determines how to calculate density independent graph Laplacian as shown by equation (5.53) and the above convergence results. Indeed, when $\beta = \frac{1}{2}$, the three graph Laplacian converge to the Laplace-Beltrami operator and are density independent. Only the normalized random walk Laplacian graph converges to the weighed Laplace-Beltrami operator for any β (since $s = 2(1-\beta)$).

5.3.4 LEM - Laplacian Eigenmaps

Laplacian eigenmaps (LEM) were introduced by Mikhail Belkin and Partha Niyogi in [15], and used in various applications of computer vision. In the continuous

framework, the solution is simply given by f^{LEM}:

$$\begin{array}{rccc} f^{\text{LEM}}: & \mathcal{M} & \longrightarrow & \mathbb{R}^m \\ & x & \longmapsto & \begin{pmatrix} f_1^*(x) \\ \cdots \\ f_m^*(x) \end{pmatrix} \end{array} \tag{5.54}$$

where f_i^* are the eigenfunctions of the Laplace-Beltrami operator.

Laplacian eigenmaps are designed to be applied to a discrete set of data. Let denote $\mathbf{y} = [y(1), \ldots, y(p)]^T$ such that $y(i) = f(\mathbf{x}_i)$ The authors minimize a discrete form of equation (5.45).

$$\mathbf{G}_{\text{LEM}}(\mathbf{y}) = \langle \mathbf{L_n y}, \mathbf{y} \rangle_M + \mu \left(1 - \langle \mathbf{y}, \mathbf{y} \rangle_M\right)$$

where $\langle \mathbf{x}, \mathbf{y} \rangle = \mathbf{y}^T \mathbf{x}$. Note that the normalized adjacency matrix $\tilde{W}$ is not used. As mentioned in the previous section, the choice of the Laplacian $\mathbf{L_n}$ is relevant only when the density $q_{\mathcal{M}}$ is uniform over the manifold. In addition, the embedding space is not equipped with an explicit metric. The Diffusion maps technique alleviates these limitations.

5.3.5 DFM - Diffusion Maps

Introduction

The major goal of Diffusion maps (DFM) is to define a metric, named the *diffusion distance*, that measures the connectivity between points in an arbitrary set. Interestingly, the diffusion distance is effectively calculated with the eigenvector of the random walk matrix or its t^{th} iterate built from the data set. The Diffusion maps technique is actually fully connected to the Laplacian eigenmaps and the Laplacian-based framework.

The DFM Kernel

The Diffusion maps technique defines an anisotropic transition matrix $\mathbf{P} = (P_{i,j})_{i,j \in 1,\ldots,p}$ such that P_{ij} is a kernel defined by

$$P_{ij} = \mathrm{p}(\mathbf{x}_j | \mathbf{x}_i) = \frac{\tilde{w}_\beta(\mathbf{x}_i, \mathbf{x}_j)}{\tilde{q}(\mathbf{x}_i)} \tag{5.55}$$

with $\tilde{q}(\mathbf{x}_i) = \sum_{\mathbf{x}_j \in \mathbf{\Gamma}} \tilde{w}_\beta(\mathbf{x}_i, \mathbf{x}_j)$
Alternatively, we have

$$\mathbf{P} = \tilde{\mathbf{D}}^{-1}\tilde{\mathbf{W}} \tag{5.56}$$

where $\tilde{\mathbf{D}} = \mathrm{diag}(\tilde{d}_1, \ldots, \tilde{d}_p)$ with $\forall i = 1, \ldots, p,\ \tilde{d}_i = \sum_{k=1}^{p} \tilde{W}_{i,k}$

For the sake of clarity, we also write $P_{ij} = \mathrm{p}(\mathbf{x}_i, \mathbf{x}_j) = \mathrm{p}(\mathbf{x}_j|\mathbf{x}_i)$. Yet again, $\mathrm{p}(\cdot\,|\cdot\,)$ (or $\mathrm{p}(\cdot\,,\cdot\,)$) denotes a transition probability between two states, and should neither be confused with the number of samples p nor with the density manifold sampling $\mathrm{p}(\cdot\,)$. The diffusion maps kernel is clearly related to the previously defined *random walk matrix* $\tilde{\mathbf{L}_r} = \mathbf{I} - \mathbf{P}$, which is a density-independent approximation of the generalized Laplace-Beltrami operator Δ_s (with $s = 2(1-\beta)$) [47, 101]. Both have the same eigenvectors and their eigen-spectrum is similar, up to an additive constant and a scalar.
We now briefly explain in two points why the diffusion distance is of particular interest.

The diffusion distance

Let $\mathrm{p_t}(\mathbf{x}_i, \mathbf{x}_j)$ denote the elements of the matrix $\mathbf{P^t}$, the t^{th} iterate of $\mathbf{P}$. We recall that $\mathrm{p_t}(\mathbf{x}_i, \mathbf{x}_j)$ represents the probability of going from $\mathbf{x}_i$ to $\mathbf{x}_j$ in t steps. The connectivity of any point $\mathbf{x}_i$ with regards to others in the graphs is characterized by the conditional probabilities $\mathrm{p_t}(\cdot|\mathbf{x}_i)$. By comparing the conditional probabilities $\mathrm{p_t}(\cdot|\mathbf{x}_i)$ and $\mathrm{p_t}(\cdot|\mathbf{x}_j)$, the diffusion distance, denoted $\mathcal{D}_t(\mathbf{x}_i, \mathbf{x}_j)$ measures the difference of connectivity between $\mathbf{x}_i$ and $\mathbf{x}_j$; it is more robust to outliers than the geodesic distances. Many distances can be used between these two probability distributions (Kullback Leibler, L^1, L^2, etc.), but a weighed L^2 is chosen by Lafon, Stéphane and Coifman, Ronald because it can be easily calculated by means of the eigenvectors of the matrix P_t, see the second point below. The reader is referred to [117] for details about the weighed L^2 distance.

Calculating the diffusion distance

Let $(\lambda_i)_{i\in 1,\ldots,p}$ with $\lambda_0 = 1 \geq \lambda_1 \geq \cdots \geq 0$ and $(\mathbf{\Psi}_i)_{i\in 1,\ldots,p}$ be respectively the eigenvalues and the associated eigenvectors of $\mathbf{P}$.

$$\mathbf{P} = \tilde{\mathbf{D}}^{-1}\tilde{\mathbf{W}} = \mathbf{\Psi\Lambda\Psi}^{-1} \tag{5.57}$$

where $\mathbf{\Psi} = [\mathbf{\Psi}_1, \ldots, \mathbf{\Psi}_p]$, $\mathbf{\Lambda} = \mathrm{diag}(\lambda_1, \ldots, \lambda_p)$ and $\forall i,\ \lambda_i = 1 - \mu_i$
The *diffusion distance* $\mathcal{D}_t^2(\mathbf{x}_i, \mathbf{x}_j)$ is expressed as an Euclidean distance in the space spanned by the eigenvectors of $\mathbf{P}$:

$$\mathcal{D}_t^2(\mathbf{x}_i, \mathbf{x}_j) = \sum_{k=1}^{p} \lambda_k^{2t} \left(\mathbf{\Psi}_k(\mathbf{x}_i) - \mathbf{\Psi}_k(\mathbf{x}_j)\right)^2 \tag{5.58}$$

where $\mathbf{\Psi}_k(\mathbf{x}_i) = (\mathbf{\Psi}_k)_i$. Furthermore, the *diffusion distance* is well approximated depending on the decay of the eigenvalues λ_k, by keeping only the first Q terms of the sum in equation (5.58). The choice of Q is also related to the dimension of the manifold $\mathcal{M}$

$$\mathcal{D}_t^2(\mathbf{x}_i, \mathbf{x}_j) \approx \sum_{k}^{Q} \lambda_k^{2t} \left(\mathbf{\Psi}_k(\mathbf{x}_i) - \mathbf{\Psi}_k(\mathbf{x}_j)\right)^2 \tag{5.59}$$

Embedding data

A mapping $\Phi^{\mathbf{DFM}}$ that embeds the data into the *Euclidean* space $\mathbb{R}^m$ *quasi isometrically*[3] with respect to a *diffusion distance* in the original space can be constructed as:

$$\begin{array}{rccc} \Phi^{\mathbf{DFM}}: & \Gamma \subset \mathcal{M} & \rightarrow & \mathbb{R}^m \\ & \mathbf{x}_i & \mapsto & \left(\lambda_1^t \left(\mathbf{\Psi}_1\right)_i, ..., \lambda_m^t \left(\mathbf{\Psi}_m\right)_i\right) \end{array} \tag{5.60}$$

Link with the continuous context

$\mathbf{\Psi}_1, \ldots \mathbf{\Psi}_m$ solve the minimization of $\mathbf{G}_{\text{DFM}}$ defined below:

$$\mathbf{G}_{\text{DFM}}(\mathbf{y}) \quad = \quad \left(\left\langle \widetilde{\mathbf{L}_r}\mathbf{y}, \mathbf{y} \right\rangle\right) + \mu \left(1 - \langle \mathbf{y}, \mathbf{y} \rangle\right) \tag{5.61}$$

In other words, the embedding in the continuous context is given as:

$$\begin{array}{rccc} f^{\text{DFM}}: & \mathcal{M} & \longrightarrow & \mathbb{R}^m \\ & x & \longmapsto & f_t^{\text{DFM}}(x) = \begin{pmatrix} (1-\lambda_1)^t\ f_1^*(x) \\ \cdots \\ (1-\lambda_m)^t\ f_m^*(x) \end{pmatrix} \end{array} \tag{5.62}$$

[3] it is an isometry when $m = p$

General remarks

Diffusion distance reflects the intrinsic geometry of the data set defined via the adjacency graph in a diffusion process (the anisotropic kernel $(P_{i,j})$ being seen as a transition matrix in a random walk process). In this formulation, t is a parameter controlling the diffusivity of the adjacency graph and can be chosen arbitrarily. We used $t = 1$ for our experiments. Diffusion distance was shown to be more robust to outliers than geodesic distances [47], thereby motivating its use to estimate the embedding (Fig. 5.6). Accordingly, in the remainder of this text, the notion of proximity in the original shape space (e.g. the "closest" neighbors of a given shape) is based on the diffusion distance. Since the embedding Φ is an isometry, proximity is advantageously deduced in the Euclidean reduced space. See (Fig. 5.6)

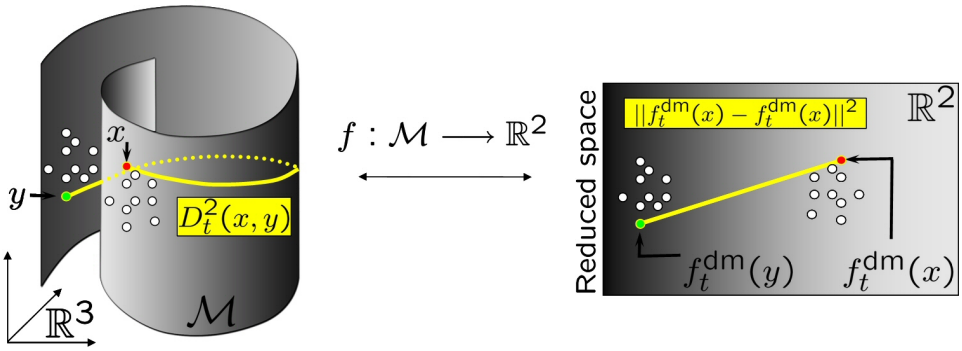

Figure 5.6: **Diffusion maps** - The diffusion distance $D_t^2(x, y)$ is approximated by the Euclidean distance in the reduced space

Link with KPCA

DFM can be related [4] to KPCA by comparing the mappings (Eq. 5.25) and (Eq. 5.60). Both techniques are very similar in spirit. Indeed, both mappings are of the

[4] For the sake of clarity, the link between KPCA and DFM is detailed only when KPCA data are supposed to be already centered in the feature space. The general case of non centered data can be processed following the same path

following form.

$$\begin{array}{rccl} \Phi^{\mathbf{GM}}: & \Gamma \subset \mathbb{R}^n & \longrightarrow & \mathbb{R}^m \\ & \mathbf{x}_i & \longmapsto & \left\{\lambda_1^t \left(\mathbf{z}_1\right)_i, \ldots, \lambda_m^t \left(\mathbf{z}_m\right)_i\right\} \end{array} \tag{5.63}$$

where $\mathbf{z}_1, \ldots, \mathbf{z}_m$ are the eigenvectors of a kernel matrix $\mathbf{M}$ such that $\mathbf{M} = \mathbf{Z}\mathbf{\Lambda}\mathbf{Z}^{-1}$, with $\mathbf{Z} = [\mathbf{z}_1, \ldots, \mathbf{z}_p]$. Now, we have:

$$\begin{array}{lllllll} \textbf{KPCA} & \textbf{case:} & \mathbf{M} & = & \mathbf{K} & = & \mathbf{W} \quad \text{with } t = 1/2 \\ \textbf{DFM} & \textbf{case:} & \mathbf{M} & = & \mathbf{P} & = & \tilde{\mathbf{D}}^{-1}\tilde{\mathbf{W}} \end{array} \tag{5.64}$$

Clearly, both techniques diagonalize the same matrix up to a normalization. Nevertheless, KPCA implicitly assumes uniform density on the manifold while most of application with real data often imply non-uniform data. KPCA depends on the sampling density on the manifold but DFM estimates its density before capturing its intrinsic geometry.

On the face of it the Diffusion maps technique does not define projections of new data onto "principal axes" as in (Eq. 5.29). However, a technique introduced by E. Nyström [145] and presented in the following chapter allows us to extend the eigenvectors of a linear operator to eigenfunction in the entire space. The Nyström extension can be proposed to calculate the embedding of a new point. In the sequel, we will highlight the link between the Nyström extension and the KPCA extension (7.1.3).

5.4 Other manifold learning methods

Many other techniques exist in literature to reduce the dimensionality of a point cloud, and most of them are related to Laplacian based methods. One can cite, for instance, Locally Linear Embedding (LLE) introduced by Sam Roweis and Lawrence Saul [166]. Note that LLE and Isomap [196] appeared in the same volume of *Science*. Briefly, LLE calculates the eigenfunction of the iterated Laplace-Beltrami operator $\mathcal{L}^2(f)$ [15], which coincide with those of $\mathcal{L}(f)$. Nevertheless, the algorithm does not search explicitly for any discrete approximation of the Laplace-Beltrami operator.
Maximum Variance Unfolding (MVU) [212] is another recent dimensionality reduction technique that makes the links with most of Laplacian-based methods such as LEM, LLE and Isomap.

One can also cite other approaches such as Tangent alignment[217] or Hessian eigenmaps [70]. The former represents tangent spaces learned by fitting an affine subspace in a neighborhood of each data point. Tangent spaces are then aligned to calculate a global parametrization of the data. Hessian eigenmaps derive from the LLE framework and is able to handle a wider class of situations, in particular when the underlying embedding is not convex. Let also mention the work on "Manifold Charting" [31] and "Minimax Embeddings" [32] proposed by Matthew Brand, the work entitled "Manifold Analysis by Topologically Constrained Isometric Embedding" [164] by Guy Rosman, Alexander Bronstein Michael Bronstein and Ron Kimmel and finally the work on "Manifold Sculpting" [94] by Mike Gashler, Dan Ventura, and Tony Martinez.

Many extensions of the techniques presented in this document can be found in the *computer vision* or *learning* literatures.

5.5 Estimating the dimension of the manifold

The dimension of a manifold is also known as its intrinsic dimension. It is simply the minimal number of parameters necessary to represent the variability of a data set. When linear methods such as PCA or MDS are used, the dimension is very often estimated by observing the eigenspectrum i.e. the spectrum of the eigenvalues output by the PCA. Indeed, the useful information is supposed to be represented in a given percentage x of the total variance. If we consider the eigenvalues $\lambda_1, \ldots, \lambda_p$ sorted in decreasing order and their corresponding eigenvectors $\mathbf{u}_1, \ldots, \mathbf{u}_p$, the dimension m^* is often estimated as follows.

$$m^* = \arg\min_m \left| \frac{\sum_{i=1}^{m} \lambda_i}{\sum_{i=1}^{p} \lambda_i} - x \right| \tag{5.65}$$

m^* is referred as the minimal number of principal axes needed to represent the data in a lower dimensional basis. Nevertheless, this method still relies on the arbitrary fixed value x.

Let also mention an important work by Thomas Minka [140] which comprehends the PCA as density estimation. The true dimensionality of the data is estimated by approximating a maximum likelihood estimator and by achieving a Bayesian

model selection.

When data lie on a smooth manifold, finding out the intrinsic dimension of the dimension estimation might become a very complex problem. One approach to estimate the dimensionality of the data is to select the dimension that minimizes a given prediction error as proposed by François Meyer and Greg Stephens in [137]. Nevertheless, the problem of estimating the dimension of a general point cloud is still an open problem in the learning community. For more details and reference, the reader is referred for example to [91, 153, 123, 100, 49].

Although existing techniques could be used to determine the dimension of our manifold, we mostly set it as a parameter. In one application on brain ventricle, we propose however a simple data dependent methodology to figure out the intrinsic dimension of the manifold (see sec. 8.3.3).

Chapter 6

Application of graph Laplacian to Interactive Image Retrieval

Abstract

Interactive image search or relevance feedback is the process which helps a user refining his query and finding difficult target categories. This consists in partially labeling a very small fraction of an image database and iteratively refining a decision rule using both the labeled and unlabeled data. Training of this decision rule is referred to as transductive learning.

*In this chapter, we present an original approach for relevance feedback based on Graph Laplacian. A modified Graph Laplacian is introduced to make a robust learning of the embedding, via diffusion maps, possible. The contribution is threefold: it allows us (**i**) to integrate all the unlabeled images in the decision process (**ii**) to robustly capture the topology of the image set and (**iii**) to perform the search process within the manifold. Relevance feedback experiments were conducted on simple databases including Olivetti and Swedish as well as challenging and large scale databases including Corel. Comparisons show clear and consistent gain, of our graph Laplacian method, over state-of-the art relevance feedback approaches.*

Contents

Original contributions

We design a *relevance feedback system* that integrates unlabeled data in the training process.

We introduce a new robust graph Laplacian operator.

Related publication: ICASSP'08 [172], CVPR'08 [173].

6.1 Introduction

At least two interrogation modes are known in content based image retrieval (CBIR); the query by example and relevance feedback (RF). In the first mode the user submits a query image as an example of his "class of interest" and the system displays the closest image(s) using a feature space and a suitable metric [27, 189]. A slight variant is category retrieval which consists in displaying images belonging to the "class of the query". In the second category (see the pioneering works [114, 155]) the user labels a subset of images as positive and/or negative according to an unknown metric defined in "his mind" and the CBIR system refines a metric and/or a decision rule and displays another set of images hopefully closing the gap between the user's intention and the response(s) of the CBIR system [218, 50, 168, 219]. This process is repeated until the system converges to the user's class of interest. The performance of an RF system is usually measured as the expectation of the number of user's responses (or iterations) necessary to focus on the targeted class. This performance depends on the capacity of an RF system (i) to generalize well on the set of unlabeled images using the labeled ones, (ii) to ask the most informative questions to the user (see for instance [200]) and (iii) the consistency and self-consistency of the user's responses. Points (i)–(ii) are respectively referred to as *the transduction* and the *display models*. Point (iii) assumes that different users have statistically the same answers according to an existing but unknown model referred to as the *user model*.

6.1.1 Related Work

Different schemes exist in literature for the purpose of RF [168, 219], which are either based on

- **density estimation** [131, 104] when they model the distribution and the positive / the (possibly) negative labeled images or on
- **discriminative training** [200], when they build a decision function which classifies the unlabeled data.

In the first category, different density estimation methods are used in RF including non parametric Parzen windows [131], Gaussian mixture models [50], logistic regression [34] and novelty detectors [41, 179]. In [50, 83], the authors introduced the notion of relative judgment of the user, i.e. the response is not binary but a

relative number measuring the relevance of a displayed set of images. The user's response is assumed as a sigmoid function of the distance, so images close to the highly numbered set are more likely to be the target than the others. The authors in [50] used Gaussian mixture models and a Bayesian framework in order to estimate (and update) a distribution through all images and display those with the highest probability. The proposed approach in [83] defines a criterion based on the mutual information between the user's responses and all the possible target images in the database and display those which maximize this criterion.

In the second family, discriminative methods learn from the aggregated set of positive and negative labeled images how to classify the unlabeled ones. Existing RF methods use support vector machines [200, 200, 87], decision trees [127], boosting [199] and Bayesian classifiers [87, 209, 50]. The RF method in [200] is of particular interest due to its important gain in the convergence speed when using active learning [178, 12].

6.1.2 Motivation and Contribution

The success of relevance feedback is largely dependent on how much **(1)** the image description (feature+similarity) fits **(2)** the semantic wanted by the user. The gap between **(1)** and **(2)** is referred to as *the semantic gap*. The reduction of this gap basically requires adapting the decision rule (as discussed earlier) and the features to the user's feedback. Many works (see, for instance [168]) consider features as a weighted combination of simple sub-features, each sub-feature capturing a particular characteristic. The weight of each sub-feature and hence the structure of the underlying embedding is adapted by taking into account the variance of the labeled set, so relevance feedback will pay more attention to the sub-features with high variances. Put differently, adapting features might be explicitly achieved as in [168] or implicitly as a part of the decision rule training (as discussed in Section 6.1.1).

When the original sub-features are highly correlated, it is difficult to find dimensions, in the original feature space, which are clearly discriminant according to the user's feedback. This happens when the Gaussian assumption (about the distribution of the data) does not hold, when the data in original feature space form a non-linear manifold, which implies that the classes become extremely not separa-

ble (see Figure 6.1, on the top). Therefore, further processing is required in order to extract dimensions with high intrinsic variances. A didactic example, shown in Figure (6.1), (the application is the search of faces by identity), follows the statement in [2]: *the variance due to the intra-class variability (pose, illumination, etc.) is larger than the inter-class variability (identity)*. Figure (6.1) illustrates this principle where clearly the intra-class variance estimated through the original feature space (resp. the intrinsic dimensions of the manifold enclosing the data) is larger (resp. smaller) than the inter-class variance. *Clearly, searching those faces through the intrinsic dimensions of the manifold is easier than in the original space.* Hence, learning the manifold enclosing the data is crucial in order to capture the *actual* structure of the data.

In this chapter, we introduce a new relevance feedback scheme based on graph Laplacian[16]. We first model the topology of the image database, including the unlabeled images, using an eigen approximation of the graph Laplacian, then we propagate the labels by projecting the whole dataset using a linear operator learned on both the labeled and the unlabeled sets. The main contributions of this work are:

- In contrast to existing relevance feedback methods which only rely on the labeled set of images, our approach integrates the unlabeled data in the training process through the cluster assumption [181] (as discussed in Section 6.3.1). These unlabeled data turn out to be very useful when only few labeled images are available since it allows us to favor decision boundaries located in low density regions of the image database, which are often encountered in practice. Although the approach proved to work in the particular task of relevance feedback, it can be easily extended to other transductive learning tasks such as database categorization.

- In the second main contribution of this work, we derive a new operator from the graph Laplacian which makes it possible to embed the dataset in *a robust way*. This graph Laplacian, based on diffusion map, captures the conditional probabilities of transition from one sample to another with a path of given length. Its main particularity is to consider the intermediate paths with high transition likelihoods only (see Section 6.3.2).

- For numerical and practical matters, we show in Section (6.4) an extension

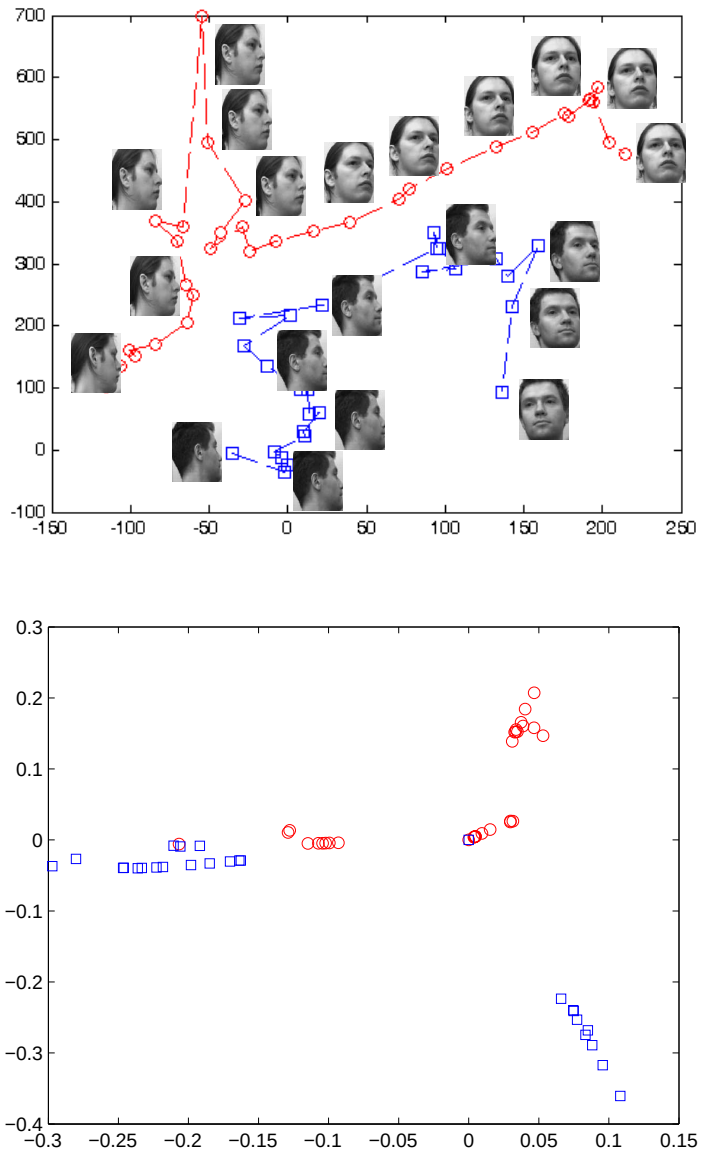

Figure 6.1: **Top: the distribution of two classes corresponding to two individuals**. It is clear that the intra class variance is larger than the inter class one. **Bottom: the distribution of the same classes inside the manifold trained using graph Laplacian.** It is clear that the converse is now true and the classification task is easier in the embedding space.

of the method in order to handle large scale databases using Nyström interpolation.

In the remainder of this chapter, we consider the following notation. $\mathbf{z}$ is a random variable standing for a training sample taken from $\mathcal{Z}$ and $\mathbf{q}$ its class label in $\{+1, -1\}$ ($\mathbf{q} = 1$ if the sample $\mathbf{z}$ belongs to the targeted class and -1 otherwise). $G = \langle V, E \rangle$ denotes a graph where V is a set of vertices and E are weighted edges. We use also l, t as indices for iterations. Among terminologies a *display* is a set of images taken from the database which are shown to the user at iteration t. The chapter is organized as follows: Section 6.2 introduces the overall architecture of the RF process. Section 6.3 describes our RF model based on the s-weighted robust graph Laplacian and the display model. Section 6.4 provides an extension of the embedding method in order to handle large scale databases which are very often encountered in practice, using the Nyström operator. Section 6.5 provides an extensive experimental study using different databases including specific ones; face databases and also generic databases. We discuss the method and conclude in Section 6.6.

6.2 Overview of the Search Process

Let $\mathcal{P} = \{\mathbf{z}_1, ..., \mathbf{z}_p\}$, $\mathcal{Q} = \{\mathbf{q}_1, ..., \mathbf{q}_p\}$ denote respectively a training set of images and the underlying unknown ground truth. Here $\mathbf{q}_i$ is equal $+1$ if the image $\mathbf{z}_i$ belongs to the user's "class of interest" and $\mathbf{q}_i = -1$ otherwise. Let us consider $\mathcal{D}_t \subset \mathcal{P}$ as a display shown at iteration t and $\mathcal{Q}_t$ the labels of $\mathcal{D}_t$. Our interaction consists in asking the user questions such that his/her responses make it possible to reduce the *semantic gap* according to the following steps:

- "Page Zero": Select a display $\mathcal{D}_1$ which might be a random set of images or the prototypes found after applying clustering or Voronoi subdivision.
- Reduce the "semantic gap" iteratively ($t = 1$,..., T):

  (1) Label the set $\mathcal{D}_t$ using a (possibly stochastic) *known-only-by-the-user* function $\mathcal{Q}_t \leftarrow \mathcal{L}(\mathcal{D}_t)$. Here $\mathcal{L}$ is referred to as the user model which, given a display $\mathcal{D}_t$, provides the labels $\mathcal{Q}_t$. When the ground truth is unique, this function is consistent (through different users) and self-consistent (with

respect to the same user) so the user's answer is coherent and objective, otherwise the labeling function becomes stochastic. The coherence issue is not in the scope of this chapter (see [83] for a comprehensive study), so we only consider consistent and self-consistent users.

(2) Train a decision function $g_t : \mathcal{Z} \rightarrow \{-1, +1\}$ on the (so far) labeled training set $\mathcal{T}_t = \bigcup_{l=1}^{t}(\mathcal{D}_l, \mathcal{Q}_l)$ and the unlabeled set of images $\mathcal{P} - \cup_{l=1}^{t}\mathcal{D}_l$. The transduction model discussed in (6.3.1) is the one used for this training. At iteration t, the target is to efficiently use both labeled and unlabeled data in order to estimate the actual decision function,

$$\underset{f:\mathcal{Z}\rightarrow\{+1;-1\}}{\operatorname{argmin}} \; P\left[g\left(\mathbf{z}\right) \neq \mathbf{q}\right]. \tag{6.1}$$

where $P[x]$ is the probability that x happens. In our setting, it is important to generalize well even when the size of the labeled training set is small. This is why this step should use transductive methods which implicitly assume that the topology of the decision boundary depends on the unlabeled set $\mathcal{P} - \cup_{k=1}^{t}\mathcal{D}_k$ as shown in (6.3). More precisely, the clustering assumption implicitly made is the following: the decision boundary is likely to be in low density regions of the input space $\mathcal{Z}$ [144].

