



PhD-FSTC-2017-67  
The Faculty of Sciences, Technology and Communication

## DISSERTATION

Presented on 20/10/2017 in Luxembourg

to obtain the degree of

DOCTEUR DE L'UNIVERSITÉ DU LUXEMBOURG

EN INFORMATIQUE

by

**Girum DEMISSE**

Born on 16 November 1987 in Addis Ababa (Ethiopia)

## DEFORMATION BASED CURVED SHAPE REPRESENTATION

### Dissertation defense committee

Dr. Mehrdad Sabetzadeh, Chairman  
*Senior research scientist, University of Luxembourg*

Dr. David Fofi, Vice-chairman  
*Professor, University of Burgundy*

Dr. Bjorn Ottersten, dissertation supervisor  
*Professor, University of Luxembourg*

Dr. Djamila Aouada  
*Research scientist, University of Luxembourg*

Dr. Stefano Berretti  
*Professor, University of Florence*



## Abstract

Representation and modelling of an objects' shape is critical in object recognition, synthesis, tracking and many other applications in computer vision. As a result, there is a wide range of approaches in formulating representation space and quantifying the notion of similarity between shapes. A similarity metric between shapes is a basic building block in modelling shape categories, optimizing shape valued functionals, and designing a classifier. Consequently, any subsequent shape based computation is fundamentally dependent on the computational efficiency, robustness, and invariance to shape preserving transformations of the defined similarity metric.

In this thesis, we propose a novel finite dimensional shape representation framework that leads to a computationally efficient, closed form solution, and noise tolerant similarity distance function. Several important characteristics of the proposed curved shape representation approach are discussed in relation to earlier works. Subsequently, two different solutions are proposed for optimal parameter estimation of curved shapes. Hence, providing two possible solutions for the point correspondence estimation problem between two curved shapes. Later in the thesis, we show that several statistical models can readily be adapted to the proposed shape representation framework for object category modelling. The thesis finalizes by exploring potential applications of the proposed curved shape representation in 3D facial surface and facial expression representation and modelling.

## Acknowledgements

I would first like to thank my advisor Professor Björn Ottersten for his insightful comments and general advice throughout the last three and half years. Our conversations were always instructive and pleasant, thank you. I would also like to thank my co-supervisor Dr. Djamila Aouada for her continuous support in both technical and personal matters. I am thankful for her openness to ideas and patience. I would like to thank my defence committee for agreeing to review this work.

I thank the whole SIGCOM group members for fostering a friendly and collaborative work environment, especially members of the computer vision subgroup. I would also like to thank Dr. Michel Antunes for his encouragement and interesting discussions. My long time office mates, Dr. Anestis Tsakmalis and Dr. Jabier Martinez, thank you for making our office a pleasant place to work.

I would like to thank my family, my mother, Azeb Kidane, for her prayers and support. My father, Lt.col Getachew Demisse, who taught me about science, literature and whatever is good in me. Thank you for your support and exciting stories. My sisters, Fiker Getachew and Bethlehem Tibebu, thank you for your support. My uncles for their never ending support and always inspiring conversations. Finally, my wife and friend Rahel Mekonnen, thank you for your support and love. Thank you for sharing your life with me.

Girum G. Demisse

Luxembourg, October 2017.

# Contents

<b>Notation</b>	<b>vii</b>
<b>List of Figures</b>	<b>ix</b>
<b>List of Tables</b>	<b>xv</b>
<b>List of Algorithms</b>	<b>xvii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 Challenges . . . . .	4
1.3 Contributions . . . . .	6
1.4 Outline . . . . .	11
<b>2 Curved shape representation</b>	<b>13</b>
2.1 Landmark based representation . . . . .	14
2.2 Immersion based representation . . . . .	17
2.3 Deformable-template based representation . . . . .	22
2.3.1 Active shape representation . . . . .	23
2.3.2 Nonlinear deformable-template representations . . . . .	24
2.4 Summary . . . . .	25
<b>3 Deformation based curved shape representation</b>	<b>27</b>
3.1 Motivation . . . . .	27
3.2 Deformation based representation . . . . .	28
3.2.1 Data structure . . . . .	28

## CONTENTS

---

3.2.2	Representation . . . . .	31
3.3	Metric and geodesic curves . . . . .	37
3.4	Characteristics and interpretation . . . . .	42
3.4.1	Properties . . . . .	42
3.4.2	Interpretation . . . . .	45
3.5	Evaluation . . . . .	46
3.5.1	Shape retrieval . . . . .	47
3.5.2	Robustness . . . . .	49
3.5.3	Computational cost . . . . .	51
3.6	Conclusion . . . . .	52
<b>4</b>	<b>Optimal parametrization</b>	<b>53</b>
4.1	Motivation . . . . .	53
4.2	Problem definition . . . . .	54
4.3	Special case: linear reparametrizations . . . . .	55
4.4	General case . . . . .	57
4.4.1	Optimal sampling . . . . .	58
4.4.2	Dynamic programming . . . . .	63
4.5	Evaluation . . . . .	65
4.5.1	Shape retrieval . . . . .	65
4.5.2	Robustness . . . . .	68
4.6	Conclusion . . . . .	70
<b>5</b>	<b>Statistical curved shape analysis</b>	<b>75</b>
5.1	Motivation . . . . .	75
5.2	Background . . . . .	75
5.3	Statistics of shapes . . . . .	77
5.4	Modelling shape distribution . . . . .	81
5.4.1	Parametric density estimation . . . . .	82
5.5	Latent variable estimation . . . . .	85
5.5.1	Dimension reduction . . . . .	85
5.5.2	Clustering . . . . .	87
5.5.3	Mixture of Gaussians . . . . .	89
5.6	Evaluation . . . . .	94

5.7 Conclusion . . . . .	97
<b>6 3D facial expression analysis</b>	<b>99</b>
6.1 Motivation . . . . .	99
6.2 Related works . . . . .	100
6.3 Surface representation . . . . .	102
6.3.1 Face representation . . . . .	104
6.3.2 Expression representation . . . . .	105
6.4 Curve correspondence . . . . .	106
6.4.1 Cost of mismatching curves . . . . .	106
6.4.2 Dynamic programming based solution . . . . .	109
6.5 Modelling expressions . . . . .	112
6.6 Experiments . . . . .	115
6.6.1 Experimental setup . . . . .	116
6.6.2 Linear versus the proposed expression space . . . . .	117
6.6.3 Results . . . . .	118
6.7 Conclusion . . . . .	119
<b>7 Conclusion and outlook</b>	<b>123</b>
7.1 Conclusion . . . . .	123
7.2 Outlook . . . . .	124
<b>A Exponential maps</b>	<b>129</b>
<b>B Computing optimal rigid transformation</b>	<b>131</b>
<b>Bibliography</b>	<b>133</b>

## CONTENTS

---



# Notation

$\mathbf{R}^n$	n-dimensional real space.
$\mathbf{C}^n$	n-dimensional complex space.
$\mathbf{CP}^n$	n-dimensional complex projective space.
$\mathbf{S}^n$	n-dimensional sphere.
$\mathbf{V}^n$	n-dimensional subspace.
$\text{Imm}(\mathbf{S}^n, \mathbf{R}^n)$	the set of immersions from $\mathbf{S}^n$ to $\mathbf{R}^n$ .
$\text{Diff}(\mathbf{S}^n)$	group of diffeomorphism from $\mathbf{S}^n$ to $\mathbf{S}^n$ .
$C^n$	n-differentiable function.
$I_n$	n-dimensional identity matrix.
$e$	group identity.
$\langle \cdot   \cdot \rangle$	inner product.
$\  \cdot \ _2$	$L_2$ norm.
$\rtimes$	semi-direct product
$\  \cdot \ _F$	Frobenius norm.
$n!$	n factorial.
$\text{vec}(\cdot)$	an operator that vectorizes a matrix.
$\min f(\cdot)$	minimum value of the function $f(\cdot)$ .
$\arg \min f(\cdot)$	minimizer of the function $f(\cdot)$ .
$\max f(\cdot)$	maximum value of the function $f(\cdot)$ .
$\arg \max f(\cdot)$	maximizer of the function $f(\cdot)$ .
$\text{SO}(n)$	n dimensional special orthogonal group.
$\text{SE}(n)$	n dimensional special Euclidean group.
$\text{mod}(x)$	modulo operation with modulus x.
$c$	continuous curve.

## CONTENTS

---

$\dot{c}, \ddot{c}$	differentials of the $c$ . The dots correspond with order of the differential.
$\circ$	function composition.
$\tilde{c}$	approximation of a continuous curve by discrete points.
$\oint$	closed line integral
$X$	random variable.
$\mathbf{X}$	random vector.

# List of Figures

1.1	Illustration of curved shapes in object representation: (a) examples of plane images from Caltech 101 data set [FFFP07] and their corresponding silhouettes. (b) 3D facial surfaces decomposed into a set of curved shapes. . . . .	2
2.1	Kendall’s shape space . . . . .	16
2.2	Variation due to reparametrization: (a) and (c) show the parameter space and its reparametrization along with the red dot to denote the starting point. (b) and (d) show the curve trace defined on (a) and (c), respectively. . . . .	19
3.1	Illustration of an arc length based curve sampling. The blue curve shows the continuous curve, and the points connected by the red lines are sampled based on arc length. . . . .	29
3.2	Illustration of the proposed representation. Given a fixed starting point $p^s$ , the curve is reconstructed by the successive action of the $(g_1, \dots, g_{z-1})$ . . . . .	33
3.3	Shapes along the geodesic path between the initial shape (first column) and target shape (last column). The odd rows show results from our approach while the even rows are results from infinite dimensional based approach called SRV function [SKJJ11]. All shapes are represented by 100 uniformly sampled and normalized points. We note that results from [SKJJ11] are smoothed and loses local features of the shapes. . . . .	40

## LIST OF FIGURES

---

3.4	Deformation transport between curves with known parametrization. The first set of shapes shows two examples where $\tilde{c}_1$ deforms to $\tilde{c}_{1*}$ . The second set shows the transported deformation to their similar objects $\tilde{c}_2$ to give $\tilde{c}_{2*}$ , respectively. . . . .	43
3.5	Two pairs of curves. Under absolute deformation (AD), the curves in (b) are more similar than the curves in (a), while under relative deformation (RD) the curves in (a) are more similar than the curves in (b). . . . .	44
3.6	Different samplings of a given continuous curve $c$ . The red dots denote sampled points. The last sampling $\xi^*$ is a uniform sampling. Note that $\xi_1$ and $\xi_2$ do not preserve shape. . . . .	45
3.7	Example leaves from different categories of the Flavia leaf dataset [WBX <sup>+</sup> 07].	47
3.8	Examples from different categories of the Swedish leaf dataset [Söd01].	49
3.9	Precision-Recall curves on the Flavia and the Swedish leaf dataset.	49
3.10	The first row shows the 7 types of fighter jets from [TGJ07]. The second row shows examples from the corrupted shapes with $\sigma = 2.5$ .	50
3.11	Precision-Recall curves on the fighter jets dataset for different noise magnitudes quantified by the standard deviation ( $\sigma$ ). . . . .	51
4.1	Point correspondence estimation based on linear parametrization assumption: The first and second columns show the curves to be matched, the last column shows the matching result. Note that in the case of the first row, the estimated point correspondence are reasonably accurate than the second row where a nonlinear local deformation is observed. . . . .	56
4.2	Geodesic paths for different values of $\alpha$ and $\lambda$ . Note the impact of a large $\lambda$ value on the deformation (geodesic) and the final result.	59

