Transformation-Invariant Analysis of Visual Signals with Parametric Models

THÈSE N^{O} 5844 (2013)

PRÉSENTÉE LE 4 OCTOBRE 2013

À LA FACULTÉ DES SCIENCES ET TECHNIQUES DE L'INGÉNIEUR
LABORATOIRE DE TRAITEMENT DES SIGNAUX 4
PROGRAMME DOCTORAL EN GÉNIE ÉLECTRIQUE

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Elif VURAL

acceptée sur proposition du jury:

Prof. P. Vandergheynst, président du jury
Prof. P. Frossard, directeur de thèse
Prof. D. Kressner, rapporteur
Dr G. Peyré, rapporteur
Prof. M. B. Wakin, rapporteur

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Suisse
2013

Acknowledgements

Everyone with a PhD would agree that PhD is a long and tough journey. Still, it was also a quite enjoyable one for me and I owe this largely to the people below.

The biggest thanks goes to my PhD supervisor Pascal Frossard, for being the best boss one can ever imagine. Not only did I learn enormously from him, but working with him was also an extremely enjoyable experience and so much fun. He has always been so considerate, kind, and understanding. I will certainly miss a lot the fabulous work environment he has built in LTS4.

Next, I would like to thank Swiss National Science Foundation (SNSF) for financing my PhD studies and giving me the flexibility to work on what I really find interesting in the past five years.

I am very grateful to my thesis jury; Daniel Kressner, Gabriel Peyré, Michael Wakin, and Pierre Vandergheynst, for sparing their valuable time to read and correct my thesis and for the very useful feedback they have provided.

Then comes all past and present LTS4 members, who have always been super-cool workmates and made the lab such a warm and pleasurable place. Many thanks to Vijay, Eirina, Ivana, Nikos, Luigi, Xiaowen (Wenwen), Dorina, Sofia, Dorna, Ana, Attilio, Lorenzo, Yannick, Hyunggon, David, Thomas, Laura, Cédric, Arthur, Yanhui, and Adnan for all the wonderful moments we had in Lausanne and all over the planet. Special thanks to Zafer, for being such a good friend and all the adventure and diversity he brought into my daily life in Lausanne; to Tamara, for always being the right person to talk to whenever I sought help and solidarity; to Alhussein, for the inspiring technical discussions, mind-relieving mid-work chats and his persistent efforts to enrich my French slang vocabulary; to Jacob (Monsieur) for delighting me every now and then with his little surprises like his delicious homemade cooking; and to Efi, for her help in the beginning of my PhD work and providing the DC optimization codes that I benefited a lot from in this thesis. Thanks also to our administrative assistant Rosie De Pietro, not only for her assistance with the paperwork but also for always being there to share our happiness on special occasions. Many thanks to all my other friends in the other SP labs as well.

A huge thanks goes to all my Turkish friends in Lausanne, who have made "Lozan" a real home for me: Işık, Mine, Tuna, Yasemin, Engin, Gürkan, Ahmed, Zafer, Anıl, Eymen, Pınar, Gökhan, Barış, Kerem, Can, Onur, Onur, Nihal, Burçak, Ebru, Zeynep, Tuğçe, Emrah, Emrah, Arda, Arda, Gözen, Ali Galip, Cihangir, Eren, Engin, Mithat, Hüma, Aydın, Mustafa, Yüksel, Selma, Cumhur, Başak, and Ali. Without them, my life in Switzerland would have never been so happy. Thanks very much also to my friends from Turkey; Hande, Burcu, Nilüfer, Murat, Mehmet, Aşkın, Gülhan, Doğacan, and Egemen. Their support from kilometers away was crucial to me.

Many thanks to A. Aydın Alatan, my master's thesis supervisor in METU, for the interesting and helpful discussions we continued to have during my PhD and for his continuing support.

I would also like to thank my piano teacher Layla Ramezan, not only for providing the distraction that was at times invaluable for me to find a balance, but also for her friendship.

I feel very lucky that I had the opportunity to assist the projects of several master students and interns at EPFL. Many thanks to Ozan, Sercan, Ahmet, Víctor, Yu, Şerife, Teodora and Ehsan. I enjoyed very much working with them. Ozan Şener also helped me a lot with preparing the

experimental setups of Chapter 3, and his internship turned out to be the beginning of a nice friendship. Special thanks to Hemant Tyagi, who did his master's thesis with me and ended up teaching me much more than I taught him.

During our whole lives, we meet very few people who influence our lives and our personalities profoundly. Eren Şems Halıcı is one such person for me. The extent of my gratitude for him is far beyond what I could express in a thesis acknowledgement. Without his support, I could have never come to Lausanne to start a PhD. Neither would I be the same person today, if I had never met him.

Next comes the time to thank my family: my mother Ayşegül, my father Abdülgaffar, and my brother Ali. Besides everything that comes as part of being a family, I want to specifically thank mom, for helping me acquire the habit of reading since my early childhood; and dad, for always encouraging me to pursue academic studies. I love them.

Finally, I would like to thank Gilles Puy with all my heart, for the unconditional love and the infinite patience he has had for me. He has always been there to multiply my joy in good times and lift my spirits in bad times. I thank him for all the support he provided during my PhD, and more importantly, for sharing life with me.

Abstract

The analysis of collections of visual data, e.g., their classification, modeling and clustering, has become a problem of high importance in a variety of applications. Meanwhile, image data captured in uncontrolled environments by arbitrary users is very likely to be exposed to geometric transformations. Therefore, efficient methods are needed for analyzing high-dimensional visual data sets that can cope with geometric transformations of the visual content of interest.

In this thesis, we study parametric models for transformation-invariant analysis of geometrically transformed image data, which provide low-dimensional image representations that capture relevant information efficiently. We focus on transformation manifolds, which are image sets created by parametrizable geometric transformations of a reference image model. Transformation manifolds provide a geometric interpretation of several image analysis problems. In particular, image registration corresponds to the computation of the projection of the target image onto the transformation manifold of the reference image. Similarly, in classification, the class label of a query image can be estimated in a transformation-invariant way by comparing its distance to transformation manifolds that represent different image classes. In this thesis, we explore several problems related to the registration, modeling, and classification of images with transformation manifolds.

First, we address the problem of sampling transformation manifolds of known parameterization, where we focus on the target applications of image registration and classification in the sampling. We first propose an iterative algorithm for sampling a manifold such that the selected set of samples gives an accurate estimate of the distance of a query image to the manifold. We then extend this method to a classification setting with several transformation manifolds representing different image classes. We develop an algorithm to jointly sample multiple transformation manifolds such that the class label of query images can be estimated accurately by comparing their distances to the class-representative manifold samples. The proposed methods outperform baseline sampling schemes in image registration and classification.

Next, we study the problem of learning transformation manifolds that are good models of a given set of geometrically transformed image data. We first learn a representative pattern whose transformation manifold fits well the input images and then generalize the problem to a supervised classification setting, where we jointly learn multiple class-representative pattern transformation manifolds from training images with known class labels. The proposed manifold learning methods exploit the information of the type of the geometric transformation in the data to compute an accurate data model, which is ignored in previous manifold learning algorithms.

Finally, we focus on the usage of transformation manifolds in multiscale image registration. We consider two different methods in image registration, namely, the tangent distance method and the minimization of the image intensity difference with gradient descent. We present a multiscale performance analysis of these methods. We derive upper bounds for the alignment errors yielded by the two methods and analyze the variations of these bounds with noise and low-pass filtering, which is useful for gaining an understanding of the performance of these methods in image registration. To the best of our knowledge, these are the first such studies in multiscale registration settings.

Geometrically transformed image sets have a particular structure, and classical image analysis methods do not always suit well for the treatment of such data. This thesis is motivated by this observation and proposes new techniques and insights for handling geometric transformations in image analysis and processing.

Keywords: image registration, pattern classification, transformation-invariance, geometric image transformations, transformation manifolds.

Résumé

L'analyse de collections de données visuelles, comme leur classification, modélisation, ou partitionnement, est un problème très important dans de nombreuses applications. Cependant les images prises par différents utilisateurs dans un environnement incontrôlé sont susceptibles de subir de nombreuses transformations géométriques. Des méthodes efficaces prenant en compte ces transformations sont donc nécessaires afin d'analyser des données visuelles de grande dimension.

Dans cette thèse, nous étudions des modèles paramétriques permettant une analyse de collections d'images invariante par transformations géométriques et fournissant une représentation en basse dimension de ces images tout en préservant leurs informations caractéristiques. Nous nous concentrons sur l'utilisation des variétés créées par des transformations géométriques, représentées par quelques paramètres, d'une image de référence. Ceci nous permet d'avoir une interprétation géométrique de plusieurs problèmes d'analyse de collections d'images. En particulier, le problème du recalage de deux images se réduit au calcul de la projection de l'image d'intérêt sur la variété générée par transformations de l'image de référence. De la même manière, pour la classification, la classe de l'image d'intérêt peut être estimée de manière invariante aux transformations en comparant les distances entre cette image et les variétés produites par transformations des images représentant chaque classe. Dans cette thèse, nous explorons plusieurs problèmes liés à la modélisation, la classification et au recalage d'images grâce à l'utilisation de ces variétés.

Nous étudions tout d'abord le problème de l'échantillonnage de variétés générées par des transformations paramétriques d'images de référence. Pour cela, nous nous concentrons plus particulièrement sur les problèmes de recalage et de classification d'images. Nous proposons tout d'abord un algorithme itératif pour l'échantillonnage d'une variété de sorte que les échantillons obtenus permettent d'estimer précisément la distance entre une image d'intérêt et la variété. Nous généralisons ensuite cette méthode au problème de la classification où nous devons échantillonner plusieurs variétés générées à partir d'images de classes différentes. Nous développons un algorithme capable d'échantillonner conjointement plusieurs variétés afin que la classe d'une image puisse être estimée correctement en comparant les distances entre cette image et les échantillons de chaque variété. Les méthodes proposées sont plus performantes que les méthodes d'échantillonnages classiques utilisées en recalage et classification d'images.

Ensuite, nous étudions le problème de l'apprentissage de variétés modélisant correctement un ensemble d'images géométriquement transformées. Nous apprenons tout d'abord un motif de base générant une variété dont les images d'entraînement sont proches. Nous généralisons ensuite cette approche pour traiter un problème de classification supervisée, où nous apprenons conjointement plusieurs motifs représentatifs de chaque classe à partir d'images dont la classe est connue. Cette méthode d'apprentissage de variétés exploite l'information connue du type de transformations existant entre images, et ignorée dans les autres méthodes d'apprentissage de variétés, pour obtenir un modèle fidèle des données.

Enfin, nous nous intéressons à l'utilisation des variétés pour le recalage d'image par des méthodes multi-échelles. Nous considérons deux différentes méthodes de recalage d'images : la minimisation de la différence d'intensité entre images par descente de gradient et la méthode de la distance aux tangentes. Nous présentons une analyse multi-échelle de la performance de ces méthodes. Nous fournissons des bornes supérieures sur les erreurs de recalage produites par ces deux méthodes et analysons les variations de ces bornes en présence de bruit ou de filtrage passe-bas. Ces résultats sont utiles pour comprendre et expliquer les performance de ces méthodes

de recalage d'images. A notre connaissance, ce sont les premières études de performance de ces méthodes dans un contexte multi-échelle.

Les images transformées géométriquement forment un ensemble avec une structure particulière et les méthodes classiques d'analyse d'images ne sont pas toujours adéquates pour le traitement de tels ensembles. Cette thèse est motivée par cette observation et nous proposons de nouvelles techniques et idées pour traiter efficacement les transformations géométriques en analyse et traitement d'images.

Mots Clés : recalage d'images, classification d'images, invariance aux transformations géométriques, transformation géométrique d'images, variétés générées par transformations géométriques.

Contents

List of Figures

Chapter 1

Introduction

1.1 Analysis of Image Sets

We are living in an era where vast collections of high-dimensional data are created every day, so that the analysis of data content in a fully-automated or computer-aided manner is of great interest in many different domains such as entertainment, security, healthcare, and surveillance. This prompts the need for intelligent systems capable of analyzing data while requiring as little human interaction as possible.

One of the main research interests of the last few decades has thus been the development of efficient algorithms for analyzing data or image sets towards effective solutions for classification, pattern recognition, and clustering tasks. However, the variation in data acquisition conditions constitutes a major challenge in these problems. Most classification and pattern recognition methods are either designed for perfectly aligned input data, in which case the data should be a priori registered with respect to a suitably selected reference model, or they require large amounts of manually labeled training data to achieve invariance to small transformations, which is often time-consuming and burdensome to supply. Invariance to geometric transformations is however an important property of effective data analysis applications. The same object captured from different perspectives, for example, should be given the same identity or label. This can be achieved by parametric data representations built on different forms of analytical models. The treatment of data analysis problems with parametric models that are appropriately designed to account for geometric transformations allows for the development of powerful classification and recognition tools that achieve transformation-invariance in a natural way. The integration of prior knowledge about data geometry into parametric models significantly reduces the required number of training samples in classification applications. Moreover, parametric models provide concise representations for data sets, and therefore, offer convenient means of encoding, processing or retrieving data information.

In this thesis, we study a class of parametric models for analyzing image sets that are based on transformation manifold models. Transformation manifolds are defined as image sets that can be described by parametrizable geometric transformations of a reference image model. An instance of a transformation manifold can be a pattern transformation manifold, which refers to the set of patterns obtained by applying geometric transformations to a reference pattern; or the observation manifold of an object, which consists of its images captured under varying viewpoints.

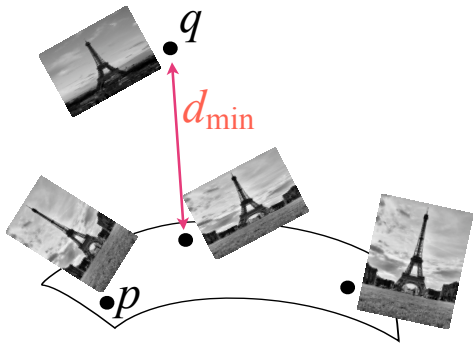


Figure 1.1: Illustration of the image registration problem. The transformation manifold of the reference pattern p consists of its rotated and scaled versions. The target pattern q can be aligned with the reference pattern p by finding the point on the pattern transformation manifold that has the smallest distance ($d_{\min}$) to q. The distance $d_{\min}$ is called the manifold distance. (Photos from [1])

In a setting where transformation manifolds are used for modeling and representing different image types or image classes, the geometric interpretation of common image analysis problems can be used to devise efficient image analysis methods for the registration [2], classification [2], [3], and clustering [4] of images. Image registration refers to the problem of estimating the coordinate transformation between a reference image and a target image, which gives the best approximation of the target image from the reference image. Provided that the coordinate transformation is globally parametrizable by a small number of parameters, the image registration problem can be geometrically regarded as the computation of the projection of the target image onto the transformation manifold of the reference image. In this case, the transformation parameters that best align the image pair are given by the transformation parameters of the manifold point that has the smallest distance to the target image. This is illustrated in Figure 1.1, where the target image q is similar to a rotated and scaled version of the reference image p, and the two images are aligned by identifying the projection of q onto the transformation manifold of p. Then, the problem of classifying images that have undergone geometric transformations can also benefit from manifold models. While common classification methods such as SVM and LDA work very well for linearly separable data sets, geometrically transformed image data usually has a highly nonlinear structure. Such generic methods attempt to handle nonlinearities by using kernel tricks, which however yields a limited performance as the prior information about the data model is not taken into account. However, in a setting where each image class is represented with a different transformation manifold, the class label of a query image can be estimated simply by comparing its distance to the candidate class-representative manifolds and selecting the label of the closest manifold. In such a setting, class-representative manifolds partition the image space into regions such that each region consists of points that correspond to the class represented by a particular manifold. Then, the consideration of each region as a different image class defines a classification rule, where the nonlinearity of data is automatically handled in the classifier as it inherently accounts for geometric transformations with the manifold models. An illustration of transformation-invariant classification with transformation manifolds is given in Figure 1.2, where the object observation manifolds $\mathcal{M}_1$ and $\mathcal{M}_2$ represent two candidate image classes. Since the query image q has smaller distance to the first transformation

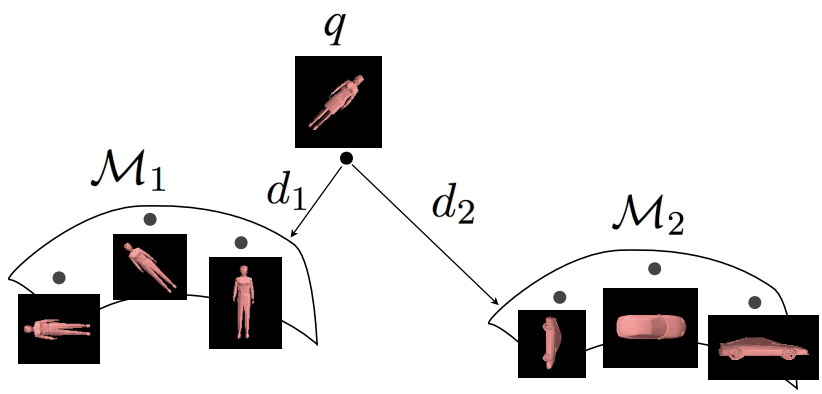


Figure 1.2: Transformation-invariant image classification with manifold models. Object observation manifolds $\mathcal{M}_1$ and $\mathcal{M}_2$ are generated by the observations of two objects representing two different image classes. The class label of a query image q can be estimated by comparing its distances d_1, d_2 to the class-representative transformation manifolds $\mathcal{M}_1$, $\mathcal{M}_2$ respectively.

manifold $\mathcal{M}_1$, it is assigned the class label of the first manifold.

The analysis and representation of image sets with parametric models however raises important research questions. First, given a transformation manifold model representing an image set, how can one use it effectively in the analysis of the image set? A key problem in transformation-invariant image analysis is the computation of the distance between an image and a transformation manifold, which is called the manifold distance. The calculation of the manifold distance requires the projection of the image onto the manifold, which is a complicated optimization problem that does not have a known optimal solution for generic transformation models. This leaves space for the exploration of constructive solutions to estimate the manifold distance by seeking alternative representations of the manifold, such as discrete or multiresolution approximations. Next, given a data set, how can one construct a manifold model that is a good representative of the data? Although the recovery of low-dimensional structures in data sets, which is known as the manifold learning problem, has been studied in many previous works in the literature (e.g., [5], [6], [7]), the problem of learning manifolds in an application-specific manner, by exploiting prior information about the data model, has not been warranted much consideration so far. Finally, what are the theoretical performance limits in the registration of images with globally parametrizable transformation models? Previous studies on the performance of image registration methods usually constrain their analysis to limited types of geometric transformation models; and the works examining the registration problem in a multiresolution setting are mostly restricted to optical flow computation with gradient-based methods. The analysis of image registration for arbitrary transformation models and various representations of the manifold, such as first-order or multiresolution approximations, remains an interesting problem. This thesis is motivated by these important observations and seeks answers to the above key questions.

1.2 Thesis Outline

The thesis is organized as follows.

In Chapter 2, we introduce the setting used in this thesis for image analysis with manifold models and give a brief overview of image registration and classification.

In Chapter 3, we consider the problem of estimating the distance between an image and a transformation manifold of known parameterization and present an effective solution based on the discretization of the manifold. An easy and computationally very simple way to estimate the distance between a query image and a transformation manifold is to sample the manifold offline and then to approximate the manifold distance with the distance of the query image to the sample set. In such a scheme, the accuracy of the manifold distance estimation depends highly on the selection of samples, especially when the manifold is represented with a limited number of samples to speed up the computation. We therefore study the problem of selecting samples from transformation manifolds that provide a good representation of manifolds. We focus on two important applications in manifold sampling; namely, image registration and image classification. We first present an iterative algorithm for sampling a single transformation manifold such that the average error in the manifold distance estimation is minimized. We then extend this solution to propose a method for the joint discretization of multiple manifolds that represent different classes, such that the selected samples give an accurate estimation of the class labels of query images. The proposed sampling methods are experimentally shown to outperform typical sampling schemes such as random and uniform discretizations in the problems of the registration and classification of images with 2-D and 3-D geometric transformations.

While we have considered that manifolds are known in Chapter 3, we study in Chapter 4 the learning of manifolds that are good representatives of image data sets. In particular, we consider the problem of learning pattern transformation manifolds from image sets that have undergone geometric transformations. We treat the manifold learning problem in an application-oriented way, where we target image approximation and classification. For the approximation problem, we propose a greedy method that constructs a representative pattern that is sparsely represented in an analytic dictionary, such that the transformation manifold of the representative pattern fits well the input images. Then, we generalize this learning approach to a supervised classification setting with multiple transformation manifolds representing different image classes. We present an iterative multiple manifold building algorithm such that the classification accuracy is promoted in the joint learning of the representative patterns from training images with known class labels. The learned manifolds are then used in the classification of test images. The most important advantage of the proposed manifold building methods over traditional manifold learning algorithms is that they integrate in the learning the knowledge of the type of geometric transformations of data. This permits us to fit a precise parametric model to the data, which enables the synthesis of novel manifold points and the use of the learned manifolds effectively in classification. We present experimental results showing that the proposed methods have good performance in the approximation and classification of handwritten digits and microbiological images.

Next, in Chapter 5 we concentrate on the utilization of pattern transformation manifolds in image registration applications based on the common tangent distance method [3]. The computation of the exact projection of a target image onto the transformation manifold of a reference image is generally a complicated optimization problem. As an alternative to manifold discretiza-

tion presented in Chapter 3, the tangent distance method estimates this projection by using a first-order approximation of the transformation manifold, in which case the distance estimation is given by a simple least-squares solution. Tangent distance is commonly used in image registration and classification, and like many registration methods, it has multiresolution extensions [2], where the alignment is achieved gradually by using a pyramid of low-pass filtered versions of the reference and target images. Meanwhile, the tangent distance works well only if the linear approximation of the manifold is accurate. The linearity assumption holds when the reference parameters used in the linearization of the manifold are sufficiently close to the optimal solution, or when the manifold has small curvature. This motivates the study presented in this chapter, where a multiscale performance analysis of the tangent distance method is proposed. We first derive an upper bound for the alignment error. Then, we analyze its variation with the size of the low-pass filter used in smoothing the images to build the multiscale representation, and with the image noise level, i.e., the distance between the target image and the transformation manifold of the reference image. Our main finding is that the alignment error bound is linearly proportional to the noise level and it generally varies non-monotonically with the filter size. Hence, there exists an optimal filter size that minimizes the alignment error, whose value depends on the image noise level and the distance between the reference and optimal transformation parameters. Our theoretical findings are confirmed by extensive experimental results.

The tangent distance method considered in Chapter 5 gives a fast estimation of the transformation parameters by linearizing the manifold. Therefore, this method minimizes an approximate version of the manifold distance. Meanwhile, many registration methods minimize the actual manifold distance in order to compute the transformation parameters. A simple and fast solution is to use local, descent-type optimizers for the calculation of the manifold distance. In Chapter 6, we focus on this setting and present a performance analysis of multiscale image registration where the global 2-D translation between a reference image and a noisy target image is estimated by smoothing the images and minimizing the distance between them with local optimizers. In order to provide a thorough characterization of the performance of descent-type minimizers in image registration, we first analyze the well-behavedness of the image distance function by estimating the neighborhood of translations that can be correctly computed with a simple gradient descent minimization. We show that the area of this neighborhood increases at least quadratically with the size of the low-pass filter used in smoothing the image pair in the multiscale representation. We then examine the deviation in the global minimum of the distance function caused by noise, which constitutes a source of error common to all region-based methods that minimize the distance function locally. We derive an upper bound for the alignment error and study its dependence on the noise properties and the filter size. Our main finding is that the error bound increases at a rate that is at least linear with respect to the filter size. We also present experimental results, which are in accordance with the theoretical results. Our detailed analysis of the effect of filtering on the local minima of the distance function and the alignment error is helpful for better understanding the performance limits of multiscale registration methods.

To conclude, this thesis studies the analysis of images with parametric models and proposes new approaches for the representation, approximation, registration and classification of images based on a geometric interpretation of several image analysis problems. Our study provides new insights into transformation-invariance in image analysis and has the potential to be used in the development of effective, geometry-aware tools for the treatment of visual data sets.

1.3 Summary of Contributions

The main contributions of this thesis are summarized below.

- We propose novel methods for sampling transformation manifolds of known parameterization, which yield accurate estimates of the manifold distance in image registration and transformation-invariant image classification applications.
- We present a new and application-oriented approach for the manifold learning problem. We propose novel methods that learn analytic pattern transformation manifolds from data sets by exploiting data model priors in the learning. The proposed methods yield a good performance in the approximation and classification of images.
- We present a first theoretical analysis of multiscale image registration with the tangent distance method for arbitrary geometric transformation models, which shows the importance of the filter size selection for the performance of this algorithm.
- We present an extensive theoretical study of multiscale image registration with descent-type minimizers for the transformation model of 2-D translations. We propose a comprehensive analysis of the well-behavedness of the image dissimilarity function and give a thorough characterization of the alignment error of local minimizers in a multiscale registration setting.

Chapter 2

Low-Dimensional Image Representations and Image Analysis

In this chapter, we introduce some of the concepts and notations used in this thesis, while giving an overview of the most relevant previous works. In Section 2.1, we discuss the representation and processing of image sets with manifold models. Then in Section 2.2, we give an overview of some recent results in image registration and classification.

2.1 Image Manifolds

Many image analysis methods rely on the geometric interpretation of image data as a subset of a high-dimensional ambient space. For example, n-pixel digital images can be represented as points in the discrete space $\mathbb{R}^n$. Another common representation for image sets relies on the continuous space of square-integrable functions, where images correspond to functions in $L^2(\mathbb{R}^2)$.

Interestingly enough, in quite many image processing applications, the image data at hand is concentrated around an intrinsically low-dimensional structure in the high-dimensional space; i.e., the images residing in the high-dimensional ambient space $\mathbb{R}^n$ can be locally approximated with a lower-dimensional space $\mathbb{R}^m$, with $m \ll n$. The representation of data sets with low-dimensional models has recently been a popular research topic as it is helpful for understanding the intrinsic and meaningful structures in large data sets and facilitates their analysis.

An image set in $\mathbb{R}^n$ or $L^2(\mathbb{R}^2)$ that can be globally parameterized by a few parameters is a manifold if around each image there exists an open neighborhood where the mapping between the parameter domain and the image set is a homeomorphism. In an image processing application, the knowledge of a global parameterization for image sets is highly desirable; the availability of a simple representation with a few degrees of freedom that captures the important characteristics of a large data set in a compact way makes it possible to analyze and process the data efficiently.

While one can find many examples of image processing methods that benefit from low-dimensional image models (e.g., as in [8]), in this thesis we rather focus on the utilization of transformation manifold models in image registration and classification problems. Hence, we begin with giving some examples of parametric image manifolds that have been studied in previous works. We then introduce the setting and notations used in this thesis for the registration and classification

of image sets with manifold models. Next, we give an overview of some manifold learning methods. Finally, we describe the representation of images in structured and parametric dictionaries, which is used in the manifold learning methods proposed in this thesis.

2.1.1 Different families of image manifolds

Since it provides concise and convenient models, the representation of signals with parametric manifolds has been the subject of many previous studies. We mention here a few of them. The work [8] by Peyré describes several manifold models for 1-D signals and image patches, corresponding to smoothly varying images, cartoon images, locally parallel textures and sparse patches. These models are used in the solution of inverse problems such as image inpainting and compressed sensing. Next, in [9], Wakin et al. study image articulation manifolds (IAMs), which are defined as image sets generated by the variation of a parameter controlling the appearance of an image. Some examples of IAMs are manifolds generated by the translations and rotations of an object; or the articulations of a composite object, which change the relative arrangement of its individual components. Lastly, the work [10] by Donoho et al. proposes an interesting study of image manifolds that are isometric to the Euclidean space, in which case the geodesic distances on a manifold of dimension m are proportional to Euclidean distances in $\mathbb{R}^m$. Several examples of image manifolds with this nice property are presented in [10], and include manifolds generated by translating a 4-fold symmetric object, pivoting an object with piecewise smooth boundaries, horizon articulations, and cartoon face articulations.

There are several additional examples of parametrizable image manifolds and the related applications in the literature. In this thesis we however consider the use of manifolds in image analysis, and focus especially on registration and classification applications. We thus discuss these in more details below.

2.1.2 Image analysis with manifold models

We define now the parametric manifold models used for analyzing images in this thesis. We adopt a notation that fits with image transformation manifolds. However, the data analysis problems that we formulate here are generalizable to arbitrary parametrizable signal manifolds.

Let H be a Hilbert space of visual signals (i.e., the discrete space $\mathbb{R}^n$ or the continuous space $L^2(\mathbb{R}^2)$), p denote a visual object, and $\Lambda \subset \mathbb{R}^d$ represent a compact d-dimensional transformation parameter domain. Assume that the transformation parameter vectors λ in Λ act on p such that $U_\lambda(p) \in H$ denotes a geometrically transformed observation of the object p in the space H, where the geometric transformation is specified by the parameter vector λ. We consider a non-degenerate geometric transformation model U_λ; i.e., we assume that, around each λ_0 in the interior of Λ, there exists an open ball $B_\epsilon(\lambda_0) \subset \Lambda$ such that the mapping $U_{(\cdot)}(p) : B_\epsilon(\lambda_0) \to U_{B_\epsilon(\lambda_0)}(p) \subset H$ is a homeomorphism. Then, we call the set $\mathcal{M}(p)$ that consists of all geometrically transformed observations of p over the parameter domain Λ as the transformation manifold[1] of p.

[1]Throughout the thesis, the term "manifold" is used to mean a *smooth manifold* rather than a *topological manifold*. Since the manifold $\mathcal{M}(p)$ is defined with a global parameterization on Λ via the mapping $U_{(\cdot)}(p)$, $\mathcal{M}(p)$ can be written as a union of compatible charts such that the transition between any two intersecting charts is given by the identity diffeomorphism. Note, however, that the assumption that $\mathcal{M}(p)$ is globally parametrizable over a domain $\Lambda \subset \mathbb{R}^d$ is

Definition 1.

$$\mathcal{M}(p) = \{U_\lambda(p), \lambda \in \Lambda\} \subset H \tag{2.1}$$

Once the visual object p is fixed, the transformation $U_{(\cdot)}(p)$ represents a mapping from the parameter domain Λ to H, which constructs the transformation manifold. As it is defined on the d-dimensional parameter domain Λ, the manifold $\mathcal{M}(p)$ has dimension d. Some examples of transformation manifolds that we study are pattern transformation manifolds and object observation manifolds.

Pattern transformation manifolds are generated by the geometric transformations of a 2-D visual pattern p. A typical pattern transformation model that we will consider is

$$\mathcal{M}(p) = \{U_\lambda(p) : \lambda = (\theta, t_x, t_y, s_x, s_y) \in \Lambda\} \tag{2.2}$$

where θ denotes a rotation, t_x and t_y represent translations in x and y directions, and s_x and s_y define an anisotropic scaling in x and y directions. In Figure 1.1, the pattern transformation manifold generated by the rotations and scale changes of a reference pattern is illustrated.

Next, the observation manifold of a 3-D object model p is given by

$$\mathcal{M}(p) = \{U_\lambda(p) : \lambda = (\psi_x, \psi_y, \psi_z) \in \Lambda\} \tag{2.3}$$

where $U_\lambda(p)$ is the image of the object p rendered from the viewpoint specified by the three rotation angles ψ_x, ψ_y, ψ_z. An illustration of object observation manifolds is given in Figure 1.2.

If the mapping $U_{(\cdot)}(p) : \Lambda \to H$ is differentiable, then the inner product $\langle \cdot, \cdot \rangle$ on H induces a Riemannian metric on $\mathcal{M}(p)$ given by

$$\mathcal{G}_{ij}(\lambda) = \left\langle \frac{\partial U_\lambda(p)}{\partial \lambda^i}, \frac{\partial U_\lambda(p)}{\partial \lambda^j} \right\rangle$$

where $\lambda = [\lambda^1 \ \lambda^2 \ \cdots \ \lambda^d]^T$. In this case, $\mathcal{M}(p)$ is a Riemannian manifold. In this thesis, we will mostly study differentiable transformation manifolds, which are Riemannian manifolds. The Riemannian manifold $\mathcal{M}(p)$ is illustrated in Figure 2.1.

Now let $x \in H$ be an image that does not necessarily belong to the manifold. We define the distance of x to the manifold $\mathcal{M}(p)$ as follows.

Definition 2.

$$d(x, \mathcal{M}(p)) = \min_{\lambda \in \Lambda} \|x - U_\lambda(p)\|$$

The distance $d(x, \mathcal{M}(p))$ is also known as the *manifold distance*. In the above definition and throughout the thesis, $\|\cdot\|$ denotes the norm induced by the inner product in H. In particular, $\|\cdot\|$ denotes the L^2-norm for vectors in $L^2(\mathbb{R}^2)$ and the ℓ^2-norm for vectors in $\mathbb{R}^n$. We define the manifold distance using this norm, since it corresponds to the "physical" distance in the signal space H. If $\lambda^* \in \Lambda$ is a parameter vector such that

$$\lambda^* = \arg\min_{\lambda} \|x - U_\lambda(p)\|$$

to make the notation and the analysis easier and is not a generally required condition in the formal definition of a manifold.

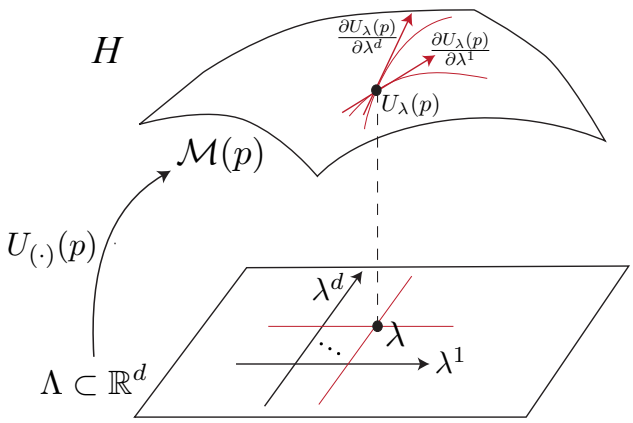


Figure 2.1: Riemannian manifold $\mathcal{M}(p)$ residing in the Hilbert space H. The Riemannian metric is given by the inner products of the tangent vectors $\partial U_\lambda(p)/\partial \lambda^i$.

then we call $U_{\lambda^*}(p)$ a projection of x on $\mathcal{M}(p)$. The projection of an image onto a transformation manifold is typically associated with the registration of the image with respect to a transformation model. Given an image x with an unknown geometric transformation and a reference transformation manifold $\mathcal{M}(p)$, the projection point $U_{\lambda^*}(p)$ gives the best estimate of x that can be obtained with a geometric transformation of p. If x and p represent 2-D images, then the computation of the optimal transformation parameters λ^* that best align the reference image p with the target image x is referred to as an *image registration* problem, which is illustrated in Figure 2.2.

Now let us overview *transformation-invariant image classification* with manifold models. We assume that $H = \mathbb{R}^n$ in this part of the discussion. Consider a collection of visual objects p^1, p^2, $\cdots$, p^M representing M different image classes. Then, the transformation manifolds $\mathcal{M}^1, \mathcal{M}^2, \cdots, \mathcal{M}^M \subset \mathbb{R}^n$,

$$\mathcal{M}^m = \mathcal{M}(p^m) = \{U_\lambda(p^m), \lambda \in \Lambda\}, \quad m = 1, \cdots, M \tag{2.4}$$

defined by the geometric transformations of the visual objects $\{p^m\}$ are transformation-invariant representatives of these M image classes.

In the following, we assume that the transformation manifolds $\{\mathcal{M}^m\}$ constitute sufficiently accurate models for the representation of the image classes $1, \cdots, M$. Note that this assumption holds when the variations within an image class are sufficiently well-captured by the transformations of a single reference model, so that the manifold distance gives a good measure of image dissimilarity. The validity of this assumption depends of course on the application. Nevertheless, we show throughout this thesis that such a setting can be applied to various classification and recognition problems. In this case, a query image is assumed to belong to the class corresponding to the closest manifold. Hence, based on the set of transformation manifolds $\{\mathcal{M}^m\}$, the class label $l(x)$ of a query image x is assigned as

$$l(x) = \arg \min_{m \in \{1, \cdots, M\}} d(x, \mathcal{M}^m). \tag{2.5}$$

Now let us consider a setting with only two image classes represented by the manifolds $\mathcal{M}^m$

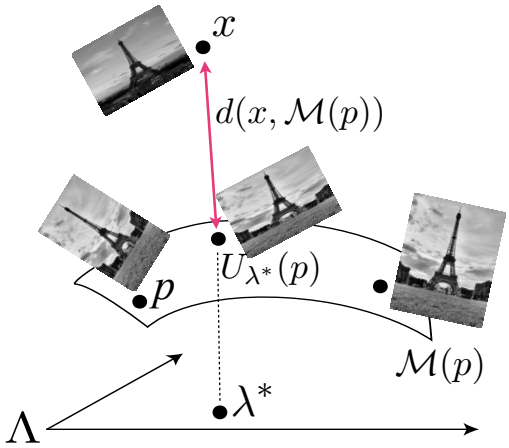


Figure 2.2: Illustration of the registration of a target pattern x with respect to a reference pattern p. The transformation manifold $\mathcal{M}(p)$ of the reference pattern p is the set of geometrically transformed versions $U_\lambda(p)$ of p. If the projection of the target pattern x onto the transformation manifold $\mathcal{M}(p)$ is given by the manifold point $U_{\lambda^*}(p)$, then the parameter vector λ^* defines the geometric transformation that best aligns p with x.

and $\mathcal{M}^r$. In order to characterize the sets of images that belong to the same class as each of these manifolds, we define the half-spaces[2] $\mathcal{H}^{mr}$ and $\mathcal{H}^{rm}$ as follows. We denote by $\mathcal{H}^{mr}$ the set of points whose distance to $\mathcal{M}^m$ is smaller than their distance to $\mathcal{M}^r$ (notice that $\mathcal{H}^{mr} \neq \mathcal{H}^{rm}$):

Definition 3.

$$\mathcal{H}^{mr} = \overline{\{x \in \mathbb{R}^n : d(x, \mathcal{M}^m) < d(x, \mathcal{M}^r)\}}. \tag{2.6}$$

The notation $\overline{(\cdot)}$ in (2.6) denotes the closure of the set. Note that $\mathcal{H}^{mr}$ is defined in this way in order to properly handle the degenerate cases that may be caused by manifold intersections. We then define the decision surface $\mathcal{B}^{mr}$ as the boundary of the half-space $\mathcal{H}^{mr}$,

Definition 4.

$$\mathcal{B}^{mr} = \partial \mathcal{H}^{mr} \tag{2.7}$$

where the notation $\partial(\cdot)$ denotes the boundary of a set. The decision surface $\mathcal{B}^{mr}$ is a combination of hypersurfaces, i.e., a union of $(n-1)$-dimensional manifolds in $\mathbb{R}^n$.

Let us now consider M class representative manifolds instead of two. We define the approximation region $\mathcal{H}^m \subset \mathbb{R}^n$ of the manifold $\mathcal{M}^m$ as follows.

Definition 5.

$$\mathcal{H}^m = \bigcap_{r \in \{1, \cdots, M\} \setminus \{m\}} \mathcal{H}^{mr} \tag{2.8}$$

[2]Although in the standard definition, the boundary surface determining a half-space is an affine hyperplane, here we generalize the term to include the case where the boundary is a hypersurface.

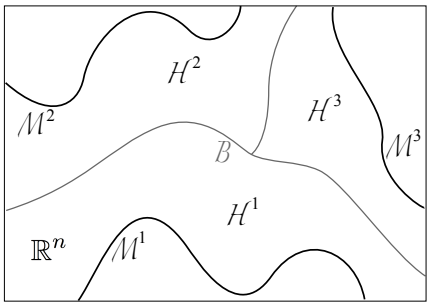


Figure 2.3: Illustration of the partitioning of the signal space into different signal classes. $\mathcal{M}^1$, $\mathcal{M}^2$, $\mathcal{M}^3$ are three manifolds representing different classes; $\mathcal{H}^1$, $\mathcal{H}^2$, and $\mathcal{H}^3$ are their respective approximation regions; and $\mathcal{B}$ is the combined decision surface.

The approximation region $\mathcal{H}^m$ defines the set of images belonging to the m-th class. Finally, in order to adapt the decision surface $\mathcal{B}^{mr}$ determined by two manifolds to the case of multiple manifolds, we define the combined decision surface $\mathcal{B}$

Definition 6.

$$\mathcal{B} = \bigcup_{m=1}^{M} \partial\mathcal{H}^m. \tag{2.9}$$

The combined decision surface $\mathcal{B}$ is a subset of the union of the decision surfaces $\mathcal{B}^{mr}$, i.e., $\mathcal{B} \subset \bigcup_{m\neq r} \mathcal{B}^{mr}$. It forms a boundary between the regions of the space that correspond to different classes, which are determined by the manifolds $\mathcal{M}^1, \mathcal{M}^2, \cdots, \mathcal{M}^M$. An illustration of the class-representative transformation manifolds, the approximation regions $\mathcal{H}^m$ of these manifolds, and the decision surface $\mathcal{B}$ is given in Figure 2.3. We observe that the decision surface $\mathcal{B}$ defines a classification rule in the ambient space $\mathbb{R}^n$. The decision surface $\mathcal{B}$ is determined by the collection $\{\mathcal{M}^m\}$ of transformation manifolds, which are defined by the geometric transformations of class-representative image models. Therefore, the surface $\mathcal{B}$ has a nonlinear structure that is specifically designed to handle geometric transformations in image classification.

Finally, we remark the following. It may not always be possible to partition the whole image space $\mathbb{R}^n$ into the approximation regions of a set of class-representative transformation manifolds. For instance, one may come across degeneracies resulting from manifold intersections and there may exist a full-dimensional subset of the image space that is equidistant to two manifolds. Yet, in this thesis, our treatment relies on the implicit assumption that the training and test images in a transformation-invariant classification application are reasonably close to the representative transformation manifolds of their class. In particular, the sampling algorithm presented in Chapter 3 assumes that most of the training and test images of class m are in the approximation region $\mathcal{H}^m$ of the predetermined class-representative manifold $\mathcal{M}^m$ to be sampled. Similarly, the manifold learning algorithm presented in Chapter 4 is based on the hypothesis that, for each class m, there exists a transformation manifold $\mathcal{M}^m$ around which the training and test images of class m are concentrated.

2.1.3 Manifold learning

Following the discussion on image manifolds and data analysis with manifold models, we provide now a brief overview of manifold learning methods. Manifold learning represents the recovery of low-dimensional structures in high-dimensional signal sets. It typically seeks a good way of mapping the data in the high-dimensional space $\mathbb{R}^n$ to a low-dimensional space $\mathbb{R}^m$, while the objective criterion in the construction of the mapping varies between different methods. For example, the ISOMAP method obtains a mapping that preserves the geodesic distances in the original space [5], whereas methods such as LLE [6] are based on retaining the weights in the locally linear representation of data samples in terms of their neighbors. Laplacian eigenmaps [11] and Hessian eigenmaps [7] methods resemble LLE; the former learns a mapping that conforms to the weights of the data graph constructed in the original space, while the latter computes local coordinates that have vanishing Hessian. Diffusion maps [12] and LTSA [13] are among other well-known manifold learning methods. In diffusion maps, data points are embedded into a low-dimensional Euclidean space such that the Euclidean distance in the low-dimensional domain corresponds to the diffusion distance defined in the original domain. Finally, in LTSA, a global parameterization is computed by aligning the local tangent spaces computed around the data points via affine transformations.

Manifold learning methods such as the ones mentioned above construct a discrete mapping that sends each data point to a vector in a small-dimensional Euclidean space, without learning a general parametric data model; i.e., without learning an explicit formulation of a manifold. For this reason, in contrast to methods that learn parametric models from data samples, they have the following main shortcomings regarding their usage in image analysis. First, they compute a parameterization only for the initially available data, and their generalization for the parameterization of additional data is not straightforward. Second, these methods lack the means of synthesizing new data samples that are on the same manifold. Third, most of the methods that do not allow the synthesis of new data do not have immediate generalizations for classification applications, since the classification problem usually requires the generation of new manifold points for the computation of the manifold distance.

A couple of previous works have sought solutions to these limitations. Concerning the generalization of the learned parameterization to initially unavailable data samples, a method has been proposed in [14], which provides out-of-sample extensions for some common manifold learning algorithms. The authors interpret these algorithms as learning the eigenvectors of a data-dependent kernel, and then generalize the eigenvectors to the continuous domain in order to compute eigenfunctions. Next, the method in [15] computes a smooth tangent field with the use of analytic functions and thus yields a smooth manifold structure that makes the generation of novel points possible. Also, a method is proposed in [16] for synthesizing new images based on the LLE algorithm. Finally, the SLLE algorithm proposed in [17] achieves dimensionality reduction while promoting classification performance, which is obtained by modifying LLE such that the discrimination between different class samples is encouraged in the computation of the data embedding. It however does not provide a solution for the data synthesis problem.

Meanwhile, all of these methods are generic and they make no assumption on the type of the manifold underlying the observed data. Therefore, if they are applied on a data set sampled from a transformation manifold, the embedding computed with these generic methods does not necessarily reflect the real transformation parameters. This significantly limits the usage of these methods for

the particular problem of the analysis of images with geometric transformations. The manifold learning method that we propose in Chapter 4 is motivated by this key observation and it aims to learn a parametric mapping that physically corresponds to actual transformation parameters. It differs from the above methods essentially in the fact that it uses the information of the model that generates the data and employs it for learning an accurate data representation that assigns an exact transformation parameter vector to each data sample. Moreover, it provides straightforward solutions for the issues mentioned above: It allows the parameterization of additional data points and the synthesis of novel manifold points, which leads to successful applications in classification and image analysis problems.

2.1.4 Image representations with parametric dictionaries

A natural way to construct a parametric manifold is to build a representative pattern by selecting atoms from a parametric dictionary, e.g., as in the study presented in [18]. An atom is a waveform that is adapted to the local structures of signals [19], and a dictionary refers to a (typically redundant) collection of atoms. In the manifold learning problem that we study in Chapter 4, it is favorable to use a structured and parametric dictionary instead of an unstructured one, since it allows the formulation of the atom selection problem as an optimization problem on a continuous parameter space. Furthermore, the geometric transformation model can be easily combined with the parametric dictionary model, which is used to devise an efficient learning algorithm. In the image registration analyses presented in Chapters 5 and 6, we also adopt a representation of patterns in a parametric dictionary as it permits us to study the registration problem in an analytic way. We now describe the parametric dictionaries used in this thesis in more details.

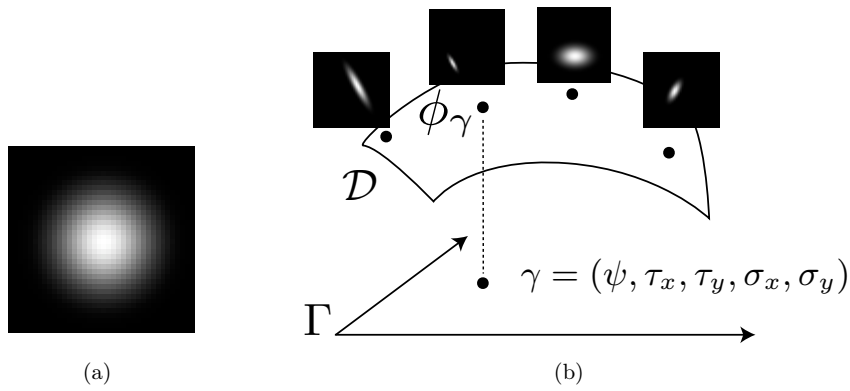


Figure 2.4: Illustration of a parametric Gaussian dictionary. (a) Gaussian mother function ϕ. (b) Dictionary manifold $\mathcal{D}$ generated by the geometric transformations of ϕ. Each atom ϕ_γ on $\mathcal{D}$ is a geometrically transformed version of the mother function ϕ.

We consider the representation of visual patterns in the continuous space of square-integrable functions $L^2(\mathbb{R}^2)$. Let $\phi(X)$ denote an analytic and square-integrable function (i.e., $\phi \in L^2(\mathbb{R}^2)$), where $X = [x\ y]^T \in \mathbb{R}^{2\times 1}$ defines the spatial coordinate variable. We will refer to ϕ as the mother

function and consider the analytic and parametric dictionary

$$\mathcal{D} = \{\phi_\gamma : \gamma = (\psi, \tau_x, \tau_y, \sigma_x, \sigma_y) \in \Gamma\} \subset L^2(\mathbb{R}^2) \tag{2.10}$$

such that each atom ϕ_γ of the dictionary $\mathcal{D}$ is derived from the mother function $\phi \in L^2(\mathbb{R}^2)$ by the geometric transformation specified by its parameter vector γ. In (2.10), ψ is a rotation parameter, τ_x and τ_y denote translations in x and y directions, and σ_x and σ_y represent an anisotropic scaling in x and y directions. The set Γ is the transformation parameter domain over which the dictionary is defined. Then, an atom ϕ_γ is given by

$$\phi_\gamma(X) = \phi(\sigma^{-1}\,\Psi^{-1}\,(X - \tau)), \tag{2.11}$$

where

$$\sigma = \begin{bmatrix} \sigma_x & 0 \\ 0 & \sigma_y \end{bmatrix}, \quad \Psi = \begin{bmatrix} \cos(\psi) & -\sin(\psi) \\ \sin(\psi) & \cos(\psi) \end{bmatrix}, \quad \tau = \begin{bmatrix} \tau_x \\ \tau_y \end{bmatrix}. \tag{2.12}$$

Since the dictionary $\mathcal{D}$ is generated by parametrizable geometric transformations of the mother function ϕ, it is a manifold. In Figure 2.4, the illustration of a dictionary manifold created with a Gaussian mother function is shown.

It is shown in [20] (in the proof of Proposition 2.1.2) that the linear span of a dictionary $\mathcal{D}$ generated with respect to the transformation model in (2.10) is dense in $L^2(\mathbb{R}^2)$ if the mother function ϕ has nontrivial support (unless $\phi(X) = 0$ almost everywhere). In this case, for any pattern $p \in L^2(\mathbb{R}^2)$, there exists a sequence $\{\phi_{\gamma_k}\}$ of atoms in $\mathcal{D}$ such that

$$p = \lim_{K\to\infty} \sum_{k=1}^{K} c_k\, \phi_{\gamma_k} \tag{2.13}$$

where c_k are the atom coefficients. Hence, the above equation gives a parametric and analytic representation of reference patterns in $L^2(\mathbb{R}^2)$. We mostly consider the Gaussian function $\phi(X) = e^{-X^T X} = e^{-(x^2+y^2)}$ for the choice of the mother function ϕ as it has a good spatial localization and it is easy to treat in derivations due to its well-studied properties. This choice also ensures that the linear span of $\mathcal{D}$ is dense in $L^2(\mathbb{R}^2)$; therefore, any pattern $p \in L^2(\mathbb{R}^2)$ can be approximated with a finite set of atoms in $\mathcal{D}$ up to a controllable accuracy. The works presented in Chapters 4-6 of this thesis benefit from the parametric representation of reference patterns given in (2.13), which is helpful for studying the learning and registration of visual patterns.

2.2 Image Analysis

This thesis focuses primarily on solving and analyzing image registration and image classification problems with manifold models. Therefore, in this section, we give a brief overview of image registration and classification methods that have been studied in the literature.

2.2.1 Image registration

Image registration refers to the problem of estimating the coordinate transformation that gives the best approximation of a target image from a reference image. More formally, if $p(X)$ denotes a reference image and $q(X)$ denotes a target image where $X = [x\, y]^T$ is a coordinate vector in $\mathbb{R}^2$, the image registration problem corresponds to the computation of a warping function $w : \mathbb{R}^2 \to \mathbb{R}^2$ that defines a mapping of the coordinates between two images such that the warped version of the reference image is as close as possible to the target image

$$q(X) \approx p(w(X)).$$

The need for aligning images arises in many different applications like image analysis and classification [2], [3], [4], biomedical imaging [21], and stereo vision [22]. The type of the warping function $w(X)$ also varies between different image registration problems. While it is assumed to be given by a parametric transformation model (e.g., translation, rotation, scale change, affine or perspective transformations) in some settings, some geometric deformations can be better modeled with nonparametric transformations such as elastic or piecewise affine transformations [23]. The image registration problem is closely related to the motion estimation problem [24], which is the computation of the displacement of pixels or image blocks between two image frames. In this case, the warping model corresponds simply to a 2-D shift, whose value changes however between different image blocks or pixels.

Image registration methods are mainly categorized as feature-based or region-based methods. Feature-based methods compute the warping function with the help of a discrete set of matched coordinates between the reference and target images, while region-based methods estimate the warping by optimizing an objective function that measures the dissimilarity or the similarity of the images. As we are particularly interested in the usage of manifold models in registration, which correspond to region-based methods with parametric transformation models, here we limit our discussion to region-based image registration. Many region-based methods use the SSD (sum-of-squared difference) as the dissimilarity measure or its approximations [24]. The SSD corresponds to the squared-norm of what is usually called the displaced frame difference (DFD) in motion estimation. The direct correlation is also widely used as a similarity measure [25], [26], and it can be shown to be equivalent to the SSD [27].

Generic registration methods

We begin with reviewing some registration and motion estimation methods that do not use manifold models. In [28], the displacement is computed with a discrete search of the match of each pixel on a search window. In motion estimation literature, several algorithms such as 2-D logarithmic search and increasing accuracy search have been proposed for speeding up the search in the block-matching, by locating a solution coarsely and then refining it progressively [24]. Such methods are based on the hypothesis that the objective image dissimilarity function is convex. In the family of motion estimation methods using the SSD measure, the Netravali-Robbins pel-recursive algorithm [29] and its variations [30], [31], estimate the displacement field by minimizing the SSD over image subregions with gradient descent. There are many methods that minimize

an approximate version of the SSD for a faster implementation, e.g., [32]. Gradient-based optical flow computation methods also belong to this type [24], [26], which exploit a linear approximation of the image intensity function in the estimation of the displacement between two image blocks.

Manifold-based registration methods

Unlike the generic methods mentioned above, manifold-based registration methods use a geometric intuition for estimating a parametric transformation model between an image pair. Some examples of manifold-based methods are as follows. The study in [18] formulates the registration problem as the minimization of the distance between a target image and a pattern transformation manifold and proposes an algorithm to solve it optimally for geometric transformations consisting of translations, rotations, and isotropic scalings. Then, the tangent distance method proposed by Simard et al. in [3] constructs a first-order approximation of the transformation manifold of the reference image by computing the tangent space of the manifold at a reference manifold point. The transformation parameters are then estimated by calculating the orthogonal projection of the target image onto the tangent space of the manifold. The study in [2] is a multiresolution extension of the tangent distance. The recent work [33] uses the tangent distance in motion compensation for video compression. In fact, gradient-based optical flow computation methods can also be interpreted as alignment algorithms that make use of manifold linearizations. Applying a first-order approximation of the intensity function of the reference image block and then computing the displacement in a least-squares manner is equivalent to projecting the target image block onto the linear approximation of the manifold formed by the translations of the reference image block. Therefore, these methods can be seen as a special instance of the tangent distance that is restricted to the transformation model of 2-D translations.

Hierarchical registration strategy

Many registration techniques adopt, or can be coupled with, a multiscale hierarchical search strategy. In hierarchical registration, reference and target images are aligned by applying a coarse-to-fine estimation of the transformation parameters, using a pyramid of low-pass filtered and downsampled versions of the images. Coarse scales of the pyramid are used for a rough estimation of the transformation parameters, where the solution is less likely to get trapped into the local minima of the dissimilarity function as the images are smoothed by low-pass filtering. Moreover, the search complexity is lower at coarse scales as the image pair is downsampled accordingly. The alignment is then refined gradually by moving on to the finer scales. The hierarchical search strategy is widely used in image registration and motion estimation, since it accelerates the algorithm and leads to better solutions with reduced sensitivity to local minima [22], [24], [28], [32], [34]. In particular, it is common practice to apply the tangent distance [2], [3] and gradient-based optical flow estimation techniques [26], [35] in a coarse-to-fine manner as that improves the accuracy of the first-order approximations used in these methods.

Theoretical analysis of registration methods

While generic methods are more suitable for handling arbitrary deformation models includ-

ing nonparametric ones, manifold-based methods are applicable when the warping function $w(X)$ admits a global parameterization. Meanwhile, for the estimation of parametric transformations, manifold models offer a good performance in general. In particular, methods that are based on manifold linearizations are very convenient in many applications, since these methods are model-independent (i.e., they are applicable to any parametric transformation model) and they estimate the transformation very easily with a simple least-squares solution. However, they have the limitation that they are accurate only for small transformations. As mentioned above, this limitation can be partially overcome by using a hierarchical search technique. The theoretical characterization of the accuracy of manifold-based registration methods is important for gaining a better understanding of transformation-invariance in image analysis, which is one of the purposes of this thesis. We overview below some theoretical studies about the performance of methods that estimate parametric transformation models.

To begin with, the work by Robinson et al. [27] studies the estimation of global translation parameters between an image pair corrupted with additive Gaussian noise. The authors derive the Cramér-Rao lower bound (CRLB) on the translation estimation. Given by the inverse of the Fisher information matrix, the CRLB is a general lower bound for the MSE of an estimator that computes a set of parameters from noisy observations. Another work that studies performance bounds in registration is [36], where the CRLB is derived for several geometric transformation models. Next, the article [37] is a recent theoretical study on the accuracy of subpixel block-matching in stereo vision. The paper first examines the relation between the discrete and continuous block-matching distances, and then presents a continuous-domain analysis of the effect of noise on the accuracy of disparity estimation from a rectified stereo pair corrupted with additive Gaussian noise. An estimation of the disparity that globally minimizes the windowed squared-distance between blocks is derived. While the main goal of the paper is to examine the theoretical limits in attaining a subpixel accuracy in block-matching, their results, which provide a characterization of the disparity estimation error with the noise level, are particularly interesting from the standpoint of our study. We give a more detailed discussion of these works in Chapters 5 and 6.

Next, we mention some results related to the performance of multiscale gradient-based optical flow estimation methods. The works in [27], [38] and [39] examine the bias on gradient-based shift estimators and show that smoothing the images reduces the bias on the estimator. However, smoothing also has the undesired effect of impairing the conditioning of the linear system to be solved in gradient-based estimators [38]. Therefore, this tradeoff must be taken into account in the selection of the filter size in coarse-to-fine gradient-based registration. The papers [27], [40] furthermore show that the bias on gradient-based estimators increases as the amount of translation increases. Robinson et al. use this observation to explain the benefits of multiscale gradient-based methods [27]. At large scales, downsampling, which reduces the amount of translation, and smoothing help to decrease the bias on the estimator. Then, as the change in the translation parameters is small at fine scales, the estimation does not suffer from this type of bias anymore. Moreover, at fine scales, the accuracy of the estimation increases as high-frequency components are no more suppressed. This is due to the fact that the CRLB of the estimation is smaller when the bandwidth of the image is larger. Lastly, the analysis in [41] studies the convergence of multiscale gradient-based registration methods where the image pair is related with a 2-D translation. It is shown that, for sufficiently small translations, coarse-to fine gradient-based registration algorithms converge to the globally optimal solution if the images are smoothed with ideal low-pass filters such that the filter

bandwidth is doubled in each stage of the pyramid. However, this convergence guarantee is limited to an ideal noiseless setting where the target image is exactly a translated version of the reference image.

Meanwhile, the previous works analyzing the performance of image registration do not address the following issues. First, the effect of image noise on the alignment error is analyzed at a single scale in works such as [36], [37], whereas the multiscale analysis in [27] is restricted to gradient-based optical flow methods. The examination of the influence of noise on the performance of registration in a multiscale setting is important since many registration methods use a hierarchical alignment strategy. Next, although it is widely known as a practical fact that smoothing an image pair is helpful for overcoming the undesired local minima of the dissimilarity function [42], [24], this has not been studied on a theoretical basis before. We address these problems in Chapter 6 for the geometric transformation model of 2-D translations, where we first characterize the variation of the local minima of the dissimilarity function with low-pass filtering and then examine the joint variation of the alignment error with image noise and smoothing. As we examine the deviation between the global minima of the noiseless distance function and the distance function obtained under noise, the alignment error bounds that we derive have implications for a wide range of displacement estimation methods such as [29]-[32], which use the SSD or its approximations. Lastly, the performance analysis of registration algorithms employing a linearization of the manifold has only been done for translations in works such as [27], [38], [39], [41], which study gradient-based methods. We present a more general treatment of image alignment with manifold linearizations by analyzing the performance of the tangent distance method for arbitrary transformation models in a multiscale setting in Chapter 5.

2.2.2 Image classification

The classification problem can be defined as the determination of the category of data samples. In Bayes decision theory, the classification problem is formalized by assuming a prior probability $P(m)$ for each class m and a conditional probability density $p(x|m)$ for the data samples x of the m-the class [43]. Then, the estimation of the class label of a data sample x is formulated as the maximization of the posterior probability $P(m|x)$, which is the probability that the observation x is an instance of the m-th class. Such a representation of data classes with probabilistic models yields well-defined classification rules defined by the decision boundaries between class probability densities. However, in practice, the probability density functions of data classes are often not known and not easy to estimate, neither. Therefore, the key problem in the design of a classifier is typically to learn a suitable classification rule from data observations. In the following, we first give an overview of generic classification methods that do not assume any data priors, and then discuss some classification methods that are designed for manifold-modeled data.

Generic classification methods

We begin with describing some common classification methods that are based on the projection of high-dimensional data onto linear subspaces in order to handle the excessive dimensionality of data. These methods typically try to learn good subspaces using the training data with known class labels. Then, query data of unknown classes can be classified according to their similarity with the

training data when projected onto the learned subspaces, for instance by nearest neighbor classification. The Principal Component Analysis (PCA) method projects the data onto low-dimensional subspaces generated by the first few principal directions corresponding to the dominant orientations of the data, which are given by the eigenvectors of the data scatter matrix [44]. However, the performance of PCA is not always optimal for classification, since the principal components are not guaranteed to be the directions that are useful for discriminating different classes. On the other hand, Linear Discriminant Analysis (LDA) methods [45], [46] seek a subspace such that the projection of the data on the subspace minimizes the scatter within the classes and maximizes the scatter between different classes; i.e., data samples of the same class are close to each other and the centers of different classes are far from each other on the projected subspace. These subspace-based classification methods have been demonstrated in several pattern recognition problems. For instance, Eigenfaces [47] and Fisherfaces [48] are face recognition algorithms based respectively on PCA and LDA. We also note that quite many recent works study the classification of images by using suitable subspaces that capture well the characteristics of different image classes. For instance, a face recognition algorithm is proposed in [49], where face images are classified by comparing their sparse representations in the subspaces generated by the training samples of each class. The study in [50] models data samples as belonging to a union of subspaces where each subspace is treated as a different class and proposes a method to compute these subspaces based on a sparse representation model. The proposed method is demonstrated in motion segmentation and face clustering problems. Finally, dictionary learning for classification tasks is an active research field, which aims at constructing dictionaries that allow an accurate classification of signals based on their representations in these dictionaries [51].

The subspace-based classification methods mentioned above work well if the data has an approximately linear structure or is linearly separable. However, in quite many applications the data has a nonlinear geometry and different classes are not linearly separable, in which case nonlinear classifiers are preferable. Some common examples of nonlinear classifiers are neural networks and Support Vector Machines (SVM). Neural networks are structures consisting of units called neurons, which are organized in an architecture with (typically multiple) layers [52]. The data sample to be classified is provided as input to the first layer of neurons, and the neurons in each layer output a response that is a linear combination of the response from the previous layer, which is provided to the next layer. In this way, the output layer determines the estimate of the class label of the input data sample. It is common to train neural networks with a back-propagation algorithm, which is based on optimizing the weights of the neuron connections such that a cost function representing the classification error is minimized. A solution for handling the nonlinearity of data in a more effective way is to design the first layer of the network such that it maps the original data to a higher-dimensional space via a nonlinear function called an interpolation function or a kernel, so that the data has higher chances to be linearly separable in the new space. Some popular interpolation functions are polynomials and radial basis functions. The idea of mapping the data to higher dimensional spaces via kernels to achieve linear separability is also at the core of the SVM method [53], which then classifies the data by separating them with hyperplanes constructed using the samples that are closest to the decision boundaries.

Note that it is common to use an alternative representation of the data in a feature space instead of the original image space in such classification methods. The types of features that give a good classification accuracy depend of course on the application and vary on a wide span including trans-

form domain features [52] (e.g., Fourier, Wavelet, Hadamard coefficients), regional features such as SIFT keypoints [54] and local binary patterns [55], and shape descriptors such as chain codes [56].