(3) Select the next display $\mathcal{D}_{t+1} \subset \mathcal{P} - \bigcup_{k=1}^{t}\mathcal{D}_k$. The convergence of the RF model to the actual decision boundary is very dependent on the amount of information provided by the user. As $P(.)$ is unknown and the whole process is computationally expensive, the display model considers a sampling strategy which selects a collection of images that improves our current estimate of the "class of interest" (see Section 6.3.3). This can be achieved by showing samples of difficult-to-classify images such as those close to the decision boundary. Given the labeled set $\mathcal{T}_t$, and let $g_{\mathcal{D}}$ be a classifier trained on $\mathcal{T}_t$ and a display $\mathcal{D}$. The issue of selecting $\mathcal{D}_{t+1}$ can be formulated at iteration $t+1$ as:

$$\begin{aligned} &\mathcal{D}_{t+1} \leftarrow \underset{\mathcal{D}}{\operatorname{argmin}} \; P\left[g_{\mathcal{D}}(\mathbf{z}) \neq \mathbf{q}\right] \\ &s.t. \quad \mathcal{D}_{t+1} \bigcap \left(\bigcup_{l=1}^{t}\mathcal{D}_l\right) = \emptyset \end{aligned} \tag{6.2}$$

6.3 Graph Laplacian and Relevance Feedback

Graph Laplacian methods emerged recently as one of the most successful in transductive inference [16], (spectral) clustering [192] and dimensionality reduction [15]. The underlying assumption is: *the probability distribution generating the (input) data admits a density with respect to the canonical measure on a sub-manifold of the Euclidean input space*. Let $\mathcal{M}$ denotes this sub-manifold and p the probability distribution of the input space with respect to the canonical measure on $\mathcal{M}$ (i.e. the one associated with the natural volume element dV). Note that $\mathcal{M}$ can be all the Euclidean space (or a subset of it of the same dimension) so that p can simply be viewed as a density with respect to the Lebesgue measure on the Euclidean space.

6.3.1 s-Weighted Transductive Learner

In transductive inference, one searches for a smooth function $g : \mathcal{Z} \rightarrow \mathcal{Q}$ from the input feature space into the output space such that $g(\mathbf{z}_i)$ is close to the associated output $\mathbf{q}_i$ on the training set and such that the function is allowed to vary only on low density regions of the input space. Let $s \geq 0$ be a parameter characterizing how low the density should be to allow large variations of g (see (6.3)). Depending on the confidence we assign to the training outputs, we obtain the following optimization problem:

$$\min_f \sum_{i=1}^{p} c_i \left[\mathbf{q}_i - g(\mathbf{z}_i)\right]^2 + \int_{\mathcal{M}} \|\nabla g\|^2 \, \mathrm{p}^s \, dV, \tag{6.3}$$

where the c_i's are positive coefficients measuring how much we want to fit the training point $(\mathbf{z}_i, \mathbf{q}_i)$. We recognize the generalized Laplace Beltrami operator in the second term. Please note the difference between p the size of the training set and $\mathrm{p}(\cdot)$ the density probability on the manifold.) Typically, $c_i = +\infty$ imposes a hard constraint on the function g so that $g(\mathbf{z}_i) = \mathbf{q}_i$. By the law of large numbers, the integral in (6.3) can then be approximated by

$$\frac{1}{n} \sum_{i=1}^{n} g(\mathbf{z}_i) \Delta_s g(\mathbf{z}_i) \mathrm{p}^{s-1}(\mathbf{z}_i). \tag{6.4}$$

Unfortunately, the direct computation of $\Delta_s g(\mathbf{z}_i)$ for every possible function g is not possible and solving (6.3) is intractable. The discrete approximation of the s-th

weighted Laplacian operator, is an alternative to this problem. See chapter 5 for details. Accordingly, we solve its discrete counterpart:

$$\min_{\mathbf{F}\in\mathbb{R}^n} \ (\mathbf{F}-\mathbf{Q})^t\mathbf{C}(\mathbf{F}-\mathbf{Q}) + \mathbf{F}^T\dot{\mathbf{L}}\mathbf{F},$$

whose solutions are those of the linear system

$$(\dot{\mathbf{L}} + \mathbf{C})\mathbf{F} = \mathbf{C}\mathbf{Q}. \tag{6.5}$$

where $\dot{\mathbf{L}} = \widetilde{\mathbf{D}}^{\mathbf{s-1}}\widetilde{\mathbf{L}}_{\mathbf{r}}$ in view of (6.4). Here $\mathbf{C}$ is the diagonal $p \times p$ matrix for which the i-th diagonal element is c_i for a labeled point, and 0 for an unlabeled point, and similarly, $\mathbf{Q}$ is the p-dimensional vector for which the i-th element is $\mathbf{q}_i$ for a labeled point, and 0 for an unlabeled point.

6.3.2 A Robust k-step random walk matrix

As previously mentioned, we focus on the random walk matrix $\mathbf{P}$ instead of the graph Laplacian matrix $\mathbf{I} - \mathbf{P}$ since they have the same eigenvectors and the same eigenvalue up to a scalar and a constant value. When embedding a dataset using the one step random walk matrix $\mathbf{P}$, the main drawback is its sensitivity to noise. This comes from "short-cuts", when building the adjacency graph (or estimating the scale parameter of the Gaussian kernel). In such cases, the *actual* topology of the manifold $\mathcal{M}$ is lost (see. Figure 6.2, top). In [118], the authors consider instead an iterated random walk matrix $\mathbf{P}^{(k)}$ (denoted also $\mathbf{P}_k$), here $\mathbf{P}_k = \mathbf{P}_{k-1} \times \mathbf{P}$. The latter models a Markovian process where the conditional k-step transition likelihood (between two data $\mathbf{z}_i$ and $\mathbf{z}_j$) is the sum of the conditional likelihoods of all the possible (k-1)-steps linking $\mathbf{z}_i$ and $\mathbf{z}_j$. This results in low transition probabilities in low density areas. Nevertheless, when those areas are noisy, the method fails to capture the correct topology (cf. Figure 6.2, middle).

A k-step random walk matrix: the above limitation motivates the introduction of a new (called robust) graph Laplacian, recursively defined as

$$\mathbf{P}_k \ = \ [\mathbf{P}_{k-1}^{\frac{1}{\alpha}} \times \mathbf{P}^{\frac{1}{\alpha}}]^{\alpha}, \quad 1/\alpha \in [1, +\infty[\tag{6.6}$$

Let $P(i,j)^{\frac{1}{\alpha}}$ denotes the j^{th} column of the i^{th} row of $\mathbf{P}^{\frac{1}{\alpha}}$. Again, $\mathbf{P}$ is the one step random walk matrix where each entry $P(i,j)$ corresponds to the probability of a walk from $\mathbf{z}_i$ to $\mathbf{z}_j$ in one step, also denoted $P_1(j|i)$. This quantity characterizes

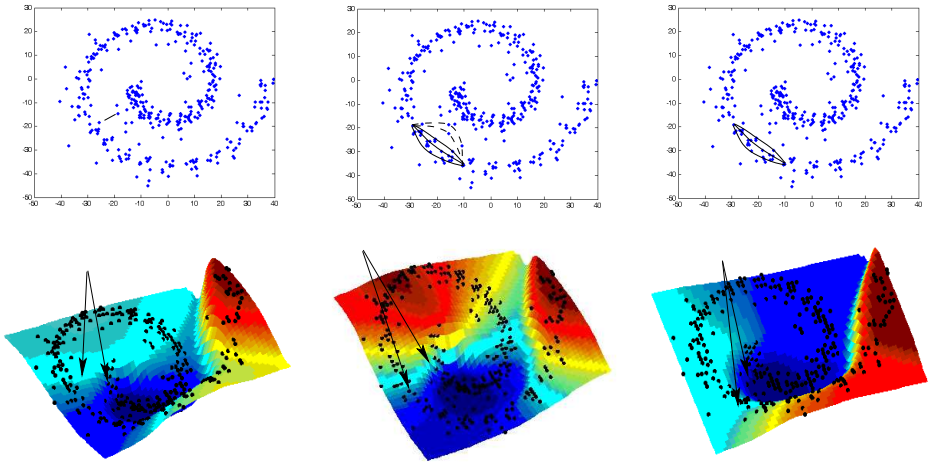

Figure 6.2: **Top: samples taken from the Swiss roll**. **On the left: A short cut makes the random walk Laplacian embedding very noise sensitive**, clearly the variation of the color map does not follow the intrinsic dimension of the actual manifold. **In the middle: when using the diffusion map, noisy paths affect the estimation of the conditional probabilities**. **On the right: when using the robust diffusion map, the color map varies following the intrinsic dimension.**

the first order neighborhood structure of the graph G. In the context of diffusion maps [118], the idea is to represent higher order neighborhood by taking powers of the matrix $\mathbf{P}$, so $P_k(i,j) = P_k(j|i)$ is the probability of a walk from $\mathbf{z}_i$ to $\mathbf{z}_j$ in k steps. Here k acts as a scale factor and makes it possible to increase the local influence of each node in the graph G. Matrix $\mathbf{P}_k$ can be inferred from matrix $\mathbf{P}_{k-1}$ and matrix $\mathbf{P}$ by summing the conditional probabilities over different paths, i.e.

$$[P_k(j|i)]^{\frac{1}{\alpha}} = \sum_{l=1}^{n} [P_{k-1}(l|i)]^{\frac{1}{\alpha}} \, [P_1(j|l)]^{\frac{1}{\alpha}} \tag{6.7}$$

We refer to a k-path as any path of k steps in the graph G. Depending on α the general form of the random walk matrix P_k implements the following random walks:

- $\alpha \to 1$: $[P_k(j|i)]^1$ is *the average* transition probability of the k-paths linking

$\mathbf{z}_i$ to $\mathbf{z}_j$. So P_k implements exactly the one in [118].

- $\alpha \rightarrow 0$: it is easy to see that $[P_k(j|i)]^{\frac{1}{\alpha}}$ converges to $\max_l \left\{ [P_{k-1}(l|i)]^{\frac{1}{\alpha}}, [P(j|l)]^{\frac{1}{\alpha}} \right\}$, so $P_k(i,j)$ corresponds to *the most likely* transition probability of k-steps.

- $\alpha \in]0,1[$: $[P_k(j|i)]^{\frac{1}{\alpha}}$ is dominated by the largest terms in $\left\{ [P_{k-1}(l|i)]^{\frac{1}{\alpha}}, [P(j|l)]^{\frac{1}{\alpha}} \right\}$. The effect of noisy terms is then reduced.

Figure (6.2, on the left) shows the application of (6.6) to the embedding of the Swiss roll data ($k = 10$ and $\alpha = 0.2$). Clearly, the topology of the data is now preserved. Figure (6.3) shows the robustness of the method to different amount of noise (again $k = 10$ and $\alpha = 0.2$).

6.3.3 Display Model

The data in $\mathcal{P}$ are mapped into a manifold $\mathcal{M}$ such that any two elements $\mathbf{z}_i$ and $\mathbf{z}_j$ in $\mathcal{P}$ with close conditional probabilities $\{P_k(i|.)\}$ and $\{P_k(j|.)\}$ will also be close in $\mathcal{M}$. We use the mapping defined in equation 5.60. The diffusion distance plays a key role in propagating the labels from the labeled to unlabeled data following the shortest path or the average path (depending on the setting of α).

We define a probabilistic framework which, given a subset of displayed images $\mathcal{D}_1$,...,$\mathcal{D}_t$ until iteration t, makes it possible to explore the manifold $\mathcal{M}$ in order to propose a subset of images $\mathcal{D}_{t+1}$. When we use the unlabeled data by using a transductive algorithm, the heuristics still rely on the following basic assumption: at each iteration, one can select the display in order to refine the current estimate of the decision boundary or one can select the display in order to find uncharted territories in which the actual decision boundary is present. The first display strategy *exploits* our knowledge of the likely position of the decision boundary while the second one *explores* new regions. We believe that any good CBIR system should find the correct balance between exploration and exploitation.

Exploitation: let $\mathcal{D} \subset \mathcal{P}$ and $\mathcal{D}' = \{\mathbf{z} \in \mathcal{D}, g_t(\mathbf{z}) > 0\}$, (6.2) is equivalent to :

$$D_{t+1} \leftarrow \arg\max_{\mathcal{D}'} P(\mathcal{D}' \mid \mathcal{D}_t, ..., \mathcal{D}_1) \tag{6.8}$$

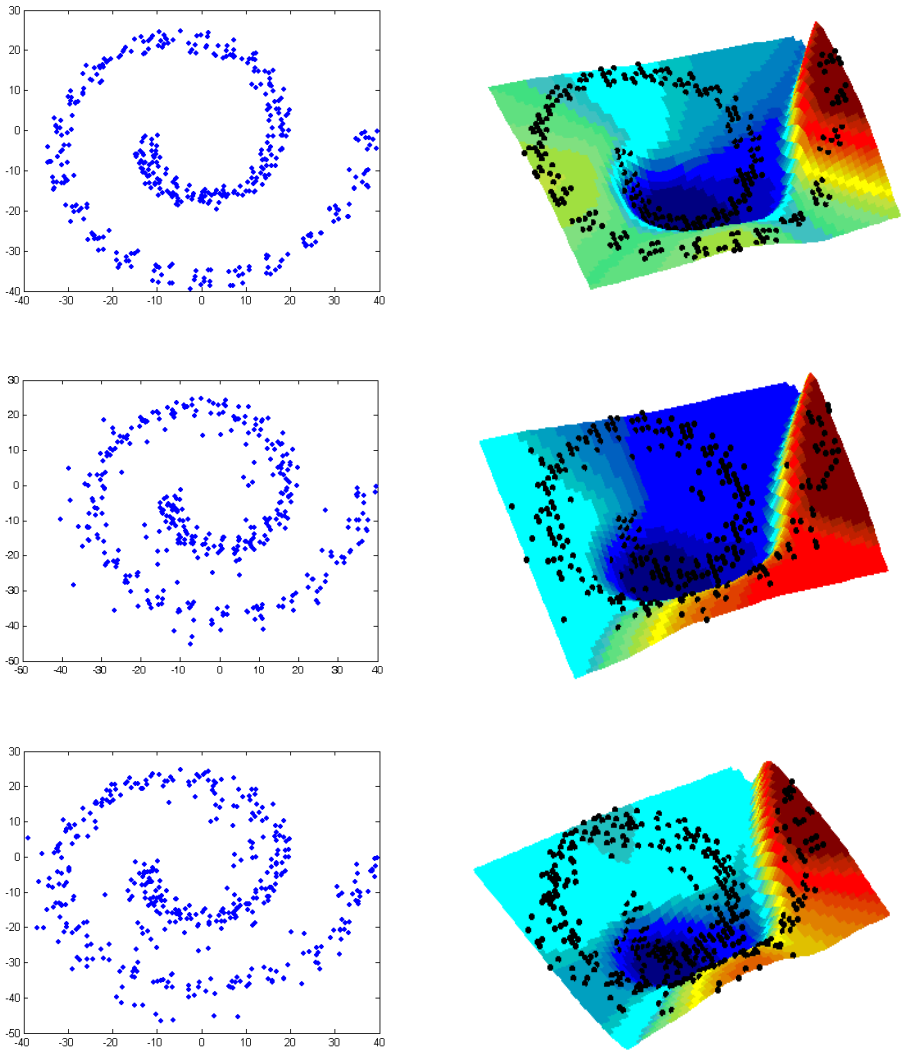

Figure 6.3: **Robustness of the embedding** with respect to uniform noise throughout the curvilinear abscissa of the Swiss roll. From top to bottom, the noise is 0%, 15% and 40%.

Assuming the data in $\mathcal{D}_{t+1}$ are chosen independently :

$$\begin{aligned} P(\mathcal{D}' \mid \mathcal{D}_t, ..., \mathcal{D}_1) &= \prod_{\mathbf{z}_j \in \mathcal{D}'} P(\mathbf{z}_j \mid \mathcal{D}_t, ..., \mathcal{D}_1) \\ P(\mathbf{z}_j \mid \mathcal{D}_t, ..., \mathcal{D}_1) &\propto \max_{\substack{\mathbf{z}_i \in \mathcal{T}_t \\ \mathbf{q}_i = +1}} \frac{1/D_{\mathcal{M}}(\mathbf{z}_i, \mathbf{z}_j)}{\sum_l 1/D_{\mathcal{M}}(\mathbf{z}_i, \mathbf{z}_l)}, \end{aligned} \tag{6.9}$$

Exploration: equivalently, the criteria is similar to (6.8) but:

$$P(\mathbf{z}_j \mid \mathcal{D}_t, ..., \mathcal{D}_1) \propto \min_{\substack{\mathbf{z}_i \in \mathcal{T}_t \\ \mathbf{q}_i = +1}} \frac{1/D_{\mathcal{M}}(\mathbf{z}_i, \mathbf{z}_j)}{\sum_l 1/D_{\mathcal{M}}(\mathbf{z}_i, \mathbf{z}_l)}, \tag{6.10}$$

We consider in this work a mixture between the two above strategies where at each iteration t of the interaction process, half of the display (of size 8 in practice) is taken from exploitation and the other set taken from exploration.

6.4 Nyström Extension

Relevance feedback usually involves databases ranging from many thousands to millions of images. The complexity of solving $\mathbf{P}_k = \mathbf{\Psi}^T \mathbf{\Lambda} \mathbf{\Psi}$ grows in $O(p^3)$ and on those databases, the problem gets quickly out of hand. For instance, for the Corel database ($p = 9.000$) it took about 15 hours to solve the eigenproblem on a standard 64 bits AMD processor of 1.8 GHz. Clearly this limits the applicability of the method for large scale databases.

Consider $\mathcal{P}' = \{\mathbf{z}_i\}_1^{p'}$ as a subset of $\mathcal{P}$ ($p' \ll p$), p' is chosen so that the above eigenproblem is numerically tractable. The Nyström extension (see details in section 7.1.3) will then be applied in order to extend the eigen-solution on the whole set $\mathcal{P}$:

$$\tilde{\Phi}_k^{\mathbf{DFM}}(\mathbf{z}) = \lambda \sum_{i=1}^{p'} \mathrm{p}(\mathbf{z}, \mathbf{z}_i) \mathbf{\Psi}_k(\mathbf{z}_i) \tag{6.11}$$

In order to show the precision of (6.11), we randomly select $\mathcal{P}'$ from Corel (see Section 6.5) with different sizes $p' = 500$, 1.000, 2.000 and 3.000. For a fixed p', we consider 15 different sampling of $\mathcal{P}$, and for each one, we estimate the embedding of $\mathcal{P}'$ using graph Laplacian and we extend on both $\mathcal{P}'$, $\mathcal{P} \setminus \mathcal{P}'$ using the Nyström interpolation. The results reported in Figure (6.4), report two measures of error:

1. The curve in green shows the expectations of the interpolation error between (i) graph Laplacian embedding and (ii) Nystroḿ interpolation, both on $\mathcal{P}'$.

2. The curve in blue shows the same measures but on $\mathcal{P} \setminus \mathcal{P}'$.

In both (1) and (2) the two errors decrease as n' increases and asymptotically converge to the same curve. This clearly corroborates the theoretical statement in [210], proving that the eigenvector-expansion of the Graph Laplacian converges to the eigenfunctions of the Laplace-Beltrami operator.

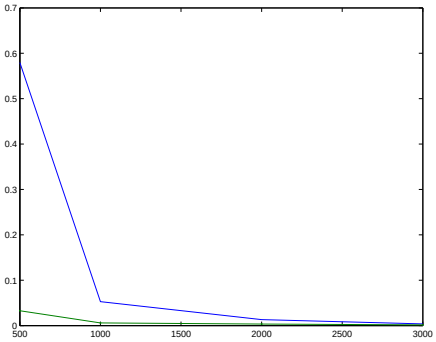

Figure 6.4: **Means of interpolation error using the Nyström extension**. In green: errors on $\mathcal{P}'$. In blue: errors on $\mathcal{P} \setminus \mathcal{P}'$.

6.5 Performances

In this section, we demonstrate the validity of relevance feedback using our graph Laplacian. We compare it to popular state-of-the-art methods including support vector machines, Bayesian inference and closely related methods, i.e. graph-cuts. The effectiveness is measured by the expected number of images per class which are displayed to the user or equivalently the average number of iterations necessary in order to show a fraction of images per class.

6.5.1 Databases

Experiments were conducted on simple databases (Olivetti and Swedish) as well as more difficult ones (Corel). The Olivetti face database contains 40 people each

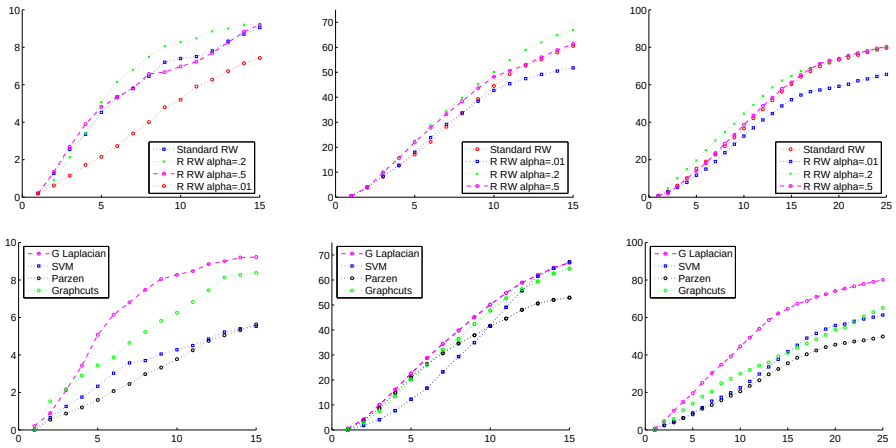

Figure 6.5: **On the top:** These figures show the **recall** for Orl, Swedish and Corel databases for different graph Laplacians. **On the Bottom: Comparison of Graph Laplacian** with respect to SVM, Parzen and Graph-cuts.

one of them being represented by 10 faces. Each face is processed using histogram equalization and encoded using kernel principal component analysis (KPCA) resulting into 20 coefficients. The Swedish set contains 15 categories of leaf silhouettes each one represented by 75 contours. Each contour $\mathcal{C}$ is encoded using 14 coefficients corresponding to the eigenvalues of KPCA on $\mathcal{C}$ [169]. The Corel database contains 90 categories each one represented by 100 images. This database is generic and images range from simple objects to natural scenes with complex background. Each image in this database is encoded simply using a $3D$ RGB color histogram of 125 dimensions. In this dataset, the classes are spread out so the relevance feedback task is more challenging. For all those databases the ground truth is provided.

6.5.2 Benchmarking

We evaluate the performance of our RF scheme using the recall that we define in the sequel. Let R_t be a random variable standing for the total number of relevant images returned by the CBIR system until iteration t, i.e., those belonging to the user's "class of interest". The recall is defined as $E(R_t) = \sum_r rP(R_t = r)$, here the randomness and the expectation of R_t is to be taken through different classes of

interest. Figures (6.5, top) show the recall for different graph Laplacians including the standard random walk (RW) and the robust random walk (R RW) for different values of α. The recall reported for the three databases (ORL, Swedish and Corel) show clearly that when $\alpha \ll 1$ (in practice $\alpha = .5$ and $\alpha = .2$), the embedding generated using the graph Laplacian is more robust and captures the topology of the data better, and hence the performance follow. Nevertheless, when $\alpha \rightarrow 0$ (in practice $\alpha = .01$), the performances degrade as the underlying graph Laplacian implements the most likely path which is more noise-sensitive (see Section 6.3.2). In all the experiments, the path length k is chosen large enough in order to make the approach robust to noise. In practice and after cross validation, we set $k = 10$.

6.5.3 Comparison

We compared our method to standard representative relevance feedback tools including inductive methods: support vector machines (SVMs), Bayesian inference (based on Parzen windows) and transductive one such as Graph cuts [170]. In all these methods, we use the same display strategy (i.e., combined exploration exploitation). We train the SVMs and Parzen classifiers using the triangular kernel since the extensive study in [87] showed that SVM based relevance feedback using the triangular kernel achieved far better results than other kernels. Thus, we limit our comparison to SVM and Parzen using this kernel only. Again, for graph Laplacian, the scale parameter of the Gaussian kernel is set as $\sigma = E_{\mathbf{z},\mathbf{z}' \in \mathcal{N}_m(\mathbf{z})}\{\|\mathbf{z} - \mathbf{z}'\|\}$, here $\mathcal{N}_m(\mathbf{z})$ denotes the set of m nearest neighbors of $\mathbf{z}$ (in practice $m = 10$). The results reported in Figure (6.5, bottom), show that in almost all cases, the recall performances of relevance feedback (using graph-Laplacian) are better than SVMs, Parzen and Graph cuts based RF. Clearly, the use of unlabeled data as a part of transductive learning (in graph Laplacian and graph cuts), makes it possible to improve the performance substantially. Furthermore, the embedding of the data through graph Laplacian makes it possible to capture the topology of the data, so learning the decision rule becomes easier.

6.6 Conclusion

In this work, we introduced an original approach for relevance feedback based on transductive learning using graph Laplacian. This work clearly demonstrates the advantages of semi supervised learning: it is effective in the sense that it **(1)** handles

transductive learning (in contrast to inductive learning), via the robust s-weighted graph Laplacian which implements the clustering assumption and uses the unlabeled data as a part of the training process **(2)** it captures the topology of the data so that the similarity measure and the propagation of the labels to unlabeled data is done through the manifold enclosing the data **(3)** it achieves a clear and consistent improvement with respect to the most powerful and used techniques in relevance feedback including SVMs, Parzen windows and graph cuts. We also showed the efficiency of this approach to handle large scale databases using the Nyström extension. The experiments reported in this chapter confirm the superiority of our approach.

Chapter 7

Dealing with new points and *attracting forces* of manifolds

Abstract

We introduce techniques which are natural extensions of manifold learning methods previously presented in this dissertation. Indeed, manifold learning is achieved given a collection of data and may imply large computational costs to embed data in a low dimensional representation. Thus, while new points are to be processed on line, it is extremely desirable to calculate the embedding of these points only, not starting from scratch with the entire collection of data plus the new points. This chapter tackles such extensions and further proposes different projection operators of data onto manifolds, which are the core of our contribution. Most importantly, this chapter provides a sound operator that attracts data toward the manifold, and that draws the contours of a new shape prior term used for image segmentation in chapter 8.

Contents

Original contributions

We introduce two extensions of the embedding to new points in the case of diffusion maps: a naive embedding regularization and the well known Nyström extension.

We design 3 *attracting forces* that attract points toward the manifold.

1. *Closest projection force*. **Related publication:** SSVM'07 [75]
2. *Same embedding force*. **Related publication:** ICIP'07 [76]
3. *Constant embedding force*. **Related publications:** ICCV'07 [81], MICCAI'07 [80]

7.1 Out-of-sample problem

7.1.1 Introduction

The mapping Φ^{DFM} (or Φ^{LEM}) often denoted Φ in the sequel is only defined on the training samples. In this approach we propose two techniques to calculate the embedding of a new point $\mathbf{x}$, given the embedding of the points $\mathbf{x}_1, \ldots, \mathbf{x}_p$. The first technique calculates a function regression in the discrete embedding space. The second one, the Nyström method, is a popular technique employed to extend empirical functions from the training set $\mathbf{\Gamma} = \{\mathbf{x}_1, \ldots, \mathbf{x}_p\}$ to new samples, i.e. the *out of sample problem* [17, 8]. Furthermore, a link with the KPCA extension (Eq. 5.29) can be pointed out.

7.1.2 First approach: embedding regularization

Let $\mathbf{\Psi}$ be a $p \times m$ matrix of which the column vectors are $\mathbf{\Psi}_1, \ldots, \mathbf{\Psi}_m$, the m eigenvectors of matrix $\mathbf{P}$, the matrix built from the DFM kernel (Eq. 5.56). Let us also denote $\mathcal{M}$ the analyzed manifold. Without loss of generality, we have from equation (5.61) and a slight generalization to m dimensions:

$$\mathbf{\Psi} = \arg\min_{\mathbf{Y}} \boldsymbol{G'}(\mathbf{Y}) \tag{7.1}$$

$$\boldsymbol{G'}(\mathbf{Y}) = \text{Tr}\left(\langle \mathbf{PY}, \mathbf{Y} \rangle\right) + \mathbf{\Lambda}\left(\mathbf{I} - \langle \mathbf{Y}, \mathbf{Y} \rangle\right) \tag{7.2}$$

where $\mathbf{Y}$ is a $p \times m$ matrix and $\mathbf{\Lambda}$ is the diagonal matrix: $\mathbf{\Lambda} = \text{diag}(\lambda_1, \ldots, \lambda_m)$.

Our goal is now to calculate the embedding of a new point $\mathbf{x}$, also denoted $\mathbf{x}_{p+1}$, and some properties are required. First, $\mathbf{x}$ may not belong to the manifold ($\mathbf{x} \notin \mathcal{M}$) and so, the relative embedding values of points $\mathbf{x}_1, \ldots, \mathbf{x}_p$ previously computed should not change. Then, even if $\mathbf{x}$ belongs to the manifold, it is neither desirable nor efficient to calculate again the embedding from points $\mathbf{x}_1, \ldots, \mathbf{x}_{p+1}$. Let $\text{p}(\cdot, \mathbf{x})$ and $\text{p}(\mathbf{x}, \cdot)$ be respectively defined by

$$\text{p}(\cdot, \mathbf{x}) = [\text{p}(\mathbf{x}_1, \mathbf{x}), \cdots, \text{p}(\mathbf{x}_p, \mathbf{x})]^T \tag{7.3}$$

$$\text{p}(\mathbf{x}, \cdot) = [\text{p}(\mathbf{x}, \mathbf{x}_1), \cdots, \text{p}(\mathbf{x}, \mathbf{x}_p)]^T \tag{7.4}$$

where $\text{p}(\mathbf{x}_i, \mathbf{x})_{i=1,\ldots,p}$ and $\text{p}(\mathbf{x}, \mathbf{x}_i)_{i=1,\ldots,p}$ can be easily deduced from equation

(5.55):

$$\begin{aligned} \forall\, i = 1, \ldots, p, \quad \mathrm{p}(\mathbf{x}_i, \mathbf{x}) &= \frac{w(\mathbf{x}_i, \mathbf{x})}{\sum\limits_{b \in \mathbf{\Gamma}} \frac{q(\mathbf{x})}{q(\mathbf{x}_b)}\, w(\mathbf{x}_i, \mathbf{x}_b)} \\ \forall\, i = 1, \ldots, p, \quad \mathrm{p}(\mathbf{x}, \mathbf{x}_i) &= \frac{w(\mathbf{x}, \mathbf{x}_i)}{\sum\limits_{b \in \mathbf{\Gamma}} \frac{q(\mathbf{x}_i)}{q(\mathbf{x}_b)}\, w(\mathbf{x}, \mathbf{x}_b)} \end{aligned} \tag{7.5}$$

We finally define $\mathbf{P}_{\text{NEW}}$ as:

$$\mathbf{P}_{\text{NEW}} = \begin{bmatrix} \dot{\mathbf{P}} & \mathrm{p}(\cdot, \mathbf{x}) \\ \mathrm{p}(\mathbf{x}, \cdot)^T & \mathrm{p}(\mathbf{x}, \mathbf{x}) \end{bmatrix} \tag{7.6}$$

Following equation (7.1), the unconstrained energy to minimize can then be written:

$$\mathbf{G}''(\mathbf{Z}) = Tr\left(\langle \mathbf{P}_{\text{NEW}} \mathbf{Z}, \mathbf{Z} \rangle\right) + \mathbf{\Lambda}\left(\mathbf{I} - \langle \mathbf{Z}, \mathbf{Z} \rangle\right) \tag{7.7}$$

Since the embeddings of the p points $\mathbf{x}_1, \ldots, \mathbf{x}_p$ are fixed, we add the following constrained problem:

$$\mathbf{z}^*_{p+1} = \arg \min_{\mathbf{z}_{p+1}} \mathbf{G}''(\mathbf{Z}) \quad \text{s.t.} \quad \mathbf{Z} = \begin{bmatrix} \mathbf{\Psi} \\ \mathbf{z}_{p+1} \end{bmatrix} \tag{7.8}$$

Deriving equation (7.8) leads to the mapping $\hat{\boldsymbol{z}} : \mathbb{R}^n \longrightarrow \mathbb{R}^m$

$$\hat{\mathbf{z}}(\mathbf{x}) = \frac{\left[\mathrm{p}(\cdot, \mathbf{x}) + \mathrm{p}(\mathbf{x}, \cdot)\right]^T \mathbf{\Psi}}{2\mathrm{p}(\mathbf{x}, \mathbf{x})} \tag{7.9}$$

The results pointed out in equation (7.9) are of particular interest since the solution is expressed by means of the Nadaraya-Watson estimator widely used in the statistical learning literature. The function $\hat{\mathbf{z}}(\mathbf{x})$ can be seen as a regression function estimating the continuous embedding. However, all the embedding is regularized, including the values $\hat{\mathbf{z}}_1, \ldots, \hat{\mathbf{z}}_p$.