4.3	Intermediate values of dynamic programming. (a) shows two input shapes. $\tilde{c}_1^*$ , coloured in green, is uniformly sampled and $c_2$ , coloured in red, is to be sampled optimally. (b) shows the search space, defined by the charts $U_i$ , and the cost of selecting a point from $c_2$ for the $i^{\text{th}}$ position with the color coding. (c-e) shows three optimal sampling paths based on (4.16) for different values of $\alpha$ and $\lambda$ . The green path is an optimal for $\alpha = 1$ and $\lambda = 0$ , the red path is an optimal for $\alpha = 0$ and $\lambda = 1$ , and finally the yellow path is an optimal sampling for $\alpha$ and $\lambda$ values shown below the figures. . . .	60
4.4	Geodesic paths between two shapes under different elasticity constraint $\eta$ . Note that an objective functional with an appropriate $\lambda$ value gives a consistent result regardless of the elasticity constraint.	61
4.5	Illustration of the symmetric and the asymmetric solutions. Each row shows the solution sampling along with the problem space and the solution paths. The first row shows the solution due to (4.16) where the blue curve is uniformly sampled with $z = 100$ . The second row shows the optimal sampling of both curves with (4.14) where $z = 50$ . . . . .	66
4.6	Linear vs area preserving sampling. For the area preserving case, the green curve is sampled uniformly, and the curve in red is sampled with area preserving parametrization. Note that the area preserving sampling adjusts the sampling rate while the linear parameter sampling does not. . . . .	70
4.7	Impact of sampling size under the optimal and the uniform sampling based point correspondence estimation. . . . .	70
4.8	A geodesic path from the first to the last curve. These curves are used in estimating the effects of sample size. . . . .	71
4.9	Impact of $\lambda$ on partial shape matching. In all of the above examples, $\eta = 37.46$ . Each row shows optimal sampling of a shape, shown in red, to match the uniformly sampled shape shown in green. Note that, a large value of $\lambda$ leads to a matching that favours area preservation. . . . .	72

## LIST OF FIGURES

---

4.10	Illustration of deformation transporting in linear vs area preserving parameter estimation. . . . .	73
5.1	Mean computation of shapes visualized using Multidimensional scaling [BG05]. The shapes depicted in red are the computed mean of each dataset. (a) and (c) are mean shapes when the point correspondence estimation is based on linear parameter estimation. (b) and (d) are the mean shapes computed from shapes dataset with area preserving parameter estimation, respectively. . . . .	80
5.2	Illustration of the tangent space at the mean shape, where the covariance matrix of a linearized data points are computed. . . . .	81
5.3	Example dataset of dogs from Kimia 1070-shape dataset. . . . .	83
5.4	Mean shapes of the dogs dataset along with a randomly sampled shapes. (a) Mean shape based on area preserving parameter estimation, (b) randomly sampled shapes from a Gaussian distribution of shapes with area preserving parameter estimation. (c) Mean shape based on linear parameter estimation, (d) randomly sampled shapes from a Gaussian distribution of shapes with linear parameter estimation. . . . .	83
5.5	Shapes in two principal directions on the dogs dataset. The first column shows the mean shape, computed from the dataset with parameter estimation based on linearity assumption and area preserving assumption, respectively. The last three columns show the shapes in the first two principal directions with scaling factor given by $\sigma$ when the dataset is aligned with linear and area preserving parameter estimations, respectively. . . . .	86
5.6	Examples of clustering results for $k=2$ . Two different clustering results for different point correspondence estimations. (a) Point correspondence estimation based on linear parametrization. (b) Point correspondence estimation based on area preserving parametrization. . . . .	88

5.7	Mixture of Gaussian result on the dogs dataset with ( $k = 3$ ), for visualization the dataset is plotted with coordinates estimated with Multidimensional scaling. The diamond shapes show the position of the means. The color of each data point is assigned to the Gaussian that maximizes its posterior (5.32). (d) Shows the Log-likelihood after each iteration. . . . .	92
5.8	Mixture of Gaussians ( $k = 2$ ) result on a car dataset [LS03] where shape variation is only due to change of viewpoint. . . . .	93
5.9	Examples from ETH80 dataset [LS03]. . . . .	94
6.1	Proposed 3D facial expression representation. A smile is captured by the deformation $\mathcal{D}_E$ and applied to two different subjects, preserving their specific shapes. . . . .	101
6.2	Facial curve extraction. . . . .	102
6.3	Commutative diagram. Deformation between two curves is represented in terms of the deformation between previously selected curves. This formulation is used to estimate curve correspondence via dynamic programming. . . . .	106
6.4	Curve matching between two different faces with “surprise”, on the far left, and “happy”, on the far right, expressions. The first row shows the geodesic deformation between the faces when both faces are represented by 50 uniformly sampled curves. The red curves are tracked along the deformation to illustrate the mismatch. In the second row, the “surprise” face is represented by 50 uniformly sampled curves while the “happy” faces is optimally sampled via dynamic programming, see Section 6.4. The cost matrix shows three solutions for different weighting factors—the blue region is the feasible set defined by the sectors size. The red curve is the optimal solution for $\alpha = 0$ and $\lambda = 1$ , the green path is optimal for $\alpha = 1$ and $\lambda = 0$ , finally the yellow path is optimal for $\alpha = 1$ and $\lambda = 1$ , the second deformation is based on the yellow path. . .	107

## LIST OF FIGURES

---

6.5	Curve matching. In (a) a cost matrix with three solutions is shown that correspond to the three deformations in (b). In all the deformations, the first faces are approximated by 50 uniformly sampled curves, i.e., they are $\Gamma$ , while the last faces are sampled optimally, i.e., they are $\Gamma^*$ . Pair of facial curves are highlighted in red in all of the deformations to illustrate the impact of the matching. The deformation in the first row is according to the red path in the cost matrix which is the solution when $\alpha = 0$ and $\lambda = 1$ , that is when both faces are sampled uniformly. The deformation in the second row is based on the green path which is the solution when $\alpha = 1$ and $\lambda = 0$ , that is when there is no volume based constraint. The last deformation is based on the yellow path which is computed for $\alpha = 1$ and $\lambda = 4$ . In this particular example, regardless of the good curve matching solution given by the yellow path, the deformation of the lower lip is not smooth. <i>This is mainly because we are only matching curves and disregarding point matching.</i> . . .	113
6.6	Curve matching for a relatively small deformation under three different $\lambda$ values. Note that, the impact of $\lambda$ is minimal on the matching results since the deformation is relatively small. . . . .	114
6.7	Curve matching for a relatively large deformation under three different $\lambda$ values. Contrary to Figure 6.6, the value of $\lambda$ has a significant impact on the matching result since the deformation is relatively large. . . . .	114
6.8	Given a vector form of an expression $\mathcal{L}(\mathcal{D}_E)$ , the figure shows the action of $\mathcal{E}(S \times \mathcal{L}(\mathcal{D}_E))$ on two different neutral faces for different scales $S$ . In (a) the scale $S = 0$ , thus the face remains neutral. . .	115
6.9	Two-dimensional expression space computed from three expressions with PCA. (a) PCA on expressions extracted as expression residues. (b) PCA on the Lie algebra of expressions extracted with our approach. . . . .	117



# List of Tables

3.1	Mean average precision (MAP) on the Flavia leaf dataset. Our result is highlighted at the bottom. . . . .	48
3.2	Nearest-neighbour recognition rate on the Swedish leaf dataset. Our result is highlighted at the bottom. . . . .	50
3.3	MAP on the fighter jets dataset with a Gaussian noise of different standard deviations. . . . .	50
4.1	Retrieval results on Kimia99 shape dataset. . . . .	68
4.2	Retrieval results on Kimia216 shape dataset. . . . .	69
5.1	Average accuracy comparison between models based on the proposed approach and earlier shape feature based classifiers on ETH80 dataset. The best performing algorithm is highlighted with bold font. . . . .	95
6.1	Comparison between performances of the proposed approach and recent results. We highlight the top score at the bottom. . . . .	119
6.2	Confusion matrix on expressions extracted from faces with uniformly sampled 50 curves (D-1-U). . . . .	120
6.3	Confusion matrix on expressions extracted from faces with optimally sampled 50 curves (D-1-O). . . . .	120
6.4	Confusion matrix on expressions extracted from faces with uniformly sampled 100 curves (D-2-U). . . . .	121
6.5	Confusion matrix on expressions extracted from faces with optimally sampled 100 curves (D-2-O). . . . .	121

## LIST OF TABLES

---

# List of Algorithms

1	Fixing one curve and optimally sampling another . . . . .	64
2	Estimation of the Karcher mean in $SO(n)$ . . . . .	79
3	Optimal sampling of curves from $\Gamma^2$ . . . . .	111

## LIST OF ALGORITHMS

---

# Chapter 1

## Introduction

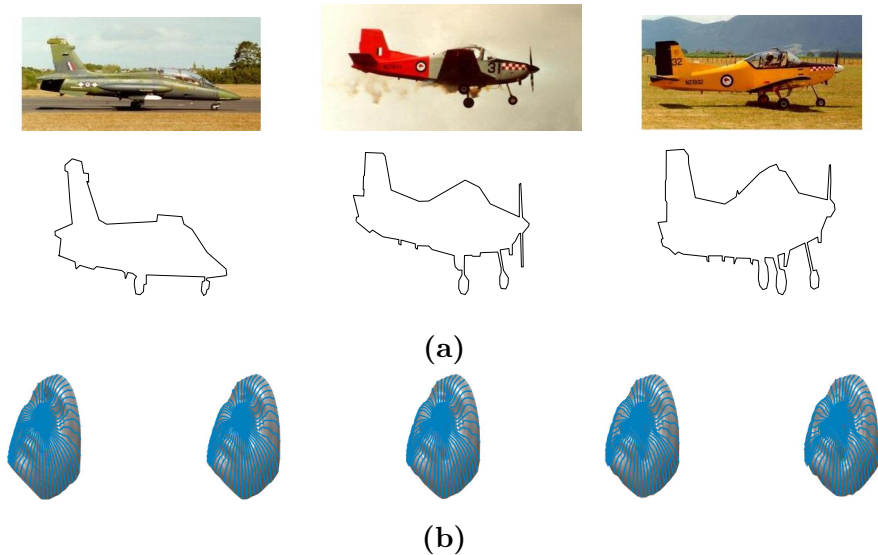
### 1.1 Motivation

The study of shapes has a long mathematical history in differential geometry and topology. The shapes studied in these fields, primarily in differential topology, are those that can locally be approximated by smooth functions. In that sense, the main goal of differential topology is to study properties of such shapes that are invariant under diffeomorphism [Mil97].