Classification methods for manifold-modeled data

The above classification methods are generic and they do not rely on any specific assumption on the geometry of data. Meanwhile, when the data is likely to be sampled from a manifold instead of being arbitrarily spread in the ambient space, it is useful to exploit this special structure in the learning. We now mention a couple of classification techniques that are based on this hypothesis.

In several classification methods, the exploitation of the manifold structure of data is through local linearity assumptions and the hypothesis that neighboring samples on the same manifold have a small distance, as opposed to the usage of a parametric representation of the data manifold. Therefore, these methods are nonparametric.

Among nonparametric methods, graph-based methods are algorithms that construct a data graph such that each node of the graph is a data sample, where the weights between connected nodes are typically set according to the distance between the data samples. In these methods, the edges between nodes are usually assigned such that each node is connected to a predetermined number of nearest neighbors (k-nearest neighbors), or its neighbors within a sphere of a predefined radius (ϵ-neighborhood). Such a construction of the data graph makes these methods a good choice when the data is concentrated around a manifold and has a locally flat structure, since neighboring nodes in the graph correspond to neighboring manifold points in this case. In graph-based methods, data classification is usually formulated as a semi-supervised learning problem, such that the data with both known and unknown class labels are used in the construction of the graph. The unknown class labels are then estimated based on the similarity between the data samples, which is captured by the edge weights. There are several ways to estimate the unknown class labels. One way is to use graph mincuts [57]. In a setting with binary class labels, this method seeks a minimum set of edges such that when these edges are removed, the resulting graph yields two disconnected components corresponding to the two classes. Another example of a graph-based method is the label propagation algorithm [58], which proposes to estimate class labels by assigning soft labels to the nodes and letting all nodes propagate their soft labels to their neighbors according to a probabilistic transition model. The nodes with known class labels are exploited by setting their soft labels to the correct class labels at each iteration, which drives the learning. In this way, the algorithm converges to a solution that gives an estimate of the labels of all nodes.

Next, the classification method proposed in [59] assumes that the data points reside on a manifold, which contain labeled and unlabeled samples from two classes. It learns a function on the data graph as a linear combination of the first few eigenvectors of the graph Laplacian such that the values of the function at labeled data samples approximate well their class labels. The construction of the classifier function in terms of the eigenvectors of the graph Laplacian is helpful for controlling its smoothness on the manifold, so that nearby samples are encouraged to have the same class labels. Finally, the SLLE method presented in [17] proposes the classification of data samples lying on a manifold via dimensionality reduction. It learns a mapping of the data to a lower-dimensional space by introducing a separation between different classes. The class labels of new data samples are then estimated in the lower-dimensional space using a simple method such

as nearest mean or nearest neighbor classifier.

While all of the above methods are based on the assumption that all data samples lie on a single manifold, there are also several studies that model different classes with different manifolds like we do in this thesis. We mention here a few of them. First, the texture classification method proposed in [60] relies on a clustering algorithm where each cluster is considered as a different manifold. The clusters are then computed by minimizing the total geodesic distance between data samples and the representatives of their cluster. Another manifold-based classification method is presented in [61], which proposes an algorithm for human face recognition in videos. The images of each person are modeled as a different manifold given by a union of planes that approximate different poses of that person, where the planes are connected to each other with transition probabilities. The recognition of faces in the video is then achieved with a maximum a posteriori estimation, which takes the transitions between consecutive frames into account. Finally, the recent study in [62] proposes a method to learn multiple manifolds that represent different facial expressions. Then the facial expression in a query image is estimated by identifying the manifold that gives the smallest reconstruction error for the image.

In contrast to nonparametric methods described above, parametric methods are algorithms that explicitly make use of a parametric representation of the manifold model in classification. Some examples to parametric manifold-based classification methods are the following. The tangent distance method [3] proposes a solution for transformation-invariant classification by estimating the manifold distance by making use of a linear approximation of the manifolds, which is demonstrated in handwritten digit recognition applications. The multiresolution extension of the tangent distance proposed in [2] is applied in face recognition and semantic video classification problems. The work in [4] proposes to measure image dissimilarities with a metric called the joint manifold distance. The joint manifold distance is given by the subspace-to-subspace distance between linear manifold approximations; however, the proposed formulation involves priors on the image distributions as well. The method is used to cluster faces in videos. Lastly, a generalized maximum likelihood classifier is described in [63], where each class is represented by a different parametrizable image appearance manifold, and the class labels of image observations are estimated according to their distances to the manifolds as in our framework. This setting is then used for developing a compressive classifier called the smashed filter, which estimates the class labels of data samples from compressed measurements.

Finally, we remark the following about the classification of image sets with geometric transformations, which is studied in this thesis. This kind of image data has a highly nonlinear structure; therefore, linear classifiers fail for such data. One may get slightly better results through the use of nonlinear classifiers; however, the generic kernels used in these classifiers do not necessarily match well the particular geometric structure of such image sets. Since geometrically transformed image sets can be assumed to be concentrated around transformation manifolds, manifold-based methods give better results for their classification in general. Meanwhile, manifold-based classification methods have some limitations as well. First, graph-based or nonparametric algorithms are susceptible to the sampling conditions of data, such as the sampling density and noise. The sampling density becomes an important issue especially when the dimension of the manifold is

high.[3] Parametric methods such as [3], [2], [4] provide an analytic way to estimate the manifold distance; however, they have the drawback that their estimation is accurate only for small transformations. In this thesis, we aim at developing efficient image classification methods based on parametric transformation-manifold models that can handle large transformations as well. We thus study the approximation of manifold distance with manifold samplings in Chapter 3 and propose methods to construct transformation manifolds that are good for classification in Chapter 4. Then, in Chapters 5 and 6 we theoretically examine how the performance of parametric methods relying on the estimation of the manifold distance can be improved with a multiscale analysis.

[3]The analysis in our recent study [64] shows that, for the local linearity assumptions to hold, the ambient space distance between data samples must decrease at a rate of $O(d^{-3/2})$ with the dimension d of the manifold.

Chapter 3

Sampling Parametrizable Manifolds

3.1 Fast Manifold Distance Estimation with Sampling

In this chapter, we address the problem of sampling transformation manifolds for accurate estimation of the manifold distance in image registration and classification. The exact computation of the manifold distance is in general a complicated problem, mainly due to the variety and complexity of the involved transformation models. Among the previous works that propose solutions for the manifold distance computation problem, studies such as [2], [3], [4] are based on first-order approximations of the manifold. However, such methods perform well especially when the relative transformation between the target image and the reference image is small, and the first-order approximation loses accuracy for large transformations. There are also some works that study the estimation of transformation parameters for specific types of geometric transformation models, e.g., [18], [65]. Meanwhile, there is no known solution for the computation of the manifold distance optimally for generic transformation models. A simple and practical way to estimate the manifold distance is then to represent the manifold with a finite grid of manifold samples, where the distance between a query image and its projection onto the manifold is approximated by the distance between the image and the nearest manifold sample. The usage of such a grid improves the complexity of distance estimation immensely, possibly at the price of a lower distance accuracy.

In image analysis applications, it is common practice to sample manifolds in a straightforward way by generating a grid regular in the parameter domain. However, a regular discretization in the parameter space is not guaranteed to offer a good performance, especially when the number of samples is limited. While the choice of the manifold grid has considerable influence on the accuracy of manifold distance estimation, the manifold sampling problem has not been given much consideration so far within the context of image analysis. Structured grid generation has been well-studied especially for analytical two-dimensional surfaces in $\mathbb{R}^3$, mostly for the purpose of obtaining finite-difference solutions to partial differential equations [66]. It is also possible to find sampling solutions for surfaces represented in non-analytical forms such as meshes [67]. Even though some of these sampling methods may in principle be generalized for image manifolds of arbitrary dimension, the targeted applications must be taken into account in grid generation.

In this chapter, we study the distance-based discretization of transformation manifolds of known parameterization. We first present a manifold discretization algorithm that minimizes the manifold

distance estimation error stemming from the representation of the manifold by finitely many grid points. Our discretization method bears some resemblance to the LBG vector quantization algorithm [68] due to the alternating optimization steps it involves, where the representative samples for a given partition of the space are computed, and then the space is repartitioned for the updated set of samples. However, the proposed method differs essentially from the LBG algorithm, since it targets the minimization of the manifold distance with samples positioned on the manifold and does not have a signal approximation objective.

Noting the dependency between the registration and classification performances, we then extend this sampling solution to the joint discretization of multiple transformation manifolds representing different classes. As discussed in Chapter 2, the estimation of the class label m of a query image x requires the determination of the approximation region $\mathcal{H}^m$ it lies in. We assume that the exact knowledge of the manifolds determines the class label of an image perfectly. A discrete representation of the manifolds reduces the complexity of the classification problem, while the classification performance in the discrete setting depends significantly on the sampling. We propose a discretization method where all manifolds are jointly sampled such that the relative geometries of different manifolds are taken into account to yield a good classification accuracy. Experimental results show that the proposed discretization methods yield better registration and classification performance than basic discretizations such as random grids or regular grids. Moreover, the consideration of the relative properties of manifolds in the sampling in addition to their individual properties improves the classification accuracy.

In the manifold discretization study presented in this chapter, we essentially focus on transformation manifolds. However, we maintain a generic formulation that it is applicable to arbitrary parametric signal manifolds. We also note that parametrizable signal manifolds are not restricted to image manifolds, which could find examples within acoustic and seismic signals for instance [8], [69].

This chapter is organized as follows. In Section 3.2 we overview the discretization of parametric manifolds based on distance estimation, and in Section 3.3 we propose an extension of the registration-based sampling solution for classification. We present experimental results in Section 3.4, and conclude in Section 3.5.

3.2 Manifold Discretization for Minimal Distance Estimation Error

In this section, we present an iterative method for the optimization of manifold samples such that the manifold distance estimation error caused by representing the manifold with a finite number of samples is minimized. Note that the main purpose of the sampling scheme proposed here is the accurate estimation of the manifold distance for registration applications rather than the approximation of the manifold, which are not necessarily equivalent.

We consider signal manifolds residing in $\mathbb{R}^n$ that are defined by a parametric model as in (2.1). We assume that a generating signal pattern p, a compact parameter domain Λ and a bounded mapping $U_{(\cdot)}(p)$ between Λ and $\mathbb{R}^n$ are given. Considering the pattern p to be fixed, we denote the manifold $\mathcal{M}(p)$ simply as $\mathcal{M}$. We formulate the discretization of the manifold $\mathcal{M}$ as the selection of a predetermined number N of manifold points; i.e., a sample set $\mathcal{S} = \{S_i\} = \{U_{\lambda_i}(p)\} \subset \mathcal{M}$, $i = 1, \cdots, N$ for some $\{\lambda_1, \cdots, \lambda_N\} \subset \Lambda$. We would like to select a set of samples that minimizes

the total manifold distance estimation error E over R, where R is a bounded region in the space $\mathbb{R}^n$. We consider R to be a region of interest, which depends on the application. We define the error E as

$$E = \int_R \left(d^2(x, \mathcal{S}) - d^2(x, \mathcal{M})\right) dx \tag{3.1}$$

where

$$d(x, \mathcal{S}) = \min_{i \in \{1,2,\cdots,N\}} \|x - S_i\| \tag{3.2}$$

denotes the distance between x and the sample set $\mathcal{S}$. The formulation of the error in terms of squared distances is for the ease of analytical manipulation.

For a given sample set, one can partition R into N regions as $R = \bigcup_{i=1}^{N} R_i$, where each R_i is a region consisting of points with smallest ℓ^2-distance to S_i among all samples, i.e.,

$$R_i = \{x \in R : \|x - S_i\| \leq \|x - S_j\|, \forall j \in \{1, \cdots, N\}\}. \tag{3.3}$$

The regions R_i are thus defined similarly to the Voronoi cells in the LBG vector quantization method or Lloyd's quantization algorithm [68]. Then, the total manifold distance estimation error can be written as

$$E = \sum_{i=1}^{N} E_i = \sum_{i=1}^{N} \int_{R_i} \left(\|x - S_i\|^2 - d^2(x, \mathcal{M})\right) dx. \tag{3.4}$$

In order to minimize the error E, we follow an iterative optimization procedure. In each iteration of the algorithm a two-stage optimization is employed: In the first stage, we fix the samples S_i and determine the partition regions R_i corresponding to the samples. In the actual implementation of the method, we numerically determine the regions R_i with the help of training data. Then, in the second stage, we fix the regions R_i and optimize each sample S_i individually such that the error E_i in the regarding region is minimized. The minimization of the manifold distance estimation error E_i within a specific region R_i is achieved as follows. The error term E_i can be rearranged as

$$E_i = \int_{R_i} \|x - S_i\|^2 \, dx - \int_{R_i} d^2(x, \mathcal{M}) \, dx$$

where the second integration depends only on R_i, and is constant with respect to S_i as R_i is treated as a fixed parameter. Therefore, E_i is given by

$$E_i = \int_{R_i} \|x - S_i\|^2 \, dx + c_i = \int_{R_i} x^{\mathrm{T}} x \, dx - 2S_i^{\mathrm{T}} \int_{R_i} x \, dx + V_i S_i^{\mathrm{T}} S_i + c_i,$$

where c_i is a constant independent of S_i, and $V_i = \int_{R_i} dx$ is the volume of the region R_i. Denoting the centroid of R_i by $G_i = (\int_{R_i} x \, dx)/(\int_{R_i} dx)$, we get

$$E_i = \int_{R_i} x^{\mathrm{T}} x \, dx + V_i(-2S_i^{\mathrm{T}} G_i + S_i^{\mathrm{T}} S_i) + c_i = V_i(S_i^{\mathrm{T}} S_i - 2S_i^{\mathrm{T}} G_i) + c_i',$$

where we express the sum of the terms independent of S_i by c_i'. As E_i differs from $\|S_i - G_i\|^2$ only up to a positive multiplicative factor and an additive term constant with respect to S_i, one can equivalently minimize

$$\varepsilon_i = \|S_i - G_i\|^2 \tag{3.5}$$

at each iteration of the algorithm. This actually means that S_i should be selected as the manifold point closest to the centroid of the region R_i.

The following is a summary of the procedure we apply for obtaining a manifold discretization that minimizes the total manifold distance estimation error. Given the available domain of parameters and the mapping defining the manifold, we begin with an initial sample set $\mathcal{S}(0) = \{S_i(0)\}$ on the manifold, which is possibly randomly selected. We optimize the sample set iteratively. In iteration k of the algorithm, we first compute the regions $\{R_i(k)\}$ that partition R with respect to the manifold samples $\{S_i(k)\}$, and then we modify each sample $S_i(k)$ individually to obtain the new sample $S_i(k+1)$ such that the manifold distance estimation error given by (3.5) is minimized in the corresponding region. The new sample $S_i(k+1)$ is the projection of the centroid $G_i(k)$ onto the manifold. Iterations are repeated until improvements become negligible. We call this algorithm Registration-Efficient Manifold Discretization (REMD). An iteration of the algorithm is illustrated in Figure 3.1, and the pseudocode is given in Algorithm 1.

As the parameter domain Λ is compact and the mapping $U_{(\cdot)}(p)$ is bounded, for a given number of samples N, there exists a solution $\mathcal{S}^*$ that globally minimizes the total error E in (3.1). At each iteration of the method, first the partition regions are updated and then the samples are readjusted, both of which are modifications that either reduce E or retain it. Since the error E is non-increasing throughout the iterations and is also lower bounded, it converges. However, in general the cost function E is a non-convex, complicated function of the optimization variables $\{\lambda_i\}_{i=1}^N$; therefore, the algorithm is not guaranteed to find a globally optimal solution. This could be mitigated by the choice of a good initial distribution of samples. For instance, in order to begin with a balanced sample distribution, a preliminary stage can be added before the main iterations. Here one can impose the condition that the pairwise distance between any two samples in the ambient space or the parameter space is larger than some threshold value.

Lastly, although the error E converges, the output of the algorithm (the sample locations) is not theoretically guaranteed to converge. The study of the convergence of the Lloyd algorithm and its extensions is indeed an active research topic and recent works such as [70] provide some conditions under which the Lloyd algorithm converges in one-dimensional and multi-dimensional data spaces. While the generalization of such results to the manifold setting considered in this chapter remains as an interesting future study, we note that the convergence of the algorithm is not critical for the utilization of the algorithm output in practical applications: Different sample sets that yield the same distance estimation error are quite likely to perform similarly in practice.

3.3 Classification-Based Discretization of Multiple Manifolds

We have examined above a discretization solution for a single signal manifold based on the minimization of the distance estimation error given by the approximation of the manifold with a set of samples. Now we consider the sampling problem with multiple signal manifolds. We consider that the manifold distance is computed with a discrete set of samples from each manifold, and the

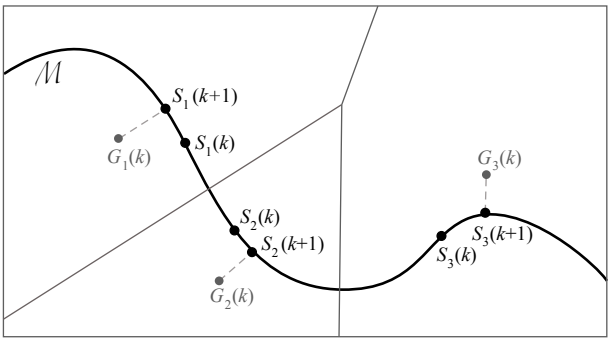


Figure 3.1: Illustration of a single iteration of the algorithm with three samples: Each $G_i(k)$ is the centroid of the partition region corresponding to the sample $S_i(k)$ at the kth iteration. The updated sample $S_i(k+1)$ is the projection of the centroid $G_i(k)$ onto the manifold.

estimated class label of a signal is the label of the nearest manifold sample. Clearly, the accuracy of a sampling in classification is highly related to its accuracy in distance approximation. One possible solution to the multiple manifold discretization problem is to sample each class-representative manifold independently with the REMD algorithm presented in Section 3.2. Although this brings a certain improvement in the classification performance compared to baseline sampling solutions such as regular or random sampling, it fails in taking into account the geometric properties of different manifolds relative to each other. A better approach to the multiple discretization problem is the joint discretization of all manifolds.

Furthermore, given a fixed budget for the total number of manifold samples, which can also be interpreted as a fixed computational complexity for classification, we would like to determine how many samples should be selected from each manifold such that the overall classification accuracy is maximized. The ease of representing a manifold by a discrete sample set is highly dependent on the manifold geometry. The sample budget may thus vary for different manifolds. Moreover, in the determination of the budget allocation, the relative configuration of the manifolds must also be taken into account. For instance, if a subgroup of manifolds are more likely to lead to misclassifications because of their internal resemblance, then it may be better to allocate them a higher number of samples.

In Section 3.3.1, we first formulate the classification accuracy of a multiple manifold discretization, then in Section 3.3.2 we describe an iterative algorithm for sampling multiple signal manifolds that aims to improve the classification accuracy gradually. In Section 3.3.3 we discuss some approaches for determining the allocation of the overall sample budget to different manifolds.

3.3.1 Classification with discrete samples on manifolds

The classification of signals with discretized manifolds can be formulated as follows. We consider M signal manifolds $\mathcal{M}^1, \cdots, \mathcal{M}^M$ generated from M class-representative signal patterns $p^1, \cdots, p^M$ with parametrizable models as defined in (2.4). Each manifold $\mathcal{M}^m$ is approximated by a finite set of N_m samples

$$\mathcal{S}^m = \{S_i^m\} = \{U_{\lambda_i^m}(p^m)\} \subset \mathcal{M}^m \tag{3.6}$$

Algorithm 1 Registration-Efficient Manifold Discretization

1: **Input:**
Λ: Parameter vector domain
$U_{(\cdot)}(p)$: Mapping from parameter domain Λ to manifold $\mathcal{M}$
N: Number of manifold samples
2: **Initialization:**
3: Choose an initial set of manifold samples $\mathcal{S}(0) = \{S_i(0)\}$, $i = 1, \cdots, N$.
4: $k = 0$.
5: **repeat**
6: Determine the partition regions $\{R_i(k)\}$.
7: Compute the centroids $\{G_i(k)\}$ of the regions.
8: Update each sample $S_i(k)$ to $S_i(k+1)$, which is the projection of the centroid $G_i(k)$ on the manifold.
9: $k = k + 1$.
10: **until** Error E converges
11: $\mathcal{S} = \mathcal{S}(k)$.
12: **Output**:
$\mathcal{S} = \{S_i\}$: A set of manifold samples

for $i = 1, \cdots, N_m$, where $\lambda_i^m \in \Lambda$ is the parameter vector generating the sample S_i^m. The classification of a test signal corresponds to the determination of the manifold with smallest distance to it. Given a signal $x \in \mathbb{R}^n$, the estimate $\hat{l}(x)$ of its true class label $l(x)$ is given by the class label of its nearest neighbour among all manifold samples

$$\hat{l}(x) = \arg\min_m d(x, \mathcal{S}^m) = \arg\min_m \left(\min_{i \in \{1,..,N_m\}} \|x - S_i^m\| \right). \tag{3.7}$$

We analyze now the classification error yielded by the discretization of manifolds. Let $R \subset \mathbb{R}^n$ denote a bounded region of interest in the signal space. For each manifold $\mathcal{M}^m$, we define a partitioning of R into regions $\{R_i^m\}$, where each region consists of points closest to a specific sample S_i^m of $\mathcal{M}^m$ among its all samples,

$$\begin{aligned} R &= \bigcup_{i=1}^{N_m} R_i^m \\ R_i^m &= \{x \in R : \|x - S_i^m\| \leq \|x - S_j^m\|, \ j \in \{1, \cdots, N_m\}\}. \end{aligned} \tag{3.8}$$

Now, consider a signal $x \in R_i^m$ that is of class m. Then $x \in R_i^m \cap \mathcal{H}^m$. Depending on the distribution of samples, x can be correctly classified only if its distance to S_i^m is the smallest among its distances to all manifold samples. Thus, we define a function $E_i^m : R_i^m \to \{0, 1\}$ such that it represents the classification error for signals of class m in the region R_i^m.

$$E_i^m(x) = \begin{cases} 1 & \text{if } x \in \mathcal{H}^m \text{ and } \|x - S_i^m\| > d\big(x, \bigcup_{r \neq m} \mathcal{S}^r\big) \\ 0 & \text{otherwise} \end{cases} \tag{3.9}$$

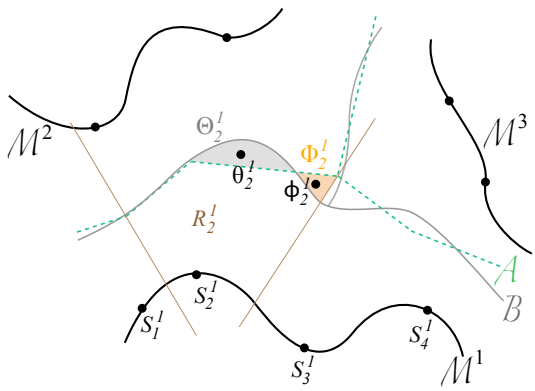


Figure 3.2: Illustration of transformation-invariant signal classification via transformation manifolds

Then, from (3.9) we define the total classification error

$$E = \sum_{m=1}^{M} \sum_{i=1}^{N_m} \int_{R_i^m} E_i^m(x)\, dx. \tag{3.10}$$

Notice that due to the definition of the error $E_i^m(x)$, the total classification error E corresponds to the sum of the volumes of the regions in R where signals are not correctly classified. Another source of misclassification associated with a sample S_i^m corresponds to the points that are actually closer to another manifold than $\mathcal{M}^m$, but are misclassified as a result of being closer to S_i^m than their nearest manifold sample of the correct class. Hence, in analogy with E_i^m, we can define an alternative classification error function $F_i^m : R_i^m \to \{0, 1\}$ as

$$F_i^m(x) = \begin{cases} 1 & \text{if } x \notin \mathcal{H}^m \text{ and } \|x - S_i^m\| < d\big(x, \bigcup_{r \neq m} \mathcal{S}^r\big) \\ 0 & \text{otherwise} \end{cases} \tag{3.11}$$

which leads to the following alternative formulation of the total classification error

$$F = \sum_{m=1}^{M} \sum_{i=1}^{N_m} \int_{R_i^m} F_i^m(x)\, dx. \tag{3.12}$$

The classification errors E in (3.10) and F in (3.12) are equal. However, due to the two mentioned sources of misclassification associated with a single sample S_i^m, we formulate the classification error as the combination of the two. The reason for this choice is made more clear in the algorithm description in Section 3.3.2. Hence we write the total classification error as

$$\varepsilon = \frac{1}{2} \sum_{m=1}^{M} \sum_{i=1}^{N_m} \int_{R_i^m} \big(E_i^m(x) + F_i^m(x)\big)\, dx \tag{3.13}$$

where $\varepsilon = E = F$.

Note that a geometric interpretation of the problem is the following. Let $\tilde{R}_i^m$ denote the region

of space with smallest distance to the sample S_i^m among the samples of all manifolds

$$\tilde{R}_i^m = \{x \in R : \|x - S_i^m\| \leq \|x - S_j^r\|,\ \forall r \in \{1, \cdots, M\}, \forall j \in \{1, \cdots, N_r\}\} \tag{3.14}$$

where $\tilde{R}_i^m \subset R_i^m$. Also, let $\mathcal{A}_{ij}^{mr}$ denote the affine hyperplane which is at equal distance to the two samples S_i^m and S_j^r belonging to two different manifolds $\mathcal{M}^m$ and $\mathcal{M}^r$. In analogy with the way that the decision surface $\mathcal{B}$ is defined in Chapter 2 as the boundary for the determination of the true classes $l(x)$ of signals, let now $\mathcal{A}$ denote the decision surface that determines the class label estimates $\hat{l}(x)$ of signals based on the manifold approximations with samples. Hence, we define

$$\mathcal{A} = \bigcup_{m=1}^{M} \partial(\bigcup_{i=1}^{N_m} \tilde{R}_i^m), \tag{3.15}$$

where $\mathcal{A} \subset \bigcup_{m,r,i,j} \mathcal{A}_{ij}^{mr}$. The definition of the classification error function ε in (3.13) corresponds to the total volume of the regions between the true boundary $\mathcal{B}$ and its approximation $\mathcal{A}$. Therefore, the problem of minimizing the classification error can be regarded geometrically as the selection of the sample sets $\bigcup_{m=1}^{M} \mathcal{S}^m$ such that the resulting $\mathcal{A}$ constitutes an accurate approximation of $\mathcal{B}$ inside R. This is illustrated in Figure 3.2.

3.3.2 Discretization algorithm

We would like to minimize the classification error ε in (3.13) by optimizing the sample sets $\bigcup_{m=1}^{M} \mathcal{S}^m$. In order to achieve this, we suggest an iterative procedure as follows. We start with an initial set of samples. Then, in each iteration we optimize one manifold sample S_i^m and try to reduce ε by perturbing S_i^m. However, the dependence of the classification error on the location of a sample S_i^m is fairly intricate, and it is not simple to determine the optimal sample location. Hence, in the minimization of the error, we adopt a constructive approach rather than optimal; therefore, the search directions in the perturbation of a sample may not always decrease the overall error. In order to handle this, we accept an update on a sample location only if it reduces the classification error. After reaching a locally optimum error with the perturbation of the single sample S_i^m, we repeat this process with different manifold samples until the stabilization of the classification error. Constraining the classification error to be non-increasing assures the termination of the algorithm. The overall procedure is not guaranteed to find the globally optimal solution and the accuracy of the final sampling is influenced by the initialization of the samples.

In a single iteration of the algorithm, we would like to find an update on S_i^m that reduces the error ε, where the rest of the samples are considered to be fixed. The examination of the error term in (3.13) reveals that the effect of the sample S_i^m on ε is twofold. The terms $E_i^m(x)$ and $F_i^m(x)$ involve the distance of space points $\|x - S_i^m\|$ to the sample S_i^m, but the region of integration R_i^m is also defined by the position of the sample S_i^m. Hence, the classification error has a complicated dependence on the sample location S_i^m. Let Θ_i^m and Φ_i^m denote the regions of R_i^m where $E_i^m(x) = 1$ and $F_i^m(x) = 1$ respectively (illustrated in Figure 3.2).

$$\Theta_i^m = \big\{x \in R_i^m \cap \mathcal{H}^m : \|x - S_i^m\| > d\big(x, \bigcup_{r\neq m} \mathcal{S}^r\big)\big\} \tag{3.16}$$

$$\Phi_i^m = \big\{x \in R_i^m \setminus \mathcal{H}^m : \|x - S_i^m\| < d\big(x, \bigcup_{r\neq m} \mathcal{S}^r\big)\big\} \tag{3.17}$$

The error terms in the expression (3.13) contributing to ε are in fact the sum of the volumes of these two regions Θ_i^m and Φ_i^m. Therefore, in order to reduce the error ε, we seek an update on S_i^m that decreases the volumes of Θ_i^m and Φ_i^m. Let $S_i^m(k)$ be the location of the sample S_i^m in iteration k of the search algorithm, and let $R_i^m(k)$, $\Theta_i^m(k)$, and $\Phi_i^m(k)$ be defined similarly. The definitions (3.16) and (3.17) suggest that decreasing the distance $\|x - S_i^m(k)\|$ between the sample and the points in $\Theta_i^m(k)$ reduces the misclassified portion of $\Theta_i^m(k)$. Similarly, it is necessary to increase the distance $\|x - S_i^m(k)\|$ between the sample and the points in $\Phi_i^m(k)$ in order to reduce the misclassified portion of $\Phi_i^m(k)$. Hence, we define the distance measures $D_\Theta(S_i^m(k))$ and $D_\Phi(S_i^m(k))$ as follows

$$D_\Theta(S_i^m(k)) = \int_{\Theta_i^m(k)} \|x - S_i^m(k)\|^2 \, dx \tag{3.18}$$

$$D_\Phi(S_i^m(k)) = \int_{\Phi_i^m(k)} \|x - S_i^m(k)\|^2 \, dx. \tag{3.19}$$

As discussed in Section 3.2, the minimization of $D_\Theta(S_i^m(k))$ is possible by minimizing the distance $\|\theta_i^m(k) - S_i^m(k)\|$, where $\theta_i^m(k)$ is the centroid of $\Theta_i^m(k)$. Similarly, in order to maximize $D_\Phi(S_i^m(k))$, one should maximize $\|\phi_i^m(k) - S_i^m(k)\|$, where $\phi_i^m(k)$ is the centroid of $\Phi_i^m(k)$. However, even an update on $S_i^m(k)$ that decreases $D_\Theta(S_i^m(k))$ and simultaneously increases $D_\Phi(S_i^m(k))$ does not guarantee that the total classification error ε decreases. This is because in general $\Theta_i^m(k+1) \not\subset \Theta_i^m(k)$ and $\Phi_i^m(k+1) \not\subset \Phi_i^m(k)$. Even if the error is reduced within $\Theta_i^m(k)$, $\Theta_i^m(k+1)$ might contain points that are not inside $\Theta_i^m(k)$ and actually increase ε. Still, when one aims to reduce ε by perturbing only $S_i^m(k)$ in a given configuration of the samples and manifolds, curing the immediate regions of misclassification $\Theta_i^m(k)$ and $\Phi_i^m(k)$ is a promising attempt. We thus propose to update the sample $S_i^m(k)$ in the following way as long as the overall error does not increase.

Let $\mu_i^m(k)$ and $\nu_i^m(k)$ denote the parameter vectors corresponding respectively to the projections of the centroids $\theta_i^m(k)$ and $\phi_i^m(k)$ on the manifold. Then, the purpose of moving $S_i^m(k)$ closer to $\theta_i^m(k)$ and away from $\phi_i^m(k)$ leads to the following two updates

$$S_i^m(k+1) = U_\eta(p^m), \qquad \eta = \big((1-\alpha)\lambda_i^m(k) + \alpha\,\mu_i^m(k)\big) \text{ such that } \alpha \text{ minimizes } \varepsilon \tag{3.20}$$

$$S_i^m(k+1) = U_\eta(p^m), \qquad \eta = \big((1+\beta)\lambda_i^m(k) - \beta\,\nu_i^m(k)\big) \text{ such that } \beta \text{ minimizes } \varepsilon \tag{3.21}$$

where $\lambda_i^m(k)$ is the parameter vector defining $S_i^m(k)$ and both α and β are positive scalars. Hence we determine the directions of perturbation with respect to the centroids of the misclassified volumes, and we adjust the amount of perturbation to obtain the largest decrease in the error. In this

way, we find a locally optimum sample location reducing the misclassified portions of $\Theta_i^m(k)$ and $\Phi_i^m(k)$. It also guarantees that the possible penalty of creating new misclassified regions by moving the sample is always smaller than the benefit of correcting previous misclassifications. In the optimization of a single sample S_i^m, we alternate between the updates in (3.20) and (3.21) until convergence, where the parameters $\mu_i^m(k)$ and $\nu_i^m(k)$ are updated after each perturbation. Then we continue the optimization process by picking other manifold samples and applying the same procedure until the classification error is stabilized. We call this algorithm Classification-Driven Manifold Discretization (CMD) and give an overview of it in Algorithm 2. Finally, we note that in each iteration the perturbation of a manifold sample in the described way corresponds to a one-dimensional search in the d-dimensional parameter space. Although the algorithm is not guaranteed to be optimal, it offers a compromise in the performance-complexity trade-off.

Algorithm 2 Classification-Driven Manifold Discretization

Input:
Λ: Parameter vector domain
$U_{(.)}(p^m)$: Mappings from parameter domain Λ to manifolds $\mathcal{M}^m$, $m = 1, \cdots, M$
$\bigcup_{m=1}^{M} \mathcal{S}^m(0) = \bigcup_{m=1}^{M} \bigcup_{i=1}^{N_m} \{S_i^m(0)\}$: Initial set of manifold samples
Initialization:
Initialize total classification error ε as defined in (3.13).
$k = 0$.
repeat
  Pick (possibly randomly) a manifold $\mathcal{M}^m$ and a sample S_i^m from this manifold, $m \in \{1, \cdots, M\}$, $i \in \{1, \cdots, N_m\}$.
  repeat
    Determine the misclassified region $\Theta_i^m(k)$ and the parameter vector $\mu_i^m(k)$.
    Update $S_i^m(k+1)$ as in (3.20).
    $k = k + 1$.
    Determine the misclassified region $\Phi_i^m(k)$ and the parameter vector $\nu_i^m(k)$.
    Update $S_i^m(k+1)$ as in (3.21).
    $k = k + 1$.
  until ε is stabilized
until ε is stabilized
$\bigcup_{m=1}^{M} \mathcal{S}^m = \bigcup_{m=1}^{M} \mathcal{S}^m(k)$.
Output:
$\bigcup_{m=1}^{M} \mathcal{S}^m$: A set of representative samples for each manifold

3.3.3 Sample budget allocation

We have considered so far that the number of samples per manifold is predetermined. We address now the problem of the allocation of samples from a total budget to the different manifolds. This allocation is driven by the properties of the different manifolds. We propose two solutions for budget allocation that can be paired with the CMD algorithm.

A first simple way of determining the sample budget between manifolds is the following. We initialize the sample set of each manifold to be a dense grid on the manifold, and delete samples progressively until the total number of samples meets the budget constraint. We determine the sample to be deleted based on the classification error associated with each sample. Following the previously mentioned arguments, we delete a grid point on one manifold where Θ_i^m is relatively small. When the number of remaining grid points reaches the budget, we optimize the resulting sample sets using CMD for the final adjustment of sample locations. Since the optimization of sample locations takes place after the budget allocation, we name this approach Manifold Discretization with Predefined Allocation (MDPA).

Second, we introduce a joint budget allocation and CMD sample optimization solution where we allow the deletion of a sample from one manifold to create a new sample in another manifold during the iterations. In order to elaborate on the transfer of samples between manifolds, we turn back to the geometric interpretation of our problem formulation. In a configuration with multiple manifolds, our definition of the classification error is the total volume of the regions lying between the piecewise planar boundary surface $\mathcal{A}$ formed by the sample sets and the true boundary surface $\mathcal{B}$. As a result, the classification accuracy of such a setting is directly dependent on the approximability of the boundary surface $\mathcal{B}$ by a piecewise linear model. A "well-behaved" manifold region corresponding to a well-approximable part of $\mathcal{B}$ is more amenable to be represented by a small set of discrete samples than a manifold region corresponding to a part of $\mathcal{B}$ that is difficult to approximate with a linear model. Therefore, the compensation of the loss of a sample from a well-behaved manifold region is relatively easier. We thus propose the following method for dynamic sample budget allocation and optimization. We start with an equal distribution of samples per manifold that satisfies the overall budget constraint, and begin optimizing them with the CMD algorithm. During iterations, we deduce that the manifold region around a sample has poor representability, whenever the misclassified region Θ_i^m has a large volume compared to the average and iterations 7-14 of Algorithm 2 fail to improve the classification accuracy. In a similar way, we determine regions of good representability around samples where the volume of the corresponding Θ_i^m is relatively small. Consequently, we add a new sample to the poorly representable region, at the cost of deleting a sample from a well-representable region selected among all manifolds. We place the new sample at the point corresponding to the projection of the centroid of Θ_i^m onto the manifold. We further adapt the discretization to the new configuration in the regions where sample deletion and creation have taken place, and apply iterations 7-14 of CMD to all neighboring samples on the manifolds of the deleted sample and the created sample. Note however that the sample transfer is not guaranteed to reduce the classification error. In order to ensure the improvement of the classification accuracy, we accept the update only if it reduces the error. We call this approach Dynamic Manifold Discretization (DMD).

3.4 Experimental Results

3.4.1 Setup

We now present experimental results demonstrating the performances of the manifold discretization algorithms discussed in Sections 3.2 and 3.3. All experiments are conducted on two kinds of transformation manifolds, namely the transformation manifold of a 2-D pattern, and the observation

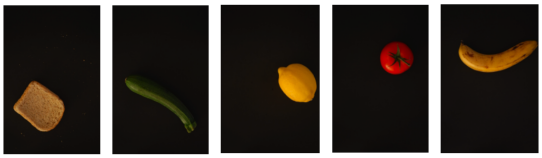


Figure 3.3: Example images from database

Figure 3.4: Pattern transformation manifold images

Figure 3.5: Example objects from airplane class

Figure 3.6: Object observation manifold images

manifold of a synthetical 3-D object.

In the first experimentation setup, we construct pattern transformation manifolds generated by the rotations and translations of 2-D visual patterns. Given a reference image p, we define its transformation manifold $\mathcal{M}(p)$ as

$$\mathcal{M}(p) = \{U_\lambda(p) : \lambda = (\theta, t_x, t_y) \in \Lambda\}, \tag{3.22}$$

where θ is the angle of rotation, and t_x and t_y are horizontal and vertical translation parameters. In the experiment, we use a database of top-view images of 5 different objects, where each object has 500 images captured under different orientations and positions. An example image for each object is given in Figure 3.3. Note that due to the positioning of the camera and the limitations on object positions, the 2-D pattern transformation model in (3.22) constitutes an approximate model for the observations. For each object we build the transformation manifold of a fixed representative pattern picked among the database images. The image set of each object is grouped randomly into 300 training and 200 test images. The categorization of the database into training and test sets is changed randomly in each repetition of the experiment. All images are converted to greyscale, downsampled to a resolution of 50×60 pixels, and background pixels are assigned zero intensity by simple thresholding. Manifold points are generated by rotating and translating the representative pattern (cropped previously near the boundary) over a 50×60 pixel zero background within the parameter range $\theta \in [-\pi, \pi)$; $t_x \in [-7, 7]$; $t_y \in [-12, 12]$. All images and generated manifold points are normalized to have unit norm. Some illustrative images from the transformation manifold of one of the objects are shown in Figure 3.4.

In the second experimentation setup, we consider the object observation manifold model defined in (2.3), which is generated by the observations of a synthetical 3-D object model p from different viewpoints. We use the Princeton Shape Benchmark database of 3-D objects [71], and conduct our experiments on 8 different classes of objects (car, airplane, ship, tank, human, animal, table, bottle) with several (4-30) objects belonging to each class. Some example objects belonging to the airplane class are shown in Figure 3.5. For each class we choose a representative object, and

generate the observation manifold of the representative object in the parameter range $\psi_x, \psi_y, \psi_z \in [-\pi/4, \pi/4]$. The representative object of each class is changed randomly in different repetitions of the experiment. All rendered images are converted to greyscale, downsampled to the resolution of 50×50 pixels and normalized to unit norm. The training and test sets for each manifold consist of 500 random observations of the objects of the same class within the same parameter range. Some images from an object observation manifold are displayed in Figure 3.6.

In both experimental setups, we use training images only for the computation of the centroids of space regions. We compute the centroid of a region of $\mathbb{R}^n$ experimentally by taking random training images, checking if they are in the inquired region, and then computing the arithmetic average of inliers when a sufficient number of them are accumulated as suggested in [72]. Once the centroids are computed, we estimate their projections onto the manifold with the aid of a dense grid on the manifold. We first locate the projection coarsely by finding the grid point that has the smallest distance to the centroid, and then refine the location of the projection by minimizing its distance to the centroid using gradient descent tools.

3.4.2 Results on registration accuracy

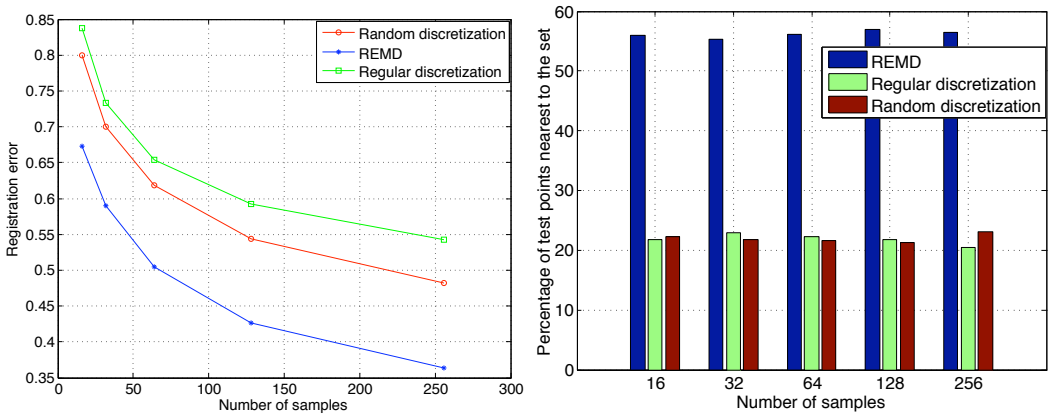
Here we test the REMD algorithm on pattern transformation and object observation manifolds. In both experimental setups, we initialize the algorithm with a randomly selected sample set. We compare the sample set determined by the REMD algorithm to the initial random sample set and to a sample set defined by a regular grid over the parameter domain. The performance evaluation criterion is the accuracy of the discretization in manifold distance estimation. For each discretization, the distances of test points to the sample set are computed and the average registration error is calculated. The registration error is taken as the ℓ^2-distance between the exact projection of the test point onto the manifold and the manifold sample with smallest ℓ^2-distance to the test point.

In the setup with pattern transformation manifolds, we build and sample the transformation manifold of each object individually. For each object the experiment is repeated 10 times with different random initializations. The results are averaged over all realizations and all objects. In Figure 3.7(a), average registration errors obtained with the REMD output, random, and regular sample sets are plotted for several numbers of samples. Then, we compute the distance of each test point to all manifold samples obtained with these three discretization approaches. For each test point we determine the sample among all three sets that has the smallest distance to the test point. In Figure 3.7(b) we report the percentage of test points that have their closest manifold sample within the REMD output, random, and regular sample sets.

These experiments intend to measure the capability of the discretization to provide an accurate approximation of the projection onto the manifold. As shown in the figures, the discretization obtained by the REMD method yields the least registration error when compared to the random discretization and the regular discretization in the parameter domain. In addition, for the majority of the test points the most accurate approximation of the projection lies within the REMD algorithm output sample set.

The experiments on object observation manifolds are conducted similarly. We construct and sample the observation manifold of each object individually. Experiments are repeated 5 times for each class with different random initializations. The results are again averaged over all realizations

and all objects. Figure 3.8(a) shows the registration errors obtained with the samplings and Figure 3.8(b) shows the percentage of test images with the best projection approximations within the REMD output, random, and the regular sample sets. The results are in accordance with the results obtained on pattern transformation manifolds. Note that, although the intrinsic dimensions of manifolds are the same in the two experimental setups, the typical number of samples required for accurately representing the pattern transformation manifolds and the object observation manifolds is quite different in these experiments. This is due to the differences in the type and range of the geometric transformation parameters that generate the manifolds. The fact that the object observation manifolds are defined over a relatively small parameter domain makes it possible to represent them with fewer samples compared to the pattern transformation manifolds for similar performance.



(a) Registration errors given by the samplings

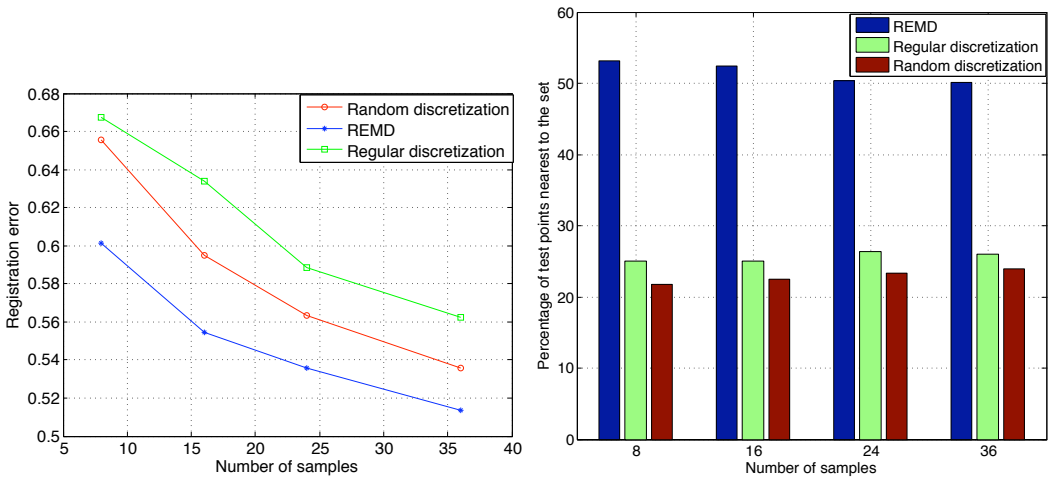
(b) Distribution of test points with respect to projection accuracy

Figure 3.7: Sampling results obtained on pattern transformation manifolds

3.4.3 Results on transformation-invariant classification

In this part, we evaluate the performances of the discretization approaches in Section 3.3 in transformation-invariant classification. In all of the pattern transformation manifold experiments, the image set of each object in the database is regarded as a different signal class. Similarly, in the synthetical objects database, the rendered images of each class of objects are considered to belong to a separate signal class. Only training images are available to the sampling algorithms, and the classification performance is measured on test images. Once sample sets are obtained, the class label of a test image is estimated as the class label of the manifold sample with smallest distance to it. In all of the following figures, the correct classification rates of test images are plotted in percentage with respect to the number of samples per manifold. All experiments are repeated 10 times with different random algorithm initializations and averaged.

First, we compare the REMD algorithm, the random discretization and the regular discretiza-



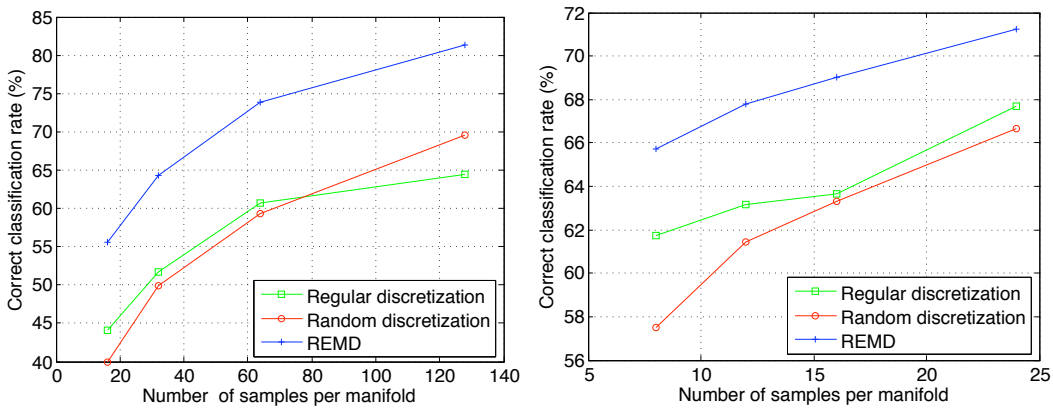
(a) Registration errors given by the samplings

(b) Distribution of test points with respect to projection accuracy

Figure 3.8: Sampling results obtained on object observation manifolds

tion in the parameter domain with respect to their classification performances. The experimental setting is the same as that of Section 3.4.2; i.e., the transformation manifold of each class is sampled individually with the REMD algorithm, random discretization, and regular discretization, where an equal number of samples are selected on each manifold. The results obtained on pattern transformation manifolds and object observation manifolds are plotted respectively in Figures 3.9(a) and 3.9(b). The plots indicate that in both setups, the REMD output sample set has higher classification performance compared to the random and regular discretizations. This is in agreement with the results of the registration experiments of Section 3.4.2, confirming the dependency of the transformation-invariant classification performance on the accuracy of manifold distance estimation. When the two plots in Figures 3.9(a) and 3.9(b) are compared, it is seen that the classification rate improvement introduced by REMD or by increasing the number of samples is higher in pattern transformation manifolds. This can be explained by the difference between the two setups. In object observation manifold experiments there are several object models belonging to the same class. Therefore, space points have a relatively large deviation from the manifold of the representative object. This deviation is smaller in pattern transformation manifolds as the space points of a specific class are the images of the same object.

Then we search the efficiency of jointly optimizing all manifold samples in comparison with sampling each manifold individually. For this purpose, we first select an equal number of samples from each manifold independently with the REMD algorithm as in the previous experiment. Then we apply an additional stage of joint optimization, where we optimize the samples from all classes together with two alternative approaches. In the first approach, we optimize the output sample set of the REMD algorithm further with the CMD algorithm (marked simply as CMD in the plots). In the second one we again begin with the REMD output sample set, but then perform the joint



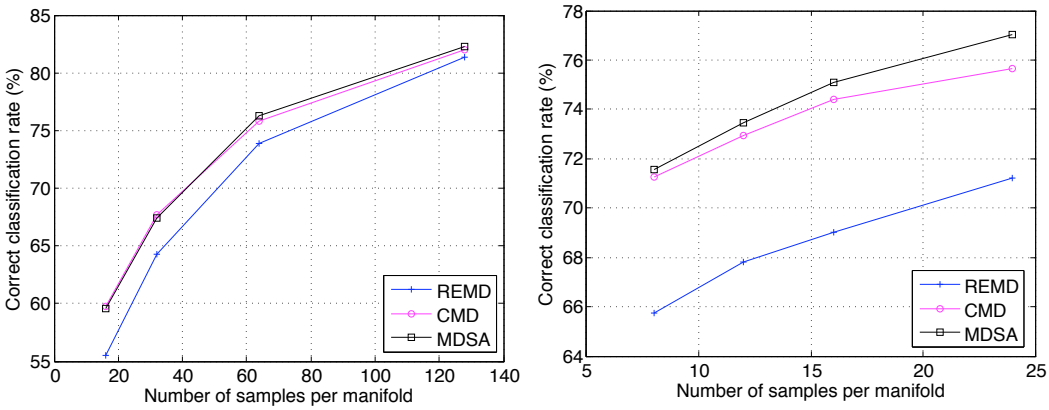
(a) Pattern transformation manifolds

(b) Object observation manifolds

Figure 3.9: Classification results obtained by sampling class representative manifolds individually

optimization of manifold samples using a simulated annealing algorithm where we define the cost function as the classification error ε in (3.13). We name this approach Manifold Discretization with Simulated Annealing (MDSA). The simulated annealing algorithm is based on seeking the global optimum by trying random search directions, whereas the CMD algorithm has a more restricted search domain. Therefore, the results with MDSA are provided as a benchmark for the evaluation of the efficiency of CMD. The correct classification rates are plotted in Figure 3.10(a) for pattern transformation manifolds and in Figure 3.10(b) for object observation manifolds. The results show that the joint optimization of manifold samples after the individual sampling stage brings a significant improvement on the classification rate. This is consistent with the expectation that the relative characteristics of manifolds, i.e., their structures with respect to each other, should also be taken into account as well as their individual characteristics in classification. Moreover, the performances of the sample sets obtained by CMD and MDSA are close to each other. This shows that CMD is an effective constructive algorithm. The slight superiority of MDSA to CMD is justifiable in the sense that the CMD algorithm performs a one-dimensional search in the parameter domain at each update step, whereas in simulated annealing the search space is full dimensional in the parameter domain.

Finally, in a third experiment we examine the effect of the uneven distribution of the total sample budget to different manifolds. We compare the performances of the CMD algorithm with equal budget distribution to different manifolds, DMD, and MDPA, which have been discussed in Section 3.3.3. In order to test CMD, we first select an equal number of samples from each manifold independently with the REMD algorithm, and then optimize the output of REMD in a further stage with CMD as in the previous experiment. We apply the same procedure for DMD as well; it is initialized with the output of REMD with equally distributed samples, where the movement of samples between different manifolds is allowed afterwards to optimize the distribution of the sample budget. The correct classification rates obtained with the three sampling approaches are plotted with respect to the average number of samples selected per manifold in Figures 3.11(a) and 3.11(b),



(a) Pattern transformation manifolds

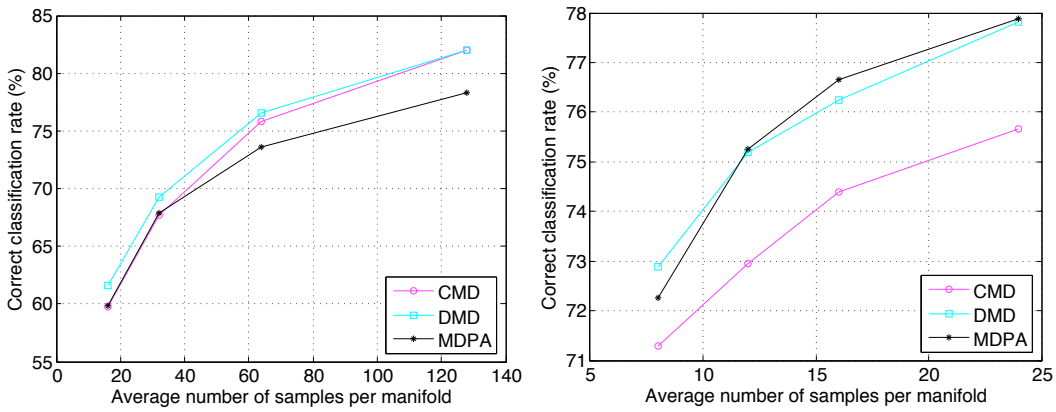
(b) Object observation manifolds

Figure 3.10: Effect of the joint optimization of samples on classification accuracy

respectively for pattern transformation and object observation manifolds. The results suggest that an uneven distribution of the sample budget to different manifolds according to their geometric properties may improve the classification accuracy when compared to the equal distribution of samples. It is seen that the performances of DMD and MDPA are close to each other in the object observation manifolds experiment. However, in the pattern transformation manifolds experiment, the number of available training images per manifold sample is much smaller, which has a negative influence on the efficiency of budget distribution in the first stage of progressive sample deletion in MDPA.

3.4.4 Discussion of results

Here we interpret our experimental results from the perspective of the trade-off between computational complexity and performance. The main motivation behind this work is the difficulty of the exact computation of the manifold distance. The state-of-the-art methods accomplishing manifold distance computation are considerably demanding. For instance, the algorithm proposed in [18] involves a complexity of $O(K\, n_1\, n_2)$, where K is the number of atoms used in the decomposition of the reference pattern and $n_1 \times n_2$ is the image resolution, while a similar algorithm complexity is reported in [2]. On the other hand, in order to estimate the manifold distance we propose the utilization of a suitable manifold grid that is to be determined offline. Once the grid is obtained, the manifold distance estimation is simplified merely to the computation of the norms of the difference vectors between the query signal and the samples, which clearly reduces the cost of distance estimation significantly. Meanwhile, it is not easy to draw a general conclusion in the comparison of the accuracies of manifold distance computation algorithms and our grid approach. This is highly dependent on the algorithms under comparison. For instance, the algorithm in [18] is guaranteed to find the global solution in the projection of query images onto pattern transformation manifolds, resulting in a perfectly accurate distance computation for the studied transformation model. However, methods such as [4], [2], [3] do not have such optimal performance guarantees. In the



(a) Pattern transformation manifolds

(b) Object observation manifolds

Figure 3.11: Effect of the uneven distribution of sample budget on classification accuracy

approximation of the manifold distance with a grid, the registration accuracy is clearly dependent on the number of samples, and the registration error asymptotically approaches zero as the number of samples increases. We also note that, in presence of large geometric transformations the grid approach may have an advantage over algorithms based on tangent distance, which are susceptible to local minima. Considering these, we conclude that sample-based approaches achieve a compromise between accuracy and computational effort.

Now, let us turn to the complexity of the discretization process. We begin with the REMD algorithm. In each iteration of REMD, first the regions R_i are computed and then the samples S_i are optimized. In the first stage, the cells R_i are numerically determined by identifying the manifold sample closest to each training image. Therefore, the complexity of computing the R_i's is $O(NTn)$, where T is the number of training images, N is the number of manifold samples and n is the data dimension. Next, in the second stage, the samples S_i are computed by projecting the centroids of the regions R_i onto the manifold. The complexity of computing the centroids of all regions is $O(Tn)$ since this requires the summation of a total of T training images. Then, in our implementation, we simply computed the projections of centroids onto the manifold with the aid of a dense grid. If the sample locations do not change much, one can also employ a descent-type optimization for this stage. Note however that centroid projection is in fact an image registration problem, and a variety of solutions may be applicable depending on the geometric transformation model (e.g., [73], [65]). Let g denote the complexity parameter of this registration stage (for instance, g may refer to the grid density, or the dimension d of the manifold if gradient descent is used). Then the complexity of centroid projection is $O(Ngn)$ as this is repeated for all N samples, which gives the complexity of the second stage as $O(Tn + Ngn)$. Summing up the complexities of the first and second stages, we obtain the overall complexity of the REMD algorithm as $O(NTn + Ngn)$. The runtime of a non-optimized MATLAB implementation of the algorithm is around a few minutes for $N = 16$ in the experiment reported in Figure 3.7.

Next, we derive the complexity of the CMD algorithm. First, in the main loop of the algorithm,

the determination of the misclassified regions $\Theta_i^m(k)$ and $\Phi_i^m(k)$ requires the comparison of the distances between all training images and all samples. The complexity of determining these regions is therefore $O(T_J\, N_J\, n)$, where T_J and N_J denote respectively the total number of training images and samples from all classes. The complexity of computing the centroids $\theta_i^m(k)$ and $\phi_i^m(k)$ of these regions is $O(T_J\, n)$. Next, consider the optimization of one sample S_i^m. As in the analysis of the REMD algorithm, the projections of $\theta_i^m(k)$ and $\phi_i^m(k)$ onto the manifold can be computed with a complexity of $O(g\, n)$. Then, the update steps in (3.20) and (3.21) require the minimization of the classification error ε, which is based on the inspection of the regions $\Theta_i^m(k)$ and $\Phi_i^m(k)$. Therefore, these updates can be achieved with a complexity of at most $O(T_J\, N_J\, n)$. This gives the complexity of the optimization of a single sample as $O(g\, n + T_J\, N_J\, n)$. Multiplying this with the total number of samples N_J and summing up the complexity of all stages, the overall complexity of the CMD algorithm is obtained as $O(N_J\, g\, n + T_J\, N_J^2\, n)$.

Finally, we note that the classification performance of CMD, which has a refined search space, is fairly close to that of MDSA where the correct classification rate is optimized by simulated annealing. Comparing these two approaches with respect to their convergence rates, we have seen that CMD terminates in a much less number of iterations, which is a result of its capability of assessing the proper search directions. Yet, the overall running time of the CMD algorithm depends on the computational time required by the projection of space points onto manifolds. The speed of the required registration block depends on the type of transformations involved. Efficient solutions exist for certain geometric transformations. For instance, the phase correlation method [73] is a well-known and fast technique that recovers image translations. We remark also that image registration is an active research field and recent works such as [74] are promising for the generalization of such techniques to handle a wider range of geometric transformations.

3.5 Conclusion

In this chapter, we have studied the sampling of signal manifolds with known parameterization. We have presented a method for the discretization of a single manifold such that the sample set yields a good registration accuracy. Then we have generalized the problem to the discretization of multiple signal manifolds representing different classes of signals; we have proposed a method for the joint optimization of all manifold samples in order to preserve high classification performance. We have also discussed possible ways of optimizing the distribution of a fixed sample budget to different class representative manifolds in order to improve the classification accuracy. We have tested the proposed sampling approaches on pattern transformation manifolds and object observation manifolds. Experimental results show that the registration and classification accuracy of the distance-based sampling is considerably higher than the ones obtained with random and regular samplings. Moreover, the consideration of the relative structures of different manifolds in the discretization improves the classification performance significantly when compared to the independent discretization of each manifold. We have also observed that distributing the total sample budget unequally to the manifolds by taking their different characteristics into account also brings an improvement. The study presented in this chapter proposes a grid-based solution for maintaining speed and accuracy at the same time in tasks related to the registration, modeling and classification of data sets.

Chapter 4

Learning Pattern Transformation Manifolds

4.1 Manifold Learning with Data Priors

In the previous chapter, we have proposed a solution for fast computation of the manifold distance based on the discretization of manifolds, where the manifolds are sampled to yield a good accuracy in the registration and transformation-invariant classification of data sets. While we have investigated how one can sample effectively a given manifold model, an important problem still needs to be resolved: how to build suitable manifolds that are good representatives of the data at hand. In this chapter, we seek a solution for this problem for the particular class of pattern transformation manifolds.

Given a set of images that are geometrically transformed observations of a visual signal, we address the problem of constructing a pattern transformation manifold (PTM) that well represents the image set. We assume that the type of transformations that generate the input images, i.e., the geometric transformation model, is known. However, we do not assume any prior alignment of the input images; i.e., the individual transformation parameters corresponding to the images are to be computed. Under these assumptions, our manifold computing problem is formulated as the construction of a representative pattern, together with the estimation of the transformation parameters approximating the input images. We consider a PTM model that is generated by smooth geometric transformations. We propose to build the representative pattern as a linear combination of parametric atoms selected from an analytic dictionary manifold (as defined in (2.10)). We study the PTM building problem in two parts, where we respectively address approximation and classification applications.

In the data approximation part, we aim at obtaining an accurate transformation-invariant approximation of input images with the learned manifold. We iteratively construct a representative pattern by successive addition of atoms such that the total squared distance between the input images and the transformation manifold is minimized. The selection of an atom is then formulated as an optimization problem with respect to the parameters and the coefficient of the atom. We propose a two-stage solution for the atom selection, where we first estimate the parameters of a good atom and then improve this solution. In the first stage, we derive an approximation of

the objective function (total squared distance) in a DC (Difference-of-Convex) form; i.e., in the form of a difference of two convex functions. We describe a procedure for computing this DC decomposition when a DC form of the geometrically transformed atom is known. The resulting DC approximation is minimized using a DC solver. Then, we refine the solution of the first stage with a gradient descent method where we approximate the manifold distance by the tangent distance in the objective function.

In the second part of our study, we extend this manifold building approach to explore transformation-invariant image classification with PTM models. We consider multiple sets of geometrically transformed observations, where each set consists of a different class of images. We study the problem of constructing multiple PTMs such that each PTM represents one image class, and the images can be accurately classified with respect to their distances to the constructed PTMs. We propose an iterative method that jointly selects atoms for the representative patterns of all classes. We define an objective function that is a weighted combination of a classification and a data approximation error term. Then, we select atoms by minimizing a two-stage approximation of the objective function as in the first part. We present experimental results showing that the approaches proposed for single and multiple manifold computation perform well in transformation-invariant approximation and classification applications in comparison with baseline methods.

Our study is linked to two main topics; manifold learning and sparse signal representations. First, our PTM building approach can be seen as a special instance of manifold learning with prior information on the data model. However, as discussed in Chapter 2, it differs from classical manifold learning methods since it uses the information of the geometric transformation model in learning a parametric and analytic manifold that fits the data. Since the manifold is constructed in a parametric form, the mapping between the parameter domain and the high-dimensional signal space is perfectly known. Thus, one can generate new samples on the manifold and compute the parametrizations of initially unavailable data simply by finding their projections on the manifold. This also permits the estimation of the distance between a test image and the computed manifolds. It is then possible to assign class labels to test images in a transformation-invariant way by comparing their distances to the computed class-representative manifolds. Finally, as demonstrated by some of our experiments, the incorporation of the model knowledge into the manifold learning procedure brings important advantages such as robustness to data noise and sparse sampling of data, in comparison with generic methods based on local linearity assumptions.