7.1.3 Nyström Extensions

The Nyström extension is a popular method that consists in extending the eigenvectors of an operator to all the space. Noticing that every training sample verifies:

$$\forall \mathbf{x}_i \in \mathbf{\Gamma} \quad \forall k \in 1, \ldots, p \quad \sum_{\mathbf{x}_j \in \mathbf{\Gamma}} \mathrm{p}(\mathbf{x}_i, \mathbf{x}_j) \mathbf{\Psi}_k(\mathbf{x}_j) = \lambda_k \mathbf{\Psi}_k(\mathbf{x}_i),$$

the embedding of new data points located outside the set $\mathbf{\Gamma}$ can similarly be computed by a smooth extension $\tilde{\Phi}^{\text{DFM}}$ of Φ

$$\begin{array}{rccc} \tilde{\Phi}^{\text{DFM}}: & \mathbb{R}^n & \rightarrow & \mathbb{R}^m \\ & \mathbf{x} & \mapsto & (\tilde{\Phi}_1(\mathbf{x}), \ldots, \tilde{\Phi}_m(\mathbf{x})) \end{array} \tag{7.10}$$

where

$$\begin{array}{rcl} \forall k \in 1, \ldots, m \quad \tilde{\Phi}_k^{\text{DFM}}(\mathbf{x}) & = & \lambda_k^{t-1} \sum_{\mathbf{x}_j \in \mathbf{\Gamma}} \mathrm{p}(x, \mathbf{x}_j) \mathbf{\Psi}_k(\mathbf{x}_j) \\ & = & \left(\mathbf{\Lambda}^{t-1} \mathbf{\Psi}^T \mathbf{P}\left(\mathbf{x}, \cdot\right)\right)_k \end{array} \tag{7.11}$$

with $\mathbf{P}(\mathbf{x}, \cdot) = [\mathrm{p}(\mathbf{x}, \mathbf{x}_1), \ldots, \mathrm{p}(\mathbf{x}, \mathbf{x}_p)]^T$.

The Nyström extension happens to be more interesting than the previous approach regularizing the embedding. First, it avoids the regularization of the embedding values, as in the previous approach. Then, following section (5.3.5), we emphasis the links between the Nyström extension (Eq. 7.11) and the KPCA projection (Eq. 5.29). First, we recall and complete the link between KPCA and DFM introduced in section (5.3.5). We compare the mappings (5.25) (KPCA) and (5.60) (DFM) and notice that they are similar in spirit. Indeed, both mappings are of the following form.

$$\begin{array}{rccc} \Phi^{\text{GM}}: & \mathbf{\Gamma} \subset \mathbb{R}^n & \longrightarrow & \mathbb{R}^m \\ & \mathbf{x}_i & \longmapsto & \left\{\lambda_1^t \left(\mathbf{z}_1\right)_i, \ldots, \lambda_m^t \left(\mathbf{z}_m\right)_i\right\} \end{array} \tag{7.12}$$

where $\mathbf{z}_1, \ldots, \mathbf{z}_m$ are the eigenvectors of a kernel matrix $\mathbf{M}$ such that $\mathbf{M} = \mathbf{Z}\mathbf{\Lambda}\mathbf{Z}^{-1}$, with $\mathbf{Z} = [\mathbf{z}_1, \ldots, \mathbf{z}_p]$. Now, we have:

$$\begin{array}{lllllll} \textbf{KPCA} & \textbf{case:} & \mathbf{M} & = & \mathbf{K} & = & \mathbf{W} \qquad \text{with } t = 1/2 \\ \textbf{DFM} & \textbf{case:} & \mathbf{M} & = & \mathbf{P} & = & \tilde{\mathbf{D}}^{-1}\tilde{\mathbf{W}} \end{array} \tag{7.13}$$

We also compare the projection of a new point $\mathbf{x}$ onto principal axes in KPCA and the Nyström extension used with DFM. Both approaches are similar in spirit as well ((Eq. 5.29) and (Eq. 7.11)):

$$\forall k = 1, \ldots, m \quad \Phi_k^{\text{GM}}(\mathbf{x}) = \left(\mathbf{\Lambda}^{t-1} \mathbf{Z}^T \mathbf{M}\left(\mathbf{x}, \cdot\right)\right)_k \tag{7.14}$$

where $M(\mathbf{x}, \cdot) = [M(\mathbf{x}, \mathbf{x}_1), \ldots, M(\mathbf{x}, \mathbf{x}_p)]$ Now, we have:

$$\begin{array}{llllllllll} \textbf{KPCA / Projection} & \textbf{case:} & \mathbf{Z} & = & \mathbf{V} & \& & \mathbf{M}\left(\mathbf{x}, \cdot\right) & = & \mathbf{K}\left(\mathbf{x}, \cdot\right) \\ \textbf{DFM / Nyström} & \textbf{case:} & \mathbf{Z} & = & \mathbf{\Psi} & \& & \mathbf{M}\left(\mathbf{x}, \cdot\right) & = & \mathbf{P}\left(\mathbf{x}, \cdot\right) \end{array} \tag{7.15}$$

where $\mathbf{K}(\mathbf{x}, \cdot)$ is calculated following the Mercer's theorem and the RKHS theory, $\mathbf{P}(\mathbf{x}, \cdot)$ is straightforwardly obtained from the DFM kernel and $t = 1/2$ in the KPCA / projection case.

7.2 Pre-image problem

Up till now, we have worked with data lying in $\mathbb{R}^n$. Henceforth, data belongs generically to a space $\mathbb{X}$, provided a differentiable distance exists in such space. Given a point in the reduced space $\mathbf{y} \in \mathbb{R}^m$, we endeavor to find the point $\mathbf{x} = \tilde{\Phi}^{-1}_{|\mathcal{M}}(\mathbf{y})$ in the manifold $\mathcal{M}$ such that $\tilde{\Phi}(\mathbf{x}) = \mathbf{y}$, i.e. the pre-image of $\mathbf{x}$ [115, 8, 60]. As noted by Pablo Arias, Gregory Randall and Guillermo Sapiro in [8] where $\mathbb{X} = \mathbb{S}$ is the shape space, such a point $\mathbf{x}$ — *i.e. a shape in their work* — might not exist since the pre-image problem is ill-posed. To circumvent this problem, they search for the shape that optimizes a given optimality criterion in the reduced space.

7.3 *Attracting forces* toward a manifold

7.3.1 Introduction

As previously introduced, our goal is to build a sound non-linear shape prior based on a category of shapes modeled as a *shape manifold*. A prior term energy used in segmentation tasks should intuitively attract the shape toward the *shape manifold*. As a first step, it is therefore natural to construct a projection operator onto manifold, which is detailed in the sequel. Note that we deliberately give formulations applicable to general data in this section and keep applications to shapes and image segmentations for chapter 8.

The first projection operator was proposed in [75] in the context of shapes. We simply determine the point on the manifold that minimizes the distance to the point to project and give an iterative algorithm that best approximates the projection. A second projection operator [76] , faster and simpler in nature, is also described. Nevertheless, in some applications such as in image segmentation, the minimizing path to the projection onto the manifold is particularly more important than the projection itself. Thus, we provide a sound efficient operator [80, 81] that attracts data toward the manifold at a given constant embedding.

7.3.2 General assumptions and Delaunay triangulation

In this work, we are interested in interpolating the manifold $\mathcal{M}$ between "neighboring" training samples. Therefore, we assume that the point $\mathbf{y} \in \mathbb{R}^m$ falls inside the convex-hull of the training samples in the reduced space (where the point $\mathbf{x} \in \mathbb{R}^m$ is outside, we would consider instead its orthogonal projection on the convex-hull). In this sense, the set of training samples must be exhaustive enough to capture the limits of the manifold $\mathcal{M}$.

We also assume that any point $\mathbf{x}$ belonging to the manifold $\mathcal{M}$, can be expressed as a weighted mean (also known as the Karcher mean) that interpolates between "neighboring" samples for the diffusion distance. (This hypothesis is applicable, for example, in the case of shape manifolds [40]). To this end, we exploit the Euclidean nature of the reduced space $\mathbb{R}^m$ to determine the $m+1$ closest neighbors of $\mathbf{x}$ (note that if the point $\mathbf{y} \in \mathbb{R}^m$ is located outside the convex-hull, then only m neighbors are identified.). In this sense, the set of training samples must be sufficiently dense for the interpolation to be meaningful.

We compute a *Delaunay triangulation* $\mathcal{D}_{\mathcal{M}}$ in the reduced space of the training data and identify the $m+1$ closest neighbors of $\mathbf{x}$ as the $m+1$-Delaunay simplex that $\mathbf{x}$ belongs to. This m-dimensional simplex is formed by $m+1$ points that correspond to the image by $\tilde{\Phi}$ of the $m+1$ closest neighbors $\mathbb{N} = (\mathbf{x}_0, ..., \mathbf{x}_m)$ of $\mathbf{x}$ in $\mathbb{X}$ for the diffusion metric [47].

7.3.3 *Attracting force* #1: closest projection

Image segmentation methods that take shape priors into account, generally require the projection (in some sense) of a shape candidate onto the set of shape samples. As previously mentioned, this projection is often just the mean of the samples, and sometimes a variation of this mean according to *deformation modes* [122, 165, 40]. Here, we propose a projection based on our local interpolation (see chapter 3):

Solution 1 (Minimizing the distance to the projection)

Let $\mathcal{M}$ be a finite m dimensional smooth manifold lying on $\mathbb{X}$.

Let $\mathbf{\Gamma} = \mathbf{x}_1, \ldots, \mathbf{x}_p$ be p points sampling the manifold $\mathcal{M}$.
Let $\mathbf{x}$ be a point of $\mathbb{X}$.

Let $\mathbb{N}(\mathbf{x}) = (\mathbf{x}_{j_0}, \cdots, \mathbf{x}_{j_m})$ $(\forall k = 0, \ldots, m \quad \mathbf{x}_{j_k} \in \mathbf{\Gamma})$ *be a neighborhood system of* $\mathcal{M}$ *close to* $\mathbf{x}$ *chosen based on the Delaunay triangulation of the reduced space obtained by Diffusion maps, and* $\mathbf{\Theta} = \{\theta_1, \ldots, \theta_m\}$ *their associated barycentric weights*
We define the local projection $\Pi_{\mathcal{M}}(\mathbf{x})$ *of* $\mathbf{x}$ *onto* $\mathcal{M}$ *to be the interpolation according to* $\mathbb{N}(\mathbf{x})$ *that is the closest to* $\mathbf{x}$ *:*

$$\begin{aligned} \Pi_{\mathcal{M}}(\mathbf{x}) &= \bar{X}_{\mathbb{N}(\mathbf{x})}(\mathbf{\Theta}_\Pi) \\ \text{with } \mathbf{\Theta}_\Pi &= \arg\min_{\mathbf{\Theta}} d(\mathbf{x}, \bar{X}_{\mathbb{N}(\mathbf{x})}(\mathbf{\Theta})) \end{aligned} \tag{7.16}$$

where $\bar{X}_{\mathbb{N}(\mathbf{x})}(\mathbf{\Theta})$ *is a weighted mean interpolation of the manifold* $\mathcal{M}$ *between the points of* $\mathbb{N}(\mathbf{x})$ *defined by the following equation (please also refer to equation* (3.24)*)*

$$\bar{X}_{\mathbb{N}(\mathbf{x})}(\Theta) = \arg\min_{X} \sum_{k=0}^{m} \theta_k d\left(\mathbf{x}_{j_k}, X\right)^2 \tag{7.17}$$

While such a projection is clearly better than choosing the nearest neighbor, the energy involved in equation (7.16) cannot be minimized easily. The variations with respect to Θ of the distance $d(\mathbf{x}, \bar{S}_{\mathbb{N}(\mathbf{x})}(\Theta))$ between the interpolation and shape $\mathbf{x}$ are intricate. The gradient of this distance could be written, but, involving second order shape derivatives, it yields a complex minimization scheme that might not be useful for our purpose. Keeping shape priors in mind, it appears that an approximation of the projection $\Pi_{\mathcal{M}}(\mathbf{x})$ is sufficient.

Many algorithms might be designed to get an approximate solution to (7.16). We suggest an iterative scheme illustrated figure 7.1 that we call the *snail algorithm*. Although it is not guaranteed to converge, it is proved to give good approximations of the projection of a candidate shape onto the shape prior manifold, in only a few iterations. Actually, we investigated more extensive searches of the minimum of (7.16) without any significant improvement. The snail algorithm is defined by:

Solution 2 (Approximation of minimization (7.16)) *Let* $\mathcal{M}$*,* $\mathbf{x}$ *and* $\mathbb{N}(\mathbf{x})$ *be defined as in solution 1. The snail algorithm proceeds as follows:*

1. Initialization: choose the shapes of the neighborhood system as initial guesses.

   For $i = 0, \cdots, m$*, let* $\mathbf{\Theta}^i = (\theta_0^i, \cdots, \theta_m^i)$ *be defined by* $\theta_j^i = \delta_{ij}$

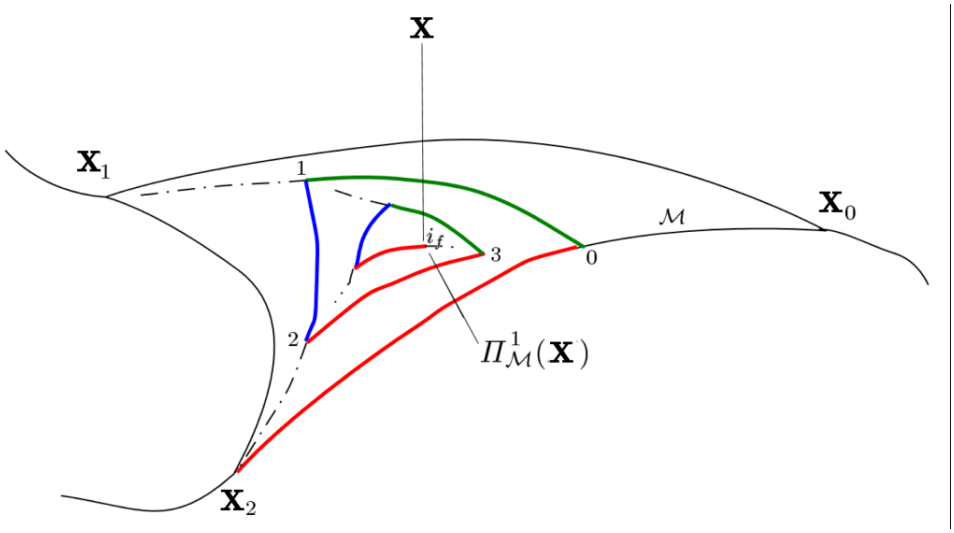

Figure 7.1: **The Snail algorithm:** steps are indexed 1, 2, ..., i_f

2. Iterations: look for a better projection between the latest estimate and the one computed $m+1$ steps before.

   For $i = m, m+1, \cdots$ until convergence, estimate:

$$\boldsymbol{\Theta}^{i+1} = \alpha_i \boldsymbol{\Theta}^i + (1-\alpha_i)\boldsymbol{\Theta}^{i-m}$$

$$\text{with } \alpha_i = \arg\min_{0\leq\alpha\leq 1} d(\mathbf{x}, \bar{X}_{\mathbb{N}(\mathbf{x})}(\alpha\boldsymbol{\Theta}^i + (1-\alpha)\boldsymbol{\Theta}^{i-m})) \quad (7.18)$$

3. Exit:

   Let i_f be the index of last iteration. Approximate the projection by:

$$\Pi^1_{\mathcal{M}}(\mathbf{x}) = \bar{X}_{\mathbb{N}(\mathbf{x})}(\boldsymbol{\Theta}^{i_f})$$

Note that we still need to design a minimization scheme to estimate the optimal α in (7.18). Again, a variational method is both too slow for our purpose and useless for an approximation. Computing a small number of interpolations and keeping the best one turns out to be satisfactory. Moreover, because these interpolations are obtained through a gradient descent (see [40]), estimating the interpolations for a series of α is efficient, each interpolation being the initial guess

for the next one.

We finally define the attracting force #1 as follows.

The attracting force #1 is a force denoted $\vec{F_1}$ that attracts a point $\mathbf{x} \notin \mathcal{M}$ toward its projection $\Pi^1_{\mathcal{M}}(\mathbf{x})$ (Fig. 7.5).
$\vec{F_1}$ is called *closest projection force*.

In the context of image segmentation, the distance $d(\mathbf{x}, \Pi^1_{\mathcal{M}}(\mathbf{x}))$ constitutes the *shape prior* term.

7.3.4 *Attracting force* #2: same embedding

The previous projection may be too long too compute and may not be easily integrated into a segmentation process. In order to overcome this limitation, we introduce a second *attracting force* and we define the projection $\Pi^2_{\mathcal{M}}(\mathbf{x})$ of point $\mathbf{x}$ onto the manifold $\mathcal{M}$

Let us denote $\mathbf{y} = \tilde{\mathbf{\Phi}}(\mathbf{x})$. Having identified the neighborhood $\mathbb{N}(\mathbf{x}) = (\mathbf{x}_{j_0}, ..., \mathbf{x}_{j_m})$ of $\mathbf{x}$, we search for a Karcher mean of $\mathbb{N}(\mathbf{x})$ such that its embedding is $\tilde{\mathbf{\Phi}}(\mathbf{x})$. The projection is then given by:

$$\Pi^2_{\mathcal{M}}(\mathbf{x}) = \arg\min_{\Theta} \left\| \mathbf{y} - \tilde{\Phi}\left(\bar{X}_{\mathbb{N}(\mathbf{x})}(\Theta)\right) \right\| \tag{7.19}$$

The process is initialized by using the barycentric coordinates of $\tilde{\mathbf{\Phi}}(\mathbf{x})$ within the simplex formed by $\{\tilde{\mathbf{\Phi}}(\mathbf{x}_{j_0}), \ldots, \mathbf{\Phi}(\mathbf{x}_{j_m})\}$ Figure 7.2 illustrates our projection operator on a 2 dimensional manifold lying in $\mathbb{R}^3$. Note that we do not try to estimate the manifold outside of its limits, as the ones defined by the convex hull of the training points in the reduced space. As a consequence, projection of a point located outside the manifold will belong to the border of the manifold.
We finally define the attracting force #2 as follows.

The attracting force #2 is a force denoted $\vec{F_2}$ that attracts a point $\mathbf{x} \notin \mathcal{M}$ toward its projection $\Pi^2_{\mathcal{M}}(\mathbf{x})$ (Fig. 7.5).
$\vec{F_2}$ is called *same embedding force*.

This low computational force however not relevant in the framework of image segmentation. In our model, the final segmentation result is indeed a trade-off between the shape prior term and the data term, which is not necessarily on the shape manifold. Then, the path toward the manifold becomes of particular importance, as well as the direction of the force in the context of image segmentation.

7.3.5 *Attracting force* #3: constant embedding

In the previous section, we defined a projection of a point $\mathbf{x}$ onto the manifold that has the same embedding. The resulting *attracting force* however does not keep the embedding constant. We propose a more elegant force that preserves the embedding along the entire evolution path. Diffusion maps create a diffusion process along and outside the manifold (it is based on an approximation of the Laplace-Beltrami operator). In figure 7.3, we represent the embedding values using the Nyström extension, "around" the 1-D swiss roll manifold. We see that attracting point $\mathbf{x}$ toward $\mathcal{M}$ while keeping a constant embedding is a natural idea.

Let $\mathcal{S}$ be the iso-level set of contant embedding points around $\mathbf{x}$.

$$\mathcal{S} = \{\mathbf{x}' \in \mathbb{X}, \tilde{\mathbf{\Phi}}(\mathbf{x}') = \tilde{\mathbf{\Phi}}(\mathbf{x})\}. \tag{7.20}$$

The problem is now the following: considering $\mathcal{S}$, we have to let $\mathbf{x}$ evolve on $\mathcal{S}$ toward $\mathcal{M}$ as fast as possible. To do so, we choose $\Pi^2_{\mathcal{M}}(\mathbf{x})$ as a target point and let evolve $\mathbf{x}$ toward $\Pi^2_{\mathcal{M}}(\mathbf{x})$ at constant embedding.

Deformation space preserving the embedding

Each point $\mathbf{x} \in \mathbb{X}$ has its associated deformation space, the tangent space denoted $\mathbb{T}_x$. For instance, when $\mathbb{X}$ is the shape space, $\mathbb{T}_x$ corresponds to normal deformation fields that can be applied to shape $\mathbf{x}$. We define $\mathbf{E_x}$ the space spanned by $\mathbf{A} = \{\vec{a}_1, \cdots \vec{a}_m\}$ where $\forall k \in \{1, \cdots, m\} \quad \vec{a}_k = \nabla\tilde{\Phi}_k(\mathbf{x})$. See equations (7.10) & (7.11). The space $\mathbf{E_x}$ is intuitively the space of deformations at $\mathbf{x}$ that maximally modify the embedding. On the opposite side, the space denoted $\mathbf{E}^{\perp}_{\mathbf{x}}$, orthogonal to $\mathbf{E_x}$, corresponds to the deformations that have a minimal influence on the embedding value. We then write the deformation space at $\mathbf{x}$ by using the direct sum symbol $\oplus$, $\mathbf{E_x}$ and $\mathbf{E}^{\perp}_{\mathbf{x}}$:

$$\mathbb{T}_{\mathbf{x}} = \mathbf{E_x} \oplus \mathbf{E}^{\perp}_{\mathbf{x}} \tag{7.21}$$

We finally calculate from $\mathbf{A}$ an orthonormal basis $\mathbf{B} = \{\vec{b}_1, \cdots, \vec{b}_m\}$ of $\mathbf{E}_\mathbf{x}$ using the orthogonalization Gram-Schmidt process . In order to preserve the embedding during the evolution, we define the projection of any velocity field $\vec{w}$ onto the space $\mathbf{E}_\mathbf{x}^\perp$:

$$\Pi_{\mathbf{E}_\mathbf{x}^\perp}(\vec{w}) = \vec{w} - \sum_{k=1}^{m} \langle \vec{w}, \vec{b}_k \rangle \vec{b}_k \tag{7.22}$$

We finally define the attracting force #3 as follows.

The attracting force #3 is a force denoted $\vec{F}_3$ that attracts a point $\mathbf{x} \notin \mathcal{M}$ toward its projection $\Pi^2_\mathcal{M}(\mathbf{x})$ such that the embedding value is preserved (Fig. 7.5).
$\vec{F}_3$ is called *constant embedding force*.

$$\vec{F}_3(\mathbf{x}) = \Pi_{\mathbf{E}_\mathbf{x}^\perp}\left(\vec{F}_2(\mathbf{x})\right) \tag{7.23}$$

We have defined a projection that attracts a point toward the manifold at constant embedding. This projection will be used in chapter 8 to achieve *manifold denoising*, and will be employed as a shape prior term in segmentation tasks.

Gradient of the embedding function

In the following lines, we detail the gradient of the embedding function $\tilde{\Phi}(x)$ (in the DFM case). The gradient is expressed as follows.

$$\nabla_\mathbf{x} \tilde{\mathbf{\Phi}}_k(\mathbf{x}) = \sum_i \nabla_\mathbf{x} \mathrm{p}(\mathbf{x}, \mathbf{x}_i) \Psi_k(\mathbf{x}_i) \tag{7.24}$$

We have

$$\mathrm{p}(\mathbf{x}_i, \mathbf{x}_j) = \frac{\tilde{w}(\mathbf{x}_i, \mathbf{x}_j)}{\tilde{q}(\mathbf{x}_i)} \quad \text{with} \quad \tilde{q}(\mathbf{x}_i) = \sum_{\mathbf{x}_j \in \Gamma} \tilde{w}(\mathbf{x}_i, \mathbf{x}_j) \tag{7.25}$$

which gives

$$\mathrm{p}(\mathbf{x}, \mathbf{x}_j) = \frac{w(\mathbf{x}, \mathbf{x}_j)}{\sum_b K_{jb} w(\mathbf{x}, \mathbf{x}_b)} \quad \text{with} \quad K_{jb} = \frac{\sum_a w(\mathbf{x}_a, \mathbf{x}_j)}{\sum_d w(\mathbf{x}_d, \mathbf{x}_b)} \tag{7.26}$$

Lastly, we have

$$\begin{aligned}\nabla_{\mathbf{x}}\mathrm{p}(\mathbf{x},\mathbf{x}_j) &= \frac{1}{\left(\sum_b K_{jb}w(\mathbf{x},\mathbf{x}_b)\right)^2}\left(\nabla_{\mathbf{x}}w(\mathbf{x},\mathbf{x}_j)\sum_b K_{jb}w(\mathbf{x},\mathbf{x}_b)\right.\\ &- \left. w(\mathbf{x},\mathbf{x}_j)\sum_b K_{jb}\nabla_x w(\mathbf{x},\mathbf{x}_b)\right) \qquad (7.27)\end{aligned}$$

Implementation

Equation (7.27) can be easily implemented but its calculation should be as fast as possible. In order to accelerate the process, we write the $\nabla_x\tilde{\Phi}_k(x)$ in the following matrix form and use an optimized matrix library to calculate the gradient.

$$\nabla_{\mathbf{x}}\tilde{\mathbf{\Phi}}_k(\mathbf{x}) = V(\mathbf{x})\left(I + B(\mathbf{x})A(\mathbf{x})\right)A(\mathbf{x})\Psi \qquad (7.28)$$

where :

$$\begin{aligned}V(\mathbf{x}) &= [\vec{v}_1,\ldots,\vec{v}_p] \quad \text{with} \quad \forall\, i=1,\ldots,p \quad \vec{v}_i = \nabla_x w(\mathbf{x},\mathbf{x}_j)\\ A(\mathbf{x}) &= \mathrm{diag}\left(\left[\frac{1}{\alpha_1},\ldots,\frac{1}{\alpha_p}\right]\right) \quad \text{with} \quad \forall\, i=1,\ldots,p \quad \alpha_i = \sum_b K_{ib}w(\mathbf{x},\mathbf{x}_b)\\ B(\mathbf{x}) &= \left[\sum_b \beta_{1b},\ldots,\sum_b \beta_{pb}\right] \quad \text{with} \quad \forall\, i=1,\ldots,p \quad \beta_{ib} = -w(\mathbf{x},\mathbf{x}_i)K_{ib}\\ \Psi &= [\Psi_1,\ldots,\Psi_m]\end{aligned}$$

7.4 Conclusion

In this chapter, we introduced three possible forces to attract points toward the manifolds (Fig. 7.5).

- **Closest projection force**: the target is defined by $\Pi^1_{\mathcal{M}}(\mathbf{x})$, the closest point of $\mathcal{M}$ to $\mathbf{x}$. **Resulting force:** $\vec{F}_1$
- **Same embedding force**: the target is defined by $\Pi^2_{\mathcal{M}}(\mathbf{x})$, the point of $\mathcal{M}$ having the same embedding as $\mathbf{x}$. **Resulting force:** $\vec{F}_2$
- **Constant embedding force**: the target is also defined by $\Pi^2_{\mathcal{M}}(\mathbf{x})$, but only the components of $\vec{F}_2$ that preserve the embedding are kept. **Resulting force:** $\vec{F}_3$

These forces will be illustrated in experiments presented in chapter 8.