In this thesis, however, the shapes that we are concerned with are curves embedded in  $\mathbf{R}^n$  and semantically associated with natural or man-made objects. For instance, objects from an image can be represented by their *outlines* or *silhouettes* which are closed curves in  $\mathbf{R}^2$ . Meanwhile, three dimensional surfaces can be decomposed into a set of curves in  $\mathbf{R}^3$  for further characterization, see Figure 1.1. Consequently, the properties we want to study are the ones that convey similarity or dissimilarity in a way that agrees with an observed dataset. Undeniably, such kind of properties exist in shape categories— mammals, humans in particular, can recognize an object from its shape with remarkable ease, even under occlusion or viewpoint variation. Nevertheless, mathematically formalizing the concept of shape similarity or dissimilarity, in the way a human’s perceptual system does, has proven to be a challenging task. To highlight the contrast, consider the silhouette of a horse and the silhouette of an apple. For a human being, these silhouettes are completely different from each other. From a differential

## 1. INTRODUCTION

---



**Figure 1.1:** Illustration of curved shapes in object representation: (a) examples of plane images from Caltech 101 data set [FFFP07] and their corresponding silhouettes. (b) 3D facial surfaces decomposed into a set of curved shapes.

topology point of view, however, shapes that can smoothly deform from to another are the same; hence a horse silhouette is the same as an apple's silhouette. As a result, a smooth deformation of one object's silhouette to another is not a strong enough criterion to characterize a given object category. Meanwhile, shapes from the same object category can exhibit large variations which can also be associated with smooth deformations. For instance, the shape of an object's silhouette might deform in some local region depending on our point of view or how the object is posed. Hence, a given smooth shape deformation can characterize both inner-category and inter-category shape variations. Consequently, the main task of a shape based object recognition approach is to be able to distinguish between inner-category and inter-category shape deformations. To that end, several and interrelated problems have to be addressed. First, a data structure has to be formulated to describe an observed object's shape— in practical applications, the observed object's shape is no more than an approximation of its trace. Subsequently, one has to prescribe a mathematical representation that can be associated with the data structure. The mathematical representation has to be, in some sense, invariant to inner-category shape variations or object identity

preserving transformations like occlusion. Additionally, the notion of similarity and dissimilarity between shapes has to be quantified by a distance function that is defined on the mathematical representation space. In most cases, the distance function is further used to approximate the distribution of a shape category with statistical models.

Considering the difficulty of the task, one might naturally ask: does a shape contain enough information to be used as an input for object recognition and analysis? We believe yes. Shape is one of the most informative features of an object. Although highly important, colour and texture are susceptible to change in time and vary regardless of the object’s identity. For instance, an orange fruit is green when unripe and orange when ripe, while its shape is more or less stable with respect to seasonal changes. Moreover, colour is sensitive to contrast and lighting conditions. Alternatively, some objects can not adequately be described either by texture or colour. This is further strengthened by the fact that, regardless of specialization, a general perceptual system in mammals has a dedicated neural pathway for shape recognition [SW12]. Lastly, an argument can be made in support of shapes from an application point of view; that is, for problem domains where colour is not available, the most prominent feature of an object is its shape. In medical imaging, shape analysis is one of the main tools used for diseases diagnostics. In [WBR<sup>+</sup>07], deformation-based mathematical shape representation, together with a statistical model called Principal Component Analysis (PCA), is used to study the progression of dementia caused by Alzheimer’s disease. In [SLM<sup>+</sup>16], a shape similarity function, defined on a mathematical representation called Square Root Velocity (SRV), is used to analyse protein structures as curves in  $\mathbf{R}^3$ —several more applications, inspired by biological problems, are discussed in [DM98, Pen09]. Furthermore, SRV based shape representation approaches, together with statistical models, have been used in leaf shape recognition [LKSM14] and face recognition from 3D point cloud [DADS10]. In [AK10], curves are used to study generic 3D surfaces. Meanwhile, a wide range of shape representation approaches, general known as “shape descriptors”, are used for object classification in computer vision [WFB<sup>+</sup>14, WBY<sup>+</sup>12, HJZG12].

In this thesis, we propose a novel deformation-based curved shape representation approach. The proposed approach leads to a similarity distance equation

## 1. INTRODUCTION

---

that has a closed-form solution unlike most infinite dimensional shape representation approaches [MM03, BBM16]. We further formulate a symmetric objective functional to estimate the point correspondence between shapes and ensure a one-to-one solution. On the contrary, in landmark-based approaches [Ken84], point correspondence between shapes is established by manual annotation. Meanwhile, infinite dimensional representations use a fixed template shape to estimate the point correspondence between shapes which leads to asymmetric objective functional and a potentially one-to-many solution [SKJJ11, BBM16]—the effectiveness of the proposed representation and the point correspondence estimation are evaluated in several shape retrieval problems. Subsequently, we show the adaptation of different statistical models to the proposed curved shape representation and evaluate their performance in shape category modelling problems. Finally, we discuss the application of the proposed curved shape representation approach in face and facial expression analysis from a 3D point cloud data.

### 1.2 Challenges

In [Mum91, Mum87], the main challenges involved in formalizing a curved shape representation are broadly categorized into two: *Data structure* and *Representation*. In this subsection, we will describe a general overview of the challenges associated with shape representation from these two points of view.

1. **Data structure:** Contours of objects, either natural or man-made, are infinite dimensional, i.e., a contour has infinitely many degrees of freedom to vary. On the other hand, existing computer systems have limited memory and computing power. Consequently, negotiating computational limitation with representational limitations has been one of the main challenges of shape representation approaches. Essentially, there are two main schools of approach to this problem. 1) *finite dimensional*: methods based on landmark points [Ken84] or shape descriptors [YKR+08]. *Infinite dimensional*: methods based on piecewise differentiable functions (splines) [SKJJ11]. Approaches based on descriptors or a set of descriptors attempt to summarize the shape with feature points extracted from the shape of the object. One



of the main advantages of feature-based representations is that the features usually define a linear and low dimensional feature space on which a distance metric can be defined and statistical models can be built. However, the feature maps are not necessarily invertible and thus it is difficult to recover computational results in the original data space. Moreover, in [SM06] it is argued that a feature vector can not uniquely represent a shape, since the shapes themselves are infinite dimensional. There will always be many shapes exhibiting a property summarized by a specific feature. Meanwhile, in the infinite dimensional setting the representational weakness of the descriptor-based approaches is tackled by approximating an observed shape by an analytical function or by studying the deformation of the shapes from one to another [You99]. In such cases, the space of shapes is usually defined as a nonlinear submanifold of a Banach space. Contrary to the descriptor-based approaches, the difficulty in this case is not the power of the representation but the computationally demanding distance equations, and in some cases intractable equations, that are associated with the infinite dimensional space. Consequently, several and different types of metrics are proposed and studied in the infinite dimensional setting [MM03].

2. **Representation:** Once an appropriate data structure is chosen, the next challenge, and perhaps the most important, is designing/selecting a mathematical structure that can be associated with observed shapes, so that statistical and geometric calculations can be done. To further clarify this challenge, we will adapt Kendall's [Ken84] definition of shape– we will refer to this definition throughout this thesis– which reads as follows.

**Definition 1.** *Shape is all the geometrical information that remains when rotation, translation and scale are removed.*

Although, one can add several additional requirements, according to Definition 1, the most basic requirement for a shape representation is to be invariant to similarity (symmetry) transformations. That is, either a pre processing stage has to be developed to clear variations due to scale, translation and rotation or the constructed representation has to be inherently

## 1. INTRODUCTION

---

invariant to such transformations. Nevertheless, Kendall's definition is restricted only to global linear transformation of shape variations. However, the notion of similarity between shapes is not only captured by global transformations but through local variations as well. For instance, the shape of an apple might exhibit nonlinear variations in some local regions due to occlusion or a bite. Consequently, the family of symmetry transformations has to be expanded to include high dimensional transformations that capture shape variations in local regions or has to include global nonlinear transformations. In the shape representation literature, such kinds of transformations are generally known as reparametrizations of the shape's parameter space or diffeomorphisms. The difficulty, however, is in practical problems neither the parametrization of a shape nor its reparametrization is known. As a result, optimal parametrizations of shapes have to be estimated with respect to one another, defining what is known as the point correspondence estimation problem.

Moreover, in addition to invariance to symmetry transformations the representation has to lead to a mathematical shape space that can at least be linearised locally. This is mainly because in modelling a category of shapes similarity metric alone is not sufficient to explain the data generating event, hence statistical models are employed in a locally linearised region of the shape space to model shape distributions.

The above two main categories of problems are often interrelated, hence different approaches in the data structuring will influence both the problem formulation and solutions in the representation construction and the computation of statistical quantities.

### 1.3 Contributions

The main contribution of this thesis is in the formulation of a computationally efficient and descriptive curved shape representation that is based on a finite dimensional approximation of curves. The thesis proposes to approximate a given continuous curved shape by a set of sampled points with respect to the curve's

arc length. The approximation of the curves by a discrete set of points is similar to approximating a curve by piecewise linear function. For most practical applications, a piecewise linear approximation is sufficient to describe a curve, while the fitness of the approximation can be tailored to specific applications by altering the number of discrete points. In a sense, a finite dimensional data structure might be considered similar to landmark or descriptor based data structure. However, unlike descriptors we do not associate any special meaning to the points except that they approximate the curve. Furthermore, we define the approximation in terms of optimal parameter estimation instead of relying on annotation or feature extraction like landmark based approaches. Nevertheless, we benefit from the finite dimensional data structure by avoiding some of the computational difficulties of the infinite dimensional approach, while formalizing and providing approximate solutions to the challenges posed in the infinite dimensional setting. Subsequently, interrelated folds of contributions are made in the formulation of the proposed curved shape representation, and its applications in shape based object retrieval, object category classification, and facial expression analysis. The following list summarizes the contributions of the thesis.

1. **Representation:** The thesis proposes a novel Lie group based representation that equates a given set of oriented points with a sequence of transformation matrices, such that their sequential action on a given fixed starting point reconstructs the set. We show that this representation defines a distance function that has several interesting properties. Assuming the knowledge of curve parametrizations, the main properties of the representation and the distance it defines are listed below:
  - Measuring the distance between two curves: In the proposed representation, measuring the deformation cost from one to another is equivalent to measuring the distance between the curves. Hence, the formalized distance function is similar to what is known as *effort functional* in infinite dimensional setting [You98].
  - Deformation between curves is defined as the difference between relative transformations of the curves. Consequently, unlike other metrics

## 1. INTRODUCTION

---

in shape representation, it is not the absolute displacement of each point that is measured but the relative displacement of each point.