The method proposed in [75] is related to our work in the sense that it computes a simultaneous alignment of a set of images that have undergone transformations, where the application of the method to classification problems is also demonstrated. However, their technique is essentially different from ours as it is based on the idea of "congealing" via the minimization of entropy in the corresponding pixels of aligned images. Next, our method uses the idea of learning by fitting a parametric model to the data. It is possible to find several other examples of this kind of approach in the literature. For example, the article [76] is a survey on locally weighted learning, where regression methods for computing linear and nonlinear parametric models are discussed. Efficient computation of locally weighted polynomial regression is the focus of [77]. Meanwhile, the method in [78] applies locally weighted regression techniques to the appearance-based pose estimation problem. Lastly, the study in [79] proposes a method to learn dictionaries in a parametric form that yield domain-invariant sparse signal representations.

Next, we remark the following about the relation between this work and the field of sparse

signal approximations. Since we construct representative patterns in a greedy way, our method bears some resemblance to sparse approximation algorithms such as Matching Pursuit (MP) [19] or Simultaneous Orthogonal Matching Pursuit (SOMP) [80]. There are also common points between our method and the Supervised Atom Selection (SAS) algorithm proposed in [81], which is a classification-driven sparse approximation method. SAS selects a subset of atoms from a discrete dictionary by minimizing a cost function involving a class separability term and an approximation term. However, the main contributions of this work in comparison with such algorithms lie in the following. First, we achieve a transformation-invariant approximation of signals due to the transformation manifold model. Furthermore, we employ an optimization procedure for computing the atom parameters that provide an accurate approximation (or classification) of signals. This corresponds to learning atoms from a dictionary manifold, whereas methods such as MP and SOMP pick atoms from a predefined discrete dictionary. This also suggests that it is possible to find connections between our work and transformation-invariant dictionary learning, where a sparse representation of signals is sought not only in terms of the original atoms but also in their geometrically transformed versions. So far, transformation-invariance in sparse approximations has been mostly studied for shift-invariance as in [82] and [83], and for scale-invariance as in [84], [85]. The work presented in [86] also achieves shift-invariance in the sparse decomposition via a continuous basis pursuit. Our PTM learning method involves the formation of atoms that ensure invariance to a relatively wide range of geometric transformations in comparison with the above works. Our study may thus provide some insight into transformation-invariance in sparse approximations as well.

This chapter is organized as follows. In Section 4.2, we discuss the manifold computation problem for transformation-invariant approximation of image signals. Then, in Section 4.3, we present an extension of the proposed scheme for transformation-invariant classification. We discuss the complexity of the proposed methods in Section 4.4. Finally, we conclude in Section 4.5.

4.2 Computation of PTMs for Signal Approximation

4.2.1 Problem formulation

The PTM computation problem can be briefly explained as follows. Given a set of observations $\{u_i\}$, we would like to compute a pattern p such that its transformation manifold $\mathcal{M}(p)$ fits the observations $\{u_i\}$. Therefore, we look for a pattern p such that the total distance between $\mathcal{M}(p)$ and $\{u_i\}$ is minimized, which is illustrated in Figure 4.1. Now we define the problem formally.

Let $p \in L^2(\mathbb{R}^2)$ be a visual pattern, $\Lambda \subset \mathbb{R}^d$ be a closed parameter domain, and $\lambda \in \Lambda$ be a parameter vector. We define $A_\lambda(p) \in L^2(\mathbb{R}^2)$ as the pattern that is generated by applying the geometric transformation specified by λ to p. The relation between the two patterns is expressed as $A_\lambda(p)(X) = p(X')$, where $X = [x\, y]^T$, $X' = [x'\, y']^T$, and the coordinate variables X, X' are related as $X' = a(\lambda, X)$. We assume that a is a smooth (C^∞) function. Also, defining $a_\lambda(X) := a(\lambda, X)$ for a fixed $\lambda \in \Lambda$, we assume that $a_\lambda : \mathbb{R}^2 \to \mathbb{R}^2$ is a bijection. Then, we define the transformation manifold of p as

$$\mathcal{M}(p) = \{U_\lambda(p) : \lambda \in \Lambda\} \subset \mathbb{R}^n, \tag{4.1}$$

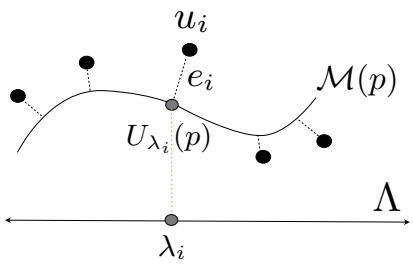


Figure 4.1: The set $\{u_i\}$ of geometrically transformed observations is approximated with the transformation manifold $\mathcal{M}(p)$ of a representative pattern p.

where $U_\lambda(p) \in \mathbb{R}^n$ is an n-dimensional discretization of $A_\lambda(p)$.[1]

Let $\mathcal{U} = \{u_i\}_{i=1}^N \subset \mathbb{R}^n$ be a set of observations of a geometrically transformed visual signal. We would like to describe these observations as $u_i = U_{\lambda_i}(p) + e_i$ by the transformations $U_{\lambda_i}(p)$ of a common representative pattern p, where the term e_i indicates the deviation of u_i from $\mathcal{M}(p)$. In the selection of p, the objective is to approximate the images in $\mathcal{U}$ accurately. We represent the approximation accuracy in terms of the distance of the input images to $\mathcal{M}(p)$. We formalize this problem as follows.

Problem 1. *Given images $\mathcal{U} = \{u_i\}_{i=1}^N$, compute a pattern $p \in L^2(\mathbb{R}^2)$ and a set of transformation parameter vectors $\{\lambda_i\}_{i=1}^N \subset \Lambda$, by minimizing*

$$E = \sum_{i=1}^{N} \|u_i - U_{\lambda_i}(p)\|^2. \tag{4.2}$$

The error E corresponds to the total squared distance of the input images to $\mathcal{M}(p)$. In order to solve Problem 1, we propose to construct p as a sparse linear combination of some parametric atoms from a dictionary manifold

$$\mathcal{D} = \{B_\gamma(\phi) : \gamma \in \Gamma\} \subset L^2(\mathbb{R}^2). \tag{4.3}$$

Here, each atom $B_\gamma(\phi) \in L^2(\mathbb{R}^2)$ is derived from the analytic mother function $\phi \in L^2(\mathbb{R}^2)$ through a geometric transformation specified by a parameter vector γ. An atom is thus given by $B_\gamma(\phi)(X) = \phi(X')$, where $X' = b(\gamma, X)$. We assume that b is a smooth function, and that $b_\gamma(X) := b(\gamma, X)$, $b_\gamma : \mathbb{R}^2 \to \mathbb{R}^2$ is a bijection for any fixed $\gamma \in \Gamma$. The parameter domain Γ is assumed to be a closed and convex subset of $\mathbb{R}^s$ for some s, where s is the number of transformation parameters generating $\mathcal{D}$. Hence, $\mathcal{D}$ is an s-manifold. Let us write $\phi_\gamma = B_\gamma(\phi)$ for simplicity. We would like to obtain the representative pattern in the form

$$p = \sum_{j=1}^{K} c_j \, \phi_{\gamma_j}$$

[1]When sampling $A_\lambda(p)$ to get $U_\lambda(p)$, we fix a rectangular window on $\mathbb{R}^2$, and a regular sampling grid once and for all. Note that defining the pattern transformations in the continuous space $L^2(\mathbb{R}^2)$ instead of $\mathbb{R}^n$, together with constructing p with parametric atoms in $L^2(\mathbb{R}^2)$, saves us from resampling and interpolation ambiguities.

as a combination of K atoms $\{\phi_{\gamma_j}\}$ with coefficients $\{c_j\}$. Under these assumptions, we reformulate the previous problem as follows.

Problem 2. *Given images $\mathcal{U} = \{u_i\}_{i=1}^N$, an analytic mother function ϕ, and a sparsity constraint K; compute a set of atom parameter vectors $\{\gamma_j\}_{j=1}^K \subset \Gamma$, a set of coefficients $\{c_j\}_{j=1}^K \subset \mathbb{R}$, and a set of transformation parameter vectors $\{\lambda_i\}_{i=1}^N \subset \Lambda$, by minimizing*

$$E = \sum_{i=1}^{N} \|u_i - U_{\lambda_i}\big(\sum_{j=1}^{K} c_j\, \phi_{\gamma_j}\big)\|^2. \tag{4.4}$$

Note that the construction of p with smooth atoms assures the smoothness of the resulting transformation manifold. A manifold point $U_\lambda(p) \in \mathbb{R}^n$ is given by the discretization of the function

$$A_\lambda(p)(X) = p(a_\lambda(X)) = \sum_{j=1}^{K} c_j\, \phi_{\gamma_j}(a_\lambda(X)) = \sum_{j=1}^{K} c_j\, \phi(b_{\gamma_j} \circ a_\lambda(X)) \tag{4.5}$$

where the notation $\circ$ stands for function composition. Here $a_\lambda(X)$ is a smooth function of λ; and b and ϕ are smooth functions, too. Therefore, $A_\lambda(p)(X)$ is a smooth function of λ. Then, each component $U_\lambda(p)(l)$ of $U_\lambda(p)$ is a smooth function of λ, for $l = 1, \ldots, n$.

4.2.2 PTM building algorithm

We now describe an algorithm for the solution of Problem 2. Due to the complicated dependence of E on the atom and projection parameters, it is hard to find an optimal solution for Problem 2. Thus, we propose a greedy method that builds a pattern p iteratively by selecting atoms from $\mathcal{D}$. Each successive version p_j of the pattern p leads to a different manifold $\mathcal{M}(p_j)$, whose form gradually converges to the final solution $\mathcal{M}(p)$. During the optimization of the atom parameters in each iteration, we first locate a good initial solution by minimizing a DC approximation of the objective function using DC programming. We then refine our solution by using a locally linear approximation of the manifold near each input image and minimizing the total tangent distance to the manifold with gradient descent. The reason for our choice of a two-step optimization in atom selection is the following. The DC solver used in our implementation is the cutting plane algorithm, which slows down as the number of vertices increases throughout the iterations. Therefore, in practice, we use the DC programming step for approaching the vicinity of a good solution and we terminate it when it slows down. Then, we continue the minimization of the function with gradient descent. Considering that the DC program is not affected by local minima and gradient descent is susceptible to local minima, using these two methods respectively for the first and second parts is a suitable choice. We start by giving a brief discussion of DC functions [87] that are used in our algorithm.

Definition 7. *A real valued function f defined on a convex set $C \subset \mathbb{R}^s$ is called DC on C if for all $x \in C$, f can be expressed in the form*

$$f(x) = g(x) - h(x) \tag{4.6}$$

where g, h are convex functions on C. The representation (4.6) is said to be a DC decomposition of f.

An important fact about DC functions is the following[2] [87].

Proposition 1. *Every function $f : \mathbb{R}^s \to \mathbb{R}$ whose second partial derivatives are continuous everywhere is DC.*

The global minimum of DC functions can be computed using DC solvers such as the cutting plane algorithm and the branch-and-bound algorithm [89], which is a major reason for the choice of DC programming in this work. There are also some DC optimization methods such as DCA [90] and the concave-convex procedure (CCCP) [91], which have favorable computational complexities and converge to a local minimum. The theoretical guarantee for finding the global minimum with the cutting plane algorithm is lost when the DC program is terminated before exact convergence as in our implementation; however, the overall two-step minimization gives good results in practice.

Equipped with the DC formalism, we can now describe our iterative manifold learning algorithm. As the atom selection procedure requires the computation of the distance between the input images and the PTM, the algorithm initially needs rough estimates of the parameter vectors. Therefore, we first assign a tentative set of parameter vectors $\{\lambda_i\}$ to the images $\{u_i\}$ by projecting $\{u_i\}$ onto some reference transformation manifold $\mathcal{M}(\Psi)$. The pattern Ψ can be possibly chosen as a typical pattern in the input set (an $L^2(\mathbb{R}^2)$-representation of some u_i). Then, the parameter vector assigned to an image is given by $\lambda_i = \arg\min_{\lambda\in\Lambda} \|u_i - U_\lambda(\Psi)\|$. We compute the transformation parameters by first roughly locating the projections with the help of a grid, and then performing a line search near the closest grid point.

Now let us describe the j-th iteration of the algorithm. Let p_{j-1} denote the pattern consisting of $j-1$ atoms (one can set p_0=0). In the j-th iteration we would like to choose an atom $\phi_{\gamma_j} \in \mathcal{D}$ and a coefficient c_j such that the data approximation error

$$E = \sum_{i=1}^{N} \|e_i\|^2 = \sum_{i=1}^{N} d^2(u_i, \mathcal{M}(p_j)) \tag{4.7}$$

is minimized, where $p_j = p_{j-1} + c_j\,\phi_{\gamma_j}$. We remark that the cost function in (4.4) is defined as a function of all atom parameters $\{\gamma_j\}_{j=1}^K$ and coefficients $\{c_j\}_{j=1}^K$, however, the one in (4.7) is considered only as a function of γ_j and c_j. For simplicity, we use the same symbol E for these two functions with an abuse of notation.

Notice that the values of $\{\lambda_i\}$ may change between iterations $j-1$ and j, because the projection points change when the manifold is updated. The alteration of $\{\lambda_i\}$ is illustrated in Figure 4.2. At the beginning of the j-th iteration, the vectors $\{\lambda_i\}$ take the values computed at the end of iteration $j-1$ by projecting $\{u_i\}$ on $\mathcal{M}(p_{j-1})$. Therefore, $d(u_i, \mathcal{M}(p_j)) \neq \|u_i - U_{\lambda_i}(p_j)\|$ in general. In the minimization of E, it is not easy to formulate and compute the exact distance $d(u_i, \mathcal{M}(p_j))$, since it

[2]Proposition 1 is the original statement of Corollary 4.1 in [87], which holds for functions defined on $\mathbb{R}^s$. However, a function defined on a convex subset of $\mathbb{R}^s$ with continuous second partial derivatives is also DC. This can be easily seen by referring to the proof of Corollary 4.1 in [87], which is based on the fact that locally DC functions are DC, and to Hartman's proof [88] that locally DC functions defined on a convex subset of $\mathbb{R}^s$ are DC on the same domain.

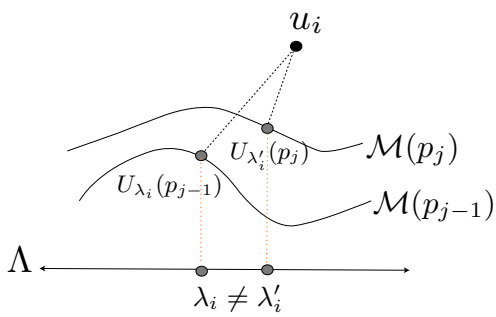


Figure 4.2: The parameter vectors corresponding to the projections of the point u_i on the previous manifold $\mathcal{M}(p_{j-1})$ and the updated manifold $\mathcal{M}(p_j)$ are shown respectively by λ_i and λ_i'.

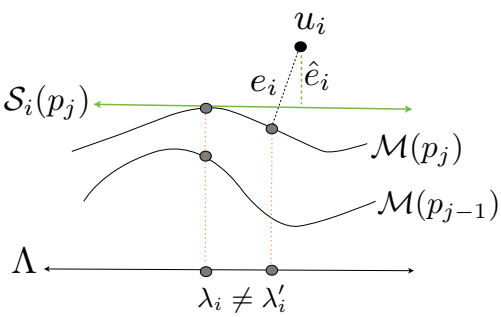


Figure 4.3: $\mathcal{S}_i(p_j)$ is the first order approximation of the manifold $\mathcal{M}(p_j)$ around $U_{\lambda_i}(p_j)$. Here, the difference vector e_i between u_i and its exact projection on $\mathcal{M}(p_j)$ is approximated by the difference vector $\hat{e}_i$ between u_i and its projection on $\mathcal{S}_i(p_j)$.

would require the formulation of λ_i as a function of the optimization variables, which does not have a known closed-form expression. Therefore, we propose to minimize E in two stages. Let $\gamma = \gamma_j$ and $c = c_j$ denote the parameters and the coefficient of the new atom for the ease of notation. In the first stage, we define a coarse approximation

$$\tilde{E} = \sum_{i=1}^{N} \|\tilde{e}_i\|^2 = \sum_{i=1}^{N} \|u_i - U_{\lambda_i}(p_{j-1} + c\,\phi_\gamma)\|^2 = \sum_{i=1}^{N} \|v_i - c\,U_{\lambda_i}(\phi_\gamma)\|^2 \tag{4.8}$$

of E, where $v_i = u_i - U_{\lambda_i}(p_{j-1})$ is a constant with respect to γ and c. Note that the operator $U_\lambda(\cdot)$ is linear, since for two patterns p, r, and a scalar c, we have

$$A_\lambda(c\,p + r)(X) = (c\,p + r)(a_\lambda(X)) = c\,p(a_\lambda(X)) + r(a_\lambda(X)) = cA_\lambda(p)(X) + A_\lambda(r)(X).$$

We have the following proposition.

Proposition 2. *$\tilde{E}$ is a DC function of γ and c. Moreover, if a DC decomposition for the components (pixels) of the transformed atom $U_\lambda(\phi_\gamma)$ is known, a DC decomposition of $\tilde{E}$ is computable.*

The proof of Proposition 2 is given in Appendix A.1. Although finding the DC decomposition of an arbitrary function is an open problem, DC decompositions are available for important function classes [89]. See, for instance, [18] for the derivation of the DC decompositions of several elementary functions, and [89] for operations with known DC decompositions. For the rest of our discussion, we assume that a DC decomposition of the components of $U_{\lambda_i}(\phi_\gamma)$ is computable. We can therefore minimize $\tilde{E}$ using the cutting plane algorithm discussed in [89] and [18]. This provides an initial solution for the atom that is optimized further in the next stage.

In the second stage of our method, we approximate E by another function $\hat{E}$, which is the sum of the squared tangent distances of $\{u_i\}$ to the updated manifold $\mathcal{M}(p_j)$. Let $\mathcal{S}_i(p_j)$ denote the first order approximation of $\mathcal{M}(p_j)$ around $U_{\lambda_i}(p_j)$, where λ_i is still as computed at the end of iteration $j-1$. Then, the distance $d(u_i, \mathcal{S}_i(p_j))$ between u_i and $\mathcal{S}_i(p_j)$ is called the tangent distance [3] and it provides an approximation for $d(u_i, \mathcal{M}(p_j))$ (illustrated in Figure 4.3). Hence, $\hat{E}$ is given by

$$\hat{E} = \sum_{i=1}^{N} \|\hat{e}_i\|^2 = \sum_{i=1}^{N} d^2(u_i, \mathcal{S}_i(p_j)). \tag{4.9}$$

The complete derivations of $\hat{E}$, $\mathcal{S}_i(p_j)$ and the distance to $\mathcal{S}_i(p_j)$ are given in Appendix A.2. We minimize $\hat{E}$ over (γ, c) using a gradient descent algorithm. At the end of this second stage, we finally obtain our solution for the atom parameters γ and the coefficient c.

The new atom is then added to the representative pattern such that $p_j = p_{j-1} + c\,\phi_\gamma$. Since p_j is updated, we recompute the projections of $\{u_i\}$ on the new manifold $\mathcal{M}(p_j)$ and update $\{\lambda_i\}$ such that they correspond to the new projection points. The projections can be recomputed by performing a search in a small region around their previous locations.

We continue the iterative approximation algorithm until the change in E becomes insignificant or a predefined sparsity constraint is reached. We also finalize the algorithm in case an update increases E, which might occur as the atom selection is done by minimizing the approximations of E. The termination of the algorithm is guaranteed as E is forced to be non-increasing throughout the iterations. However, due to the complicated structure of the method that uses several approximations of E, it is hard to provide a theoretical guarantee that the solution p, $\{\lambda_i\}_{i=1}^N$ converges, even if that has been the case in all experiments. We name this method Parameterized Atom Selection (PATS) and summarize it in Algorithm 3. The complexity of the algorithm is discussed in Section 4.4. As a final remark, we discuss the accuracy of the reformulation of the objective function E in (4.2) in several stages of the algorithm. To begin with, the error arising from approximating (4.2) with (4.4) asymptotically approaches 0 as the number of atoms in the sparse approximation is increased, provided that the span of $\mathcal{D}$ is dense in $L^2(\mathbb{R}^2)$. Then, the gradual minimization of (4.4) via minimizing (4.7) also introduces an error, which is a common feature of greedy algorithms. Next, the deviation of $\tilde{E}$ in (4.8) from E in (4.7) mainly depends on the amount of change in the transformation parameters between consecutive iterations. Starting the algorithm with a good initialization of parameters helps to reduce this error. Moreover, the inaccuracy caused by this approximation is partially compensated for in the next stage as $\hat{E}$ accounts for parameter changes. The accuracy of this second approximation essentially depends on the nonlinearity of the manifold; i.e., $\hat{E} = E$ if the manifold is linear. However, even if the manifold has high curvature, the approximation $\hat{E} \approx E$ is accurate if the change in the transformation parameters is small between adjacent iterations, which is often the case especially in late phases of the algorithm.

Algorithm 3 Parameterized Atom Selection (PATS)

1: **Input:**
$\mathcal{U} = \{u_i\}_{i=1}^N$: Set of observations
2: **Initialization:**
3: Determine a tentative set of parameter vectors $\{\lambda_i\}$ by projecting $\{u_i\}$ on the transformation manifold $\mathcal{M}(\Psi)$ of a reference pattern Ψ.
4: $p_0 = 0$.
5: $j = 0$.
6: **repeat**
7: $j = j + 1$.
8: Optimize the parameters γ and the coefficient c of the new atom with DC programming such that the error $\tilde{E}$ in (4.8) is minimized.
9: Further optimize γ and c with gradient descent by minimizing the error $\hat{E}$ in (4.9).
10: Update $p_j = p_{j-1} + c\,\phi_\gamma$.
11: Update parameter vectors $\{\lambda_i\}$ by projecting $\{u_i\}$ onto $\mathcal{M}(p_j)$.
12: **until** the approximation error E converges or increases
13: **Output**:
$p = p_j$: A representative pattern whose transformation manifold $\mathcal{M}(p)$ fits the input data $\mathcal{U}$

4.2.3 Experimental results

We now present experimental results demonstrating the application of PATS in transformation-invariant image approximation. We first describe the experimental setup. We experiment on the PTM model given in (2.2). The transformed image $U_\lambda(p)$ is a discretization of $A_\lambda(p)$, where $A_\lambda(p)(X) = p(X')$ and the coordinate transformation between p and $A_\lambda(p)$ is given by

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} s_x^{-1} & 0 \\ 0 & s_y^{-1} \end{bmatrix} \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x - t_x \\ y - t_y \end{bmatrix}. \tag{4.10}$$

We use the dictionary manifold model given in (2.10), where we take the mother function ϕ as the normalized Gaussian function

$$\phi(X) = \sqrt{\frac{2}{\pi}} e^{-(x^2+y^2)}. \tag{4.11}$$

In Appendix A.3, we describe the computation of the DC decompositions of $U_\lambda(\phi_\gamma)$ and the error $\tilde{E}$ for this setup.

In the first set of experiments, we test the PATS algorithm on two data sets, which consist of handwritten "5" digits and face images. The first data set is generated from the MNIST handwritten digits database [92] by applying random geometric transformations to 30 randomly selected images of the "5" digit. The second data set consists of 35 geometrically transformed face images of a single subject with facial expression variations [93], which is regarded as a source of deviation of the data from the manifold. Both data sets are generated by applying rotations, anisotropic scalings and translations.

In the experiments we measure the data approximation error of the learned pattern, which is

the average squared distance of input images to the computed transformation manifold. In the plots, the data approximation error is normalized with respect to the average squared norm of input images.

In order to evaluate the performance of the PATS method, we compare it with the following baseline approaches.

- MP on average pattern: We determine a representative pattern (average pattern) by picking the untransformed image that is closest to the centroid of all untransformed data set images. Then, we obtain progressive approximations of the average pattern with Matching Pursuit [19].
- SMP on aligned patterns: We obtain a progressive simultaneous approximation of untransformed images with the Simultaneous Matching Pursuit algorithm explained in [94]. SMP selects in each iteration one atom that approximates all images simultaneously, but the coefficient of the atom is different for each image. We construct a pattern gradually by adding the atoms chosen by SMP and weighting them with their average coefficients.
- Locally linear approximation: We compute the locally linear approximation error, which is the average distance between an image and its projection onto the plane passing through its nearest neighbors. We include this error, since typical manifold learning algorithms such as [6] and [7] use a linear approximation of the manifold.

The dictionary used in the first two methods above is a redundant sampling of the dictionary manifold in (2.10). The results obtained on the digit and face images are given respectively in Figures 4.4 and 4.5. Some images from each data set are shown in Figures 4.4(a) and 4.5(a). The patterns built with the proposed method are displayed in Figures 4.4(b) and 4.5(b). It is seen that the common characteristics of the input images are well captured in the learned patterns. The data approximation errors of the compared methods are plotted in Figures 4.4(c) and 4.5(c). The errors of the PTM-based methods are plotted with respect to the number of atoms used in the progressive generation of patterns. The results show that the proposed method provides a better approximation accuracy than the other approaches. The approximation accuracies of MP and SMP are better in the face images experiment compared to the digits experiment. This can be explained by the fact that face images of the same subject have smaller numerical variation with respect to handwritten digit images; therefore, an average pattern in the data set can approximate the others relatively well.[3] One can also observe that the locally linear approximation error is significantly high. The local linearity assumption fails in these experiments because of the sparse sampling of the data (small number of images), whereas PTM-based methods are much less affected by such sampling conditions.

In a second experiment, we study the effect of occlusions and outliers in PTM building. We experiment on the same digits data set as before, with a transformation model consisting of 2-D translations, where only the parameters t_x, t_y in (2.2) are used. The images are randomly occluded with horizontal and vertical stripes as shown in Figure 4.6(a). We generate four different data sets, where the first one consists of 150 images of only the digit "2". We obtain the other data sets

[3] In an evaluation on the aligned and normalized versions of the input images, the average squared distance to the centroid is found as 0.40 for the digit images and 0.01 for the face images.

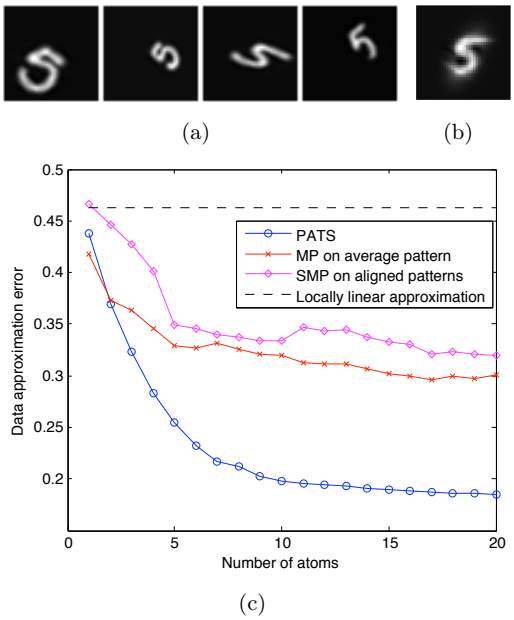


Figure 4.4: Manifold approximation results with handwritten "5" digits. (a) Images from the digits data set. (b) Learned pattern. (c) Approximation error.

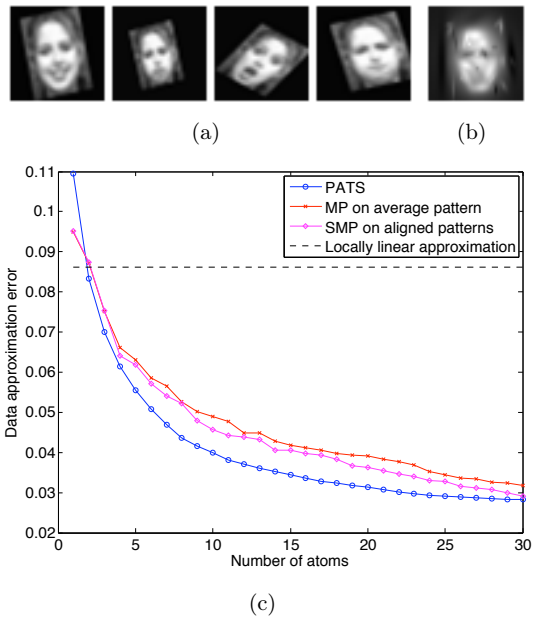


Figure 4.5: Manifold approximation results with face images. (a) Images from the face data set. (b) Learned pattern. (c) Approximation error.

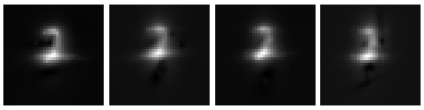
by adding the first data set outliers consisting of a mixture of "3", "5" and "8" digits, where the outlier/inlier ratio is 10%, 20% and 30%. We test the PATS method using a dictionary generated with the inverse multiquadric mother function given by

$$\phi(X) = (1 + x^2 + y^2)^{\mu}, \;\; \mu < 0.$$

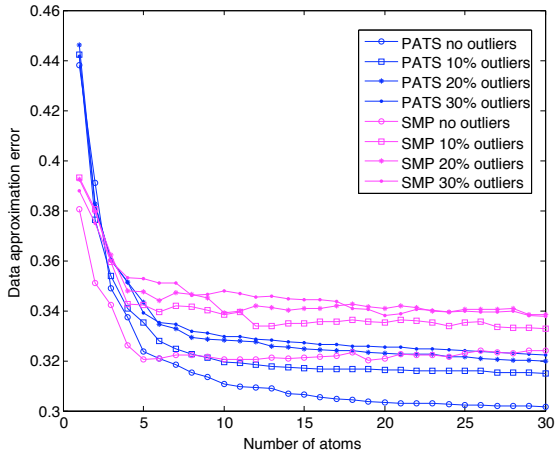
We have set $\mu = -3$ in the experiments. The computation of the DC decomposition for this mother function is explained in Appendix A.3. The patterns learned with all four data sets are shown in Figure 4.6(b), and the errors are plotted in Figure 4.6(c). The errors obtained with SMP on aligned patterns are also given for comparison. It is shown that the proposed method can recover a representative "2" digit in spite of the occlusions. As the ratio of outliers is augmented, the characteristics of the learned pattern gradually diverge from the "2" digit; and the approximation error increases as the average deviation of the data from the "2" manifold is increased.

Then, in a third experiment, we search the effect of some algorithm settings on the performance of PATS. We experiment on a data set from the Extended Yale Face Database B [95] where face images are captured under varying illumination conditions. We create a data set of 90 images by applying geometric transformations consisting of anisotropic scaling to the images of a single subject, where only the parameters s_x, s_y in (2.2) are used. Some sample data set images are shown in Figure 4.7(a). We apply the PATS algorithm in three different settings. In the first setting, the algorithm is used in its normal mode; i.e., in line 3 of Algorithm 3, parameters are

(a)

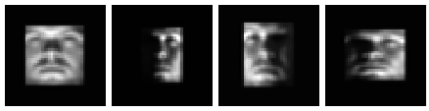


(b)



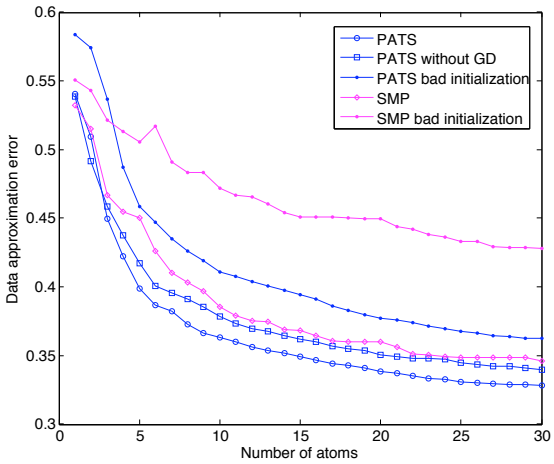
(c)

Figure 4.6: Manifold approximation results with occluded digit images with outliers. (a) Images from the occluded digits data set. (b) Learned patterns, from left to right: 0%, 10%, 20%, 30% outliers. (c) Approximation error.



(a)

(b)



(c)

Figure 4.7: Manifold approximation results with face images with varied illumination conditions. (a) Images from the Yale face data set. (b) Learned patterns, from left to right: Normal setting, without gradient descent, bad initialization. (c) Approximation error.

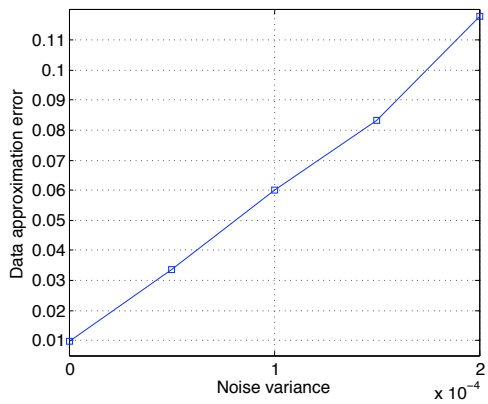


Figure 4.8: Dependence of the approximation error on data noise. The largest noise variance 2×10^{-4} corresponds to an SNR of 9.054 dB.

initialized with respect to the data set image having the smallest distance to the centroid of all images. In the second setting, the initialization is done in the same way; however, line 9 of the algorithm (gradient descent) is omitted. The third setting is the same as the first setting except that the algorithm is started with a bad initialization, where the alignment in line 3 is done with respect to the data set image having the largest distance to the centroid. The patterns learned in all three settings are shown in Figure 4.7(b), and the approximation errors are plotted in Figure 4.7(c). The algorithm does not output clear facial features due to the variation of illumination. The gradient descent step is seen to bring a certain improvement in the performance. The results also show that the algorithm has a sensitivity to initialization. A significant change in the initial values of the transformation parameters causes the algorithm to compute a different solution. In order to provide a comparison, we also plot the results obtained with "SMP on aligned patterns" with default and bad alignments. The fact that the error difference between the two cases is much larger in SMP compared to PATS suggests that PATS can nevertheless compensate for a bad initialization of transformation parameters to some extent.

Finally, in a last experiment we examine the approximation accuracy of the learned manifold with respect to the noise level of the data set. We form a synthetic pattern r that is composed of 10 randomly selected atoms from $\mathcal{D}$. Then, we generate a data set $\mathcal{U}$ of 50 images by applying to r random geometric transformations of the form (2.2). We derive several data sets from $\mathcal{U}$ by corrupting its images with additive Gaussian noise, where each data set has a different noise variance. Then, we run the PATS algorithm on each data set. In Figure 4.8, the data approximation error is plotted with respect to the noise variance. The deviation between $\mathcal{U}$ and $\mathcal{M}(r)$ depends on the noise level, and the ideal approximation error is linearly proportional to the noise variance. Such a linear dependency can be observed in Figure 4.8. However, it is observed that the curve does not pass through the origin, which is due to the suboptimal greedy nature of the algorithm.

4.3 Joint Computation of PTMs for Classification

In this section we consider multiple image sets, where each set consists of geometrically transformed observations of a different visual signal class. We build on the scheme presented in Section 4.2.2 and extend the PATS algorithm for joint manifold computation in classification applications.

4.3.1 Problem formulation

Consider a collection of images $\mathcal{U} = \bigcup_{m=1}^{M} \mathcal{U}^m \subset \mathbb{R}^n$ consisting of M classes, where each subset $\mathcal{U}^m = \{u_i^m\}_{i=1}^{N_m}$ consists of N_m geometrically transformed observations of a visual signal of class m. We would like to represent each set $\mathcal{U}^m$ by a transformation manifold $\mathcal{M}(p^m) \subset \mathbb{R}^n$ that is generated by the geometric transformations of a representative pattern p^m as in (2.4). We would like to build $\{\mathcal{M}^m\}$ such that they provide a good representation of the images in $\mathcal{U}$ and also permit us to classify them accurately by manifold distance computation. Hence, in the construction of the manifolds, we formulate the objective function as a weighted combination of two terms E_a and E_c, which respectively represent approximation and classification errors. The approximation error E_a is given by the sum of the squared distances of images to the manifold of the same class

$$E_a = \sum_{m=1}^{M} \sum_{i=1}^{N_m} \|e_i^m\|^2 = \sum_{m=1}^{M} \sum_{i=1}^{N_m} d^2(u_i^m, \mathcal{M}^m). \tag{4.12}$$

We assume that an image is assigned the class label of the manifold with smallest distance to it. We define a misclassification indicator function I such that for $u_i^m \in \mathcal{U}^m$

$$I(u_i^m) = \begin{cases} 0, & \text{if } d(u_i^m, \mathcal{M}^m) < \min_{r \neq m} d\big(u_i^m, \mathcal{M}^r\big) \\ 1, & \text{otherwise.} \end{cases} \tag{4.13}$$

Then, the classification error E_c is the total number of misclassified data points.

$$E_c = \sum_{m=1}^{M} \sum_{i=1}^{N_m} I(u_i^m) \tag{4.14}$$

We would like to compute $\{\mathcal{M}^m\}_{m=1}^{M}$ such that the weighted error

$$E = E_a + \alpha\, E_c \tag{4.15}$$

is minimized, where $\alpha > 0$ is a coefficient adjusting the weight between the approximation and classification terms. We formulate a generic PTM learning problem as follows.

Problem 3. *Given image sets $\{\mathcal{U}^m\}$, compute patterns $\{p^m\} \subset L^2(\mathbb{R}^2)$ and transformation parameters $\{\lambda_i^m\} \subset \Lambda$, $m = 1, \ldots, M$ and $i = 1, \ldots, N_m$, by minimizing*

$$E = \sum_{m=1}^{M} \sum_{i=1}^{N_m} \left(\|u_i^m - U_{\lambda_i^m}(p^m)\|^2 + \alpha\, I(u_i^m) \right). \tag{4.16}$$

Our solution is based on constructing each p^m with atoms from the dictionary manifold $\mathcal{D}$ defined in (4.3). We reformulate Problem 3 under these assumptions.

Problem 4. *Given image sets $\{\mathcal{U}^m\}$, a mother function ϕ and sparsity constraints $\{K_m\}$; compute a set of atom parameters $\{\gamma_j^m\} \subset \Gamma$, coefficients $\{c_j^m\} \subset \mathbb{R}$, and transformation parameters $\{\lambda_i^m\} \subset \Lambda$ for $m = 1, \ldots, M$, $j = 1, \ldots, K_m$ and $i = 1, \ldots, N_m$, by minimizing*

$$E = \sum_{m=1}^{M} \sum_{i=1}^{N_m} \big(\|u_i^m - U_{\lambda_i^m}\big(\sum_{j=1}^{K_m} c_j^m \phi_{\gamma_j^m}\big)\|^2 + \alpha\, I(u_i^m)\big). \tag{4.17}$$

4.3.2 Classification-driven PTM learning

Problem 4 is similar to Problem 2, except that it also involves a classification error term that has a quite complex dependence on the optimization variables. Therefore, it is hard to solve optimally. We present a greedy solution based on building $\{p^m\}$ iteratively with joint atom selection.

We begin with a tentative assignment of parameter vectors. In (4.17) each vector λ_i^m corresponds to the projection of u_i^m on $\mathcal{M}^m$. We assign $\{\lambda_i^m\}$ by picking a reference pattern Ψ^m for each class and then projecting each $\mathcal{U}^m$ onto $\mathcal{M}(\Psi^m)$. We also compute the cross-projection vectors $\{\lambda_i^{m,r}\}$, where

$$\lambda_i^{m,r} = \arg\min_{\lambda \in \Lambda} \|u_i^m - U_\lambda(p^r)\|$$

corresponds to the projection of u_i^m onto $\mathcal{M}^r$.

Then, we construct $\{p^m\}$ by gradually adding new atoms to each p^m. In the j-th iteration of the algorithm, we would like to optimize the parameters γ_j^m and coefficients c_j^m of the new atoms such that the weighted error E is minimized. Now we consider the j-th iteration and denote $\gamma^m = \gamma_j^m$, $c^m = c_j^m$. Then $\gamma = [\gamma^1\, \gamma^2\, \ldots\, \gamma^M]$ and $c = [c^1\, c^2\, \ldots\, c^M]$ are the optimization variables of the j-th iteration. We consider E as a function of γ and c similarly to Section 4.2 and propose to minimize E through a two-stage optimization. We first obtain an approximation $\tilde{E}$ of E, which is in a DC form. We minimize $\tilde{E}$ using the cutting plane algorithm and estimate a coarse solution, which is used as an initial solution in the second stage. Then in the second stage, we define a refined approximation $\hat{E}$ of E based on the tangent distances of images to the manifolds and minimize it with a gradient-descent algorithm.

The minimization of $\tilde{E}$ and $\hat{E}$ determines a solution for γ and c. We update the pattern p^m of each class by adding it the selected atom with parameters γ^m and coefficient c^m (in practice, we add an atom only if its coefficient is significant enough). Then, we recompute the transformation parameters $\{\lambda_i^m\}$ and $\{\lambda_i^{m,r}\}$ by projecting the images onto the new manifolds. We have observed that selecting the atoms by minimizing a combination of approximation and classification terms instead of only a classification term gives better results, especially for robustness to data noise. Still, we would like to make sure that the selected atoms improve the classification performance at the end of an iteration. Therefore, the decision of accepting the updates on the manifolds is taken according to the classification error E_c in (4.14). If E_c is not reduced we reject the updates and pass to the next iteration.[4] We continue the iterations until the classification error E_c converges. The termination of the algorithm is guaranteed by constraining E_c to be non-increasing during

[4] In the course of the algorithm, parameters β and α are adapted such that the emphasis is shifted from approximation capabilities in early phases to classification capabilities in later phases. This is explained in more detail in Section 4.3.3. For this reason, even if the classification error does not decrease in one iteration, it may do in the next one.

Algorithm 4 Joint Parameterized Atom Selection (JPATS)

1: **Input:**
$\mathcal{U} = \bigcup_{m=1}^{M} \mathcal{U}^m$: Set of observations for M signal classes
2: **Initialization:**
3: Determine tentative parameter vectors $\{\lambda_i^{m,r}\}$ by projecting $\{u_i^m\}$ on the transformation manifolds $\{\mathcal{M}(\Psi^m)\}$ of reference patterns $\{\Psi^m\}$.
4: $p_0^m = 0$ for $m = 1, \ldots, M$.
5: $j = 0$.
6: Initialize the sigmoid parameter β and the weight parameter α.
7: **repeat**
8: $j = j + 1$.
9: Optimize the joint atom parameters $\gamma = [\gamma^1\, \gamma^2 \ldots \gamma^M]$ and coefficients $c = [c^1\, c^2 \ldots c^M]$ with DC programming such that the error $\tilde{E}$ in (4.24) is minimized.
10: Further optimize γ and c with gradient descent such that the refined error $\hat{E}$ in (4.27) is minimized.
11: Update $p_j^m = p_{j-1}^m + c^m\, \phi_{\gamma^m}$ for $m = 1, \ldots, M$ if c^m is significant.
12: Update the parameter vectors $\{\lambda_i^{m,r}\}$.
13: Update β and α.
14: Check if the new manifolds reduce the classification error E_c. If not, reject the updates on p^m and $\{\lambda_i^{m,r}\}$, and go back to 9.
15: **until** the classification error E_c converges
16: **Output**:
$\{p^m\} = \{p_j^m\}$: A set of patterns whose transformation manifolds $\{\mathcal{M}^m\}$ represent the data classes $\mathcal{U}^m$

the iterations, which in return stabilizes the objective function E. We call this method Joint Parameterized Atom Selection (JPATS) and summarize it in Algorithm 4.

Let us come to the detailed description of the approximations of E in the two-stage optimization. Let $\{p_{j-1}^m\}$ and $\{\mathcal{M}_{j-1}^m\}$ denote the patterns and the corresponding transformation manifolds computed after $j-1$ iterations. For simplicity of notation, we will use the convention $\mathcal{M}^m = \mathcal{M}_j^m$ and $p^m = p_j^m$ throughout the derivations of $\tilde{E}$ and $\hat{E}$.

In the first step, we obtain $\tilde{E}$ in the form $\tilde{E} = \tilde{E}_a + \alpha\, \tilde{E}_c$, where $\tilde{E}_a$ and $\tilde{E}_c$ are respectively the approximations of E_a and E_c. The first term $\tilde{E}_a$ is simply given by the generalization of the approximation error in (4.8) to the multiple manifold case.

$$\tilde{E}_a = \sum_{m=1}^{M} \sum_{i=1}^{N_m} \|\tilde{e}_i^m\|^2 = \sum_{m=1}^{M} \sum_{i=1}^{N_m} \|v_i^m - c^m\, U_{\lambda_i^m}(\phi_{\gamma^m})\|^2 \tag{4.18}$$

where the parameters λ_i^m are the ones computed at the end of iteration $(j-1)$, and $v_i^m = u_i^m - U_{\lambda_i^m}(p_{j-1}^m)$.

Then, we derive $\tilde{E}_c$ in the following way. Notice that the classification error E_c in (4.14) is a discontinuous function of γ and c due to the discontinuity of the misclassification indicator function

I. Let $r(u_i^m)$ denote the index of the manifold with smallest distance to an image u_i^m among the manifolds of all classes except its own class m; i.e.,

$$r(u_i^m) = \arg\min_{r \neq m} d(u_i^m, \mathcal{M}^r).$$

Clearly, $r(u_i^m)$ can take different values throughout the iterations. However, for simplicity, in the j-th iteration we fix the indices $r(u_i^m)$ to their values attained at the end of iteration $(j-1)$ and denote them by the constants r_i^m. Then we can define the function

$$f(u_i^m) = d^2(u_i^m, \mathcal{M}^m) - d^2(u_i^m, \mathcal{M}^{r_i^m})$$

such that $I(u_i^m)$ corresponds to the unit step function of $f(u_i^m)$; i.e., $I(u_i^m) = u(f(u_i^m))$. Thus, if we replace the unit step function with the sigmoid function $S(x) = (1 + e^{-\beta x})^{-1}$, which is a common analytical approximation of the unit step, we obtain the approximation

$$S\big(f(u_i^m)\big) = \Big(1 + e^{-\beta f(u_i^m)}\Big)^{-1}$$

of $I(u_i^m)$. As the value of the positive scalar β tends to infinity, the sigmoid function approaches the unit step function. A continuous approximation of E_c is thus given by

$$\sum_{m=1}^{M} \sum_{i=1}^{N_m} S\big(f(u_i^m)\big). \tag{4.19}$$

Now, in order to minimize the function in (4.19) we do the following. We first compute

$$f_0(u_i^m) = d^2\left(u_i^m, \mathcal{M}_{j-1}^m\right) - d^2\left(u_i^m, \mathcal{M}_{j-1}^{r_i^m}\right)$$

for each image u_i^m. Then, applying a first-order expansion of S around each $f_0(u_i^m)$, we obtain the following approximation of the error term in (4.19)

$$\sum_{m=1}^{M} \sum_{i=1}^{N_m} \left(S\big(f_0(u_i^m)\big) + \left.\frac{dS}{df}\right|_{f=f_0(u_i^m)} \big(f(u_i^m) - f_0(u_i^m)\big) \right). \tag{4.20}$$

Since $f_0(u_i^m)$ and $S(f_0(u_i^m))$ are constants, the minimization of the expression in (4.20) becomes equivalent to the minimization of

$$\sum_{m=1}^{M} \sum_{i=1}^{N_m} \left.\frac{dS}{df}\right|_{f=f_0(u_i^m)} f(u_i^m) = \sum_{m=1}^{M} \sum_{i=1}^{N_m} \eta_i^m f(u_i^m) \tag{4.21}$$

where

$$\eta_i^m = \left.\frac{dS}{df}\right|_{f=f_0(u_i^m)} = \left.\frac{\beta\, e^{-\beta f}}{(1 + e^{-\beta f})^2}\right|_{f=f_0(u_i^m)}.$$

Let us rearrange (4.21) in a more convenient form. For each class index m, let $R^m = \{(i,k) :$

$r_i^k = m\}$ consist of the pairs of data and class indices of images that do not belong to class m but have $\mathcal{M}^m$ as their closest manifold among all manifolds except the one of their own class. Then (4.21) can be rewritten as

$$\sum_{m=1}^{M}\sum_{i=1}^{N_m} \eta_i^m \, d^2(u_i^m, \mathcal{M}^m) - \sum_{m=1}^{M}\sum_{(i,k)\in R^m} \eta_i^k \, d^2(u_i^k, \mathcal{M}^m). \tag{4.22}$$

As it is not easy to compute the distance terms $d^2(u_i^k, \mathcal{M}^m)$ directly, we proceed with the approximation $d^2(u_i^k, \mathcal{M}^m) \approx \|u_i^k - U_{\lambda_i^{k,m}}(p_{j-1}^m + c^m \, \phi_{\gamma^m})\|^2$, where the value of $\lambda_i^{k,m}$ is the one computed in iteration $(j-1)$. We finally get $\tilde{E}_c$ from (4.22) with this approximation.

$$\tilde{E}_c = \sum_{m=1}^{M}\sum_{i=1}^{N_m} \eta_i^m \, \|v_i^m - c^m \, U_{\lambda_i^m}(\phi_{\gamma^m})\|^2 - \sum_{m=1}^{M}\sum_{(i,k)\in R^m} \eta_i^k \, \|v_i^{k,m} - c^m \, U_{\lambda_i^{k,m}}(\phi_{\gamma^m})\|^2 \tag{4.23}$$

where $v_i^{k,m} = u_i^k - U_{\lambda_i^{k,m}}(p_{j-1}^m)$. Now, from (4.18) and (4.23) we can define

$$\tilde{E} = \tilde{E}_a + \alpha \, \tilde{E}_c. \tag{4.24}$$

Proposition 3. *$\tilde{E}$ is a DC function of γ and c. Moreover, if a DC decomposition for the components of the transformed atom $U_\lambda(\phi_\gamma)$ is known, a DC decomposition of $\tilde{E}$ is computable.*

The proof of Proposition 3 is given in Appendix A.4.

Now let us describe the term $\hat{E}$ that is used in the second stage of the optimization of E. We derive $\hat{E}$ by replacing the manifold distances by tangent distances; i.e., we use the approximation $d^2(u_i^k, \mathcal{M}^m) \approx d^2(u_i^k, \mathcal{S}_i^k(p^m))$, where $\mathcal{S}_i^k(p^m)$ is the first-order approximation of $\mathcal{M}^m$ around the point $U_{\lambda_i^{k,m}}(p^m)$. The tangent distance is derived in Appendix A.2. Let $w_i^m = u_i^m - U_{\lambda_i^m}(p^m)$ and $w_i^{k,m} = u_i^k - U_{\lambda_i^{k,m}}(p^m)$. Then the function E_a in (4.12) is approximated by

$$\hat{E}_a = \sum_{m=1}^{M}\sum_{i=1}^{N_m} \|w_i^m - T_i^m \left((T_i^m)^T \, T_i^m\right)^{-1} \, (T_i^m)^T w_i^m\|^2. \tag{4.25}$$

Similarly, the classification error function in (4.22) is approximated by

$$\begin{aligned}\hat{E}_c = &\sum_{m=1}^{M}\sum_{i=1}^{N_m} \eta_i^m \|w_i^m - T_i^m \left((T_i^m)^T \, T_i^m\right)^{-1} \, (T_i^m)^T w_i^m\|^2 \\ &- \sum_{m=1}^{M}\sum_{(i,k)\in R^m} \left(\eta_i^k \|w_i^{k,m} - T_i^{k,m} \left((T_i^{k,m})^T \, T_i^{k,m}\right)^{-1} \, (T_i^{k,m})^T w_i^{k,m}\|^2\right).\end{aligned} \tag{4.26}$$

Here T_i^m and $T_i^{k,m}$ denote the $n \times d$ matrices whose columns are the tangent vectors to the manifold

$\mathcal{M}^m$ at respectively the points $U_{\lambda_i^m}(p^m)$ and $U_{\lambda_i^{k,m}}(p^m)$. From (4.25) and (4.26) we can finally define

$$\hat{E} = \hat{E}_a + \alpha\, \hat{E}_c. \tag{4.27}$$

Let us briefly discuss the effect of the approximations made on the original cost function E. The accuracy of approximating the unit step function with a sigmoid in (4.19) can be adjusted by changing the slope of the sigmoid (see also the note in Section 4.3.3). Then, in order for the linear approximation of the sigmoid in (4.20) to be valid, the values of $f(u_i^m)$ must be sufficiently close to their base values $f_0(u_i^m)$. The effect of this linearization can be alleviated by updating the base values $f_0(u_i^m)$ several times in an iteration. The rest of the approximations are similar to those discussed in Section 4.2.2.

4.3.3 Implementation details

We now discuss some points related to the implementation of JPATS. We first explain the choice of the parameter β in Algorithm 4. Notice that the function $S\big(f(u_i^m)\big)$ can also be interpreted as the probability of misclassifying u_i^m upon updating the manifolds at the end of the iteration. When u_i^m gets closer to its true manifold $\mathcal{M}^m$, $f(u_i^m)$ decreases and $S\big(f(u_i^m)\big)$ decays to 0. Similarly, when u_i^m gets away from $\mathcal{M}^m$, $S\big(f(u_i^m)\big)$ approaches 1. The probabilistic interpretation of the function $S\big(f(u_i^m)\big)$ stems naturally from its shape. Consequently, the approximate error in (4.19) corresponds to the sum of the probabilities of misclassifying the input images. Based on this interpretation, we propose to update β according to the statistics drawn from the data. For each u_i^m, we examine the value of $f(u_i^m)$ at the beginning the iteration and the value of $I(u_i^m)$ at the end of the iteration. Then we pick β such that the shape of the sigmoid matches the $I(u_i^m)$ vs. $f(u_i^m)$ plot. Such an adaptive choice of β also provides the following flexibility. In early phases of the process where the total misclassification rate is relatively high, β usually has small values, which yields slowly changing sigmoids. Therefore, a relatively large portion of the input images have an effect on the choice of the new atoms. However, in later phases, as the total misclassification rate decreases, β usually takes larger values resulting in sharper sigmoids, which gives misclassified images more weight in atom selection.

Then, we comment on the choice of the weight parameter α. In principle, α can be set to have any nonnegative value. Setting $\alpha = 0$ corresponds to a purely approximation-based procedure that computes the manifolds individually with PATS, whereas a large α yields a learning algorithm that is rather driven by classification objectives. However, we have observed that a good choice in practice consists of selecting a small value for α at the beginning and increasing it gradually.[5] This guides the algorithm to first capture the main characteristics of input signals, and then encourage the selection of features that enhance class-separability.

Finally, we have made the following simplification in the implementation of the DC programming block. The number of optimization variables is $(s+1)M$ in our problem, where s is the dimension of $\mathcal{D}$ and M is the number of classes. Although the cutting plane algorithm works well for low-dimensional solution spaces, it becomes computationally very costly in high dimensions. Therefore, in the implementation of JPATS we partition the variables into subsets and optimize the subsets one

[5] In our setup, we control the α parameter by using a shifted and scaled sigmoid function. The initial and final values of the sigmoid are around 0.5 and 10; and its center is typically attained at iterations 5-7 of Algorithm 4.

by one. Although there is no guarantee of finding the globally optimal solution in this case, we have experimentally observed that one can still obtain reasonably good results regarding the complexity-accuracy trade-off. In order to handle high-dimensional solution spaces, one can alternatively replace the cutting plane algorithm with another DC solver such as DCA [90] or CCCP [91]. These methods reduce the original DC program to the iterative solution of a pair of dual convex programs, which improves the computational complexity significantly at the expense of losing global optimality guarantees. Another issue affecting the efficiency of the DC programming block is the size of the solution space. We have seen that it is useful to add a preliminary block that locates a good search region before the DC block. This can be achieved using a coarse grid in the solution space or a global search method such as the genetic algorithm or particle swarm optimization. Note that one may also minimize the objective function by using only a global search method. However, in experiments we have seen that the final value of the objective function is the smallest when both global search and DC optimization are employed.

4.3.4 Experimental results

We now evaluate the performance of JPATS with experiments on transformation-invariant classification. We test the algorithm on two data sets consisting of handwritten digits [92] and microbiological images [96]. In the digits experiment, we use the transformation manifold model in (2.2). In the microbiological images experiment, we use the model

$$\mathcal{M}(p) = \{U_\lambda(p) : \lambda = (\theta, t_x, t_y, s) \in \Lambda\} \subset \mathbb{R}^n, \tag{4.28}$$

where s denotes an isotropic scale change. In both experiments, we use the dictionary model in (2.10) and the Gaussian mother function in (4.11).

The first experiment is conducted on the images of the "2,3,5,8,9" digits, which lead to a relatively higher misclassification rate than the rest of the digits. The data sets are generated by randomly selecting 200 training and 200 test images for each digit and applying random geometric transformations consisting of rotation, anisotropic scaling and translation. The images of each digit are considered as the observations of a different signal class.

The second experiment is done on some sequences from the microbiology video collection of the Natural History Museum [96], which contains short video clips of living protists. We run the experiment on 6 different species (Discocephalus sp., Epiclintes ambiguus, Oxytricha sp., Scyphidia sp., Stentor roeseli, Stylonychia sp.), and we use three sample videos for each one. Each species is considered as a different class. The manifold in (4.28) provides a suitable model, as the rotation and translations describe well the movements of the protists, and the isotropic scaling compensates for zoom changes. However, there is still some deviation from the manifold, as a result of noise, small nonrigid protist articulations and occasional recording of different individuals in different videos. For each species, we experiment on a subset of frames from all three sequences. We preprocess the frames by conversion to greyscale, smoothing and thresholding. Then, for each class, we randomly select 70 training and 35 test images.

In the experiments we compare the methods listed below. In the first four methods, we apply the algorithms on the training images in order to build PTMs. Then we compute the misclassification rate of the test images. The class label of a test image is estimated by identifying the smallest

distance between the image and the computed manifolds. The algorithms work as follows.

- JPATS: We jointly build PTMs for all classes with the proposed method.
- PATS: We compute individual PTMs for each class with PATS.
- SMP on aligned patterns: We compute individual PTMs for each class as explained in Section 4.2.3.
- SAS on aligned patterns: We use the untransformed/aligned images of all classes and select a set of Gaussian atoms with SAS [81]. We set the weight factor to $\lambda = 2$ in [81]. Then, for each class we build a PTM by forming a pattern, where the selected atoms are weighted with their average coefficients.
- LLA: We compute a locally linear approximation using the training images of each class. A test image is classified by identifying its $(d+1)$-nearest neighbors among the training images of each class, computing its distance to the plane passing through the nearest neighbors, and comparing its distances to the planes of different classes.
- SLLE: We compute a low-dimensional embedding of the training images with the Supervised Locally Linear Embedding algorithm [17] and assign the class labels of the test images via nearest-neighbor classification in the embedded domain.
- LDA: Linear Discriminant Analysis on aligned data. The better one of linear and quadratic kernels is picked in each experiment.
- Neural Network: A feed-forward backpropagation network for pattern recognition is used on aligned data.

The results are presented in Figures 4.9 and 4.10 respectively for the digit and microbiological image experiments. In Figures 4.9(a) and 4.10(a), a data set image from each class is shown. Some typical representative patterns computed with JPATS, PATS and the reference methods are shown in Figures 4.9(b)-4.9(e) and 4.10(b)-4.10(e). Figures 4.9(f) and 4.10(f) show the misclassification rates of test images (in percentage) vs. the number of atoms per class. Both plots are obtained by averaging the results of 5 repetitions of the experiment with different training and test sets. The results show that JPATS yields the best classification performance in general, although PATS produces visually more pleasant patterns. This can be explained as follows. PATS is designed to minimize the approximation error; and the assessment of the visual quality of the computed patterns is rather dependent on their approximation capabilities. The local features that are common to different classes appear in the representative patterns of all these classes built with PATS, which produces an output that matches visual perception. However, if a local feature is common to several classes, its inclusion in the representative patterns does not contribute much to the discrimination among classes; therefore, these non-distinctive features are not emphasized in the output of JPATS. On the other hand, the local features that are rather special to one class are more pronounced in JPATS compared to PATS. In fact, due to the classification error term in JPATS, the algorithm tends to select atoms that "push" a manifold away from the samples of other classes. For instance, in Figure 4.9(b), the top and bottom arcs of the "8" digit are not as apparent, since the other digits

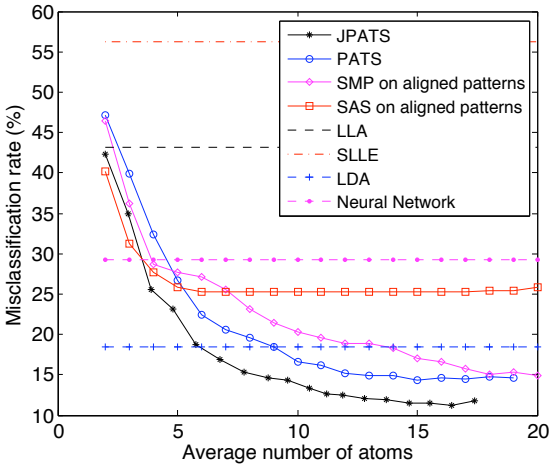
(a) Sample images from data set

(b) Patterns built with JPATS

(c) Patterns built with PATS

(d) Patterns built with SMP on aligned patterns

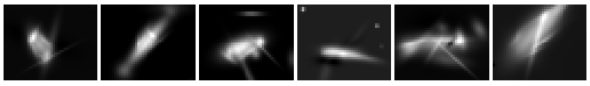
(e) Patterns built with SAS on aligned patterns



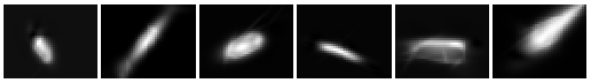
(f) Classification errors of test images

Figure 4.9: Performance of the classification-driven learning algorithms on handwritten digits data set

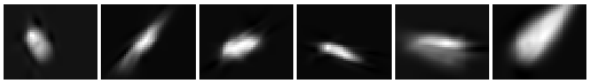
(a) Sample images from data set



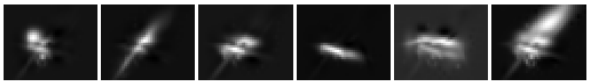
(b) Patterns built with JPATS



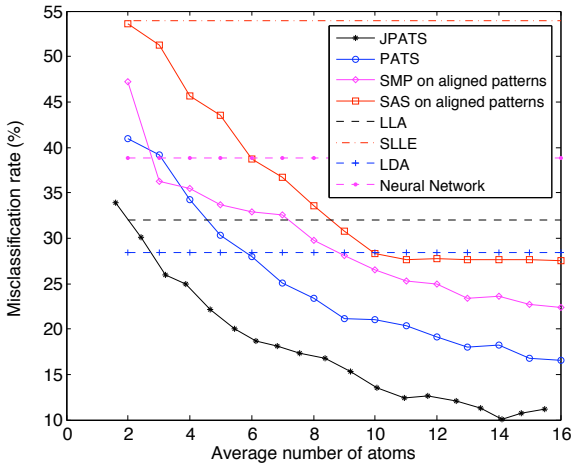
(c) Patterns built with PATS



(d) Patterns built with SMP on aligned patterns



(e) Patterns built with SAS on aligned patterns



(f) Classification errors of test images

Figure 4.10: Performance of the classification-driven learning algorithms on microbiological images data set

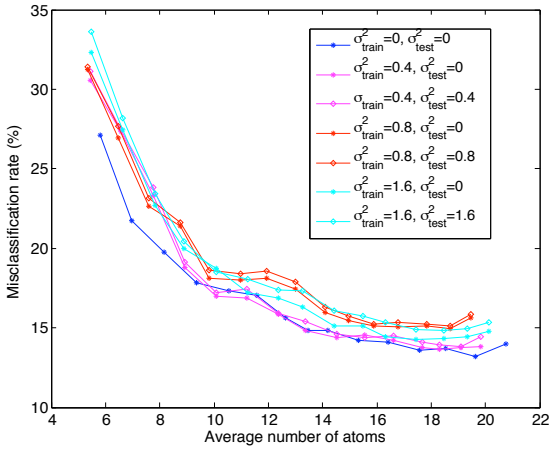


Figure 4.11: Performance of JPATS on noisy data. The noise variance $\sigma^2 = 1.6$ corresponds to an SNR of 6.35 dB.

also have similar features. However, the crossover of "8" is specific to this class; therefore, it is prominent in the output. Similarly, the straight edge of "9" is also characteristic of this class and emphasized in the learned pattern.