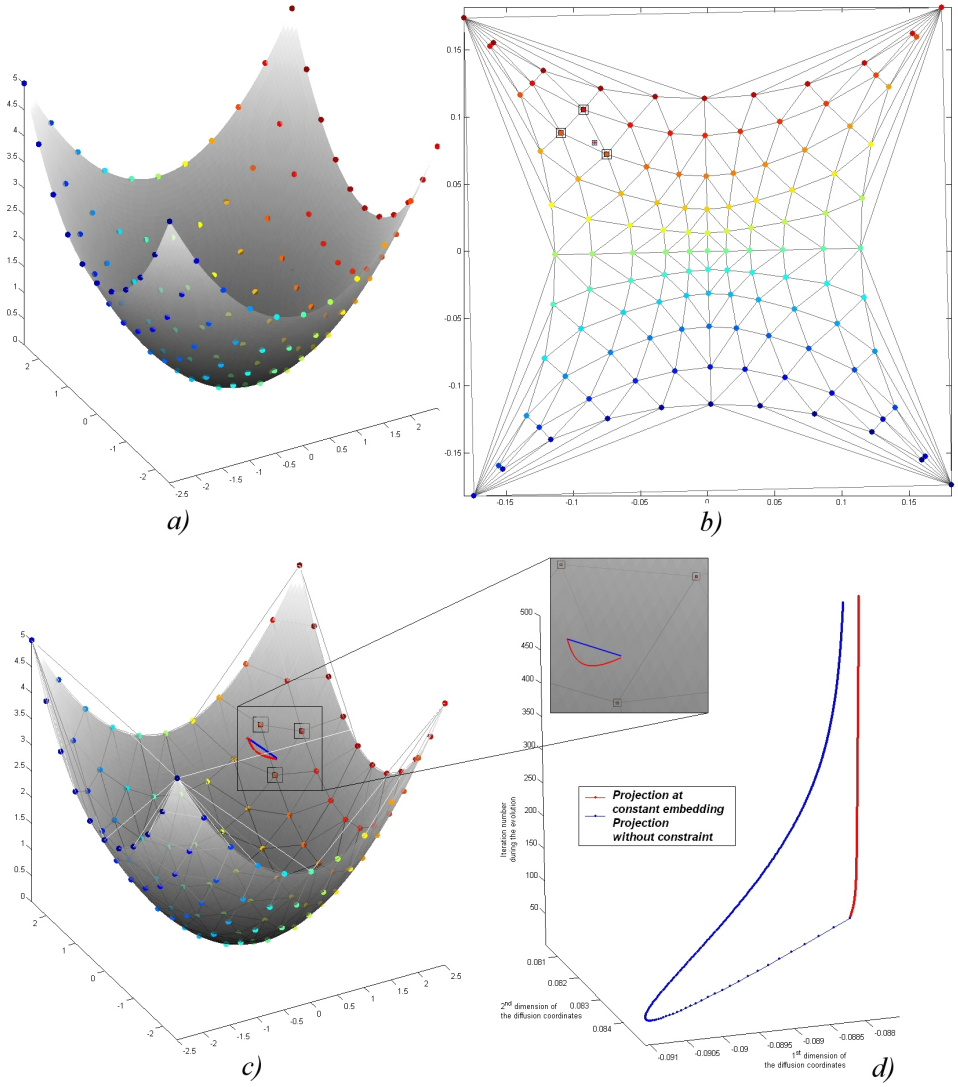

Figure 7.2: **Projection onto the manifold: attracting forces # 2 & # 3** - a) Set of point samples lying on the surface given by the equation $f(x, y) = x^2 + y^2$. b) The reduced space and the Delaunay triangulation. c) Steepest descent evolution toward the weighted mean $\Pi^2_{\mathcal{M}}(\mathbf{x})$ (in blue) and at constant embedding (in red). The Delaunay triangulation is represented in the original space. d) Values of the embedding during the two evolutions

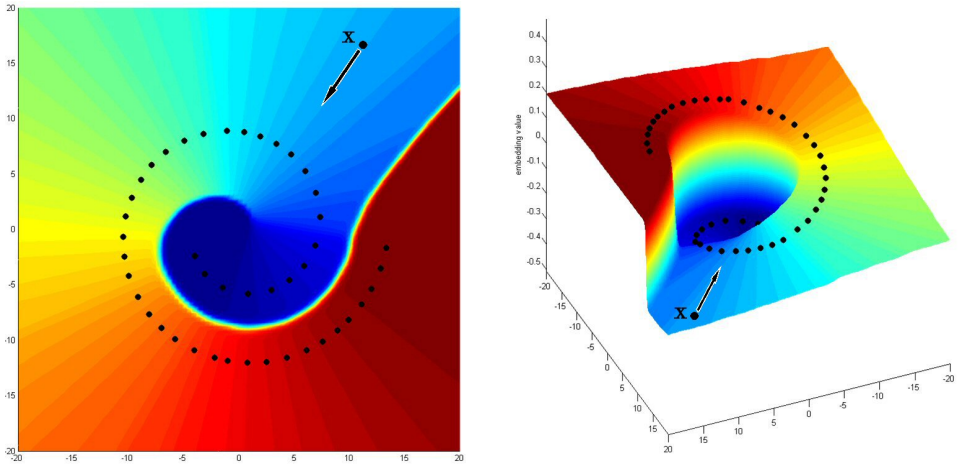

Figure 7.3: **Being attracted toward the manifold (here, black dots) at constant embedding** The color code represents the embedding value (also visualized along the z-axis on the right). The force #3 enforces the point to be evolved toward the manifold by preserving the embedding value.

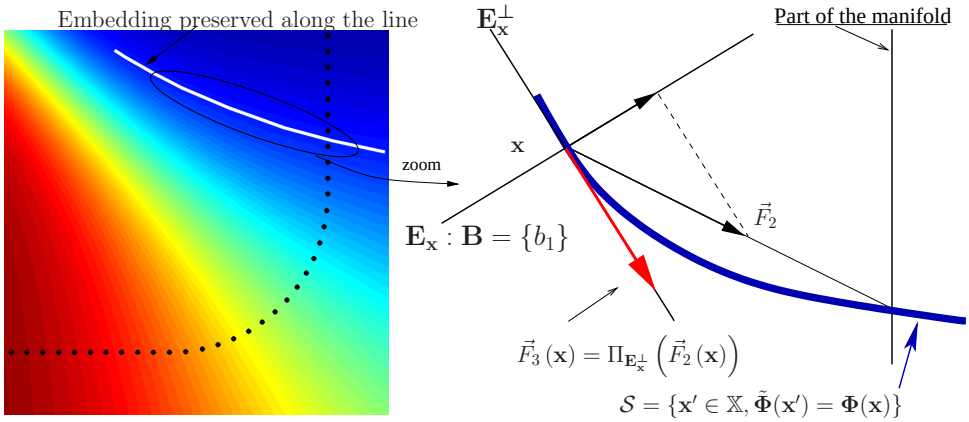

Figure 7.4: **Evolution toward the manifold at constant embedding value** On the left: Sampled manifold (black dots) and visualization of the embedding value (color code). On the right: schematic detail. The blue line represents the set of point having the same embedding value as $\mathbf{x}$. The force #2 modifies the embedding value during the evolution. The force #3 (red arrow in the tangent direction of the blue line) enforces the point $\mathbf{x}$ to evolve at constant embedding value.

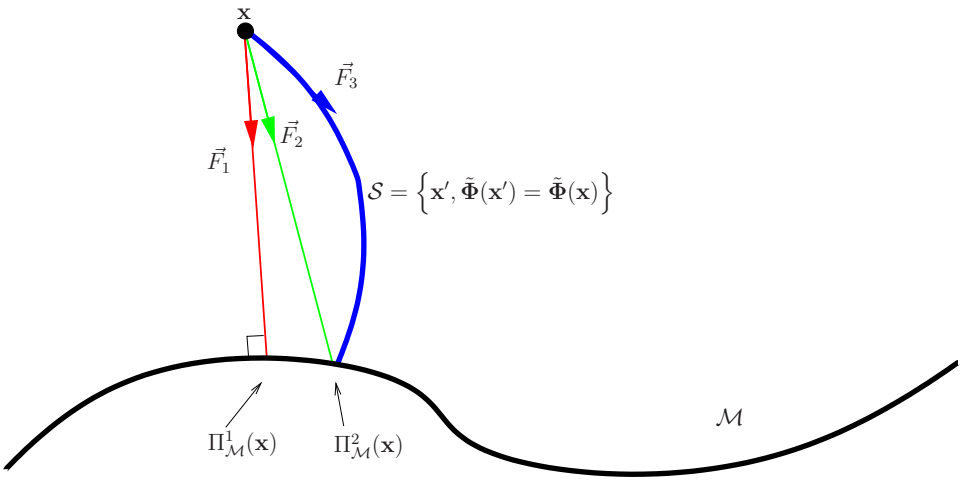

Figure 7.5: **Three forces to attract points toward the manifold $\mathcal{M}$ -** **1.** $\vec{F}_1$: force directed toward the closest point of $\mathcal{M}$ to $\mathbf{x}$ **2.** $\vec{F}_2$: force directed toward the point of $\mathcal{M}$ having the same embedding as $\mathbf{x}$ **3.** $\vec{F}_3$: force at constant embedding toward $\mathcal{M}$

Chapter 8

Applications of *attracting forces* to manifold denoising and image segmentation with priors

Abstract

The tools developed in chapters 5 and 7 of this dissertation are successfully applied to manifold denoising and image segmentation with non linear shape priors. In particular, this chapter provides results obtained on various 2D and 3D examples corresponding to different shape prior manifold*: fish shapes, cars shapes and anatomical structures.*

Contents

Original contributions

We design a new approach to manifold denoising, which relies on the *constant embedding force*.
Related publication: ICCV'07 [81].

We project shapes onto *shape manifolds* using *attracting forces* $\vec{F}_1$ (*closest projection force*), $\vec{F}_2$ (*same embedding force*) and $\vec{F}_3$ (*constant embedding force*).
Related publications: SSVM'07 [75], ICIP'07 [76], ICCV'07 [81].

We achieve image segmentation with non linear shape priors using *attracting forces* $\vec{F}_2$ (*same embedding force*) and $\vec{F}_3$ (*constant embedding force*).
Related publications: ICIP'07 [76], ICCV'07 [81], MICCAI'07 [80].

8.1 Manifold Denoising

In section 7.3.4, we estimate the manifold $\mathcal{M}$ by interpolating between training shape samples (i.e. by minimization of an energy functional) subject to constant embedding constraints (Eq. 7.19). As such, the manifold $\mathcal{M}$ is assumed to go through every training sample. Unfortunately, this implies that our manifold reconstruction is sensitive to outliers that are mapped among other training samples into the reduced space through the embedding $\tilde{\Phi}$ (Fig 8.1-a). To alleviate this problem, we propose to use the mapping $\tilde{\Phi}$ and the Euclidean nature of the reduced space to design a denoising functional $E^{\text{denoising}}$.

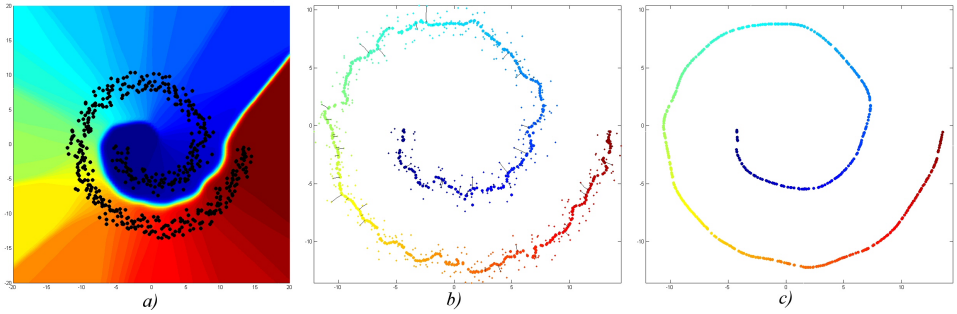

Figure 8.1: **Manifold Denoising** - a) The set of point sample with the iso-level set of the one dimensional embedding. b) After 5 iterations of denoising. Smaller points correspond to original data, bigger to denoised data. The black lines are the paths of some randomly points during the evolution c) Final result.

The embedding $\tilde{\Phi}$ captures the intrinsic geometry of the manifold $\mathcal{M}$ by mapping training samples into $\mathbb{R}^m$ isometrically with respect to a diffusion distance in the original shape space. It is useful to interpret the mapping as a smoothing filter that absorbs the "noise components orthogonal to the manifold" and maps outliers among valid training samples. In light of this, we propose to use the connectedness of the Delaunay triangulation $\mathcal{D}_{\mathcal{M}}$ in the reduced space to infer connectedness of the training samples in the original space $\mathbb{X}$. For each training sample $\mathbf{x}_i \in \mathbf{\Gamma}$, we identify its set $\mathbb{N}_i = \mathbb{N}(\mathbf{x}_i)$ of adjacent neighbors that are connected in the Delaunay triangulation $\mathcal{D}_{\mathcal{M}}$ (see section 7.3.2). We then define the denoising functional

over all training samples:

$$E^{\text{denoising}}(\Gamma) = \sum_{\mathbf{x}_i \in \Gamma} \sum_{\mathbf{x}_{i,k} \in \mathbb{N}_i} d^2(\mathbf{x}_i, \mathbf{x}_{i,k}), \tag{8.1}$$

The functional $E^{\text{denoising}}$ is minimized by gradient descent with the additional constraints of preserving the embedding. To do so, we enforce the additional constraint $\forall \mathbf{x}_i \in \Gamma\ \tilde{\Phi}(\mathbf{x}_i) = \text{constant}$, which can be expressed by $m \times p$ orthogonality constraints in the tangent space as presented in the previous chapter (sec 7.3.5).

Minimization of the functional $E^{\text{denoising}}$ implements the well-known umbrella-operator, which is a linear approximation of the Laplacian operator [68]. As such, our denoising framework acts as a diffusion process, attracting every shape sample toward the mean shape of its neighbors. In spirit, it is similar to the approach proposed by Hein and Maier in [102]. Yet, it is different in two essential aspects. First, the diffusion process is based on the diffusion distance, which is more robust to outliers than geodesic distance. The connectivity of the manifold $\mathcal{M}$ is directly derived from the Delaunay triangulation $\mathcal{D}_{\mathcal{M}}$. Also, during the evolution, we avoid the time consuming procedure which consists of updating the whole connectivity graph, since we enforce the embedding to remain the same. Finally, as noted in [68, 102], a trade-off between reducing the noise and smoothing the manifold exists. Minimization of the energy (Eq. 8.1) leads to a global flow which smooths the manifold via mean curvature.

8.2 Projections and shapes

Introduction

In this section, we fulfil the purposes and initial goals of this dissertation. The shape deformation framework and manifold learning techniques such as DFM are finally employed together in order to design attractive forces toward *shape manifolds*. In this context, the main tools developed in this thesis — i.e. the three *attracting forces* — are successfully applied to various shapes in the rest of this section. We provide illustrations for each force, using *shape manifolds* (with synthetic and real shapes: rectangles, crosses, fishes, and even 3D brain ventricles). As previously mentioned, our approach is neither built on a particular shape representation, nor a particular shape distance. The method only requires a shape representation endowed with a differentiable distance.

8.2.1 Attracting force #1, rectangle shape manifold & fish shape manifold

Synthetic example

In this example, the *shape manifold* is a set of rectangles (orientation between $-\frac{\pi}{6}$ and $\frac{\pi}{6}$, length between 2 and 4 times the width). The training set corresponds to shapes randomly sampled such that the distribution of their corners is the uniform law in an authorized area . In order to show the robustness of the method, we constructed deformed shapes and computed their projection. To do so, we extract the neighbor system defined above The dimension of the *shape manifold* is 2 and thus interpolation is obviously involves 3 shapes. A rectangle is chosen to lie between two angular positions and two different sizes, and is then altered by slightly deforming the shape (so it does not belong to the *shape manifold* anymore). We show in Figure 8.2 the altered shape, the neighbor system chosen and its projection following the *snail algorithm* presented in section (Eq. 7.3.3).

Real shapes : fishes

We provide also prominent results with a *fish manifold* built from the SQUID[1] database. We have strongly deformed a fish shape S: the head is deformed and the shape suffers from many occlusions. Of course, such a shape does not belong to the set used to build the graph Laplacian. Then, we determined the neighbor system $S_0, \dots S_3$ and the projection $\Pi^1_{\mathcal{M}}(S)$ onto the *shape prior manifold*. Such a projection, which is clearly better than the nearest neighbor, is able to handle shape occlusions and great deformations as it recovers most of the original shape (Fig. 8.3).

8.2.2 Attracting force #2, cross shape manifold

The *snail algorithm* used in the previous examples gives interesting results but this approach has some serious drawbacks. Convergence of the target shape should be achieved before designing any attractive force toward the manifold. Such a process cannot be easily integrated into a segmentation task. Consequently, the second force is designed to overcome such limitations and can be jointly used with the embedding regularization (Sec. 7.1.2) or the Nyström extension (Sec. 7.1.3)

[1] SQUID: Shape Queries Using Image Databases. SQUID was used for MPEG-7 evaluation of shape descriptors.

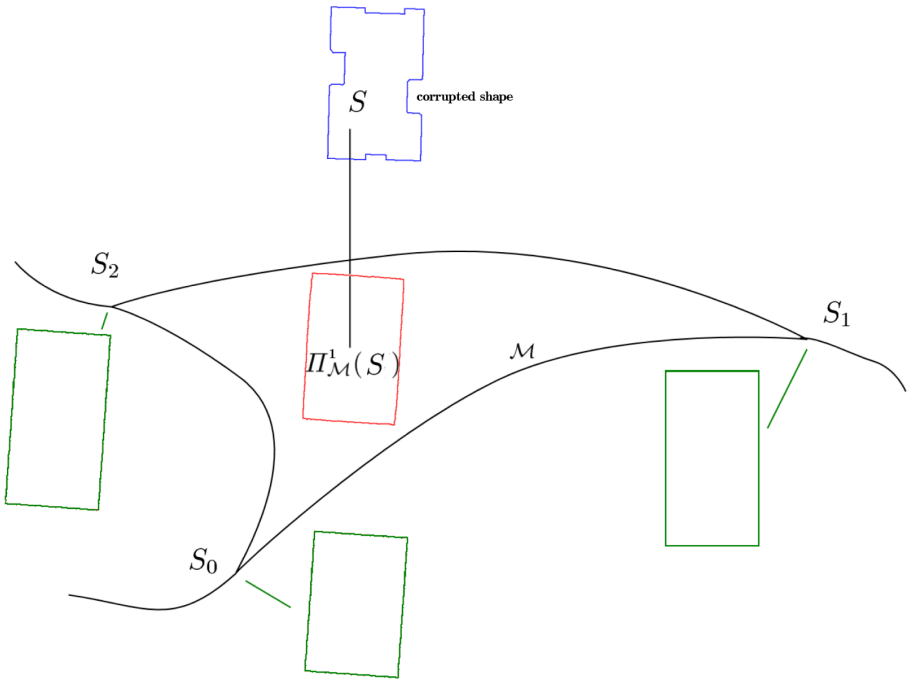

Figure 8.2: **Projection onto the *rectangle manifold*** using the *snail algorithm* (*attracting force #1* - **direct projection**): $\{S_0, S_1, S_2\}$ is the neighborhood system

to solve the pre-image problem. In this experiment, the training set consists of 105 crosses, whose arms, although of various sizes, maintain the horizontal and vertical symmetry of the shape. Figure 8.4 (above) illustrates the *shape manifold* represented in a 2 dimensional reduced space. Note that, the dimension is deduced straightforwardly from the construction of the manifold itself. As previously, a cross S is modified by slightly deforming the shape. Figure 8.4 (below) shows the altered shape, the neighborhood system chosen ($\{S_0, S_1, S_2\}$), and its projection by using the **same embedding force** and an embedding regularization (sec. 7.1.2).

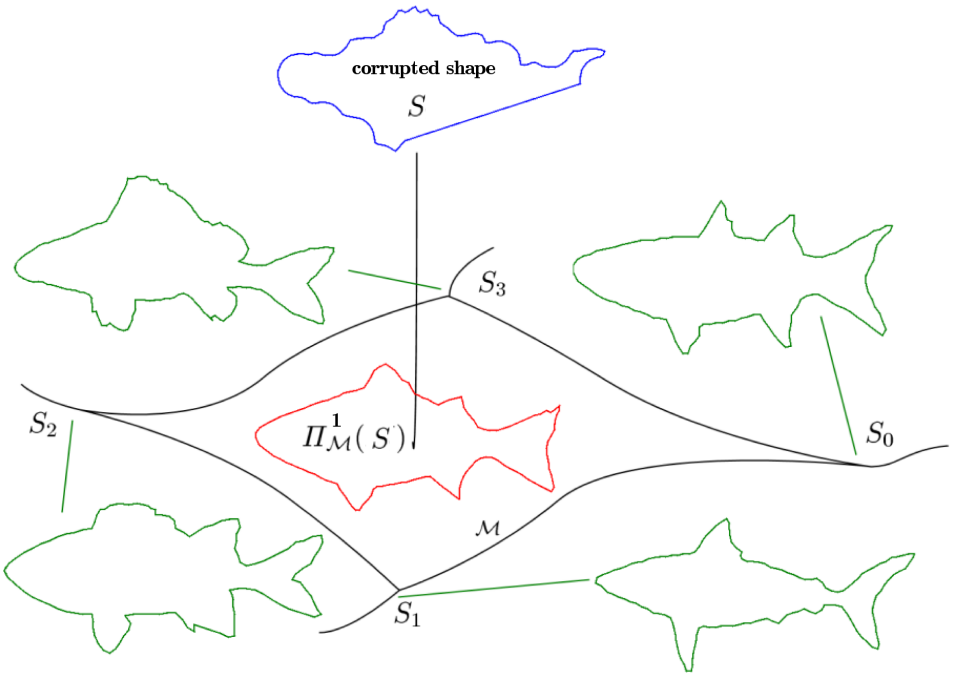

Figure 8.3: **Projection onto the *fish manifold*** using the *snail algorithm* (*attracting force #1* - **closest projection**): $\{S_0, S_1, S_2, S_3\}$ is the neighborhood system

8.2.3 Attracting force #3, ventricle shape manifold

In this subsection, we apply the third force and compare it to the second one. We recall that the third *attracting force* is a force that attracts points toward the manifold at constant embedding, and has better signification as shown in the following lines.

We use a dataset of 39 ventricle nuclei from Magnetic Resonance Imaging (MRI). The shapes are aligned using their principal moment before computing their diffusion coordinates. In this experiment, we compare the projection at constant embedding, the neighbors in the Delaunay triangulation of the reduced space and the mean shape obtained from these neighbors. Our deformation surface is implemented in the *Level Set* framework: the distance functions of the ventricle shapes are encoded in $140 \times 75 \times 60$ images. To perform the projection, we start from an

ellipsoid aligned on the 3D shape set. Its embedding is indicated by the black point in Figure 8.5. The nearest shapes in the corresponding Delaunay triangle are easily identified in order to compute the mean shape target and the projection at constant embedding . The projection at constant embedding captures details (on the right side of the ventricle) of closest shapes (38 & 22) that the mean shape loses due to its smoothing properties.

8.3 Image segmentation with general non linear shape priors

8.3.1 Fish shapes, attracting force #2

We propose to apply the method presented in section 7.3.4 in the context of image segmentation with shape priors. Without loss of generality, the method is stated as a variational problem attempting to minimize the energy $E^T(S)$ of an evolving curve S.

$$E^T(S) = E^{ac}(S) + \alpha E^p(S) \tag{8.2}$$

$E^a(S)$ is the common energy used in the active contour framework. $E^p(S)$ is the prior term that attracts the evolving shape toward the *shape prior manifold* according to $\Pi^2_{\mathcal{M}}(\cdot)$.

$$E^p(S) = d\left(S, \Pi^2_{\mathcal{M}}(S)\right)^2 \tag{8.3}$$

α is a weighting parameter that influences the importance of the prior term. The energy $E^T(S)$ is minimized using the calculus of variations. Results are presented in Figures 8.6 (embedding of the *shape manifold*) 8.7 (segmentation of an occluded fish).

8.3.2 Surveillance : Cars, attracting force #3

In this example, we illustrate the *attracting force #3* (as a shape prior term) in segmentation tasks of 2D car shapes. We are aiming at segmenting partly occluded cars. In this experiment, the non-linear prior is the manifold of the 2D shapes observed while turning around different cars. The dataset used is made up of 17 cars whose shapes are quite different : Audi A3, Audi TT, BMW Z4, Citroën C3, Chrysler Sebring, Honda Civic, Renault Clio, Delorean DMC-12, Ford Mustang Coupe, Lincoln MKZ, Mercedes S-Class, Lada Oka, Fiat Palio, Nissan 200sx , Nissan Primera , Hyundai Santa Fe and Subaru Forester. For each car, we extracted

12 shapes from the projection of the 3D CAD[2] model (Fig. 8.8 forming a dataset of 204 shape samples). The shapes are finally stored in the form of distance functions by means of a 160×120 image domain. In the learning stage, the embedding of the *car shape manifold* is estimated using diffusion maps over the dataset. Note that cars are registered under an affine transformation. In Figure 8.9 b, we represented the first two dimensions of the diffusion coordinates, which constitute the *reduced space*, and the corresponding Delaunay triangulation. Note that the car shapes have a coherent spatial organization in the *reduced space*.

Without loss of generality, we implement our surface deformation in the *Level Set* framework. We use a simple data term designed to attract the curve toward image edges [146], which gives the following evolution equation:

$$\begin{aligned} \frac{\partial \phi(x,y)}{\partial t} &= g\left(\nabla_{x,y,z}\hat{I}(x,y)\right)\left[\varepsilon\kappa(\phi(x,y)+\nu\right]\left|\nabla\phi(x,y)\right| \\ &\quad -\alpha\,\overrightarrow{v}_{sp}\cdot\bar{D}_S(x,\tau) \end{aligned} \tag{8.4}$$

where ϕ is the signed distance function of the evolving contour. $\kappa = \text{div}\left(\frac{\nabla\phi}{|\nabla\phi|}\right)$, $I(x,y)$ and ν are respectively the mean curvature, the image intensity at location (x,y) and a constant speed term to push or pull the contour. $g(\overrightarrow{z}) = \frac{1}{1+||\overrightarrow{z}||^2}$ is a stopping function used to extract edges.

In order to demonstrate the influence of our shape prior, we segment partly occluded cars which are not in the initial data set. We also choose images with different points of view. We initialize the contour with an ellipse around the car to segment and report the evolution in both cases, with and without our shape prior. The final results are presented in Figure 8.10. Without the shape prior, the energy is obviously minimized on the image edges. However, when the shape prior is incorporated, the new energy overcomes local minima of the data term energy and finally gives a "better" segmentation.

8.3.3 Bio Medical Imaging : Ventricle Nuclei, attracting force #3

We illustrate the potential benefits of our approach on another simple segmentation task, the segmentation of the ventricle nucleus from Magnetic Resonance Imaging (MRI). The *shape priors* term is built from the *attracting force #3*. Training

[2]CAD: Computed Assisted Design

shape samples were obtained from 39 manually segmented images of 10 young, 9 middle-aged, 9 old normal controls and 11 demented adults (Fig. 8.11). 39 data points probably form an insufficiently small data set and more shape samples are desirable to recover a satisfactory embedding. Note also that the artificial nature of the proposed segmentation task is only dedicated to reveal the influence of the shape prior term.

Estimating the dimension of the shape prior manifold

The dimension m of the reduced space is usually estimated from the profile of the eigenspectrum (Fig. 8.11-a), which is not always justified. However, there is not always an obvious choice (especially when the number of data points is insufficient). In our case, $m = 2$, $m = 3$, or $m = 4$ appear to be a realistic guess. Nevertheless, in the case of labeled data, one can disambiguate this choice by also requiring the embedding $\tilde{\Phi}_t$ to separate/cluster "well" the different groups. We simply define the degree of separability $d_{i,j}$ between two groups i and j by the distance $d_{i,j} = \frac{\|\mu_i - \mu_j\|}{\sqrt{\sigma_i^2 + \sigma_j^2}}$, where μ_i and σ_i^2 are the mean and variance in $\mathbb{R}^m$ of data points corresponding to group $\#i$. The degree of separability of the mapping $\tilde{\Phi}_t$ is then $\sum_{i,j} d_{i,j}$. Note that this method can also be used to determine an optimal value for the parameter t. Finally, on this unsatisfactory small data set, we find that the optimal mapping requires $m = 2$ (Fig. 8.11-a,b).

Closest neighbors

Diffusion maps embed *advantageously* the data set in the Euclidean space $\mathbb{R}^m$ *isometrically* with respect to a Diffusion distance in $\mathbb{S}$. This distance was shown to be more robust to outliers than geodesic distances [47]. To illustrate this point, we show in Fig. 8.11-c a manually altered shape with its two closest neighbors in $\mathbb{S}$ and $\mathbb{R}^m$. At least visually, the identified shapes in $\mathbb{R}^m$ appear more similar to the altered shape than the ones in $\mathbb{S}$.

Ventricle nucleus segmentation from MRI with occlusion

We consider a simple segmentation task which consists of segmenting the ventricle nucleus from an MRI that was modified by adding a white noise and degraded with an artificial occlusion (clearly visible in Fig. 8.12). Motivated by our choice of representing a shape S by its signed distance function $\bar{D}_S$, our surface deformation is implemented in the *Level Set* framework. The *Level Set* evolution is guided by a *simple* intensity-based velocity term, a curvature term, and the non-linear shape prior term:

$$\begin{aligned} \frac{\partial \phi(x,y,z)}{\partial t} &= [\beta\,(I(x,y,z) - T(x,y,z)) - \kappa]\,|\nabla\phi(x,y,z)| \\ &\quad -\alpha\,\overrightarrow{v}_{sp} \cdot \nabla\phi(x,y,z) \end{aligned} \tag{8.5}$$

where $I(x,y,z)$ and κ represent respectively the image intensity and mean curvature respectively at location (x,y,z), T is a threshold computed locally from image intensities, β and α two weighting coefficients equal to $\beta = 0.1$ and $\alpha = 0.1$. Figure 8.12 displays our segmentation results. Despite the artificial occlusion, the shape prior term was able to recover the correct shape by attracting the shape onto the *shape prior manifold*. Yet, the final surface is geometrically-accurate because the active contour can evolve freely inside the manifold $\mathcal{M}$ subject to the image term. The red-cross in Fig. 8.11 locates the final segmented shape in the embedding. Finally, note that, in practice, the shape prior term is not used during the first steps of the evolution since a robust alignment is impossible.

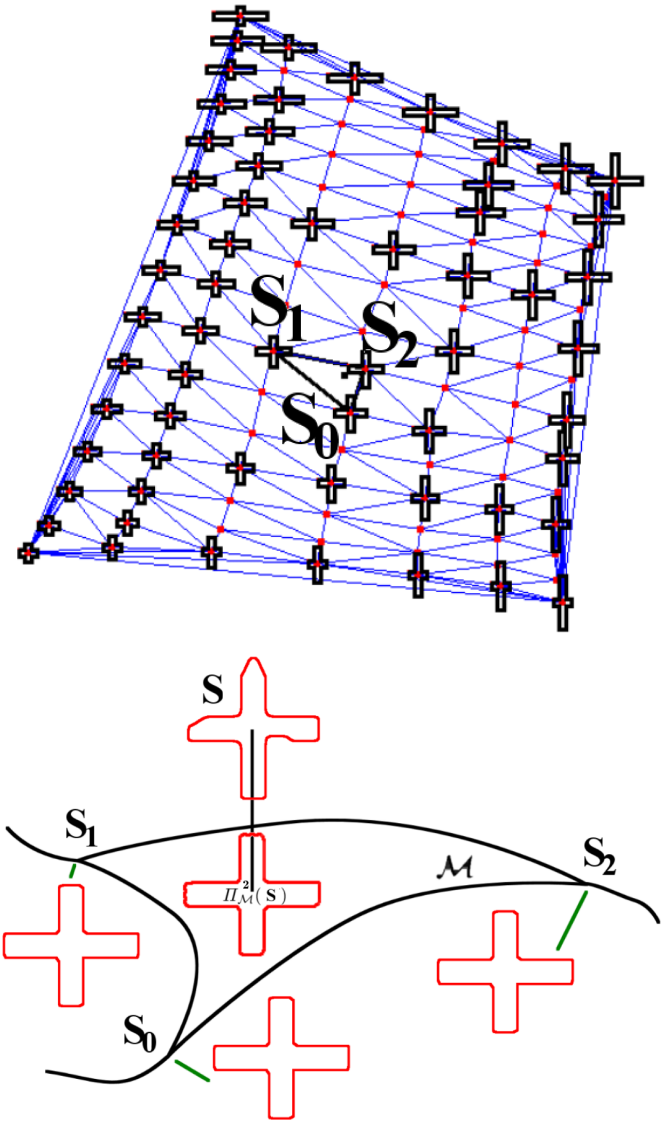

Figure 8.4: **Projection onto the *cross manifold*** using the **same embedding force** (attracting force #2): **On the top**: Reduced space. The black points is the embedding of the altered shape. **On the bottom:** Corrupted shape, its neighborhood system $\{S_0, S_1, S_2\}$, and its projection.