- In the proposed framework, the distance and optimal deformation between two curves are computed in a closed form; in the infinite dimensional setting most metric definitions do not lead to a closed form distance measure.
- The defined distance function ultimately depends on the arc length approximation of the curves, hence is more tolerant to local noise and shape perturbation as opposed to metrics defined in the infinite dimensional setting, which usually are based on second or higher order derivatives of the curves.

*This work is discussed in detail in Chapter 3. The discussion is based on the following publications:*

- Demisse. GG, Aouada. D, Ottersten. B. “Deformation based curved shape representation”, in *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2017.
- Demisse. G, Aouada. D, Ottersten. B. “Similarity Metric For Curved Shapes In Euclidean Space”, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

2. **Optimal parametrization:** The landmark based approach usually depends on a manual annotation of the landmark points. Consequently, two potentially different sets of landmarks can be defined for the same observed curved shape. Moreover, the problem of optimal point correspondence is not defined, since the matching landmarks are assumed to be given. In this thesis, contrary to landmark-based approaches, the points approximating a curve are selected based on arc length. Consequently, the sampling can be made invariant to the curve parametrization if the parametrization is known a priori. In most computer vision applications, however, the parametrization of curves is not known a priori; what is observed is only the approximation of the curve’s trace. As a result, we formulate optimal

parameter estimation under a linearity assumption and later extend it to a general case. Optimal parameter estimation, in both cases, can be computed in polynomial time and is consistent with the notion of deformation minimizing distance function. We note that in most infinite dimensional settings, an optimal parametrization of curves is estimated for a given curve with respect to a selected template curve. As discussed in [BBM16], such approach can lead to a one-to-many point correspondence solution. In our case, we solve the optimal parameter estimation for two different cases: 1) optimal parameter estimation of a shape argument with respect to the parametrization of a template shape, 2) optimal parameter estimation for both curve arguments. In both cases, we ensure a one-to-one point correspondence by restricting the family of parametrizations to injective and monotonic functions with strictly positive derivatives with respect to the curve’s arc length.

*This work is discussed in detail in Chapter 4. The discussion is based on the following publication:*

- Demisse. GG, Aouada. D, Ottersten. B. “Deformation based curved shape representation”, in *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2017.

3. **Statistical models:** We discuss established statistical models and latent variable estimation approaches for the modelling and simulation of shape categories and shape deformations. Since the proposed representation can locally be linearised, the adaptation of statistical models follows naturally. We show the applicability of the statistical models in shape based object category recognition. Results show that all of the statistical models performed as well or better than several “shape descriptor” based object category modelling approaches.

*This work is discussed in detail in Chapter 5. The discussion is partly based on the following publication:*

## 1. INTRODUCTION

---

- Demisse. G, Aouada. D, Ottersten. B. “Template-based Statistical Shape Modelling On Deformation Space”.in *IEEE International Conference on Image Processing (ICIP)*, 2015.

4. **Application:** Finally, we employ the proposed curved shape representation for 3D facial surface and expression representation. We extend the optimal parameter estimation in curves to optimal parameter estimation in facial surfaces. We evaluate the proposed approach on expression recognition problem from a statistic 3D point cloud dataset. The approach performed comparably with feature based 3D facial expression classification techniques, when the curves are not optimally sampled. In case of the optimal curve sampling, the approach outperformed the feature based methods.

*This work is discussed in detail in Chapter 6. The discussion is based on a work that is under major revision:*

- Demisse. GG, Aouada. D, Ottersten. B. *Deformation Based Facial Expression Representation, ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, under review.

Other publications not included in this thesis:

- Oyedotun. O, Demisse. G, Shabayek. A, Aouada. D, Ottersten. B. “Facial Expression Recognition via Joint Deep Learning of RGB-Depth Map Latent Representations”. in *IEEE International Conference on Computer Vision Workshop (ICCVW)*, 2017.
- Antunes. M, Baptista. R, Demisse. G, Aouada. D, and Ottersten. B. “Visual and human-interpretable feedback for assisting physical activity”. in *European Conference on Computer Vision (ECCV) Workshop on Assistive Computer Vision and Robotics*, 2016.

## 1.4 Outline

The thesis is organized as follows:

Chapter 2 describes earlier works in shape representation and metric definition. The discussion is mainly from the perspective of the shape spaces and the metric that is defined on them.

In Chapter 3, we describe the proposed shape representation approach and the distance metric defined on it. We will further discuss important properties and relations of the proposed approach with what is known as *effort functional*. The Chapter concludes with experimental evaluations of the proposed representation and its metric in shape retrieval problems.

Chapter 4 builds on the proposed curved shape representation by extending it to curves with unknown parametrization. Optimal curve parameter estimation is solved for a special case, when reparametrization is assumed to be linear, and for the general case, for reparametrizations that preserve a given geometric property. Moreover, Chapter 4 discusses a dynamic programming-based solution and introduces free parameters into the optimization problem.

In Chapter 5, we describe how to compute statistics of a shape dataset represented by the proposed curve representation. Later on, we present different parametric distribution models for estimating the distribution of shapes represented by the proposed approach. Chapter 5 concludes by evaluating the statistical models in shape based object category recognition.

In Chapter 6 we discuss an application of the proposed curve representation in representing and analyzing face and facial expressions from 3D data. Subsequently, a restricted version of optimal parameter estimation, in 3D face representation, and modelling facial expressions are discussed. The Chapter concludes with an evaluation of the proposed approach for facial expression recognition.

The thesis ends with concluding remarks in Chapter 7. Furthermore, potential extension directions of the proposed curve representation approach are discussed.

## 1. INTRODUCTION

---



## Chapter 2

# Curved shape representation

In this chapter, we discuss different curved shape representation approaches that are closely related to what is proposed in this thesis.

In general, a curved shape representation can be understood as a function that identifies a given shape with a mathematical quantity. In most cases, these mathematical quantities aim to satisfy a general and/or application specific properties. There are several works that classify shape representation approaches using different criteria (processing approach, problem domain of the representation, co-domain of the representation, etc), see [YKR<sup>+</sup>08, You12]. However, in this chapter and throughout the thesis we discuss shape representations from two fundamental points of view:

1. *Completeness.* A representation is called complete if it is one-to-one, that is two distinct shapes can never have the same representation. Completeness is fundamental to address *invariance* to symmetry transformations—transformations that preserve the nature of the shape—and *robustness*—tolerance to small transformations or perturbations. Completeness is also closely related with the data structuring of the curves.
2. *Shape space and metrics.* In broad terms, a shape space is the co-domain of a shape representation with a distance *metric*. In most cases, it is further endowed with additional mathematical structures, e.g., differential and topological structures. Constructing a shape space and defining a metric is

## 2. CURVED SHAPE REPRESENTATION

---

not always straightforward for all representations. Among the basic structures of a shape space, the most important ones are differential structures and smoothly varying *metrics*. These two are fundamental to compute geometric notions like distance, differential notions like derivatives, and to tackle optimization problems in the constructed shape space [Men13]. Consequently, statistical and machine learning algorithms can be designed to explain a dataset of observations in the defined shape space. However, the complexity and effectiveness of the algorithms will ultimately depend on the shape representation and the metric defined on it. As discussed in Section 2.2, in most infinite dimensional formulations the defined distance metric is computed iteratively.

Consequently, Chapter 2 will only cover shape representations that are studied as shape spaces and discusses the *completeness* of the spaces and the *metrics* defined on them. Nevertheless, there is a whole category of interesting shape representations known as “shape descriptors” that are studied mainly from the *completeness* point of view. This type of shape representations will not be discussed here, for further details refer to [BMP02, YKR+08].

### 2.1 Landmark based representation

Representing a shape with a set of landmarks (key-points extracted from the outline of an object’s shape) has a long history in biology [BR+71, Boo94]. Most of the early approaches use length or angle between landmarks to analyse variations due to growth or anatomical changes, while neglecting the general geometric information associated with the coordinates of the landmarks. Considering landmarks along with the coordinates was developed, independently, by Bookstein [Boo84, Boo86] and Kendall [Ken84, Sma96]. Bookstein’s representation<sup>1</sup> is mainly motivated and developed from the point of view of biological applications. While Kendall’s representation, which is more theoretical, was developed for astronomical and archaeological applications. In what follows, we will mainly discuss Kendall’s representation.

---

<sup>1</sup>Sometimes referred to as Bookstein’s shape coordinates.

## 2.1 Landmark based representation

---

Let  $(p_1, \dots, p_z) \in \mathbf{R}^{2z}$ , where  $p_i \in \mathbf{R}^2$ , be a set of  $z$  landmarks extracted from the outline of a planar shape<sup>1</sup>. Subsequently, following Kendall's definition of shape, Definition 1 in Chapter 1, transformations of landmark points due to translation, scale and rotation are filtered out. To that end, location and scale are removed from the contour using  $\kappa(\cdot)$ , which is defined as follows

$$\kappa(p_1, \dots, p_z) = (p_1^*, \dots, p_z^*) = \left( \frac{p_1 - \bar{p}}{h}, \dots, \frac{p_z - \bar{p}}{h} \right), \quad (2.1)$$

where

$$\bar{p} = \frac{1}{z} \sum_{i=1}^z p_i \in \mathbf{R}^2, \quad \text{and} \quad h = \sqrt{\sum_{i=1}^z \|p_i - \bar{p}\|_2^2} \in \mathbf{R}. \quad (2.2)$$

The function  $\kappa(\cdot)$  removes location and scale by centering the landmarks to zero mean and scaling them to unit norm. As a result, the dimensionality of the landmarks is reduced. That is, centering the landmarks confines our observation to a subspace which we denote with  $\mathbf{V}$  and defined as

$$\mathbf{R}^{2z} \supset \mathbf{V}^{2z-2} = \left\{ (p_1, \dots, p_z) : \frac{1}{z} \sum_{i=1}^z p_i = 0 \right\}. \quad (2.3)$$

Further rescaling the landmarks to unit norm leads to a hypersphere in  $\mathbf{V}^{2z-2}$ . Thus,  $\kappa$  can be defined as

$$\kappa : \mathbf{R}^{2z} \rightarrow \mathbf{S}^{2z-3}. \quad (2.4)$$

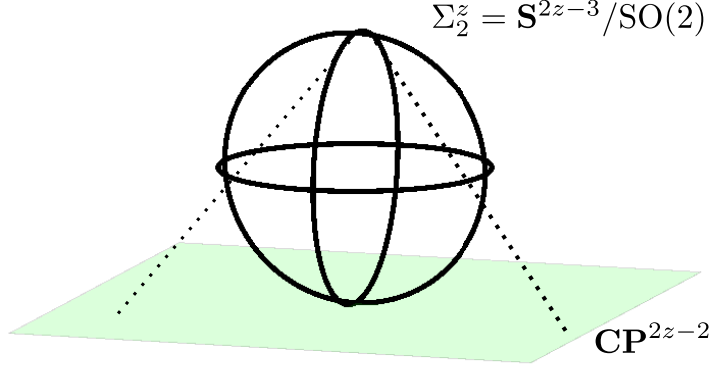
According to Kendall, the space  $\mathbf{S}^{2z-3}$  is called *pre-shape* space. Note that, rotational variation is not yet removed, hence the name *pre-shape* space. In order to remove rotational variation, it is necessary to define the orientation of the landmarks as a function of the points; this is similar to defining scale and location as the norm and the mean of the landmarks, see (2.1). However, there exists no such function for  $z > 2$  landmark points [Sma96]. Consequently, in

---

<sup>1</sup>Note that, in biological and archaeological applications a landmark does not necessarily have to be on the contour of the shape.