Next, we examine the effect of data noise on the performance of JPATS. We create several data sets by corrupting the digits data set used in the previous experiment with additive Gaussian noise of different variances. For each noise level, we look into two cases, where only training images are corrupted in the first one, and both training and test images are corrupted in the second one. The misclassification rate of test images are plotted in Figure 4.11, where σ^2_{train} and σ^2_{test} denote the noise variances of training and test images. The data noise has a small influence on the performance of the algorithm. The final increase in the misclassification rate is bounded by 2.7% even when the noise energy reaches 23% of the signal energy. The robustness to noise is achieved due to the fact that the algorithm is designed to generate a smooth pattern that fits all images simultaneously, which enables it to smooth data noise. The other PTM-based methods are also expected to exhibit similar noise behaviors.

Finally, we evaluate the performances of PATS and JPATS in a setting where the test images contain some outliers that do not belong to any of the classes. We run the experiment on the digit data set used in the experiment of Figure 4.9. The training phase of the algorithms is as before: In both methods, the manifolds are learned using only training images of known classes. However, test images are contaminated with 200 outlier images that do not belong to any of the target classes, where the number of test images in each class is also 200. Each outlier image is generated by randomly selecting one test image from each class, taking the average of these images, corrupting the average image with additive Gaussian noise, and finally normalizing it. Thus, all outlier images have unit norm, while a typical class-sample test image with unit scale ($s_x = 1, s_y = 1$) also has unit norm. Then, test images are classified using the manifolds learned with PATS and JPATS as follows. If the distance between a test image and the closest manifold is larger than a threshold, the image is labeled as an "outlier"; and if this distance is smaller than the threshold, it is assigned

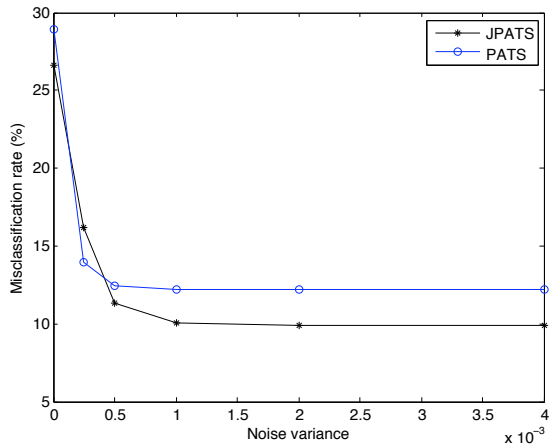


Figure 4.12: Performance of PATS and JPATS in a classification setting with outlier test images that do not belong to any class. For the noise variance $\sigma^2 = 0.5 \times 10^{-3}$, the ratio between the norms of the noise component and the average component of an outlier image is 0.65.

the class label of the closest manifold as before. The experiment is repeated for different values of the noise variance for the Gaussian noise component of the outlier images. The threshold used for each noise level is numerically selected in a sample run of the experiment such that it gives the best classification rate and fixed for the other runs, separately for PATS and JPATS. The results are presented in Figure 4.12, which are the average of 5 runs. The misclassification rate is the percentage of test images that have not been assigned the correct class label or the correct "outlier" label. In the plots shown in Figure 4.12, the noise variance 0 corresponds to the case that outlier images are the averages of some test images coming from different classes. This is the most challenging instance of the experiment, as outliers come from a region close to class samples and it is relatively difficult to distinguish them from class samples. Consequently, the optimal threshold that gives the smallest misclassification rate is high for this instance, resulting in labeling all outliers as class samples. The performance of JPATS is better than PATS in this case, as the overall classification rate is determined by the classification rate of class samples. Then, as the noise variance is increased, the components of the outlier images in random directions in the image space are amplified, making it thus easier to distinguish them from class samples. It is seen that JPATS performs better than PATS in most cases. However, for the smallest nonzero noise variance value $\sigma^2 = 0.25 \times 10^{-3}$, PATS is observed to give a better classification rate than JPATS. This can be explained as follows. In this experiment, JPATS is trained according to the hypothesis that all test images belong to a valid class. For this reason, in order to increase the distance between a manifold and the samples of other classes, JPATS may occasionally pick some atoms that push a manifold away from the samples of its own class as well. This slight increase in the distance between the manifold and the samples of its own class renders it difficult for JPATS to distinguish between real class samples and challenging outliers that are very close to class samples. In order to get the best performance from JPATS in such a setting with outliers, one can tune the α parameter to a suitable value depending on outlier characteristics such that a sufficiently strict control is imposed

on the distance between the learned manifolds and the samples of their own classes through the approximation term E_a.

4.4 Complexity Analysis

Let us now examine the complexities of the proposed algorithms. We begin with the PATS method summarized in Algorithm 3. There are three blocks in the main loop of the algorithm. The first one minimizes $\tilde{E}$ with DC programming, the second one minimizes $\hat{E}$ via gradient descent, and the third one computes the projections of $\{u_i\}$ on the manifold. In the analysis of the first block, it is important to distinguish between the complexities of the DC solver and the computation of the DC decomposition of $\tilde{E}$. The former depends on the selected solver. The cutting plane method involves the construction of polytopes in the search space; therefore, it has an exponential complexity in the number of atom parameters s (dimension of $\mathcal{D}$). However, one can also use a technique such as DCA [90]. In this case, the solution of the dual convex programs involves the evaluation of the subdifferentials of the functions constituting the DC decomposition. In our problem, this corresponds to the evaluation of gradients since the decomposing functions are differentiable. The gradient can be numerically evaluated using finite differences; therefore, the complexity of such a solver is at most linear in s. Next, it can be seen from equations (A.1) and (A.2) that the cost of computing the DC decomposition of $\tilde{E}$ is linear in the image resolution n and the number of samples N. Hence, the complexity of the first block becomes $O(2^s\, n\, N)$ for cutting plane, and $O(s\, n\, N)$ for DCA. In the analysis of the second block, it can be easily shown that the complexity of calculating the vector $w_i - T_i(T_i^T\, T_i)^{-1}\, T_i^T w_i$ in (A.3) is $O(d\, n^2)$, where d is the dimension of $\mathcal{M}(p)$. Therefore, the cost of computing the total squared tangent distance $\hat{E}$ in (A.3) is obtained as $O(d\, n^2\, N)$. As we minimize $\hat{E}$ with gradient descent using finite differences, the complexity of the second block is $O(s\, d\, n^2\, N)$. Finally, the cost of updating the projections of $\{u_i\}$ on $\mathcal{M}(p)$ is $O(d\, n\, N)$ in our actual implementation, because we minimize the distance between each $\{u_i\}$ and $\mathcal{M}(p)$ by performing a line search along each dimension of $\mathcal{M}(p)$. Thus, taking DCA as a reference for the first block, we can summarize the overall complexity of PATS as $O(s\, d\, n^2\, N)$. In the experiments in Figure 4.9, the practical runtime of a non-optimized MATLAB implementation of the PATS algorithm to learn one atom from 200 training images was around twenty minutes.

We examine the complexity of JPATS given in Algorithm 4 similarly. From (A.4) it is seen that the cost of computing the DC decomposition of $\tilde{E}$ is linear in N_J and n, where $N_J = \sum_{m=1}^{M} N_m = \sum_{m=1}^{M} |R^m|$. The complexity of the first block with respect to DCA is therefore $O(s\, n\, N_J)$. Then, (4.25) and (4.26) show that the cost of computing $\hat{E}$ is $O(d\, n^2\, N_J)$. The complexity of the second block is thus $O(s\, d\, n^2\, N_J)$. Finally, the third block has complexity $O(d\, n\, N_J\, M)$, since each image is reprojected on each manifold. Therefore, the overall complexity of JPATS is $O\big(N_J(s\, d\, n^2 + d\, n\, M)\big)$. The complexity of selecting an atom with "SMP on aligned patterns", which has the closest performance to the proposed methods, can be similarly obtained as $O(N_J\, n\, D)$, where D denotes the cardinality of the discrete dictionary used. We remark that the proposed method is more suitable for applications where the manifolds are learned "offline" and then used for the classification of test data. Moreover, there might be ways to improve the complexity-accuracy trade-off depending on the application. For instance, one might prefer to sacrifice on accuracy for a less complex solution by omitting step 9 or 10 of Algorithm 4. Also, if the class-representative

manifolds are well-separated, it may be sufficient to use the PATS algorithm instead of JPATS. An option for achieving a high-speed PTM learning is to build a tentative representative pattern, for instance with "SMP on aligned patterns", in a preliminary analysis step and register the input images with respect to this pattern. Then, one may speed up the learning significantly by discarding the projection update steps and optimizing the atoms of the representative pattern by minimizing only the error in (4.8) with a fast minimizer such as the gradient descent algorithm.

4.5 Conclusion

In this chapter, we have studied the problem of building smooth pattern transformation manifolds for the transformation-invariant representation of sets of visual signals. The manifold learning problem is cast as the construction of a representative pattern as a linear combination of smooth parametric atoms. The manifold is then created by geometric transformations of this pattern. The smoothness of the computed manifolds is ensured by the smoothness of the constituting parametric atoms, which is a desirable property that facilitates the usage of the manifolds in image analysis. We have described a single manifold learning algorithm for approximation and a multiple manifold learning algorithm for classification. Experimental results show that the proposed methods provide a good approximation and classification accuracy compared to reference methods. The proposed methods are applicable to unregistered data that can be approximated by 2-D pattern transformations with a known geometric transformation model. Our study shows the potentials of parametric models for data representations, which can be effectively used to achieve high performance in the registration, coding and classification of geometrically transformed image sets.

Chapter 5

Analysis of Image Registration with Tangent Distance

5.1 Overview of Tangent Distance Analysis

In the previous chapter, we have studied the problem of learning pattern transformation manifold models from image data sets. We have seen that one can attain a good performance in transformation-invariant image analysis applications with properly constructed PTM models. Meanwhile, given a reference transformation manifold model representing a group of visual signals, the analysis of a query image requires its registration with respect to the transformation manifold, which is an essential and challenging problem. In Chapter 3, we have proposed a constructive solution for image registration based on the discretization of transformation manifolds. However, sampling may not be the optimal solution for registration in some applications, e.g. due to memory requirements or other constraints. Hence, in this chapter, we study the image registration problem in a different setting. We examine the alignment of images using linear approximations of manifolds, which is known as the tangent distance method. We derive performance bounds in order to understand the limits of this popular method, which is helpful for using it more efficiently in registration problems. We specifically focus on pattern transformation manifolds that are used in this thesis.

The tangent distance method is based on estimating the projection of a target image onto the transformation manifold of a reference image by exploiting a first-order Taylor approximation of the transformation manifold [3]. In image alignment with tangent distance, the reference transformation parameters around which the manifold is linearized are required to be sufficiently close to the optimal transformation parameters corresponding to the exact projection of the target image onto the manifold, so that the linear approximation of the manifold is valid and the optimal transformation parameters can be estimated accurately. When the distance between the reference and optimal transformation parameters is large, an efficient way to get around this limitation is to apply the tangent distance method in a hierarchical manner [3]. In hierarchical alignment, a pyramid of low-pass filtered and downsampled versions of the reference and target images is built, and the alignment is achieved in a coarse-to-fine manner. The transformation parameters are first roughly estimated using the smoothest images in the pyramid, and then refined progressively by passing

to the fine scales. The low-pass filtering applied in coarse scales helps to reduce the nonlinearity of the manifold, which renders the linear approximation more accurate and allows the recovery of relatively large transformations. Once the transformation parameters are estimated roughly in coarse scales, the remaining increment in the transformation parameters to be computed in fine scales is relatively small and the linear approximation of the manifold is therefore accurate. The hierarchical estimation of transformation parameters using manifold linearizations is very common in image registration [2], motion estimation [26], [24] and stereo vision [22].

Although the tangent distance method is frequently used in image analysis applications, its performance has not been theoretically studied for general transformation models yet. The treatments in [27], [39] and [40] limit their scope to gradient-based optical flow estimation, which corresponds to a transformation model consisting only of 2-D translations. The examination of the effect of filtering on the accuracy of tangent distance is especially important in hierarchical alignment, so that the filter size at each stage of the pyramid can be properly selected. In this chapter, we present a theoretical analysis of hierarchical image alignment with the tangent distance method, where we aim to characterize the alignment error as a function of the filter size. Previous works such as [3] and [97] using the tangent distance in image classification and clustering compute the distance in a symmetric fashion; i.e., they linearize the transformation manifolds of both the reference and target images and compute the subspace-to-subspace distance. In our analysis we focus on the alignment accuracy aspect of the tangent distance method and consider the point-to-subspace distance obtained by linearizing the transformation manifold of only the reference image, e.g., as in [2], [22], which is more suitable for image registration purposes. We consider a setting where the reference image is noiseless, the target image is a noisy observation of a transformed version of the reference image, and both the reference and target images are presmoothed with low-pass filters before alignment.

We first derive an upper bound for the alignment error, which is defined as the parameter-domain distance between the optimal transformation parameter vector and its estimation with the tangent distance method. The upper bound for the alignment error is obtained in terms of the noise level of the target image, the parameter-domain distance between the reference transformation parameters around which the manifold is linearized and the optimal transformation parameters, and some geometric parameters of the transformation manifold such as curvature and metric tensor. The alignment error is shown to be linearly increasing with the manifold curvature and the noise level, and monotonically increasing with the parameter-domain distance between the reference and optimal transformation parameters. Then, in order to study the relation between the alignment error and the amount of low-pass filtering in a hierarchical setting, we employ an analytic and parametric representation of the reference pattern and analyze the dependence of the alignment error on the size of the low-pass filter kernel. Using the notation $O(\cdot)$ to represent the approximate rate of variations of parameters with the noise level and the filter size, we show that the alignment error decreases with the filter size ρ for small filter kernels at a rate of $O(1+(1+\rho^2)^{-1/2})$. This is due to the fact that filtering smooths the manifold and decreases its nonlinearity, which improves the accuracy of the linear approximation of the manifold. However, as one keeps increasing the filter size, the decrease in the alignment error due to the improvement of the manifold nonlinearity converges, and the error starts to increase with filtering at an approximate rate of $O(\rho)$ at relatively large values of the filter size. The increase in the error stems from the adverse effect of filtering, which amplifies the alignment error caused by image noise. Therefore, our main finding is that, in a

noisy setting where the target image is not exactly on the transformation manifold of the reference image, the filter kernel has an optimal size where the alignment error takes its minimum value. We also examine the dependence of the alignment error obtained with the filtered images on the initial noise level (the distance between the target image and the transformation manifold of the reference image before filtering), and show that the error is linearly proportional to the initial noise level. Another finding of our study is that, for transformation models that involve a scale change of the reference pattern, the operations of filtering and applying a geometric transformation do not commute and filtering causes a secondary source of noise. Hence, even if the initial noise level is zero, the alignment error increases with the filter size at relatively big filter sizes and the filter size has an optimal value minimizing the alignment error for such transformation models. Finally, we obtain an approximate expression for the optimal filter size that minimizes the alignment error, which indicates that the filter size should be chosen as inversely proportional to the square-root of the noise level, and proportional to the square-root of the distance between the reference and the optimal transformation parameters. This justifies the usage of the coarse-to-fine strategy in image alignment with tangent distance: Large filters are used at coarse scales, where the transformation to be estimated is relatively large. Then, at fine scales small filters are preferable since the remaining amount of transformation to be estimated is smaller.

This chapter is organized as follows. First, in Section 5.2, we give an overview of image registration with the tangent distance method and formulate the registration analysis problem. Then, in Section 5.3, we present a theoretical analysis of the tangent distance method, where we first state an upper bound for the alignment error and then examine its variation with the noise level and filtering. In Section 5.4, we evaluate our analysis with some experiments and in Section 5.5 we give a discussion of our results in comparison with previous works. We conclude in Section 5.6.

5.2 Image Registration with Tangent Distance

The computation of the exact projection of a target image onto a reference transformation manifold is a complicated optimization problem, especially when the manifold is high-dimensional and generated by complex geometric transformations. The tangent distance method proposes to solve this problem by using a first-order approximation of the transformation manifold, which is illustrated in Figure 5.1. In the figure, $\mathcal{M}(p)$ is the transformation manifold of the reference pattern p defined over the parameter domain Λ, and q is the target image to be aligned with p. The exact projection of q on $\mathcal{M}(p)$ is the point p_{λ_o}, so that λ_o is the optimal transformation parameter vector that best aligns p with q. In order to estimate λ_o with the tangent distance method, a first order approximation $\mathcal{S}_{\lambda_r}(p)$ of the manifold $\mathcal{M}(p)$ is computed at a reference point p_{λ_r}, which is preferably not too distant from p_{λ_o}. The distance of q to $\mathcal{S}_{\lambda_r}(p)$ can be easily computed with a least squares solution and the point of projection on $\mathcal{S}_{\lambda_r}(p)$ gives the transformation parameter vector λ_e, which is the estimate of λ_o.

In the following, we first settle the notations and describe the tangent distance method formally. We then formulate the registration analysis problem studied in this chapter.

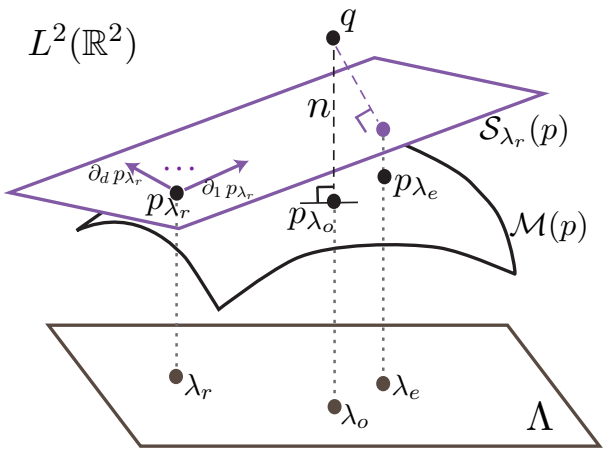


Figure 5.1: Illustration of image alignment with the tangent distance method. $\mathcal{S}_{\lambda_r}(p)$ is the first-order approximation of the transformation manifold $\mathcal{M}(p)$ around the reference point p_{λ_r}. The estimate λ_e of the optimal transformation parameters λ_o is obtained by computing the orthogonal projection of the target image q onto $\mathcal{S}_{\lambda_r}(p)$.

5.2.1 Notations

Let $p \in L^2(\mathbb{R}^2)$ be a reference pattern that is C^2-smooth with square-integrable derivatives and $q \in L^2(\mathbb{R}^2)$ be a target pattern. Let $\Lambda \subset \mathbb{R}^d$ denote a compact, d-dimensional transformation parameter domain and

$$\lambda = [\lambda^1\ \lambda^2\ \cdots\ \lambda^d] \in \Lambda$$

be a transformation parameter vector. We denote the pattern obtained by applying p the geometric transformation specified by λ as $A_\lambda(p) \in L^2(\mathbb{R}^2)$. We use the notation of Chapter 4 for geometrically transformed images and denote $X = [x\ y]^T$,

$$A_\lambda(p)(X) = p(a(\lambda, X)) \tag{5.1}$$

and $a_\lambda(X) = a(\lambda, X)$. We assume that $a : \Lambda \times \mathbb{R}^2 \to \mathbb{R}^2$ is a C^2-smooth function representing the change of coordinates defined by λ; and $a_\lambda : \mathbb{R}^2 \to \mathbb{R}^2$ is a bijection for a fixed λ.

Let us write $p_\lambda = A_\lambda(p)$ for convenience. Then, the transformation manifold $\mathcal{M}(p)$ of the pattern p is given by

$$\mathcal{M}(p) = \{p_\lambda : \lambda \in \Lambda\} \subset L^2(\mathbb{R}^2)$$

which consists of transformed versions of p over the parameter domain Λ. Since a and p are C^2-smooth, the local embedding of $\mathcal{M}(p)$ in $L^2(\mathbb{R}^2)$ is C^2-smooth. Therefore, the first and second-order derivatives of manifold points with respect to the transformation parameters exist. Let us denote the derivative of the manifold point p_λ with respect to the i-th transformation parameter λ^i as $\partial_i\, p_\lambda$, where

$$\partial_i\, p_\lambda(X) = \frac{\partial\, p_\lambda(X)}{\partial \lambda^i}.$$

The derivatives $\partial_i p_\lambda$ are also called tangent vectors. Similarly, we denote the second-order derivatives by

$$\partial_{ij} p_\lambda(X) = \frac{\partial^2 p_\lambda(X)}{\partial \lambda^i \partial \lambda^j}.$$

Then, the tangent space $T_\lambda \mathcal{M}(p)$ of the manifold at a point p_λ is the subspace generated by the tangent vectors at p_λ

$$T_\lambda \mathcal{M}(p) = \left\{ \sum_{i=1}^{d} \partial_i p_\lambda \, \zeta^i : \zeta \in \mathbb{R}^d \right\} \subset L^2(\mathbb{R}^2) \tag{5.2}$$

where $\{\partial_i p_\lambda\}_{i=1}^d$ are the basis vectors of $T_\lambda \mathcal{M}(p)$, and $\{\zeta^i\}_{i=1}^d$ are the coefficients in the representation of a vector in $T_\lambda \mathcal{M}(p)$ in terms of the basis vectors.

Lastly, given a reference pattern p and a target pattern q, we denote the optimal transformation parameter vector as

$$\lambda_o = \arg\min_{\lambda \in \Lambda} \|q - p_\lambda\|^2 \tag{5.3}$$

which gives the projection p_{λ_o} of q onto $\mathcal{M}(p)$. The purpose of image registration methods is to compute λ_o. However, the exact calculation of λ_o is difficult in general, since the nonlinear and highly intricate geometric structure of pattern transformation manifolds renders the distance minimization problem quite complicated. The tangent distance method simplifies this problem to a least squares problem, which is described below.

5.2.2 Tangent distance algorithm

In alignment with the tangent distance, transformation parameters are estimated by using a linear approximation of the manifold $\mathcal{M}(p)$ and then computing λ_o by minimizing the distance of q to the linear approximation of $\mathcal{M}(p)$ [2]. The first-order approximation of $\mathcal{M}(p)$ around a reference manifold point p_{λ_r} is given by

$$\mathcal{S}_{\lambda_r}(p) = \{p_{\lambda_r} + \sum_{i=1}^{d} \partial_i p_{\lambda_r} (\lambda^i - \lambda_r^i) : \lambda \in \mathbb{R}^d\} \subset L^2(\mathbb{R}^2).$$

Then, the estimate λ_e of λ_o with the tangent distance method is given by the solution of the least squares problem

$$\lambda_e = \arg\min_{\lambda \in \mathbb{R}^d} \|q - p_{\lambda_r} - \sum_{i=1}^{d} \partial_i p_{\lambda_r} (\lambda^i - \lambda_r^i)\|^2. \tag{5.4}$$

The solution of the above problem can be obtained as

$$\lambda_e = \lambda_r + [\mathcal{G}_{ij}(\lambda_r)]^{-1} [\langle q - p_{\lambda_r}, \partial_i p_{\lambda_r} \rangle] \tag{5.5}$$

where $[\mathcal{G}_{ij}(\lambda)] \in \mathbb{R}^{d \times d}$ is the matrix representation of the metric tensor $\mathcal{G}_{ij}(\lambda) = \langle \partial_i p_\lambda, \partial_j p_\lambda \rangle$ induced from the standard inner product on $L^2(\mathbb{R}^2)$. Hence, the (i,j)-th entry of $[\mathcal{G}_{ij}(\lambda)]$ is $\mathcal{G}_{ij}(\lambda)$. Similarly, $[\langle q - p_{\lambda_r}, \partial_i p_{\lambda_r} \rangle]$ represents the $d \times 1$ matrix whose i-th entry is $\langle q - p_{\lambda_r}, \partial_i p_{\lambda_r} \rangle$. The estimate λ_e of the transformation parameters obtained by solving (5.4) is expected to be closer

to the optimal solution λ_o than the reference parameters λ_r; therefore, λ_e can be regarded as a refinement of λ_r if the reference parameters λ_r are considered as an initial guess for the optimal ones λ_o. The estimation of transformation parameters with the tangent distance method is illustrated in Figure 5.1.

5.2.3 Problem formulation

From (5.3), we can decompose the target image q as

$$q = p_{\lambda_o} + n$$

where p_{λ_o} is the projection onto the manifold $\mathcal{M}(p)$ and $n \in L^2(\mathbb{R}^2)$ is the noise representing the deviation of q from $\mathcal{M}(p)$. We define the noise level parameter

$$\nu = \|n\|$$

as the distance of the target pattern to the translation manifold of the reference pattern.

We can now formulate the problems that we study in this chapter. Our first purpose is to examine the deviation between the optimal transformation parameter vector λ_o and its estimate λ_e, which defines the alignment error of the tangent distance method. In particular, we would like to find an upper bound for the alignment error $\|\lambda_e - \lambda_o\|$ in terms of the noise level ν of the target image, the known geometric parameters of the manifold $\mathcal{M}(p)$ that can be computed from p (such as its curvature and metric tensor), and the distance $\|\lambda_o - \lambda_r\|$ between the optimal and the reference transformation parameters. This states a bound on how much the initial guess λ_r for λ_o can be improved, given the proximity of λ_r with respect to λ_o. We thus present an upper bound for the alignment error $\|\lambda_e - \lambda_o\|$ in Section 5.3.1. Note that it is also possible to formulate the alignment error as the manifold distance estimation error measured in the ambient space $L^2(\mathbb{R}^2)$. However, in this study, we characterize the error in the parameter space Λ instead of the ambient space $L^2(\mathbb{R}^2)$ because of the following reason. The errors in the parameter domain and the ambient space are expected to have similar behaviors. Meanwhile, since we examine the problem in a multiscale setting, it is easier to characterize the error in the parameter domain as the distances in the ambient space are not invariant to smoothing.

Next, our second and main goal is to examine how the alignment error varies when the reference and target patterns are smoothed with a low-pass filter. We formalize this problem as follows. We consider a Gaussian kernel for the low-pass filter, since it is a popular smoothing kernel whose distinctive properties have been well-studied in scale-space theory [98]. Let

$$\phi(X) = e^{-X^T X} = e^{-(x^2+y^2)}$$

denote a Gaussian mother function. Then, the family of functions

$$\frac{1}{\pi\rho^2}\phi_\rho(X) \tag{5.6}$$

define variable-sized, unit L^1-norm Gaussian low-pass filters, where $\phi_\rho(X) = \phi(\Upsilon^{-1}(X))$ is a scaled

version of the mother function $\phi(X)$ with

$$\Upsilon = \begin{bmatrix} \rho & 0 \\ 0 & \rho \end{bmatrix}. \tag{5.7}$$

Here, the scale parameter ρ corresponds to the radius of the filter kernel, which controls the filter size. When the tangent distance method is used in a multiscale registration setting, the transformation parameters are estimated using the filtered versions of the reference and target patterns

$$\hat{p}(X) = \frac{1}{\pi\rho^2}\,(\phi_\rho * p)(X) \qquad\qquad \hat{q}(X) = \frac{1}{\pi\rho^2}\,(\phi_\rho * q)(X)$$

where $*$ denotes a convolution.

We write the parameters that are associated with the filtered versions of the reference and target patterns with the notation $\hat{(\cdot)}$. Now let $\hat{\lambda}_o$ be the transformation parameter vector corresponding to the projection of $\hat{q}$ onto the transformation manifold $\mathcal{M}(\hat{p})$ of the filtered reference pattern $\hat{p}$

$$\hat{\lambda}_o = \arg\min_{\lambda\in\Lambda} \|\hat{p}_\lambda - \hat{q}\|^2. \tag{5.8}$$

Hence, $\hat{\lambda}_o$ is the optimal transformation parameter vector that aligns $\hat{p}$ with $\hat{q}$. Let $\partial_i\,\hat{p}_\lambda$ and $\hat{\mathcal{G}}_{ij}$ denote respectively the first derivatives and the metric tensor of the manifold $\mathcal{M}(\hat{p})$. From (5.5), the transformation estimate $\hat{\lambda}_e$ obtained with the filtered versions of the reference and target patterns by linearizing the manifold $\mathcal{M}(\hat{p})$ is given by

$$\hat{\lambda}_e = \lambda_r + [\hat{\mathcal{G}}_{ij}(\lambda_r)]^{-1}[\langle \hat{q} - \hat{p}_{\lambda_r}, \partial_i\,\hat{p}_{\lambda_r}\rangle]$$

where λ_r is the reference parameter vector. The alignment error obtained with the smoothed patterns is given as $\|\hat{\lambda}_e - \hat{\lambda}_o\|$, which we are interested in in this study. In particular, we would like to characterize the variation of $\|\hat{\lambda}_e - \hat{\lambda}_o\|$ with the size ρ of the low-pass filter used for smoothing the images in multiscale alignment, and the initial noise level ν of the target image before filtering. We thus examine in Section 5.3.2 the variation of the alignment error with noise and filtering.

5.3 Analysis of Tangent Distance

5.3.1 Upper bound for the alignment error

We now present an upper bound for the error of the alignment computed with the tangent distance method. We can assume that the parameter domain Λ is selected sufficiently large, so that p_{λ_o} is not on the boundary of $\mathcal{M}(p)$. Then, the noise pattern n is orthogonal to the tangent space of $\mathcal{M}(p)$ at p_{λ_o}. In other words, we have

$$\langle n, \partial_i\, p_{\lambda_o}\rangle = 0, \quad \forall i = 1, \cdots, d. \tag{5.9}$$

The deviation of the target image from the transformation manifold model impairs the estimation of transformation parameters. In our analysis of the alignment error, this deviation is characterized by the distance ν between q and $\mathcal{M}(p)$. Then, there is another source of error that causes the

deviation of the estimated parameters λ_e from the optimal ones λ_o. It is related to the nonzero curvature of the manifold, as a result of which $\mathcal{M}(p)$ diverges from its linear approximation $\mathcal{S}_{\lambda_r}(p)$. In the derivation of the component of the alignment error associated with manifold nonlinearity, we make use of a quadratic approximation of the manifold around the reference point p_{λ_r}

$$p_\lambda \approx p_{\lambda_r} + \sum_{i=1}^{d} \partial_i \, p_{\lambda_r}(\lambda^i - \lambda_r^i) + \frac{1}{2} \sum_{i=1}^{d} \sum_{j=1}^{d} \partial_{ij} \, p_{\lambda_r}(\lambda^i - \lambda_r^i)(\lambda^j - \lambda_r^j). \tag{5.10}$$

This approximation is treated as an equality in the derivation of the alignment error bound in Theorem 1. Equation (5.10) shows that the nonlinearity of the manifold can be characterized with an upper bound $\mathcal{K}$ on the norm of the second derivatives of the manifold

$$\mathcal{K} := \max_{i,j=1,\cdots,d} \sup_{\lambda \in \Lambda} \|\partial_{ij} \, p_\lambda\|.$$

Since $\mathcal{K}$ is an upper bound for the norms of the derivatives of tangent vectors, it can be regarded as a uniform curvature bound parameter for $\mathcal{M}(p)$.

We can now state our result that defines an upper bound on the alignment error.

Theorem 1. *The parameter-domain distance between the optimal transformation λ_o and its estimate λ_e given by the tangent distance method can be upper bounded as*

$$\|\lambda_e - \lambda_o\| \leq \; E := \mathcal{K} \, \lambda_{\min}^{-1} \left([\mathcal{G}_{ij}(\lambda_r)]\right) \left(\frac{1}{2} \, d^2 \sqrt{\mathrm{tr}([\mathcal{G}_{ij}(\lambda_r)])} \; \|\lambda_o - \lambda_r\|_\infty^2 + \sqrt{d} \; \nu \; \|\lambda_o - \lambda_r\|_1 \right) \tag{5.11}$$

where $\lambda_{\min}(\cdot)$ and $\mathrm{tr}(.)$ denote respectively the smallest eigenvalue and the trace of a matrix, and the notations $\|\cdot\|_\infty$ and $\|\cdot\|_1$ stand for the ℓ^∞ and ℓ^1-norms in $\mathbb{R}^n$.

Theorem 1 is proved in Appendix B.1. The result is obtained by examining the effects of both the nonlinearity of the manifold and the image noise on the alignment error. The theorem shows that the alignment error augments with the increase in the manifold curvature parameter $\mathcal{K}$ and the noise level ν, as expected. Moreover, another important factor affecting the alignment error is the distance $\|\lambda_o - \lambda_r\|$ between the reference and the optimal transformation parameters. If the reference manifold point p_{λ_r} around which the manifold is linearized is sufficiently close to the true projection of the target image onto the manifold, the tangent distance method is more likely to give a good estimate of the registration parameters.

5.3.2 Alignment error with low-pass filtering

We now analyze the influence of the low-pass filtering of the reference and target patterns on the accuracy of alignment with the tangent distance method as it is the case in multiscale registration algorithms. We consider a setting where the reference pattern p and the target pattern q are low-pass filtered and the transformation parameters are estimated with the smoothed versions of p and q. The purpose of this section is then to analyze the variation of the alignment error bound given in Theorem 1 with respect to the kernel size of the low-pass filter used in smoothing.

We first remark the following. The optimal transformation parameter vector $\hat{\lambda}_o$ corresponding to the smoothed patterns is in general different from the optimal transformation parameter vector λ_o corresponding to the unfiltered patterns p and q. This is due to the fact that both the image noise and the filtering cause a perturbation in the global minimum of the function $f(\lambda) = \|q - p_\lambda\|^2$, which gives the distance between the target pattern q and the transformed versions of the reference pattern p. Note that the overall error in the transformation parameter estimation is $\|\hat{\lambda}_e - \lambda_o\|$ and it can be upper bounded as

$$\|\hat{\lambda}_e - \lambda_o\| \leq \|\hat{\lambda}_e - \hat{\lambda}_o\| + \|\hat{\lambda}_o - \lambda_o\|.$$

Here, the first error term $\|\hat{\lambda}_e - \hat{\lambda}_o\|$ results from the linearization of the manifold, whereas the second error term $\|\hat{\lambda}_o - \lambda_o\|$ is due to the shift in the global minimum of the distance function $f(\lambda)$. In this chapter, we aim to analyze how the linearization of the manifold affects the estimation of the transformation parameters for generic transformation models. Therefore, we focus on the first error term $\|\hat{\lambda}_e - \hat{\lambda}_o\|$ associated particularly with the registration of the images using the tangent distance, and examine its variation with the noise level and the action of smoothing the images. Note that the second error term $\|\hat{\lambda}_o - \lambda_o\|$ depends on the geometric transformation model. For example, in Chapter 6, we examine it for the transformation model of 2-D translations and study its dependence on the noise level and low-pass filtering in details. We remark however that the error term $\|\hat{\lambda}_e - \hat{\lambda}_o\|$ caused by the manifold linearization is in general expected to be dominant over the second error term $\|\hat{\lambda}_o - \lambda_o\|$ unless the reference parameters λ_r are really close to the optimal parameters λ_o.

The filtered target pattern can be decomposed as

$$\hat{q} = \hat{p}_{\hat{\lambda}_o} + \tilde{n}$$

where the noise pattern $\tilde{n}$ is orthogonal to the tangent space $T_{\hat{\lambda}_o}\mathcal{M}(\hat{p})$ at $\hat{p}_{\hat{\lambda}_o}$. Let $\partial_{ij}\,\hat{p}_\lambda$ and $\hat{\mathcal{K}}$ denote the second order derivatives and the curvature bound parameter of the manifold $\mathcal{M}(\hat{p})$. Then, from Theorem 1, the alignment error obtained with the smoothed patterns can be upper bounded as $\|\hat{\lambda}_e - \hat{\lambda}_o\| \leq \hat{E}$, where

$$\hat{E} = \hat{\mathcal{K}}\ \lambda_{\min}^{-1}\ \left([\hat{\mathcal{G}}_{ij}(\lambda_r)]\right) \left(\frac{1}{2}\, d^2 \sqrt{\mathrm{tr}([\hat{\mathcal{G}}_{ij}(\lambda_r)])}\ \|\hat{\lambda}_o - \lambda_r\|_\infty^2 + \sqrt{d}\ \|\tilde{n}\|\ \|\hat{\lambda}_o - \lambda_r\|_1\right). \tag{5.12}$$

In order to analyze the variation of $\hat{E}$ with filtering and noise, we examine the dependence of each term in the expression of $\hat{E}$ in (5.12) on the filter size ρ and the initial noise level ν of the unfiltered target image. The curvature parameter $\hat{\mathcal{K}}$ of the smoothed manifold is given by

$$\hat{\mathcal{K}} = \max_{i,j=1,\cdots,d}\ \sup_{\lambda\in\Lambda} \|\partial_{ij}\,\hat{p}_\lambda\|.$$

Hence, if a uniform estimate can be found for the rate of variation of $\|\partial_{ij}\,\hat{p}_\lambda\|$ with the filter size ρ that is valid for all λ and (i,j), the curvature parameter $\hat{\mathcal{K}}$ will also have the same order of variation with ρ. Next, the metric tensor of the smoothed manifold is given by $\hat{\mathcal{G}}_{ij}(\lambda_r) = \langle \partial_i\,\hat{p}_{\lambda_r}, \partial_j\,\hat{p}_{\lambda_r}\rangle$, and

its trace is

$$\operatorname{tr}\big([\hat{\mathcal{G}}_{ij}(\lambda_r)]\big) = \sum_{i=1}^{d} \|\partial_i\, \hat{p}_{\lambda_r}\|^2.$$

Therefore, if the variation of $\|\partial_i\, \hat{p}_{\lambda_r}\|^2$ with the filter size ρ can be characterized uniformly (in a way that is valid for all λ_r and i), the trace $\operatorname{tr}\big([\hat{\mathcal{G}}_{ij}(\lambda_r)]\big)$ of the metric tensor will also have the same order of variation with ρ as $\|\partial_i\, \hat{p}_{\lambda_r}\|^2$. Since the trace is given by the sum of the eigenvalues, one can reasonably expect the smallest eigenvalue $\lambda_{\min}\ \big([\hat{\mathcal{G}}_{ij}(\lambda_r)]\big)$ to have the same variation with ρ as well. Lastly, the norm $\|\tilde{n}\|$ of the noise component of $\hat{q}$ depends on both the filter size ρ and the initial noise level ν before filtering.

We study now Equation (5.12) in more details and derive first a relation between the norms $\|\partial_i\, \hat{p}_\lambda\|$, $\|\partial_{ij}\, \hat{p}_\lambda\|$ of the first and second-order manifold derivatives and the norms $\|N_\nabla \hat{p}\|$, $\|N_h \hat{p}\|$ of the gradient and Hessian magnitudes of the filtered reference pattern $\hat{p}$. We state the dependences of $\|N_\nabla \hat{p}\|$ and $\|N_h \hat{p}\|$ on the filter size ρ in Lemma 1, which is then used to obtain the variation of the manifold derivatives $\|\partial_i\, \hat{p}_\lambda\|$, $\|\partial_{ij}\, \hat{p}_\lambda\|$ with ρ in Corollary 1. Next, we establish the dependence of the norm $\|\tilde{n}\|$ of the noise component on ρ and ν in Lemma 2. Finally, all of these results are put together in our main result Theorem 2, where we present the rate of variation of the alignment error bound $\hat{E}$ with the filter size ρ and the initial noise level ν of the target image.

Examination of $\|\partial_i\, \hat{p}_\lambda\|$ and $\|\partial_{ij}\, \hat{p}_\lambda\|$

Let us begin with the computation of the terms $\|\partial_i\, \hat{p}_\lambda\|$ and $\|\partial_{ij}\, \hat{p}_\lambda\|$. First, from the relation (5.1), we have

$$p_\lambda(X) = p(X')$$

where $X' = a_\lambda(X)$. Let us denote the transformed coordinates as $X' = [x'\ y']^T$ and write the derivatives of the transformed coordinates with respect to the transformation parameters as

$$\partial_i\, x' = \frac{\partial x'}{\partial \lambda^i}, \qquad \partial_i\, y' = \frac{\partial y'}{\partial \lambda^i}, \qquad \partial_{ij}\, x' = \frac{\partial^2 x'}{\partial \lambda^i\, \partial \lambda^j}, \qquad \partial_{ij}\, y' = \frac{\partial^2 y'}{\partial \lambda^i\, \partial \lambda^j}.$$

Also, let

$$\partial_x\, p(X') = \left.\frac{\partial\, p(X)}{\partial x}\right|_{X=X'}, \qquad \partial_y\, p(X') = \left.\frac{\partial\, p(X)}{\partial y}\right|_{X=X'}$$
$$\partial_{xx}\, p(X') = \left.\frac{\partial^2\, p(X)}{\partial x^2}\right|_{X=X'}, \qquad \partial_{xy}\, p(X') = \left.\frac{\partial^2\, p(X)}{\partial x\, \partial y}\right|_{X=X'}, \qquad \partial_{yy}\, p(X') = \left.\frac{\partial^2\, p(X)}{\partial y^2}\right|_{X=X'}$$

denote the partial derivatives of the reference pattern p evaluated at the point X'. Then, the derivatives of the manifold $\mathcal{M}(p)$ at p_λ are given by

$$\begin{aligned}
\partial_i\, p_\lambda(X) &= \partial_x\, p(X')\partial_i\, x' + \partial_y\, p(X')\partial_i\, y' \\
\partial_{ij}\, p_\lambda(X) &= \partial_{xx}\, p(X')\, \partial_i\, x'\partial_j\, x' + \partial_{xy}\, p(X')\, (\partial_i\, x'\partial_j\, y' + \partial_j\, x'\partial_i\, y') + \partial_{yy}\, p(X')\, \partial_i\, y'\partial_j\, y' \\
&\quad + \partial_x\, p(X')\, \partial_{ij}\, x' + \partial_y\, p(X')\, \partial_{ij}\, y'.
\end{aligned}$$

One can generalize this to the smoothed versions $\hat{p}$ of the reference pattern as

$$\begin{aligned}\partial_i\,\hat{p}_\lambda(X) &= \partial_x\,\hat{p}(X')\partial_i\,x' + \partial_y\,\hat{p}(X')\partial_i\,y' \\ \partial_{ij}\,\hat{p}_\lambda(X) &= \partial_{xx}\,\hat{p}(X')\,\partial_i\,x'\partial_j\,x' + \partial_{xy}\,\hat{p}(X')\,(\partial_i\,x'\partial_j\,y' + \partial_j\,x'\partial_i\,y') + \partial_{yy}\,\hat{p}(X')\,\partial_i\,y'\partial_j\,y' \\ &\quad + \partial_x\,\hat{p}(X')\,\partial_{ij}\,x' + \partial_y\,\hat{p}(X')\,\partial_{ij}\,y'.\end{aligned} \tag{5.13}$$

Notice that, in the above equations, the filtering applied on the reference pattern influences only the spatial derivatives of the reference pattern ($\partial_x\,\hat{p}$, $\partial_y\,\hat{p}$, $\partial_{xx}\,\hat{p}$, $\partial_{xy}\,\hat{p}$, $\partial_{yy}\,\hat{p}$), whereas the derivatives of the transformed coordinates ($\partial_i\,x'$, $\partial_i\,y'$, $\partial_{ij}\,x'$, $\partial_{ij}\,y'$) depend solely on the transformation model λ and are constant with respect to the filter size ρ. Therefore, the variation of $\|\partial_i\,\hat{p}_\lambda\|$ and $\|\partial_{ij}\,\hat{p}_\lambda\|$ with ρ is mostly determined by the variation of the spatial derivatives of the pattern with the filter size. We denote the gradient of $\hat{p}$ as

$$\nabla\hat{p}(X) = [\partial_x\,\hat{p}(X)\;\partial_y\,\hat{p}(X)]^T$$

and the vectorized Hessian of $\hat{p}$ as

$$(h\hat{p})(X) = [\partial_{xx}\,\hat{p}(X)\;\partial_{xy}\,\hat{p}(X)\;\partial_{xy}\,\hat{p}(X)\;\partial_{yy}\,\hat{p}(X)]^T. \tag{5.14}$$

We then define the functions $N_\nabla\hat{p},\; N_h\hat{p}:\; \mathbb{R}^2 \to \mathbb{R}$

$$N_\nabla\hat{p}\,(X) = \|\nabla\hat{p}(X)\|, \qquad\qquad N_h\hat{p}\,(X) = \|(h\hat{p})(X)\|$$

which give the ℓ^2-norms of the gradient and the Hessian of $\hat{p}$ at X. Since we assume that the spatial derivatives of the pattern are square-integrable, the functions $N_\nabla\hat{p}$ and $N_h\hat{p}$ are in $L^2(\mathbb{R}^2)$. The equations in (5.13) show that the first derivatives of the manifold are proportional to the first derivatives of the pattern; and the second derivatives of the manifold depend linearly on both the first and the second derivatives of the pattern. One thus expects the L^2-norms of the manifold derivatives to be related to the L^2-norms of $N_\nabla\hat{p}$ and $N_h\hat{p}$ as

$$\begin{aligned}\|\partial_i\,\hat{p}_\lambda\| &= O\left(\|N_\nabla\hat{p}\|\right) \\ \|\partial_{ij}\,\hat{p}_\lambda\| &= O\left(\|N_\nabla\hat{p}\| + \|N_h\hat{p}\|\right)\end{aligned} \tag{5.15}$$

from the perspective of their dependence on the filter size ρ. These relations indeed hold and they are formally shown in Appendix B.2.

Since we have established the connection between the manifold derivatives and the pattern spatial derivatives, it suffices now to determine how the spatial derivatives $\|N_\nabla\hat{p}\|$ and $\|N_h\hat{p}\|$ depend on the filter size ρ. In order to examine this, we adopt a parametric representation of the reference pattern p in the analytic dictionary $\mathcal{D}$ defined in (2.10), where we consider the Gaussian function $\phi(X) = e^{-X^TX}$ as the mother function in building the dictionary. Since the linear span of this dictionary is dense in $L^2(\mathbb{R}^2)$, any pattern $p \in L^2(\mathbb{R}^2)$ can be represented as the linear combination of a sequence of atoms in $\mathcal{D}$. In the rest of our analysis, we adopt a representation of p in $\mathcal{D}$

$$p(X) = \sum_{k=1}^{\infty} c_k\,\phi_{\gamma_k}(X) \tag{5.16}$$

where γ_k are the atom parameters and c_k are the atom coefficients. Our derivation of the variations of $\|N_\nabla \hat{p}\|$ and $\|N_h \hat{p}\|$ is based on the representation of patterns with Gaussian atoms. Nevertheless, the conclusions of our analysis are general and valid for all reference patterns in $L^2(\mathbb{R}^2)$ since any square-integrable pattern can be represented in the above form (5.16).

Now, applying the Gaussian filter in (5.6) on the reference pattern in (5.16), we obtain the filtered pattern as

$$\frac{1}{\pi\rho^2}\,(\phi_\rho * p)(X) = \frac{1}{\pi\rho^2}\sum_{k=1}^{\infty} c_k\,(\phi_\rho * \phi_{\gamma_k})(X)$$

from the linearity of the convolution operator. In order to evaluate the convolution of two Gaussian atoms, we use the following proposition [99].

Proposition 4. *Let* $\phi_{\gamma_1}(X) = \phi(\sigma_1^{-1}\,\Psi_1^{-1}\,(X-\tau_1))$ *and* $\phi_{\gamma_2}(X) = \phi(\sigma_2^{-1}\,\Psi_2^{-1}\,(X-\tau_2))$. *Then*

$$(\phi_{\gamma_1} * \phi_{\gamma_2})(X) = \frac{\pi|\sigma_1\sigma_2|}{|\sigma_3|}\phi_{\gamma_3}(X) \tag{5.17}$$

where

$$\phi_{\gamma_3}(X) = \phi(\sigma_3^{-1}\,\Psi_3^{-1}\,(X-\tau_3))$$

and the parameters of ϕ_{γ_3} *are given by*

$$\tau_3 = \tau_1 + \tau_2, \qquad \Psi_3\,\sigma_3^2\,\Psi_3^{-1} = \Psi_1\,\sigma_1^2\,\Psi_1^{-1} + \Psi_2\,\sigma_2^2\,\Psi_2^{-1}.$$

Proposition 4 implies that, when an atom ϕ_{γ_k} of p is convolved with the Gaussian kernel, it becomes

$$\frac{1}{\pi\rho^2}\,(\phi_\rho * \phi_{\gamma_k})(X) = \frac{|\sigma_k|}{|\hat{\sigma}_k|}\phi_{\hat{\gamma}_k}(X) \tag{5.18}$$

where $\phi_{\hat{\gamma}_k}(X) = \phi(\hat{\sigma}_k^{-1}\,\hat{\Psi}_k^{-1}\,(X-\hat{\tau}_k))$ and

$$\hat{\tau}_k = \tau_k, \quad \hat{\Psi}_k = \Psi_k, \quad \hat{\sigma}_k = \sqrt{\Upsilon^2 + \sigma_k^2}. \tag{5.19}$$

Hence, when p is smoothed with a Gaussian filter, the atom $\phi_{\gamma_k}(X)$ with coefficient c_k is replaced by the smoothed atom $\phi_{\hat{\gamma}_k}(X)$ with coefficient

$$\hat{c}_k = \frac{|\sigma_k|}{|\hat{\sigma}_k|}c_k = \frac{|\sigma_k|}{\sqrt{|\Upsilon^2 + \sigma_k^2|}}c_k = \frac{\sigma_{x,k}\,\sigma_{y,k}}{\sqrt{(\rho^2+\sigma_{x,k}^2)(\rho^2+\sigma_{y,k}^2)}}c_k \tag{5.20}$$

where $\sigma_k = \mathrm{diag}(\sigma_{x,k},\,\sigma_{y,k})$. This shows that the change in the pattern parameters due to filtering can be captured by substituting the scale parameters σ_k with $\hat{\sigma}_k$ and replacing the coefficients c_k with $\hat{c}_k$. Then, the smoothed pattern $\hat{p}$ has the following representation in the dictionary $\mathcal{D}$

$$\hat{p}(X) = \sum_{k=1}^{\infty}\hat{c}_k\,\phi_{\hat{\gamma}_k}(X). \tag{5.21}$$

One can observe from (5.20) that the atom coefficients $\hat{c}_k$ of the filtered pattern $\hat{p}$ change with the filter size ρ at a rate

$$\hat{c}_k = O((1+\rho^2)^{-1}). \tag{5.22}$$

Also, from (5.19), the atom scale parameters of $\hat{p}$ are given by

$$\hat{\sigma}_{x,k} = \sqrt{\sigma_{x,k}^2 + \rho^2}, \qquad \hat{\sigma}_{y,k} = \sqrt{\sigma_{y,k}^2 + \rho^2} \tag{5.23}$$

which have the rate of increase

$$\hat{\sigma}_{x,k},\ \hat{\sigma}_{y,k} = O((1+\rho^2)^{1/2}) \tag{5.24}$$

with the filter size ρ.

We are now equipped with the necessary tools for examining the variations of $\|N_\nabla \hat{p}\|$ and $\|N_h \hat{p}\|$ with the filter size ρ. We state these in the following lemma.

Lemma 1. *The norms $\|N_\nabla \hat{p}\|$ and $\|N_h \hat{p}\|$ of the gradient and Hessian magnitudes decrease with the filter size ρ at the following rates*

$$\|N_\nabla \hat{p}\| = O((1+\rho^2)^{-1})$$
$$\|N_h \hat{p}\| = O((1+\rho^2)^{-3/2}).$$

The proof of Lemma 1 is given in Appendix B.3. The above dependences are shown by deriving approximations of $\|N_\nabla \hat{p}\|$ and $\|N_h \hat{p}\|$ in terms of the atom parameters $\{\gamma_k\}$ and coefficients $\{c_k\}$. Their variations with the filter size ρ are then determined by building on the relations (5.24) and (5.22). The lemma not only confirms the intuition that the norms of the pattern gradient and Hessian should decrease with filtering, but also provides expressions for their rate of decrease with the filter size ρ.

An immediate consequence of Lemma 1 is the following.

Corollary 1. *The norms $\|\partial_i \hat{p}_\lambda\|$, $\|\partial_{ij} \hat{p}_\lambda\|$ of the first and second-order manifold derivatives decrease with the filter size ρ at the following rates*

$$\|\partial_i \hat{p}_\lambda\| = O((1+\rho^2)^{-1})$$
$$\|\partial_{ij} \hat{p}_\lambda\| = O\left((1+\rho^2)^{-3/2} + (1+\rho^2)^{-1}\right).$$

Proof: The corollary follows directly from Lemma 1 and the relation between the manifold derivatives and the pattern derivatives given in (5.15). □

Note that for large values of ρ, the second additive term of $O(1+\rho^2)^{-1}$ in $\|\partial_{ij} \hat{p}_\lambda\|$ dominates the first term of $O(1+\rho^2)^{-3/2}$, therefore $\|\partial_{ij} \hat{p}_\lambda\| = O((1+\rho^2)^{-1})$ for large ρ. However, we keep both additive terms in $\|\partial_{ij} \hat{p}_\lambda\|$ as we will see that the first term is important for characterizing the behavior of the alignment error bound for small values of the filter size. Corollary 1 will be helpful for determining the dependences of the curvature bound $\hat{\mathcal{K}}$ and the parameters related to the metric tensor $\hat{\mathcal{G}}_{ij}$ on the filter size in our main result Theorem 2.

Examination of $\|\tilde{n}\|$

We examine now the variation of $\|\tilde{n}\|$ with both the filter size ρ and the initial noise level ν of the unfiltered target pattern. In the following lemma, we summarize the dependence of the noise level $\|\tilde{n}\|$ of the filtered target pattern on ν and ρ.

Lemma 2. *The distance $\|\tilde{n}\|$ between the filtered target pattern $\hat{q}$ and the transformation manifold $\mathcal{M}(\hat{p})$ of the filtered reference pattern $\hat{p}$ has a rate of variation of*

$$\|\tilde{n}\| = O\left((\nu+1)(1+\rho^2)^{-1/2}\right)$$

with the filter size ρ and the initial noise level ν for geometric transformation models that allow the change of the scale of the pattern p. The variation of $\|\tilde{n}\|$ is however given by

$$\|\tilde{n}\| = O\left(\nu(1+\rho^2)^{-1/2}\right)$$

if the geometric transformation model does not include a scale change.

The proof of Lemma 2 is given in Appendix B.4. The presented dependences are obtained by deriving a relation between the norm of the noise component $\tilde{n} = \hat{q} - \hat{p}_{\hat{\lambda}_o}$ and the filtered version $\hat{n}$ of the initial noise component $n = q - p_{\lambda_o}$. The lemma states that $\|\tilde{n}\|$ decreases with the filter size ρ at a rate of $O\left((1+\rho^2)^{-1/2}\right)$. Meanwhile, its dependence on the initial noise level ν differs slightly between transformation models that include a scale change or not. The noise term $\|\tilde{n}\|$ increases at a rate of $O(\nu)$ for transformations without a scale change; however, transformations with a scale change introduce an offset to the initial noise level to yield a variation of $O(\nu+1)$. This is due to the following reason. The initial noise level before filtering is given by the norm of $n = q - p_{\lambda_o}$, where $p_{\lambda_o} \in \mathcal{M}(p)$. Meanwhile, when the transformation model λ includes a scale change, the actions of filtering and transforming a pattern do not commute, and the filtered version $\widehat{p_{\lambda_o}}$ of p_{λ_o} does not lie on the transformation manifold $\mathcal{M}(\hat{p})$ of the filtered reference pattern $\hat{p}$ (see Appendix B.4 for more details). The "lifting" of the base point $\widehat{p_{\lambda_o}}$ of $\hat{q}$ (remember the decomposition $\hat{q} = \widehat{p_{\lambda_o}} + \hat{n}$) from the manifold $\mathcal{M}(\hat{p})$ further increases the distance of $\hat{q}$ to $\mathcal{M}(\hat{p})$, in addition to the deviation $\hat{n}$. The overall noise level in case of filtering is therefore larger than the norm of the filtered version $\hat{n}$ of n. Note that for transformations involving a scale change, even if the initial noise level ν is zero, which means $q \in \mathcal{M}(p)$, we have $\hat{q} \notin \mathcal{M}(\hat{p})$ after filtering. This creates a source of noise when the filtered versions of the image pair are used in the alignment.

Examination of $\hat{E}$

We are now ready to present our main result, which states the dependence of the alignment error $\hat{E}$ on the initial noise level of the target pattern and the filter size.

Theorem 2. *The alignment error bound $\hat{E}$ obtained when the smoothed image pair is aligned with the tangent distance method is given by*

$$\hat{E} = \hat{E}_1 + \hat{E}_2$$

where the error component $\hat{E}_1$ resulting from manifold nonlinearity decreases at rate

$$\hat{E}_1 = O\left(1 + (1+\rho^2)^{-1/2}\right)$$

with the size ρ of the low-pass filter kernel used for smoothing the reference and target images. The second component $\hat{E}_2$ of the alignment error associated with image noise has the variation

$$\hat{E}_2 = O\left((\nu+1)\,(1+\rho^2)^{1/2}\right)$$

with the filter size ρ and the noise level ν if the geometric transformation model includes a scale change. The variation of $\hat{E}_2$ with ρ and ν is

$$\hat{E}_2 = O\left(\nu\,(1+\rho^2)^{1/2}\right)$$

if the geometric transformation model does not change the scale of the pattern.

Proof: Remember from (5.12) that the alignment error bound is given by

$$\hat{E} = \hat{E}_1 + \hat{E}_2$$

where the error terms

$$\begin{aligned} \hat{E}_1 &= \frac{1}{2}\, d^2\, \hat{\mathcal{K}}\; \lambda_{\min}^{-1}\left([\hat{\mathcal{G}}_{ij}(\lambda_r)]\right)\sqrt{\mathrm{tr}([\hat{\mathcal{G}}_{ij}(\lambda_r)])}\; \|\hat{\lambda}_o - \lambda_r\|_\infty^2 \\ \hat{E}_2 &= \sqrt{d}\; \hat{\mathcal{K}}\; \lambda_{\min}^{-1}\left([\hat{\mathcal{G}}_{ij}(\lambda_r)]\right)\; \|\tilde{n}\|\; \|\hat{\lambda}_o - \lambda_r\|_1 \end{aligned} \tag{5.25}$$

are associated respectively with the nonzero manifold curvature (lifting of the manifold from the tangent space) and the noise on the target image. Also, remember that the variation of $\hat{\mathcal{K}}$ with ρ is the same as that of $\|\partial_{ij}\,\hat{p}_\lambda\|$, and that $\lambda_{\min}\left([\hat{\mathcal{G}}_{ij}(\lambda_r)]\right)$ and $\mathrm{tr}([\hat{\mathcal{G}}_{ij}(\lambda_r)])$ have the same variation with ρ as $\|\partial_i\,\hat{p}_{\lambda_r}\|^2$. Hence, using Corollary 1, we obtain

$$\hat{\mathcal{K}}\; \lambda_{\min}^{-1}\left([\hat{\mathcal{G}}_{ij}(\lambda_r)]\right) = O\left(1 + (1+\rho^2)^{-1/2}\right) O(1+\rho^2) \tag{5.26}$$

$$\sqrt{\mathrm{tr}([\hat{\mathcal{G}}_{ij}(\lambda_r)])} = O\left((1+\rho^2)^{-1}\right) \tag{5.27}$$

which gives

$$\hat{E}_1 = O\left(1 + (1+\rho^2)^{-1/2}\right).$$

Then, from Lemma 2 and Equation (5.26), we determine the variation of $\hat{E}_2$ as

$$\hat{E}_2 = O\left((\nu+1)\,(1+\rho^2)^{1/2}\right) O\left(1 + (1+\rho^2)^{-1/2}\right) \approx O\left((\nu+1)\,(1+\rho^2)^{1/2}\right)$$

for transformations involving a scale change, and as

$$\hat{E}_2 = O\left(\nu\,(1+\rho^2)^{1/2}\right) O\left(1 + (1+\rho^2)^{-1/2}\right) \approx O\left(\nu\,(1+\rho^2)^{1/2}\right)$$

for transformations without a scale change, which finishes the proof of the theorem. □

Theorem 2 can be interpreted as follows. The first error component $\hat{E}_1$ related to manifold nonlinearity is of $O\left(1+\,(1+\rho^2)^{-1/2}\right)$. Since filtering the patterns makes the manifold smoother and decreases the manifold curvature, it improves the accuracy of the first-order approximation of the manifold used in tangent distance. Therefore, the first component of the alignment error decreases with the filter size ρ. Then, we observe that the second error component $\hat{E}_2 = O\left((\nu+1)\,(1+\rho^2)^{1/2}\right)$ resulting from image noise, is proportional to the noise level as expected, but also increases with the filter size ρ. The increase of the error with smoothing is due to the fact that filtering has the undesired effect of amplifying the alignment error caused by noise. This result is in line with our study in Chapter 6 and previous works such as [27], [36] examining Crámer-Rao lower bounds in image registration, which are discussed in more detail in Section 5.5.

The dependence of the overall alignment error on the filter size can be interpreted as follows. For reasonably small values of the image noise level, the overall error $\hat{E}$ first decreases with the filter size ρ at small filter sizes due to the decrease in the first term $\hat{E}_1$, since filtering improves the manifold linearity. As one keeps increasing the filter size, the first error term $\hat{E}_1 = O\left(1+\,(1+\rho^2)^{-1/2}\right)$ gradually decreases and finally converges to a constant value . After that, the second error term $\hat{E}_2$ takes over and the overall alignment error $\hat{E}$ starts to increase with the filter size. The amplification of the registration error resulting from image noise then becomes the prominent factor that determines the overall dependence of the error on the filter size. As the alignment error first decreases and then increases with filtering, there exists an optimal value of the filter size ρ for a given noise level ν. In the noiseless case where $\nu = 0$, our result shows that applying a big filter is favorable as it flattens the manifold, provided that the transformation model does not involve a scale change. Meanwhile, for geometric transformations involving a scale change, there exists a nontrivial optimal filter size even in the noiseless case $\nu = 0$. In this case, the non-commutativity of filtering and pattern transformation processes creates a secondary source of error that is an increasing function of the filter size.

Remark. In hierarchical image registration, in the early stages of alignment where the distance $\|\hat{\lambda}_o - \lambda_r\|$ between the reference and optimal parameters is relatively large, the image pair is smoothed with big filters. Then, in the progressive refinement of the transformation estimates, the reference parameter vector λ_r of each stage is taken as the estimate $\hat{\lambda}_e$ of the previous stage, while the filter size is decreased gradually at the same time [2], [100]. We now interpret this strategy in the light of Theorem 2 by deriving the optimal filter size that minimizes the alignment error in terms of the distance between the optimal and reference transformation parameters.

Observe that, from the expressions of the error components $\hat{E}_1$ and $\hat{E}_2$ in (5.25) and the variations of $\hat{E}_1$ and $\hat{E}_2$ with ν and ρ derived in Theorem 2, the alignment error is roughly given by

$$\hat{E} \approx \left(1+\,(1+\rho^2)^{-1/2}\right)\,\|\hat{\lambda}_o - \lambda_r\|^2 + (\nu+1)\,(1+\rho^2)^{1/2}\,\|\hat{\lambda}_o - \lambda_r\|. \tag{5.28}$$

In this approximate expression, we ignore the constants. We rely on the equivalence of norms and replace the ℓ^∞ and ℓ^1-norms of $\hat{\lambda}_o - \lambda_r$ by its ℓ^2-norm. Then, the optimal filter size ρ_{opt} minimizing $\hat{E}$ satisfies

$$1+\rho_{opt}^2 = \frac{\|\hat{\lambda}_o - \lambda_r\|}{\nu+1} \tag{5.29}$$

so that the dependence of the optimal filter size on the distance $\|\hat{\lambda}_o - \lambda_r\|$ and the noise level ν is given by

$$\rho_{opt} \approx O\left(\sqrt{\frac{\|\hat{\lambda}_o - \lambda_r\|}{\nu + 1}}\right). \tag{5.30}$$

Hence, the filter size should be chosen proportionally to the square root of the distance $\|\hat{\lambda}_o - \lambda_r\|$ between the reference transformation parameters and the optimal ones in each stage of hierarchical registration. This provides a justification of the strategy of reducing the filter size gradually in coarse-to-fine alignment, since the estimate $\hat{\lambda}_e$, which is used as the reference parameter vector λ_r in the next stage, is expected to approach the optimal solution progressively, i.e., the distance $\|\hat{\lambda}_o - \lambda_r\|$ decreases throughout the hierarchical alignment process. Another observation is that the noise level of the target image also influences the optimal filter size. The filter size must be chosen inversely proportional to the square root $\sqrt{\nu}$ of the noise level, which is due to the increase in the alignment error with filtering in the presence of noise.