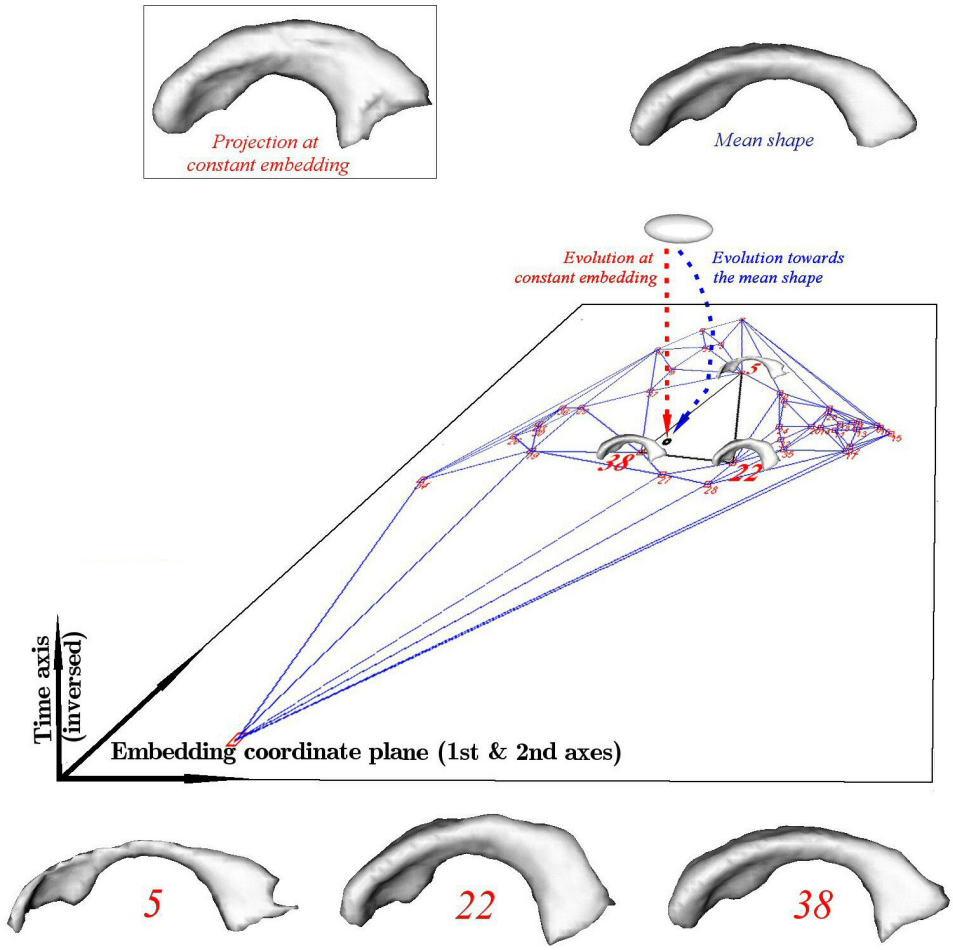

Figure 8.5: **The ventricle manifold**: Comparison of the evolution toward the Karcher mean shape $\Pi^2_{\mathcal{M}}(\cdot)$ (in blue) and the evolution at constant embedding(in red). The plane is represents the embedding coordinates of the 1^{st} and the 2^{nd} axes.

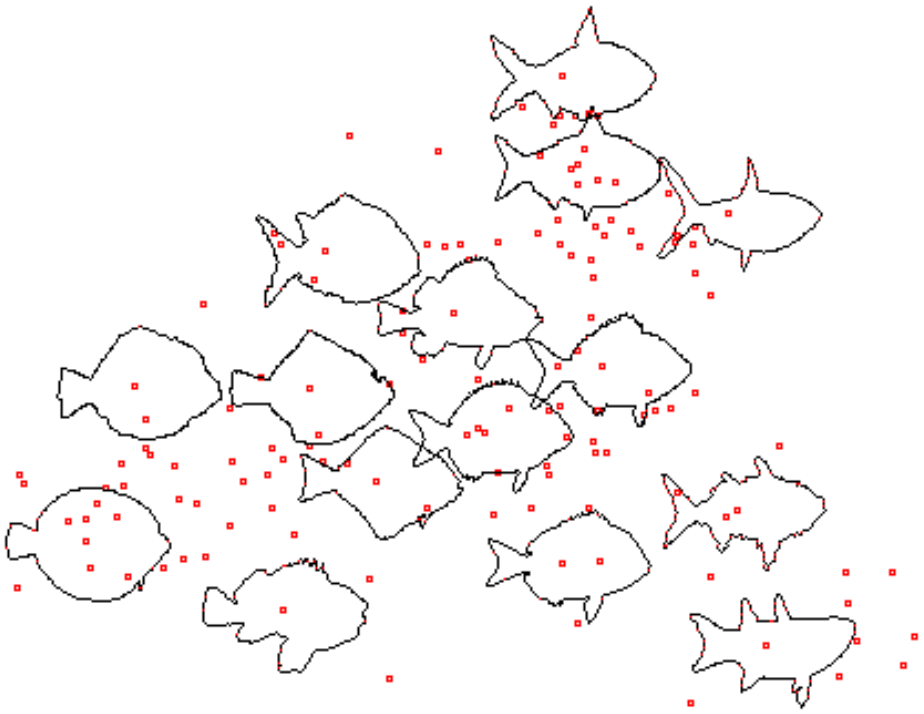

Figure 8.6: **The fish embedding:** 2-dimensional representation of 150 fishes [SQUID database]

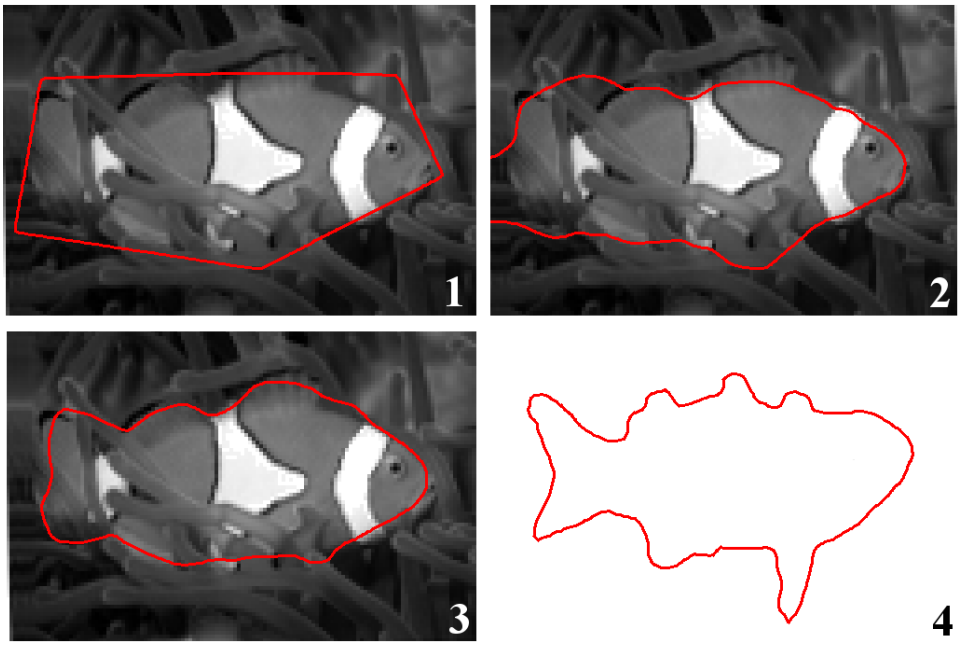

Figure 8.7: **Fish segmentation - attracting force # 2** **1**: initial contour **2**: active contour without shape prior **3**: active contour with shape prior **4**: reprojection of the final result on the *shape manifold*

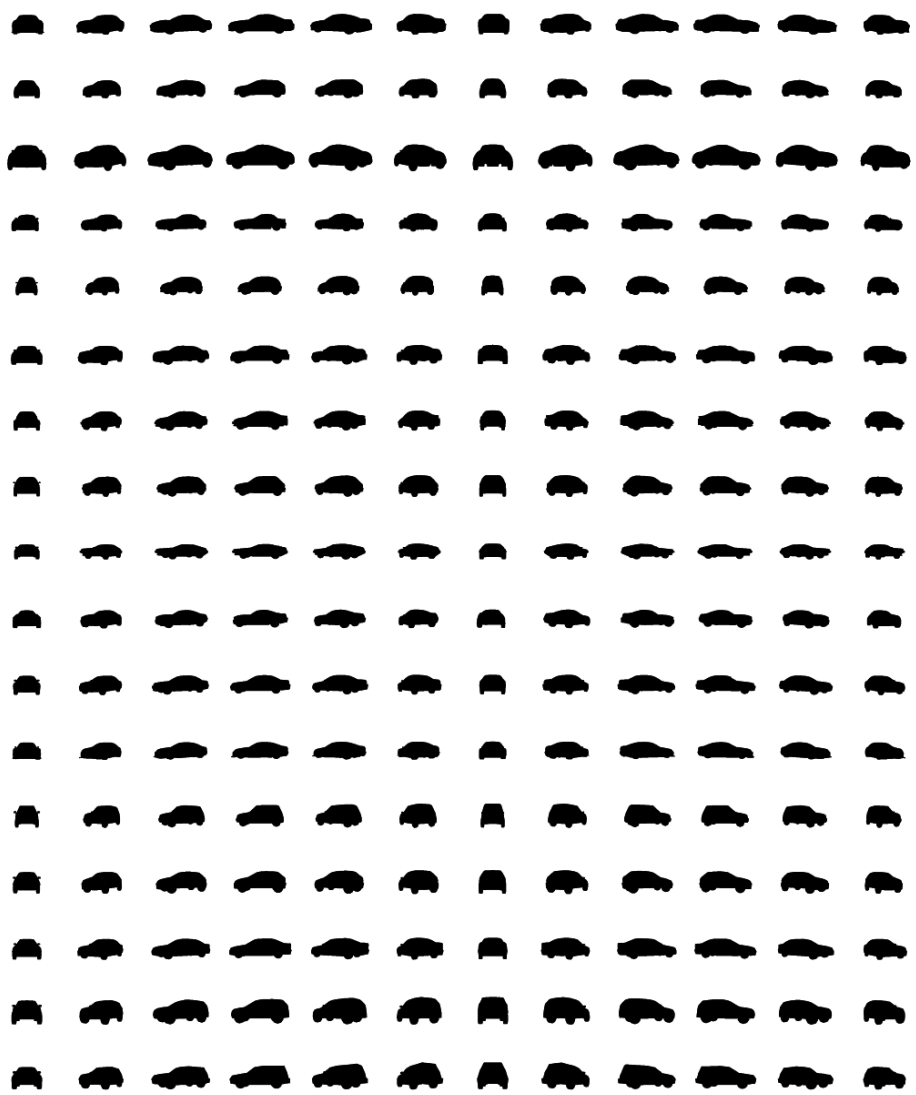

Figure 8.8: **Training set of the car manifold** is made of 192 car shapes. It is comprised of 17 different cars (1 row $\leftrightarrow$ 1 car) : Audi A3, Audi TT, BMW Z4, Citroën C3, Chrysler Sebring, Honda Civic, Renault Clio, Delorean DMC-12, Ford Mustang Coupe, Lincoln MKZ, Mercedes S-Class, Lada Oka, Fiat Palio, Nissan 200sx , Nissan Primera , Hyundai Santa Fe and Subaru Forester. For each car, 12 views are taken while turning around (1 column $\leftrightarrow$ 1 view). Although we do complete a full turn, the manifold does not result in a spherical topology.

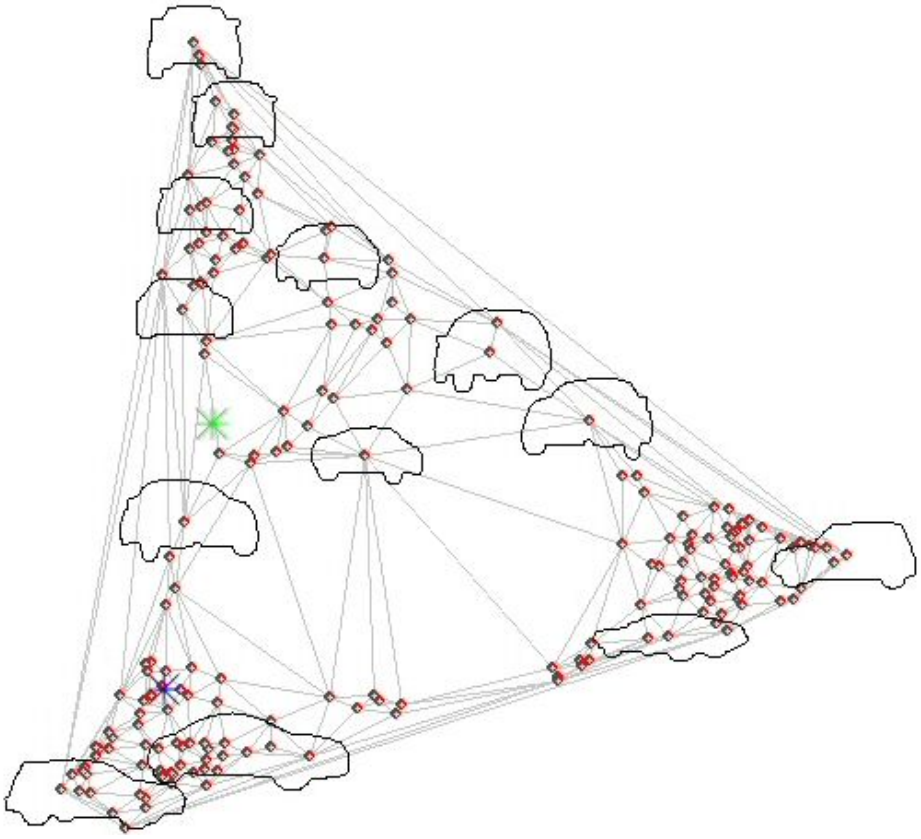

Figure 8.9: **Reduced space of the car data set and its Delaunay triangulation.**

Figure 8.10: **Segmentation with shape priors (car manifold), attracting force # 3** - Segmentation of a Peugeot 206 (left) and a Suzuki Swift (right). First row: Segmentation with data term only. Second row: segmentation with our shape prior. The embedding of the final shape is denoted by a blue cross and a green cross respectively for the Peugeot 206 and the Suzuki Swift in Figure 8.9

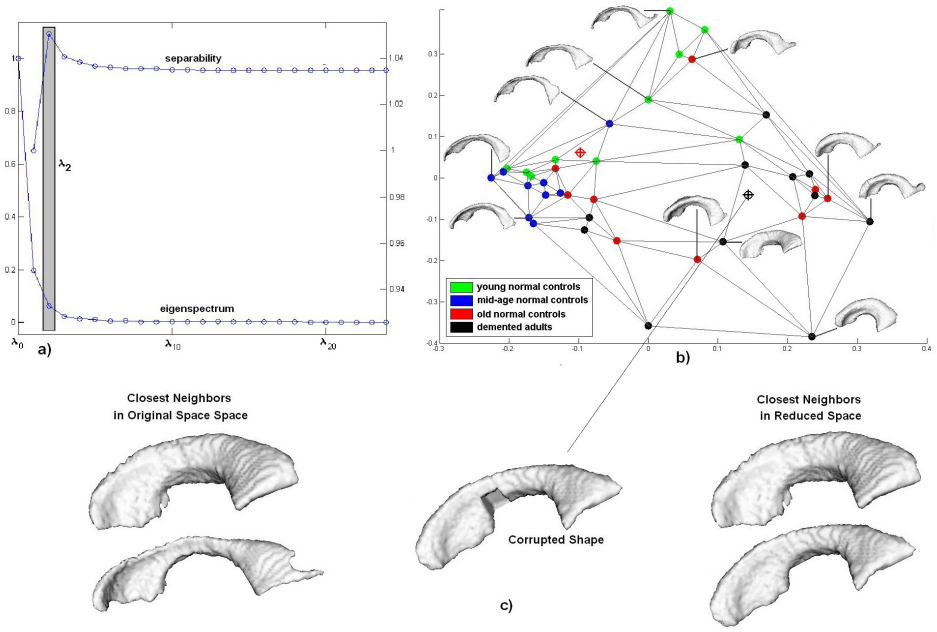

Figure 8.11: **Segmentation with shape priors (ventricle manifold), attracting force # 3, 1/2** - a) Eigenspectrum profile and degree of separability: on this restricted data set with 39 shapes only, $m = 2$ appears to be the optimal dimension. b) The two-dimensional embedding partitioned by a Delaunay triangulation. c) A manually altered shape and its two closest neighbors in $\mathbb{S}$ and in the reduced space: visually, the ones in the reduced space appear more similar.

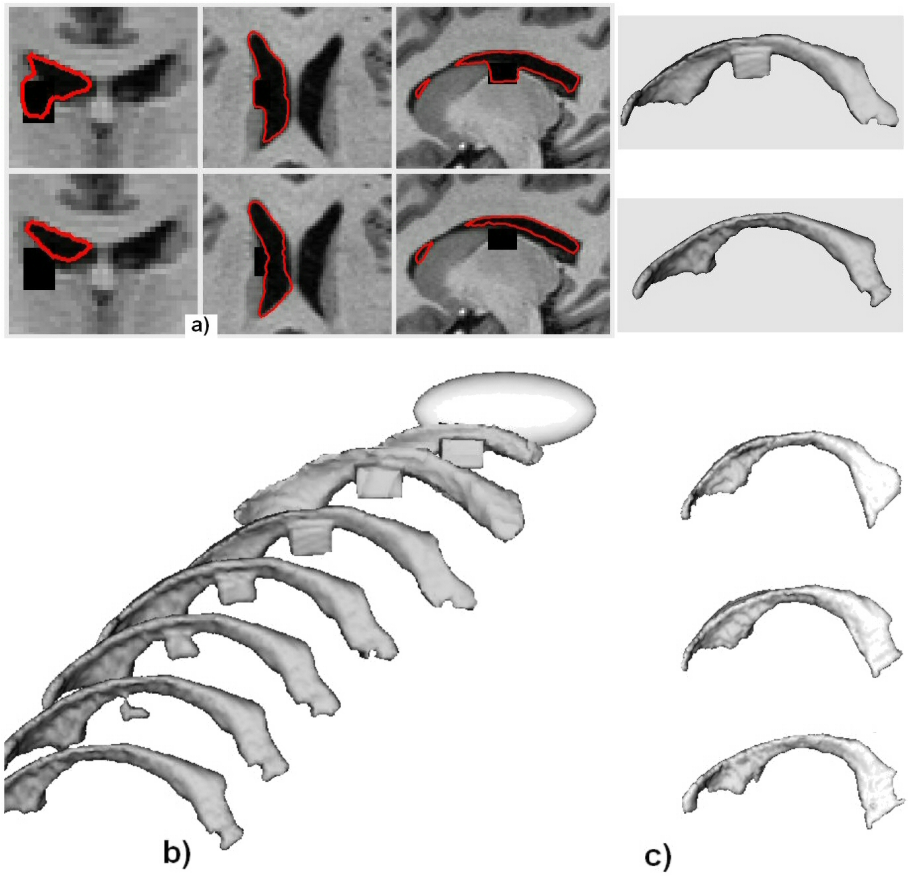

Figure 8.12: **Segmentation with shape priors (ventricle manifold), attracting force # 3, 2/2** - a) Coronal, horizontal, and sagital slices of the MRI volume with the final segmentations without (top) and with (bottom) the shape prior. b) Some snapshots of the shape evolution - the shape prior term was not used during the first steps. c) The closest neighbors of the final surface.

Chapter 9

Conclusion

In this work, not only do we fulfill our initial goal of designing non-linear shape priors for image segmentations, but we also proposed a manifold denoising algorithm. In addition, we developed a new approach to the problem of *interactive image retrieval*.

More precisely, we introduced a framework to work on general shapes in the context of manifold learning techniques and provided a solution to the interpolation problem between shape samples. We introduced three forces with different properties, designed to attract shapes toward *shape manifolds*. We formulated these forces in a generic framework since it can be applied to more general data than shapes. We sum up the three *attracting forces*:

- **Closest projection force:** We defined a projection operator onto the *shape manifold* and suggested estimating the projection of a random shape by means of an efficient iterative process.
  The projection onto the manifold minimizes the distance to the random initial shape. It is finally used as a target while evolving toward the *shape manifold*, resulting in the *closest projection force*.

- **Same embedding force:** the *attracting force* #2 calculates a local weighted Karcher mean shape that has the same embedding of the random initial shape. Again, the projection is used as a target to attract shapes of a given embedding, resulting in the *same embedding forces*. This low computational force is however not relevant in the image segmentation framework.

- **Constant embedding force:** This is an elegant formulation to attract shapes toward a *shape manifold*. Our prior is fully integrated into the Deformable model framework. Based on diffusion maps, the Nyström extension and a Delaunay partition in the reduced space, this force deforms the shape toward the manifold at constant embedding value.

We have also proposed a new deformable model framework for image segmentation that incorporates general non-linear shape priors by learning a shape prior manifold and by modeling the *attracting force* #3 . Our approach carefully exploits the properties of diffusion maps to derive an innovative shape prior term designed to attract an active contour toward the shape manifold. We demonstrated the strength of our approach by applying these ideas in different experiments (chapter 8) either with synthetic or real data, including in segmentation tasks.

We stress the fact that the proposed method is general and is not necessarily restricted to specific shape representations or 2D/3D-dimensional segmentation tasks. In particular, the only requirement is a differentiable distance (and differentiable kernel). Our approach can be applied to more general data sets, such as diffusion weighted images as well as combined anatomical and functional Magnetic Resonance Imaging (MRI) in medical imaging.

Finally, we described a variational solution for *manifold denoising* as illustrated in section 8.1 (fig. 8.1) based on the *attracting force* #3. Some simple experiments demonstrated the potential of our approach, although more experiments are required.

The research presented in this dissertation introduces a new general framework to deal with projections onto manifolds, based on dimensionality reduction techniques, and their applications to image segmentation with general non linear shape priors. However, note that our approach is limited to open manifolds and further research on this topic should alleviate this problem. Future works may also include new applications, not limited to segmentation tasks, that exploit the concepts presented in this dissertation. Yet again, it is expected to use more general data since the only requirement to apply our method is a differentiable distance. Finally, a promising direction would be to achieve non rigid registration in the shape manifold.

Chapitre 10

Conclusion (Version Française)

Dans ce travail, nous avons atteint notre objectif initial de construire des *a priori* non linéaires pour la segmentation d'images, mais nous avons aussi introduit un algorithme de débruitage de variétés. Nous avons également été impliqué dans le développement d'une nouvelle méthode de *contrôle de pertinence en recherche d'image*

Plus précisément, nous avons introduit un nouveau cadre pour utiliser des formes dans le contexte des techniques d'apprentissages de variétés et nous proposons une solution pour interpoler entre les échantillons de forme. Trois forces avec différentes propriétés, ont été construites pour attirer les formes vers les *variétés de formes*. Ces forces sont formulées dans un cadre générique puisque elles peuvent être appliquée à des types de données plus généraux. Nous résumons ces trois forces :

- **Force vers la projection la plus proche** Nous avons défini un opérateur de projection sur la *variété des formes a priori* et suggéré son estimation rapide au moyen d'un processus itératif.
  La projection minimise la distance à la forme corrompue, et peut être utilisé en tant que cible pendant l'évolution vers la *variété des formes*.

- **Force vers la projection de même valeur de plongement** La *force attractive* #2 calcule une moyenne locale pondérée de Karcher de forme qui a la même valeur de plongement que la forme initiale à projeter. La projection est de nouveau utiliser comme une cible pour attirer l'ensemble des formes ayant une valeur de plongement donnée. Cette force ayant un faible coût

n'est cependant pas pertinente dans le cadre de la segmentation d'image.

– **Force à plongement constant** Il s'agit d'une formulation élégante pour attirer les formes vers la *la variété des formes a priori*. En particulier, cette approche qui est complètement intégrée à la technique des *diffusions maps* (DFM), repose sur l'extension de Nyström et une partition de Delaunay. Cette force #3 assure que l'évolution se fait à valeur de plongement constant.

Nous avons également proposé un nouveau cadre de modèle déformable pour la segmentation d'image avec des a priori de forme. Pour cela, la variété des formes a priori est apprise et la *force attractive* #3 est utilisée comme un terme d'énergie d'a priori de forme. Notre approche exploite soigneusement les propriétés des *diffusion maps* afin de construire un terme d'a priori de forme innovant, qui attire un contour actif vers la variété des formes a priori. Nous avons démontré la force de notre approche en appliquant ces idées dans différentes expériences (chapitre 8), avec des données synthétiques ou réelles, y compris dans des tâches de segmentation.

Nous insistons sur le fait que la méthode proposée est générale et n'est pas nécessairement restreinte à des représentations spécifiques de forme ou à des tâches de segmentation en dimension 2D/3D. En particulier, seulement une distance et un noyau différentiable sont nécessaires. Notre approche peut être appliquée à des données plus générales telles que les images de diffusion pondérées ou IRMf (Imagerie de Résonance Magnétique fonctionnelle) en imagerie médicale.

Enfin, nous avons décrit une solution variationnelle pour le *débruitage de variété* comme illustré en section 8.1 (fig. 8.1) reposant sur la *force attractive* #3. Quelques expériences basiques ont montré le potentiel de notre approche, bien que d'autres seraient nécessaires.

Les recherches présentées dans ce manuscrit introduisent un nouveau cadre général pour projeter des données sur des variétés, reposant sur des techniques de réduction de dimension, et les applications qui en découlent en segmentation d'image avec des a priori de forme non linéaire généraux. Cependant, notre approche est limitée à des variétés ouvertes et d'autres recherches sont nécessaires pour résoudre ce problème. Des recherches futures pourraient s'orienter vers de nouvelles applications, non limitées à des tâches de segmentation, qui exploitent les concepts pré-

sentés dans cette thèse. De nouveau, le prochain objectif est d'utiliser des données plus générales puisque seulement un noyau différentiable est nécessaire. Finalement, une direction de recherche prometteuse serait de développer une méthode de recalage non rigide dans une variété de forme apprise à l'aide d'un échantillon de formes.

Appendix A

Radon/Hough space for pose estimation

Nota Bene: This work was achieved during my thesis under the supervision of Nikos Paragios when he was at Ecole des Ponts (CERTIS Lab.) and with the partial collaboration of Yakup Genc, Program manager at Siemens Corporate Research in Princeton New-Jersey, USA. It is about 3D reconstruction and camera calibration, which is not the main topic of this dissertation. However, since this project relies on machine learning and was my first step using these techniques , it seems natural to include it for the sake of completeness.

Abstract

In this chapter, we present a method for camera pose estimation from one single image in a known environment. This method comprises two stages, a learning step and an inference stage where given a new image we recover the exact camera position. We focus on lines and the Radon/Hough transform to model features. The question to be answered is what can be learnt from lines in order to estimate the camera position ?.

Lines that are recovered in the Radon space consist of our feature space. Such features are associated with [AdaBoost] learners that capture the wide image feature spectrum of a given 3D line. Such a framework is used through inference for pose estimation. Given a new image, we extract features which are consis-

tent with the ones learnt, and we associate such features with a number of lines in the 3D plane that are pruned through the use of geometric constraints. Once correspondence between lines has been established, pose estimation is done in a straightforward fashion. Promising experimental results based on a real case are presented in this chapter.

Contents

Original contributions

We propose a new approach to one-shot fast pose estimation based on a machine learning technique.

Related publication: ICPR'06 [79]

A.1 Introduction

Pose estimation has been extensively studied in the past years. Nevertheless, it is still an open problem particularly in the context of real time vision. Robot navigation, autonomous systems and self-localization are some of the domains in computational vision where pose estimation is important. One can also cite a number of applications in augmented and mixed reality where a solution to this problem is critical. In prior literature pose estimation methods are either feature-driven [167] or geometry-driven [4, 71, 154, 44].

The solution proposed aims to combine feature-based methods and geometry-driven approaches. To this end, we consider geometric elements such as lines to be the most appropriate feature space. Such a selection is motivated from a number of reasons. Lines are simple geometric structures that refer to a compact representation of the scene, while at the same time one can determine angles and orientations that relate their relative positions. Parallel to that, in the image projection space appropriate feature spaces (Hough [72, 207], Radon [207]) and methods exist for fast extraction and tracking [66] of such geometric elements with important precision.

A promising solution is both feature-and-geometry driven. Lines are characterized by their projection in the Radon space, forming a feature space. In addition, the geometry of 2D-line configuration can be easily recovered through a 3D reconstruction of the scene. The scheme of our method is thus to reconstruct line while their geometry and features are learnt. Once this is done, a simple line detector coupled with the information previously learnt can be implemented in order to infer the pose estimation from a single view. We are aiming at real-time applications such as augmented reality based on a head mounted device or robot navigation.

The remainder of the document is organized in the following way. In section A.2, we present basics of line detection based on the Hough and Radon transform. A matching and tracking process are also presented in section A.2.3. Then, section A.3 details our approach, the feature space being based on the Radon space. Experimental results and discussion are finally presented in the last section.

A.2 Feature detection, matching & tracking

The detection of primitives in images is a recurrent problem in computer vision, particularly for points and lines. Feature detection is a key point of our approach and we will only be interested in line extraction in the sequel. In this section, we present one of the most powerful tools to robustly detect lines in images, the *Hough transform*. Nevertheless, the voting space of the Hough transform has some discretization defects that might be unsatisfactory.To cope with such limitations, the *Radon transform* may be employed. We also point out the links between the *Hough transform* and the *Radon transform* up.