## 2. CURVED SHAPE REPRESENTATION

---



**Figure 2.1:** Kendall's shape space

Kendall's formalism, an equivalence class is defined for the set of all possible rotational variations of a given landmark set as follows

$$[\kappa(p_1^*, \dots, p_z^*)] = \{(R(p_1^*, \theta), \dots, R(p_z^*, \theta)) : 0 \leq \theta \leq 2\pi\} \subset \mathbf{S}^{2z-3}, \quad (2.5)$$

where  $R(p_i^*, \theta)$  defines rotation of the point  $p_i^*$  by  $\theta$  about the origin; we remind the reader that the landmarks are on a plane. As a result, a shape is not identified with an element of  $\mathbf{S}^{2z-3}$  but with an equivalence class that is defined as (2.5). Hence, the *shape space* of landmarks, using Kendall's formalism, is the set of all equivalent classes defined as

$$\Sigma_2^z = \{[\kappa(p_1^*, \dots, p_z^*)] : \kappa(p_1, \dots, p_z) \in \mathbf{S}^{2z-3}\} = \mathbf{S}^{2z-3}/\text{SO}(2), \quad (2.6)$$

the subscript on  $\Sigma$  shows the dimension of the landmark points, while the superscript shows the number of the landmarks. In other words,  $\Sigma_2^z$  is a quotient space of the *pre-shape* space. The shape space given (2.6) is further shown to be isometric to the *complex projective space*,  $\mathbf{CP}^{2z-2}$ . Intuitively, one can understand the shape space  $\Sigma_2^z$  as a Riemannian submersion<sup>1</sup> of  $\mathbf{S}^{2z-3}$  to  $\mathbf{CP}^{2z-2}$ . In effect, every orbit in  $\mathbf{S}^{2z-3}$  (2.5) is identified with a line in  $\mathbf{CP}^{2z-2}$ . Next, we will consider (2.6) as the shape space of landmarks and briefly discuss it from the two main perspectives introduced in Chapter 2.

---

<sup>1</sup>Submersion is a differentiable map between two manifolds such that its differential at each point is a linear surjective map between the tangent spaces.

1. *Completeness*: Although elegant, Kendall's shape space  $\Sigma_2^z$ , which is developed based on a set of landmarks, does not define how to select the landmark points on a shape outline. Consequently, the landmark points are selected either based on problem specific knowledge or defined to satisfy a certain mathematical criteria. Hence it is entirely possible to have two different sets of landmarks for a given shape which will be mapped to different representations. Consequently, landmark-based representations do not meet the criterion of completeness.
2. *Metric*: The geodesic distance between two shapes, in a *pre-shape* space, can be defined in a closed form as follows

$$d_{\mathbf{S}^{2z-3}}(\kappa(X_1), \kappa(X_2)) = \cos^{-1} \langle \kappa(X_1), \kappa(X_2) \rangle \quad (2.7)$$

where  $\langle \cdot, \cdot \rangle$  is used to denote the inner product. The distance defined in the *pre-shape* is further generalized to the *shape space* as

$$d_{\Sigma_2^z}(\kappa(X_1), \kappa(X_2)) = \min_{\theta_1, \theta_2} \{d_{\mathbf{S}^{2z-3}}(\kappa(X_1, \theta_1), \kappa(X_2, \theta_2))\}. \quad (2.8)$$

As we alluded earlier,  $\Sigma_2^z$  is isometric to  $\mathbf{CP}^{2z-2}$ . Consequently, (2.8) can be expressed in a closed form since orbits in  $\mathbf{S}^{2z-3}$  correspond to set of complex lines in a complex projective space. This leads to what is known as the Fubini-Study metric [MM05, Hel62], in mathematics, and Procrustean distance in Kendall's shape space [Sma96].

## 2.2 Immersion based representation

In a direct contrast to landmark-based approaches, one can construct a shape representation by starting from a general mathematical structure— the space of all smooth functions which we denote by  $C^\infty(\mathbf{R}^n, \mathbf{R}^m)$ . Such a space, however, is not a Banach space<sup>1</sup>. Consequently, a much more general topological space is considered for function based shape space construction; that is to consider curved

---

<sup>1</sup>A vector space with a distance function such that the distance defines a topology that is complete. This is analogous to  $\mathbf{R}^n$  except that it is infinite dimensional.

## 2. CURVED SHAPE REPRESENTATION

---

shapes as an embedding or immersion of  $\mathbf{S}^1$  to  $\mathbf{R}^n$  [Men13, MM03, SKJJ11, MM07, MMSY07]. As such, a curve is defined as

$$c : \mathbf{S}^1 \rightarrow \mathbf{R}^n, \quad \text{s.t.} \quad \dot{c} \neq 0, \quad (2.9)$$

where  $\dot{c}$  represents the first derivative with respect to its parametrization. The space of all possible curves, defined as (2.9), is denoted by  $\text{Imm}(\mathbf{S}^1, \mathbf{R}^n)$ , where “Imm” stands for immersion; in [MM03], a detailed account is given on how  $\text{Imm}(\mathbf{S}^1, \mathbf{R}^n)$  can be made into a manifold.

Similar to the landmark-based approach, variation due to rotation and translation can be filtered out by filtering rotation,  $\text{SO}(n)$ , and translation,  $\mathbf{R}^n$ , from  $\text{Imm}(\mathbf{S}^1, \mathbf{R}^n)$ . More importantly, however, in the immersion based approach variation in the parameter space  $\mathbf{S}^1$  is also considered, unlike the landmark based approach which relies on a predefined point selection (parametrization). To further explain, let  $c_1(\mathbf{S}^1)$  be the analytical representation of an observed curve. Intuitively, if we trace the path of  $c_1(\mathbf{S}^1)$ , its shape does not change with respect to how fast we travel, which direction we travel, or our starting point. To be more precise, consider the immersion  $c_1$  defined below

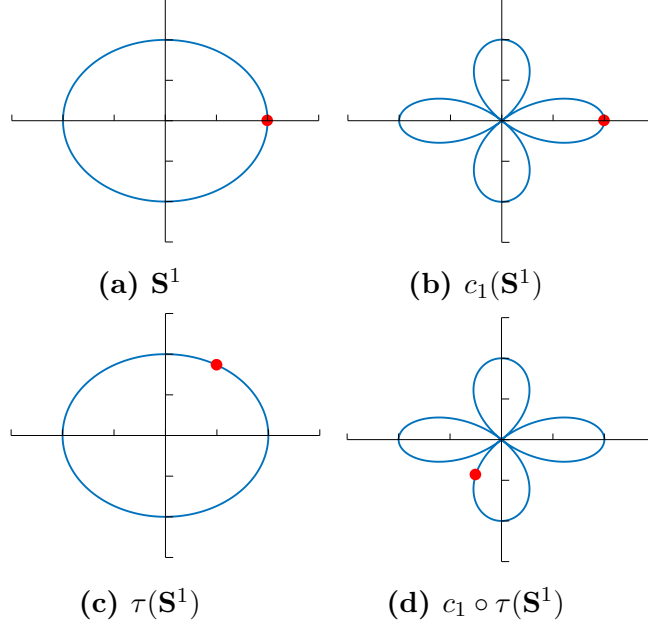
$$c_1 : \mathbf{S}^1 \ni \theta \rightarrow (\cos 2\theta \cos \theta, \cos 2\theta \sin \theta) \in \mathbf{R}^2. \quad (2.10)$$

Subsequently, let  $\tau$  be a reparametrization that maps  $\mathbf{S}^1$  to itself, defined as

$$\tau : \theta \rightarrow (\varphi + \theta)_{\text{mod}(2\pi)}, \quad (2.11)$$

for some  $\varphi \in [0, 2\pi]$ . Using (2.11) it is possible to select different starting points, from which we can trace the shape of  $c_1$ , by selecting different values for  $\varphi$ . Regardless of the starting point, however, the curve’s shape remains the same with respect to Kendall’s definition, see Figure 2.2. Nevertheless, the composition of  $c_1$  with  $\tau$  is different from its original definition, that is,  $c_1(\mathbf{S}^1) \neq c_1 \circ \tau(\mathbf{S}^1)$ . This kind of representational variations are referred to as variation due to reparametrization, or as deformation of the parameter space.

Similar to location and rotation, variation due to reparametrization has to be removed from the representation space to avoid redundancy. Consequently, we denote all possible smooth and invertible reparametrizations of  $\mathbf{S}^1$  as  $\text{Diff}(\mathbf{S}^1)$ .



**Figure 2.2:** Variation due to reparametrization: (a) and (c) show the parameter space and its reparametrization along with the red dot to denote the starting point. (b) and (d) show the curve trace defined on (a) and (c), respectively.

Note that  $\text{Diff}(\mathbf{S}^1)$  is a group under composition. As a result, the composition of a given curve  $c_1 \in \text{Imm}(\mathbf{S}^1, \mathbf{R}^n)$  with  $\tau \in \text{Diff}(\mathbf{S}^1)$  can be seen as a group action from the right.

Similar to (2.5), we can use  $\text{Diff}(\mathbf{S}^1)$  to define an equivalence class of shapes and consider the collection of these equivalent classes as a shape space. The equivalence class of a given shape  $c_1$  under  $\text{Diff}(\mathbf{S}^1)$  is defined as follows

$$[c_i] = \{c_i \circ \tau : \forall \tau \in \text{Diff}(\mathbf{S}^1)\}, \quad (2.12)$$

Subsequently, the collection of all equivalent classes of type (2.12) are given by taking the quotient space of  $\text{Imm}(\mathbf{S}^1, \mathbf{R}^n)$ , defined by  $\text{Diff}(\mathbf{S}^1)$  group, which is written as

$$B(\mathbf{S}^1, \mathbf{R}^n) = \text{Imm}(\mathbf{S}^1, \mathbf{R}^n) / \text{Diff}(\mathbf{S}^1). \quad (2.13)$$

Finally, by removing variations due to rotation and location, the space of immersed function-based shape representation is given as

$$B(\mathbf{S}^1, \mathbf{R}^n) / SO(n) \times \mathbf{R}^n. \quad (2.14)$$

## 2. CURVED SHAPE REPRESENTATION

---

In the literature,  $\text{Imm}(\mathbf{S}^1, \mathbf{R}^n)$  is referred to as space of parametrized curves, and  $B(\mathbf{S}^1, \mathbf{R}^n)$  as space of unparametrized curves [BBMM14]. Subsequently, we briefly comment on the completeness of the immersion-based shape space and possible metrics that can be defined on it.