5.4 Experimental Results

We now present experimental results that illustrate our alignment error bounds. In all settings, we experiment on three different geometric transformation models, namely a two-dimensional translation manifold

$$\mathcal{M}(p) = \{A_\lambda(p) : \lambda = (t_x, t_y) \in \Lambda\}, \tag{5.31}$$

a three-dimensional manifold given by the translations and rotations of a reference pattern

$$\mathcal{M}(p) = \{A_\lambda(p) : \lambda = (\overline{\theta}, t_x, t_y) \in \Lambda\}, \tag{5.32}$$

and a four-dimensional manifold generated by the translations, rotations and isotropic scalings of a reference pattern

$$\mathcal{M}(p) = \{A_\lambda(p) : \lambda = (\overline{\theta}, t_x, t_y, \overline{s}) \in \Lambda\}. \tag{5.33}$$

In the above models, t_x and t_y represent translations in x and y directions, $\overline{\theta}$ denotes a rotation parameter, and $\overline{s}$ is a scale change parameter. The parameters $\overline{\theta}$ and $\overline{s}$ are normalized versions of the actual rotation angle θ and scale change factor s, so that the magnitudes of the manifold derivatives with respect to t_x, t_y, $\overline{\theta}$, and $\overline{s}$ are proportional.

In all experiments, several target patterns are generated from a reference pattern by applying a random geometric transformation according to the above models. The target patterns are then corrupted with additive noise patterns at different noise levels ν. For each reference and target pattern pair (p, q), a sequence of image pairs $(\hat{p}, \hat{q})$ are obtained by smoothing p and q with low-pass filters having a range of kernel size ρ. Then, the target pattern $\hat{q}$ in each image pair is aligned with the reference pattern $\hat{p}$ using the tangent distance method, where the reference parameter vector λ_r is taken as identity such that $\hat{p}_{\lambda_r} = \hat{p}$. The experimental alignment error is measured as the parameter domain distance $\|\hat{\lambda}_e - \hat{\lambda}_o\|$ between the optimal transformation parameter vector $\hat{\lambda}_o$ and its estimate $\hat{\lambda}_e$. Then, the experimental alignment error is compared to its theoretical upper bound $\hat{E}$ given in Theorem 1. The curvature parameter $\mathcal{K}$ is computed numerically in the

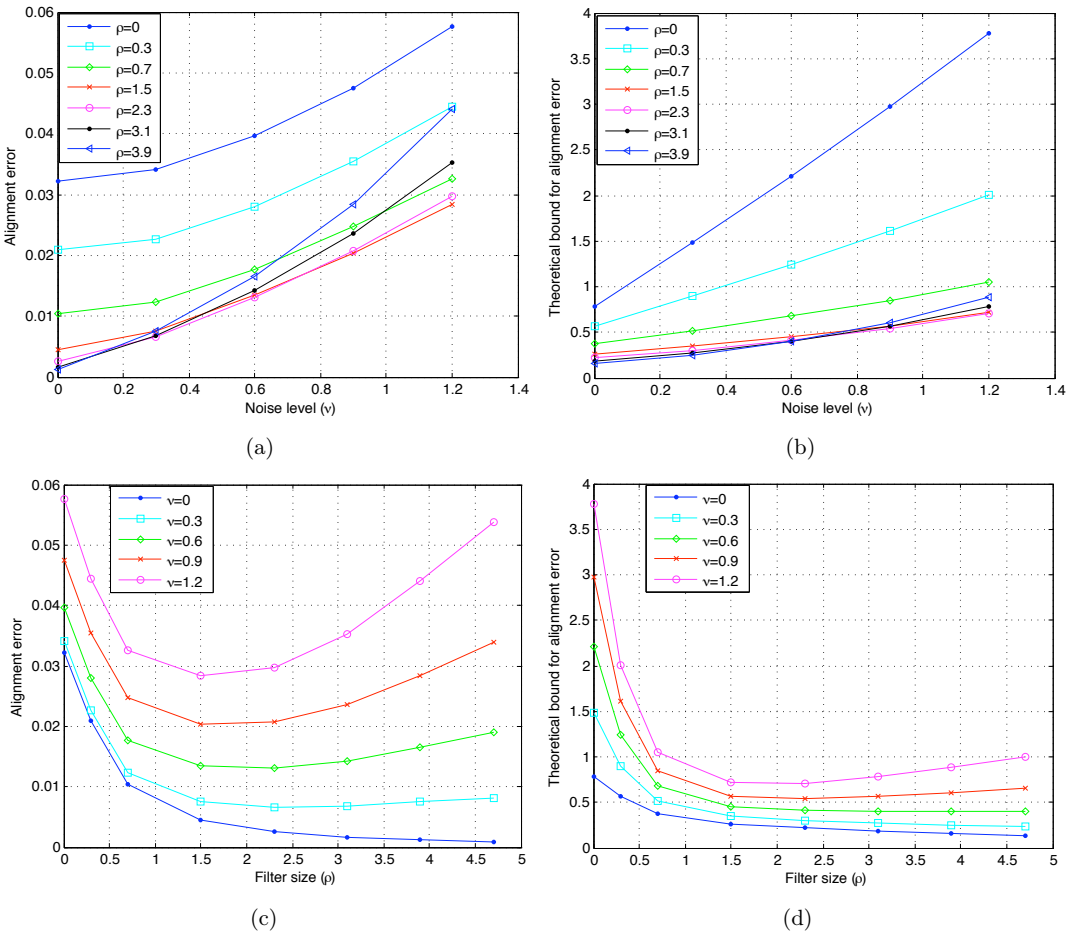


Figure 5.2: Alignment errors of random patterns for 2-D manifolds generated by translations.

implementation of the theorem.

In the first set of experiments, we experiment on 50 different reference patterns that consist of 20 atoms randomly selected from the Gaussian dictionary $\mathcal{D}$. The atom parameters are randomly drawn from the intervals $\psi \in [-\pi, \pi)$; $\tau_x, \tau_y \in [-4, 4]$; $\sigma_x, \sigma_y \in [0.3, 2.3]$; and the atom coefficients are randomly selected within the range $[-1, 1]$. Then, for each one of the models (5.31)-(5.33), 10 target patterns are generated for each reference pattern. The transformation parameters of target patterns are selected randomly within the ranges $\overline{\theta} \in [-0.4, 0.4]$; $t_x, t_y \in [-0.4, 0.4]$; and $\overline{s} \in [0.4, 1.6]$. The above ranges for the normalized rotation and scale parameters $\overline{\theta}$ and $\overline{s}$ correspond to the actual rotation angles $\theta \in [-0.04\pi, 0.04\pi]$ and scale change factors $s \in [0.87, 1.13]$. Each target pattern is corrupted with a different realization of a noise pattern that consists of 100 small-scale Gaussian atoms with random coefficients drawn from a normal distribution, which demonstrates a random noise pattern in the continuous domain. The noise patterns are normalized

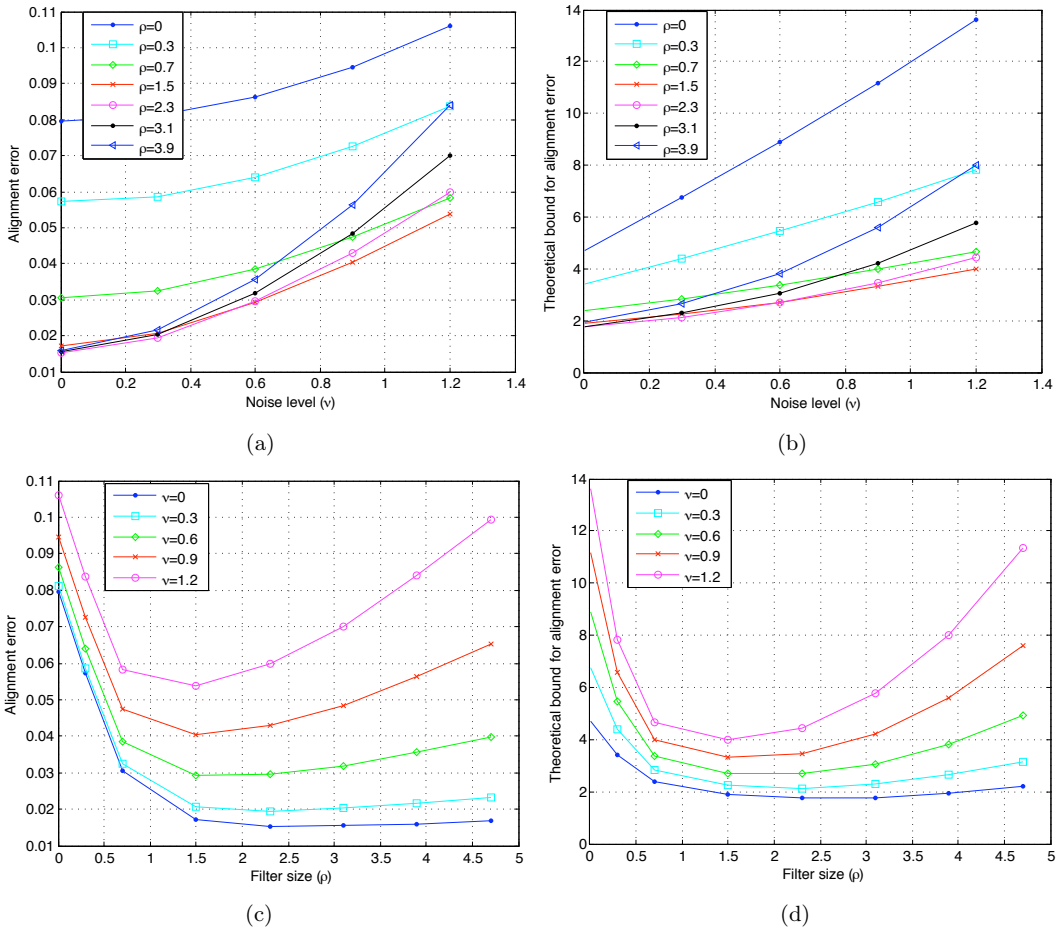


Figure 5.3: Alignment errors of random patterns for 3-D manifolds generated by translations and rotations.

to match a range of noise levels ν.

The results obtained for the transformation models (5.31), (5.32), and (5.33) are presented respectively in Figures 5.2, 5.3 and 5.4, where the performance is averaged over all reference and target patterns. In all figures, the experimental alignment errors and their theoretical upper bounds are plotted with respect to the noise level ν in panels (a) and (b), where the noise level ν is normalized with the norm $\|p\|$ of the reference pattern. The same experimental errors and theoretical bounds are plotted as functions of the filter size ρ in panels (c) and (d) of all figures.

The results of this experiment can be interpreted as follows. First, the plots in panels (a) and (b) of Figures 5.2-5.4 show that the variation of the alignment error with the noise level ν generally follows approximately a linear rate both in the empirical and the theoretical plots. This confirms the estimations $\hat{E} = O(\nu)$, $\hat{E} = O(\nu+1)$ of Theorem 2. Next, the plots in (c) and (d) of the figures show that the actual alignment error and its theoretical upper bound decrease with filtering at

small filter sizes ρ, as smoothing decreases the nonlinearity of the manifold. The error then begins to increase with the filter size ρ at larger values of ρ in the presence of noise. This confirms that the filter size has an optimal value when the target image is noisy, as predicted by Theorem 2. The shift in the optimal value of the filter size with the increase in the noise level is observable especially in Figures 5.2 and 5.3, which is in agreement with the approximate relation between ρ_{opt} and ν given in (5.30). Moreover, in most plots, the optimal value of the filter size that minimizes the theoretical upper bound in (d) is seen to be in the vicinity of the optimal filter size minimizing the actual alignment error in (c), which shows that the theoretical bound provides a good prediction of suitable filter sizes in alignment. The results also show that the variation of the alignment error with the filter size matches the approximately linear rate $\hat{E} = O\left((1+\rho^2)^{1/2}\right) \approx O(\rho)$ at large filter sizes in most plots.

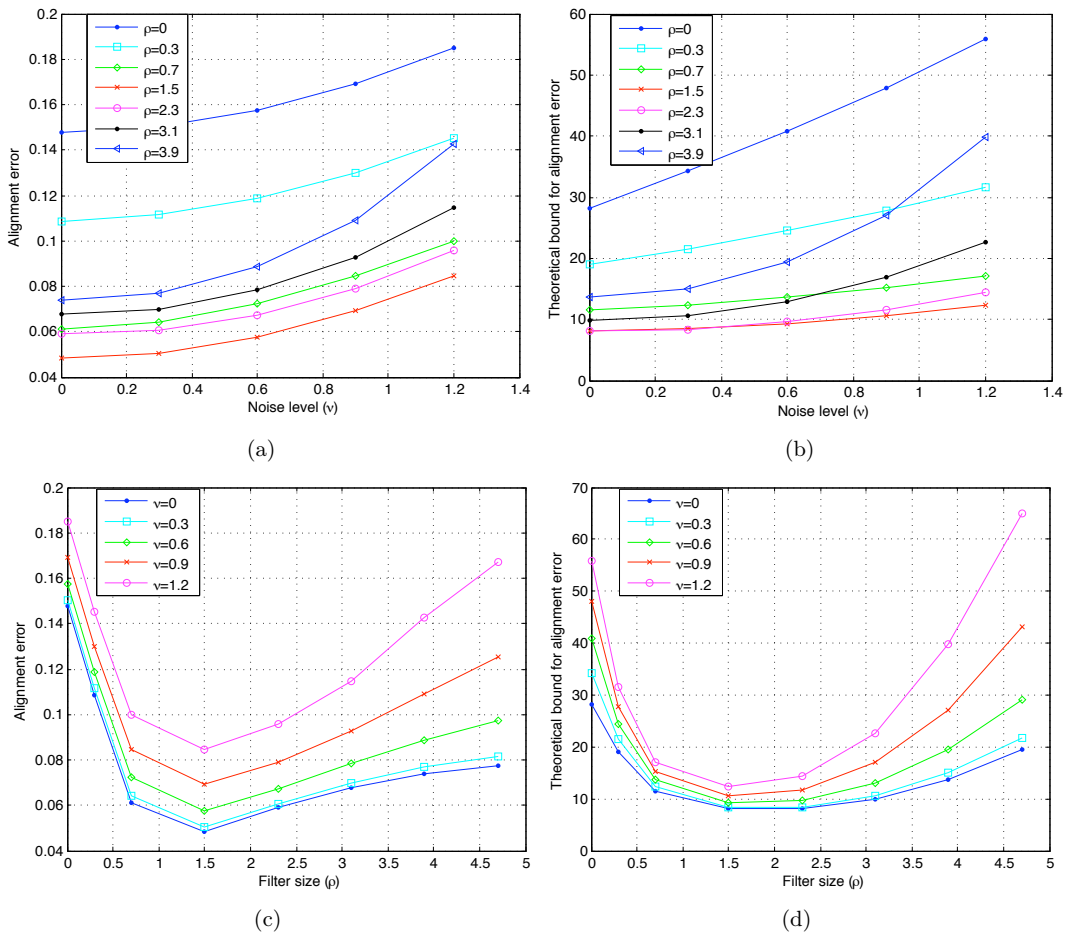


Figure 5.4: Alignment errors of random patterns for 4-D manifolds generated by translations, rotations, and scale changes.

It is also interesting to compare the behavior of the alignment error between different transformation models. To begin with, one can observe in Figures 5.2(c) and 5.2(d) that, for two-dimensional translation manifolds, the alignment error and its theoretical bound asymptotically approach 0 when the filter size ρ increases in the noiseless setting $\nu = 0$. The monotonic decay of the error with filtering is expected since Theorem 2 predicts a variation of $\hat{E} = O\left(1 + (1+\rho^2)^{-1/2}\right)$ for the noiseless case. Meanwhile, the convergence of the error to 0 for the specific transformation model of translations can be explained as follows. In this special case, the variation of the second derivatives of the manifold with the filter size is given by $\|\partial_{ij}\,\hat{p}_\lambda\| = O\left((1+\rho^2)^{-3/2}\right)$, which follows from the fact that the second derivatives of the transformed coordinates in (5.13) vanish; i.e., $\partial_{ij}\,x', \partial_{ij}\,y' = 0$. This gives the rate of decrease of the alignment error with ρ as $\hat{E} = O\left((1+\rho^2)^{-1/2}\right)$ for translation manifolds in the noiseless case. Therefore, the alignment error approaches 0 as ρ increases.

Next, Figures 5.3(c) and 5.3(d) obtained with three-dimensional manifolds generated by translations and rotations show that the experimental and theoretical alignment errors approach a nonzero value in the noiseless case $\nu = 0$ as suggested by the prediction $\hat{E} = O(1 + \left(1+\rho^2\right)^{-1/2})$. However, there is a slight increase in the error at large values of the filter size ρ. This can be explained as follows. In the derivations of $\|\partial_i\,\hat{p}_\lambda\|$ and $\|\partial_{ij}\,\hat{p}_\lambda\|$ in Appendix B.2, we have defined a support region Ω outside which the intensities of the pattern and its derivatives are insignificant, and ignored the dependence of this region on the filter size by assuming that a sufficiently large Ω is selected with respect to the largest realistic value of the filter size. Meanwhile, since a tightly selected support region should expand with the filter size, the behavior of the exact value of the alignment error may deviate slightly from the theoretical prediction for the noiseless case $\hat{E} = O(1 + \left(1+\rho^2\right)^{-1/2})$, which is obtained under the assumption that Ω is fixed. This deviation is especially observable for transformation models where the transformed coordinates x', y' have large derivatives close to the support boundary, where the expansion of Ω with filtering gets important. This is indeed the case for image rotations. However, the plots in Figure 5.3 show that this effect remains negligible. Lastly, we comment on the plots in Figure 5.4 obtained for four-dimensional transformation manifolds generated by translations, rotations, and isotropic scale changes. One can observe in Figures 5.4(c) and 5.4(d) that both the experimental alignment error and its theoretical upper bound increase significantly with the filter size ρ in the noiseless case $\nu = 0$ when transformations include scale changes. This is due to the secondary source of noise demonstrated in Lemma 2. Theorem 2 suggests that the error increases with filtering at a rate $\hat{E} = O\left((\nu+1)(1+\rho^2)^{1/2}\right)$ at large values of ρ, which corresponds to a variation $\hat{E} = O\left((1+\rho^2)^{1/2}\right)$ in the noiseless case.

We perform a second set of experiments on five real images, which are shown in Figure 5.5. The images are resized to the resolution of 60×60 pixels, and for each image an analytical approximation in the Gaussian dictionary $\mathcal{D}$ is computed with 100 atoms. The dictionary is defined over the parameter domain $\psi \in [-\pi, \pi)$; $\tau_x, \tau_y \in [-6, 6]$; $\sigma_x, \sigma_y \in [0.05, 3.5]$. Two reference patterns are considered for each image; namely, the digital image itself, and its analytical approximation in $\mathcal{D}$. For each one of the transformation models (5.31)-(5.33), 40 test patterns are generated for each reference pattern by applying a geometric transformation and adding with a digital Gaussian noise image that is i.i.d. for each pixel. The geometric transformations are randomly selected from the transformation parameter domain $\overline{\theta} \in [-0.6, 0.6]$; $t_x, t_y \in [-0.6, 0.6]$; $\overline{s} \in [0.1, 2.1]$. The normalized rotation and scale parameters $\overline{\theta}$ and $\overline{s}$ correspond to the actual rotation angle and scale change factors $\theta \in [-0.07\pi, 0.07\pi]$ and $s \in [0.89, 1.13]$. The experimental alignment errors $\|\hat{\lambda}_e - \hat{\lambda}_o\|$ are

Figure 5.5: Images used in the second set of experiments

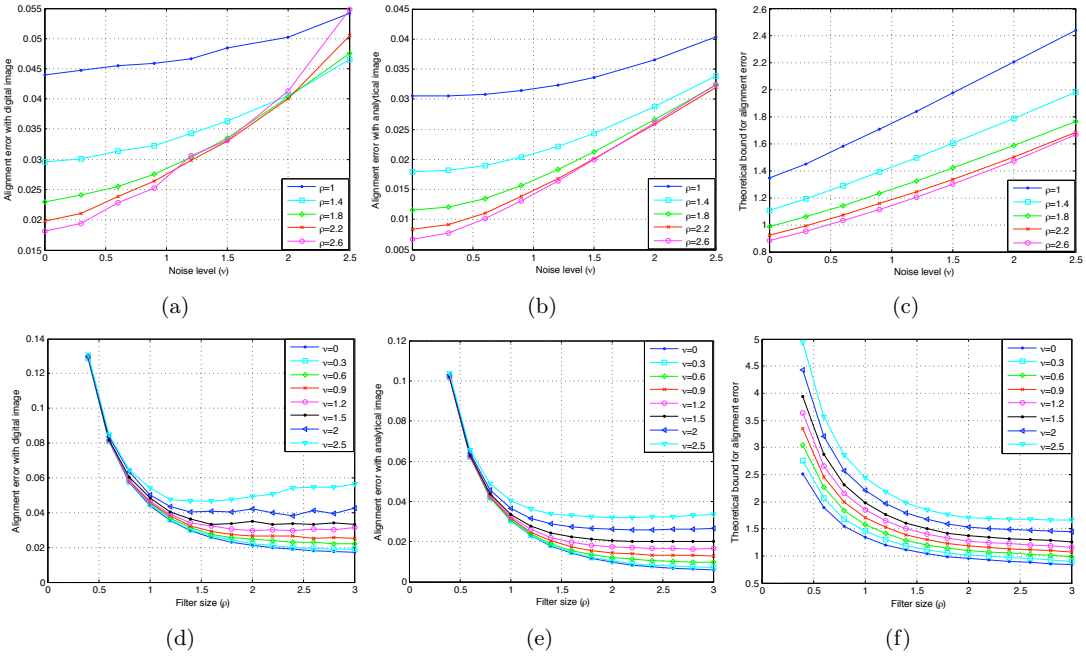


Figure 5.6: Alignment errors of real images for 2-D manifolds generated by translations.

computed by aligning the target patterns with the reference patterns, for both the original digital images and their approximations in the analytical dictionary $\mathcal{D}$. The theoretical upper bounds $\hat{E}$ are computed based on the analytical representations of the reference patterns. The alignment errors are plotted in Figures 5.6-5.8, which are averaged over all reference and target patterns. Figures 5.6, 5.7, and 5.8 show the errors obtained with the 2-D, 3-D and 4-D manifold models given respectively in (5.31), (5.32), and (5.33). In all figures, the alignment errors of the digital images, the alignment errors of the analytical approximations of images, and the theoretical upper bounds for the alignment error are plotted with respect to the noise level ν in panels (a)-(c), and with respect to the filter size ρ in panels (d)-(f).

The results of the experiment show that the behavior of the alignment error for digital image representations is very similar to the behavior of the error obtained with the analytical approximations of the images in $\mathcal{D}$. They mostly agree with the theoretical curves as well. The plots confirm that the increase in the alignment error with the noise level converges to a linear rate as

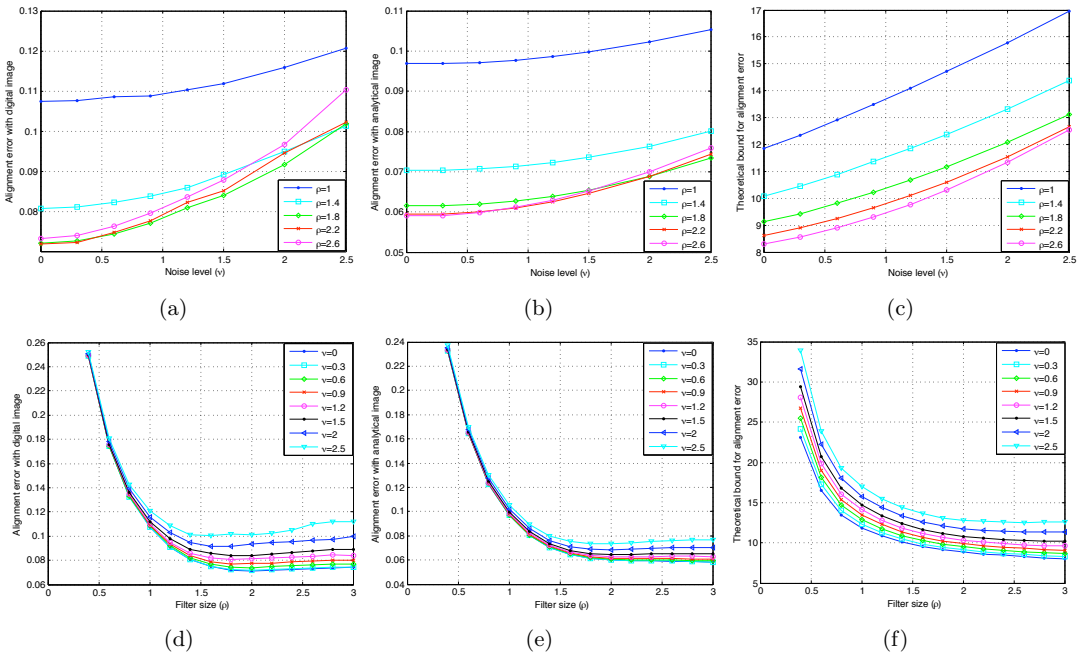


Figure 5.7: Alignment errors of real images for 3-D manifolds generated by translations and rotations.

predicted by the theoretical results. The variation of the error with filtering is also in agreement with Theorem 2, and different transformation models lead to different behaviors for the alignment error as in the previous set of experiments. Meanwhile, it is observable that the dependence of the alignment error $\hat{E}$ on the filter size ρ in these experiments is mostly determined by its first component $\hat{E}_1$ related to manifold nonlinearity, even at large filter sizes. This is in contrast to the results obtained in the first setup with synthetically generated random patterns. The difference between the two setups can be explained as follows. Real images generally contain more high-frequency components than synthetical images generated in the smooth dictionary $\mathcal{D}$. These are captured with fine, small-scale atoms in the analytical approximations (the smallest atom scale used in this setup is 0.05, while it is 0.3 in the previous setup). The high-frequency components increase the manifold nonlinearity, which causes the error $\hat{E}_1$ to be the determining factor in the overall error. In return, the positive effect of filtering that reduces the alignment error is more prominent in these experiments, while the non-monotonic variation of the error with the filter size is still observable at large noise levels or for the transformation model (5.33) involving a scale change. The comparison of the two experimental setups shows that the exact variation of the error with filtering is influenced by the frequency characteristics of the reference patterns.

The plots in panels (d)-(f) of the figures also show that, at small filter sizes, experimental errors are relatively high and very similar for different noise levels, while this is not the case in the theoretical plots. This suggests that numerical errors in the estimation of the tangent vectors with finite differences must have some influence on the overall error in practice, which is not

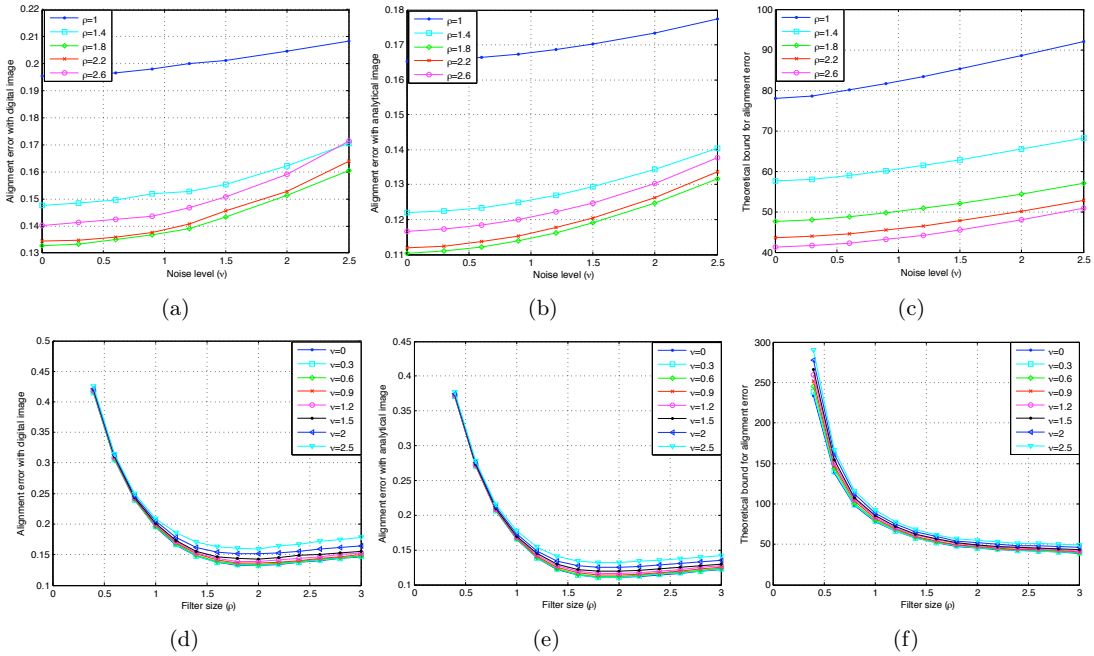


Figure 5.8: Alignment errors of real images for 4-D manifolds generated by translations, rotations, and scale changes.

taken into account in the theoretical bound. This error is higher for images with stronger high-frequency components and diminishes with smoothing (see the study in [39] for example). Lastly, one can observe that the alignment errors obtained with digital images are slightly larger than the alignment errors given by the analytic approximations of the images. This can be explained by the difference in the numerical computation of the tangent vectors in these two experimental settings. The analytic representation of the images in terms of parametric Gaussian atoms allows a more accurate computation of the tangent vectors, while the numerical interpolations employed in the computation of the tangents in the digital setting create an additional error source.

The overall conclusions of the experiments can be summarized as follows. The theoretical alignment error upper bound given in Theorem 1 gives a numerically pessimistic estimate of the alignment error as it is obtained with a worst-case analysis. However, it reflects well the actual dependence of the true alignment error both on the noise level and the filter size, and the results confirm the approximate variation rates given in Theorem 2. The theoretical upper bounds can be used in the determination of appropriate filter sizes in image registration with tangent distance.

5.5 Discussion of Results

We have derived an upper bound for the alignment error of the tangent distance method for generic transformation models. Our analysis shows that the alignment error decreases with the filter size

ρ for small values of ρ. However, in the presence of noise, the error starts increasing with ρ at relatively large filter sizes and there exists an optimal value of ρ that minimizes the alignment error. We have also shown that the alignment error bound is linearly proportional to the noise level of the target image.

We now discuss some previous studies about the performance of image registration. We begin with the works that analyze the dependence of the alignment error on noise. First, the study in [27] derives the Crámer-Rao lower bound (CRLB) for the registration of two images that differ by a 2-D translation. The CRLB gives a general lower bound for the MSE of any estimator; therefore, the lower bounds derived in [27] are valid for all registration algorithms that aim to recover the translation between two images. A Gaussian noise model is assumed in [27], and the CRLB of a translation estimator is shown to be proportional to the noise variance. One can consider the noise standard deviation in the analysis in [27] to be proportional to our noise level parameter ν, which implies that the alignment error has a lower bound of $O(\nu)$. Then, the study in [36] explores the CRLB of registration for a variety of geometric transformation models and shows that the linear variation of the CRLB with the noise level derived in [27] for translations can be generalized to several other models such as rigid, shear and affine transformations. Being a generic bound valid for any estimator, the Crámer-Rao lower bound is also valid for the tangent distance method. In our main result Theorem 2, the second component $\hat{E}_2$ of the alignment error, which is related to image noise, increases at a rate of $O(\nu)$ with the noise level ν for any geometric transformation model. Therefore, the results in [27] and [36] are consistent with ours. Finally, let us remark the following about the variation of $\hat{E}_2$ with the filter size. The studies [27] and [36] show that the CRLB of transformation estimators increases when the magnitudes of the spatial derivatives of patterns decrease. Since low-pass filtering reduces the magnitudes of spatial derivatives, it increases the MSE of estimators that compute the transformation parameters between an image pair. Our main result, which indicates that the error component $\hat{E}_2$ associated with image noise increases with filtering, is in line with these previous works.

Let us now compare our results with some previous analyses on the performance of gradient-based methods in optical flow computation, which can be regarded as the restriction of the tangent distance method to estimate 2-D translations between image patches. First, the work [27] studies the bias on gradient-based estimators, which employ a first-order approximation of the image intensity function. The bias is the difference between the expectation of the translation parameter estimates and the true translation parameters, and it results from the first-order approximation of the image intensity function. It is therefore associated with the first error term $\hat{E}_1$ in Theorem 2 in our analysis. Note that the second error term $\hat{E}_2$ results from image noise and is related to the variance of the estimator when a zero-mean random noise model is assumed. It is shown in [27] that the bias is more severe if the image has larger bandwidth, i.e., if it has stronger high-frequency components. Hence, as smoothing the images with a low-pass filter reduces the image bandwidth, it decreases the bias. The studies in [38] and [39] furthermore report that smoothing diminishes the systematic error in the estimation of the image gradients from finite differences in optical flow computation, as it reduces the second and higher-order derivatives of the image intensity function. The results in [27] are consistent with our analysis, which shows that the component of the alignment error associated with manifold nonlinearity decreases with the filter size ρ. Our result is however valid not only for translations, but for other transformation models as well. Moreover, it provides an exact rate of decrease for the error, which is given by $O\left((1+\rho^2)^{-1/2}\right)$ for translations, and

$O\left(1+(1+\rho^2)^{-1/2}\right)$ for other transformation models. Lastly, the analysis in [27] reports that the bias due to series truncation has a polynomial dependence on the amount of translation. In the bound given in Theorem 1, the alignment error term E_1 associated with manifold nonlinearity is seen to be proportional to the square $\|\lambda_o - \lambda_r\|_\infty^2$ of the distance between the transformation parameters. This quadratic dependence is due to the fact that we have used a second-order approximation of the transformation manifold; a higher-order approximation clearly yields a polynomial dependence of higher-degree as obtained in [27].

5.6 Conclusion

In this chapter, we have presented a performance analysis of hierarchical image registration with the tangent distance method, which uses a first-order approximation of the transformation manifold in the estimation of the geometric transformation between two images. We have derived an upper bound for the alignment error and analyzed its variation with the noise level and the size of the low-pass filter used for smoothing the images in hierarchical algorithms. Our main finding is that the alignment error decreases with filtering for small filters, as filtering reduces manifold nonlinearity. It however increases with filtering for large filters due to the effect of image noise, while it is linearly proportional to the noise level. Therefore, there exists an optimal value of the filter size that minimizes the alignment error, which depends on the noise level and the geometric structure of the manifold. Our treatment is generic and valid for arbitrary geometric transformation models, and provides an exact prediction for the joint rate of variation of the alignment error with the filter size and the noise level. The presented study provides insight for the understanding of multiscale registration methods that are based on manifold linearizations, and are helpful for obtaining a better performance with these methods in image registration and transformation-invariant image analysis applications.

Chapter 6

Analysis of Image Registration with Descent Methods

6.1 Overview of Image Registration Analysis

Manifold distance computation is a difficult problem, which is tightly linked to the image registration problem. In Chapter 5, we have studied the registration of images with the tangent distance method, which gives an easy solution based on minimizing the distance to the linear approximation of the manifold. In this chapter, we focus on another simple and fast method for solving the registration problem, which is the minimization of the actual distance to the manifold with a simple local optimizer. We consider the particular transformation model of 2-D translations and study a rather basic problem concerning the performance of multiscale image registration algorithms that use local optimization methods. Smoothing the images is heuristically known to improve the well-behavedness of the dissimilarity function in image registration by reducing the local minima [42], [24], [101]. Meanwhile, it also causes an increase in the alignment error in a noisy setting, which has already been observed in studies such as [27] examining the CRLB of registration. The selection of good low-pass filters in image registration requires the consideration of these two effects together.

Despite the awareness of the link between smoothing and the ease of registration, the influence of smoothing on the density of local minima of the dissimilarity function has never been theoretically studied before. Moreover, none of the previous studies characterize the exact relation between the properties of the low-pass filter used in smoothing and the alignment error inherent in and common to all region-based methods, which is the change in the location of the global minimum of the dissimilarity function due to the perturbation caused by the additive noise on the images. In this chapter, we present a study that aims to respond to these two important issues, which influence the performance of all region-based registration methods. The density of the local minima of the dissimilarity function is directly related to the effectiveness of local descent-type optimizers in registration. Also, assuming that the translation between two images is sufficiently small, the perturbation in the global minimum of the dissimilarity function corresponds exactly to the alignment error of descent methods in image registration. Hence, in order to provide a solid illustration of our alignment analysis, we consider the gradient descent method as the optimization technique. Note however that the alignment error bounds derived in this chapter are relevant to the performance of

not only gradient descent, but other registration methods as well. For instance, it has been seen in Section 5.3.2 of Chapter 5 that the actual error of the tangent distance method is affected by the perturbation in the global minimum of the distance function.

Although the registration problem is formulated for the estimation of the global 2-D translation between a reference image and a target image in this chapter, one can equivalently assume that the considered reference and target patterns are image patches rather than complete images. For this reason, our study is of interest not only for registration applications where the transformation between the image pair is modeled by a pure translation (e.g., as in satellite images), but also for various motion estimation techniques, such as block-matching algorithms and region-based matching techniques in optical flow that assign constant displacement vectors to image subregions. We adopt an analytic and parametric model for the reference and target patterns and formulate the registration problem in the continuous domain of square-integrable functions $L^2(\mathbb{R}^2)$. We use the squared-distance between the image intensity functions as the dissimilarity measure. This distance function is the continuous domain equivalent of SSD. We study two different aspects of image registration; namely, *alignment regularity* and *alignment accuracy*.

We first look at *alignment regularity*; i.e., the well-behavedness of the distance function, and estimate the largest neighborhood of translations such that the distance function has only one local minimum, which is also the global minimum. Then we study the influence of smoothing the reference and target patterns on the neighborhood of translations recoverable with local minimizers such as descent-type algorithms without getting trapped in a local minimum. In more details, we consider the translation manifold of the reference pattern, which is the set of patterns generated by its translations. In the examination of the alignment regularity, we assume that the target pattern lies on the translation manifold of the reference pattern. We then consider the distance function $f(\lambda) = \|p_\lambda - q\|^2$ between the target pattern q and the translated version p_λ of the reference pattern p, where λ denotes a translation parameter. The global minimum of f is at the origin $\lambda = 0$. Then, in the translation parameter domain, we consider the largest open neighborhood around the origin within which f is an increasing function along any ray starting out from the origin. We call this neighborhood the Single Distance Extremum Neighborhood (SIDEN). The SIDEN of a reference pattern is important in the sense that it defines the translations that can be correctly recovered by minimizing f with a descent method. We derive an analytic estimation of the SIDEN. Then, in order to study the effect of smoothing on the alignment regularity, we consider the registration of low-pass filtered versions of the reference and target patterns and examine how the SIDEN varies with the filter size. Our main result is that the volume (area) of the SIDEN increases at a rate of at least $O(1 + \rho^2)$ with respect to the size ρ of the low-pass filter kernel, which controls the level of smoothing. This formally shows that, when the patterns are low-pass filtered, a wider range of translation values can be recovered with descent-type methods; hence, smoothing improves the regularity of alignment. Then, we demonstrate the usage of our SIDEN estimate for constructing a regular multiresolution grid in the translation parameter domain with exact alignment guarantees. Based on our estimation of the neighborhood of translations that are recoverable with descent methods, we design an adaptive search grid in the translation parameter domain such that large translations can be recovered by locating the closest solution on the grid and then refining this estimation with a descent method.

Then we look at *alignment accuracy* and study the effect of image noise on the accuracy of image alignment. We also characterize the influence of low-pass filtering on the alignment accuracy

in a noisy setting. This is an important matter, as the target image is rarely an exactly translated version of the reference image in practice. When the target pattern is noisy, it is not exactly on the translation manifold of the reference pattern. The noise on the target pattern causes the global minimum of the distance function to deviate from the solution $\lambda = 0$. We formulate the alignment error as the perturbation in the global minimum of the distance function, which corresponds to the misalignment between the image pair due to noise. We focus on two different noise models. In the first setting, we look at Gaussian noise. In the second setting, we examine arbitrary square-integrable noise patterns, where we consider general noise patterns and noise patterns that have small correlation with the points on the translation manifold of the reference pattern. We derive upper bounds on the alignment error in terms of the noise level and the pattern parameters in both settings. We then consider the smoothing of the reference and target patterns in these settings and look at the variation of the alignment error with the noise level and the filter size. It turns out that the alignment error bound increases at a rate of $O\left(\eta^{1/2}\,(1-\eta)^{-1/2}\right)$ and $O\left(\nu^{1/2}\,(1-\nu)^{-1/2}\right)$ in respectively the first and second settings with respect to the noise level, where η is the standard deviation of the Gaussian noise, and ν is the norm of the noise pattern. Another observation is that the alignment error is small if the noise pattern has small correlation with translated versions of the reference pattern. Moreover, the alignment error bounds increase at the rates $O\left(\rho^{3/2}\,(1-\rho)^{-1/2}\right)$ and $O\left((1+\rho^2)^{1/2}\right)$ in the first and second settings, with respect to the filter size ρ. Therefore, our main finding is that smoothing the image pair tends to increase the alignment error when the target pattern does not lie on the translation manifold of the reference pattern. The experimental results confirm that the behavior of the theoretical bound as a function of the noise level and filter size reflects well the behavior of the actual error.

This chapter is is organized as follows. In Section 6.2, we focus on the alignment regularity problem, where we first derive an estimation of the SIDEN and then examine its variation with filtering. Then in Section 6.3, we look into the alignment accuracy problem and present our results regarding the influence of noise on the alignment accuracy. In Section 6.4, we present experimental results. In Section 6.5, we give a discussion of our results and interpret them in comparison with the previous studies in the literature. Finally, we conclude in Section 6.6.

6.2 Analysis of Alignment Regularity

6.2.1 Notation and problem formulation

Let $p \in L^2(\mathbb{R}^2)$ be a visual pattern with a non-trivial support on $\mathbb{R}^2$ (i.e., $p(X)$ is not equal to 0 almost everywhere on $\mathbb{R}^2$). In order to study the image registration problem analytically, we adopt a representation of p in the analytic and parametric dictionary manifold $\mathcal{D}$ defined in (2.10) with the Gaussian function $\phi(X) = e^{-X^TX} = e^{-(x^2+y^2)}$ as the mother function. We assume that a sufficiently accurate approximation of p with finitely many atoms in $\mathcal{D}$ is available; i.e.,

$$p(X) \approx \sum_{k=1}^{K} c_k\, \phi_{\gamma_k}(X) \tag{6.1}$$

where K is the number of atoms used in the representation of p, γ_k are the atom parameters and c_k are the atom coefficients.

Throughout this chapter, $T = [T_x\ T_y]^T \in S^1$ denotes a unit-norm vector and S^1 is the unit circle in $\mathbb{R}^2$. We use the notation tT for translation vectors, where $t \geq 0$ denotes the magnitude of the vector (amount of translation) and T defines the direction of translation. Then, the translation manifold $\mathcal{M}(p)$ of p is the set of patterns generated by translating p

$$\mathcal{M}(p) = \{p(X - t\,T) : T \in S^1,\, t \in [0, +\infty)\} \subset L^2(\mathbb{R}^2). \tag{6.2}$$

We consider the squared-distance between the reference pattern $p(X)$ and its translated version $p(X - tT)$. This distance is the continuous domain equivalent of the SSD measure that is widely used in registration methods. The squared-distance in the continuous domain is given by

$$f(tT) = \|p(X) - p(X - tT)\|^2 = \int_{\mathbb{R}^2} (p(X) - p(X - tT))^2 dX. \tag{6.3}$$

The global minimum of f is at the origin $tT = 0$. Therefore, there exists a region around the origin within which the restriction of f to a ray tT_a starting out from the origin along an arbitrary direction T_a is an increasing function of $t > 0$ for all T_a. This allows us to define the Single Distance Extremum Neighborhood (SIDEN) as follows.

Definition 8. *We call the set of translation vectors*

$$\mathcal{S} = \{0\} \cup \{\omega_T T : T \in S^1, \omega_T > 0,\ \text{and}\ \frac{df(tT)}{dt} > 0\ \ \text{for all}\ 0 < t \leq \omega_T\} \tag{6.4}$$

the Single Distance Extremum Neighborhood (SIDEN) of the pattern p.

Note that the origin $\{0\}$ is included separately in the definition of SIDEN since the gradient of f vanishes at the origin and therefore $df(tT)/dt|_{t=0} = 0$ for all T. The SIDEN $\mathcal{S} \subset \mathbb{R}^2$ is an open neighborhood of the origin such that the only stationary point of f inside $\mathcal{S}$ is the origin. We formulate this in the following proposition.

Proposition 5. *Let $tT \in \mathcal{S}$. Then $\nabla f(tT) = 0$ if and only if $tT = 0$.*

Proof: Let $\nabla f(tT) = 0$ for some $tT \in \mathcal{S}$. Then, $\nabla_T f(tT) = 0$, which is the directional derivative of f along the direction T at tT. This gives

$$\nabla_T f(tT) = \frac{d}{du} f(tT + uT)\bigg|_{u=0} = \frac{d}{du} f\left((t+u)T\right)\bigg|_{u=0} = \frac{d}{du} f(uT)\bigg|_{u=t} = \frac{df(tT)}{dt} = 0$$

which implies that $t = 0$, as $tT \in \mathcal{S}$. The second part $\nabla f(0) = 0$ of the statement also holds clearly, since the global minimum of f is at 0. □

Proposition 5 can be interpreted as follows. The only local minimum of the distance function f is at the origin in $\mathcal{S}$. Therefore, when a translated version $p(X - tT)$ of the reference pattern is aligned with $p(X)$ with a local optimization method like a gradient descent algorithm, the local minimum achieved in $\mathcal{S}$ is necessarily also the global minimum.

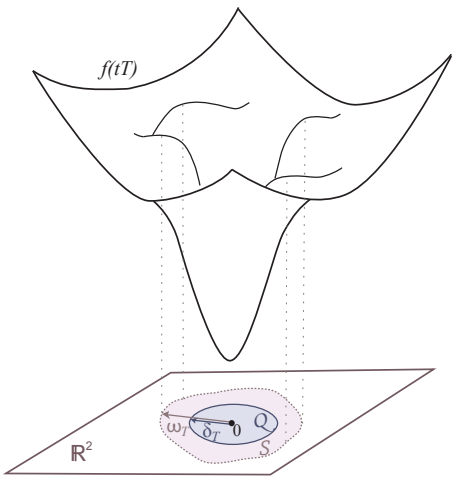


Figure 6.1: SIDEN $\mathcal{S}$ is the largest open neighborhood around the origin within which the distance f is increasing along all rays starting out from the origin. Along each unit direction T, $\mathcal{S}$ covers points $\omega_T T$ such that $f(tT)$ is increasing between 0 and $\omega_T T$. The estimate $\mathcal{Q}$ of $\mathcal{S}$ is obtained by computing a lower bound δ_T for the first zero-crossing of $df(tT)/dt$.

The goal of our analysis is now the following. Given a reference pattern p, we would like to find an analytical estimation of $\mathcal{S}$. However, the exact derivation of $\mathcal{S}$ requires the calculation of the exact zero-crossings of $df(tT)/dt$, which is not easy to do analytically. Instead, one can characterize the SIDEN by computing a neighborhood $\mathcal{Q}$ of 0 that lies completely in $\mathcal{S}$; i.e., $\mathcal{Q} \subset \mathcal{S}$. $\mathcal{Q}$ can be derived by using a polynomial approximation of f and calculating, for all unit directions T, a lower bound δ_T for the supremum of ω_T such that $\omega_T T$ is in $\mathcal{S}$. This does not only provide an analytic estimation of the SIDEN, but also defines a set that is known to be completely inside the SIDEN. The regions $\mathcal{S}$ and $\mathcal{Q}$ are illustrated in Figure 6.1.

In Section 6.2.2 we derive $\mathcal{Q}$. In particular, $\mathcal{Q}$ is obtained in the form of a compact analytic set and f is a differentiable function. This guarantees that, if the translation that aligns the image pair perfectly is in the set $\mathcal{Q}$, the distance function f can be minimized with gradient descent algorithms; the solution converges to a local minimum of f in $\mathcal{Q}$, which is necessarily the global minimum of f, resulting in a perfect alignment. Moreover, we will see in Section 6.4 that the knowledge of a set $\mathcal{Q} \subset \mathcal{S}$ permits us to design a registration algorithm that can recover large translations perfectly.

Finally, as $\mathcal{Q}$ is obtained analytically and parametrically, it is simple to examine its variation with the low-pass filtering applied to p. This is helpful for gaining an understanding of the relation between the alignment regularity and smoothing. We study this relation in Section 6.2.3.

6.2.2 Estimation of SIDEN

We now derive an estimation $\mathcal{Q}$ for the Single Distance Extremum Neighborhood $\mathcal{S}$. In the following, we consider T to be a fixed unit direction in S^1. We derive $\mathcal{Q} \subset \mathcal{S}$ by computing a δ_T which guarantees that $df(tT)/dt > 0$ for all $0 < t \leq \delta_T$. In the derivation of $\mathcal{Q}$, we need a closed-form expression for $df(tT)/dt$. Since f is the distance between the patterns $p(X)$ and $p(X - tT)$ that

are represented in terms of Gaussian atoms, its derivation requires the integration of products of Gaussian atom pairs. We thus use the formula provided by Proposition 8 given in Appendix B.3 in the examination of the distance function f.

The terms

$$\begin{aligned}\Sigma_{jk} &= \frac{1}{2}\left(\Psi_j\,\sigma_j^2\,\Psi_j^{-1} + \Psi_k\,\sigma_k^2\,\Psi_k^{-1}\right)\\ Q_{jk} &= \frac{\pi\,|\sigma_j\sigma_k|}{\sqrt{|\Sigma_{jk}|}}\exp\left(-\frac{1}{2}(\tau_k-\tau_j)^T\,\Sigma_{jk}^{-1}\,(\tau_k-\tau_j)\right)\end{aligned}$$

defined in Proposition 8 are functions of the parameters of the j-th and k-th atoms. We also denote

$$\begin{aligned}a_{jk} &:= \frac{1}{2}\,T^T\,\Sigma_{jk}^{-1}\,T, \qquad b_{jk} := \frac{1}{2}\,T^T\,\Sigma_{jk}^{-1}\,(\tau_k-\tau_j)\\ c_{jk} &:= \frac{1}{2}\,(\tau_k-\tau_j)^T\,\Sigma_{jk}^{-1}\,(\tau_k-\tau_j).\end{aligned} \tag{6.5}$$

Notice that $a_{jk} > 0$ and $c_{jk} \geq 0$ since $\|T\| = 1$ and Σ_{jk}, Σ_{jk}^{-1} are positive definite matrices. By definition, $Q_{jk} > 0$ as well. Note also that a_{jk} and b_{jk} are functions of the unit direction T; however, for the sake of simplicity we avoid expressing their dependence on T explicitly in our notation.

We can now give our result about the estimation of the SIDEN.

Theorem 3. *The region $\mathcal{Q} \subset \mathbb{R}^2$ is a subset of the SIDEN $\mathcal{S}$ of the pattern p if*

$$\mathcal{Q} = \{tT : T \in S^1,\ 0 \leq t \leq \delta_T\}$$

where δ_T is the only positive root of the polynomial $|\alpha_4|t^3 - \alpha_3 t^2 - \alpha_1$ and

$$\begin{aligned}\alpha_1 &= \sum_{j=1}^{K}\sum_{k=1}^{K} c_j c_k\,Q_{jk}\,(2\,a_{jk} - 4\,b_{jk}^2)\\ \alpha_3 &= \sum_{j=1}^{K}\sum_{k=1}^{K} c_j c_k\,Q_{jk}\left(-\frac{8}{3}\,b_{jk}^4 + 8\,b_{jk}^2\,a_{jk} - 2\,a_{jk}^2\right)\\ \alpha_4 &= -1.37\sum_{j=1}^{K}\sum_{k=1}^{K}|c_j c_k|\,Q_{jk}\,\exp\left(\frac{b_{jk}^2}{a_{jk}}\right)a_{jk}^{5/2}\end{aligned}$$

are constants depending on T and on the parameters γ_k of the atoms in p.

The proof of Theorem 3 is given in the technical report [102, Appendix A.1]. The proof applies a Taylor expansion of $df(tT)/dt$ and derives δ_T such that $df(tT)/dt$ is positive for all $t \leq \delta_T$. Therefore, along each direction T, δ_T constitutes a lower bound for the first zero-crossing of $df(tT)/dt$ (see Figure 6.1 for an illustration of δ_T). By varying T over the unit circle, one obtains a closed neighborhood $\mathcal{Q}$ of 0 that is a subset of $\mathcal{S}$. This region can be analytically computed using only

the parametric representation of p and provides an estimate for the range of translations tT over which $p(X)$ can be exactly aligned with $p(X - tT)$.

6.2.3 Variation of SIDEN with smoothing

We now examine how smoothing the reference pattern p with a low-pass filter influences its SIDEN. We consider the Gaussian filter kernel we have studied in Chapter 5, which is given in (5.6). Remember from Chapter 5 that the smoothed version $\hat{p}$ of the reference pattern p can be obtained by replacing the original atom coefficients c_k and atom scale matrices σ_k by $\hat{c}_k$ and $\hat{\sigma}_k$, whose expressions are given in (5.20) and (5.19). Therefore, the filtered version of the reference pattern in (6.1) is given in the form

$$\hat{p}(X) = \sum_{k=1}^{K} \hat{c}_k \phi_{\hat{\gamma}_k}(X). \tag{6.6}$$

Considering the same setting as in Section 6.2.1, where the target pattern $p(X - tT)$ is exactly a translated version of the reference pattern $p(X)$, we now assume that both the reference and target patterns are low-pass filtered as it is typically done in hierarchical image registration algorithms. When a pattern is low-pass filtered, the scale parameters of its atoms increase and the atom coefficients decrease proportionally to the filter kernel size, leading to a spatial diffusion of the image intensity function. The goal of this section is to show that this diffusion increases the volume of the SIDEN. We achieve this by analyzing the variation of the smoothed SIDEN estimate $\hat{\mathcal{Q}}$ corresponding to the smoothed distance

$$\hat{f}(tT) = \int_{\mathbb{R}^2} \big(\hat{p}(X) - \hat{p}(X - tT)\big)^2 dX \tag{6.7}$$

with respect to the filter size ρ. Since the smoothed pattern has the same parametric form (6.6) as the original pattern, the variation of $\hat{\mathcal{Q}}$ with ρ can be analyzed easily by examining the dependence of the parameters involved in the derivation of $\hat{\mathcal{Q}}$ on ρ. In the following, we express the terms in Section 6.2.2 that have a dependence on ρ with the notation $\hat{(.)}$, such as $\hat{a}_{jk}$, $\hat{b}_{jk}$, $\hat{c}_k$, $\hat{\sigma}_k$. We write the terms that do not depend on ρ in the same way as before; e.g., t, T, τ_k, Ψ_k.

Now, we can use Theorem 3 for the smoothed pattern $\hat{p}(X)$. For a given kernel size ρ, the smoothed versions $\hat{a}_{jk}$, $\hat{b}_{jk}$, $\hat{c}_{jk}$, $\hat{Q}_{jk}$ of the parameters in Section 6.2.2 can be obtained by replacing the scale parameters σ_k with $\hat{\sigma}_k$ defined in (5.19). Then, the smoothed SIDEN corresponding to ρ is given as $\hat{\mathcal{Q}} = \{tT : T \in S^1,\ 0 \le t \le \hat{\delta}_T\}$ where $\hat{\delta}_T$ is the positive root of the polynomial $|\hat{\alpha}_4| t^3 - \hat{\alpha}_3 t^2 - \hat{\alpha}_1$ such that

$$\hat{\alpha}_1 = \sum_{j=1}^{K}\sum_{k=1}^{K} \hat{c}_j \hat{c}_k \, \hat{Q}_{jk} \, (2\, \hat{a}_{jk} - 4\, \hat{b}_{jk}^2), \qquad \hat{\alpha}_3 = \sum_{j=1}^{K}\sum_{k=1}^{K} \hat{c}_j \hat{c}_k \, \hat{Q}_{jk} \left(-\frac{8}{3}\, \hat{b}_{jk}^4 + 8\, \hat{b}_{jk}^2\, \hat{a}_{jk} - 2\, \hat{a}_{jk}^2 \right)$$

$$\hat{\alpha}_4 = -1.37 \sum_{j=1}^{K}\sum_{k=1}^{K} |\hat{c}_j \hat{c}_k| \, \hat{Q}_{jk} \, \exp\left(\frac{\hat{b}_{jk}^2}{\hat{a}_{jk}} \right) \hat{a}_{jk}^{5/2}.$$

Similarly to the derivation in Section 6.2.2, the terms $\hat{a}_{jk}$, $\hat{b}_{jk}$, $\hat{c}_{jk}$, $\hat{Q}_{jk}$ are associated with the

integration of the products of smoothed Gaussian atom pairs, and they appear in the closed-form expression of $d\hat{f}(tT)/dt$.

We are now ready to give the following result, which summarizes the dependence of the smoothed SIDEN estimate on the filter size ρ.

Theorem 4. *Let $V(\hat{\mathcal{Q}})$ denote the volume (area) of the SIDEN estimate $\hat{\mathcal{Q}}$ for the smoothed pattern $\hat{p}$. Then, the order of dependence of the volume of $\hat{\mathcal{Q}}$ on ρ is given by $V(\hat{\mathcal{Q}}) = O(1+\rho^2)$.*

Theorem 4 is proved in [102, Appendix A.2]. The proof is based on the examination of the order of variation of $\hat{a}_{jk}$, $\hat{b}_{jk}$, $\hat{c}_{jk}$, $\hat{Q}_{jk}$ with ρ, which is then used to derive the dependence of $\hat{\delta}_T$ on ρ.

Theorem 4 is the main result of this section. It states that the volume of the SIDEN estimate increases with the size of the filter applied on the patterns to be aligned. The theorem shows that the area of the region of translation vectors for which the reference pattern $\hat{p}(X)$ can be perfectly aligned with $\hat{p}(X-tT)$ using a descent method expands at the rate $O(1+\rho^2)$ with respect to the increase in the filter size ρ. Here, the order of variation $O(1+\rho^2)$ is obtained for the estimate $\hat{\mathcal{Q}}$ of the SIDEN. Hence, one may wonder if the volume $V(\hat{\mathcal{S}})$ of the SIDEN $\hat{\mathcal{S}}$ has the same dependence on ρ. Remembering that $\hat{\mathcal{Q}} \subset \hat{\mathcal{S}}$ for all ρ, one immediate observation is that the rate of expansion of $\hat{\mathcal{S}}$ must be at least $O(1+\rho^2)$; otherwise, there would exist a sufficiently large value of ρ such that $\hat{\mathcal{Q}}$ is not included in $\hat{\mathcal{S}}$. One can therefore conclude that $V(\hat{\mathcal{S}}) \geq V(\hat{\mathcal{Q}}) = O(1+\rho^2)$. However, this only gives a lower bound for the rate of expansion of $\hat{\mathcal{S}}$ and the exact rate of expansion of $\hat{\mathcal{S}}$ may be larger. In the following, we make a few comments about the variation of $\hat{\mathcal{S}}$ with ρ.

Remark. As shown in the proof of Theorem 3, the derivative of the distance function $f(tT)$ is of the form

$$\frac{df(tT)}{dt} = \sum_{j=1}^{K}\sum_{k=1}^{K} c_j c_k \; Q_{jk} \; s_{jk}(t) \tag{6.8}$$

where

$$s_{jk}(t) = e^{-\left(a_{jk}\,t^2 + 2\,b_{jk}\,t\right)}\,(a_{jk}\,t \,+\, b_{jk}) \,+\, e^{-\left(a_{jk}\,t^2 - 2\,b_{jk}\,t\right)}\,(a_{jk}\,t \,-\, b_{jk})\,. \tag{6.9}$$

In order to derive $\mathcal{S}$, one needs to exactly locate the smallest zero-crossing of $\frac{df(tT)}{dt}$. This is not easy to do analytically due to the complicated form of the functions $s_{jk}(t)$, which we handle with polynomial approximations in the derivation of $\mathcal{Q}$. However, in order to gain an intuition about how the zero-crossings change with filtering, one can look at the dependence of the extrema of the two additive terms in $s_{jk}(t)$ on ρ. The function $e^{-\left(a_{jk}\,t^2 + 2\,b_{jk}\,t\right)}\,(a_{jk}\,t \,+\, b_{jk})$ has two extrema at

$$\mu_0 = \frac{1}{a_{jk}}\left(-\sqrt{\frac{a_{jk}}{2}} - b_{jk}\right), \qquad \mu_1 = \frac{1}{a_{jk}}\left(\sqrt{\frac{a_{jk}}{2}} - b_{jk}\right) \tag{6.10}$$

and $e^{-\left(a_{jk}\,t^2 - 2\,b_{jk}\,t\right)}\,(a_{jk}\,t \,-\, b_{jk})$ has two extrema at

$$\mu_2 = \frac{1}{a_{jk}}\left(-\sqrt{\frac{a_{jk}}{2}} + b_{jk}\right), \qquad \mu_3 = \frac{1}{a_{jk}}\left(\sqrt{\frac{a_{jk}}{2}} + b_{jk}\right). \tag{6.11}$$

Now replacing the original parameters a_{jk}, b_{jk} with their smoothed versions $\hat{a}_{jk}$, $\hat{b}_{jk}$ and using the

result from the proof of Theorem 4 that $\hat{a}_{jk}$ and $\hat{b}_{jk}$ decrease at a rate of $O\left((1+\rho^2)^{-1}\right)$, it is easy to show that the locations of the extrema $\hat{\mu}_0, \hat{\mu}_1, \hat{\mu}_2, \hat{\mu}_3$ change with a rate of $O\big((1+\rho^2)^{1/2}\big)$. One may thus conjecture that the zero-crossings of $df(tT)/dt$ along a fixed direction T might also move at the same rate, which gives the volume of $\hat{\mathcal{S}}$ as $V(\hat{\mathcal{S}}) = O(1+\rho^2)$.

On the other hand, $V(\hat{\mathcal{S}})$ may also exhibit a different type of variation with ρ depending on the atom parameters of p. In particular, $V(\hat{\mathcal{S}})$ may expand at a rate greater than $O(1+\rho^2)$ for some patterns. For example, as shown in [102, Proposition 4], there exists a threshold value ρ_0 of the filter size such that for all $\rho > \rho_0$, $\hat{\mathcal{S}} = \mathbb{R}^2$ and thus $V(\hat{\mathcal{S}}) = \infty$ for patterns that consist of atoms with coefficients of the same sign. In addition, patterns whose atoms with positive (or negative) coefficients are dominant over the atoms with the opposite sign are likely to have this property due to their resemblance to patterns consisting of atoms with coefficients of the same sign.

Theorem 4 describes the effect of smoothing images before alignment. One may then wonder what the optimal filter size to be applied to the patterns before alignment is, given a reference and a target pattern. Theorem 4 suggests that, if the target pattern is on the translation manifold of the reference pattern, applying a large filter is always preferable as it provides a large range of translations recoverable by descent algorithms. The accuracy of alignment does not change with the filter size in this noiseless setting, since a perfect alignment is always guaranteed with descent methods as long as the amount of translation is inside the SIDEN. However, the assumption that the target pattern is exactly of the form $p(X - tT)$ is not realistic in practice; i.e., in real image processing applications, the target image is likely to deviate from $\mathcal{M}(p)$ due to the noise caused by image capture conditions, imaging model characteristics, etc. Hence, we examine in Section 6.3 if filtering affects the accuracy of alignment when the target image deviates from $\mathcal{M}(p)$.

6.3 Analysis of Alignment Accuracy in Noisy Settings

We now analyze the effect of noise and smoothing on the accuracy of the estimation of translation parameters. In general, noise causes a perturbation in the location of the global minimum of the distance function. The perturbed version of the single global minimum of the noiseless distance function f will remain in the form of a single global minimum for the noisy distance function with high probability if the noise level is sufficiently small. The noise similarly introduces a perturbation on the SIDEN as well. The exact derivation of the SIDEN in the noisy setting requires the examination of the first zero-crossings of the derivative of the noisy distance function along arbitrary directions T around its global minimum. At small noise levels, these zero-crossings are expected to be perturbed versions of the first zero-crossings of $df(tT)/dT$ around the origin, which define the boundary of the noiseless SIDEN $\mathcal{S}$. The perturbation on the zero-crossings depends on the noise level. If the noise level is sufficiently small, the perturbation on the zero-crossings will be smaller than the distance between $\mathcal{S}$ and its estimate $\mathcal{Q}$. This is due to the fact that $\mathcal{Q}$ is a worst-case estimate for $\mathcal{S}$ and its boundary is sufficiently distant from the boundary of $\mathcal{S}$ in practice, which is also confirmed by the experiments in Section 6.4. In this case, the estimate $\mathcal{Q}$ obtained from the noiseless distance function f is also a subset of the noisy SIDEN. Therefore, under the small noise assumption, $\mathcal{Q}$ can be considered as an estimate of the noisy SIDEN as well and it can be

used in the alignment of noisy images in practice.[1] Our alignment analysis in this section relies on this assumption. Since we consider that the reference and target patterns are aligned with a descent-type optimization method, the solution will converge to the global minimum of the noisy distance function in the noisy setting. The alignment error is then given by the change in the global minimum of the distance function, which we analyze now.