A.2.1 The Hough transform

The *Hough transform*, which has been the purpose of a lot of research since the 60's, is a method capable of finding parameterized shapes in images. The idea of this transform is to express a mapping between an image space and a parameter space which constitutes a dual space. Obviously, the parameter space depends on the shape of the primitive we work on. In its initial forms, the *Hough transform* [103, 163] was designed to detect only 2-lines[1] in binary images (A binary image is, for instance, obtained from the edge map of a given image, see (sec. 4.2.1)). Paul V.C Hough [103] chose the slope and the intercept as parameters of the line. We will be using the projective representation in the rest. Let a be the slope and b be the intercept of a given line whose projective representation is $l = [a, 1, -b]^T$. Also, let $p = [x, y, 1]^T$ be a point. The method relies on the principle of duality in projective geometry.

$$l^T p = 0 \tag{A.1}$$

Equation (A.1) has two dual interpretations [85]:

1. line l goes through point p if equation (A.1) is verified
2. point p belongs to line l if equation (A.1) is verified

Finally, let $I \subseteq \mathbb{R}^2$ be the image space, $P \subseteq \mathbb{R}^2$ be the parameter space (for instance, the x-axis represents the slope and the y-axis represents the intercept). By using the duality principle, we see that a point in I corresponds to a line in P. We also notice that the set of points belonging to a given line $l^* = [a^*, 1, -b^*]^T$ in the

[1] n-line: line in a n dimensional space

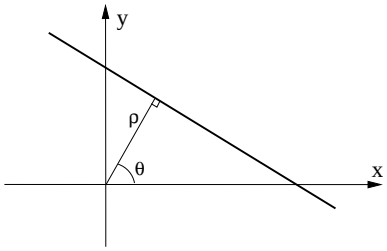

Figure A.1: **Parametrization of the Hough space**

image space I corresponds to the set of lines intersecting at the coordinates (a^*, b^*) in the parameter space P.
In practice, an accumulator array of the size of the parameter space is set up to zero and each point p in the image space votes for the cells corresponding the lines going through p. The line detection is finally achieved by fixing a [detection] threshold in the accumulator array.
More recently, A. S. Aguado, E. Montiel and M. S. Nixon [3, 21] have formalized and generalized the relationship between the principle of duality and the Hough transform, not only to projective geometry.

The line slope parametrization is not optimal because both parameters are not bounded. Richard O. Duda and Peter E. Hart [73] proposed a commonly accepted line parametrization, which circumvents such a limitation. The authors wrote the line l as

$$l = \{(x, y)\,, x\cos(\theta) + y\sin(\theta) - \rho\} \tag{A.2}$$

where the two parameters θ and ρ are respectively the angle of its normal and the distance to the origin of l as represented in figure (A.1). If we choose to restrict θ to $[0, \pi]$, ρ is an algebraic distance, otherwise, $\theta \in [0, 2\pi]$ and ρ is an absolute distance. This parametrization, which is unique, maps a point in image space to a sinusoid in the parameter space. Figure (A.2) shows an example of the Hough transform on a very simple example.

The *Hough transform* as described so far is known as the Standard Hough Transform (SHT) and belongs to a classification called *one to many* $(1 \rightarrow m)$. Each point produces indeed a bench of points in the parameter space. Although the *Hough transform* is robust, it is very costing from a computational point of view,

Figure A.2: **Example of Hough Transform**: Image space on the left, parameter space on the right. The three highest values of the parameter space represented by an accumulator give the parametrization of the 3 lines in the image space

particularly when the dataset of points in the image space is large. In order to reduce the time of detection, N. Kiryati and Y. Eldar and A. M. Bruckstein [113] have proposed the *Probabilistic Hough Transform* (PHT) that selects a poll of sample in the image space . They speed up the process by using probabilities and by achieving a kind of "coarse to fine" *Hough transform*. This idea has been extended in [93].

Another classification, called *many to one* $(m \rightarrow 1)$, was introduced by Lei Xu and Erkkii Oja: the *Randomized Hough Transform* (RHT) [213]. Rather than taking a single point in the image space, the authors prefer to compute only one point in the parameter space by taking randomly several points in the image space. For the case of a line, two random points define a line and so, vote for one point in the parameter space. When a threshold is reached in the parameter space, the corresponding line is detected and masked out of the image space. The algorithm starts again until it does not find any line after a certain number of polls.
The PHT and RHT have been unified later by H. Kalviainen, N. Kiryati and S. Alaoutinen in [106]. The reader is also referred to [124, 186] for more details.
Note that the *Hough transform* has been widely extended to more general shapes, even in higher dimensions.

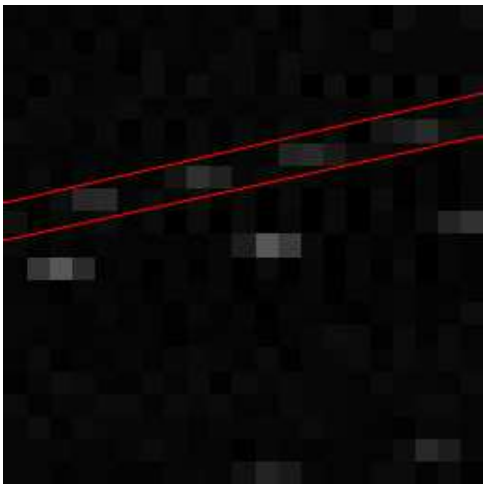

Figure A.3: **Discretization defects** using standard Hough transform between the two red lines

Unfortunately, the SHT parameter space has discretization defects as shown in figure (A.3) in the stripe between the two red lines. The *Radon transform* [159] does not suffer from such a defect and can be efficiently implemented by using the Fourier Transform. Both transformations, *Hough transform* and *Radon transform* are derived from the same concept and the output spaces are the same when the Radon space is computed on the edge map.

A.2.2 The Radon transform

Let g be a mapping defining an image over a domain space U such that:

$$\begin{aligned} g: \quad U &\longmapsto \quad \mathbb{R} \\ u &\longrightarrow \quad g(u) \end{aligned}$$

and let $f_{\mathbf{p}}(u) = 0$ define a shape described by the vector parameter $\mathbf{p}$. The Radon transform of g regarding the shape $f_{\mathbf{p}}(u) = 0$ is given by:

$$\mathcal{R}(g)(\mathbf{p}) = \int_U g(x)\delta\left[f_{\mathbf{p}}(x)\right] du \tag{A.3}$$

where $\delta(.)$ is the Delta-Dirac function. The discrete form of the *Radon transform* is extensively used in tomography image reconstruction but it can also be very useful

for line detection.
In that particular case, $U = \mathcal{R}^2$ ie $u = (x, y)$ and let $\mathbf{p} = (\rho, \theta)$ such that:

$$f_{\mathbf{p}=(\rho,\theta)}(x, y) = \rho - x\cos(\theta) - y\sin(\theta) \tag{A.4}$$

and thus, equation (A.3) can written as follows.

$$\mathcal{R}(I)(\rho, \theta) = \iint_{\mathbb{R}} I(x, y)\delta\left(\rho - x\cos(\theta) - y\sin(\theta)\right) dxdy \tag{A.5}$$

where $g = I$ is the transformed image.
Finally, local maxima are thresholded by using the median value of neighboring pixels.

A.2.3 Tracking / Matching lines in the Radon space

Basic image to image tracking

Local maxima in the parameter space correspond to lines in the image space and can be extracted in a straightforward fashion. The *Radon transform* is a global transformation that encodes the entire line structure in a compact fashion. It is capable of accounting for occlusions, local and global changes of the illumination as well as strong presence of noise.

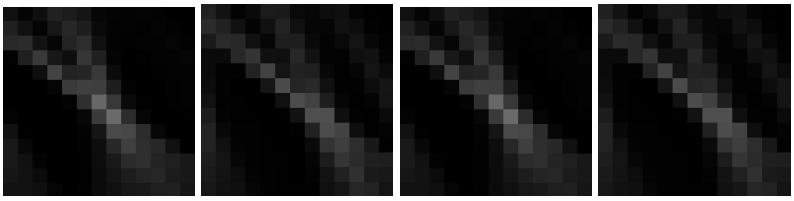

Figure A.4: **Line signature in the Radon space** for a given number of consecutive images.

Tracking lines is a feasible task by capturing the line displacement from one image to the next in the parameter space. The problem is indeed simplified due to the constraint that lines correspond to local maxima — in the parameter space — and therefore simple comparison between local Radon patches could provide explicit correspondences between lines. To this end, we consider simple normalized correlation criteria. We seek to recover the optimal displacement $\mathbf{du} = (dx, dy)$ between two Radon images such that the distance between the corresponding patches

is minimal. Basically, the algorithm works with the Radon spaces ($\mathcal{R}_1$ & $\mathcal{R}_2$) of two successive images (I_1 & I_2) and for each local maximum detected previously in $\mathcal{R}_1$ - i.e. a line in I_1 - it searches for the 2D-dimensional shift in $\mathcal{R}_2$ such that an energy E is minimized:

$$\min_{\substack{(dx,dy)\in \\ \Omega(X,Y)}} E(dx, dy) \tag{A.6}$$

where $\Omega(X, Y)$ is the neighborhood of (X, Y).
The search can be constrained on local maximums of $\mathcal{R}_2$ but experiments show that a free shift search is preferred. We assume that the transformation between Radon spaces $\mathcal{R}_1$ and $\mathcal{R}_2$ corresponding to two consecutive images I_1 and I_2 is slight. Thus, we simply chose to compute a cross normalized sum of differences:

$$E(dx, dy) = \frac{\sum_{u,v} \left[\left(\mathcal{W}_{X,Y} \left\{ I_1 \right\} (u, v) \right) \left(\mathcal{W}_{X',Y'} \left\{ I_2 \right\} (u, v) \right) \right]^2}{\sum_{u,v} \left[\mathcal{W}_{X,Y} \left\{ I_1 \right\} (u, v) \right]^2 \sum_{u,v} \left[\mathcal{W}_{X',Y'} \left\{ I_2 \right\} (u, v) \right]^2} \tag{A.7}$$

where $X' = X + dx$, $Y' = Y + dy$, $\mathcal{W}_{X,Y}$ is a designed window centered in (X, Y) such that the values $\mathcal{W}_{X,Y}(u, v)$ are centered (the mean over the windows is subtracted).
Note that the particular structure of the Radon space, which folds up, is taken into consideration. We tried other forms of similar energy (correlation etc.) but none showed real improvements.

Tracking over a sequence

In the previous lines, we presented a simple image to image line tracking. We are however interested in tracking lines over a video sequence. Thus, dying lines — i.e. *lines that are not present anymore in an image* — and new line detection should be taken into consideration. Without loss of generality, algorithm (1) outlines the procedure implemented to achieve such a task. It is based on three main functions: **image to image line detection**, **new line detection** and **outgoing line detection**. The former has been described previously. Then, the algorithm tries to keep up to $N_l^{\max}$ during the tracking within N^{seq} images. **New line detection** has been already detailed and is used to maintain the number of lines tracked in the current image (up to $N_l^{\max}$). The last function, **outgoing line detection** , ensures that a

line will not be tracked if it is not anymore in the current image. In order to decide if a line should be tracked in the following images, the algorithm analyses with the help of variance the patches over N images. It avoids removing and detecting again continuously the same line along the tracking within the video sequence.

Algorithm 1 Tracking lines in a video sequence

INPUTS: N, N^{seq},$N_l^{\max}$ be initialized
Initialize $\mathcal{O} = \emptyset$, $\mathcal{N} = \emptyset$, $t \leftarrow 0$
while $t + N <= N^{\text{seq}}$ **do**
  for all *line* $\in \mathcal{O}$ **do**
    Trackline(*line*,$t + N - 2$,$t + N - 1$);
  end for
  $n \leftarrow N_l^{\max} - |\mathcal{O}|$ {number of lines to detect in image}
  $\mathcal{N} \leftarrow$ detect n new lines in image t
  for all *line* $\in \mathcal{N}$ **do**
    Trackline(*line*,$\{t, \ldots, t + N - 1\}$);
  end for
  $\mathcal{O} \leftarrow \mathcal{O} \cap \mathcal{N}$
  for all *line* $\in \mathcal{O}$ **do**
    OutGoing_Detection(*line*,t,$\{t, \ldots, t + N - 1\}$);
  end for
  $t \leftarrow t + 1$
end while

Trackline(*l*,$\{a, \ldots, b\}$) is the basic tracking function of the line *l* between images a and b, in the corresponding Radon/Hough spaces (see A.2.3).

OutGoing_Detection(*l*,*t*,$\{t, \ldots, t + N - 1\}$) is the procedure that detects if the tracking of the line *l* should be stopped or not. (see A.2.3).

Conclusion

We have presented an efficient method for line tracking in the Radon space based on correlation patches. Note that the correlation patches in the Radon space can also be used in order to improve semi-automatic line matching in a sequence of images.

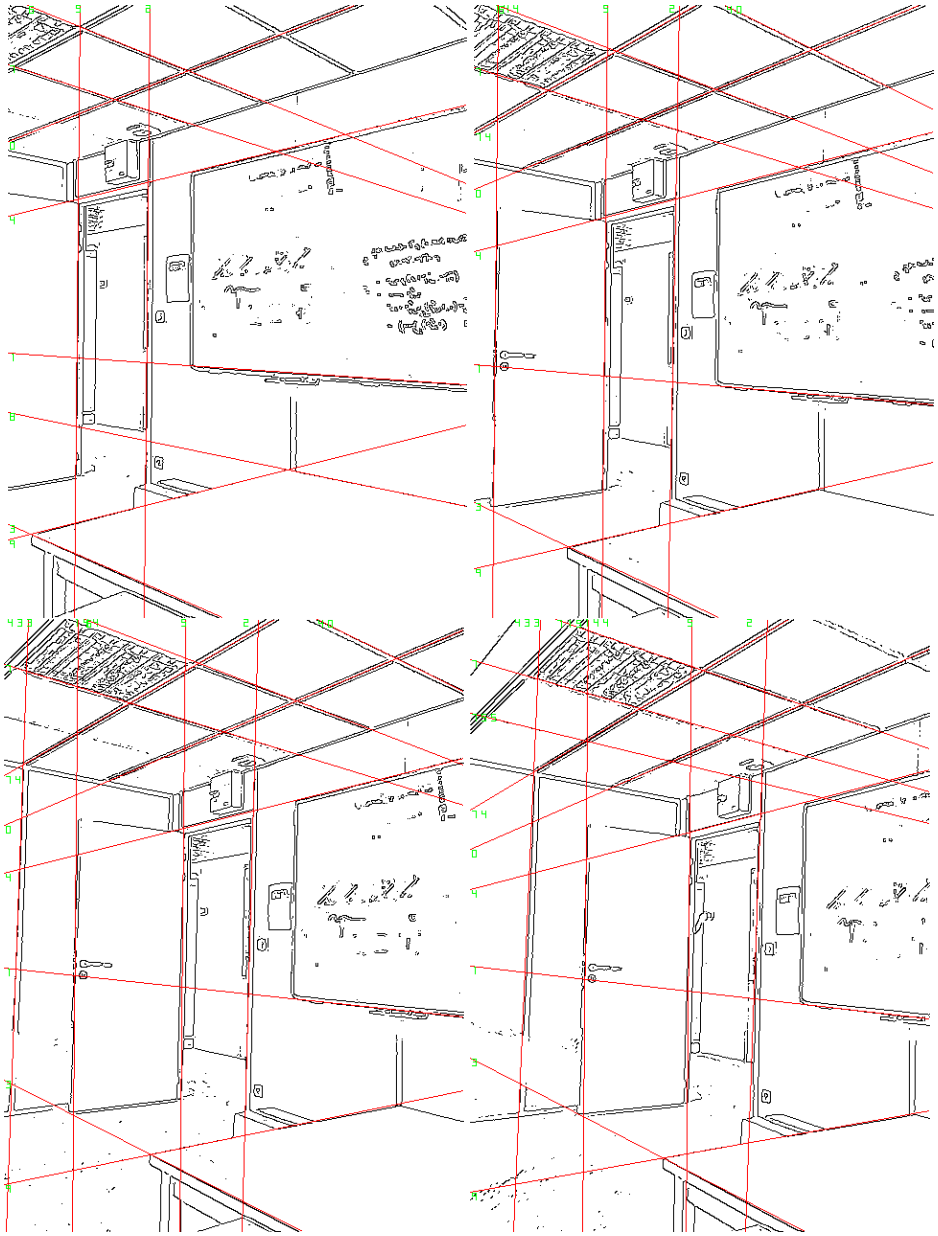

Figure A.5: **Tracking lines in the Radon space** and their projections in the corresponding image space. Results are presented in raster-scan format.

A.3 Inference from complete Hough/Radon space

A.3.1 Objectives & Problem formulation

Our method consists of a learning and an inference step. During the learning stage, the scene is learnt from an image sequence and its corresponding 3D reconstruction. A geometry-based learning is achieved by recovering geometric relations between lines and consequently between their projections. In parallel to the feature-based learning, 3D lines are associated through AdaBoost learners with their 2D projection in the Radon space (local maxima). Such an information space is used within a matching process to recover the camera's pose from a new image. Matching between plausible line candidates in a new image dictates multiple correspondences between the 2D new image lines and the 3D reconstructed lines. The most probable configuration in terms of appearance while satisfying geometric consistency constraints provides the camera position. The overview of such an approach is shown in [Fig. (A.6)].

Let us consider a centered coordinate system that is defined with the camera lens center or with the observer located at the origin. The view axis is supposed to be colinear to the z axis and perpendicular to the image plane. Using the perspective model, the image of any point in space is equal to the intersection of the image plane and the line joining the point to the center of the camera lens.

The main stream of research in 3D reconstruction and pose estimation has been

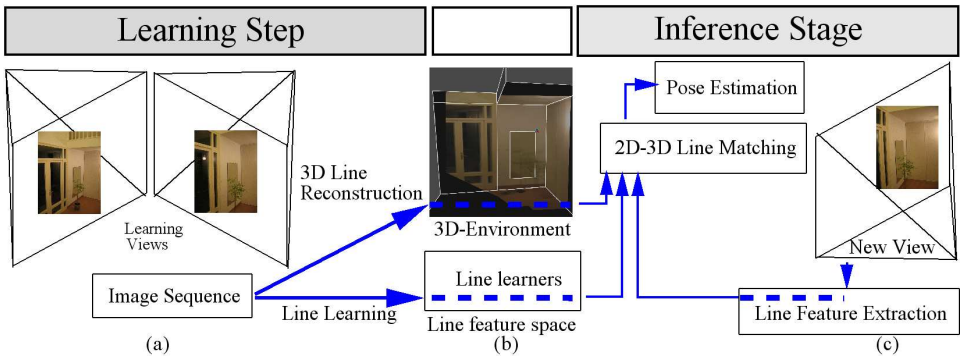

Figure A.6: **Overview of the proposed pose estimation approach** where both learning and estimation steps are delineated.

devoted to point correspondence [84, 156, 167]. Line correspondences could be an efficient alternative to such an approach [126, 120]. Such a feature space inherits the advantage of being more robust than point correspondences as well as more global. On the other hand, line tracking algorithms are computationally intensive and low sampling frequency and long time delays can therefore be expected. Such a limitation can be addressed using image transformations like the Hough [72] and the Radon space [207] which projects images into convenient line spaces.

The remainder of the current section is organized in the following way: Section (A.2.3) described our approach toward line tracking. Since lines are tracked through a video sequence, their 3D information is recovered and the 3D-2D relation can be indirectly learned by a boosting algorithm as presented in section (A.3.2). Section (A.3.3) is devoted to inference and pose estimation and finally a discussion for this approach is given in the conclusion of this appendix in section (A.4).

A.3.2 3D-2D Line Relation through Boosting

As previously mentionned, the first step is to compute a three dimensional model of the scene and more particularly the lines. 3D reconstruction from image sequences of video sequence has been widely studied in the past years. Figure (A.7) is a 3D reconstruction of an indoor scene based on lines. Once the scene and 3D lines have been reconstructed (Fig. A.7), one would like to establish a connection between such 3D lines and their corresponding projections. Since our approach is both features and geometric-based, we aim at learning both kinds of constraints.

First, geometrical constraints can be straight and naturally deduced from the 3D reconstructed scene implying 2D constraints on the projected lines. Since extraction of the relative geometry is not critical —once 3D reconstruction has been completed— more attention is to be paid to feature extraction, learning and modeling.

Let us consider that our feature learning stage consists of $\mathcal{L} = \{l_1, l_2, ..., l_n\}$ 3D lines, and our training consists of c images. Without loss of generality we assume that such geometric elements were successfully detected within these c images. Let $\mathcal{P}_k = \{p_k^1, p_k^2, ..., p_k^c\}$ being the projections in the radon space of line l_k at these c images. Such projections correspond to the 2D local radon patches represented as d-dimensional vectors.

Traditional statistical inference techniques can be used to recover a distribu-

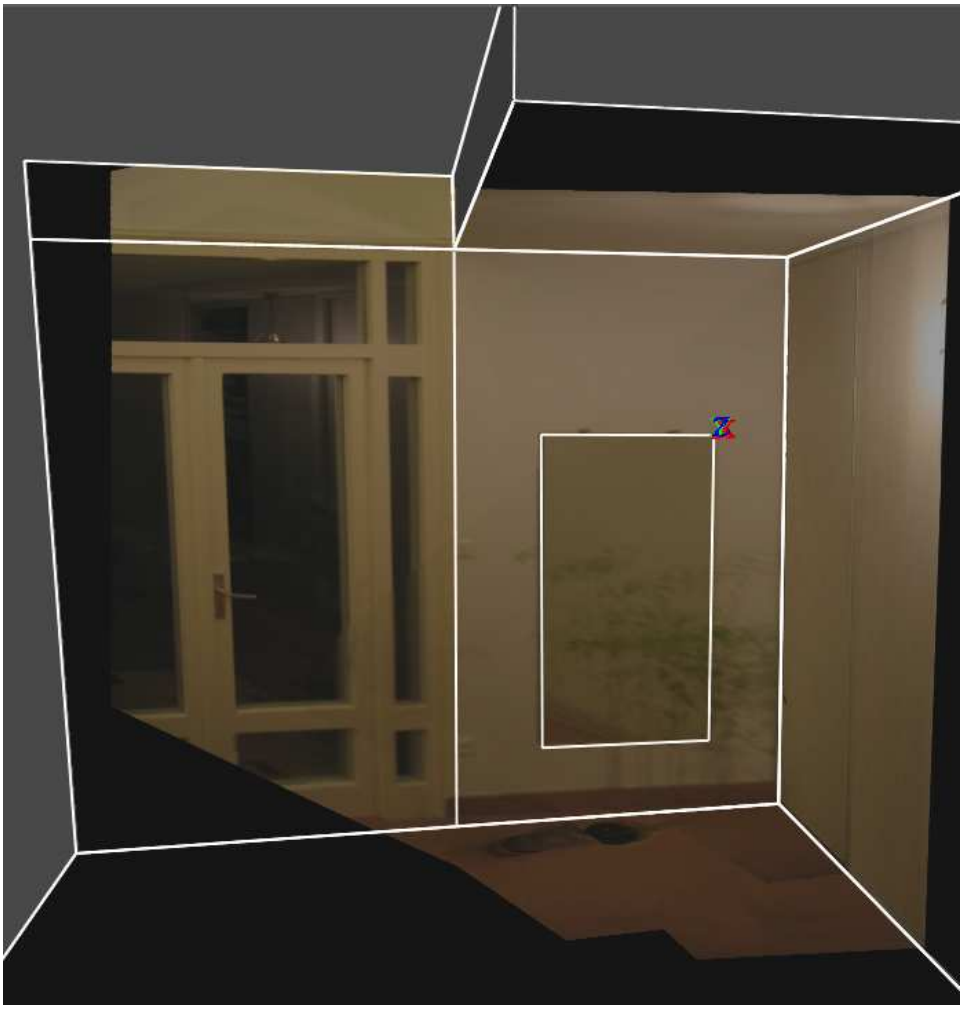

Figure A.7: **3D reconstruction of an indoor scene**

tion of such d-dimensional vectors. To this end, one can consider simple Gaussian assumptions and classic dimensionality reduction techniques like principal component analysis. Such a selection could fail to account for the highly no-linear structure of the Radon space and so of the corresponding features. Furthermore, since recovering a training shots from all possible virtual positions of the observer is almost impossible, one should also account for sparse observations and learning from small training sets. Therefore, more advanced classification techniques are to be considered which are able to cope with some of the above limitations.

Our basic classifier consists of: given two classes $\mathcal{C}_1$ and $\mathcal{C}_2$ find an appropriate transformation/function F that can measure the distance between a sample p and these classes $F(C_k, p)$. To this end, within the context of our application one can consider n bin classification problems F_k,

$$F_k(p) = \begin{cases} 1, & p \in C_k \\ 0, & p \in C_j,\ j \neq k \end{cases}$$

In other words, we are looking for a way to compute the boundary of a binary partition between the features corresponding to line l_k versus the others. Stump classification can deal with this problem: it tests binary partitions along all the d dimensions and all possible thresholds. The model is given by:

$$\begin{aligned} R = &\{\alpha_0 \mathbb{1}_{x_j < \tau} + \alpha_1 \mathbb{1}_{x_j \geq \tau} : j \in 1, \ldots, d, \tau \in \mathbb{R}, \\ &\alpha_0 \in [0; 1], \alpha_1 \in [0; 1]\} \end{aligned} \tag{A.8}$$

The threshold τ^* and the dimension j^* that minimizes the desired criteria $\mathcal{W}(j, \tau)$ are kept to form the partition parameters. The reader can refer to [10] to get further details about stumps and more particularly about the criteria $\mathcal{W}$ we used.

Consequently, stump classification returns a function f_m that defines a partition of the space according to a hyperplane which is orthogonal to the canonical basis of $\mathcal{X}$:

$$f_m = f_{m,<} \mathbb{1}_{x \in \mathcal{X}^{<}_{j,\tau}} + f_{m,\geq} \mathbb{1}_{x \in \mathcal{X}^{\geq}_{j,\tau}}$$

Stumps were implemented and tested with a synthetic data set formed with a video sequence of a basic 3D structure. Towards producing a realistic test case, a set of perturbations (random lines) are introduced in the 3D scene [Fig. (A.10)]. In order to account for possible sensor noise, the corresponding video images are convolved with a Gaussian operator (white noise), and additional lines are also added in the observation set. Radon transformations of such images are used to

recover local patches that guide the learning step while a test set is also created. The classification error for all experiments which were conducted was close to 0.5. Therefore, one can conclude that bin classification on such a space induces a high risk in pose estimation.

One can overcome such a limitation through the transformation of the stump classifier into a "weak" learner. In addition, the learning algorithm should determine the origin of the sample as accurate as possible by the use of a multitude of "weak" learners. AdaBoost [89, 90] is one of the most prominent techniques to address such a task among others such as neural networks [72] and support vector machines [208]. Boosting improves significantly the accuracy of any given learning algorithm, even in the case of a "weak" learner. Such techniques have a number of interesting empirical properties. It has been shown [90] that boosting does not perfom an overfit to the training data.

The general idea of boosting is to **1-** repeatedly use a "weak" learner [stumps returning a regression function f_m in our case] with some weights w_i^m on the training data - *m being the iteration index* - **2-** focus on misclassified data from one iteration to the next through the update of w_i^m:

$$w_i^m = \frac{w_i^{m-1} e^{-Y_i f_m(X_i)}}{\mathrm{K}} \quad \begin{array}{l} \forall i \in \{1,\ldots,N\} \\ \text{K: normalizing constant} \end{array} \tag{A.9}$$

where Y_i is the classication corresponding to the feature X_i, (X_i, Y_i) being an element of the learning and N its size.

Then, at each step a weight c_m associated with the current learner is determined according to the corresponding classification performance. The final classification is given by the thresholded regression function $\mathbb{1}_{G_M(x)>T}$, $G_M(x)$ being the weighted combination of the "weak" learners:

$$G_M(x) = \sum_{m=1}^{M} c_m f_m \tag{A.10}$$

This is a slightly modified version of the "real AdaBoost algorithm" [175, 10] presented in figure (A.8). Indeed, at the end of the "real" AdaBoost algorithm, the decision is based on $\mathbb{1}_{G_M(x)\geq\frac{1}{2}}$ implying implicitly a fixed threshold on the regression function $G_M(x) = \sum_{i=1}^{M} c_i f_i$. Instead of doing this and since $G_M(x)$ is by definition piece-wise constant, we preferred to choose dynamically the threshold T among the finite set of possible values so that the error can be decreased.

Finally, the feature learning stage outputs n classifiers

$$\mathcal{S}^n = \{\mathbb{1}_{G_M^1(x)>T_1}, \ldots, \mathbb{1}_{G_M^k(x)>T_k}, \ldots \mathbb{1}_{G_M^n(x)>T_n}\}$$

-one for each line- that are going to be used for line inference and pose estimation.

(X, Y) forms the learning set, $X \in \mathcal{X} = R^d$ is a feature and Y the corresponding true classification decision
Start with weights $w_i^0 = \frac{1}{N}$ for any $i \in \{1, \ldots, N\}$.
For $m = 1$ **to** M **do**

- Determine $j \in \{1, \ldots, N\}$ and $\tau \in \mathbb{R}$ minimizing $\mathcal{W}_{w^{m-1}}(j, \tau)$.
- Choose $f_m = f_{m,<}\mathbb{1}_{x \in \mathcal{X}_{j,\tau}^<} + f_{m,\geq}\mathbb{1}_{x \in \mathcal{X}_{j,\tau}^\geq}$ where

$$\begin{cases} f_{m,<} & \triangleq & \frac{1}{2}\log\left(\frac{\mathbb{P}_{w^{m-1}}(Y=1; X \in \mathcal{X}_{j,\tau}^<)+\beta}{\mathbb{P}_{w^{m-1}}(Y=0; X \in \mathcal{X}_{j,\tau}^<)+\beta}\right) \\ f_{m,\geq} & \triangleq & \frac{1}{2}\log\left(\frac{\mathbb{P}_{w^{m-1}}(Y=1; X \in \mathcal{X}_{j,\tau}^\geq)+\beta}{\mathbb{P}_{w^{m-1}}(Y=0; X \in \mathcal{X}_{j,\tau}^\geq)+\beta}\right) \end{cases}$$

  and $\beta = \frac{1}{4N}$
- Set $w_i^m = \frac{w_i^{m-1} e^{-Y_i f_m(X_i)}}{\text{Cst}}$ for any $i \in \{1, \ldots, N\}$, where Cst is the normalizing constant.

EndFor

- Output the classifier $\mathbb{1}_{G_M(x) \geq \frac{1}{2}} = \frac{1+\text{sign}[G_M(x)]}{2}$ where $G_M(x) = \sum_{i=1}^{M} c_i f_i$

Figure A.8: **"Real" AdaBoost [175] using stumps** as defined in [10]

A.3.3 Line Inference & Pose estimation

Line inference consists of recovering the most probable 2D patches-to-3D lines configuration using the set of classifiers

$$\mathcal{S}^n = \left\{\mathbb{1}_{G_M^1(x)>T_1}, \ldots, \mathbb{1}_{G_M^k(x)>T_k}, \ldots \mathbb{1}_{G_M^n(x)>T_n}\right\}$$

. In this section, we first explore the straighforward solution and then we propose an objective function that couples the outcome of the weak learners with geometric constraints inherited from the learning stage. Such an objective function also solves pose estimation since the optimal camera parameters refer to its lowest potential.