1. *Completeness*: Immersion based representations are complete. That is, unlike landmark based representations, curved shapes are treated as infinite dimensional objects and no prior point selection is imposed, mainly in the space of parametrized curves. For practical purposes, however, such representations are approximated with piecewise polynomial functions which can potentially introduce redundancies in case of unparametrized curves [BBM16].
2. *Metric*: Several metrics have been studied in the immersion based shape representation [MM07, SKJJ11, MM03, SYM07]. Among them, the simplest metric is the  $L_2$  metric, which is defined as

$$G_c(h, h) = \int \langle h, h \rangle ds, \quad (2.15)$$

where  $h$  is a vector field along the curve  $c$ , and is integrated with respect to arc length of the curve  $c$ ,  $ds = \|\dot{c}(\theta)\| d\theta$ . Note that  $G_c$  is a smoothly varying metric in  $\text{Imm}(\mathbf{S}^1, \mathbf{R}^n)$  and is invariant to reparametrization. In general, the distance between two curves for a given metric, in our case using (2.15), is given as

$$d_{Imm}(c_1, c_2) = \min_{\gamma(t)} \int_0^1 \sqrt{G_c(\dot{\gamma}(t), \dot{\gamma}(t))} dt, \quad (2.16)$$

where  $\gamma(t)$  is a given curve with the initial conditions of  $\gamma(0) = c_1$  and  $\gamma(1) = c_2$ , and  $\dot{\gamma} = d\gamma(t)/dt$ . Equation (2.16), which is a distance in  $\text{Imm}(\mathbf{S}^1, \mathbf{R}^n)$  space, can be generalized to  $B(\mathbf{S}^1, \mathbf{R}^n)$  as

$$d_B([c_1], [c_2]) = \min_{\tau \in \text{Diff}(\mathbf{S}^1)} \min_{\gamma(t)} \int_0^1 \sqrt{G_c(\dot{\gamma}(t), \dot{\gamma}(t))} dt, \quad (2.17)$$

such that  $\gamma(0) = c_1$  and  $\gamma(1) = c_2 \circ \tau$ , for some  $\tau \in \text{Diff}(\mathbf{S}^1)$ . Nevertheless, there exists no closed form solution for the geodesic equation of the metric (2.15). More importantly, the induced geodesic distance, using (2.15),



## 2.2 Immersion based representation

---

between two given curves vanishes [MM05, MM03]. That is, for some  $\epsilon > 0$  there can always be a path between two given curves with length smaller than  $\epsilon$ . Consequently, to overcome the vanishing distance, several works have considered what are called Sobolev type metrics [BBM16]. A typical Sobolev metric, in a shape space, is the first-order Sobolev metric which adds a weighted first derivative of the vector fields to (2.15) as follows

$$H_c(h, h) = \int \langle h, h \rangle + A \langle D_s h, D_s h \rangle ds, \quad (2.18)$$

where  $A > 0$  and  $D_s h$  is the derivative of the vector field with respect to the arc length, i.e.,  $D_s h = \frac{1}{\|\dot{c}(\theta)\|} \dot{h}(\theta)$ . The geodesic equation for (2.18) involves second-order partial differential equations, which are numerically difficult to solve [BBMM14]. Note that, we are referring to the geodesic equation of parameterized curves. In a particular case [SKJJ11, MMSY07], a closed form geodesic equation or numerically efficient solvers are provided. In most cases, however, the geodesic equation for the Sobolev metric is solved iteratively. In [BBMM14], the efficient solutions for Sobolev type metrics, [SKJJ11, MMSY07, MSJ07], are all discussed as the pullback<sup>1</sup> metric of a generalized isometric mapping<sup>2</sup> that maps a shape space to a simple constant metric. To further explain, let  $F$  be an isometric map of type defined as follows

$$F : \text{Imm}(\mathbf{S}^1, \mathbf{R}^n, G_c) \rightarrow C^\infty(\mathbf{S}^1, \mathbf{R}^n, L_2). \quad (2.19)$$

Subsequently, the metric  $G_c$  can be defined using the much simpler  $L_2$  norm, since (2.19) is isometric, as follows

$$G_c(h, h) = L_2(D_c F(h), D_c F(h)), \quad (2.20)$$

where  $G_c$  and  $L_2$  are the respective metrics of the spaces, and  $D_c F$  is the derivative of  $F$  at  $c \in \text{Imm}(\mathbf{S}^1, \mathbf{R}^n)$  and  $h$  is a vector at  $c$ 's tangent space.

---

<sup>1</sup>The derivative of a function between one manifold and another defines a linear map between the tangent spaces of its domain and co-domain. These maps are known as pullback or pushforward, depending on the mapping directions.

<sup>2</sup>A function is isometric if it preserves the distance function defined in its domain space.

## 2. CURVED SHAPE REPRESENTATION

---

For example, the Square Root Velocity (*SRV*) transform [SKJJ11], is a mapping of the form (2.19) defined as

$$F(c) = \frac{\dot{c}}{\sqrt{\|\dot{c}\|}}, \quad (2.21)$$

which essentially describes an immersion  $c$  by its normalized tangent vector. Subsequently, (2.21) induces the following metric

$$G_c(h, h) = \int_{\mathbf{S}^1} \langle D_s h, \dot{c} \rangle^2 + \frac{1}{4} \langle D_s h, \ddot{c} \rangle^2 ds, \quad (2.22)$$

note that  $\dot{c}$  and  $\ddot{c}$ , represent tangential speed and curvature of the curve, respectively. As a result, one can interpret (2.22) as measuring the curve’s deformation, described by a tangent vector  $h$ , components in the tangential and curvature direction. The generalized form of (2.22) are known as “elastic metrics” [SKJJ11, MSJ07].

### 2.3 Deformable-template based representation

An alternative to both immersion based and landmark based representations is to represent curves based on their difference from a given *template* shape. In such a scenario, one needs to define the *template* shape, and the type of *deformations* allowed. The earliest work, in such direction is by D’Arcy Thompson, where similarity transformations of animal body forms are studied [D92]. Particularly with the work of Grenander and his collaborators [GM07, KMG98, Gre97, GM94], advanced pattern theory was developed with an emphasis on the deformation and similarity group of shapes. Following the work of Grenander, in [You98] a metric defined on a shape deformation space is shown to induce a distance metric on its shape space, given the deformation space is a group and acts transitively on elements of the shape space. Subsequently, deformable-template based analysis has been formulated to study deformation of landmark points, images, and curved shapes [Tro95, You10b, You98, BMTY05].

Nevertheless, in this section, we will discuss deformable-template based representations in the context of curved shape representation.

### 2.3.1 Active shape representation

One of the main approaches in analysing shape variations from a template shape is introduced in [CTCG95] as *active shape representation*. Active shape representation is based on Principal Component Analysis (PCA), hence considers only linear deformations of the shape. It starts by approximating a curve  $c$  by a manually selected landmark points as  $c = (p_1, \dots, p_z)$ . Note that, in active shape representation, curve variation due to the deformation of the parameter space is not considered since the landmark points are selected manually they are considered to be matching points. Next, for a collection of such curved shapes,  $M^q = \{c_1, \dots, c_q\}$ , the mean shape  $\bar{c}$  is computed as

$$\bar{c} = \frac{1}{q} \sum_{i=1}^q c_i. \quad (2.23)$$

Subsequently, the mean shape is selected as a *template* shape and a linear deformation model is defined as

$$\bar{c} + \sum_{j=1}^l a_j b^j, \quad (2.24)$$

such that the  $b_{j=1, \dots, l}$  are linearly independent  $l$  directional vectors in which the template shape  $\bar{c}$  can vary– the  $a_j$  are scalar weighting factors for each basis. The basis and the scalar weights can be learned from  $M^q$  using PCA. Consequently, the *pre-shape* space of active shape representations is the vector space spanned by the  $b^j$ . However, similar to earlier approaches, variation due to rotation and/or translation has to be cleared from the dataset.

1. *Completeness*: According to our definition, active shape representations are not complete. Since the landmark points are selected manually, there is no criterion on how to select them.
2. *Metric*: Since the deformation space is a vector space, the metric used in active shape representation is the  $L_2$  norm. As a result, the deformation of a given shape  $c_1$  from the template shape  $\bar{c}$  is measured as

$$\left\| \sum_{i=1}^l (b^i)^T (c_1 - \bar{c}) \right\|_2. \quad (2.25)$$

## 2. CURVED SHAPE REPRESENTATION

---

Subsequently, the geodesic deformation is a simple straight line given as

$$t \sum_{i=1}^l (b^i)^T (c_1 - \bar{c}) + (1 - t)\bar{c} \quad \text{s.t.} \quad t \in [0, 1]. \quad (2.26)$$

One of the main advantages of active shape models is that both the metric and the deformation space are very simple and computationally appealing. However, the model works based on the assumption that variations of a given shape category, from a template shape, are small enough to be captured by a linear model. The difficulty, however, is when the variations in a shape category are due to large nonlinear deformations, e.g., occlusion, articulation, viewpoint variation. In such cases, an active shape model is not adequate.

### 2.3.2 Nonlinear deformable-template representations

Mainly, with the work of Grenander et al. [GM94, GK93, AGP91], deformation from a template shape is studied as a group action of piecewise linear or nonlinear group.

There are two basic approaches in modelling nonlinear deformations from a template shape: 1) modelling the deformation space as finite dimensional group, which usually leads to a matrix group, and 2) modelling the deformation space as an infinite dimensional group, which leads to a Hilbert space with function composition as the group operator— note that, the choice of the deformation space is highly correlated with the choice of the data structure, see Chapter 1. In [You98, Tro95], the considered deformation space is a set of  $C^d$  functions, for some  $d \geq 0$ . Together with a composition operation, the space of  $C^d$  functions is shown to be an infinite dimensional Lie group. In [GM94], a general pattern theory that analyses configurations generated by geometric units (lines, points, etc.) based on the deformation from a template configuration was presented. In this case, the considered deformation space is a direct product of matrix Lie groups, see [Gre97] for a brief overview. Although not discussed in detail here, in [KMG98], the Frenet-Serret frame has been used to represent deformations of a curved shape without referring to a template shape. However, the Frenet-Serret frame is applicable only to curves with a non-vanishing curvature.