The selection of the noise model for the representation of the deviation of the target pattern from the translation manifold of the reference pattern depends on the imaging application. It is common practice to represent non-predictable deviations of the image intensity function from the image model with additive Gaussian noise. This noise model fits well the image intensity variations due to imperfections of the image capture system, sensor noise, etc. Meanwhile, in some settings, one may have a prior knowledge of the type of the deviation of the target image from the translation manifold of the reference image. For instance, the deviation from the translation manifold may be due to some geometric image deformations, non-planar scene structures, etc. In such settings, one may be able to bound the magnitude of the deviation of the image intensity function from the translation model. Considering these, we examine two different noise models in our analysis. We first focus on a setting where the target pattern is corrupted with respect to an analytic noise model in the continuous space $L^2(\mathbb{R}^2)$. The analytic noise model is inspired by the i.i.d. Gaussian noise in the discrete space $\mathbb{R}^n$. In Section 6.3.1, we derive a probabilistic upper bound on the alignment error for this setting in terms of the parameters of the reference pattern and the noise model. Then, in Section 6.3.2, we generalize the results of Section 6.3.1 to arbitrary noise patterns in $L^2(\mathbb{R}^2)$ and derive an error bound in terms of the norm of the noise pattern. The influence of smoothing the reference and target patterns on the alignment error is discussed in Section 6.3.3.

Throughout Section 6.3, we use the notations $\overline{(\cdot)}$ and $\underline{(\cdot)}$ to refer respectively to upper and lower bounds on a variable $(\cdot)$. The parameters corresponding to smoothed patterns are written as $\hat{(.)}$ as in Section 6.2.3. The notations $R_{(.)}$ and $C_{(.)}$ are used to denote important upper bounds appearing in the main results, which are associated with the parameter in the subscript.

6.3.1 Derivation of an upper bound on alignment error for Gaussian noise

We consider the noiseless reference pattern p in (6.1) and a target pattern that is a noisy observation of a translated version of p. We assume an analytical noise model given by

$$w(X) = \sum_{l=1}^{L} \zeta_l \, \phi_{\xi_l}(X), \tag{6.12}$$

where the noise units $\phi_{\xi_l}(X)$ are Gaussian atoms of scale ϵ. The coefficients ζ_l and the noise atom parameters ξ_l are assumed to be independent. The noise atoms are of the form $\phi_{\xi_l}(X) = \phi\big(E^{-1}(X - \delta_l)\big)$ where

$$E = \begin{bmatrix} \epsilon & 0 \\ 0 & \epsilon \end{bmatrix}, \quad \delta_l = \begin{bmatrix} \delta_{x,l} \\ \delta_{y,l} \end{bmatrix}.$$

The vector δ_l is the random translation parameter of the noise atom ϕ_{ξ_l} such that the random variables $\{\delta_{x,l}\}_{l=1}^{L}, \{\delta_{y,l}\}_{l=1}^{L} \sim U[-b, b]$ have an i.i.d. uniform distribution. Here, b is a fixed parameter

[1] The validity of this approximation is confirmed by the numerical simulation results in Section 6.4.

used to define a region $[-b, b] \times [-b, b] \subset \mathbb{R}^2$ in the image plane, which is considered as a support region capturing a substantial part of the energy of reference and target images. The centers of the noise atoms are assumed to be uniformly distributed in this region. In order to have a realistic noise model, the number of noise units $L \gg K$ is considered to be a very large number and the scale $\epsilon > 0$ of noise atoms is very small. The parameters L and ϵ will be treated as noise model constants throughout the analysis. The coefficients $\zeta_l \sim N(0, \eta^2)$ of the noise atoms are assumed to be i.i.d. with a normal distribution of variance η^2.

The continuous-space noise model $w(X)$ is chosen in analogy with the digital i.i.d. Gaussian noise in the discrete space $\mathbb{R}^n$. The single isotropic scale parameter ϵ of noise units bears resemblance to the one-pixel support of digital noise units. The uniform distribution of the position δ_l of noise units is similar to the way digital noise is defined on a uniform pixel grid. The noise coefficients ζ_l have an i.i.d. normal distribution as in the digital case. If our noise model $w(X)$ has to approximate the digital Gaussian noise in a continuous setting, the noise atom scale ϵ is chosen comparably to the pixel width and L corresponds to the resolution of the discrete image.

Let now p_n be a noisy observation of p such that $p_n(X) = p(X) + w(X)$, where w and p are independent according to the noise model (6.12). We assume that the target pattern is a translated version of $p_n(X)$ so that it takes the form $p_n(X - tT)$. Then, the noisy distance function between $p(X)$ and $p_n(X - tT)$ is given by

$$g(tT) = \int_{\mathbb{R}^2} \big(p(X) - p_n(X - tT)\big)^2 \, dX = \int_{\mathbb{R}^2} \big(p(X) - p(X - tT) - w(X - tT)\big)^2 \, dX. \tag{6.13}$$

This can be written as $g(tT) = f(tT) + h(tT)$, where

$$h(tT) := -2 \int_{\mathbb{R}^2} \big(p(X) - p(X - tT)\big) w(X - tT) \, dX + \int_{\mathbb{R}^2} w^2(X - tT) \, dX. \tag{6.14}$$

The function h represents the deviation of g from f. We call h the distance deviation function. The expected value of h is independent of the translation tT and given by

$$\mu_h := E[h(tT)] = \frac{\pi}{2} L \eta^2 \epsilon^2$$

where $E[.]$ denotes the expectation [102, Appendix B.1]. Therefore, $E[g(tT)] = f(tT) + \mu_h$ and the global minimum of $E[g(tT)]$ is at $tT = 0$. However, due to the probabilistic perturbation caused by the noise w, the global minimum of g is not at $tT = 0$ in general. We consider g to have a single global minimum and denote its location by $t_0 T_0$. Nevertheless, the single global minimum assumption is not a strict hypothesis of our analysis technique; i.e., the upper bound that we derive for the distance between $t_0 T_0$ and the origin is still valid if g has more than one global minimum. In this case, the obtained upper bound is valid for all global minima.

We now continue with the derivation of a probabilistic upper bound on the distance t_0 between the location $t_0 T_0$ of the global minimum of g and the location 0 of the global minimum of f. We show in [102, Appendix B.2] that t_0 satisfies the equation

$$\frac{t_0^2}{2} \left(\left. \frac{d^2 f(tT_0)}{dt^2} \right|_{t=t_1} + \left. \frac{d^2 f(tT_0)}{dt^2} \right|_{t=t_2} + \left. \frac{d^2 h(tT_0)}{dt^2} \right|_{t=t_1} \right) = |h(0) - h(t_0 T_0)| \tag{6.15}$$

for some $t_1 \in [0, t_0]$ and $t_2 \in [0, t_0]$. Our derivation of an upper bound for t_0 will be based on (6.15). The above equation shows that t_0 can be upper bounded by finding a lower bound on the term

$$\left.\frac{d^2 f(tT_0)}{dt^2}\right|_{t=t_1} + \left.\frac{d^2 f(tT_0)}{dt^2}\right|_{t=t_2} + \left.\frac{d^2 h(tT_0)}{dt^2}\right|_{t=t_1} \tag{6.16}$$

and an upper bound on the term $|h(0) - h(t_0T_0)|$. However, h is a probabilistic function; i.e., $h(tT)$ and its derivatives are random variables. Therefore, the upper bound that we will obtain for t_0 is a probabilistic bound given in terms of the variances of $h(0) - h(t_0T_0)$ and $d^2h(tT_0)/dt^2$.

In the rest of this section, we proceed as follows. First, in order to be able to bound $|h(0)-h(tT)|$ probabilistically, we present in Lemma 3 an upper bound on the variance of $h(0) - h(tT)$. Next, in order to bound the term in (6.16), we state a lower bound for $d^2f(tT)/dt^2$ in Lemma 4 and an upper bound for the variance of $d^2h(tT)/dt^2$ in Lemma 5. These results are finally put together in the main result of this section, namely Theorem 5, where an upper bound on t_0 is obtained based on (6.15). Theorem 5 applies Chebyshev's inequality to employ the bounds derived in Lemmas 3 and 5 to define probabilistic upper bounds on the terms $|h(0) - h(t_0T_0)|$ and $|d^2h(tT_0)/dt^2|$. Then, this is combined with the bound on $d^2f(tT)/dt^2$ in Lemma 4 to obtain a probabilistic upper bound on t_0 from the relation (6.15).

In the derivation of this upper bound, the direction T_0 of the global minimum of g is treated as an arbitrary and unknown unit-norm vector. Moreover, the variances of $h(0) - h(tT)$ and $d^2h(tT)/dt^2$ have a complicated dependence on t, which makes it difficult to use them directly in (6.15) to obtain a bound on t_0. In order to cope with the dependences of these terms on t and T, the upper bounds presented in Lemmas 3 and 5 are derived as uniform upper bounds over the closed ball of radius $\bar{t}_0 > 0$, $B_{\bar{t}_0}(0) = \{tT : T \in S^1, 0 \leq t \leq \bar{t}_0\}$. The upper bounds are thus independent of tT and valid for all tT vectors in $B_{\bar{t}_0}(0)$. In these lemmas, the parameter $\bar{t}_0$ is considered to be a known threshold for t_0, such that $t_0 \leq \bar{t}_0$. This parameter will be assigned a specific value in Theorem 5.

We begin with bounding the variance of term $h(0) - h(tT)$ in order to find an upper bound for the right hand side of (6.15). Let us denote

$$\Delta h(tT) := h(0) - h(tT).$$

From (6.14),

$$\Delta h(tT) = h(0) - h(tT) = 2\int_{\mathbb{R}^2} \left(p(X) - p(X - tT)\right) w(X - tT) dX$$

where we have used the fact that $\int_{\mathbb{R}^2} w^2(X - tT)dX = \int_{\mathbb{R}^2} w^2(X)dX$. Let $\sigma^2_{\Delta h(tT)}$ denote the variance of $\Delta h(tT)$. In the following lemma, we state an upper bound on $\sigma^2_{\Delta h(tT)}$. Let us define beforehand the following constants for the k-th atom of p

$$\Phi_k := \Psi_k(\sigma_k^2 + E^2)^{-1}\Psi_k^{-1}, \qquad \kappa_k := \frac{\pi\,|\sigma_k|\,|E|}{\sqrt{|\sigma_k^2 + E^2|}}.$$

Also, let $J^- = \{(j,k) : c_jc_k < 0\}$ and $J^+ = \{(j,k) : c_jc_k > 0\}$ denote the set of pairs (j,k) of atom

indices with negative and positive coefficient products.

Lemma 3. *Let $\bar{t}_0 > 0$, and let $tT \in B_{\bar{t}_0}(0)$. Then, the variance $\sigma^2_{\Delta h(tT)}$ of $\Delta h(tT)$ can be upper bounded as*

$$\sigma^2_{\Delta h(tT)} < R_{\sigma^2_{\Delta h}} := C_{\sigma^2_{\Delta h}} \eta^2 \tag{6.17}$$

where

$$C_{\sigma^2_{\Delta h}} := 4\,L \left(\sum_{(j,k)\in J^+} c_j \kappa_j \, c_k \kappa_k \; \overline{\mathfrak{c}}_{jk} + \sum_{(j,k)\in J^-} c_j \kappa_j \, c_k \kappa_k \; \underline{\mathfrak{d}}_{jk} \right).$$

Here the terms $\overline{\mathfrak{c}}_{jk}$ and $\underline{\mathfrak{d}}_{jk}$ are constants depending on $\bar{t}_0$ and the atom parameters of p. In particular, $\overline{\mathfrak{c}}_{jk}$ and $\underline{\mathfrak{d}}_{jk}$ are bounded functions of $\bar{t}_0$, given in terms of exponentials of second-degree polynomials of $\bar{t}_0$ with negative leading coefficients.

The proof of Lemma 3 is presented in [102, Appendix B.4]. In the proof, a uniform upper bound $\overline{\mathfrak{c}}_{jk}$ and a uniform lower bound $\underline{\mathfrak{d}}_{jk}$ are derived for the additive terms[2] constituting the variance of $\Delta h(tT)$. The exact expressions of $\overline{\mathfrak{c}}_{jk}$ and $\underline{\mathfrak{d}}_{jk}$ are given in Appendix C.1.

We have thus stated a uniform upper bound $R_{\sigma^2_{\Delta h}}$ for the variance of $\Delta h(tT)$ which will be used to derive an upper bound for the right hand side of (6.15) in Theorem 5. We now continue with the examination of the left hand side of (6.15). We begin with the term $d^2 f(tT)/dt^2$. The following lemma gives a lower bound on the second derivative of the noiseless distance function $f(tT)$ in terms of the pattern parameters.

Lemma 4. *The second derivative of $f(tT)$ along the direction T can be uniformly lower bounded for all $t \in [0, t_0]$ and for all directions $T \in S^1$ as follows*

$$\frac{d^2 f(tT)}{dt^2} \geq \underline{r}_0 + \underline{r}_2 t_0^2 + \underline{r}_3 t_0^3. \tag{6.18}$$

Here $\underline{r}_0 > 0$, $\underline{r}_2 \leq 0$, and $\underline{r}_3 < 0$ are constants depending on the atom parameters of p. In particular, $\underline{r}_0$, $\underline{r}_2$, $\underline{r}_3$ are obtained from the eigenvalues of some matrices derived from the parameters c_j, τ_j, Q_{jk}, Σ_{jk}.

The proof of Lemma 4 is given in [102, Appendix B.5] and the exact expressions of $\underline{r}_0$, $\underline{r}_2$, $\underline{r}_3$ are given in Appendix C.2. The above lower bound on the second derivative of $f(tT)$ is independent of the direction T and the amount t of translation, provided that t is in the interval $[0, t_0]$. In fact, the statement of Lemma 4 is general in the sense that t_0 can be any positive scalar. However, in the proof of Theorem 5, we use Lemma 4 for the t_0 value that represents the deviation between the global minima of f and g.

Lemma 4 will be used in Theorem 5 in order to lower bound the second derivative of f in (6.15). We now continue with the term $d^2 h(tT)/dt^2$ in (6.15). Let $h''(tT) := d^2 h(tT)/dt^2$ denote the second derivative of the deviation function h along the direction T. Since $h''(tT)$ can take both positive and negative values, in the calculation of a lower bound for the term (6.16), we need a

[2] $\overline{\mathfrak{c}}_{jk}$ and $\underline{\mathfrak{d}}_{jk}$ are upper and lower bounds for the terms $\mathfrak{c}_{jk}$, $\mathfrak{d}_{jk}$ used in [102, Lemma 1].

bound on the magnitude $|h''(tT)|$ of this term. It can be bounded probabilistically in terms of the variance of $h''(tT)$. We thus state the following uniform upper bound on the variance of $h''(tT)$.

Lemma 5. *Let $\bar{t}_0 > 0$, and let $tT \in B_{\bar{t}_0}(0)$. Then, the variance $\sigma^2_{h''(tT)}$ of $h''(tT)$ can be upper bounded as*

$$\sigma^2_{h''(tT)} < R_{\sigma^2_{h''}} := C_{\sigma^2_{h''}} \eta^2 \tag{6.19}$$

where

$$C_{\sigma^2_{h''}} := 4L\Bigg(\sum_{(j,k)\in J^+} c_j c_k \kappa_j \kappa_k \; \overline{\mathfrak{e}}_{jk} + \sum_{(j,k)\in J^-} c_j c_k \kappa_j \kappa_k \; \underline{\mathfrak{f}}_{jk} \Bigg).$$

Here $\overline{\mathfrak{e}}_{jk}$ is a constant depending on the atom parameters of p; and the term $\underline{\mathfrak{f}}_{jk}$ depends on the atom parameters of p and $\bar{t}_0$. In particular, $\overline{\mathfrak{e}}_{jk}$ is given in terms of rational functions of the eigenvalues of Φ_k matrices; and $\underline{\mathfrak{f}}_{jk}$ is a bounded function of $\bar{t}_0$ given in terms of exponentials of second-degree polynomials of $\bar{t}_0$ with negative leading coefficients.

The proof of Lemma 5 is given in [102, Appendix B.7]. The proof derives uniform upper and lower bounds $\overline{\mathfrak{e}}_{jk}$, $\underline{\mathfrak{f}}_{jk}$ for the additive terms[3] in the representation of $\sigma^2_{h''(tT)}$. The exact expressions for $\overline{\mathfrak{e}}_{jk}$ and $\underline{\mathfrak{f}}_{jk}$ are given in Appendix C.3.

Now we are ready to present our main result about the bound on the alignment error. The following theorem states an upper bound on the distance between the locations of the global minima of f and g in terms of the noise standard deviation η and the atom parameters of p, provided that η is smaller a threshold η_0. The threshold η_0 is obtained from the bounds derived in Lemmas 3, 4 and 5 such that the condition $\eta < \eta_0$ guarantees that the assumption $t_0 < \bar{t}_0$ holds. In the theorem, the parameter $\bar{t}_0$, which is treated as a predefined threshold on t_0 in the previous lemmas, is also assigned a specific value in terms the constants $\underline{r}_0$, $\underline{r}_2$, $\underline{r}_3$ of Lemma 4.

Theorem 5. *Let*

$$\bar{t}_0 := \sqrt{\frac{\underline{r}_0}{2|\underline{r}_2| + 2^{2/3}\underline{r}_0^{1/3}|\underline{r}_3|^{2/3}}}. \tag{6.20}$$

Let $R_{\sigma_{\Delta h}} := \sqrt{R_{\sigma^2_{\Delta h}}}$ and $R_{\sigma_{h''}} := \sqrt{R_{\sigma^2_{h''}}}$, where $R_{\sigma^2_{\Delta h}}$ and $R_{\sigma^2_{h''}}$ are as defined in (6.17) and (6.19), and evaluated at the value of $\bar{t}_0$ given above. Also, let $C_{\sigma_{\Delta h}} := \sqrt{C_{\sigma^2_{\Delta h}}}$ and $C_{\sigma_{h''}} := \sqrt{C_{\sigma^2_{h''}}}$.

Assume that for some $s > \sqrt{2}$, the noise standard deviation η is smaller than η_0 such that

$$\eta \le \eta_0 := \frac{\bar{t}_0^2\, \underline{r}_0}{2\, s\, C_{\sigma_{\Delta h}} + \bar{t}_0^2\, s\, C_{\sigma_{h''}}}. \tag{6.21}$$

Then, with probability at least $1 - \frac{2}{s^2}$, the distance t_0 between the global minima of f and g is bounded as

[3] $\overline{\mathfrak{e}}_{jk}$ and $\underline{\mathfrak{f}}_{jk}$ are upper and lower bounds for the terms $\mathfrak{e}_{jk}$, $\mathfrak{f}_{jk}$ used in [102, Lemma 3].

$$t_0 < R_{t_0} := \sqrt{\frac{2s\, R_{\sigma_{\Delta h}}}{\underline{r}_0 - s\, R_{\sigma_{h''}}}}. \tag{6.22}$$

The proof of Theorem 5 is given in [102, Appendix B.8]. In the proof, we make use of the upper bounds $R_{\sigma^2_{\Delta h}}$, $R_{\sigma^2_{h''}}$ on $\sigma^2_{\Delta h(tT)}$, $\sigma^2_{h''(tT)}$, and the lower bound on $d^2 f(tT)/dt^2$ given in (6.18). The upper bound R_{t_0} in (6.22) shows that the alignment error increases with the increase in the noise level, since $R_{\sigma_{\Delta h}}$ and $R_{\sigma_{h''}}$ are linearly proportional to the noise standard deviation η. The increase of the error with the noise is expected. It can also be seen from (6.22) that the increase in the term $\underline{r}_0$, which is proportional to the second derivative of the noiseless distance f, reduces the alignment error; whereas an increase in the term $R_{\sigma_{h''}}$, which is related to the second derivative of h, increases the error. This can be explained as follows. If f has a sharp increase around its global minimum at 0, i.e., f has a large second derivative, the location of its minimum is less affected by h. Likewise, if the distance deviation function h has a large second derivative, it introduces a larger alteration around the global minimum of f, which causes a bigger perturbation on the position of the minimum.

Theorem 5 states a bound on t_0 under the condition that the noise standard deviation η is smaller than the threshold value η_0, which depends on the pattern parameters (through the terms $\underline{r}_0$, $\underline{r}_2$, $\underline{r}_3$, $\bar{t}_0$) as well as the noise parameters L and ϵ (through the terms $C_{\sigma_{\Delta h}}$ and $C_{\sigma_{h''}}$). The threshold η_0 thus defines an admissible noise level such that the change in the location of the global minimum of f can be properly upper bounded. This admissible noise level is derived from the condition $R_{t_0} \leq \bar{t}_0$, which is partially due to our proof technique. However, we remark that the existence of such a threshold is intuitive in the sense that it states a limit on the noise power in comparison with the signal power. Note also that the denominator $\underline{r}_0 - s\, R_{\sigma_{h''}}$ of R_{t_0} should be positive, which also yields a condition on the noise level

$$\eta < \eta_0' = \frac{\underline{r}_0}{s\, C_{\sigma_{h''}}}.$$

However, this condition is already satisfied due to the hypothesis $\eta \leq \eta_0$ of the theorem, since $\eta_0 < \eta_0'$ from (6.21).

6.3.2 Generalization of the alignment error bound to arbitrary noise models

Here, we generalize the results of the previous section in order to derive an alignment error bound for arbitrary noise patterns. In general, the characteristics of the noise pattern vary depending on the imaging application. In particular, while the noise pattern may have high correlation with the reference pattern in some applications (e.g., noise resulting from geometric deformations of the pattern), its correlation with the reference pattern may be small in some other settings where the noise stems from a source that does not depend on the image. We thus focus on two different scenarios. In the first and general setting, we do not make any assumption on the noise characteristics and bound the alignment error in terms of the norm of the noise pattern. Then, in the second setting, we consider that the noise pattern has small correlation with the points on the translation manifold of the reference pattern and show that the alignment error bound can be made sharper in this case.

We assume that the reference pattern $p(X)$ is noiseless and we write the target pattern as $p_g(X-tT)$, where $p_g(X) = p(X)+z(X)$ is a generalized noisy observation of p such that $z \in L^2(\mathbb{R}^2)$ is an arbitrary noise pattern. Then, the generalized noisy distance function is

$$g_g(tT) = \int_{\mathbb{R}^2} \big(p(X) - p_g(X - tT)\big)^2 \, dX$$

and the generalized deviation function is $h_g = g_g(tT) - f(tT)$. Let us call u_0U_0 the point where g_g has its global minimum. Then the distance between the global minima of g_g and f is given by u_0.

We begin with the first setting and state a generic bound for the alignment error u_0 in terms of the norm of the noise $\nu := \|z\|$. In our main result, we denote by $R_p := \|p\|$ the norm of the pattern p, and make use of an upper bound $R_{p''}$ for the norm $\|d^2p(X + tT)/dt^2\|$ of the second derivative of $p(X + tT)$. The parameter $R_{p''}$ is derived in terms of the atom parameters of p in [102, Lemma 4]. We state below our generalized alignment error result for arbitrary noise patterns.

Theorem 6. *Let $\bar{t}_0$ be defined as in (6.20). Assume that the norm ν of z is smaller than ν_0 such that*

$$\nu \leq \nu_0 := \frac{\bar{t}_0^2 \underline{r}_0}{8R_p + 2R_{p''}\bar{t}_0^2} \tag{6.23}$$

where $\underline{r}_0$ is the constant in Lemma 4. Then, the distance u_0 between the global minima of f and g_g is bounded as

$$u_0 \leq R_{u_0} := \sqrt{\frac{8R_p\nu}{\underline{r}_0 - 2R_{p''}\nu}}. \tag{6.24}$$

Theorem 6 is proved in [102, Appendix C.2]. The theorem states an upper bound on the alignment error for the general case where the only information used about the noise pattern is its norm. The alignment error bound R_{u_0} is a generalized and deterministic version of the probabilistic bound R_{t_0} derived for the Gaussian noise model. In the proof of the theorem, the change $h_g(0) - h_g(u_0U_0)$ in the distance deviation function is bounded by $4R_p\nu$. The second derivative of the noiseless distance function f is captured by $\underline{r}_0$ as in Section 6.3.1. Finally, the term $2R_{p''}\nu$ bounds the second derivative of the deviation h_g. Based on these, the above result is obtained by following similar steps as in Section 6.3.1.

We now continue with the second setting where the noise pattern z has small correlation with the points on the translation manifold $\mathcal{M}(p)$ of p. We characterize the correlation of two patterns with their inner product. Assume that a uniform correlation upper bound r_{pz} is available such that

$$\left| \int_{\mathbb{R}^2} p(X + tT)z(X)dX \right| \leq r_{pz} \tag{6.25}$$

for all t and T. The following corollary builds on Theorem 6 and states that the bound on the alignment error can be made sharper if the correlation bound is sufficiently small.

Corollary 2. *Let $\bar{t}_0$ be defined as in (6.20) and let a uniform upper bound r_{pz} for the correlation be given such that $r_{pz} < \bar{t}_0^2\underline{r}_0/8$.*

Assume that the norm ν of z is smaller than ν_0 such that

$$\nu \leq \nu_0 := \frac{\bar{t}_0^2 \underline{r}_0 - 8 r_{pz}}{2 R_{p''} \bar{t}_0^2}. \tag{6.26}$$

Then, the distance u_0 between the global minima of f and g_g is bounded as

$$u_0 \leq Q_{u_0} := \sqrt{\frac{8\, r_{pz}}{\underline{r}_0 - 2 R_{p''} \nu}}. \tag{6.27}$$

The proof of Corollary 2 is given in [102, Appendix C.3]. One can observe that the alignment error bound Q_{u_0} approaches zero as the uniform correlation bound approaches zero. Therefore, if r_{pz} is sufficiently small, Q_{u_0} will be smaller than the general bound R_{u_0}. This shows that, regardless of the noise level, the alignment error is close to zero if the noise pattern z is almost orthogonal to the translation manifold $\mathcal{M}(p)$ of the reference pattern.

6.3.3 Influence of filtering on alignment error

In this section, we examine how the alignment error resulting from image noise is affected when the reference and target patterns are low-pass filtered. We consider the Gaussian kernel in (5.6) and analyze the dependence of the alignment error bounds obtained for the Gaussian noise and generalized noise models in Sections 6.3.1 and 6.3.2 on the filter size ρ and the noise level parameters η and ν.

We begin with the Gaussian noise model $w(X)$. The filtered reference pattern $\hat{p}(X)$ and the filtered noisy observation $\hat{p}_n(X)$ of the reference pattern are given by

$$\hat{p}(X) = \sum_{k=1}^{K} \hat{c}_k \phi_{\hat{\gamma}_k}(X), \qquad \hat{p}_n(X) = \hat{p}(X) + \hat{w}(X) = \sum_{k=1}^{K} \hat{c}_k \phi_{\hat{\gamma}_k}(X) + \sum_{l=1}^{L} \hat{\zeta}_l\, \phi_{\hat{\xi}_l}(X).$$

Remember that the rotation and translation parameters of the atoms of $\hat{p}$ do not depend on ρ; and the scale matrices vary with ρ such that $\hat{\sigma}_k^2 = \sigma_k^2 + \Upsilon^2$. The parameters of the smoothed noise atoms can be obtained similarly to the atom parameters of $\hat{p}$; i.e., $\phi_{\hat{\xi}_l}(X) = \phi\big(\hat{E}^{-1}(X - \delta_l)\big)$, where $\hat{E}^2 = E^2 + \Upsilon^2$. This gives the scale parameter of smoothed noise atoms as

$$\hat{\epsilon} = \sqrt{\epsilon^2 + \rho^2}. \tag{6.28}$$

The smoothed noise coefficients are given by

$$\hat{\zeta}_l = \zeta_l \frac{|E|}{|\hat{E}|} = \zeta_l \frac{\epsilon^2}{(\epsilon^2 + \rho^2)}.$$

Since all the coefficients ζ_l are multiplied by a factor of $\epsilon^2/(\epsilon^2 + \rho^2)$, the variance of smoothed noise atom coefficients is

$$\hat{\eta}^2 = \left(\frac{\epsilon^2}{\epsilon^2 + \rho^2}\right)^2 \eta^2. \tag{6.29}$$

As the noise atom units are considered to have very small scale, one can assume that $\rho \gg \epsilon$ for typical values of the filter size ρ. Then the relations in (6.28) and (6.29) give the joint variations of $\hat{\epsilon}$ and $\hat{\eta}$ with η and ρ as

$$\hat{\epsilon} = O(\rho), \qquad \hat{\eta} = O(\eta\rho^{-2}). \tag{6.30}$$

We now state the dependence of the bound $\hat{R}_{t_0}$ on ρ and η in the following main result.

Theorem 7. *The joint variation of the alignment error bound $\hat{R}_{t_0}$ for the smoothed image pair with respect to η and ρ is given by*

$$\hat{R}_{t_0} = O\left(\sqrt{\frac{\eta\,\rho^{-1}}{(1+\rho^2)^{-2} - \eta\,\rho^{-3}}}\right) = O\left(\sqrt{\frac{\eta\,\rho^3}{1-\eta\,\rho}}\right).$$

Therefore, for a fixed noise level, $\hat{R}_{t_0}$ increases at a rate of $O\left(\rho^{3/2}\,(1-\rho)^{-1/2}\right)$ with the increase in the filter size ρ. Similarly, for a fixed filter size, the rate of increase of $\hat{R}_{t_0}$ with the noise standard deviation η is $O\left(\eta^{1/2}\,(1-\eta)^{-1/2}\right)$.

The proof of Theorem 7 is presented in [102, Appendix D.2]. The stated result is obtained by using the relations in (6.30) to determine how the terms $\hat{R}_{\sigma_{\Delta h}}$, $\hat{\underline{r}}_0$, $\hat{R}_{\sigma_{h''}}$ in the expression of $\hat{R}_{t_0}$ vary with ρ and η.

Theorem 7 is the summary of our analysis about the effect of filtering on the alignment accuracy for the Gaussian noise model. While the aggravation of the alignment error with the increase in the noise level is an intuitive result, the theorem states that filtering the patterns under the presence of noise decreases the accuracy of alignment as well. Remember that this is not the case for noiseless patterns. The result of the theorem can be interpreted as follows. Smoothing the reference and target patterns diffuses the perturbation on the distance function, which is likely to cause a bigger shift in the minimum of the distance function and hence reduce the accuracy of alignment. The estimation $\hat{R}_{t_0} = O\left(\rho^{3/2}\,(1-\rho)^{-1/2}\right)$ of the alignment error suggests that the dependence of the error on ρ is between linear and quadratic for small values of ρ, whereas it starts to increase more dramatically when ρ takes larger values. Similarly, $\hat{R}_{t_0}$ is proportional to the square root of η for small η and it increases at a sharper rate as η grows.

Next, we look at the variation of the bounds $\hat{R}_{u_0}$ and $\hat{Q}_{u_0}$ for arbitrary noise patterns, which are respectively obtained for the general and small-correlation cases. We present the following theorem, which is the counterpart of Theorem 7 for arbitrary noise models.

Theorem 8. *The alignment error bounds $\hat{R}_{u_0}$ and $\hat{Q}_{u_0}$ for arbitrary noise patterns have a variation of*

$$O\left(\sqrt{\frac{\nu\,(1+\rho^2)}{1-\nu}}\right) \tag{6.31}$$

with the noise level ν and the filter size ρ. Therefore, for a fixed noise level, the errors $\hat{R}_{u_0}$ and $\hat{Q}_{u_0}$ increase at a rate of $O\left((1+\rho^2)^{1/2}\right)$ with the increase in the filter size ρ. Similarly, for a fixed filter size, $\hat{R}_{u_0}$ and $\hat{Q}_{u_0}$ increase at a rate of $O\left(\nu^{1/2}(1-\nu)^{-1/2}\right)$ with respect to the noise norm ν.

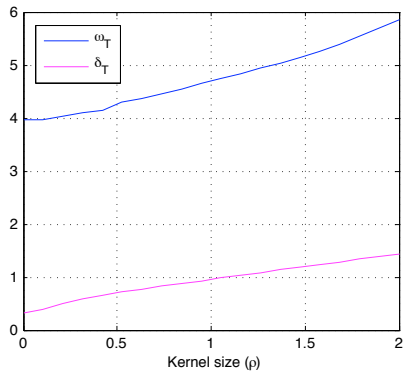


Figure 6.2: The variations of the true distance $\hat{\omega}_T$ of the boundary of $\hat{\mathcal{S}}$ to the origin and its estimate $\hat{\delta}_T$ with respect to the filter size

The proof of Theorem 8 is given in [102, Appendix E.1]. The dependence of the generalized bounds $\hat{R}_{u_0}$ and $\hat{Q}_{u_0}$ on the noise norm ν is the same as the dependence of $\hat{R}_{t_0}$ on η. However, the variation of $\hat{R}_{u_0}$ and $\hat{Q}_{u_0}$ with ρ is seen to be slightly different from that of $\hat{R}_{t_0}$. This stems from the difference between the two models. In the generalized noise model z, we have treated the norm ν of z as a known fixed number and we have characterized the alignment error in terms of ν. On the other hand, w is a probabilistic Gaussian noise model; therefore, it is not possible to bound its norm with a fixed parameter. For this reason, the alignment error for w has been derived probabilistically in terms of the standard deviations of the involved parameters. Since the filter size ρ affects the norm of z and the standard deviations of the terms related to w in different ways, it has a different effect on these two type of alignment error bounds. The reason why the two error bounds have the same kind of dependence on the noise level parameters η and ν can be explained similarly. The standard deviations of the terms related to w have a simple linear dependence on η, which is the same as the dependence of the counterparts of these terms in the generalized model on ν.

6.4 Experimental Results

6.4.1 Evaluation of alignment regularity analysis

We first evaluate our theoretical results about SIDEN estimation with an experiment that compares the estimated SIDEN to the true SIDEN. We generate a reference pattern consisting of 40 randomly selected Gaussian atoms with random coefficients, and choose a random unit direction T for pattern displacement. Then, we determine the distance $\hat{\omega}_T$ of the true SIDEN boundary from the origin along T, and compare it to its estimation $\hat{\delta}_T$ for a range of filter sizes ρ (With an abuse of notation, the parameter denoted as $\hat{\omega}_T$ here corresponds in fact to $\sup \hat{\omega}_T$ in the definition of SIDEN in (6.4)). The distance $\hat{\omega}_T$ is computed by searching the first zero-crossing of $d\hat{f}(tT)/dt$ numerically, while its estimate $\hat{\delta}_T$ is computed according to Theorem 3. We repeat the experiment 300 times with different random reference patterns p and directions T and average the results of the cases where $d\hat{f}(tT)/dt$ has zero-crossings for all values of ρ (i.e., 56% of the tested cases). The distance $\hat{\omega}_T$ and

its estimate $\hat{\delta}_T$ are plotted in Figure 6.2. The figure shows that $\hat{\delta}_T$ has an approximately linear dependence on ρ. This is an expected behavior, since $\hat{\delta}_T = O\left((1+\rho^2)^{1/2}\right) \approx O(\rho)$ for large ρ. The estimate $\hat{\delta}_T$ is smaller than $\hat{\omega}_T$ since it is a lower bound for $\hat{\omega}_T$. Its variation with ρ is seen to capture well the variation of the SIDEN boundary $\hat{\omega}_T$ with ρ.

6.4.2 Evaluation of alignment accuracy analysis

We now present experimental results evaluating the alignment error bounds derived in Section 6.3. We conduct the experiments on reference and target patterns made up of Gaussian atoms, where the target pattern is generated by corrupting the reference pattern with noise and applying a random translation tT. In all experiments, an estimate t_eT_e of tT is computed by aligning the reference and target images with a gradient descent algorithm[4], which gives the experimental alignment error as $\|tT - t_eT_e\|$. The experimental error is then compared to the theoretical bounds derived in Section 6.3.

Gaussian noise model

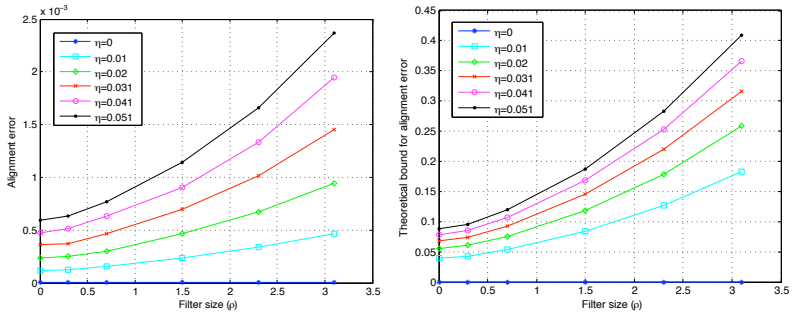
In the first set of experiments, we evaluate the results for the Gaussian noise model. We compare the experimental alignment error to the theoretical bound given in Theorem 5. [5] In all experiments, the probability constant s in Theorem 5 is chosen such that $t_0 < R_{t_0}$ holds with probability greater than 0.5. For each reference pattern, the experiment is repeated for a range of values for noise variances η^2 and filter sizes ρ. The maximum value of the noise standard deviation is taken as the admissible noise level η_0 in Theorem 5.

We first experiment on reference patterns built with 20 Gaussian atoms with randomly chosen parameters. The atom coefficients c_k in the reference patterns are drawn from a uniform distribution in $[-1, 1]$; and the position and scale parameters of the atoms are selected such that $\tau_x, \tau_y \in [-4, 4]$ and $\sigma_x, \sigma_y \in [0.3, 2]$. The noise model parameters are set as $L = 750$, $\epsilon = 0.1$. The experiment is repeated on 50 different reference patterns. Then, 50 noisy target patterns are generated for each reference pattern according to the Gaussian noise model w in (6.12) with a random translation tT in the range $tT_x, tT_y \in [-4, 4]$. The results are averaged over all reference and target patterns. In Figure 6.3, the experimental and theoretical values of the alignment error are plotted with respect to the filter size ρ, where different curves correspond to different η values. Figures 6.3(a) and 6.3(b) show respectively the experimental value $\|tT - t_eT_e\|$ and the theoretical upper bound R_{t_0} of the alignment error. Figure 6.4 shows the same results, where the error is plotted as a function of η. The experimental values and the theoretical bounds are given respectively in Figures 6.4(a) and 6.4(b).

The results in Figure 6.3 show that, although the theoretical upper bound is pessimistic (which is due to the fact that the bound is a worst-case analysis), the variation of the experimental value of the

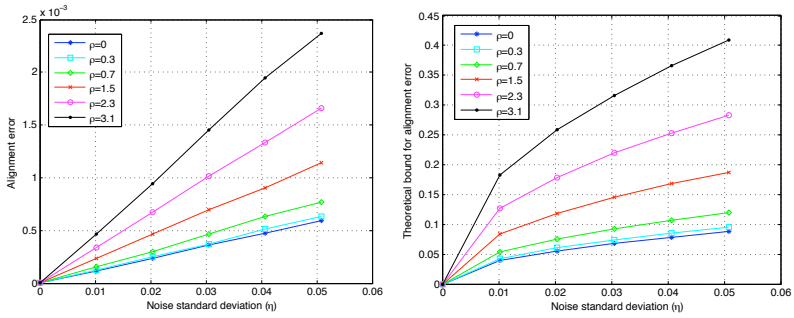
[4] In the computation of t_eT_e, in order to be able to handle large translations, before the optimization with gradient descent we first do a coarse preregistration of the reference and target images with a search on a coarse grid in the translation parameter domain, whose construction is explained in Section 6.4.3.

[5] The bound $R_{\sigma^2_{h''}}$ given in Lemma 5 is derived from the preliminary bound $R_{\sigma^2_{h''(tT)}}$ in [102, Lemma 3]. In the implementation of Theorem 5, in order to obtain a sharper estimate of $R_{\sigma^2_{h''}}$, we compute it by searching the maximum value of $\sigma^2_{h''(tT)}$ over t and T from the expressions for $\mathcal{E}_j$ and $\mathcal{F}_j$ used in the derivation of $R_{\sigma^2_{h''(tT)}}$.



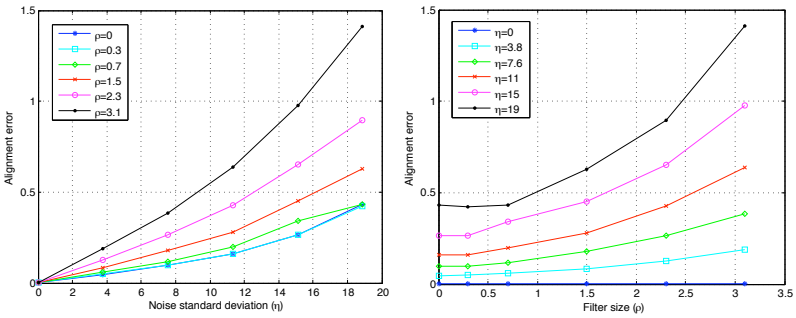
(a) (b)

Figure 6.3: Alignment error of random patterns as a function of filter size ρ.



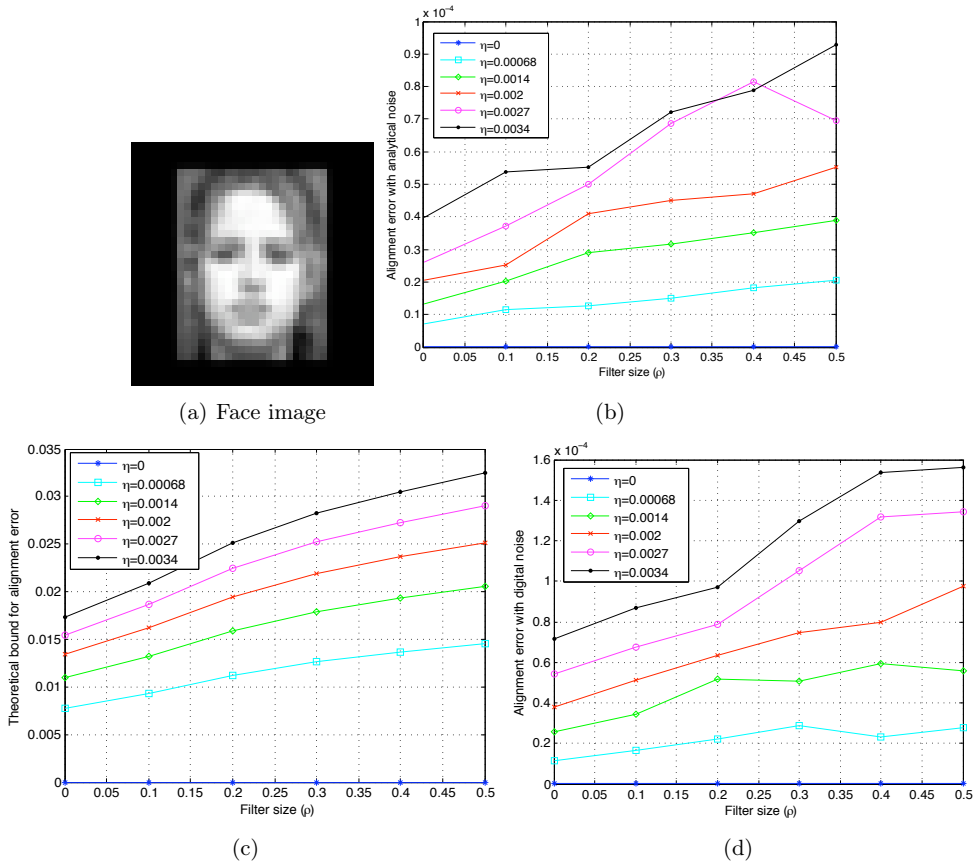
(a) (b)

Figure 6.4: Alignment error of random patterns as a function of noise standard deviation η.



(a) (b)

Figure 6.5: Alignment error of random patterns as functions of the noise standard deviation and the filter size, at high noise levels.

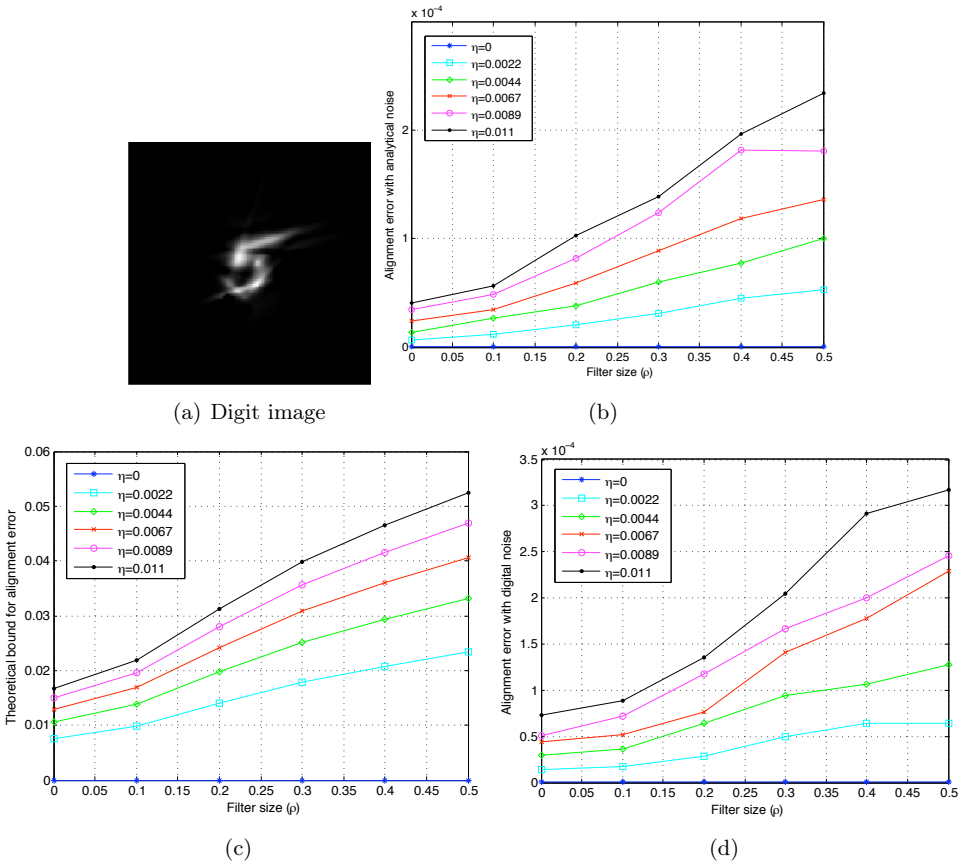


(a) Face image (b)

(c) (d)

Figure 6.6: Face pattern and alignment error as a function of filter size ρ.

alignment error as a function of the filter size is in agreement with that of the theoretical bound. The experimental behavior of the error conforms to the theoretical prediction $\hat{R}_{t_0} \approx O\left(\rho^{3/2}\,(1-\rho)^{-1/2}\right)$ of Theorem 7. Next, the plots of Figure 6.4 suggest that the variation of the theoretical bound R_{t_0} as a function of η is consistent with the result of Theorem 7, which can be approximated as $\hat{R}_{t_0} \approx O(\sqrt{\eta})$ for small values of η. On the other hand, the experimental value of the alignment error seems to exhibit a more linear behavior. However, this type of dependence is not completely unexpected. Theorem 7 predicts that $\hat{R}_{t_0}$ is of $O(\sqrt{\eta})$ for small η; and $O\left(\eta^{1/2}(1-\eta)^{-1/2}\right)$ for large η, while the experimental value of the error can be rather described as $\|tT - t_e T_e\| = O(\eta)$, which is between these two orders of variation. In order to examine the dependence of the error on η in more detail, we have repeated the same experiments with much higher values of η. The experimental alignment error is given in Figure 6.5, where the error is plotted with respect to the noise standard deviation in Figure 6.5(a) and the filter size in Figure 6.5(b). The results show that, at high noise levels, the variation of the error with η indeed increases above the linear rate $O(\eta)$. The noise levels tested in this high-noise experiment are beyond the admissible noise level derived



(a) Digit image (b)

(c) (d)

Figure 6.7: Digit pattern and alignment error as a function of filter size ρ.

in Theorem 5; therefore, we cannot apply Theorem 5 directly in this experiment. However, in view of Theorem 7, which states that the error is of $O\left(\eta^{1/2}(1-\eta)^{-1/2}\right)$, these results can be interpreted to provide a numerical justification of our theoretical finding: at relatively high noise levels, the error is expected to increase with η at a sharply increasing rational function rate above the linear rate. The variation of the error with ρ at high noise levels plotted in Figure 6.5(b) is seen to be similar to that of the previous experiments.

We now evaluate our alignment accuracy results under Gaussian noise on face and digit images. First, the reference face pattern is obtained by approximating the face image shown in Figure 6.6(a) with 50 Gaussian atoms. The average atom coefficient magnitude of the face pattern is $|c| = 0.14$, and the position and scale parameters of its atoms are in the range $\tau_x, \tau_y \in [-0.9, 0.9]$ and $\sigma_x, \sigma_y \in [0.04, 1.1]$. For the digit experiments, the reference pattern shown in Figure 6.7(a) is the approximation of a handwritten "5" digit with 20 Gaussian atoms. The pattern parameters are such that the average atom coefficient magnitude is $|c| = 0.87$, and the position and scale parameters of the atoms are in the range $\tau_x, \tau_y \in [-0.7, 0.7]$, and $\sigma_x, \sigma_y \in [0.05, 1.23]$. The range

of translation values tT_x and tT_y is selected as $[-1, 1]$ in both settings. Two different noise models are tested on both images. First, the target patterns are corrupted with respect to the analytical Gaussian noise model w of (6.12), where the noise parameters are set as $L = 750$, $\epsilon = 0.04$. Then, a digital Gaussian noise model is tested, where the pixels in the discrete representation of the images are corrupted with additive i.i.d. Gaussian noise having the same standard deviation η as w. The digital Gaussian noise model is supposed to be well-approximated by the analytical noise model. Again, 50 target patterns are generated with random translations. The alignment errors are plotted with respect to ρ in Figures 6.6 and 6.7 respectively for the face and digit patterns. The experimental error with the analytical noise model, the theoretical upper bound obtained for the analytical noise model, and the experimental error with the digital noise model are plotted respectively in panels (b), (c) and (d) in both figures. The results are averaged over all target patterns.

The plots in Figures 6.6 and 6.7 show that the experimental and theoretical errors have a similar variation with respect to ρ. The dependence of the error on ρ in these experiments seems to be different from that of the previous experiment of Figure 6.3. Although the theory predicts the variation $\hat{R}_{t_0} = O\left(\rho^{3/2}\left(1-\rho\right)^{-1/2}\right)$, this result is average and approximate. The exact variation of the error with ρ may change between different individual patterns, as the constants of the variation function are determined by the actual pattern parameters. The similarity between the plots for the analytical and digital noise models suggests that the noise model w used in this study provides a good approximation for the digital Gaussian noise, which is often encountered in digital imaging applications. In [102], the alignment error for face and digit patterns is also plotted with respect to η, and the results are similar to those of the previous experiment with random patterns.

Generic noise model

In the second set of experiments we evaluate the results of Theorem 6 and Corollary 2 for the generic noise model z. In each experiment, the target patterns are generated by corrupting the reference pattern p with a noise pattern z and by applying random translations in the range $tT_x, tT_y \in [-4, 4]$. In order to study the effect of the correlation between z and the points on $\mathcal{M}(p)$ on the actual alignment error and on its theoretical bound, we consider two different settings. In the first setting, the noise pattern z is chosen as a pattern that has high correlation with p. In particular, z is constructed with a subset of the atoms used in p with the same coefficients. The general bound R_{u_0} is used in this setting. In the second setting, the noise z is constructed with randomly selected Gaussian atoms so that it has small correlation with p. The bound Q_{u_0} for the small correlation case is used in the second setting, where the correlation parameter r_{pz} in (6.25) is computed numerically for obtaining the theoretical error bound. In both cases, the atom coefficients of z are normalized such that the norm ν of z is below the admissible noise level ν_0. The theoretical bounds[6] are then compared to the experimental errors for different values of the filter size ρ and the noise level ν.

We conduct the experiment on the random patterns used above, in Figures 6.3-6.5. The noise pattern z is constructed with 10 atoms. The average alignment errors are plotted with respect to the filter size ρ in Figure 6.8. The plots in Figure 6.8 show that the variation of the theoretical

[6]We compute the bound for the second derivative of p numerically by minimizing $\|d^2p(X+tT)/dt^2\|$ over t and T. While the bound $R_{p''}$ is useful for the theoretical analysis as it has an open-form expression, the numerically computed bound is sharper.

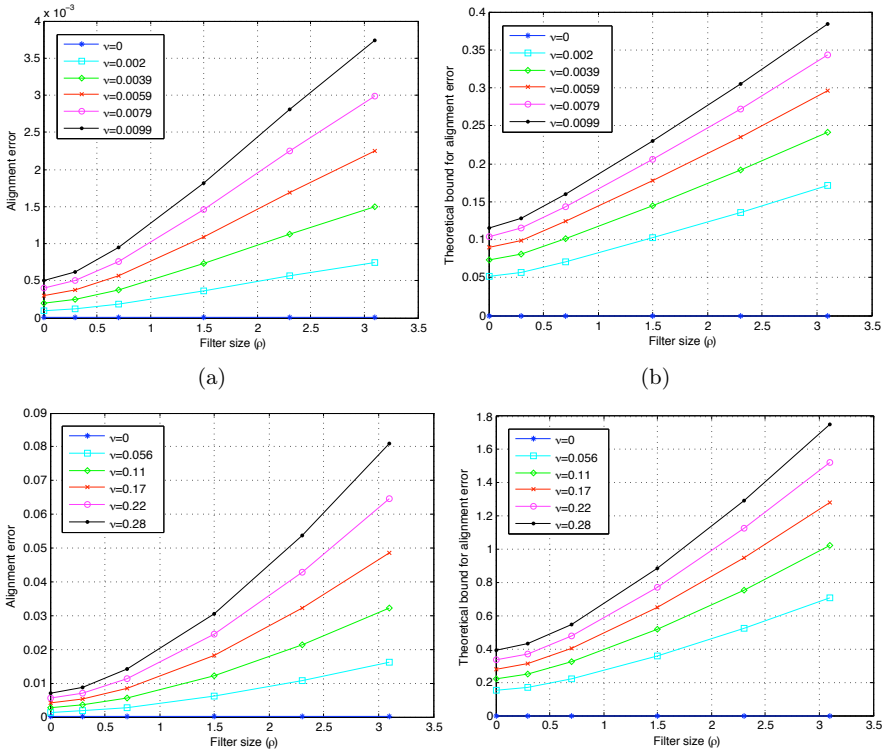


Figure 6.8: Alignment error for random patterns and generic noise, as a function of filter size ρ. (a) and (c) show the error for noise patterns respectively with high and low correlation with p. Corresponding theoretical bounds R_{u_0} and Q_{u_0} are given respectively in (b) and (d).

upper bounds with ρ fits well the behavior of the actual error in both settings. The results are in accordance with Theorem 8, which states that $\hat{R}_{u_0}$ and $\hat{Q}_{u_0}$ are of $O((1+\rho^2)^{1/2})$. The results of the experiment show that $\hat{Q}_{u_0}$ is less pessimistic than $\hat{R}_{u_0}$ as an upper bound since it makes use of the information of the maximum correlation between z and the points on $\mathcal{M}(p)$. Moreover, comparing Figures 6.8(b) and 6.8(d) we see that, when a bound r_{pz} on the correlation is known, the admissible noise level increases significantly (from around $\nu_0 = 0.01$ to $\nu_0 = 0.28$). The plots of the errors with respect to ν are also available in [102]; in short, they show that the variation of the alignment error with ν bears resemblance to its variation with η observed in the previous experiment of Figure 6.4. This is an expected result, as it has been seen in Theorem 8 that the dependences of $\hat{R}_{u_0}$ and $\hat{Q}_{u_0}$ on ν are the same as the dependence of $\hat{R}_{t_0}$ on η. These plots show also that, at the same noise level, the actual alignment error is slightly smaller when z has small correlation with the points on $\mathcal{M}(p)$. In [102], this experiment is also repeated with the face and digit patterns, with results that are similar to those obtained with random test patterns.

The overall conclusion of the experiments is that increasing the filter kernel size results in

a bigger alignment error when the target image deviates from the translation manifold of the reference image due to noise. The results show also that the theoretical bounds for the alignment error capture well the order of dependence of the actual error on the noise level and the filter size, for both the Gaussian noise model and the generalized noise model. Also, the knowledge of the correlation between the noise pattern and the translated versions of the reference pattern is useful for improving the theoretical bound for the alignment error in the general setting.

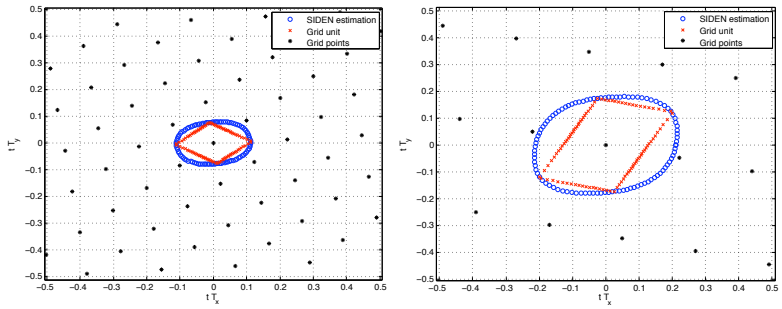
6.4.3 Application: Design of an optimal registration algorithm

We now demonstrate the usage of our SIDEN estimate in the construction of a grid in the translation parameter domain that is used for image registration. In Section 6.2.2, we have derived a set $\mathcal{Q}$ of translation vectors that can be correctly computed by minimizing the distance function with descent methods, where $\mathcal{Q}$ is a subset of the SIDEN $\mathcal{S}$ corresponding to the noiseless distance function. As discussed in the beginning of Section 6.3, in noisy settings, one can assume that $\mathcal{Q}$ is also a subset of the perturbed SIDEN corresponding to the noisy distance function, provided that the noise level is sufficiently small. Therefore the estimate $\mathcal{Q}$ can be used in the registration of both noiseless and noisy images; small translations that are inside $\mathcal{Q}$ can be recovered with gradient descent minimization. However, the perfect alignment guarantee is lost for relatively large translations that are outside $\mathcal{Q}$ and the descent method may terminate in a local minimum other than the global minimum. Hence, in order to overcome this problem, we propose to construct a grid in the translation parameter domain and estimate large translations with the help of the grid. In particular, we describe a grid design procedure such that any translation vector tT lies inside the SIDEN of at least one grid point. Such a grid guarantees the recovery of the translation parameters if the distance function is minimized with a gradient descent method that is initialized with the grid points. In order to have a perfect recovery guarantee, each one of the grid points must be tested. However, as this is computationally costly, we use the following two-stage optimization instead, which offers a good compromise with respect to the accuracy-complexity tradeoff. First, we search for the grid vector that gives the smallest distance between the image pair, which results in a coarse alignment. Then, we refine the alignment with a gradient descent method initialized with this grid vector. In practice, this method is quite likely to give the optimal solution, which has been the case in all of our simulations.

We now explain the construction of the grid. First, notice from (6.5) that $a_{jk}(T) = a_{jk}(-T)$ and $b_{jk}(T) = -b_{jk}(-T)$. Therefore, the function $s_{jk}(t)$ given in (6.9) is the same for T and $-T$ by symmetry. As Q_{jk} does not depend on T, from the form of $df(tT)/dt$ in (6.8) we have $df(tT)/dt = df(-tT)/dt$. Hence, the SIDEN is symmetric with respect to the origin. It is also easy to check that the estimation δ_T of the SIDEN boundary along the direction T satisfies $\delta_T = \delta_{-T}$. One can easily determine a grid unit in the form of a parallelogram that lies completely inside the estimate $\mathcal{Q}$ of the SIDEN and tile the (tT_x, tT_y)-plane with these grid units. This defines a regular grid in the (tT_x, tT_y)-plane such that each point of the plane lies inside the SIDEN of at least one grid point. Note that the complexity of image registration based on a grid search is given by the number of grid points. In our case, the number of grid points is determined by the area of $\mathcal{Q}$; and therefore, the alignment complexity depends on the well-behavedness of the distance function f. In particular, as $V(\mathcal{Q})$ increases with the filter size, the area of the grid units expand at the rate $O(1+\rho^2)$ and the number of grid points decrease at the rate $O\left((1+\rho^2)^{-1}\right)$ with ρ. Therefore, the

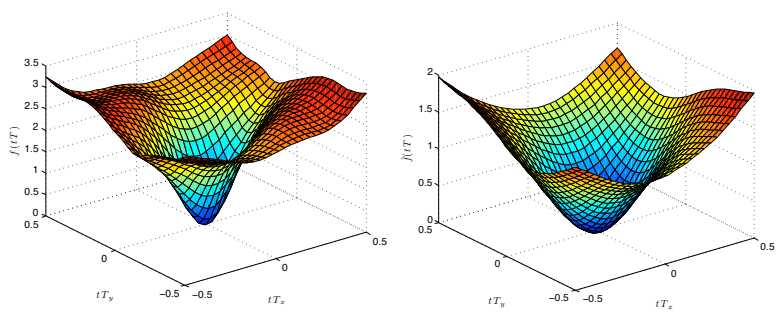
(a) Original pattern and grid images (b) Smoothed pattern and grid images

Figure 6.9: Neighboring grid patterns for the original and smoothed images.



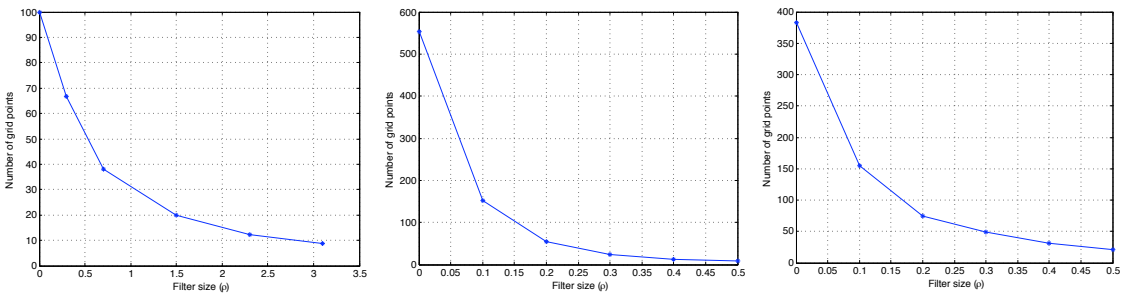
(a) Original grid (b) Smoothed grid

Figure 6.10: Grid construction



(a) Original distance function $f(tT)$ (b) Smoothed distance function $\hat{f}(tT)$

Figure 6.11: The variation of the distance function with smoothing.



(a) Random patterns (b) Face pattern (c) Digit pattern

Figure 6.12: Number of grid points. The decay rate is of $O\left((1+\rho^2)^{-1}\right)$.

alignment complexity with the proposed method is of $O\left((1+\rho^2)^{-1}\right)$.

The construction of a regular grid in this manner is demonstrated for the image of the "5" digit used in the experiments of Section 6.4.2. In Figure 6.9(a), the reference pattern and its translated versions corresponding to the neighboring grid points in the first and second directions of sampling are shown. In Figure 6.9(b), the reference pattern is shown when smoothed with a filter of size $\rho = 0.15$, as well as the neighboring patterns in the smoothed grid. The original grid and the smoothed grid for $\rho = 0.15$ are displayed in Figures 6.10(a) and 6.10(b), where the SIDEN estimates $\mathcal{Q}$, $\hat{\mathcal{Q}}$ and the grid units are also plotted. One can observe that smoothing the pattern is helpful for obtaining a coarser grid that reduces the computational complexity of image registration in hierarchical methods. The corresponding distance functions $f(tT)$ and $\hat{f}(tT)$ are plotted in Figure 6.11, which shows that smoothing eliminates undesired local extrema of the distance function and therefore expands the SIDEN.

Then, in order to demonstrate the relation between alignment complexity and filtering, we build multiscale grids corresponding to different filter sizes and plot the variation of the number of grid points with the filter size. The results obtained with the random patterns and the face and digit patterns used in Section 6.4.2 are presented in Figure 6.12. The results show that the number of grid points decreases monotonically with the filter size, as predicted by Theorem 4, which suggests that the number of grid points must be of $O\left((1+\rho^2)^{-1}\right)$.