In order to validate the performance of the AdaBoost classifier, we have created a realistic synthetic environment where inference results can be compared with the true configuration. The feature vector for one-preselected line has been learnt, and the corresponding classifier was tested with new images. Results for the 30 first iterations of the real AdaBoost are presented in [Fig. (A.9)]. We can clearly make several observations. First, learning error converges to zero while the error of the classification in the test remains stable. Such a remark is consistent with the expected behavior of the classifier; boosting does not overfit as previously mentioned. Then, samples from Class II are almost never misclassified while classification error of Class I is very important and therefore direct pose estimation is almost impossible. On top of that, one can claim that the lines that are visible change from one image to the next; therefore pose is ill-posed. Such a limitation can be dealt with the use of geometrical constraints encoded in the learning state during the 3D reconstruction step. Such an assumption could allow us to relax the AdaBoost, since classification errors become less significant once geometry is introduced.

A modified classification model is now constructed based on the previous oberservations. Let j be a new image outside the vido sequence. Any sample p such that $\left(G_M^k(p) > T_k\right)$ (Class I) is a potential match. Moreover, classification confidence depends on the distance of the data to be classified from the the boundary and so on the value of $\text{sd}^k(x) = G_M^k(x) - T_k$: the greater is $|\text{sd}^k(x)|$ the more confident is the classification. Thus, the easiest classification choice is:

$$\arg \max_{\substack{i \in \{1,\ldots,n\} \\ \text{st:} G_M^k(p_i^j) > T_k}} G_M^k(p_i^j) - T_k \tag{A.11}$$

The correspondance expressed in eqn. (A.11) is not sufficient since the most important value does not necessarily correspond to the real match. Let us assume for a line k, we are interested in the B best potential matches $\{p_{n_1}[k], \ldots, p_{n_B}[k]\}$. Such candidates are determined through the eqn. (A.11). If less than B lines verify the constraint $G_M^k(p_i[k])) > T_k \, \forall i$, then it is "relaxed" as explained earlier. In other words, lines misclassified are authorized to be taken into consideration by

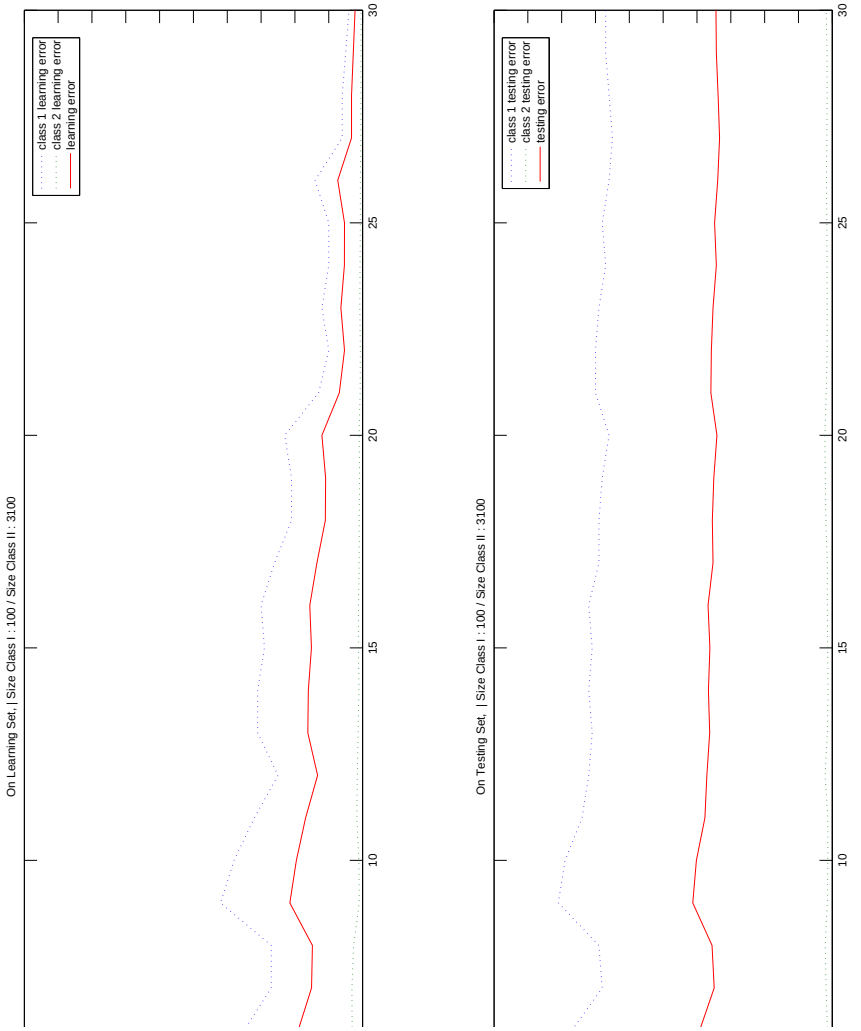

Figure A.9: **Error rates during 30 iterations of real AdaBoost - Top row:** learning set. **Bottom row:** test set. Red: mean error - Blue: error rate of Class I, class of the line learnt - Green: error rate of Class II, class of the other lines

removing the constraint in eqn. (A.11). A weighting function $h(.)$ is also used to influence the importance of a potential match based on the quantity $\text{sd}^k(.)$.

Actually we want to express a geometrical constraint GC between the projections of C lines $\{l_{s_1}, \ldots, l_{s_c}, \ldots, l_{s_C}\}$ ($C < B$). For each line s_c we keep the B best potential matches $\{p_{n_1}[s_c], \ldots, p_{n_b}[s_c], \ldots, p_{n_B}[s_c]\}$. Finally, the energy to be minimized is given by:

$$\min_{\substack{(i_1 \ldots ß_C) \in \\ (\mathcal{A}_1, \ldots, \mathcal{A}_C)}} \sum_{c=1}^{C} h(\text{sd}^{i_c}(p_{i_c}[s_c])) \text{ subject to } \text{GC}(p_{i_1}[s_1], \ldots, p_{i_C}[s_C]) \tag{A.12}$$

where:

- $\mathcal{A}_c$ is the indice set of potential matches with line l_{s_c}
- $h(x)$ is as in our implementation inversely proportional to x. More complex models can however be imagined.

with GC being the geometric constraint. One can recover the lowest potential of such a cost function using classical optimization methods but at the sight of the small number of lines detected, we consider an exhaustive search approach. Numerous formulations can be considered for the GC term. Corners are prominent characteristics of 3D scenes. Therefore, 3D lines going through the same point (that can also define an orthogonal basis) is a straighforward geometry-driven constraint. One can use such an assumption to define constraints in their projection space; that is:

$$\text{GC}(l_1, l_2, l_3) = |(l_1 \times l_2)^T l_3| \tag{A.13}$$

where $\times$ is the cross product of 2 projective points/lines and ^{T} is the transpose sign.

Such a term takes into account the scene context. Offices, buildings, etc. are scenes where the use of such a constraint is mostly justified (corners, vanishing points etc.). For example in figure (A.11), the learning step of lines 1,2 and 3 gives a set $\left\{\mathbb{1}_{G_M^1(x)>T_1}, \mathbb{1}_{G_M^2(x)>T_2}, \mathbb{1}_{G_M^3(x)>T_3}\right\}$. If only feature constraint is used through eqn. (A.11), only line 2 is well matched. However, by using relaxation and the geometrical constraint associated to these lines, the algorithm retrieves good matching. In more complex scenes, more advanced terms can be considered to

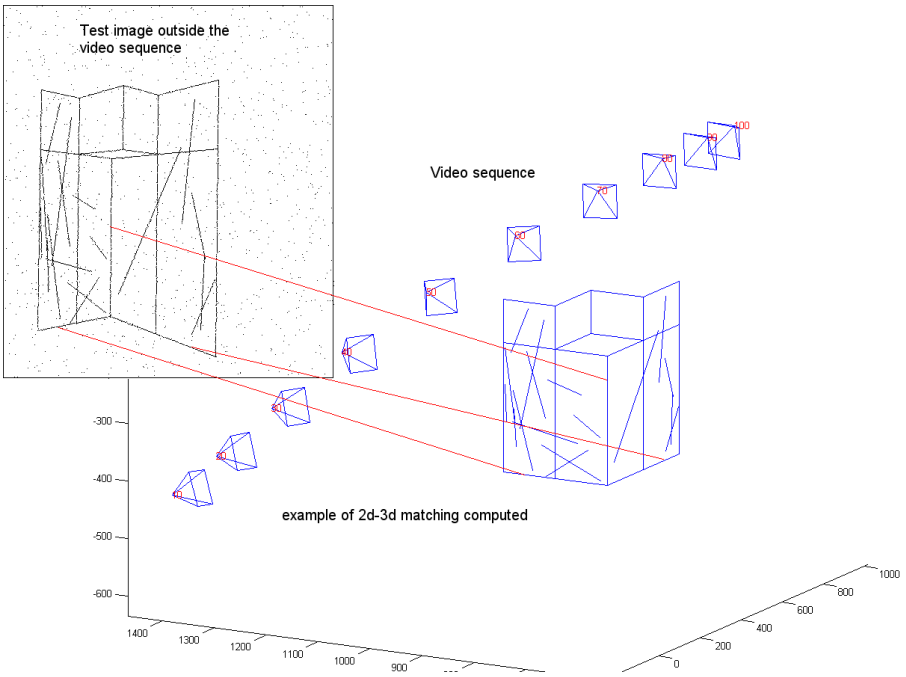

Figure A.10: **Example of learning results on synthetic data**. Red lines: matching of 3 lines using geometrical constraint

improve the robustness of the method. Once the line correspondence problem has been solved, the pose parameters of the camera can be determined using a number of methods [71, 154, 44], but we choose to implement a fast efficient linear method presented in [6].

A.4 Conclusion & Discussion

We have proposed a new technique for one shot pose estimation from images in known environments, which gives promising results. Several experiments were conducted to determine the performance of the method. To this end, first a video stream along with the corresponding 3D geometry [that can be recovered by using standard reconstruction techniques] of the scene were used to learn the model. Such a model refers to n classifiers with their features space being patches of the Radon transformation of the original image. Then, new images of the same scene

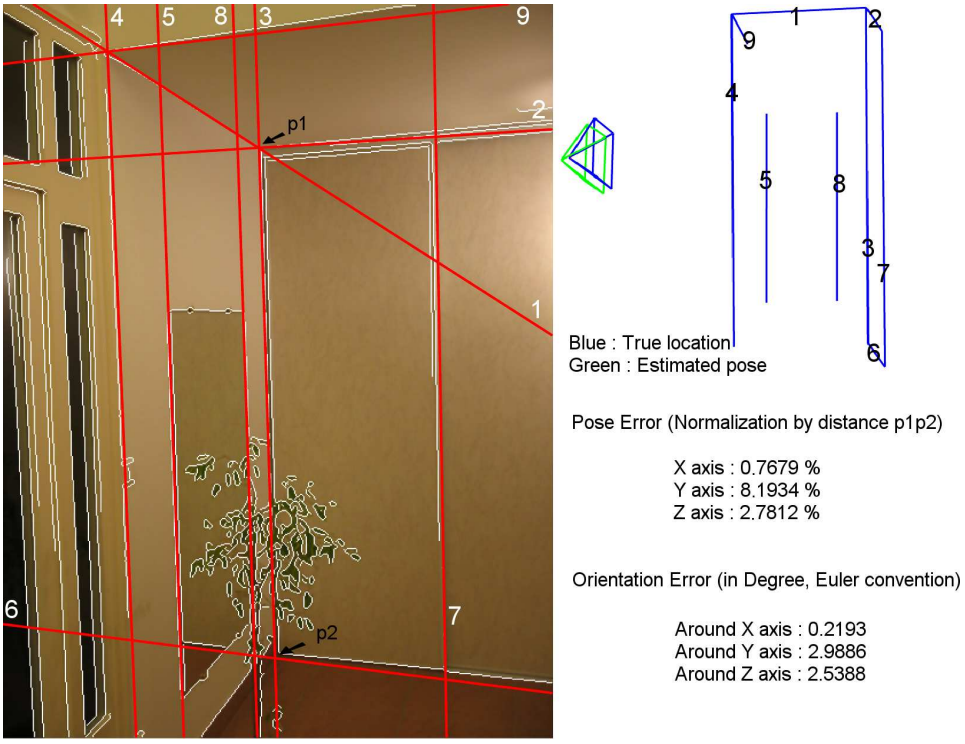

Figure A.11: **Final calibration:** the image to be calibrated is overlaid by the edge map (in white) and the 3D line reprojection (in red)

were considered and self-localization of the observer based on 2D-3D line matching [Fig. (A.10) & (A.11)] was performed.

Our method comprises a learning step where a direct association between 3D lines and Radon patches is obtained. Boosting is used to model statistical characteristics of these patches and weak classifiers are used to determine the most optimal match for a given observation. Such a classification process provides multiple possible matches for a given line and therefore a fast pruning technique that encodes geometric consistency in the process is proposed. Such additional constraints overcome the limitation of classification errors and increase the performance of the method.

Better classification and more appropriate statistical models of lines in Radon space is the most promising direction. The use of Radon patches encodes some extended clutter and therefore separating lines from irrelevant information could improve the performance of the method. Better tracking of lines through linear prediction techniques such as Kalman filter could improve the learning stage and make the method more appropriate for real-time autonomous systems. Last, but not least, representing the camera's pose parameters using non-parametric kernel-based statistical models seems to be more suitable terms to further develop the inference process.

Bibliography

[1] Rolf Adams and Leanne Bischof. Seeded region growing. *IEEE Transaction Pattern Analysis Machine Intelligence*, 16(6):641–647, 1994. 4.2.2

[2] Adini Adini, Yael Moses, and Shimon Ullman. Face recognition: The problem of compensating for changes in illumination direction. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 19(7), 1997. 6.1.2

[3] Alberto S. Aguado, Eugenia Montiel, and Mark S. Nixon. On the intimate relationship between the principle of duality and the hough transform. *Proceedings of the Royal Society, London*, A-456:503–526, 2000. A.2.1

[4] Omar Ait-Aider, Philippe Hoppenot, and Etienne Colle. Adaptation of lowe's camera pose recovery algorithm to mobile robot self-localisation. *Robotica*, 20(4):385–393, 2002. A.1

[5] Cédric Allène, Jean-Yves Audibert, Michel Couprie, Jean Cousty, and Renaud Keriven. Some links between min-cuts, optimal spanning forests and watersheds. In *Proceedings of the 8th International Symposium on Mathematical Morphology*, 2007. 4.2.2

[6] Adnan Ansar and Konstantinos Daniilidis. Linear pose estimation from points or lines. In *ECCV '02: Proceedings of the 7th European Conference on Computer Vision-Part IV*, pages 282–296, London, UK, 2002. Springer-Verlag. A.3.3

[7] Ben Appleton and Hugues Talbot. Globally optimal geodesic active contours. *Journal of Mathematics and Imaging Vision*, 23(1):67–86, 2005. 4.3.3

[8] 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. 7.1.1, 7.2

[9] Nachman Aronszajn. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68(3):337–404, 1950. 5.1.2, 5.1.2

[10] Jean-Yves Audibert. *PAC-Bayesian Statistical Learning Theory*. PhD thesis, Université Paris 6, Francec, Jul 2004. A.3.2, A.3.2, A.8

[11] Jean-Francois Aujol and Gilles Aubert. Signed distance functions and viscosity solutions of discontinuous hamilton-jacobi equations. Research Report 4507, Inria, France, July 2002. 3.2.3

[12] F.R. Bach. Active learning for misspecified generalized linear models. *Advances in Neural Information Processing Systems (NIPS)*, 19, 2006. 6.1.1

[13] Eric Bardinet, Laurent D. Cohen, and Nicolas Ayache. Tracking and motion analysis of the left ventricle with deformable superquadrics. *Medical Image Analysis*, 1(2), 1996. also INRIA Research Report RR-2797, INRIA Sophia-Antipolis, February 1996. 3.2.3

[14] Simon A. Barker. *Image Segmentation using Markov Random Field Models*. PhD thesis, University of Cambridge, Signal Processing and Communications Larboratory, Department of Engineering, July 1998. 4.2.2

[15] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, 2003. 1.3, 2.3, 5.3, 5.3.1, 5.3.4, 5.4, 6.3

[16] Mikhail Belkin and Partha Niyogi. Semi-supervised learning on manifolds. *Machine Learning*, 56:209–239, 2004. 6.1.2, 6.3

[17] 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, editors, *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA, 2004. 7.1.1

[18] Serge Beucher. Sets, partitions and functions interpolations. In *ISMM '98: Proceedings of the fourth international symposium on Mathematical mor-*

phology and its applications to image and signal processing, pages 307–314, Norwell, MA, USA, 1998. Kluwer Academic Publishers. 3.4.3

[19] Serge Beusher and Christian Lantuejoul. Use of watershed in contour detection. In *International Workshop in Image Processing, Real-time Edge and Motion Detection/Estimation*, Rennes, France, September 1979. 4.2.2

[20] J. Ross Beveridge, Joey S. Griffith, Ralf R. Kohler, Allen R. Hanson, and Edward M. Riseman. Segmenting images using localized histograms and region merging. *IJCV*, 2(3):311–352, January 1989. 4.1, 4.2.2

[21] Prabir Bhattacharya, Azriel Rosenfeld, and Isaac Weiss. Point-to-line mappings as hough transforms. *Pattern Recognition Letter*, 23(14):1705–1710, 2002. A.2.1

[22] Griff Bilbro, Reinhold Mann, Thomas K. Miller, Wesley E. Snyder, David E. Van den Bout, and Mark White. Optimization by mean field annealing. In *Advances in neural information processing systems 1*, pages 91–98, San Francisco, CA, USA, 1989. Morgan Kaufmann Publishers Inc. 4.2.2

[23] Andrew Blake and Andrew Zisserman. Some properties of weak continuity constraints and the gnc algorithm. In *CVPR '86: Proceedings of the International Conference on Computer Vision and Pattern Recognition*, volume 86, pages 656–661. 4.2.2, 4.2.2

[24] Andrew Blake and Andrew Zisserman. *Visual Reconstruction*. MIT Press, 1987. 4.2.2, 4.2.2

[25] Jean-Daniel Boissonnat and Mariette Yvinec. *Algorithmic geometry*. Cambridge University Press, New York, NY, USA, 1998. 3.2.2

[26] Gunilla Borgefors. Distance Transformations in Digital Images. *Computer Vision, Graphic, and Image Processing*, 34:344–371, 1986. 3.2.3

[27] Nozha Boujemaa, Fran¸cois Fleuret, Valérie Gouet, and Hichem Sahbi. Visual content extraction for automatic semantic annotation of video news. *In the proceedings of the SPIE Conference, San Jose, CA*, 2004. 6.1

[28] Yuri Boykov and Gareth Funka-Lea. Graph cuts and efficient n-d image segmentation. *Int. J. Comput. Vision*, 70(2):109–131, 2006. 4.2.2

[29] Yuri Boykov and Marie-Pierre Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in n-d images. In *ICCV'01: Proceedings of the 8th IEEE International Conference on Computer Vision*, volume 1, pages 105–112, 2001. 4.2.2

[30] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, 2001. 4.2.2

[31] Matthew Brand. Charting a manifold. In Sebastian Thrun, Lawrence Saul, and Bernhard Schölkopf, editors, *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA, 2004. 5.4

[32] Matthew Brand. Minimax embeddings. In Sebastian Thrun, Lawrence Saul, and Bernhard Schölkopf, editors, *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA, 2004. 5.4

[33] Claude R. Brice and Claude L. Fennema. Scene analysis using regions. *Artificial Intelligence*, 1(3):205–226, 1970. 4.1, 4.2.2

[34] G. Caenen and E.J. Pauwels. Logistic regression model for relevance feedback in content based image retrieval. *In proceedings of SPIE*, 4676:49–58, 2002. 6.1.1

[35] John F. Canny. A computational approach to edge detection. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986. 4.1, 4.2.1

[36] Vicent Caselles, Francine Catté, Tomeu Coll, and Francoise Dibos. A geometric model for active contours. *Nümerische Mathematik*, 66:1–31, 1993. 4.3.3, 4.3.3

[37] Vicent Caselles, Ron Kimmel, and Guillermo Sapiro. Geodesic active coutours. In *Fifth IEEE Internation Conference on Computer Vision*, pages 694–699, October 1995. 4.3.3, 4.3.3

[38] Tony F. Chan and Luminita A. Vese. Active contours without edges. *IEEE Transactions on Image Processing*, 10(2):266–277, 2001. 4.3.3, 4.3.3

[39] Guillaume Charpiat. *Shape Statistics for Image Segmentation with Priors*. PhD thesis, Ecole Polytechnique, Dec 2006. 3, 3.1, 3.2.1, 3.3.1, 3.4.2, 3.4.2, 3.4.2, 3.4.3

[40] Guillaume Charpiat, Olivier Faugeras, and Renaud Keriven. Approximations of shape metrics and application to shape warping and empirical shape statistics. *Foundations of Computational Mathematics*, 5(1):1–58, 2005. 1.3, 2.3, 3.3.1, 3.3.1, 3.4.2, 3.4.3, 3.4.3, 4.4.2, 7.3.2, 7.3.3, 7.3.3

[41] Y. Chen, X.S. Zhou, and T.S. Huang. One-class svm for learning in image retrieval. *Int'l Conf on Image Processing*, 2001. 6.1.1

[42] Yunmei Chen, Hemant D. Tagare, Sheshadri Thiruvenkadam, Feng Huang, David Wilson, Kaundinya S. Gopinath, Richard W. Briggs, and Edward A. Geiser. Using prior shapes in geometric active contours in a variational framework. *The International Journal of Computer Vision*, 50(3):315–328, 2002. 1.3, 2.3, 4.4.2

[43] David L. Chopp. Computing minimal surfaces via level set curvature flow. *Journal of Computational Physics*, 106:77–91, 1993. 4.3.3, 4.3.3

[44] Stéphane Christy and Radu Horaud. Iterative pose computation from line correspondences. *Computer Vision and Image Understanding*, 73(1):137–144, January 1999. A.1, A.3.3

[45] Fan R. K. Chung. *Spectral Graph Theory (CBMS Regional Conference Series in Mathematics, No. 92) (Cbms Regional Conference Series in Mathematics)*. American Mathematical Society, February 1997. 5.3.2, 5.3.2

[46] Isaac Cohen and Laurent D. Cohen. A hybrid hyperquadric model for 2-D and 3-D data fitting. Research report, INRIA, 1992. 3.2.3

[47] Ronald Coifman, Stéphane Lafon, Ann B. Lee, Mauro Maggioni, Boaz Nadler, Fred Warner, and Steven Zucker. Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps. *PNAS*, 102(21):7426–7431, 2005. 5.3.5, 5.3.5, 7.3.2, 8.3.3

[48] Timothy F. Cootes, Christopher J. Taylor, David H. Cooper, and Jim Graham. Active shape models-their training and application. *Computer Vision and Image Understanding*, 61(1):38–59, 1995. 4.4.2

[49] Jose A Costa and Alfred O Hero. *Statistics and Analysis of Shapes. H. Krim and A. Yezzi Eds, Springer*, chapter Determining Intrinsic Dimension and Entropy of high-Dimensional Shape Spaces. 2006. 5.5

[50] Ingemar J. Cox, Matthew L. Miller, Thomas P. Minka, and Peter N. Yianilos. An optimized interaction strategy for bayesian relevance feedback. *CVPR' 98: IEEE Conference on Computer Vision and Pattern Recognition, Santa Barbara, California*, pages 553–558, 1998. 6.1, 6.1.1

[51] Trevor F. Cox and Michael A. A. Cox. *Multidimensional Scaling, Second Edition*. Chapman & Hall/CRC, September 2000. 1.1.2, 1.3, 2.1.2, 2.3, 5.2.1

[52] Daniel Cremers. *Statistical shape knowledge in variational image segmentation*. PhD thesis, University of Mannheim, 2002. 4.3.3

[53] Daniel Cremers, Timo Kohlberger, and Christoph Schnörr. Nonlinear shape statistics in mumford shah based segmentation. In *European Conference on Computer Vision*, pages 93–108, 2002. 1.3, 2.3, 3.2.2, 4.4.3

[54] Daniel Cremers, Stanley Osher, and Stefano Soatto. Kernel density estimation and intrinsic alignment for knowledge-driven segmentation: Teaching level sets to walk. In C. E. Rasmussen, editor, *Pattern Recognition (Proc. DAGM)*, volume 3175 of *LNCS*, pages 36–44. Springer, 2004. 3.3.1

[55] Daniel Cremers, Stanley Osher, and Stefano Soatto. Kernel density estimation and intrinsic alignment for shape priors in level set segmentation. *International Journal of Computer Vision*, 69(3):335–351, September 2006. 4.4.2

[56] Daniel Cremers, Mikael Rousson, and Rachid Deriche. A review statistical approaches to level set segmentation: integrating colors, texture, motion and shape. *International Journal of Computer Vision*, 72(2):195–215, 1989. 4.3.3

[57] Daniel Cremers, Christoph Schnörr, and Joachim Weickert. Diffusion snakes: Combining statistical shape knowledge and image information in a variational framework. In N. Paragios, editor, *IEEE First Int. Workshop on Variational and Level Set Methods*, pages 137–144, Vancouver, 2001. 1.2, 2.2

[58] Daniel Cremers, Christoph Schnörr, Joachim Weickert, and Christian Schellewald. Diffusion snakes using statistical shape knowledge. In C. Sommer and Y.Y. Zeevi, editors, *Algebraic frames for the perception-action cycle*, volume 1888 of *LNCS*, pages 164–174, Kiel, Germany, Sept. 2000. Springer. 1.2, 2.2

[59] Olivier Cuisenaire. *Distance transformations: fast algorithm and applications to medical image processing*. PhD thesis, Universitet' Catholique de Louvain, October 1999. 3.2.3

[60] Samuel Dambreville, Yogesh Rathi, and Allen Tannenbeaum. Shape-based approach to robust image segmentation using kernel pca. In *CVPR '06: Proceedings of the International Conference on Computer Vision and Pattern Recognition*, pages 977–984, 2006. 7.2

[61] Per-Erik Danielsson. Euclidean distance mapping. *Computer Graphics and Image Processing*, 14:227–248, 1980. 3.2.3

[62] Michel C. Delfour and Jean-Paul Zolésio. *Shapes and geometries: analysis, differential calculus, and optimization*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2001. 3, 3.1

[63] Hervé Delingette. General object reconstruction based on simplex meshes. *Int. J. Comput. Vision*, 32(2):111–146, 1999. 3.2.2

[64] Hervé Delingette and Johan Montagnat. Shape and topology constraints on parametric active contours. *Computer Vision and Image Understanding*, 83(2):140–171, August 2001. 3.2.2

[65] Rachid Deriche. Using Canny's criteria to derive a recursively implemented optimal edge detector. *The International Journal of Computer Vision*, 1(2):167–187, May 1987. 4.1, 4.2.1

[66] Rachid Deriche and Olivier Faugeras. Tracking line segments. *Image and Vision Computing*, 8(4):261–270, 1990. A.1

[67] Alain Dervieux and François Thomasset. A finite element method for the simulation of rayleigh-taylor instability. *Lecture Notes on Mathematics*, (771):145–159, 1979. 3.2.3

[68] Mathieu Desbrun, Mark Meyer, Peter Schröder, and Alan H. Barr. Implicit fairing of irregular meshes using diffusion and curvature flow. *Computer Graphics*, 33(Annual Conference Series):317–324, 1999. 8.1

[69] Konstantinos I. Diamantaras and Sun Yuan Kung. *Principal component neural networks: theory and applications*. John Wiley & Sons, Inc., New York, NY, USA, 1996. 5.1.1, 5.1.1

[70] David L. Donoho and Carrie Grimes. Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data. *PNAS: Proceedings of the National Academy Sciences*, 100(10):5591–5596, May 2003. 5.4

[71] Fadi Dornaika and Christophe Garcia. Pose estimation using point and line correspondences. *Journal of Real-Time Imaging, Academic Press*, 5(3):217–232, June 1999. A.1, A.3.3

[72] Richard O. Duda and Peter E. Hart. Use of the hough transformation to detect lines and curves in pictures. *Commun. ACM*, 15(1):11–15, 1972. A.1, A.3.1, A.3.2

[73] Richard O. Duda and Peter E. Hart. Use of the hough transformation to detect lines and curves in pictures. *Commun. ACM*, 15(1):11–15, 1972. A.2.1

[74] Patrick Etyngier, Renaud Keriven, and Jean-Philippe Pons. Projecting onto a shape prior manifold. Technical Report 06-25, CERTIS, Nov 2006.

[75] Patrick Etyngier, Renaud Keriven, and Jean-Philippe Pons. Towards segmentation based on a shape prior manifold. In *SSVM' 07: 1st International Conference on Scale Space and Variational Methods in Computer Vision*, Ishia, Italy, May 2007. 3, 1, 7.3.1, 8

[76] Patrick Etyngier, Renaud Keriven, and Florent Ségonne. Projection onto a shape manifold for image segmentation with prior. In *ICIP' 07: 14th IEEE International Conference on Image Processing*, San Antonio, Texas, USA, Sep 2007. 3, 2, 7.3.1, 8

[77] Patrick Etyngier, Renaud Keriven, and Florent Ségonne. Projection onto manifolds, manifold denoising and applications. Technical Report 07-33, CERTIS, Apr 2007.

[78] Patrick Etyngier, Nikos Paragios, Jean-Yves Audibert, and Renaud Keriven. Radon/hough space for pose estimation. Technical Report 06-22, CERTIS, Jan 2006.