1. *Completeness*: Apart from the data structure, the completeness of a given deformable-template based shape representation depends on how the group acts on the shape space. In the strictest sense, a deformation group  $\Psi$  of a shape space can be considered complete if its action is *regular*. That is, for every  $c_1$  and  $c_2$  in a given shape space there exists exactly one  $\psi_i \in \Psi$  such that  $\psi_i c_1 = c_2$ . Alternatively, completeness can be framed as an optimization problem in cases where several possible deformations exist that can map  $c_1$  to  $c_2$ – the distance metric in deformation based representation is mostly defined in terms of optimization.
2. *Metric*: The general distance framework for a group based shape deformation analysis is formalized in [You98], for both finite or infinite dimensional representations. Assuming the group action is *transitive* (there exists  $\psi_i \in \Psi$  for every  $c_1, c_2 \in \mathcal{S}$  such that  $\psi_i c_1 = c_2$ ), the similarity between two shapes is measured by a deformation that minimizes an objective function called *effort functional*– the distance between the group’s identity and the given deformation, see [You98, You10a]. Similar to the landmark and immersion based approaches variation due to scale, location and rotation are cleared from the considered shape space. Subsequently, the distance between a template shape  $c_t$  and a given shape  $c_1$  is formulated as

$$d_s(c_1, c_t) = \min_{\psi \in \Psi} \{d_\Psi(e, \psi) : c_1 = \psi(c_t)\}, \quad (2.27)$$

where  $d_\Psi$  is the metric in the deformation space. In the infinite dimensional setting, distance in the deformation space is closely related with distance in the original shape space. Moreover, (2.27) can further be formalized to address optimal parameter estimation of shape representations. Computational cost of (2.27) is dependent on the type of deformation space, and distance metric defined on it.

## 2.4 Summary

In Chapter 2, we have described three distinct types of curved shape representation approaches and discussed the advantages and challenges of each approach.

## 2. CURVED SHAPE REPRESENTATION

---

Despite the apparent incompleteness of the landmark based approach, there are several landmark based applications in medical image analysis problems mainly because of its computational efficiency [DM98]. Meanwhile, the immersed function based representation has been used in specific classification problems in biology [LSZ10, LKSM14], where human annotation of the landmark points are either unavailable or are unreliable. Nevertheless, most immersion based representations are solved with numerically intensive computational schemes, except for a few special cases [SKJJ11, MMSY07]. Moreover, shapes are assumed to be locally smooth thus the distance function is sensitive to local noise. Alternatively to both landmark and immersion based representations, deformable-template models emphasize the deformation space, which usually is a group, for curve representations. Deformable-template models are generally good modelling approaches to represent a class-specific shape variation. The hypothesis is, as in active shape representation, that shape variations in a class-specific shape category are small with respect to a template shape. Consequently, the deformation space can be described with significantly fewer variables as opposed to the general deformation space— we will discuss PCA, which can be treated as deformable-template model, more specifically in Chapter 5. Nevertheless, deformable-template models are not purely a shape representation framework, since the representation is conditioned on the shapes category and has to refer to a template shape for distance computation. However, the general framework of quantifying distance as the minimum of the effort functional (2.27) is independent of the template notion; in Chapter 3 we will show the relationship between the proposed metric and (2.27).

In this thesis, using a novel finite dimensional curved shape representation, we address the high computational cost and noise sensitivity associated with distance functions in immersed function-based representations. Furthermore, the dependence of finite-dimensional representations, landmark-based approaches, on manually annotated points is addressed.

# Chapter 3

## Deformation based curved shape representation

### 3.1 Motivation

In this chapter, we present a novel curved shape representation framework that draws its motivation from the three distinct shape representation approaches discussed in Chapter 2. The representation presented here aims to embody some of the advantages of the landmark based, immersion based, and deformable-template based representations in a single framework.

Similar to the landmark-based representation, we aim to develop a representation with a computationally efficient distance function that is based on the approximation of a continuous curve by a finite set of points. Nevertheless, we do not depend on annotated points for the approximation as is done in landmark-based approaches. Instead, we use arc length based point sampling as an approximation of a continuous curve with a known curve parametrization. Consequently, general questions which usually are framed in the immersed function based setting, e.g., deformation transportation and optimal parameter estimation, can be asked in the proposed curve representation, and approximate solutions can be provided without incurring the mathematical and computational complexity that comes along with infinite dimensional space representations. Finally, similar to deformable-template based approaches our representation is based on a deformation group. Hence, we exploit the advantages of the group structure in

### 3. DEFORMATION BASED CURVED SHAPE REPRESENTATION

---

our proposed framework. Nevertheless, we do not refer to a template shape. As a result, our framework can be used to address model-free problems like shape retrieval from a given dataset. However, distinctively to our representation, the notion of distance emphasizes the difference between intrinsic characteristics of curved shapes. In most of the earlier approaches discussed in Chapter 2, distance is measured by the absolute deformation of each point—the difference between the two distance notions will be discussed in detail later. In that regard, what is proposed in this chapter is similar to the Frenet-Serret frame presented in [KMG98]. In summary, the main goals of the proposed curved shape representation are: 1) to develop a robust representation that leads to computationally efficient (closed form) geodesic curve and geodesic distance equations, 2) to develop a framework to handle optimal parameter estimation of curves and deformation analysis so that the representation can be used to later develop statistical models of a shape category.

The chapter is organized as follows: in Section 3.2, we present the proposed curved shape representation by starting from the data structure in Section 3.2.1 leading to the proposed representation in Section 3.2.2. In Section 3.3, we discuss distance metric and geodesic curves in the proposed curved shape representation space. In Section 3.4, the properties and interpretation of the proposed distance function is presented. Experimental results and comparisons are presented in Section 3.5. The chapter concludes with final remarks in Section 3.6.

## 3.2 Deformation based representation

### 3.2.1 Data structure

We start by defining a continuous curve on a given parameter space as

$$c : I \rightarrow \mathbf{R}^n, \quad (3.1)$$

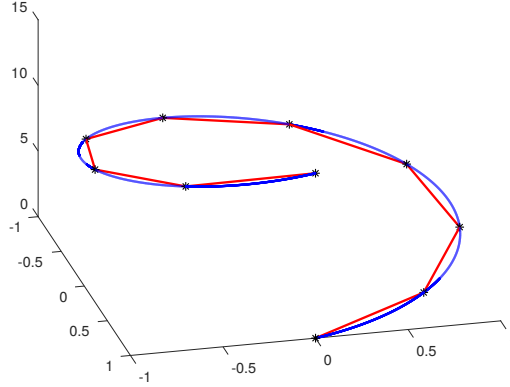
where  $I$  is the parameter space such that  $I \subset \mathbf{R}$ , e.g.,  $[0, 1]$ . Subsequently, we approximate the continuous curve  $c$  by defining a *monotonic and injective* map from a discrete parameter space to the arc length parameter space as

$$\xi : [0, z] \rightarrow s(t), \quad \text{s.t.} \quad \xi(0) = s(0) \quad \text{and} \quad \xi(z) = s(1), \quad (3.2)$$



### 3.2 Deformation based representation

---



**Figure 3.1:** Illustration of an arc length based curve sampling. The blue curve shows the continuous curve, and the points connected by the red lines are sampled based on arc length.

where  $s(\cdot)$  denotes the arc length function defined on  $I$  as  $s(t) = \int_0^t \|\dot{c}(t)\|_2 dt$ . Note that  $[0, z]$  is a discrete set of points starting from 0 to  $z$ , unlike  $[0, 1]$  which is a closed interval. Geometrically,  $\xi$  gives a polygon inscribed in  $c$ , see Figure 3.1. To simplify our discussion, we will always assume a continuous curve  $c$  is parametrized by arc length, that is  $\|\dot{c}(t)\|_2 = 1$ . Subsequently, for any given sampler  $\xi$ , the following relation

$$\|c \circ \xi(i+1) - c \circ \xi(i)\|_2 \leq \int_{\xi(i)}^{\xi(i+1)} \|\dot{c}\|_2 ds, \quad (3.3)$$

is true, since a straight line is the shortest distance between any two points. If we further impose the following constraint

$$\|c \circ \xi(i+1) - c \circ \xi(i)\|_2 = k, \quad \forall i \in [0, z] \quad (3.4)$$

for some positive  $k > 0$ , we get a set of points sampled with a uniform Euclidean distance. We denote samplers that satisfy (3.4) with  $*$  as  $\xi^*$ . It is possible to select different  $z$  points for the approximation. In fact, as  $z$  gets larger the limiting case is the continuous curve  $c$ ; the length,  $\ell$ , of the approximated curve  $c \circ \xi^*$  is bounded from above by the length of the curve  $c$

$$\ell(c \circ \xi^*) = \sum_{i=1}^z \|c \circ \xi^*(i+1) - c \circ \xi^*(i)\| \leq \sum_{i=1}^z \int_{\xi(i)}^{\xi(i+1)} \|\dot{c}\| ds. \quad (3.5)$$

### 3. DEFORMATION BASED CURVED SHAPE REPRESENTATION

---

Subsequently, we will use  $\xi$  and  $\xi^*$ , alternatively, to approximate a continuous curve  $c$  by  $\tilde{c} = c \circ \xi$  and  $\tilde{c}^* = c \circ \xi^*$ , respectively.

More importantly, since  $\xi$  is defined to partition the arc length parametrization space, it is invariant to reparametrization. To show this, we first define reparametrization of a curve as a function from the parameter space to itself as  $\beta : I \rightarrow I$ , such that  $\beta$  is a diffeomorphism (bijective and smooth). Now the sampling of the curve  $c$  under  $\beta$  is given as

$$\xi : [0, z] \rightarrow s \circ \beta(t), \quad \text{s.t. } t \in I. \quad (3.6)$$

and

$$s(t) = \int_{t_0}^t \|\dot{c}\| dt = \int_{\beta^{-1}(t_0)}^{\beta^{-1}(t)} \|\dot{c} \circ \beta\| \frac{d\beta}{dt} dt = s \circ \beta(t). \quad (3.7)$$

Hence, for some given  $\beta$  one can use the sampling  $\xi$  invariantly by correcting the rate of change, and the starting and ending points of the integral to the original parameter space  $\beta^{-1}(I)$ . Consequently, in cases where the parameter space and its reparametrization is given, the sampling of a curve can be done invariantly with respect to any reparametrization. Nevertheless, particularly in computer vision problems, one often does not have explicit knowledge of the parametrization and only observes the outline of an object's shape. In such a case, additional criteria are used to estimate an optimal parametrization of curves.

In contrast to landmark points that are discussed in Chapter 2, the data structure obtained by a given sampler  $\xi$  associates no special meaning to the points except that they approximate a curve  $c$  asymptotically as  $z \rightarrow \infty$ . This flexibility on the data structuring part will later be used to estimate optimal curve parametrization by setting different geometric criteria on the sampler functions  $\xi$ . Thereby, overcoming the main difficulty of using landmark based curve representations in cases where parametrization of the curve is not known or human annotation is not provided.

In the next section, we present a deformation based curved shape representation that is based on the data structure obtained by  $\xi^*$ – throughout Chapter 3, we assume the parametrization of the curves is given.

### 3.2.2 Representation

Let  $C$  be the set of all possible continuous curved shapes in  $\mathbf{R}^n$  approximated by  $z$  points sampled with  $\xi^*$ , as discussed in Section 3.2.1. Subsequently, we would like to study variations in  $C$  in a way that satisfies Kendall's definition of shape. Hence, we completely remove variations due to similarity transformations from  $C$ . To achieve this, the main idea we employ here is to define a similarity transformation as a function of  $\tilde{c}^*$  such that the definition satisfies  $A(a\tilde{c}^*) = aA(\tilde{c}^*)$  for some  $a \in \mathbf{R}$ , which is needed to normalize its effect in  $C$ . The following list shows how each of the similarity transformations are cleared from  $C$ .