Finally, we remark that the performance guarantee of this two-stage registration approach is confirmed by the experiments of Section 6.4.2, which use this registration scheme. In the plots of Figures 6.3-6.8, where the coarse alignment in each experiment is done with the help of a grid adapted to the filter size using the grid design procedure explained above, we see that the proposed registration technique results in an alignment error of 0 for the noiseless case ($\eta = 0$ or $\nu = 0$) for all values of the filter size ρ. These alignment error results, together with the grid size plots of Figure 6.12, show that increasing the filter size reduces the alignment complexity while retaining the perfect alignment guarantee in the noiseless case. Figures 6.3-6.8 show also that the proposed grid can be successfully used in noisy settings. The alignment error in these experiments stems solely from the change in the global minimum of the distance function due to noise, and not from the grid; otherwise, we would observe much higher alignment errors that are comparable to the distance between neighboring grid points. This confirms that the estimate $\mathcal{Q}$ remains in the

perturbed SIDEN and its usage does not lead to an additional alignment error if the noise level is relatively small.

6.5 Discussion of Results

The results of our analysis show that smoothing improves the regularity of alignment by increasing the range of translation values that are computable with descent-type methods. However, in the presence of noise, smoothing has a negative influence on the accuracy of alignment as it amplifies the alignment error caused by the image noise; and this increases with the increase in the filter size. Therefore, considering the computation cost - accuracy tradeoff, the optimal filter size in image alignment with descent methods must be chosen by taking into account the deviation between the target pattern and the translation manifold of the reference pattern; i.e., the expected noise level.

Our study constitutes a theoretical justification of the principle behind hierarchical registration techniques that use local optimizers. Coarse scales are favorable at the beginning of the alignment as they permit the computation of large translation amounts with low complexity using simple local optimizers; however, over-filtering decreases the accuracy of alignment as the target image is in general not exactly a translated version of the reference image. This is compensated for at finer scales where less filtering is applied, thus avoiding the amplification of the alignment error resulting from noise. Since images are already roughly registered at coarse scales, at fine scales the remaining increment to be added to the translation parameters for fine-tuning the alignment is small; it can be achieved at a relatively low complexity in a small search region.

We now interpret the findings of our work in comparison with some previous results. We start with the article [27] by Robinson et al., which studies the Cramér-Rao lower bound (CRLB) of the registration. Since the CRLB is related to the inverse of the Fisher information matrix (FIM) J, the authors suggest to use the trace of J^{-1} as a general lower bound for the MSE of the translation estimation. Therefore, the square root of $tr(J^{-1})$ can be considered as a lower bound for the alignment error. It has been shown in [27] that $\sqrt{tr(J^{-1})} = O(\eta)$, where η is the standard deviation of the Gaussian noise. In fact, this result tells that the alignment error with any estimator is lower bounded by $O(\eta)$, i.e., its dependence on the noise level is at least linear. Meanwhile, our study, which focuses on estimators that minimize the SSD with local optimizers, concludes that the alignment error is at most $O(\sqrt{\eta/(1-\eta)})$ for these estimators. Notice that for small η, $O(\sqrt{\eta/(1-\eta)}) \approx O(\sqrt{\eta}) > O(\eta)$; and for large η, we still have $O(\sqrt{\eta/(1-\eta)}) > O(\eta)$ due to the sharply increasing rational function form of the bound. Therefore, the result in [27] and our results are consistent and complementary, pointing together to the fact that the error of an estimator performing a local optimization of the SSD must lie between $O(\eta)$ and $O(\sqrt{\eta/(1-\eta)})$. Note also that, as it has been seen in the experiments of Figure 6.5(a), the error of this type of estimators may indeed increase with η at a rate above $O(\eta)$ in practice, as predicted by our upper bound. Next, as for the effect of filtering on the estimation accuracy, the authors of [27] experimentally observe that $tr(J^{-1})$ decreases as the image bandwidth increases, which suggests that the lower bound on the MSE of a translation estimator is smaller when the image has more high-frequency components. This is stated more formally in [40]. It is shown that the estimation of the x component of the translation has variance larger than $\eta^2/\|(\partial p/\partial x)^2\|^2$, and similarly for the y component, where $\partial p/\partial x$ is the partial derivative of the pattern p with respect to the spatial

variable x.[7] Therefore, as smoothing decreases the norm of the partial derivatives of the pattern, it leads to an increase in the variance of the estimation. These observations are also consistent with our theoretical results.

Next, we discuss some results from the recent article [37], which presents a continuous-domain noise analysis of block-matching. The blocks are assumed to be corrupted with additive Gaussian noise and the disparity estimate is given by the global minimum of the noisy distance function as in our work. Although there are differences in their setting and ours, such as the horizontal and non-constant disparity field assumption in [37], it is interesting to compare their results with ours. We consider the disparity of the block to be constant in [37], such that it fits our global translation assumption. In [37], an analysis of the deviation between the estimated disparity and the true disparity is given, which is similar to the distance t_0 between the global minima of f and g in our work. Sticking to the notation of this chapter, let us denote this deviation by t_0. In Theorem 3.2 of [37], t_0 is estimated as the sum of three error terms, where the variances of the first and second terms are respectively of $O(\eta^2)$ and $O(\eta^4)$ with respect to the noise standard deviation η. These two terms are stated as dominant noise terms. Then, the third term represents the high-order Taylor terms of some approximations made in the derivations, which however depends on the value of t_0 itself. As the overall estimation of t_0 is formulated using the term t_0 itself, their main result is interesting especially for small values of t_0, since the third term is then negligible. It can be concluded from [37] that $t_0 \approx O(\eta + \eta^2 + H.O.)$, where $H.O.$ represents high-order terms. This result is consistent with the CRLB of t_0 in [27] stating that t_0 is at least of $O(\eta)$, and our upper bound $O(\sqrt{\eta}/\sqrt{1-\eta})$. The analysis in [37] and ours can be compared in the following way. First, since our derivation is rather rigorous and does not neglect high-order terms, these terms manifest themselves in the rational function form of the resulting bound. Meanwhile, they are represented as $H.O.$ and not explicitly examined in the estimation $O(\eta + \eta^2 + H.O.)$ in [37]. For small η, this result can be approximated as $t_0 = O(\eta)$, while our result states that $t_0 \leq O(\sqrt{\eta})$. As $\sqrt{\eta} > \eta$ for small η, this also gives a consistent comparison since our estimation is an upper bound and the one in [37] is not. Indeed, the experimental results in [37] suggest that their derivation gives a slight underestimation of the error. Lastly, our noise analysis treats the image alignment problem in a multiscale setting and analyzes the joint variation of the error with the noise level η and the filter size ρ, whereas the study in [37] only concentrates on the relation between the error and the noise level.

Finally, we mention some facts from scale-space theory [98], which may be useful for the interpretation of our findings regarding the variation of the SIDEN with the filter size ρ. The scale-space representation of a signal is given by convolving it with kernels of variable scale. The most popular convolution kernel is the Gaussian kernel, as it has been shown that under some constraints, it is the unique kernel for generating a scale-space. An important result in scale-space theory is [103], which states that the number of local extrema of a 1-D function is a decreasing function of ρ. This provides a mathematical characterization of the well-known smoothing property of the Gaussian kernel. However, it is known that this cannot be generalized to higher-dimensional signals; e.g., there are no nontrivial kernels on $\mathbb{R}^2$ with the property of never introducing new local extrema when the scale increases [98]. One interesting result that can possibly be related to our analysis is

[7] This bound is obtained by assuming Gaussian noise on both reference and target patterns and deriving the CRLB.

about the density of local extrema of a signal as a function of scale. In order to gain an intuition about the behavior of local extrema, the variation of the local extrema is examined in [98] for 1-D continuous white noise and fractal noise processes. It has been shown that the expected density of the local minima of these signals decreases at rate ρ^{-1}. In the estimation of the SIDEN in our work, we have analyzed how the first zero crossing of $d\hat{f}(tT)/dt$ along a direction T around the origin varies with the scale. Therefore, what we have examined is the distance $\hat{f}$ between the scale space $\hat{p}(X)$ of an image and the scale space $\hat{p}(X - tT)$ of its translated version. Since this is different from the scale-space of the distance function f itself, it is not possible to compare the result in [98] directly to ours. However, we can observe the following. Restricting $\hat{f}$ to a specific direction T_a so that we have a 1-D function $\hat{f}(tT_a)$ of t as in [98], our estimation for the stationary point of $\hat{f}(tT_a)$ closest to the origin expands at a rate of $O((1 + \rho^2)^{1/2})$, which is $O(\rho)$ for large ρ. One can reasonably expect this distance to be roughly inversely proportional to the density of the local extrema of $\hat{f}$. This leads to the conclusion that the density of the distance extrema is expected to be around $O(\rho^{-1})$, which interestingly matches the density obtained in [98].

6.6 Conclusion

In this chapter, we have presented a theoretical analysis of image alignment with descent-type local minimizers, where we have specifically focused on the effect of low-pass filtering and noise on the regularity and accuracy of alignment. First, we have examined the problem of aligning with gradient descent a reference and a target pattern that differ by a two-dimensional translation. We have derived a lower bound for the range of translations for which the reference pattern can be exactly aligned with its translated versions, and investigated how this region varies with the filter size when the images are smoothed. Our finding is that the volume of this region increases quadratically with the filter size, showing that smoothing the patterns improves the regularity of alignment. Then, we have considered a setting with noisy target images and examined Gaussian noise and arbitrary noise patterns, which may find use in different imaging applications. We have derived a bound for the alignment error and searched the dependence of the error on the noise level and the filter size. Our main results state that the alignment error bound is proportional to the square root of the noise level at small noise, whereas this order of dependence increases at larger noise levels. More interestingly, the alignment error is also significantly affected by the filter size. The probabilistic error bound obtained with the Gaussian noise model has been seen to increase with the filter size at a sharply increasing rational function rate, whereas the deterministic bound obtained for arbitrary noise patterns of deterministic norm increases approximately linearly with the filter size. These theoretical findings are also confirmed by experiments. To the best of our knowledge, none of the previous works about image registration has studied the alignment regularity problem. Meanwhile, our alignment accuracy analysis is consistent with previous results, provides a more rigorous treatment, and studies the problem in a multiscale setting unlike the previous works. The results of our study show that, in multiscale image registration, filtering the images with large filter kernels improves the alignment regularity in early phases, while the use of smaller filters improves the accuracy at later phases. From this aspect, our estimations of the regularity and accuracy of alignment in terms of the noise and filter parameters provide insight for the principles behind hierarchical registration techniques and are helpful for the design of efficient,

low-complexity registration algorithms.

Chapter 7

Conclusions

In this thesis we have studied several problems related to the analysis of image sets with geometric transformations using parametric models. We have mostly focused on the registration and transformation-invariant classification of visual data with manifold models. We have examined the sampling and learning of transformation manifolds and derived some performance bounds for the multiscale registration of images with globally parametrizable transformations.

We have first addressed the problem of sampling transformation manifolds. Since solving the manifold distance computation problem is computationally complex, we have presented a constructive solution based on sampling data-representative transformation manifolds offline such that the resulting sample sets give an accurate estimation of the manifold distance. We have proposed two manifold discretization algorithms, which promote registration and classification accuracy with the selected sample set. An important observation is that the joint consideration of the relative structures of different class-representative manifolds in the sampling improves the classification performance significantly, compared to the individual sampling of each manifold. The sampling of manifolds has not been studied before in the context of image analysis and grid construction has typically been achieved with straightforward approaches such as uniform sampling. However, our results show that there is room for attaining much better performance in data analysis applications by using more sophisticated sampling schemes such as the methods proposed in this thesis.

Next, we have studied the learning of pattern transformation manifolds (PTMs) from a given set of image data with geometric transformations. We have presented two PTM building methods for image approximation and supervised image classification. The proposed methods are based on a greedy construction of patterns representing image sets or image classes through the selection of atoms from a parametric dictionary. The proposed algorithms use the knowledge of the type of geometric transformation that generates the data when learning a parametric model. This brings several benefits in comparison with classical manifold learning methods, such as robustness to the conditions of the input data (e.g., small number of data samples, data noise), data parameterization with a known physical meaning, and flexibility to synthesize novel data samples. The presented methods learn a representative pattern and estimate the individual transformation parameters of input images simultaneously. An alternative approach for solving the manifold learning problem can be to first align the input images with respect to a suitable reference image and then learn a model from the aligned images. However, experimental results show that the proposed idea of

simultaneous learning and aligning gives better results in image approximation and classification.

We have finally considered the multiscale alignment of images via globally parametrizable warping functions and performed a theoretical analysis of image registration performance. We have focused on two settings, where we have first examined the alignment of an image pair with the tangent distance method for arbitrary transformation models, and then studied the estimation of the 2-D translation between an image pair with descent-type local optimizers. In both settings, we have derived upper bounds for the alignment error and examined the variation of the alignment error with the image noise level and the size of the low-pass filter used for smoothing the images in multiscale alignment. The main results of our study show that smoothing has the desired effect of improving the manifold linearity in the first setting and the well-behavedness of the distance function in the second setting. However, in both settings smoothing leads to an aggravation of the alignment error resulting from image noise. This shows that, in both registration settings, the optimal filter size must be chosen by taking the expected noise level into account. The analyses of image registration with tangent distance and local optimizers presented in this thesis differ from previous works in that they provide a joint formulation of the alignment error in terms of the noise level and the filter size and give expressions for the rate of variation of the error in both settings. Another main contribution of the thesis is that it presents a first study of the well-behavedness of the dissimilarity function in alignment problems. In particular, it reports that the area of the region in the translation parameter domain where the image dissimilarity function is free of undesired local minima expands at least at a quadratic rate with respect to the size of the low-pass filter.

To sum up, this thesis proposes several approaches and ideas for benefiting from parametrizable models in an effective way in order to tackle the challenges arising from geometric transformations in the analysis of image sets. In particular, the main contributions of the thesis can be summarized as the development of novel image analysis methods designed particularly for geometrically transformed image data sets, and new insights into the performance limits of popular image alignment methods, which are helpful for employing these techniques more efficiently towards achieving transformation-invariance in image analysis.

The thesis opens a few future directions of research. First, the manifold sampling methods that we have presented rely on the assumption that the total number of samples to be selected from the manifolds is fixed and given a priori. However, in a sampling scenario where a predefined registration or classification accuracy constraint is to be met, the number of samples needs to be determined with respect to performance criteria. Therefore, it is important to investigate the relations between the registration and classification accuracy and the sampling density, or the number of manifold samples. The sampling density and the precision of the manifold representation are expected to be linked by the geometric properties of the manifold such as its nonlinearity. However, contrary to our initial intuition, we have observed that the curvature of the manifold may fail to characterize the registration accuracy of the sampling. This is because curvature is a very local property, whereas the accuracy of the estimation of the manifold distance with a sample set depends on the variation of the manifold geometry at a relatively large scale. One may perhaps consider analyzing the nonlinearity of smoothed versions of the manifold rather than the original manifold to understand the relation between registration accuracy and curvature.

Then, we have constructed our PTM learning methods only with generic dictionaries, which are derived from Gaussian or multiquadric mother functions. However, one can possibly obtain better performance with dictionaries adapted to the type of images under construction. Therefore,

an extension of the presented study is the learning of parametric dictionaries that facilitate the approximation and classification of signals with manifold models. Furthermore, the manifolds considered in this thesis are defined in the original image space. This approach gives a good classification performance if the dissimilarity between data samples can be well identified with the ambient space distance. If this is not the case, one usually seeks alternative data representations by changing the basis, projecting the data to suitable subspaces or treating the data in a feature space. Hence, the generalization of the ideas used in this thesis to the construction of class-representative parametric models based on arbitrary data representations is an interesting open problem.

Next, our analysis of the tangent distance method focuses on its registration accuracy. The extension of the presented study to examine the classification accuracy of this method remains as future work. In particular, the analysis of the influence of filtering on the classification performance of tangent distance would provide useful insights in many image analysis and pattern recognition applications. Finally, in our performance analysis of multiscale image registration with local optimizers, we have only considered the geometric transformation model of 2-D translations. The influences of smoothing on the alignment regularity and the alignment accuracy of descent methods are likely to be dependent on the global geometric transformation model. Therefore, a future research direction resides in the extension of our multiscale registration analysis to cover other transformation models as well. Such future efforts would generalize and extend the approaches presented in this thesis for applicability to a wider range of data representations and models, towards the development of strong and robust tools for the analysis of image data sets.

Appendix A

Appendix

A.1 Proof of Proposition 2

Before showing the DC property of the objective function, we list below some useful properties of DC functions from [87] and [18].

Proposition 6. *(a) Let $\{f_i\}_{i=1}^m$ be a set of DC functions with decompositions $f_i = g_i - h_i$ and let $\{\lambda_i\}_{i=1}^m \subset \mathbb{R}$. Then $\sum_{i=1}^m \lambda_i f_i$ has the following DC decomposition [87], [18].*

$$\sum_{i=1}^{m} \lambda_i f_i = \left(\sum_{i:\lambda_i>0} \lambda_i g_i - \sum_{i:\lambda_i<0} \lambda_i h_i \right) - \left(\sum_{i:\lambda_i>0} \lambda_i h_i - \sum_{i:\lambda_i<0} \lambda_i g_i \right)$$

(b) Let $f_1 = g_1 - h_1$ and $f_2 = g_2 - h_2$ be DC functions with nonnegative convex parts g_1, h_1, g_2, h_2. Then the product $f_1 f_2$ has the DC decomposition

$$f_1 f_2 = \frac{1}{2}\left((g_1+g_2)^2 + (h_1+h_2)^2\right) - \frac{1}{2}\left((g_1+h_2)^2 + (g_2+h_1)^2\right),$$

which has nonnegative convex parts [87], [18].

Now we can give a proof of Proposition 2.

Proof: Let $\tilde{e} = v - cU_\lambda(\phi_\gamma)$ denote the difference vector between an image v and an atom ϕ_γ transformed by λ with a coefficient c. We first show that the components (pixels) of $U_\lambda(\phi_\gamma)$ are DC functions of γ and c. Remember that $U_\lambda(\phi_\gamma)$ has been defined as a discretization of $A_\lambda(\phi_\gamma) = A_\lambda(B_\gamma(\phi))$. We can write

$$A_\lambda(B_\gamma(\phi))(X) = B_\gamma(\phi)(X') = \phi(\tilde{X})$$

where all coordinate variables are related by

$$\tilde{X} = b_\gamma(X') = (b_\gamma \circ a_\lambda)(X).$$

Then, we get

$$A_\lambda(B_\gamma(\phi))(X) = \phi(b_\gamma(a_\lambda(X))).$$

The l-th component $U_\lambda(\phi_\gamma)(l)$ of $U_\lambda(\phi_\gamma)$ corresponds to a certain point with coordinate vector X such that

$$U_\lambda(\phi_\gamma)(l) = A_\lambda(B_\gamma(\phi))(X) = \phi(b_\gamma(a_\lambda(X))).$$

Here, b is a smooth function of γ, and ϕ is also a smooth function. Therefore, being a composition of two smooth functions, $U_\lambda(\phi_\gamma)(l)$ is smooth as well ([104], Corollary 7.2), and thus DC by Proposition 1.

In the following, we show the DC property of $\tilde{E}$. We describe at the same time a procedure to compute the DC decomposition of $\tilde{E}$ if a DC decomposition of $U_\lambda(\phi_\gamma)(l)$ is available. We expand $\|\tilde{e}\|^2$ in terms of the errors at individual pixels as

$$\|\tilde{e}\|^2 = \|v - c\,U_\lambda(\phi_\gamma)\|^2 = \sum_{l=1}^{n} \left(v^2(l) - 2\,v(l)\,c\,U_\lambda(\phi_\gamma)(l) + c^2\,U_\lambda^2(\phi_\gamma)(l)\right) \tag{A.1}$$

where $v(l)$ is the l-th component of v. The term $v(l)$ is constant with respect to γ and c. Using the DC decomposition of $U_\lambda(\phi_\gamma)(l)$ and decomposing c as $c = 0.5\,(c+1)^2 - 0.5\,(c^2+1)$, we can compute the DC decomposition of the second term $-2\,v(l)\,c\,U_\lambda(\phi_\gamma)(l)$ from Propositions 6.b and 6.a. One can also obtain the decomposition of the last term $c^2\,U_\lambda^2(\phi_\gamma)(l)$ by applying Proposition 6.b. Finally, the DC decompositions of $\|\tilde{e}\|^2$ and

$$\tilde{E} = \sum_{i=1}^{N} \|\tilde{e}_i\|^2 \tag{A.2}$$

simply follow from Proposition 6.a. □

A.2 Derivation of total squared tangent distance $\hat{E}$

The first order approximation of the manifold $\mathcal{M}(p_j)$ around the point $U_{\lambda_i}(p_j)$ is given by

$$\mathcal{M}(p_j) \approx \mathcal{S}_i(p_j) = \{U_{\lambda_i}(p_j) + T_i\,z : z \in \mathbb{R}^{d\times 1}\}$$

where T_i is an $n \times d$ matrix consisting of tangent vectors. The k-th column of T_i is the tangent vector that is the derivative of the manifold point $U_{\lambda_i}(p_j)$ with respect to the k-th transformation parameter $\lambda_i(k)$. The orthogonal projection of u_i on $\mathcal{S}_i(p_j)$ is given by $\hat{u}_i = U_{\lambda_i}(p_j) + T_i z^*$, where the coefficient vector z^* of the projection is

$$z^* = (T_i^T\,T_i)^{-1}\,T_i^T(u_i - U_{\lambda_i}(p_j)).$$

Hence, the difference vector $\hat{e}_i$ between u_i and $\hat{u}_i$ is

$$\hat{e}_i = u_i - \hat{u}_i = u_i - U_{\lambda_i}(p_j) - T_i(T_i^T\,T_i)^{-1}\,T_i^T(u_i - U_{\lambda_i}(p_j)).$$

Letting $w_i = u_i - U_{\lambda_i}(p_j)$, we get $\hat{E}$ as

$$\hat{E} = \sum_{i=1}^{N} \|w_i - T_i(T_i^T T_i)^{-1} T_i^T w_i\|^2. \tag{A.3}$$

A.3 Computation of the DC Decompositions in Section 4.2.3

In the derivation of the DC decompositions of $U_\lambda(\phi_\gamma)(l)$ and $\tilde{E}$, we build on the results from [18], where a DC decomposition of the distance between a query pattern and the 4-dimensional transformation manifold of a reference pattern is derived. We first give the following results from [18].

Proposition 7. *(a) Let $f : \mathbb{R}^s \to \mathbb{R}$ be a DC function with decomposition $f(x) = g(x) - h(x)$ and $q : \mathbb{R} \to \mathbb{R}$ be a convex function. Then $q(f(x))$ is DC and has the decomposition*

$$q(f(x)) = p(x) - K[g(x) + h(x)],$$

where $p(x) = q(f(x)) + K[g(x) + h(x)]$ is a convex function and K is a constant satisfying $K > |q'(f(x))|$.

(b) Let $\psi \in [0, 2\pi)$ and $\sigma \in \mathbb{R}^+$. Then the following functions have DC decompositions with nonnegative convex parts: $\cos(\psi)$, $\sin(\psi)$, $\frac{\cos(\psi)}{\sigma}$, $\frac{\sin(\psi)}{\sigma}$ (see [18] for computation details).

The relation between the transformed atom and the mother Gaussian function is given by the change of variables

$$A_\lambda(\phi_\gamma)(X) = \phi(\tilde{X}) = \sqrt{\frac{2}{\pi}} e^{-(\tilde{x}^2 + \tilde{y}^2)}$$

where $\tilde{X} = [\tilde{x}\ \tilde{y}]^T$. Let the l-th pixel of $U_\lambda(\phi_\gamma)$ correspond to the coordinate vector X in $A_\lambda(\phi_\gamma)$ and the coordinate vector $\tilde{X}$ in ϕ. From (4.10) and (2.10), $(\tilde{x}, \tilde{y})$ can be derived in the form

$$\begin{aligned}
\tilde{x} &= \nu \frac{\cos(\psi)}{\sigma_x} + \xi \frac{\sin(\psi)}{\sigma_x} - \frac{\cos(\psi)\tau_x}{\sigma_x} - \frac{\sin(\psi)\tau_y}{\sigma_x} \\
\tilde{y} &= \xi \frac{\cos(\psi)}{\sigma_y} - \nu \frac{\sin(\psi)}{\sigma_y} - \frac{\cos(\psi)\tau_y}{\sigma_y} + \frac{\sin(\psi)\tau_x}{\sigma_y}
\end{aligned}$$

where

$$\begin{aligned}
\nu &= \frac{\cos(\theta)x + \sin(\theta)y - \cos(\theta)t_x - \sin(\theta)t_y}{s_x} \\
\xi &= \frac{-\sin(\theta)x + \cos(\theta)y + \sin(\theta)t_x - \cos(\theta)t_y}{s_y}.
\end{aligned}$$

Here ν and ξ are functions of the transformation parameters λ and the coordinates (x, y) but they stay constant with respect to the atom parameters γ. Now we explain how the coordinate variables $\tilde{x}$ and $\tilde{y}$ can be expanded in the DC form in terms of the atom parameter variables.

First, from Proposition 7.b, notice that the decompositions of the functions $\{\frac{\cos(\psi)}{\sigma_x}, \frac{\sin(\psi)}{\sigma_x}, \frac{\cos(\psi)}{\sigma_y}, \frac{\sin(\psi)}{\sigma_y}\}$ can be computed as explained in [18]. The first two terms in $\tilde{x}$ and $\tilde{y}$ are given by the product of one of these functions with a constant term (ν or ξ). Therefore, we can get the decompositions of these terms using Proposition 6.a.

Then, observe that τ_x has a decomposition as $\tau_x = 0.5\,(\tau_x + 1)^2 - 0.5\,(\tau_x^2 + 1)$ [18] and the decomposition of τ_y is obtained in the same manner. Thus, one can obtain the DC decompositions of the last two terms $\{\frac{\cos(\psi)\tau_x}{\sigma_x}, \frac{\sin(\psi)\tau_y}{\sigma_x}, \frac{\cos(\psi)\tau_y}{\sigma_y}, \frac{\sin(\psi)\tau_x}{\sigma_y}\}$ in $\tilde{x}$ and $\tilde{y}$ by applying the product property in Proposition 6.b on the decompositions of the terms in $\{\frac{\cos(\psi)}{\sigma_x}, \frac{\sin(\psi)}{\sigma_x}, \frac{\cos(\psi)}{\sigma_y}, \frac{\sin(\psi)}{\sigma_y}\}$ and $\{\tau_x, \tau_y\}$. Hence, having computed the decompositions of all additive terms in $\tilde{x}$ and $\tilde{y}$, one can obtain the decompositions of $\tilde{x}$ and $\tilde{y}$ from Proposition 6.a.

Let $\tilde{z} = \tilde{x}^2 + \tilde{y}^2$. The decompositions of $\tilde{x}^2$ and $\tilde{y}^2$ can be obtained by applying the product property in Proposition 6.b on the decompositions of $\tilde{x}$ and $\tilde{y}$. Then, the decomposition of $\tilde{z}$ follows from Proposition 6.a. Expressing the mother function

$$\phi(\tilde{X}) = \sqrt{\frac{2}{\pi}} e^{-\tilde{z}}$$

as a convex function of $\tilde{z}$, Proposition 7.a provides the decomposition of $\phi(\tilde{X})$. Thus, we obtain the decomposition of $U_\lambda(\phi_\gamma)(l)$.

After this point, the decomposition of $\tilde{E}$ can be computed based on the description in Appendix A.1. However, notice that for this special case of Gaussian mother function, the decomposition of the term $U_\lambda^2(\phi_\gamma)(l)$ in (A.1) can also be obtained by writing $A_\lambda^2(\phi_\gamma)(X) = 2/\pi \exp(-2\,\tilde{z})$, and using the decomposition of $\tilde{z}$ and Proposition 7.a.

Now, let us describe the computation of the DC decomposition for the inverse multiquadric mother function $\phi(X) = (1+x^2+y^2)^\mu,\ \ \mu < 0$. Notice that the decomposition of the terms $\tilde{x}$ and $\tilde{y}$ depend only on the PTM and dictionary models (4.10) and (2.10); therefore, they are the same as in the previous case. Since $\phi(\tilde{X}) = (1 + \tilde{z})^\mu$ is a convex function of $\tilde{z}$ for $\tilde{z} \geq 0$, the decomposition of $\phi(\tilde{X})$ can be obtained using Proposition 7.a. (Although the domain of the function q is $\mathbb{R}$ in Proposition 7.a, an examination of the proof in [18] shows that the property can still be applied for a convex function q defined on a domain that is a subset of $\mathbb{R}$.) This gives the decomposition of $U_\lambda(\phi_\gamma)(l)$. The decomposition of $U_\lambda^2(\phi_\gamma)(l)$ can be similarly computed by applying Proposition 7.a to the function $\phi^2(\tilde{X}) = (1 + \tilde{z})^{2\mu}$. Then, the computation of $\tilde{E}$ is the same as in the previous case.

A.4 Proof of Proposition 3

Proof: We rearrange $\tilde{E}$ in the following form

$$\tilde{E} = \sum_{m=1}^{M} \sum_{i=1}^{N_m} (1 + \alpha\, \eta_i^m)\, \|v_i^m - c^m\, U_{\lambda_i^m}(\phi_{\gamma^m})\|^2 - \sum_{m=1}^{M} \sum_{(i,k)\in R^m} \alpha\, \eta_i^k\, \|v_i^{k,m} - c^m\, U_{\lambda_i^{k,m}}(\phi_{\gamma^m})\|^2. \tag{A.4}$$

Equivalently, one can write

$$\tilde{E} = \sum_{m=1}^{M} \tilde{E}^m(\gamma^m, c^m)$$

where

$$\tilde{E}^m(\gamma^m, c^m) = \sum_{i=1}^{N_m} (1 + \alpha\, \eta_i^m)\, \|v_i^m - c^m\, U_{\lambda_i^m}(\phi_{\gamma^m})\|^2 - \sum_{(i,k)\in R^m} \alpha\, \eta_i^k\, \|v_i^{k,m} - c^m\, U_{\lambda_i^{k,m}}(\phi_{\gamma^m})\|^2.$$

From Proposition 6.a, in order to show that $\tilde{E}$ is a DC function of γ and c, it is enough to show that $\tilde{E}^m(\gamma^m, c^m)$ is a DC function of γ^m and c^m for all $m = 1, \ldots, M$.

In the proof of Proposition 2 we have already shown that the squared norm of the difference vector $\tilde{e} = v - c\, U_\lambda(\phi_\gamma)$ is a DC function of c and γ, and we have described a way to compute its DC decomposition when a DC decomposition of the components of $U_\lambda(\phi_\gamma)$ is available. Therefore, one can obtain the DC decompositions of the terms $\|v_i^m - c^m\, U_{\lambda_i^m}(\phi_{\gamma^m})\|^2$ and $\|v_i^{k,m} - c^m\, U_{\lambda_i^{k,m}}(\phi_{\gamma^m})\|^2$ in $\tilde{E}^m(\gamma^m, c^m)$. Then, $\tilde{E}^m(\gamma^m, c^m)$ is a DC function of γ^m and c^m as it is a linear combination of these terms, and its decomposition is simply given by Proposition 6.a.

□

Appendix B

Appendix

B.1 Proof of Theorem 1

Proof: Now we derive the upper bound on the alignment error given in Theorem 1. First, notice from (5.5) that the difference between the optimal and estimated transformation parameters is given by

$$\lambda_e - \lambda_o = [\mathcal{G}_{ij}(\lambda_r)]^{-1}[\langle q - p_{\lambda_r}, \partial_i\, p_{\lambda_r}\rangle] - (\lambda_o - \lambda_r).$$

Using the second-order approximation of the manifold given in (5.10), one can write the target pattern as

$$q = p_{\lambda_o} + n = p_{\lambda_r} + l_{\lambda_o} + \kappa_{\lambda_o} + n$$

where

$$l_{\lambda_o} = \sum_{i=1}^{d} \partial_i\, p_{\lambda_r}(\lambda_o^i - \lambda_r^i) \tag{B.1}$$

is the linear term and

$$\kappa_{\lambda_o} = \frac{1}{2}\sum_{i=1}^{d}\sum_{j=1}^{d} \partial_{ij}\, p_{\lambda_r}(\lambda_o^i - \lambda_r^i)(\lambda_o^j - \lambda_r^j)$$

is the quadratic term in the expansion of p_{λ_o} around p_{λ_r}. In particular, the linear component $l_{\lambda_o} \in T_{\lambda_r}\mathcal{M}(p)$ belongs to the tangent space of the manifold at p_{λ_r}, and κ_{λ_o} is the component of p_{λ_o} that represents the deviation of p_{λ_o} from the linear approximation $\mathcal{S}_{\lambda_r}(p)$ as a result of curvature. The alignment error is thus obtained as

$$\begin{aligned}\lambda_e - \lambda_o &= [\mathcal{G}_{ij}(\lambda_r)]^{-1}[\langle l_{\lambda_o} + \kappa_{\lambda_o} + n,\ \partial_i\, p_{\lambda_r}\rangle] - (\lambda_o - \lambda_r)\\ &= [\mathcal{G}_{ij}(\lambda_r)]^{-1}[\langle l_{\lambda_o},\ \partial_i\, p_{\lambda_r}\rangle] - (\lambda_o - \lambda_r)\\ &+ [\mathcal{G}_{ij}(\lambda_r)]^{-1}[\langle \kappa_{\lambda_o} + n,\ \partial_i\, p_{\lambda_r}\rangle].\end{aligned} \tag{B.2}$$

One can show from (5.2) that the orthogonal projection of a vector $v \in L^2(\mathbb{R}^2)$ onto the subspace $T_\lambda\mathcal{M}(p)$ is represented in the basis $\{\partial_i\, p_\lambda\}_{i=1}^d$ with the coefficient vector

$$\zeta = [\mathcal{G}_{ij}(\lambda)]^{-1}[\langle v,\ \partial_i\, p_\lambda\rangle].$$

Therefore, in (B.2), the term $[\mathcal{G}_{ij}(\lambda_r)]^{-1}[\langle l_{\lambda_o},\ \partial_i\, p_{\lambda_r}\rangle]$ corresponds to the coordinates of $l_{\lambda_o} \in T_{\lambda_r}\mathcal{M}(p)$ in the basis $\{\partial_i\, p_{\lambda_r}\}_{i=1}^d$. However, one can observe from (B.1) that this coordinate vector is given by $\lambda_o - \lambda_r$. Hence, using the equality

$$[\mathcal{G}_{ij}(\lambda_r)]^{-1}[\langle l_{\lambda_o},\ \partial_i\, p_{\lambda_r}\rangle] = \lambda_o - \lambda_r$$

in (B.2), we obtain the alignment error as

$$\lambda_e - \lambda_o = [\mathcal{G}_{ij}(\lambda_r)]^{-1}[\langle \kappa_{\lambda_o},\ \partial_i\, p_{\lambda_r}\rangle] \ + \ [\mathcal{G}_{ij}(\lambda_r)]^{-1}[\langle n,\ \partial_i\, p_{\lambda_r}\rangle].$$

The above equality shows that the alignment error is given by the sums of the projections of the quadratic component κ_{λ_o} and the noise component n onto the tangent space $T_{\lambda_r}\mathcal{M}(p)$. The norm of the alignment error can thus be upper bounded as

$$\|\lambda_e - \lambda_o\| \leq \left\|[\mathcal{G}_{ij}(\lambda_r)]^{-1}[\langle \kappa_{\lambda_o},\ \partial_i\, p_{\lambda_r}\rangle]\right\| + \left\|[\mathcal{G}_{ij}(\lambda_r)]^{-1}[\langle n,\ \partial_i\, p_{\lambda_r}\rangle]\right\|. \tag{B.3}$$

In the rest of our derivation, we proceed by finding an upper bound for the two terms in the above expression. We begin with the norm of $[\mathcal{G}_{ij}(\lambda_r)]^{-1}[\langle \kappa_{\lambda_o},\ \partial_i\, p_{\lambda_r}\rangle]$. We have

$$\begin{aligned}\left\|[\mathcal{G}_{ij}(\lambda_r)]^{-1}[\langle \kappa_{\lambda_o},\ \partial_i\, p_{\lambda_r}\rangle]\right\| &\leq \lambda_{\max}\left([\mathcal{G}_{ij}(\lambda_r)]^{-1}\right)\ \left\|[\langle \kappa_{\lambda_o},\ \partial_i\, p_{\lambda_r}\rangle]\right\| \\ &= \lambda_{\min}^{-1}\left([\mathcal{G}_{ij}(\lambda_r)]\right)\ \left\|[\langle \kappa_{\lambda_o},\ \partial_i\, p_{\lambda_r}\rangle]\right\|\end{aligned}$$

where $\lambda_{\max}(\cdot)$ and $\lambda_{\min}(\cdot)$ denote respectively the maximum and minimum eigenvalues of a matrix. We have

$$\begin{aligned}\left\|[\langle \kappa_{\lambda_o},\ \partial_i\, p_{\lambda_r}\rangle]\right\| &= \left(\sum_{i=1}^d |\langle \kappa_{\lambda_o},\ \partial_i\, p_{\lambda_r}\rangle|^2\right)^{1/2} \leq \left(\sum_{i=1}^d \|\kappa_{\lambda_o}\|^2 \|\partial_i\, p_{\lambda_r}\|^2\right)^{1/2} \\ &= \|\kappa_{\lambda_o}\| \sqrt{\operatorname{tr}\big([\mathcal{G}_{ij}(\lambda_r)]\big)}\end{aligned}$$

which gives

$$\left\|[\mathcal{G}_{ij}(\lambda_r)]^{-1}[\langle \kappa_{\lambda_o},\ \partial_i\, p_{\lambda_r}\rangle]\right\| \leq \lambda_{\min}^{-1}\left([\mathcal{G}_{ij}(\lambda_r)]\right) \sqrt{\operatorname{tr}\big([\mathcal{G}_{ij}(\lambda_r)]\big)}\, \|\kappa_{\lambda_o}\|. \tag{B.4}$$

One can upper bound the norm of the quadratic term κ_{λ_o} as

$$\begin{aligned}\|\kappa_{\lambda_o}\| &= \frac{1}{2}\left\|\sum_{i=1}^d \sum_{j=1}^d \partial_{ij}\, p_{\lambda_r}(\lambda_o^i - \lambda_r^i)(\lambda_o^j - \lambda_r^j)\right\| \leq \frac{1}{2}\sum_{i=1}^d \sum_{j=1}^d \|\partial_{ij}\, p_{\lambda_r}\| \|\lambda_o - \lambda_r\|_\infty^2 \\ &\leq \frac{1}{2}\, d^2\, \mathcal{K}\ \|\lambda_o - \lambda_r\|_\infty^2.\end{aligned}$$

Using this in (B.4) we bound the norm of the projection of κ_{λ_o} on $T_{\lambda_r}\mathcal{M}(p)$ as follows

$$\left\|[\mathcal{G}_{ij}(\lambda_r)]^{-1}[\langle \kappa_{\lambda_o},\ \partial_i\, p_{\lambda_r}\rangle]\right\| \leq \frac{1}{2}\, d^2\ \mathcal{K}\ \lambda_{\min}^{-1}\ \left([\mathcal{G}_{ij}(\lambda_r)]\right) \sqrt{\operatorname{tr}\big([\mathcal{G}_{ij}(\lambda_r)]\big)}\ \|\lambda_o - \lambda_r\|_\infty^2. \tag{B.5}$$

Having thus obtained an upper bound for the first additive term in (B.3), we now continue with the second term $\left\|[\mathcal{G}_{ij}(\lambda_r)]^{-1}[\langle n,\ \partial_i\, p_{\lambda_r}\rangle]\right\|$. First, remember from (5.9) that the noise component n is orthogonal to the tangent space $T_{\lambda_o}\mathcal{M}(p)$ at p_{λ_o}. The term $[\mathcal{G}_{ij}(\lambda_r)]^{-1}[\langle n,\ \partial_i\, p_{\lambda_r}\rangle]$ represents the coordinates of the projection of n onto the tangent space $T_{\lambda_r}\mathcal{M}(p)$ at p_{λ_r}. Due to manifold curvature, there is a nonzero angle between these two tangent spaces; therefore, the orthogonal projection of n onto $T_{\lambda_r}\mathcal{M}(p)$ is a nonzero vector in general. In the following, we derive an upper bound for the magnitude of this projection by looking at the change in the tangent vectors between the two manifold points p_{λ_r} and p_{λ_o}. Let us define

$$\Delta_i := \partial_i\, p_{\lambda_r} - \partial_i\, p_{\lambda_o}$$

which gives the change in the i-th tangent vector between the points p_{λ_r} and p_{λ_o}. We have

$$\begin{aligned}[\mathcal{G}_{ij}(\lambda_r)]^{-1}[\langle n,\ \partial_i\, p_{\lambda_r}\rangle] &= [\mathcal{G}_{ij}(\lambda_r)]^{-1}[\langle n,\ \partial_i\, p_{\lambda_o}\rangle] + [\mathcal{G}_{ij}(\lambda_r)]^{-1}[\langle n,\ \Delta_i\rangle] \\ &= [\mathcal{G}_{ij}(\lambda_r)]^{-1}[\langle n,\ \Delta_i\rangle]\end{aligned} \tag{B.6}$$

since $\langle n,\ \partial_i\, p_{\lambda_o}\rangle = 0$ for all $i = 1, \cdots, d$. We now derive an upper bound for the norm of Δ_i as follows. Let us define a curve

$$p_{\lambda(t)} : [0, 1] \to \mathcal{M}(p)$$

such that

$$\lambda(t) = \lambda_o + t(\lambda_r - \lambda_o).$$

Hence, $p_{\lambda(0)} = p_{\lambda_o}$ and $p_{\lambda(1)} = p_{\lambda_r}$. For each $i = 1, \cdots, d$ we have

$$\begin{aligned}\partial_i\, p_{\lambda_r} &= \partial_i\, p_{\lambda_o} + \int_0^1 \frac{d\,\partial_i p_{\lambda(t)}}{dt}\ dt = \partial_i\, p_{\lambda_o} + \int_0^1 \sum_{j=1}^{d} \partial_{ij} p_{\lambda(t)} \frac{d\lambda^j(t)}{dt}\ dt \\ &= \partial_i\, p_{\lambda_o} + \int_0^1 \sum_{j=1}^{d} \partial_{ij} p_{\lambda(t)} (\lambda_r - \lambda_o)^j\ dt.\end{aligned}$$

We thus get the following upper bound on $\|\Delta_i\|$

$$\begin{aligned}\|\Delta_i\| &= \left\| \int_0^1 \sum_{j=1}^{d} \partial_{ij} p_{\lambda(t)} (\lambda_r - \lambda_o)^j\ dt \right\| = \left\| \sum_{j=1}^{d} \int_0^1 \partial_{ij} p_{\lambda(t)} (\lambda_r - \lambda_o)^j\ dt \right\| \\ &\leq \sum_{j=1}^{d} \int_0^1 \left\|\partial_{ij} p_{\lambda(t)}\right\| |(\lambda_r - \lambda_o)^j|\ dt \quad \leq \quad \sum_{j=1}^{d} \mathcal{K}\ |(\lambda_r - \lambda_o)^j| \\ &= \mathcal{K}\ \|\lambda_r - \lambda_o\|_1.\end{aligned}$$

It follows from (B.6) that

$$\left\|[\mathcal{G}_{ij}(\lambda_r)]^{-1}[\langle n,\ \partial_i\, p_{\lambda_r}\rangle]\right\| = \left\|[\mathcal{G}_{ij}(\lambda_r)]^{-1}[\langle n,\ \Delta_i\rangle]\right\| \leq\ \lambda_{\min}^{-1}\left([\mathcal{G}_{ij}(\lambda_r)]\right) \left\|[\langle n,\ \Delta_i\rangle]\right\|$$

where

$$\left\|[\langle n,\ \Delta_i\rangle]\right\| = \left(\sum_{i=1}^{d} |\langle n,\ \Delta_i\rangle|^2\right)^{1/2} \leq \left(\sum_{i=1}^{d} \nu^2 \|\Delta_i\|^2\right)^{1/2}.$$

Using the bound $\|\Delta_i\| \leq \mathcal{K}\ \|\lambda_r - \lambda_o\|_1$ above we get

$$\left\|[\langle n,\ \Delta_i\rangle]\right\| \leq \mathcal{K}\sqrt{d}\ \nu\ \|\lambda_r - \lambda_o\|_1$$

which gives

$$\left\|[\mathcal{G}_{ij}(\lambda_r)]^{-1}[\langle n,\ \partial_i\, p_{\lambda_r}\rangle]\right\| \leq \mathcal{K}\sqrt{d}\ \nu\ \lambda_{\min}^{-1}\left([\mathcal{G}_{ij}(\lambda_r)]\right)\ \|\lambda_r - \lambda_o\|_1. \tag{B.7}$$

This finishes the derivation of the upper bound for the norm of the projection of the noise component on $T_{\lambda_r}\mathcal{M}(p)$. We finally put together the results (B.5) and (B.7) in (B.3) and get the stated bound on the norm of the alignment error $\|\lambda_e - \lambda_o\|$

$$\|\lambda_e - \lambda_o\| \leq\ \mathcal{K}\ \lambda_{\min}^{-1}\ \left([\mathcal{G}_{ij}(\lambda_r)]\right)\left(\frac{1}{2}\, d^2\sqrt{\mathrm{tr}([\mathcal{G}_{ij}(\lambda_r)])}\ \|\lambda_o - \lambda_r\|_\infty^2 + \sqrt{d}\ \nu\ \|\lambda_o - \lambda_r\|_1\right)$$

which concludes the proof.

□

B.2 Derivations of $\|\partial_i\, \hat{p}_\lambda\|$ and $\|\partial_{ij}\, \hat{p}_\lambda\|$ in terms of pattern spatial derivatives

As the pattern $\hat{p}$ and its derivatives are square-integrable, there exists a bounded support $\Omega \in \mathbb{R}^2$ such that the intensities of $\hat{p}$ and its derivatives are significantly reduced outside Ω; i.e., [1]

$$\hat{p}(X),\ \partial_x\, \hat{p}(X),\ \partial_y\, \hat{p}(X),\ \partial_{xx}\, \hat{p}(X),\ \partial_{xy}\, \hat{p}(X),\ \partial_{yy}\, \hat{p}(X) \approx 0$$

for $X \notin \Omega$. Since the coordinate change function a is C^2-smooth, the derivatives of the transformed coordinates are bounded over Ω. Hence, there exists a constant $M > 0$ such that

$$|\partial_i\, x'|,\ |\partial_i\, y'|, |\partial_{ij}\, x'|,\ |\partial_{ij}\, y'| \leq M$$

for all $i, j = 1, \cdots, d$ and $X' \in \Omega$.

Let us first clarify the notation used in the rest of our derivations. For a vector-valued function $g : \mathbb{R}^2 \to \mathbb{R}^n$, the notation g denotes the function considered as an element of the function space it belongs to, while the notation $g(X)$ always stands for the value of g evaluated at X; i.e., a vector in $\mathbb{R}^n$.

We begin with the term $\|\partial_i\, \hat{p}_\lambda\|$. For all X, we have

$$|\partial_i\, \hat{p}_\lambda(X)| = |\nabla\hat{p}(X')^T \partial_i\, X'| \leq \|\nabla\hat{p}(X')\|\|\partial_i\, X'\|$$

[1] As filtering leads to a spatial diffusion in the intensity functions of the pattern and its derivatives, the size of the support Ω in fact depends on the filter size ρ. However, for the sake of simplicity of analysis, we ignore the dependence of Ω on ρ and assume a single and sufficiently large support region Ω, which can be selected with respect to the largest value of the filter size used in a hierarchical registration application.

where $\partial_i X' = [\partial_i x' \ \partial_i y']^T$. Then, for $X \in a_\lambda^{-1}(\Omega)$, $|\partial_i \hat{p}_\lambda(X)|$ can be upper bounded as

$$|\partial_i \hat{p}_\lambda(X)| \leq \sqrt{2}M \ \|\nabla \hat{p}(X')\|.$$

We thus get

$$\begin{aligned}
\|\partial_i \hat{p}_\lambda\|^2 &= \int_{\mathbb{R}^2} |\partial_i \hat{p}_\lambda(X)|^2 dX = \int_{\mathbb{R}^2} |\nabla \hat{p}(X')^T \partial_i X'|^2 dX \\
&\approx \int_{a_\lambda^{-1}(\Omega)} |\nabla \hat{p}(X')^T \partial_i X'|^2 dX \leq 2M^2 \int_{a_\lambda^{-1}(\Omega)} \|\nabla \hat{p}(X')\|^2 dX \\
&= 2M^2 \int_\Omega \|\nabla \hat{p}(X)\|^2 \ |\det(Da_\lambda^{-1})(X)| \ dX
\end{aligned}$$

where $\det(Da_\lambda^{-1})(X)$ is the Jacobian of the coordinate change function a_λ^{-1}. In the above equations, when approximating the integration on $\mathbb{R}^2$ with the integration on $a_\lambda^{-1}(\Omega)$, we implicitly assume that $\nabla \hat{p}(X')^T \partial_i X' \approx 0$ outside the inverse image of the support region Ω. Such an assumption is reasonable as the transformed coordinates X' are typically polynomial functions of the original coordinates X and their rate of increase with X is therefore dominated by the decay of the image intensity function with X in a typical representation in $L^2(\mathbb{R}^2)$ such as the Gaussian dictionary we use in this work, which is introduced in Section 5.3.2. Since the function a_λ is a smooth bijection on $\mathbb{R}^2$, the Jacobian $\det(Da_\lambda^{-1})(X)$ is bounded on the bounded region Ω. Therefore, there exists a constant $C > 0$ such that $|\det(Da_\lambda^{-1})(X)| \leq C$ for $X \in \Omega$. Hence, we obtain

$$\|\partial_i \hat{p}_\lambda\| \leq \sqrt{2M^2C} \left(\int_{\mathbb{R}^2} \|\nabla \hat{p}(X)\|^2 dX \right)^{1/2} = \sqrt{2M^2C} \ \|N_\nabla \hat{p}\|$$

which shows that $\|\partial_i \hat{p}_\lambda\|$ and $\|N_\nabla \hat{p}\|$ have approximately the same rate of change with the filter size ρ; i.e.,

$$\|\partial_i \hat{p}_\lambda\| = O(\|N_\nabla \hat{p}\|).$$

Next, we look at the term $\|\partial_{ij} \hat{p}_\lambda\|$. From triangle inequality we have

$$\|\partial_{ij} \hat{p}_\lambda\| \leq \|v\| + \|w\|$$

where

$$\begin{aligned}
v(X) &= \partial_{xx} \hat{p}(X') \, \partial_i x' \partial_j x' + \partial_{xy} \hat{p}(X') \, (\partial_i x' \partial_j y' + \partial_j x' \partial_i y') + \partial_{yy} \hat{p}(X') \, \partial_i y' \partial_j y' \\
w(X) &= \partial_x \hat{p}(X') \, \partial_{ij} x' + \partial_y \hat{p}(X') \, \partial_{ij} y'.
\end{aligned}$$

Since w is in the same form as $\partial_i \hat{p}_\lambda$, one can upper bound it in the same way.

$$\|w\| \leq \sqrt{2M^2C} \ \|N_\nabla \hat{p}\|. \tag{B.8}$$

We now examine the term $\|v\|$. Defining the derivative product vector

$$B(X') = [\partial_i x' \partial_j x' \quad \partial_i x' \partial_j y' \quad \partial_j x' \partial_i y' \quad \partial_i y' \partial_j y']^T,$$

we have

$$|v(X)| = |(h\hat{p})(X')^T\ B(X')| \leq \|(h\hat{p})(X')\|\ \|B(X')\|.$$

At $X \in a_\lambda^{-1}(\Omega)$, the upper bound $\|B(X')\| \leq 2M^2$ yields

$$|v(X)| \leq 2M^2\ \|(h\hat{p})(X')\|.$$

Hence,

$$\begin{aligned}
\|v\|^2 &= \int_{\mathbb{R}^2} |v(X)|^2 dX = \int_{\mathbb{R}^2} |(h\hat{p})(X')^T\ B(X')|^2 dX \\
&\approx \int_{a_\lambda^{-1}(\Omega)} |(h\hat{p})(X')^T\ B(X')|^2 dX \leq 4M^4 \int_{a_\lambda^{-1}(\Omega)} \|(h\hat{p})(X')\|^2 dX \\
&= 4M^4 \int_\Omega \|(h\hat{p})(X)\|^2\ |\det(Da_\lambda^{-1})(X)|\ dX \quad \leq \quad 4M^4 C \int_\Omega \|(h\hat{p})(X)\|^2 dX
\end{aligned}$$

and therefore

$$\|v\| \leq 2M^2\sqrt{C}\left(\int_{\mathbb{R}^2} \|(h\hat{p})(X)\|^2 dX\right)^{1/2} = 2M^2\sqrt{C}\ \|N_h\hat{p}\|. \tag{B.9}$$

Finally, putting together (B.8) and (B.9), we obtain the following upper bound on $\|\partial_{ij}\,\hat{p}_\lambda\|$

$$\|\partial_{ij}\,\hat{p}_\lambda\| \ \leq\ 2M^2\sqrt{C}\ \|N_h\hat{p}\| + \sqrt{2M^2C}\,\|N_\nabla\hat{p}\|$$

which gives

$$\|\partial_{ij}\,\hat{p}_\lambda\| = O\big(\|N_\nabla\hat{p}\| + \|N_h\hat{p}\|\big).$$

B.3 Proof of Lemma 1

Since the reference pattern consists of Gaussian atoms, the derivation of the norms of its gradient and Hessian involves the integration of products of Gaussian atom pairs. Therefore, in our analysis we make use of the following proposition, which gives the expression for the integration of the product of two Gaussian atoms [99].

Proposition 8. *Let* $\phi_{\gamma_j}(X) = \phi(\sigma_j^{-1}\,\Psi_j^{-1}\,(X-\tau_j))$ *and* $\phi_{\gamma_k}(X) = \phi(\sigma_k^{-1}\,\Psi_k^{-1}\,(X-\tau_k))$. *Then*

$$\int_{\mathbb{R}^2} \phi_{\gamma_j}(X)\phi_{\gamma_k}(X)dX = \frac{Q_{jk}}{2}$$

where

$$\begin{aligned}
Q_{jk} &:= \frac{\pi\,|\sigma_j\sigma_k|}{\sqrt{|\Sigma_{jk}|}} \exp\left(-\frac{1}{2}(\tau_k-\tau_j)^T\,\Sigma_{jk}^{-1}\,(\tau_k-\tau_j)\right) \\
\Sigma_{jk} &:= \frac{1}{2}\left(\Psi_j\,\sigma_j^2\,\Psi_j^{-1} + \Psi_k\,\sigma_k^2\,\Psi_k^{-1}\right).
\end{aligned} \tag{B.10}$$

We now prove Lemma 1.

Proof: In order to determine the variations of $\|N_\nabla \hat{p}\|$ and $\|N_h \hat{p}\|$ with the filter size ρ, we first derive approximations for these terms in terms of the atom parameters of the reference pattern, which makes it easier to analyze them analytically. We then examine the dependence of these terms on ρ with the help of their approximations.

Derivation of $\|N_\nabla \hat{p}\|$

We begin with the norm $\|N_\nabla \hat{p}\|$ of the gradient magnitude. In order to lighten the notation, we do the derivations for the unfiltered reference pattern p, which are directly generalizable for its filtered versions. We have

$$\|N_\nabla p\|^2 = \int_{\mathbb{R}^2} \|\nabla p(X)\|^2 dX = \int_{\mathbb{R}^2} \left(\sum_{j=1}^{\infty} c_j (\nabla \phi_{\gamma_j}(X))^T \right) \left(\sum_{k=1}^{\infty} c_k \nabla \phi_{\gamma_k}(X) \right) dX.$$

It is easy to show that the gradient $\nabla \phi_{\gamma_j}(X)$ of the atom $\phi_{\gamma_j}(X)$ is given by

$$\nabla \phi_{\gamma_j}(X) = -2\, \phi_{\gamma_j}(X)\, \Psi_j\, \sigma_j^{-2} \Psi_j^{-1} (X - \tau_j)$$

which yields

$$(\nabla \phi_{\gamma_j}(X))^T \nabla \phi_{\gamma_k}(X) = 4\, \phi_{\gamma_j}(X) \phi_{\gamma_k}(X)\, (X - \tau_j)^T \Theta_j^T \Theta_k (X - \tau_k)$$

where $\Theta_j := \Psi_j\, \sigma_j^{-2} \Psi_j^{-1}$. Putting this in the expression of $\|N_\nabla p\|^2$, we obtain

$$\|N_\nabla p\|^2 = 4 \sum_{j=1}^{\infty} \sum_{k=1}^{\infty} c_j c_k\, L_{jk} \tag{B.11}$$

where

$$L_{jk} = \int_{\mathbb{R}^2} \phi_{\gamma_j}(X) \phi_{\gamma_k}(X)\, (X - \tau_j)^T \Theta_j^T \Theta_k (X - \tau_k)\, dX. \tag{B.12}$$

The evaluation of the above integral would give the exact expression of L_{jk} in terms of the atom parameters of p, which would however have a quite complicated form. On the other hand, we are interested in determining the variation of L_{jk} with filtering rather than obtaining its exact expression. Hence, in order to make the derivation simpler, we approximate the above expression for L_{jk} with another term $\overline{L}_{jk}$, which is easier to evaluate analytically and provides an upper bound for L_{jk} at the same time. Let us denote the smaller and greater eigenvalues of Θ_j as

$$\iota_j = \lambda_{\min}(\Theta_j), \qquad \vartheta_j = \lambda_{\max}(\Theta_j).$$

From Cauchy-Schwarz inequality,

$$|(X - \tau_j)^T \Theta_j^T \Theta_k (X - \tau_k)| \leq \|\Theta_j (X - \tau_j)\|\, \|\Theta_k (X - \tau_k)\| \leq \vartheta_j \|X - \tau_j\|\, \vartheta_k \|X - \tau_k\|.$$

Using this in the expression of L_{jk}, we get

$$\begin{aligned} L_{jk} \leq |L_{jk}| &\leq \int_{\mathbb{R}^2} \phi_{\gamma_j}(X)\phi_{\gamma_k}(X)\, \vartheta_j\vartheta_k\, \|X-\tau_j\|\|X-\tau_k\|\, dX \\ &\leq \overline{L}_{jk} := \vartheta_j\vartheta_k\sqrt{\overline{L}_j}\sqrt{\overline{L}_k} \end{aligned}$$

where

$$\overline{L}_j = \int_{\mathbb{R}^2} \phi^2_{\gamma_j}(X)\, \|X-\tau_j\|^2 dX.$$

Evaluating the above integral, we obtain

$$\overline{L}_j = \frac{\pi}{8}\, |\sigma_j|\, (\sigma^2_{x,j} + \sigma^2_{y,j}).$$

This gives the following upper bound for L_{jk}

$$\overline{L}_{jk} = \frac{\pi}{8}\vartheta_j\vartheta_k \left(|\sigma_j\sigma_k|\, (\sigma^2_{x,j} + \sigma^2_{y,j})(\sigma^2_{x,k} + \sigma^2_{y,k})\right)^{1/2}. \tag{B.13}$$

Now, generalizing (B.11) to filtered versions of the reference pattern, we have

$$\|N_\nabla \hat{p}\|^2 = 4\sum_{j=1}^{\infty}\sum_{k=1}^{\infty} \hat{c}_j\hat{c}_k\, \hat{L}_{jk}. \tag{B.14}$$

We now determine the dependence of $\|N_\nabla \hat{p}\|$ on the filter size ρ. First, from (5.22), the coefficient products have the variation

$$\hat{c}_j\hat{c}_k = O((1+\rho^2)^{-2}) \tag{B.15}$$

with the filter size. Next, we look at the term $\hat{L}_{jk}$. Note that the low-pass filter applied on the pattern p increases the atom scale parameters $\sigma_{x,j}$, $\sigma_{y,j}$ and therefore decreases the eigenvalues of the matrices Θ_j, Θ_k in the exact expression for L_{jk} in (B.12). Filtering also influences the terms $\phi_{\gamma_j}(X)$ and $\phi_{\gamma_k}(X)$ in (B.12). The variations of these terms with ρ are captured in the approximation $\overline{L}_{jk}$ through the terms ϑ_j, ϑ_k, $\overline{L}_j$, and $\overline{L}_k$. Therefore, L_{jk} and $\overline{L}_{jk}$ have the same rate of change with the filter size ρ. From (B.13), the approximation $\overline{\hat{L}}_{jk}$ of $\hat{L}_{jk}$ is given by

$$\overline{\hat{L}}_{jk} = \frac{\pi}{8}\hat{\vartheta}_j\hat{\vartheta}_k \left(|\hat{\sigma}_j\hat{\sigma}_k|\, (\hat{\sigma}^2_{x,j} + \hat{\sigma}^2_{y,j})(\hat{\sigma}^2_{x,k} + \hat{\sigma}^2_{y,k})\right)^{1/2} \tag{B.16}$$

which is simply obtained by replacing the parameters σ_j and ϑ_j with their filtered versions $\hat{\sigma}_j$ and $\hat{\vartheta}_j$. From (5.24), we have

$$\begin{aligned} \hat{\sigma}_{x,j},\, \hat{\sigma}_{y,j} &= O((1+\rho^2)^{1/2}) \\ |\hat{\sigma}_j\hat{\sigma}_k| &= O((1+\rho^2)^2) \\ \hat{\vartheta}_j &= \max(\hat{\sigma}^{-2}_{x,j}\,,\, \hat{\sigma}^{-2}_{y,j}) = O((1+\rho^2)^{-1}). \end{aligned} \tag{B.17}$$

Putting these relations together in (B.16), we obtain

$$\overline{\hat{L}}_{jk} = O(1)$$

with respect to ρ. Combining this with the rate of change of the coefficient product $\hat{c}_j\hat{c}_k$ in (B.15) yields $\hat{c}_j\hat{c}_k\hat{L}_{jk} = O((1+\rho^2)^{-2})$. Since each one the additive terms in the expression of $\|N_\nabla \hat{p}\|^2$ in (B.14) has the same rate of decrease with ρ, the infinite sum also decreases with ρ at the same rate. Therefore, we get $\|N_\nabla \hat{p}\|^2 = O((1+\rho^2)^{-2})$, which gives

$$\|N_\nabla \hat{p}\| = O((1+\rho^2)^{-1}).$$

Derivation of $\|N_h\hat{p}\|$

We now continue with the norm $\|N_h\hat{p}\|$ of the Hessian magnitude. From (5.14),

$$(N_h p\,(X))^2 = \|(hp)(X)\|^2 = (\partial_{xx}\,p(X))^2 + 2(\partial_{xy}\,p(X))^2 + (\partial_{yy}\,p(X))^2.$$

Hence,

$$\|N_h p\|^2 = \int_{\mathbb{R}^2} (N_h p\,(X))^2\,dX = \int_{\mathbb{R}^2} (\partial_{xx}\,p(X))^2 + 2(\partial_{xy}\,p(X))^2 + (\partial_{yy}\,p(X))^2 dX.$$

The second derivatives of the pattern are of the form

$$\partial_{xx}\,p(X) = \sum_{k=1}^{\infty} c_k \frac{\partial^2 \phi_{\gamma_k}(X)}{\partial x^2}$$

and $\partial_{xy}\,p(X)$, $\partial_{yy}\,p(X)$ are obtained similarly. Then, $\|N_h p\|^2$ is given by

$$\begin{aligned}\|N_h p\|^2 &= \sum_{j=1}^{\infty}\sum_{k=1}^{\infty} c_j c_k \int_{\mathbb{R}^2} \left(\frac{\partial^2 \phi_{\gamma_j}(X)}{\partial x^2}\frac{\partial^2 \phi_{\gamma_k}(X)}{\partial x^2} + 2\frac{\partial^2 \phi_{\gamma_j}(X)}{\partial x \partial y}\frac{\partial^2 \phi_{\gamma_k}(X)}{\partial x \partial y} + \frac{\partial^2 \phi_{\gamma_j}(X)}{\partial y^2}\frac{\partial^2 \phi_{\gamma_k}(X)}{\partial y^2} \right) dX \\ &= \sum_{j=1}^{\infty}\sum_{k=1}^{\infty} c_j c_k \int_{\mathbb{R}^2} \operatorname{tr}\big(H(\phi_{\gamma_j}(X))H(\phi_{\gamma_k}(X))\big)\,dX\end{aligned}$$

where

$$H(\phi_{\gamma_j}(X)) = \begin{bmatrix} \frac{\partial^2 \phi_{\gamma_j}(X)}{\partial x^2} & \frac{\partial^2 \phi_{\gamma_j}(X)}{\partial x \partial y} \\ \frac{\partial^2 \phi_{\gamma_j}(X)}{\partial x \partial y} & \frac{\partial^2 \phi_{\gamma_j}(X)}{\partial y^2} \end{bmatrix}$$

denotes the Hessian matrix of $\phi_{\gamma_j}(X)$. It is easy to show that

$$\begin{aligned}H(\phi_{\gamma_j}(X)) &= -2\Theta_j(X-\tau_j)\,\nabla^T \phi_{\gamma_j}(X) - 2\phi_{\gamma_j}(X)\Theta_j \\ &= \phi_{\gamma_j}(X)\big(4\Theta_j(X-\tau_j)(X-\tau_j)^T\Theta_j - 2\Theta_j\big)\end{aligned}$$

which yields

$$\begin{aligned}\operatorname{tr}\big(H(\phi_{\gamma_j}(X))H(\phi_{\gamma_k}(X))\big) = \phi_{\gamma_j}(X)\phi_{\gamma_k}(X)\Big[&16\operatorname{tr}\big(\Theta_j(X-\tau_j)(X-\tau_j)^T\Theta_j\Theta_k(X-\tau_k)(X-\tau_k)^T\Theta_k\big)\\ &-8\operatorname{tr}\big(\Theta_j(X-\tau_j)(X-\tau_j)^T\Theta_j\Theta_k\big)-8\operatorname{tr}\big(\Theta_j\Theta_k(X-\tau_k)(X-\tau_k)^T\Theta_k\big)\\ &+4\operatorname{tr}\big(\Theta_j\Theta_k\big)\Big].\end{aligned}$$

The squared norm of the Hessian magnitude can then be written as

$$\|N_h p\|^2 = \sum_{j=1}^{\infty}\sum_{k=1}^{\infty} c_j c_k \left(16M_{jk} - 8N_{jk} - 8N_{kj} + 4P_{jk}\right) \tag{B.18}$$

where

$$\begin{aligned}M_{jk} &= \int_{\mathbb{R}^2} \phi_{\gamma_j}(X)\phi_{\gamma_k}(X)\operatorname{tr}\big(\Theta_j(X-\tau_j)(X-\tau_j)^T\Theta_j\Theta_k(X-\tau_k)(X-\tau_k)^T\Theta_k\big)\,dX\\ N_{jk} &= \int_{\mathbb{R}^2} \phi_{\gamma_j}(X)\phi_{\gamma_k}(X)\operatorname{tr}\big(\Theta_j(X-\tau_j)(X-\tau_j)^T\Theta_j\Theta_k\big)\,dX\\ P_{jk} &= \int_{\mathbb{R}^2} \phi_{\gamma_j}(X)\phi_{\gamma_k}(X)\operatorname{tr}\big(\Theta_j\Theta_k\big)\,dX.\end{aligned} \tag{B.19}$$

We now derive approximations $\overline{M}_{jk}$, $\overline{N}_{jk}$, $\overline{P}_{jk}$ for the terms written above, which are easier to treat analytically and constitute upper bounds for these terms as well.