[79] Patrick Etyngier, Nikos Paragios, Renaud Keriven, Yakup Genc, and Jean-Yves Audibert. Radon space and adaboost for pose estimation. In *ICPR' 06: 18th International Conference on Pattern Recognition*, Hong-Kong, Aug 2006. A

[80] Patrick Etyngier, Florent Ségonne, and Renaud Keriven. Active-contour-based image segmentation using machine learning techniques. In *MICCAI' 07: 10th International Conference on Medical Image Computing and Computer Assisted Intervention*, Brisbane, Australia, Oct 2007. 3, 3, 7.3.1, 8

[81] Patrick Etyngier, Florent Ségonne, and Renaud Keriven. Shape priors using manifold learning techniques. In *ICCV' 07: 11th IEEE International Conference on Computer Vision*, Rio de Janeiro, Brazil, Oct 2007. 3, 3, 7.3.1, 8

[82] L. C. Evans and J. Spruck. Motion of level sets by mean curvature. ii. *Transactions of the American Mathematical Society*, 330(1), Mar 1992. 3.4.2

[83] Y. Fang and D. Geman. Experiments in mental face retrieval. *Proceedings AVBPA 2005, Lecture Notes in Computer Science*, pages 637–646, July 2005. 6.1.1, 6.2

[84] Olivier Faugeras. *Three-dimensional computer vision: a geometric viewpoint*. MIT Press, Cambridge, MA, USA, 1993. A.3.1

[85] Olivier Faugeras. *Three-dimensional computer vision. A geometric view point*. MIT Press, 2003. ISBN : 0-262-06158-9. A.2.1

[86] Herbert Federer. Hausdorff measure and lebesgue area. *Proceedings of the National Academy of Sciences U.S.A.*, 37(2):90–94, 1951. 6

[87] Marin Ferecatu, Michel Crucianu, and Nozha Boujemaa. Retrieval of difficult image classes using svd-based relevance feedback. *MIR '04: Proceedings of the 6th ACM SIGMM international workshop on Multimedia information retrieval*, pages 23–30, 2004. 6.1.1, 6.5.3

[88] Céline Fouard and Grégoire Malandain. 3-d chamfer distances and norms in anisotropic grids. *Image and Vision Computing*, 23(2):143–158, February 2004. 3.2.3

[89] Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *EuroCOLT*, pages 23–37, 1995. A.3.2

[90] Yoav Freund and Robert E. Schapire. Experiments with a new boosting algorithm. In *ICML*, pages 148–156, 1996. A.3.2

[91] Keinosuke Fukunaga and David R. Olsen. An algorithm for finding intrinsic dimensionality of data. *IEEE Transaction on Computers*, 20(2):176–183, 1971. 5.5

[92] Taubi Gabriel, Fernando Cukierman, Steve Sullivan, Jean Ponce, and David J. Kriegman. Parameterized families of polynomials for bounded algebraic curve and surface fitting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(3):287–303, 1994. 3.2.3

[93] Charles Galambos, Josef Kittler, and Jiri Matas. Progressive probabilistic hough transform for line detection. In *CVPR'99: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, pages 1554–1560, 1999. A.2.1

[94] Michael Gashler, Dan Ventura, and Tony Martinez. Iterative non-linear dimensionality reduction with manifold sculpting. In J.C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *NIPS '08: Advances in Neural Information Processing Systems 20*, pages 513–520. MIT Press, Cambridge, MA, 2008. 5.4

[95] Muriel Gastaud, Michel Barlaud, and Gilles Aubert. Combining shape prior and statistical features for active contour segmentation. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(5):726–734, 2004. 4.4.2

[96] Stuart Geman and Geman Donald. Stochastic relaxation, gibbs distributions and the bayesian restoration of images. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 6:721–741, November 1984. 4.2.2

[97] José Gomes and Olivier D. Faugeras. Reconciling distance functions and level sets. In *SCALE-SPACE '99: Proceedings of the Second International Conference on Scale-Space Theories in Computer Vision*, pages 70–81, London, UK, 1999. Springer-Verlag. 3.4.2

[98] Leo Grady. Random walks for image segmentation. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 28(11):1768–1783, 2006. Member-Leo Grady. 4.2.2

[99] Xiaofei He and Partha Niyogi. Locality preserving projections. In Sebastian Thrun, Lawrence Saul, and Bernhard Schölkopf, editors, *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA, 2004. 1.1.2, 2.1.2, 5.3

[100] Matthias Hein and Jean-Yves Audibert. Intrinsic dimensionality estimation of submanifolds in r^d. In *ICML '05: In the proceedings of the 22nd International Conference on Machine Learning*, Aug 2005. 5.5

[101] Matthias Hein, Jean-Yves Audibert, and Ulrike von Luxburg. From graphs to manifolds - weak and strong pointwise consistency of graph Laplacians. ArXiv Preprint, Journal of Machine Learning Research, forthcoming, 2006. 5.3.1, 5.3.2, 5.3.3, 5.3.3, 5.3.3, 5.3.5

[102] Matthias Hein and Markus Maier. Manifold denoising. In *Advances in Neural Information Processing Systems*, Cambridge, MA, USA, June 2006. MIT Press. 8.1

[103] Paul V.C. Hough. Method and means for recognizing complex patterns, Dec 1962. A.2.1

[104] Y. Ishikawa, R. Subramanya, and C. Faloutsos. Mindreader: query databases through multiple examples. *Int'l Conf. on Very Large Data Bases (VLDB), NY*, 1998. 6.1.1

[105] Marcin Iwanowski. *Application de la morphologie mathématique pour l'interpolation d'images numériques.* PhD thesis, Ecole Nationale Supérieure des Mines de Paris (ENSMP) / Ecole Polytechnique de Varsovie, November 2000. 3.2.1

[106] Heikki Kälviäinen, Nahum Kiryati, and Satu Alaoutinen. Randomized or probabilistic hough transform: unified performance evaluation, 1999. A.2.1

[107] H. Karcher. Riemannian center of mass and mollifier smoothing. 30(5):509–541, 1977. 1.3, 3.4.1, 3.4.3

[108] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active Contour Models. *Academic Publishers*, 1987. 3.2.2, 4.3.2

[109] John Kaufhold and Anthony Hoogs. Learning to segment images using region-based perceptual features. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 02:954–961, 2004. 4.2.2

[110] Jens Keuchel, Christoph Schnörr, Christian Schellewald, and Daniel Cremers. Binary partitioning, perceptual grouping, and restoration with semidefinite programming. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 25(11):1364–1379, 2003. 4.2.2

[111] Satyanad Kichenassamy, Arun Kumar, Peter Olver, Allen Tannenbaum, and Anthony J. Yezzi. Conformal curvature flows: from phase transitions to active vision. *Archive for Rational Mechanics and Analysis*, 134:275–301, 1996. 4.3.3, 4.3.3

[112] Scott Kirkpatrick, D. Gelatt, Jr., and Mario P. Vecchi. Optimization by simmulated annealing. *Science*, 220(4598):671–680, 1983. 4.2.2

[113] Nahum Kiryati, Yuval Eldar, and Alfred M. Bruckstein. A probabilistic hough transform. *Pattern Recogn.*, 24(4):303–316, 1991. A.2.1

[114] T. Kurita and T. Kato. Learning of personal visual impression for image database systems. *In the proceedings of the international conference on Document Analysis and Recognition*, 1993. 6.1

[115] James T. Kwok and Ivor W. Tsang. The pre-image problem in kernel methods. In *ICML*, pages 408–415, 2003. 7.2

[116] Jacques-Olivier Lachaud and Annick Montanvert. Deformable meshes with automated topology changes for coarse-to-fine 3D surface extraction. *Medical Image Analysis*, 3(2):187–207, 1999. 3.2.2

[117] Stéphane Lafon and Ronald R. Coifman. Diffusion maps. *Applied and Computational Harmonic Analysis : Special issue on Diffusion Maps and Wavelets*, 21(1):5–30, July 2006. 1.1.2, 1.3, 1.3, 2.1.2, 2.3, 2.3, 3.4.3, 5.3, 5.3.3, 5.3.5

[118] Stéphane Lafon, Yosi Keller, and Ronald R. Coifman. Data fusion and multicue data matching by diffusion maps. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(11):1784–1797, 2006. 6.3.2, 6.3.2, 6.3.2

[119] Cheolha Pedro Lee. *Robust Image Segmentation using Active Contours: Level Set Approaches*. PhD thesis, North Carolina State University, Department of Electrical and Computer Engineering, May 2005. 4.3.3

[120] Sung Chung Lee, Soon Ki Jung, and Ramakant Nevatia. Automatic pose estimation of complex 3d building models. In *WACV*, pages 148–152, 2002. A.3.1

[121] François Leitner and Philippe Cinquin. From splines and snakes to snake splines. In *Selected Papers from the Workshop on Geometric Reasoning for Perception and Action*, pages 264–281, London, UK, 1993. Springer-Verlag. 3.2.2

[122] Michael E. Leventon, Eric Grimson, and Olivier Faugeras. Statistical shape influence in geodesic active contours. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 316–323, 2000. 1.3, 2.3, 3.3.1, 4.4.2, 7.3.3

[123] Elizaveta Levina and Peter J. Bickel. Maximum likelihood estimation of intrinsic dimension. In *NIPS '04: Advances in Neural Information Processing Systems*, 2004. 5.5

[124] Qiang Li and Y. Xie. Randomized hough transform with error propagation for line and circle detection. *Pattern Anal. Appl.*, 6(1):55–64, 2003. A.2.1

[125] Tony Lindeberg. Edge detection and ridge detection with automatic scale selection. *Int. J. Comput. Vision*, 30(2):117–156, 1998. 4.1, 4.2.1

[126] Yuncai Liu, Thomas S. Huang, and Olivier Faugeras. Determination of camera location from 2-d to 3-d line and point correspondences. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 12(1):28–37, 1990. A.3.1

[127] S.D. MacArthur, C.E. Brodley, and C. Shyu. Relevance feedback decision trees in content-based image retrieval. *IEEE Workshop CBAIVL*, 2000. 6.1.1

[128] Ravikanth Malladi, James A. Sethian, and Baba C. Vemuri. A topology independent shape modeling scheme. In *SPIE Geometric Methods in Computer Vision II*, volume 2031, page 246, 1993. 4.3.3, 4.3.3

[129] Tim McInerney and Demetri Terzopoulos. Topology adaptive deformable surfaces for medical image volume segmentation. *IEEE Transactions on Medical Imaging*, pages 840–850, October 1999. 3.2.2, 4.3.2

[130] Tim McInerney and Demetri Terzopoulos. T-snakes: Topology adaptive snakes. *Medical Image Analysis*, 4(2):73–91, 2000. 3.2.2, 4.3.2

[131] Christophe Meilhac, Matthias Mitschke, and Chahab Nastar. Relevance feedback in surfimage. *ACV'98: Proceedings of the 4th IEEE Workshop on Applications of Computer Vision*, 1998. 6.1.1

[132] Andrea C. G. Mennucci and Anthony Yezzi. Metrics in the space of curves. eprint arXiv:math/0412454, 2004. 3.4.1

[133] James Mercer. Functions of positive and negative type and their connection with the theory of integral equations. *Philosophical Transaction of the Royal Society*, 1909. 5.1.2, 5.1.2

[134] Fernand Meyer. Un algorithme optimal de ligne de partage des eaux. In *8e Conférence sur la Reconnaissance des Formes et Intelligence Artificielle*, volume 2, pages 847–859, 1992. 4.2.2

[135] Fernand Meyer. Topographic distance and watershed lines. *Signal Processing*, 38:113–125, 1994. 4.2.2

[136] Fernand Meyer. A morphological interpolation method for mosaic images. In Petros Maragos, Ronald W. Schafer, and Muhammad A. Butt, editors, *Mathematical Morphology and its Applications to Image and Signal Processing*, pages 337–344, Atlanta, May 1996. Kluwer Academic Press. 3.4.3

[137] François Meyer and Greg Stephens. Locality and low-dimensions in the prediction of natural experience from fmri. In J.C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 1001–1008. MIT Press, Cambridge, MA, 2008. 5.5

[138] Peter W. Michor and David Mumford. Riemannian geometries on spaces of plane curves. *J. Eur. Math. Soc.*, 8:1–48, 2006. 3.1, 3.4.1

[139] Erik G. Miller and Christophe Chefd'hotel. Practical non-parametric density estimation on a transformation group for vision. In *CVPR '03: Proceedings of the International Conference on Computer Vision and Pattern Recognition*, volume 02, page 114, Los Alamitos, CA, USA, 2003. IEEE Computer Society. 3.4.2

[140] Thomas P. Minka. Automatic choice of dimensionality for pca. In *NIPS '00: Advances in Neural Information Processing Systems*, pages 598–604, 2000. 5.5

[141] Johan Montagnat, Hervé Delingette, and Nicolas Ayache. A review of deformable surfaces: topology, geometry and deformation. *Image and Vision Computing*, 19(14):1023–1040, December 2001. 3.2.1

[142] David Mumford and Jayant M. Shah. Boundary detection by minimizing functionals. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, June 1985. 4.1, 4.2.2

[143] David Mumford and Jayant M. Shah. Optimal approximation by piecewise smooth functions and associated variational problems. *Communication on Pure and Applied Mathemtics*, 42:577–685, 1989. 4.2.2

[144] H. Narayanan and M. Belkin. On the relation between low density separation, spectral clustering and graph cuts. *Advances on Neural information processing systems NIPS*, 2006. 6.2

[145] E.J. Nyström. über die praktische auflösung von linearen integralgleichungen mit anwendungen auf randwertaufgaben der potentialtheorie. *Commentationes Physico-Mathematicae*, 4(15):1–52, 1928. 1.3, 2.3, 5.3.5

[146] Stanley Osher and Red Fedkiw. Level set methods: an overview and some recent results. *Journal of Computational Physics*, 169(2):463–502, 2001. 3.2.3, 4.4.2, 8.3.2

[147] Stanley Osher and James A. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton–Jacobi formulations. *Journal of Computational Physics*, 79(1):12–49, 1988. 3.2.3, 4.3.3, 4.4.2

[148] Nikos Paragios. *Geodesic Active Regions and Level Set Methods: Contribution and Application in Artificial Vision*. PhD thesis, INRIA: Institut National de Recherche en Informatique et Automatique, 2000. 4.3.3, 4.3.3

[149] 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. 4.3.3, 4.3.3

[150] Nikos Paragios and Rachid Deriche. Geodesic active regions: A new framework to deal with frame partition problems in computer vision. *Journal of Visual Communication and Image Representation (JVCIR)*, 13(1/2):249–268, March 2002. 4.3.3, 4.3.3

[151] Kenneth H. Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2:559–572, 1901. 1.1.2, 1.3, 2.1.2, 2.3, 5.1.1

[152] Pietro Perona and Jitendra Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Transaction on Pattern Analysis Machine Intelligence*, 12(7):629–639, 1990. 4.1, 4.2.2

[153] Karl W. Pettis, T.A. Bailey, Anil K. Jain, , and Richard C. Dubes. An intrinsic dimensionality estimator from near-neighbor information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37:1–25, 1979. 5.5

[154] Thal Quynh Phong, Radu Horaud, Adnan Yassine, and Pham Dinh Tao. Object pose from 2d to 3d point and line correspondences. *International Journal of Computer Vision*, 15(3):225–243, July 1995. A.1, A.3.3

[155] R.W. Picard, T.P. Minka, and M. Szummer. Modeling user subjectivity in image libraries. *In the proceedings of the international conference on Image Processing*, 1996. 6.1

[156] Marc Pollefeys. Obtaining 3d models with a hand-held camera/3d modeling from images. Course/Turorial notes http://www.cs.unc.edu/ marc/tutorial/. A.3.1

[157] Jean-Philippe Pons and Jean-Daniel Boissonnat. Delaunay deformable models: Topology-adaptive meshes based on the restricted delaunay triangulation. In *IEEE Conference on Computer Vision and Pattern Recognition*, Minneapolis, USA, Jun 2007. 3.2.2

[158] Judith M. S. Prewitt. Object enhancement and extraction. *Picture Processing and PSchophysics*, pages 75–149, 1970. 4.1, 4.2.1

[159] J. Radon. über die bestimmung von funktionen durch ihre integralwerte längs gewisser mannigfaltigkeiten. *Mathematisch-Physikalische Klasse*, 69:262–277, 1917. A.2.1

[160] Xiaofeng Ren and Jitendra Malik. Learning a classification model for segmentation. In *ICCV '03: Proceedings of the Ninth IEEE International Conference on Computer Vision*, page 10, Washington, DC, USA, 2003. IEEE Computer Society. 4.2.2

[161] Lawrence G. Roberts. *Machine Perception of Three-Dimensional Solids*. Outstanding Dissertations in the Computer Sciences. Garland Publishing, New York, 1963. 4.1, 4.2.1

[162] Sami Romdhani, Shaogang Gong, and Ahaogang Psarrou. A multi-view nonlinear active shape model using kernel pca. In *BMVC' 99: Proceedings of the British Machine Vision Conference*, 1999. 4.4.3

[163] Azriel Rosenfeld. *Picture Processing by Computer*. Academic Press, 1969. A.2.1

[164] Guy Rosman, Alexander M. Bronstein, Michael M. Bronstein, and Ron Kimmel. Manifold analysis by topologically constrained isometric embedding. In *MPLR '06: Proceeding of the Conference on Machine Learning and Pattern Recognition*, 2006. 5.4

[165] Mikael Rousson and Nikos Paragios. Shape priors for level set representations. In *ECCV'02: Proceedings of the European Conference on Computer Vision*, volume 2, pages 78–92, 2002. 1.3, 2.3, 3.3.1, 4.4.2, 7.3.3

[166] Sam Roweis and Lawrence K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290:2323–2326, 2000. 1.1.2, 2.1.2, 5.3, 5.4

[167] Eric Royer, Maxime Lhuillier, Michel Dhome, and Thierry Chateau. Localization in urban environments: Monocular vision compared to a differential gps sensor. In *CVPR'05: Proceedins of the IEEE International Conference on Computer Vision and Pattern Recognition*, pages 114–121, 2005. A.1, A.3.1

[168] Yong Rui, Thomas S. Huang, and Sharad Mehrotra. Relevance feedback techniques in interactive content-based image retrieval. *Storage and Retrieval for Image and Video Databases (SPIE)*, pages 25–36, 1998. 6.1, 6.1.1, 6.1.2

[169] H. Sahbi. Kernel pca for similarity invariant shape recognition. *In the Journal of Neurocomputing*, 2006. 6.5.1

[170] Hichem Sahbi, Jean-Yves Audibert, and Renaud Keriven. Graph cut transducers for relevance feedback in content based image retrieval. *Certis Research Report, N 07-30, Ecole Nationale des Ponts et Chaussees*, February 2007. 6.5.3

[171] Hichem Sahbi, Patrick Etyngier, Jean-Yves Audibert, and Renaud Keriven. Graph laplacian for interactive image retrieval. Technical Report 07-32, CERTIS, Apr 2007.

[172] Hichem Sahbi, Patrick Etyngier, Jean-Yves Audibert, and Renaud Keriven. Graph laplacian for interactive image retrieval. In *ICASSP'08: International Conference on Acoustics, Speech, and Signal Processing*, Las Vegas, April 2008. 6

[173] Hichem Sahbi, Patrick Etyngier, Jean-Yves Audibert, and Renaud Keriven. Manifold learning using robust graph laplacian for interactive image retrieval. In *CVPR'08: Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, Anchorage, June 2008. 6

[174] Philippe Saint-Marc, Hillel Rom, and Gérard Medioni. B-spline contour representation and symmetry detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(11):1191–1197, 1993. 3.2.2

[175] Robert E. Schapire and Yoram Singer. Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, 37(3):297–336, 1999. A.3.2, A.8

[176] Thomas Schoenemann and Daniel Cremers. Globally optimal image segmentation with an elastic shape prior. In *ICCV'07: Proceedings of the 11th IEEE International Conference on Computer Vision*, Rio de Janeiro, Brazil, October 2007. 4.2.2

[177] Thomas Schoenemann and Daniel Cremers. Introducing curvature into globally optimimal image segmentation: Minimum ratio cycles on product graphs. In *ICCV'07: Proceedings of the 11th IEEE International Conference on Computer Vision*, Rio de Janeiro, Brazil, October 2007. 4.2.2

[178] Greg Schohn and David Cohn. Less is more: Active learning with support vector machines. *ICML'00: Proceedings of the Seventeenth International Conference on Machine Learning*, pages 839–846, 2000. 6.1.1

[179] B. Scholkopf, R. Williamson, A. Smola, J.S. Taylor, and J. Platt. Support vector method for novelty detection. *Adv. in Neural Information Processing Systems, MIT Press.*, 2000. 6.1.1

[180] Bernhard Schölkopf, Alexander J. Smola, and Klaus-Robert Müller. Kernel principal component analysis. In *ICANN '97: Proceedings of the 7th International Conference on Artificial Neural Networks*, pages 583–588, London, UK, 1997. Springer-Verlag. 1.1.2, 2.1.2, 5.1.2, 5.1.2

[181] Mathias Seeger. Learning with labeled and unlabeled data. *In Technical Report, University of Edinburgh*, 2001. 6.1.2

[182] Jean Serra. Hausdorff distances and interpolations. In *International Symposium on Mathematical Morphology and its Applications to Image and Signal Processing*, pages 107–114, 1998. 3.3.1, 3.4.3

[183] James A. Sethian. Curvature flow and entropy conditions applied to grid generation. *J. Comput. Phys.*, 115(2):440–454, 1994. 3.2.3

[184] James A. Sethian. Fast Marching Methods. *SIAM Review*, 41:199–235, 1999. 3.2.3

[185] James 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. 3.2.3, 4.4.2

[186] Dorron Shaked, O. Yaron, and Nahum N. Kiryati. Deriving stopping rules for the probabilistic hough transform by sequential analysis. *Computer Vision and Image Understanding*, 63(3):512–526, 1996. A.2.1

[187] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000. 4.1, 4.2.2

[188] Ali Kemal Sinop and Leo Grady. Uninitialized, globally optimal, graph-based rectilinear shape segementation — The opposing metrics method. In *Proc. of ICCV 2007*. IEEE Computer Society, IEEE, Oct. 2007. 4.2.2

[189] Arnold W. M. Smeulders, Marcel Worring, Simone Santini, Amarnath Gupta, and Ramesh Jain. Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12):1349–1380, 2000. 6.1

[190] Jan Erik Solem. Geodesic curves for analysis of continuous implicit shapes. In *ICPR '06: Proceedings of the 18th International Conference on Pattern Recognition*, pages 852–855, Washington, DC, USA, 2006. IEEE Computer Society. 3.2.1

[191] Jan Erik Solem. Separating rigid motion for continuous shape evolution. In *ICPR '06: Proceedings of the 18th International Conference on Pattern Recognition*, pages 43–46, Washington, DC, USA, 2006. IEEE Computer Society. 3.2.1

[192] Daniel A. Spielman and Shuang-Hua Teng. Spectral partioning works: planar graphs and finite element meshes. In *FOCS' 96: 37th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 96–105. IEEE Computer Society Press, 1996. 6.3

[193] Lawrence H. Staib and James S. Duncan. Boundary finding with parametrically deformable models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 14(11):1061–1075, 1992. 3.2.2

[194] Mark Sussman, Peter Smereka, and Stanley Osher. A level set approach for computing solutions to incompressible two-phase flow. *J. Comput. Phys.*, 114(1):146–159, 1994. 3.2.3

[195] Andrew Swann and Niels Holm Olsen. Linear transformation groups and shape space. *Journal of Mathematical Imaging and Vision*, 19(1):49–62, 2003. 3.4.2

[196] Joshua B. Tenenbaum, Vin de Silva, and John C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, December 2000. 5.2.2, 5.4

[197] Demetri Terzopoulos and Dimitris N. Metaxas. Dynamic 3d models with local and global deformations: Deformable superquadrics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(7):703–714, 1991. 3.2.2

[198] Édouard Thiel. *Géométrie des distances de chanfrein*. Université de la Méditerranée, Aix-Marseille 2, Aix-Marseille 2, Déc 2001. `http://www.lif-sud.univ-mrs.fr/~thiel/hdr` . 3.2.3

[199] K. Tieu and P. Viola. Boosting image retrieval. *IEEE Conf. Computer Vision and Pattern Recognition*, 2000. 6.1.1

[200] Simon Tong and Edward Chang. Support vector machine active learning for image retrieval. *MULTIMEDIA '01: Proceedings of the ninth ACM international conference on Multimedia*, pages 107–118, 2001. 6.1, 6.1.1

[201] Warren S. Torgerson. Multidimensional scaling: I. theory and method. *Psychometrika*, 17:401–419, 1952. 1.1.2, 1.3, 2.1.2, 2.3, 5.2.1, 5.2.1

[202] Warren S. Torgerson. Multidimensional scaling of similarity. *Psychometrika*, 30:379–393, 1952. 1.3, 2.3, 5.2.1

[203] Andy Tsai, Anthony J. Yezzi, William M. Wells, Clare Tempany, Dewey Tucker, Ayres Fan, W. Eric L. Grimson, and Allan S. Willsky. A shape-based approach to the segmentation of medical imagery using level sets. *IEEE Transactions on Medical Imaging*, 22(2):137–154, 2003. 1.3, 2.3, 4.4.2

[204] Andy Tsai, Anthony J. Yezzi, Jr., and Allan S. Willsky. Curve evolution implementation of the mumford-shah functional for image segmentation, denoising, interpolation, and magnification. *IEEE Transactions on Image Processing*, 10(8):1169–1186, August 2001. 4.3.3, 4.3.3

[205] Zhuowen Tu and Song-Chun Zhu. Image segmentation by data-driven markov chain monte carlo. *IEEE Transaction Pattern Analysis Machicne Intelligence*, 24(5):657–673, 2002. 4.2.2

[206] Carole J. Twining and Christopher J. Taylor. Kernel principal component analysis and the construction of non-linear active shape models. In *BMVC' 01: Proceedings of the British Machine Vision Conference*, 2001. 4.4.3

[207] Michael van Ginkel, Cris L. Luengo Hendriks, and Lucas J. van Vliet. A short introduction to the radon and hough transforms and how they relate to each other. Technical Report QI-2004-01, Quantitative Imaging Group, Delft University of Technology, 2004. A.1, A.3.1

[208] Vladimir Vapnik. *Estimation of Dependences Based on Empirical Data*. Springer-Verlag, New York, 1982. A.3.2

[209] N. Vasconcelos and A. Lippman. Learning from user feedback in image retrieval. *in Neural Information Processing Systems MIT press*, 2000. 6.1.1

[210] Ulrike von Luxburg, Mikhail Belkin, and Olivier Bosquet. Consistency of spectral clustering. *Annals of Statistics*, 2007. 6.4

[211] Jingbin Wang, Erdan Gu, and Margrit Betke. Mosaicshape: Stochastic region grouping with shape prior. In *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1*, pages 902–908, Washington, DC, USA, 2005. IEEE Computer Society. 4.2.2

[212] Kilian Q. Weinberger and Lawrence K. Saul. An introduction to nonlinear dimensionality reduction by maximum variance unfolding. In *AAAI*. AAAI Press, 2006. 5.3, 5.4

[213] Lei Xu and Erkki Oja. Randomized hough transform (rht): basic mechanisms, algorithms and computational complexities. *Computer Vision, Graphics and Image Processing (CVGIP)*, 57(2):131–154, 1993. A.2.1

[214] Anthony J. Yezzi, Andy Tsai, and Allan S. Willsky. A statistical approach to snakes for bimodal and trimodal imagery. In *Seventh IEEE International Conference on Computer Vision (ICCV)*, pages 898–903, 1999. 4.3.3, 4.3.3

[215] Laurent Younes. *Invariance, déformations et reconnaissance de formes*. Mathématiques et Applications. Springer, 2004. 3.4.2

[216] Gale Young and A. S. Householder. A note on multidimensional psychophysical analysis. *Psychometrika*, 6:331–333, 1941. 1.1.2, 2.1.2, 5.2.1

[217] Zhenyue Zhang and Hongyuan Zha. Principal manifolds and nonlinear dimensionality reduction via tangent space alignment. *SIAM: Journal of Scientific Computing*, 26(1):313–338, 2004. 5.4

[218] Y. Zhao, Yao Zhao, and Z. Zhu. Relevance feedback based on query refining and feature database updating in cbir system. *Signal Processing, Pattern Recognition, and Applications*, 2006. 6.1

[219] X.S. Zhou and T.S. Huang. Relevance feedback in image retrieval: A comprehensive review. *in IEEE CVPR Workshop on Content-based Access of Image and Video Libraries (CBAIVL)*, 2006. 6.1, 6.1.1

[220] Song Chun Zhu and Alan L. Yuille. Region competition: Unifying snakes, region growing, and bayes/mdl for multiband image segmentation. *IEEE Transaction on Pattern Analysis Machine Intelligence*, 18(9):884–900, 1996. 4.1, 4.2.2

Publications of the author

Conferences

- Patrick Etyngier, Nikos Paragios, Renaud Keriven, Yakup Genc, and Jean-Yves Audibert. **Radon space and adaboost for pose estimation.** *ICPR' 06: 18th International Conference on Pattern Recognition*, Hong-Kong, August 2006.

- Patrick Etyngier, Renaud Keriven, and Jean-Philippe Pons. **Towards segmentation based on a shape prior manifold.** *SSVM' 07: Proceedings of the 1st International Conference on Scale Space and Variational Methods in Computer Vision*, Ishia, Italy, May 2007.

- Patrick Etyngier, Renaud Keriven, and Florent Ségonne. **Projection onto a shape manifold for image segmentation with priors.** *ICIP' 07: Proceedings of the 14th IEEE International Conference on Image Processing*, San Antonio, Texas, USA, September 2007.

- Patrick Etyngier, Florent Ségonne, and Renaud Keriven. **Active-contour based image segmentation using machine learning techniques.** *MICCAI' 07: Proceedings of the 10th International Conference on Medical Image Computing and Computer Assisted Intervention*, Brisbane, Australia, October 2007.

- Patrick Etyngier, Florent Ségonne, and Renaud Keriven. **Shape priors using manifold learning techniques.** *ICCV' 07: Procedings of the 11th IEEE International Conference on Computer Vision, Rio de Janeiro*, Brazil, October 2007.

- Hichem Sahbi, Patrick Etyngier, Jean-Yves Audibert, and Renaud Keriven.

Graph laplacian for interactive image retrieval. *ICASSP' 08: Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, Las Vegas, April 2008.

- Hichem Sahbi, Patrick Etyngier, and Jean-Yves Audibert and Renaud Keriven. **Manifold Learning using Robust Graph Laplacian for Interactive Image Retrieval.** *CVPR' 08: In proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, Anchorage, Alaska, June 2008.

Technical reports

- Patrick Etyngier, Nikos Paragios, Jean-Yves Audibert, and Renaud Keriven. **Radon/hough space for pose estimation.** *Technical Report 06-22, CERTIS*, January 2006.
- Patrick Etyngier, Renaud Keriven, and Jean-Philippe Pons. **Projecting onto a shape prior manifold.** *Technical Report 06-25, CERTIS*, Nov 2006.
- Hichem Sahbi, Patrick Etyngier, Jean-Yves Audibert, and Renaud Keriven. **Graph laplacian for interactive image retrieval.** *Technical Report 07-32, CERTIS*, April 2007.
- Patrick Etyngier, Renaud Keriven, and Florent Ségonne. **Projection onto manifolds, manifold denoising and applications.** *Technical Report 07-33, CERTIS*, April 2007.