We begin with M_{jk}. Denoting $A_j = \Theta_j(X-\tau_j)(X-\tau_j)^T\Theta_j$,

$$M_{jk} \le |M_{jk}| \le \int_{\mathbb{R}^2} \phi_{\gamma_j}(X)\phi_{\gamma_k}(X)\,|\operatorname{tr}(A_jA_k)|dX. \tag{B.20}$$

Since A_j is a rank-1 matrix,

$$|\operatorname{tr}(A_jA_k)| = |\lambda_{\max}(A_jA_k)| \le \|A_jA_k\| \le \|A_j\|\,\|A_k\|$$

where $\|\cdot\|$ denotes the operator norm for matrices. The first inequality above follows from the fact that the spectral radius of a matrix is smaller than its operator norm, and the second inequality comes from the submultiplicative property of the operator norm. From the inequality

$$\|A_j\| = \|\Theta_j(X-\tau_j)(X-\tau_j)^T\Theta_j\| \le \vartheta_j^2\,\|X-\tau_j\|^2$$

we get

$$|\operatorname{tr}(A_jA_k)| \le \vartheta_j^2\vartheta_k^2\,\|X-\tau_j\|^2\|X-\tau_k\|^2.$$

Using this bound in (B.20) yields

$$M_{jk} \le \vartheta_j^2\vartheta_k^2 \int_{\mathbb{R}^2} \phi_{\gamma_j}(X)\phi_{\gamma_k}(X)\|X-\tau_j\|^2\|X-\tau_k\|^2 dX$$

which gives the upper bound

$$M_{jk} \leq \overline{M}_{jk} := \vartheta_j^2 \vartheta_k^2 \sqrt{\overline{M}_j} \sqrt{\overline{M}_k}$$

where

$$\overline{M}_j = \int_{\mathbb{R}^2} \phi_{\gamma_j}^2(X) \|X - \tau_j\|^4 dX.$$

Evaluating the above integral, we get

$$\overline{M}_j = \pi |\sigma_j| \left(\frac{3}{32} \sigma_{x,j}^4 + \frac{1}{16} \sigma_{x,j}^2 \sigma_{y,j}^2 + \frac{3}{32} \sigma_{y,j}^4 \right).$$

This finishes the derivation of $\overline{M}_{jk}$.

Next, we look at the term N_{jk}. Performing similar steps as in M_{jk}, we obtain

$$|\mathrm{tr}(A_j \Theta_k)| \leq \vartheta_j^2 \vartheta_k \|X - \tau_j\|^2.$$

This gives $N_{jk} \leq \vartheta_j^2 \vartheta_k \sqrt{\overline{M}_j} \, \|\phi_{\gamma_k}\|$. The norm $\|\phi_{\gamma_k}\|$ of the atom ϕ_{γ_k} is

$$\|\phi_{\gamma_k}\| = \sqrt{\frac{\pi |\sigma_k|}{2}}.$$

Hence, the term N_{jk} is upper bounded as

$$N_{jk} \leq \overline{N}_{jk} := \sqrt{\frac{\pi |\sigma_k|}{2}} \, \vartheta_j^2 \vartheta_k \sqrt{\overline{M}_j}.$$

Lastly, we derive a bound for the term P_{jk}. The magnitude of the trace of $\Theta_j \Theta_k$ can be bounded as

$$|\mathrm{tr}(\Theta_j \Theta_k)| = |\lambda_{\min}(\Theta_j \Theta_k) + \lambda_{\max}(\Theta_j \Theta_k)| \leq 2\, r(\Theta_j \Theta_k) \leq 2\, \|\Theta_j \Theta_k\| \leq 2\, \|\Theta_j\| \, \|\Theta_k\| = 2 \vartheta_j \vartheta_k$$

where $r(\cdot)$ denotes the spectral radius of a matrix. The term P_{jk} can thus be bounded as

$$P_{jk} \leq 2 \vartheta_j \vartheta_k \int_{\mathbb{R}^2} \phi_{\gamma_j}(X) \phi_{\gamma_k}(X) dX.$$

From Proposition 8, we get

$$P_{jk} \leq \overline{P}_{jk} := \vartheta_j \vartheta_k Q_{jk}$$

where Q_{jk} is as defined in (B.10).

Having thus derived approximations $\overline{M}_{jk}$, $\overline{N}_{jk}$, $\overline{P}_{jk}$ for the terms M_{jk}, N_{jk}, P_{jk} in (B.18), we now have an analytical approximation of the norm $\|N_h p\|$ of the Hessian magnitude in terms of the atom parameters of the pattern. We now determine the order of variation of $\|N_h p\|$ with the filter size ρ using this approximation. From (B.18), we obtain the norm of the Hessian magnitude of the

filtered pattern $\hat{p}$ as

$$\|N_h\hat{p}\|^2 = \sum_{j=1}^{\infty}\sum_{k=1}^{\infty} \hat{c}_j\hat{c}_k\,(16\hat{M}_{jk} - 8\hat{N}_{jk} - 8\hat{N}_{kj} + 4\hat{P}_{jk}). \tag{B.21}$$

In the expressions of M_{jk}, N_{jk}, P_{jk} in (B.19), we see that filtering affects the terms Θ_j and the atoms $\phi_{\gamma_j}(X)$. Comparing these terms with their approximations $\overline{M}_{jk}$, $\overline{N}_{jk}$, $\overline{P}_{jk}$, we observe that the influence of smoothing on the matrices Θ_j is captured in the approximations via its influence on their eigenvalues ϑ_j, while the influence of smoothing on the atoms is also preserved in the approximations as the atoms appear in the expressions of $\overline{M}_{jk}$, $\overline{N}_{jk}$, $\overline{P}_{jk}$. Hence, the terms $\hat{M}_{jk}$, $\hat{N}_{jk}$, $\hat{P}_{jk}$ have the same rate of change with the filter size ρ as their approximations $\overline{\hat{M}}_{jk}$, $\overline{\hat{N}}_{jk}$, $\overline{\hat{P}}_{jk}$. In the following, we determine the order of dependence of these terms on ρ.

We begin with $\hat{M}_{jk}$. The relations in (B.17) imply that

$$\overline{\hat{M}}_j = \pi|\hat{\sigma}_j|\left(\frac{3}{32}\hat{\sigma}_{x,j}^4 + \frac{1}{16}\hat{\sigma}_{x,j}^2\hat{\sigma}_{y,j}^2 + \frac{3}{32}\hat{\sigma}_{y,j}^4\right)$$

increases with ρ at a rate of $O((1+\rho^2)^3)$ and the product $\hat{\vartheta}_j^2\hat{\vartheta}_k^2$ decreases with ρ at a rate of $O((1+\rho^2)^{-4})$. Therefore, the overall rate of variation of

$$\overline{\hat{M}}_{jk} = \hat{\vartheta}_j^2\hat{\vartheta}_k^2\sqrt{\overline{\hat{M}}_j}\sqrt{\overline{\hat{M}}_k}$$

with the filter size is given by

$$\overline{\hat{M}}_{jk} = O((1+\rho^2)^{-1}). \tag{B.22}$$

We similarly obtain the dependence of

$$\overline{\hat{N}}_{jk} = \sqrt{\frac{\pi|\hat{\sigma}_k|}{2}}\,\hat{\vartheta}_j^2\hat{\vartheta}_k\sqrt{\overline{\hat{M}}_j}$$

on the filter size as

$$\overline{\hat{N}}_{jk} = O((1+\rho^2)^{-1}). \tag{B.23}$$

Lastly,

$$\overline{\hat{P}}_{jk} = \hat{\vartheta}_j\hat{\vartheta}_k\hat{Q}_{jk}$$

where

$$\hat{Q}_{jk} = \frac{\pi\,|\hat{\sigma}_j\hat{\sigma}_k|}{\sqrt{|\hat{\Sigma}_{jk}|}}\exp\left(-\frac{1}{2}(\tau_k-\tau_j)^T\,\hat{\Sigma}_{jk}^{-1}\,(\tau_k-\tau_j)\right)$$
$$\hat{\Sigma}_{jk} = \frac{1}{2}\left(\Psi_j\,\hat{\sigma}_j^2\,\Psi_j^{-1} + \Psi_k\,\hat{\sigma}_k^2\,\Psi_k^{-1}\right).$$

One can determine the rate of change of $\hat{Q}_{jk}$ with ρ as follows. First, since the eigenvalues of the matrix $\hat{\Sigma}_{jk}$ increase with ρ, the term in the exponential approaches 0 as ρ increases. The variation

of $\hat{Q}_{jk}$ is thus given by the variation of $\pi\,|\hat{\sigma}_j\hat{\sigma}_k|/\sqrt{|\hat{\Sigma}_{jk}|}$. The term $\sqrt{|\hat{\Sigma}_{jk}|}$ has the same rate of change with ρ as $|\hat{\sigma}_j|$; therefore, $\sqrt{|\hat{\Sigma}_{jk}|} = O(1+\rho^2)$. This gives

$$\hat{Q}_{jk} = O(1+\rho^2) \tag{B.24}$$

and

$$\overline{\hat{P}}_{jk} = O((1+\rho^2)^{-1}). \tag{B.25}$$

Finally, combining the results (B.22), (B.23) and (B.25) in (B.21), and remembering that the coefficient products vary with ρ as $\hat{c}_j\hat{c}_k = O((1+\rho^2)^{-2})$, we conclude that the norm $\|N_h\hat{p}\|$ of the Hessian magnitude decreases with the filter size ρ at a rate of

$$\|N_h\hat{p}\| = O((1+\rho^2)^{-3/2})$$

which finishes the proof of the lemma.

□

B.4 Proof of Lemma 2

Proof: Remember from (5.3) and (5.8) that the projection of the unfiltered target pattern q onto $\mathcal{M}(p)$ is p_{λ_o}, and the projection of the filtered target pattern $\hat{q}$ onto $\mathcal{M}(\hat{p})$ is $\hat{p}_{\hat{\lambda}_o}$. Since $\hat{p}_{\hat{\lambda}_o}$ is the point on $\mathcal{M}(\hat{p})$ that has the smallest distance to $\hat{q}$, we have the following for the distance $\|\tilde{n}\|$ between $\hat{q}$ and $\mathcal{M}(\hat{p})$

$$\|\tilde{n}\| = \|\hat{q} - \hat{p}_{\hat{\lambda}_o}\| \leq \|\hat{q} - \hat{p}_{\lambda_o}\|$$

where $\hat{p}_{\lambda_o}$ is the filtered pattern $\hat{p}$ transformed by the transformation vector λ_o that is optimal in the alignment of the unfiltered patterns.

As discussed in Section 5.3.2, the deviation between the transformations λ_o and $\hat{\lambda}_o$ depends on the transformation model. Here we do not go into the investigation of the difference between $\hat{p}_{\hat{\lambda}_o}$ and $\hat{p}_{\lambda_o}$, and content ourselves with the upper bound $\|\hat{q}-\hat{p}_{\lambda_o}\|$ for $\|\tilde{n}\|$ in order to keep our analysis generic and valid for arbitrary transformation models. Our purpose is then to determine how the distance $\|\hat{q} - \hat{p}_{\lambda_o}\|$ depends on the initial noise level

$$\nu = \|n\| = \|q - p_{\lambda_o}\|$$

and the filter size ρ. The noise pattern n becomes

$$\hat{n} = \hat{q} - \widehat{p_{\lambda_o}}$$

when filtered by the filter kernel in (5.6), where $\widehat{p_{\lambda_o}}$ is the filtered version of p_{λ_o} with the same kernel. Now, an important observation is that $\hat{n} \neq \hat{q} - \hat{p}_{\lambda_o}$ for geometric transformations that change the scale of the pattern, because

$$\widehat{p_{\lambda_o}} \neq \hat{p}_{\lambda_o} \tag{B.26}$$

i.e., the operations of filtering a pattern and applying it a geometric transformation do not commute for such transformation models. The reason is that filtering modifies the scale matrices σ_k of atoms, and when the geometric transformation involves a scale change, the commutativity of these two operations fails. For geometric transformations that do not involve a scale change, the equality $\widehat{p_{\lambda_o}} = \hat{p}_{\lambda_o}$ holds. This is explained in more detail in the rest of this section. For the sake of generality, we base our derivation on the hypothesis (B.26) and proceed by bounding the deviation of $\hat{q} - \hat{p}_{\lambda_o}$ from $\hat{n}$. We thus use the following inequality for bounding $\|\tilde{n}\|$

$$\begin{aligned} \|\tilde{n}\| &\leq \|\hat{q} - \hat{p}_{\lambda_o}\| \leq \|\hat{q} - \widehat{p_{\lambda_o}}\| + \|\widehat{p_{\lambda_o}} - \hat{p}_{\lambda_o}\| \\ &= \|\hat{n}\| + \|\widehat{p_{\lambda_o}} - \hat{p}_{\lambda_o}\|. \end{aligned} \tag{B.27}$$

Hence, we achieve the examination of $\|\tilde{n}\|$ in two steps. We first determine the variation of $\|\hat{n}\|$ with the initial noise level ν and the filter size ρ. Then, we study the second term $\|\widehat{p_{\lambda_o}} - \hat{p}_{\lambda_o}\|$ as a function of the filter size. We finally put together these two results in order to obtain the variation of the term $\|\tilde{n}\|$.

Derivation of $\|\hat{n}\|$

We begin with deriving an analytical expression for the norm ν of the noise pattern n, whose variation with filtering is then easy to determine. Since the noise pattern n is in $L^2(\mathbb{R}^2)$, and the linear span of the Gaussian dictionary $\mathcal{D}$ is dense in $L^2(\mathbb{R}^2)$, n can be represented as the linear combination of a sequence of atoms in $\mathcal{D}$

$$n(X) = \sum_{k=1}^{\infty} \varsigma_k \, \phi_{\chi_k}(X)$$

where ς_k are the atom coefficients and χ_k are the atom parameters. Then,

$$\nu^2 = \|n\|^2 = \sum_{j=1}^{\infty}\sum_{k=1}^{\infty} \varsigma_j \varsigma_k \int_{\mathbb{R}^2} \phi_{\chi_j}(X)\phi_{\chi_k}(X)dX = \sum_{j=1}^{\infty}\sum_{k=1}^{\infty} \varsigma_j \varsigma_k R_{jk}$$

where the term R_{jk} is in the same form as the term Q_{jk} given in (B.10) and obtained with the atom parameters of n. Then, the squared norm of the filtered version of n is

$$\|\hat{n}\|^2 = \sum_{j=1}^{\infty}\sum_{k=1}^{\infty} \hat{\varsigma}_j \hat{\varsigma}_k \hat{R}_{jk}.$$

Now, the coefficients $\hat{\varsigma}_j$ have the same variation with ρ as $\hat{c}_j$; therefore, from (5.22), we obtain

$$\hat{\varsigma}_j \hat{\varsigma}_k = O((1+\rho^2)^{-2}).$$

Next, $\hat{R}_{jk}$ and $\hat{Q}_{jk}$ have the same variation with ρ since they are of the same form. Thus, the relation in (B.24) implies that

$$\hat{R}_{jk} = O(1+\rho^2). \tag{B.28}$$

Putting these results in the expression of $\|\hat{n}\|^2$, we see that the norm $\|\hat{n}\|$ of the filtered noise pattern decreases with ρ at a rate

$$\|\hat{n}\| = O((1+\rho^2)^{-1/2}).$$

Lastly, we look at the dependence of $\|\hat{n}\|$ on the initial noise level $\nu = \|n\|$. Since convolution with a filter kernel is a linear operator, the norm of the filtered noise pattern is linearly proportional to the norm of the initial noise pattern. Therefore, $\|\hat{n}\|$ varies linearly with ν. Combining this with the above result, we obtain the joint variation of $\|\hat{n}\|$ with ν and ρ as

$$\|\hat{n}\| = O(\nu\,(1+\rho^2)^{-1/2}). \tag{B.29}$$

Derivation of $\|\widehat{p_{\lambda_o}} - \hat{p}_{\lambda_o}\|$

In order to study the variation of the term $\|\widehat{p_{\lambda_o}} - \hat{p}_{\lambda_o}\|$ with the filter size in a convenient way, we assume that the composition of the geometric transformation $\lambda \in \Lambda$ generating the manifold $\mathcal{M}(p)$ and the geometric transformation $\gamma \in \Gamma$ generating the dictionary $\mathcal{D}$ can be represented as a transformation vector in Γ; i.e., for all $\lambda \in \Lambda$ and $\gamma \in \Gamma$, there exists $\gamma \circ \lambda \in \Gamma$ such that

$$A_\lambda(\phi_\gamma)(X) = \phi_{\gamma\circ\lambda}(X).$$

Note that this assumption holds for common geometric transformation models λ such as translations, rotations, scale changes and their combinations.

In order to ease the notation, we derive the variation of $\|\widehat{p_\lambda} - \hat{p}_\lambda\|$ for an arbitrary transformation vector λ, which is also valid for the optimal transformation vector λ_o. The transformed version p_λ of p can be represented as

$$p_\lambda(X) = \sum_{k=1}^{\infty} c_k\, \phi_{\gamma_k\circ\lambda}(X).$$

Let us denote the scale, rotation and translation matrices corresponding to the composite transformation vector $\gamma_k \circ \lambda$ respectively as $\sigma_k \diamond \lambda$, $\Psi_k \diamond \lambda$, and $\tau_k \diamond \lambda$. Then the filtered version of the transformed pattern p_λ is given by

$$\widehat{p_\lambda}(X) = \sum_{k=1}^{\infty} c_k\, \frac{|\sigma_k \diamond \lambda|}{|\widehat{\sigma_k \diamond \lambda}|}\, \phi_{\widehat{\gamma_k\circ\lambda}}(X)$$

where $\widehat{\sigma_k \diamond \lambda} = \sqrt{(\sigma_k \diamond \lambda)^2 + \Upsilon^2}$ is the scale matrix of the filtered atom parameters $\widehat{\gamma_k \circ \lambda}$. The rotation and translation matrices $\Psi_k \diamond \lambda$ and $\tau_k \diamond \lambda$ do not change as filtering affects only the scale matrix.

Now we derive the expression of $\hat{p}_\lambda$, which is obtained by filtering p first, and then applying it

a geometric transformation. Remember from Section 5.3.2 that the filtered pattern $\hat{p}$ is

$$\hat{p}(X) = \sum_{k=1}^{\infty} c_k \frac{|\sigma_k|}{|\hat{\sigma}_k|} \phi_{\hat{\gamma}_k}(X)$$

and the transformed version of $\hat{p}$ by λ is

$$\hat{p}_\lambda(X) = \sum_{k=1}^{\infty} c_k \frac{|\sigma_k|}{|\hat{\sigma}_k|} \phi_{\hat{\gamma}_k \circ \lambda}(X)$$

where the atom parameter vector $\hat{\gamma}_k \circ \lambda$ has the scale matrix $\hat{\sigma}_k \diamond \lambda = \sqrt{\sigma_k^2 + \Upsilon^2} \diamond \lambda$, rotation matrix $\Psi_k \diamond \lambda$ and translation vector $\tau_k \diamond \lambda$. Comparing the expressions of $\widehat{p_\lambda}$ and $\hat{p}_\lambda$, we see that these patterns have different atom scale matrices and atom coefficients if the transformation λ involves a scale change. The atoms of $\widehat{p_\lambda}$ and $\hat{p}_\lambda$ have the same rotation and translation matrices. Hence, if λ does not modify the scale matrices of atoms, we have $\sigma_k \diamond \lambda = \sigma_k$; therefore, $\widehat{p_\lambda} = \hat{p}_\lambda$.

The modification that the transformation λ makes in the atom scale parameters can be represented with a scale change matrix

$$S = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix}$$

such that

$$\sigma_k \diamond \lambda = S\, \sigma_k.$$

Here we avoid writing the dependence of S on λ for notational convenience. We also represent the scale change of all atoms with the same matrix S to ease the notation. However, this is not a strict hypothesis; i.e., since we treat the scale change parameters s_x and s_y as constants when examining the variation of $\|\widehat{p_\lambda} - \hat{p}_\lambda\|$ with the filter size ρ, our result is generalizable to the case when different atoms have different scale change matrices S_k.

With this representation, the atom scale matrices of $\widehat{p_\lambda}$ and $\hat{p}_\lambda$ are respectively obtained as

$$\widehat{\sigma_k \diamond \lambda} = \sqrt{S^2 \sigma_k^2 + \Upsilon^2}, \qquad \hat{\sigma}_k \diamond \lambda = S\sqrt{\sigma_k^2 + \Upsilon^2}$$

and the atom coefficients in these two patterns are respectively given by

$$c_k \frac{|\sigma_k \diamond \lambda|}{|\widehat{\sigma_k \diamond \lambda}|} = c_k \frac{|S\sigma_k|}{|\sqrt{S^2\sigma_k^2 + \Upsilon^2}|}\ , \qquad c_k \frac{|\sigma_k|}{|\hat{\sigma}_k|} = c_k \frac{|\sigma_k|}{|\sqrt{\sigma_k^2 + \Upsilon^2}|}.$$

The difference between the two patterns can then be upper bounded as

$$\begin{aligned} \|\widehat{p_\lambda} - \hat{p}_\lambda\| &= \left\| \sum_{k=1}^{\infty} c_k \frac{|S\sigma_k|}{|\sqrt{S^2\sigma_k^2 + \Upsilon^2}|} \phi_{\widehat{\gamma_k \circ \lambda}} - \sum_{k=1}^{\infty} c_k \frac{|\sigma_k|}{|\sqrt{\sigma_k^2 + \Upsilon^2}|} \phi_{\hat{\gamma}_k \circ \lambda} \right\| \\ &\leq \|e_1\| + \|e_2\| \end{aligned} \tag{B.30}$$

where

$$e_1 = \sum_{k=1}^{\infty} c_k \frac{|S\sigma_k|}{|\sqrt{S^2\sigma_k^2 + \Upsilon^2}|}(\phi_{\widehat{\gamma_k \circ \lambda}} - \phi_{\hat{\gamma}_k \circ \lambda})$$

$$e_2 = \sum_{k=1}^{\infty} c_k \left(\frac{|S\sigma_k|}{|\sqrt{S^2\sigma_k^2 + \Upsilon^2}|} - \frac{|\sigma_k|}{|\sqrt{\sigma_k^2 + \Upsilon^2}|} \right) \phi_{\hat{\gamma}_k \circ \lambda}.$$

In the following, we determine the rate of change of the terms $\|e_1\|$ and $\|e_2\|$ with the filter size ρ, which will then be used to estimate the dependence of $\|\widehat{p_\lambda} - \hat{p}_\lambda\|$ using (B.30). We momentarily omit the atom index k for lightening the notation. We begin with $\|e_1\|$. Since e_1 is a linear combination of atom differences, its variation with ρ is given by the product of the variations of the coefficients and the atom difference norms with ρ.

$$\|e_1\| = O\left(c \frac{|S\sigma|}{|\sqrt{S^2\sigma^2 + \Upsilon^2}|} \right) O\left(\|\phi_{\widehat{\gamma \circ \lambda}} - \phi_{\hat{\gamma} \circ \lambda}\| \right). \tag{B.31}$$

The coefficients decrease with ρ at a rate

$$c \frac{|S\sigma|}{|\sqrt{S^2\sigma^2 + \Upsilon^2}|} = O((1+\rho^2)^{-1}). \tag{B.32}$$

Next, we look at the dependence of the term $\|\phi_{\widehat{\gamma \circ \lambda}} - \phi_{\hat{\gamma} \circ \lambda}\|$ on ρ.

$$\begin{aligned} \|\phi_{\widehat{\gamma \circ \lambda}} - \phi_{\hat{\gamma} \circ \lambda}\|^2 = \int_{\mathbb{R}^2} \Big[& \phi\left((S^2\sigma^2 + \Upsilon^2)^{-1/2} (\Psi \diamond \lambda)^{-1} (X - \tau \diamond \lambda) \right) \\ & - \phi\left((S^2\sigma^2 + S^2\Upsilon^2)^{-1/2} (\Psi \diamond \lambda)^{-1} (X - \tau \diamond \lambda) \right) \Big]^2 dX \end{aligned}$$

Defining $a_x := s_x^2\sigma_x^2 + \rho^2$, $b_x := s_x^2(\sigma_x^2 + \rho^2)$, and defining a_y and b_y similarly, the evaluation of the above integral yields

$$\|\phi_{\widehat{\gamma \circ \lambda}} - \phi_{\hat{\gamma} \circ \lambda}\|^2 = \frac{\pi}{2}(\sqrt{a_x a_y} + \sqrt{b_x b_y}) - 2\pi \sqrt{\frac{a_x a_y b_x b_y}{(a_x + b_x)(a_y + b_y)}}.$$

As the parameters a_x, b_x, a_y, b_y increase with ρ at a rate of $O(1+\rho^2)$, the rate of increase of the squared norm of the atom difference $\phi_{\widehat{\gamma \circ \lambda}} - \phi_{\hat{\gamma} \circ \lambda}$ with ρ is given by

$$\|\phi_{\widehat{\gamma \circ \lambda}} - \phi_{\hat{\gamma} \circ \lambda}\|^2 = O(1+\rho^2).$$

Putting this result in (B.31) together with the decay rate of coefficients given in (B.32) yields

$$\|e_1\| = O((1+\rho^2)^{-1/2}). \tag{B.33}$$

Let us now examine the term $\|e_2\|$. The rate of change of $\|e_2\|$ can be estimated from the variation of the coefficients and the atom norms as follows

$$\|e_2\| = O\left(c\left[\frac{|S\sigma|}{|\sqrt{S^2\sigma^2+\Upsilon^2}|} - \frac{|\sigma|}{|\sqrt{\sigma^2+\Upsilon^2}|}\right]\right) O(\|\phi_{\hat{\gamma}\circ\lambda}\|).$$

The coefficients decay with ρ at a rate

$$c\left(\frac{|S\sigma|}{|\sqrt{S^2\sigma^2+\Upsilon^2}|} - \frac{|\sigma|}{|\sqrt{\sigma^2+\Upsilon^2}|}\right) = O((1+\rho^2)^{-1}).$$

Next, the squared norm of the atom is calculated as

$$\begin{aligned}\|\phi_{\hat{\gamma}\circ\lambda}\|^2 &= \int_{\mathbb{R}^2} \phi^2\left((S^2\sigma^2+S^2\Upsilon^2)^{-1/2}(\Psi\diamond\lambda)^{-1}(X-\tau\diamond\lambda)\right)dX \\ &= \frac{\pi}{2}s_x s_y\sqrt{(\sigma_x^2+\rho^2)(\sigma_y^2+\rho^2)}\end{aligned}$$

which shows that the atom norm increases with ρ at a rate

$$\|\phi_{\hat{\gamma}\circ\lambda}\| = O((1+\rho^2)^{1/2}).$$

Hence, we obtain the order of dependence of $\|e_2\|$ on ρ as

$$\|e_2\| = O((1+\rho^2)^{-1/2}). \tag{B.34}$$

Finally, from (B.33), (B.34), and the inequality in (B.30), we obtain the variation of the error term $\|\widehat{p_\lambda} - \hat{p}_\lambda\|$ with ρ as

$$\|\widehat{p_\lambda} - \hat{p}_\lambda\| = O((1+\rho^2)^{-1/2}). \tag{B.35}$$

Variation of $\|\tilde{n}\|$ with noise level and filter size

We can now put together the results obtained so far to determine the variation of the noise term $\|\tilde{n}\|$. Using the upper bound on $\|\tilde{n}\|$ given in (B.27) and the variations of $\|\hat{n}\|$ and $\|\widehat{p_{\lambda_o}} - \hat{p}_{\lambda_o}\|$ given in (B.29) and (B.35), the joint variation of the noise term $\|\tilde{n}\|$ with the initial noise level ν and the filter size ρ is obtained as

$$\|\tilde{n}\| = O\left((\nu+1)(1+\rho^2)^{-1/2}\right)$$

for geometric transformations that change the scale of the pattern. We see that the initial noise level ν is augmented by an offset term, which results from the fact that the operations of filtering and applying a geometric transformation do not commute when the transformation involves a scale change. Since filtering and transforming commute for transformation models that do not modify the scales of atoms, the second error term $\|\widehat{p_{\lambda_o}} - \hat{p}_{\lambda_o}\|$ in (B.27) vanishes for such geometric transformations. Thus, if the transformation model λ does not involve a scale change, the variation

of $\|\tilde{n}\|$ is given by

$$\|\tilde{n}\| = O\left(\nu(1+\rho^2)^{-1/2}\right).$$

This finishes the proof of the lemma.

□

Appendix C

Appendix

C.1 Exact expressions for the parameters in Lemma 3

Here we present the expressions for the terms $\overline{\mathfrak{c}}_{jk}$ and $\underline{\mathfrak{d}}_{jk}$ used in Lemma 3. Let $\alpha_k = \lambda_{\min}(\Phi_k)$ and $\beta_k = \lambda_{\max}(\Phi_k)$ denote respectively the smaller and greater eigenvalues of Φ_k. Since Φ_k is a positive definite matrix, $\beta_k \geq \alpha_k > 0$. Let $\tau_k = [\tau_{x,k} \;\; \tau_{y,k}]^T$,

$$
\begin{aligned}
\mathfrak{b}^x_{jk} &= (\beta_j + \beta_k) \max\left\{ \left(b + \bar{t}_0 - \frac{\beta_j \tau_{x,j} + \beta_k \tau_{x,k}}{\beta_j + \beta_k} \right)^2, \left(-b - \bar{t}_0 - \frac{\beta_j \tau_{x,j} + \beta_k \tau_{x,k}}{\beta_j + \beta_k} \right)^2 \right\} \\
\mathfrak{c}^x_{jk} &= (\beta_j + \beta_k) \max\left\{ \left(b + \bar{t}_0 \frac{\beta_j}{\beta_j + \beta_k} - \frac{\beta_j \tau_{x,j} + \beta_k \tau_{x,k}}{\beta_j + \beta_k} \right)^2, \left(-b - \bar{t}_0 \frac{\beta_j}{\beta_j + \beta_k} - \frac{\beta_j \tau_{x,j} + \beta_k \tau_{x,k}}{\beta_j + \beta_k} \right)^2 \right\} \\
\mathfrak{d}^x_{jk} &= \frac{\beta_j \beta_k}{\beta_j + \beta_k} \max\left\{ (-\bar{t}_0 + \tau_{x,k} - \tau_{x,j})^2, (\bar{t}_0 + \tau_{x,k} - \tau_{x,j})^2 \right\}
\end{aligned}
$$

and $\mathfrak{b}^y_{jk}$, $\mathfrak{c}^y_{jk}$ and $\mathfrak{d}^y_{jk}$ be defined similarly by replacing x with y in the above expressions. Let

$$
\underline{H}^x_{jk} = -\frac{\beta_j \, \tau_{x,j} + \beta_k \, \tau_{x,k}}{\beta_j + \beta_k}, \qquad \underline{G}^x_{jk} = \frac{\beta_j \, \tau^2_{x,j} + \beta_k \, \tau^2_{x,k}}{\beta_j + \beta_k}
$$

and $\underline{H}^y_{jk}$, $\underline{G}^y_{jk}$ be defined similarly by replacing x with y in the above expressions. Also, let $\overline{H}^x_{jk}$, $\overline{H}^y_{jk}$, $\overline{G}^x_{jk}$, $\overline{G}^y_{jk}$ denote the terms obtained by substituting β_j, β_k with α_j, α_k in the expressions of respectively $\underline{H}^x_{jk}$, $\underline{H}^y_{jk}$, $\underline{G}^x_{jk}$, $\underline{G}^y_{jk}$. [1] Then, let

$$
\begin{aligned}
\underline{\mathcal{D}}^x_{jk} = &\frac{\sqrt{\pi}}{4b} \frac{1}{\sqrt{\beta_j + \beta_k}} \exp\left(-(\beta_j + \beta_k) \left(\underline{G}^x_{jk} - (\underline{H}^x_{jk})^2 \right) \right) \\
&\cdot \left[\operatorname{erf}\left(\sqrt{\beta_j + \beta_k} \, (b + \underline{H}^x_{jk}) \right) - \operatorname{erf}\left(\sqrt{\beta_j + \beta_k} \, (-b + \underline{H}^x_{jk}) \right) \right]
\end{aligned}
$$

[1] The terms written as $\underline{H}^x_{jk}, \ldots, \overline{G}^y_{jk}$ here correspond to the terms $\underline{H}^x_{jk}(0,0), \ldots, \overline{G}^y_{jk}(0,0)$ in [102].

and $\underline{\mathcal{D}}^y_{jk}$ be defined similarly, and let us denote by $\overline{\mathcal{D}}^x_{jk}$, $\overline{\mathcal{D}}^y_{jk}$ the terms obtained by replacing β_j, β_k with α_j, α_k in the expressions of $\underline{\mathcal{D}}^x_{jk}$, $\underline{\mathcal{D}}^y_{jk}$. Then defining

$$\begin{aligned}
\underline{\underline{\mathcal{B}}}_{jk} &= \exp\left(-\mathfrak{b}^x_{jk} - \mathfrak{b}^y_{jk} - \frac{\beta_j\beta_k\,\|\tau_k-\tau_j\|^2}{(\beta_j+\beta_k)}\right), \qquad \overline{\overline{\mathcal{B}}}_{jk} = \frac{\pi}{4b^2\,(\alpha_j+\alpha_k)}\exp\left(-\frac{\alpha_j\alpha_k\,\|\tau_k-\tau_j\|^2}{(\alpha_j+\alpha_k)}\right)\\
\underline{\underline{\mathcal{C}}}_{jk} &= \exp\left(-\mathfrak{c}^x_{jk} - \mathfrak{c}^y_{jk} - \mathfrak{d}^x_{jk} - \mathfrak{d}^y_{jk}\right), \quad \overline{\overline{\mathcal{C}}}_{jk} = \frac{\pi}{4b^2\,(\alpha_j+\alpha_k)}, \quad \underline{\mathcal{D}}_{jk} = \underline{\mathcal{D}}^x_{jk}\,\underline{\mathcal{D}}^y_{jk}, \quad \overline{\mathcal{D}}_{jk} = \overline{\mathcal{D}}^x_{jk}\,\overline{\mathcal{D}}^y_{jk}
\end{aligned}$$

the parameters $\overline{\mathfrak{c}}_{jk}$ and $\underline{\mathfrak{d}}_{jk}$ are given by

$$\begin{aligned}
\overline{\mathfrak{c}}_{jk} &= \overline{\overline{\mathcal{B}}}_{jk} - 2\underline{\underline{\mathcal{C}}}_{jk} + \overline{\mathcal{D}}_{jk}\\
\underline{\mathfrak{d}}_{jk} &= \underline{\underline{\mathcal{B}}}_{jk} - 2\overline{\overline{\mathcal{C}}}_{jk} + \underline{\mathcal{D}}_{jk}.
\end{aligned}$$

C.2 Exact expressions for the parameters in Lemma 4

We now give the expressions for the terms $\underline{r}_0$, $\underline{r}_2$ and $\underline{r}_3$ used in Lemma 4. Let $\underline{r}_0$ be the smaller eigenvalue of the following positive definite matrix

$$R_0 = \sum_{j=1}^{K}\sum_{k=1}^{K} c_j c_k\, Q_{jk}\left(\Sigma^{-1}_{jk} - \Sigma^{-1}_{jk}(\tau_k-\tau_j)(\tau_k-\tau_j)^T\Sigma^{-1}_{jk}\right).$$

Next, as shown in [102], the following upper and lower bounds can be obtained

$$\begin{aligned}
a^2_{jk} &\geq \underline{a}^2_{jk} = \frac{1}{4}\lambda^2_{\min}(\Sigma^{-1}_{jk}), \qquad a^2_{jk} \leq \overline{a}^2_{jk} = \frac{1}{4}\lambda^2_{\max}(\Sigma^{-1}_{jk})\\
b^2_{jk}a_{jk} &\leq \overline{b}^2_{jk}\overline{a}_{jk} = \frac{1}{8}\lambda_{\max}(R^{jk}_2)\lambda_{\max}(\Sigma^{-1}_{jk}), \qquad b^4_{jk} \leq \overline{b}^4_{jk} = \frac{1}{16}\lambda^2_{\max}(R^{jk}_2)
\end{aligned}$$

where $R^{jk}_2 = \Sigma^{-1}_{jk}(\tau_k-\tau_j)(\tau_k-\tau_j)^T\Sigma^{-1}_{jk}$. Then, we have

$$\underline{r}'_2 = \sum_{(j,k)\in J^+} c_j c_k Q_{jk}(-8\,\overline{b}^4_{jk} + -6\,\overline{a}^2_{jk}) \quad + \sum_{(j,k)\in J^-} c_j c_k Q_{jk}(24\,\overline{b}^2_{jk}\overline{a}_{jk} - 6\,\underline{a}^2_{jk})$$

and $\underline{r}_2$ is given by $\underline{r}_2 = \min(\underline{r}'_2, 0)$. Finally,

$$\underline{r}_3 = -\sum_{j=1}^{K}\sum_{k=1}^{K}\frac{5.46}{2^{5/2}}\,|c_j c_k|\,\frac{\pi|\sigma_j\sigma_k|}{\sqrt{|\Sigma_{jk}|}}\,(\lambda_{\max}(\Sigma^{-1}_{jk}))^{5/2}.$$

C.3 Exact expressions for the parameters in Lemma 5

Now we present the terms $\overline{\mathfrak{e}}_{jk}$ and $\underline{\mathfrak{f}}_{jk}$ used in Lemma 5. Let

$$\overline{L}_j = \left(\frac{(3^{3/4} + 3^{5/4})e^{-\frac{\sqrt{3}}{2}}}{16} + \frac{3\sqrt{\pi}}{2^{9/2}} \right) \frac{1}{\alpha_j^{5/2}}, \qquad \overline{M}_j = \sqrt{\frac{\pi}{2\alpha_j}}, \quad \overline{N}_j := \left(\frac{e^{-\frac{1}{2}}}{4} + \frac{\sqrt{\pi}}{2^{5/2}} \right) \frac{1}{\alpha_j^{3/2}}$$

$$\overline{\overline{\mathcal{E}}}_j = \frac{\beta_j^4}{4b^2} \left(2\overline{L}_j \overline{M}_j + 2\overline{N}_j^2 \right), \qquad \overline{\overline{\mathcal{F}}}_j = \frac{1}{4b^2} \overline{M}_j^2.$$

Then, the terms $\overline{\mathfrak{e}}_{jk}$ and $\underline{\mathfrak{f}}_{jk}$ are given by

$$\overline{\mathfrak{e}}_{jk} = 16\sqrt{\overline{\overline{\mathcal{E}}}_j \overline{\overline{\mathcal{E}}}_k} + 4\beta_j\beta_k \, \overline{\overline{\mathcal{B}}}_{jk}, \qquad \underline{\mathfrak{f}}_{jk} = -8 \, \beta_k \sqrt{\overline{\overline{\mathcal{E}}}_j \overline{\overline{\mathcal{F}}}_k} - 8 \, \beta_j \sqrt{\overline{\overline{\mathcal{F}}}_j \overline{\overline{\mathcal{E}}}_k} + 4\alpha_j\alpha_k \, \underline{\underline{\mathcal{B}}}_{jk}.$$

Bibliography

[1] Online photo gallery "Copyright-free Paris". [Online]. Available: http://www.flickr.com/photos/54156444@N05/galleries/72157624889329591

[2] N. Vasconcelos and A. Lippman, "A multiresolution manifold distance for invariant image similarity," *IEEE Transactions on Multimedia*, vol. 7, no. 1, pp. 127–142, 2005.

[3] P. Simard, B. Victorri, Y. LeCun, and J. Denker, "Tangent prop: a formalism for specifying selected invariances in adaptive networks," in *Advances in Neural Information Processing Systems (NIPS 1991)*, vol. 4, Denver, CO, 1992.

[4] A. W. Fitzgibbon and A. Zisserman, "Joint manifold distance: A new approach to appearance based clustering," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2003.

[5] J. B. Tenenbaum, V. de Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction." *Science*, vol. 290, no. 5500, pp. 2319–2323, December 2000.

[6] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, pp. 2323–2326, 2000.

[7] D. L. Donoho and C. Grimes, "Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 100, no. 10, pp. 5591–5596, May 2003.

[8] G. Peyré, "Manifold models for signals and images," *Computer Vision and Image Understanding*, vol. 113, no. 2, pp. 249–260, 2009.

[9] M. B. Wakin, D. L. Donoho, H. Choi, and R. G. Baraniuk, "The multiscale structure of non-differentiable image manifolds," vol. 5914, no. 1. SPIE, 2005.

[10] D. L. Donoho and C. Grimes, "Image manifolds which are isometric to Euclidean space," *Journal of Mathematical Imaging and Vision*, vol. 23, no. 1, pp. 5–24, July 2005.

[11] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Computation*, vol. 15, no. 6, pp. 1373–1396, Jun. 2003.

[12] R. R. Coifman, S. Lafon, A. B. Lee, M. Maggioni, B. Nadler, F. Warner, and S. W. Zucker, "Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 102, no. 21, pp. 7426–7431, May 2005.

[13] Z. Zhang and H. Zha, "Principal manifolds and nonlinear dimension reduction via local tangent space alignment," *SIAM Journal of Scientific Computing*, vol. 26, pp. 313–338, 2005.

[14] Y. Bengio, J. F. Paiement, P. Vincent, O. Delalleau, N. L. Roux, and M. Ouimet, "Out-of-sample extensions for LLE, ISOMAP, MDS, Eigenmaps, and Spectral Clustering," in *Adv. Neural Inf. Process. Syst.* MIT Press, 2004, pp. 177–184.

[15] P. Dollár, V. Rabaud, and S. Belongie, "Non-isometric manifold learning: Analysis and an algorithm," in *Int. Conf. Mach. Learn.*, June 2007.

[16] A. M. Álvarez-Meza, J. Valencia-Aguirre, G. Daza-Santacoloma, C. D. Acosta-Medina, and G. Castellanos-Domínguez, "Image synthesis based on manifold learning," in *CAIP (2)*, 2011, pp. 405–412.

[17] D. de Ridder, O. Kouropteva, O. Okun, M. Pietikainen, and R. P. W. Duin, "Supervised locally linear embedding." in *Proc. Int. Conf. Art. Neur. Networks*, 2003, pp. 333–341.

[18] E. Kokiopoulou and P. Frossard, "Minimum distance between pattern transformation manifolds: Algorithm and applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 7, pp. 1225–1238, Jul. 2009.

[19] S. G. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Trans. Signal Process.*, vol. 41, no. 12, pp. 3397–3415, Dec 1993.

[20] J. P. Antoine, R. Murenzi, P. Vandergheynst, and S. Ali, *Two-Dimensional Wavelets and their Relatives*, ser. Signal Processing. Cambridge University Press, 2004.

[21] J. Maintz and M. Viergever, "A survey of medical image registration," *Medical Image Analysis*, vol. 2, no. 1, pp. 1–36, 1998.

[22] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proc. 7th Intl. Joint Conf. on Artificial Intelligence*, 1981, pp. 674–679.

[23] C. A. Glasbey and K. V. Mardia, "A review of image warping methods," *Journal of Applied Statistics*, vol. 25, no. 2, pp. 155–171, 1998.

[24] G. Tziritas and C. Labit, *Motion Analysis for Image Sequence Coding.* New York, NY, USA: Elsevier Science Inc., 1994.

[25] L. G. Brown, "A survey of image registration techniques," *ACM Comput. Surv.*, vol. 24, no. 4, pp. 325–376, Dec. 1992.

[26] J. L. Barron, D. J. Fleet, and S. S. Beauchemin, "Performance of optical flow techniques," *International Journal of Computer Vision*, vol. 12, no. 1, pp. 43–77, Feb. 1994.

[27] D. Robinson and P. Milanfar, "Fundamental performance limits in image registration," *IEEE Trans. Img. Proc.*, vol. 13, no. 9, pp. 1185–1199, Sep. 2004.

[28] P. Anandan, "A computational framework and an algorithm for the measurement of visual motion," *International Journal of Computer Vision*, vol. 2, no. 3, pp. 283–310, 1989.

[29] A. N. Netravali and J. D. Robbins, "Motion-compensated television coding: Part I," *Bell System Tech. Jour.*, vol. 58, pp. 631–670, Mar. 1979.

[30] D. Walker and K. Rao, "Improved pel-recursive motion compensation," *IEEE Trans. Communications*, vol. 32, no. 10, pp. 1128 – 1134, Oct. 1984.

[31] C. Cafforio and F. Rocca, *The differential method for image motion estimation.* Springer-Verlag, 1983.

[32] J. R. Bergen, P. Anandan, K. J. Hanna, and R. Hingorani, "Hierarchical model-based motion estimation." Springer-Verlag, 1992, pp. 237–252.

[33] J. Fabrizio, S. Dubuisson, and D. Béréziat, "Motion compensation based on tangent distance prediction for video compression," *Sig. Proc.: Image Comm.*, vol. 27, no. 2, pp. 153–171, 2012.

[34] O. Nestares and D. J. Heeger, "Robust multiresolution alignment of MRI brain volumes," *Magnetic Resonance in Medicine*, vol. 43, no. 5, pp. 705–715, 2000.

[35] E. Mémin and P. Pérez, "Hierarchical estimation and segmentation of dense motion fields," *International Journal of Computer Vision*, vol. 46, pp. 129–155, Feb. 2002.

[36] İ. Ş. Yetik and A. Nehorai, "Performance bounds on image registration," *IEEE Trans. Signal Proc.*, vol. 54, no. 5, pp. 1737 – 1749, May 2006.

[37] N. Sabater, J. M. Morel, and A. Almansa, "How accurate can block matches be in stereo vision?" *SIAM Journal on Imaging Sciences*, vol. 4, no. 1, pp. 472–500, 2011.

[38] J. K. Kearney, W. B. Thompson, and D. L. Boley, "Optical flow estimation: An error analysis of gradient-based methods with local optimization," *IEEE Trans. Pattern Anal. Machine Intel.*, Mar. 1987.

[39] J. W. Brandt, "Analysis of bias in gradient-based optical flow estimation," in *1994 Conf. Rec. of the 28th Asilomar Conf. on Signals, Systems and Computers*, vol. 1, 1994, pp. 721–725.

[40] T. Q. Pham, M. Bezuijen, L. J. van Vliet, K. Schutte, and C. L. Luengo, "Performance of optimal registration estimators," in *Proc. SPIE*, 2005, pp. 133–144.

[41] M. Lefébure and L. Cohen, "Image registration, optical flow and local rigidity," *Journal of Mathematical Imaging and Vision*, vol. 14, no. 2, pp. 131–147, 2001.

[42] L. Alvarez, J. Weickert, and J. Sánchez, "A scale-space approach to nonlocal optical flow calculations," in *Proc. 2nd Inter. Conf.on Scale-Space Theories in Computer Vision*, ser. SCALE-SPACE '99. London, UK, UK: Springer-Verlag, 1999, pp. 235–246.

[43] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification (2nd Edition).* Wiley-Interscience, 2000.

[44] I. T. Jolliffe, *Principal Component Analysis.* Springer Verlag, New York, 1986.

[45] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals Eugen.*, vol. 7, pp. 179–188, 1936.

[46] G. J. Mclachlan, *Discriminant Analysis and Statistical Pattern Recognition (Wiley Series in Probability and Statistics).* Wiley-Interscience, 2004.

[47] M. A. Turk and A. P. Pentland, "Face recognition using eigenfaces," in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition.* IEEE Comput. Sco. Press, 1991, pp. 586–591.

[48] P. N. Belhumeur, J. Hespanha, and D. J. Kriegman, "Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 711–720, 1997.

[49] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 210–227, Feb. 2009.

[50] E. Elhamifar and R. Vidal, "Sparse subspace clustering: Algorithm, theory, and applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2013.

[51] I. Tosic and P. Frossard, "Dictionary learning: What is the right representation for my signal?" *IEEE Signal Processing Magazine*, vol. 28, no. 2, pp. 27–38, 2011.

[52] S. Theodoridis and K. Koutroumbas, *Pattern Recognition, Fourth Edition*, 4th ed. Academic Press, 2008.

[53] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proc. 5th annual workshop on computational learning theory*, ser. COLT '92. New York, NY, USA: ACM, 1992, pp. 144–152.

[54] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, pp. 91–110, 2004.

[55] T. Ojala, M. Pietikäinen, and D. Harwood, "A comparative study of texture measures with classification based on featured distributions," *Pattern Recognition*, vol. 29, no. 1, pp. 51–59, Jan. 1996.

[56] H. Freeman, "On the encoding of arbitrary geometric configurations," *IRE Transactions on Electronic Computers*, vol. EC-10, no. 2, pp. 260–268, 1961.

[57] A. Blum and S. Chawla, "Learning from labeled and unlabeled data using graph mincuts," in *Proc. Int. Conf. Machine Learning*, 2001, pp. 19–26.

[58] X. Zhu and Z. Ghahramani, "Learning from labeled and unlabeled data with label propagation," *Technical Report CMU-CS-03-175*, 2003.

[59] M. Belkin and P. Niyogi, "Using manifold structure for partially labelled classification," in *NIPS*, 2002.

[60] M. Gong, L. Jiao, L. Bo, L. Wang, and X. Zhang, "Image texture classification using a manifold distance based evolutionary clustering method," *Optical Engineering*, vol. 47, no. 7, 2008.

[61] K. Lee, J. Ho, M. H. Yang, and D. Kriegman, "Video-based face recognition using probabilistic appearance manifolds," in *CVPR*, 2003, pp. 313–320.

[62] R. Xiao, Q. Zhao, D. Zhang, and P. Shi, "Facial expression recognition on multiple manifolds," *Pattern Recogn.*, vol. 44, no. 1, pp. 107–116, Jan. 2011.

[63] M. A. Davenport, M. F. Duarte, M. B. Wakin, J. N. Laska, D. Takhar, K. F. Kelly, and R. Baraniuk, "The smashed filter for compressive classification and target recognition," in *Proc. SPIE Computational Imaging*, 2007, p. 6498.

[64] H. Tyagi, E. Vural, and P. Frossard, "Tangent space estimation for smooth embeddings of Riemannian manifolds," *Information and Inference*, vol. 2, no. 1, pp. 69–114, 2013.

[65] A. Neri and G. Jacovitti, "Maximum likelihood localization of 2-d patterns in the Gauss-Laguerre transform domain: Theoretic framework and preliminary results," *IEEE Transactions on Image Processing*, vol. 13, no. 1, pp. 72–86, January 2004.

[66] M. Farrashkhalvat and J. P. Miles, *Basic Structured Grid Generation with an Introduction to Unstructured Grid Generation.* Butterworth-Heinemann, 2003.

[67] G. Peyré and L. D. Cohen, "Geodesic remeshing using front propagation," *International Journal of Computer Vision*, vol. 69, no. 1, pp. 145–156, 2006.

[68] Y. Linde, A. Buzo, and R. Gray, "An algorithm for vector quantizer design," *IEEE Transactions on Communications*, vol. 28, no. 1, pp. 84–95, Jan 1980.

[69] S. Anderson and A. Nehorai, "Analysis of a polarized seismic wave model," *IEEE Transactions on Signal Processing*, vol. 44, no. 2, pp. 379–386, 1996.

[70] Q. Du, M. Emelianenko, and L. Ju, "Convergence of the Lloyd algorithm for computing centroidal Voronoi tessellations," *SIAM J. Numerical Analysis*, vol. 44, no. 1, pp. 102–119, 2006.

[71] P. Shilane, P. Min, M. Kazhdan, and T. Funkhouser, *The Princeton Shape Benchmark.* Shape Modeling International, Genova, Italy, June 2004.

[72] A. Gersho and R. M. Gray, *Vector quantization and signal compression.* Norwell, MA, USA: Kluwer Academic Publishers, 1991.

[73] C. Kuglin and D. Hines, "The phase correlation image alignment method," *IEEE International Conference on Cybernetics and Society*, pp. 163–165, September 1975.

[74] R. Matungka, Y. F. Zheng, and R. L. Ewing, "Image registration using adaptive polar transform," *IEEE Transactions on Image Processing*, vol. 18, pp. 2340–2354, October 2009.

[75] E. Learned-Miller, "Data driven image models through continuous joint alignment," *IEEE Trans. on Pattern Anal. and Machine Intel.*, vol. 28, pp. 236–250, 2006.

[76] C. Atkeson, A. W. Moore, and S. Schaal, "Locally weighted learning," *AI Review*, vol. 11, pp. 11–73, April 1997.

[77] A. W. Moore, J. Schneider, and K. Deng, "Efficient locally weighted polynomial regression predictions," in *Int. Conf. on Machine Learning*, 1997, pp. 236–244.

[78] E. Jonsson and M. Felsberg, "Accurate interpolation in appearance-based pose estimation," in *Proc. 15th Scandinavian Conf. on Image Anal.*, ser. SCIA'07. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 1–10.

[79] Q. Qiu, V. M. Patel, P. Turaga, and R. Chellappa, "Domain adaptive dictionary learning," in *Proceedings of the 12th European conference on Computer Vision*, ser. ECCV'12. Berlin, Heidelberg: Springer-Verlag, 2012, pp. 631–645.

[80] J. A. Tropp, A. C. Gilbert, and M. J. Strauss, "Algorithms for simultaneous sparse approximation part I: Greedy pursuit," *Signal Processing*, vol. 86, no. 3, pp. 572–588, 2006.

[81] E. Kokiopoulou and P. Frossard, "Semantic coding by supervised dimensionality reduction," *IEEE Trans. Multimedia*, vol. 10, no. 5, pp. 806–818, 2008.

[82] B. Mailhé, S. Lesage, R. Gribonval, F. Bimbot, and P. Vandergheynst, "Shift-invariant dictionary learning for sparse representations: Extending K-SVD," in *Proc. Eur. Sig. Proc. Conf.*, 2008.

[83] P. Jost, S. Lesage, P. Vandergheynst, and R. Gribonval, "MoTIF: An efficient algorithm for learning translation-invariant dictionaries," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Proc.*, vol. 5, 2006, pp. 857–860.

[84] J. Mairal, G. Sapiro, and M. Elad, "Multiscale sparse image representation with learned dictionaries," in *Proc. IEEE Int. Conf. Image Proc.*, vol. 3, Sep 2007, pp. 105–108.

[85] P. Sallee and B. Olshausen, "Learning sparse multiscale image representations," in *Adv. in Neur. Inf. Proc. Sys.* MIT Press, 2002.

[86] C. Ekanadham, D. Tranchina, and E. P. Simoncelli, "Sparse decomposition of transformation-invariant signals with continuous basis pursuit," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Proc.*, 2011, pp. 4060–4063.

[87] R. Horst, P. M. Pardalos, and N. V. Thoai, *Introduction to Global Optimization.* Dordrecht, The Netherlands: Kluwer Academic Publishers, 2000.

[88] P. Hartman, "On functions representable as a difference of convex functions," *Pacific Journal of Math.*, no. 9, pp. 707–713, 1959.

[89] R. Horst and N. V. Thoai, "DC programming: overview," *J. Optim. Theory Appl.*, vol. 103, pp. 1–43, October 1999.

[90] P. Tao and L. An, "Convex analysis approach to DC programming: Theory, algorithms and applications," *Acta Mathematica Vietnamica*, vol. 22, no. 1, pp. 289–355, 1997.

[91] A. L. Yuille and A. Rangarajan, "The concave-convex procedure (CCCP)," in *Adv. in Neur. Inf. Proc. Sys.* MIT Press, 2002.

[92] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[93] T. Kanade, J. F. Cohn, and Y. Tian, "Comprehensive database for facial expression analysis," *Fourth IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 46–53, 2000.

[94] R. Sala Llonch, E. Kokiopoulou, I. Tošić, and P. Frossard, "3D face recognition with sparse spherical representations," *Pattern Recogn.*, vol. 43, no. 3, pp. 824–834, Mar. 2010.

[95] K. Lee, J. Ho, and D. Kriegman, "Acquiring linear subspaces for face recognition under variable lighting," *IEEE Trans. Pattern Anal. Mach. Intelligence*, vol. 27, no. 5, pp. 684–698, 2005.

[96] Natural History Museum. (2011) Microbiology video collection. [Online]. Available: http://www.nhm.ac.uk/research-curation/research/projects/protistvideo/about.dsml

[97] A. W. Fitzgibbon and A. Zisserman, "Joint manifold distance: a new approach to appearance based clustering," *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, p. 26, 2003.

[98] T. Lindeberg, *Scale-Space Theory in Computer Vision.* Kluwer Academic Publishers, 1994.

[99] M. P. Wand and M. C. Jones, *Kernel Smoothing.* Chapman and Hall/CRC, 1995.

[100] P. Y. Simard, Y. LeCun, J. S. Denker, and B. Victorri, "Transformation invariance in pattern recognition – tangent distance and tangent propagation," *International Journal of Imaging Systems and Technology*, vol. 11, no. 3, 2001.

[101] H. Mobahi, C. L. Zitnick, and Y. Ma, "Seeing through the blur," in *CVPR*, 2012, pp. 1736–1743.

[102] E. Vural and P. Frossard, "Analysis of descent-based image registration," *EPFL-REPORT-183845*, Available at: http://infoscience.epfl.ch/record/183845.

[103] A. P. Witkin, "Scale-space filtering." in *8th Int. Joint Conf. Artificial Intelligence*, vol. 2, Karlsruhe, Aug. 1983, pp. 1019–1022.

[104] J. Munkres, *Analysis on manifolds*, ser. Advanced Book Classics. Addison-Wesley Pub. Co., Advanced Book Program, 1991.

CURRICULUM VITAE

EL İF VURAL

Address: EPFL STI IEL LTS4, Station 11, Lausanne CH-1015, Switzerland
Tel: 0041 21 693 7806
Email: elif.vural@epfl.ch

EDUCATION

Ph.D. in Electrical Engineering
Ecole Polytechnique Fédérale de Lausanne, Lausanne, Switzerland *2008-2013*

M.Sc. in Electrical and Electronics Engineering
Middle East Technical University, Ankara, Turkey *2006-2008*

B.Sc. in Electrical and Electronics Engineering
Middle East Technical University, Ankara, Turkey *2002-2006*

B.Sc. in Mathematics
Middle East Technical University, Ankara, Turkey *2002-2006*

RESEARCH EXPERIENCE

Ph.D. Research at Ecole Polytechnique Fédérale de Lausanne *2008-2013*
Supervised by: Prof. Pascal Frossard
Research on: Image analysis and classification, low-dimensional data representations, multi-view classification, sampling of Riemannian manifolds.

M.Sc. Research at Middle East Technical University *2006-2008*
3DTV European Commission NoE Project
Supervised by: Prof. A. Aydın Alatan
Research on: 3-D scene reconstruction from image sequences

TEACHING EXPERIENCE

Teaching Assistant at Ecole Polytechnique Fédérale de Lausanne

Courses: Digital Signal Processing

Master Projects Assisted: Calibration of images with 3-D range scanner data (Víctor Adalid López), Approximation of face images with analytical models (Yu Zhan), Local sampling analysis for quadratic embeddings of Riemannian manifolds (Hemant Tyagi)

Student Internship Projects Assisted: Similarity analysis of observations under affine projections (Ozan Şener), Alignment of uncalibrated images for multi-view classification (Ömer Sercan Arık), Learning analytic dictionaries for classification (Ahmet Caner Yüzügüler)

Teaching Assistant at Middle East Technical University

Courses: Electrical Circuits Laboratory

LANGUAGES

English (proficient), French (upper-intermediate), Turkish (native)

PROFESSIONAL ACTIVITIES

Reviewer in Journals: IEEE Transactions on Image Processing, IEEE Signal Processing Letters, Information and Inference: A Journal of the IMA

Reviewer in Conferences: 3DTV-Con, ICASSP

Invited talks in institutions: Aix-Marseille Université (Transformation-invariant image analysis with manifold models)

PUBLICATIONS

JOURNAL PAPERS

• E. Vural and P. Frossard, "Analysis of Descent-Based Image Registration", Submitted to SIAM Journal on Imaging Sciences.

• H. Tyagi, E. Vural and P. Frossard, "Tangent Space Estimation for Smooth Embeddings of Riemannian Manifolds", Information and Inference, vol. 2, no. 1, pp. 69-114, 2013.

• E. Vural and P. Frossard, "Learning Smooth Pattern Transformation Manifolds", IEEE Transactions on Image Processing, vol. 22, no. 4, pp. 1311-1325, 2013.

• E. Vural and P. Frossard, "Discretization of Parametrizable Signal Manifolds", IEEE Transactions on Image Processing, vol. 20, no. 12, pp. 3621-3633, 2011.

CONFERENCE PAPERS

• E. Vural and P. Frossard, "Analysis of Hierarchical Image Alignment with Descent Methods", 10th International Conference on Sampling Theory and Applications, 2013.

• H. Tyagi, E. Vural and P. Frossard, "Tangent Space Estimation Bounds for Smooth Manifolds", 10th International Conference on Sampling Theory and Applications, 2013.

• E. Vural and P. Frossard, "Learning Patten Transformation Manifolds for Classification", IEEE International Conference on Image Processing, 2012.

• E. Vural and P. Frossard, "Learning Pattern Transformation Manifolds with Parametric Atom Selection", 9th International Conference on Sampling Theory and Applications, 2011.

• E. Vural and P. Frossard, "Approximation of Pattern Transformation Manifolds with Parametric Dictionaries", IEEE International Conference on Acoustics, Speech and Signal Processing, 2011.

• S. Ö. Arik, E. Vural and P. Frossard, "Alignment of Uncalibrated Images for Multi-View Classification", IEEE International Conference on Image Processing, 2011.

• E. Vural and P. Frossard, "Curvature Analysis of Pattern Transformation Manifolds", IEEE International Conference on Image Processing, 2010.

• E. Vural and P. Frossard, "Distance-Based Discretization of Parametric Signal Manifolds", IEEE International Conference on Acoustics, Speech and Signal Processing, 2010.

• E. Vural and A. A. Alatan, "Outlier Removal for Sparse 3-D Reconstruction from Video", 3DTV-Con, 2008.