1. **Scale:** We define a uniform scaling of a curve  $\tilde{c}^*$  as

$$S(\tilde{c}^*) = \sqrt{\sum_{i=1}^z \|p_i - \bar{p}\|_2^2} \quad \text{s.t.} \quad \bar{p} = \frac{1}{z} \sum_{i=1}^z p_i, \quad (3.8)$$

which computes the  $L_2$  norm of the points when considered as a vector in  $\mathbf{R}^{nz}$ . Note that (3.8) satisfies  $S(a\tilde{c}^*) = aS(\tilde{c}^*)$  for some  $a > 0$ . Furthermore, according to (3.8) the scale of a shape is characterized by a positive scalar multiplication of the curve  $\tilde{c}^*$ . Similar to earlier approaches discussed in Chapter 2, every curve  $\tilde{c}^* \in C$  is normalized to unit norm as follows

$$\tilde{c}^* = (p_1/S(\tilde{c}^*), \dots, p_z/S(\tilde{c}^*)). \quad (3.9)$$

Note that, one can replace (3.8) with a much broader definition of scale, e.g., non uniform scaling.

2. **Translation:** Translation of a given curve  $\tilde{c}^*$  is defined with respect to a fixed starting point  $p^s$  as

$$T(\tilde{c}^*) = p_1 - p^s. \quad (3.10)$$

Using the above definition we filter translational variations from  $C$  as

$$\tilde{c}^* = ((p_1 - T(\tilde{c}^*)), \dots, (p_z - T(\tilde{c}^*))). \quad (3.11)$$

In effect, the translation of a curve is characterized by the vector difference between the starting point  $p_1$  and the fixed location  $p^s$ . In this case,

### 3. DEFORMATION BASED CURVED SHAPE REPRESENTATION

---

$T(\tilde{c}^* + b) = T(\tilde{c}^*) + b$ , where  $b$  is a  $z$  times concatenation of some  $p^t \in \mathbf{R}^n$ . Consequently, (3.11) sets (3.10) to zero, thereby clearing effects of translation from a given curve. An alternative definition to (3.10) would be to define translation as the mean of the points in  $\tilde{c}^*$ .

3. **Rotation:** Unlike the earlier transformations, rotation can not be defined as a function of curve  $\Theta(\tilde{c}^*)$  such that  $\Theta(\theta\tilde{c}^*) = \theta\Theta(\tilde{c}^*)$ , see [Sma96] for further details. As a result, it cannot be filtered out from  $C$ . Instead, we define an equivalence class in  $C$  as

$$[\tilde{c}_i^*] = \{\tilde{c}_j^* \in C \mid \exists R \in \text{SO}(n) : R\tilde{c}_j^* = \tilde{c}_i^*\}. \quad (3.12)$$

Subsequently, the family of all equivalent classes in  $C$  is denoted as

$$\mathcal{C} = C/\text{SO}(n). \quad (3.13)$$

Now, what is left in  $\mathcal{C}$  is shape informative deformation of the curves that we would like to study and quantify. To that end, consider the mapping function  $f$  defined on  $\mathcal{C}$  as follows

$$\begin{aligned} f : \mathcal{C} &\rightarrow \text{SE}(n)^{z-1} \\ f(\tilde{c}_1^*) &= G = (g_1, \dots, g_{z-1}), \end{aligned} \quad (3.14)$$

such that

$$g_i \times p_i = p_{i+1}, \quad (3.15)$$

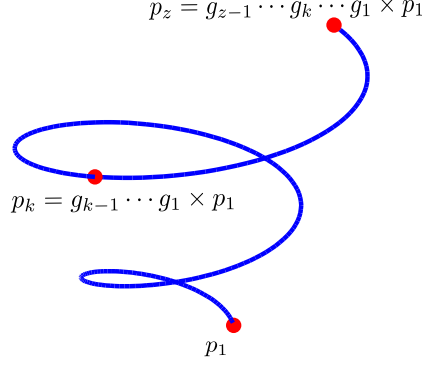
where  $g_i$ ,  $i = 1, \dots, z - 1$ , are rigid transformation matrices that map one point to the next point successively with respect to a reference coordinate frame. Intuitively,  $f(\cdot)$  attempts to encode every curve  $\tilde{c}^* \in \mathcal{C}$  as the path of a travelling particle by the rigid transformation matrices, see Figure. 3.2. Meanwhile, a rigid transformation matrix  $g_i$  is defined as

$$g_i = \begin{pmatrix} R & v \\ 0 & 1 \end{pmatrix} \quad \text{s.t.} \quad R \in \text{SO}(n) \quad \text{and} \quad v \in \mathbf{R}^n, \quad (3.16)$$

where  $\text{SO}(n)$  is a group of rotation matrices in  $n$ -dimensional space, and  $v$  is a translation vector. Consequently, the semi-direct product of the rotation and

## 3.2 Deformation based representation

---



**Figure 3.2:** Illustration of the proposed representation. Given a fixed starting point  $p^s$ , the curve is reconstructed by the successive action of the  $(g_1, \dots, g_{z-1})$ .

translation groups define the group of rigid transformation matrices  $\text{SE}(n)$ , denoted as

$$\text{SE}(n) = \text{SO}(n) \times \mathbf{R}^n. \quad (3.17)$$

Ideally, we would like the representation  $f(\cdot)$  to be *complete*, as discussed in Chapter 2. Unfortunately, there is no unique rigid transformation matrix between any two given points  $p_1$  and  $p_2$ . For instance, there exists a translation vector  $v \in \mathbf{R}^n$  for every  $R \in \text{SO}(n)$  such that  $p_2 = Rp_1 + v$ . To overcome this challenge, we further restrict  $f$  to only consider optimal rotation, i.e., the closest rotation matrix to the identity  $e$ , and optimal translation vector, the closest to zero vector. A unique optimal rotation matrix between two points is computed, using Rodrigues' rotation formula, as

$$R(\theta) = e + \sin(\theta)n + (1 - \cos(\theta))n^2, \quad (3.18)$$

where  $e$  is the group identity,  $\theta = \cos^{-1}(\langle p_1, p_2 \rangle)$ , and  $n$  is the axis of rotation given by the cross product as  $n = p_1 \times p_2$ . Note that (3.18) defines a geodesic curve between the identity matrix  $e$  and  $R(\theta)$ , parametrized by  $\theta$ . Such geodesic curves in  $\text{SO}(n)$  are known as exponential maps, see Appendix A for details. Subsequently, the optimal translation vector is given as

$$v = R(\theta)p_1 - p_2. \quad (3.19)$$

### 3. DEFORMATION BASED CURVED SHAPE REPRESENTATION

---

The translation is optimal since the shortest distance between points is a straight line. As a result of the restriction, the inverse of the mapping function  $f(\cdot)$  can be defined on the set of optimal rigid transformation matrices as follows

$$\begin{aligned} f^{-1} : \text{SE}(n)^{z-1} &\rightarrow \mathcal{C} \\ &= (p^s, g_1 p^s, g_2 g_1 p^s, \dots, (\prod_{i=1}^{z-1} g_i) p^s). \end{aligned} \quad (3.20)$$

As a result,  $f$  defines an injective map between  $\mathcal{C}$  and  $\text{SE}(n)^{z-1}$ . Moreover, the equivalence relationship defined on the set  $\mathcal{C}$  (3.12) is also carried over to the representation space  $\text{SE}(n)^{z-1}$ . That is, shapes that are defined as equivalent in  $\mathcal{C}$  are also equivalent in the representation space  $\text{SE}(n)^{z-1}$ .

**Proposition 1.** *If  $G^1$  and  $G^2$  are the representations of  $\tilde{c}_1^*, \tilde{c}_2^* \in [\tilde{c}_3^*]$  then  $G^1$  is equivalent to  $G^2$  by conjugacy,  $G^1 \sim G^2$ .*

*Proof.* Let  $\tilde{c}_1^* = (p^s, \dots, p_z^1)$  and  $\tilde{c}_2^* = (p^s, \dots, p_z^2)$ . Since,  $\tilde{c}_1^*, \tilde{c}_2^* \in [\tilde{c}_3^*]$  we can write  $\tilde{c}_2^* = R\tilde{c}_1^*$ , where  $R \in \text{SO}(n)$ , see (3.12). Rotation is an isometry, thus  $R\tilde{c}_1^* = (Rp^s, \dots, Rp_z^1)$ . The representation  $f(\tilde{c}_2^*) = (g_1^2, \dots, g_z^2)$ , such that

$$\begin{aligned} g_i^2 \times p_i^2 &= p_{i+1}^2 \\ &= R \times p_{i+1}^1, \end{aligned}$$

it can then be expressed in terms of  $f(\tilde{c}_1^*)$  as follows

$$\begin{aligned} g_i^2 \times R \times p_i^1 &= R \times g_i^1 \times p_i^1 \\ g_i^2 \times R &= R \times g_i^1. \end{aligned}$$

Thus,  $f(\tilde{c}_1^*)$  and  $f(\tilde{c}_2^*)$  are equivalent by conjugacy, i.e.,  $f(\tilde{c}_2^*) = Rf(\tilde{c}_1^*)R^{-1}$ . ■

Consequently, the equivalence relationship defined in (3.12) can be written in the representation space as

$$f([\tilde{c}_i^*]) = \{f(\tilde{c}_j^*) \mid \exists R \in \text{SO}(n)^* : Rf(\tilde{c}_j^*)R^{-1} = f(\tilde{c}_i^*)\}, \quad (3.21)$$

where  $\text{SO}(n)^*$  is the subgroup of  $\text{SO}(n)^{z-1}$  defined as

$$\text{SO}(n)^* = \{\forall i, j \in [0, z-1] : R_i R_j^T = e\}. \quad (3.22)$$

### 3.2 Deformation based representation

---

Computationally, if two given shapes belong to the same equivalent class, i.e.,  $f(\tilde{c}_1^*), f(\tilde{c}_2^*) \in f([\tilde{c}_3^*])$ , then the corresponding eigenvalues of the transformation matrices in  $f(\tilde{c}_1^*)$  and  $f(\tilde{c}_2^*)$  are the same. Finally, the representation space of equivalent classes is written as

$$\text{SE}(n)^{z-1}/\text{SO}(n)^* = \{f([\tilde{c}_i^*]) : \tilde{c}_i^* \in \mathcal{C}\}. \quad (3.23)$$

In what follows, we will describe deformation in the representation space and enforcing constraints defined in  $\mathcal{C}$  to the representation space.

**Shape deformation in the representation space:** Any given shape representation in  $\text{SE}(n)^{z-1}/\text{SO}(n)^*$  is deformable to any other curved shape representation in  $\text{SE}(n)^{z-1}/\text{SO}(n)^*$ . Let  $f([\tilde{c}_1^*])$  and  $f([\tilde{c}_2^*])$  be the representations of two different equivalent classes of curves,  $\tilde{c}_1^*$  and  $\tilde{c}_2^*$ . Given the knowledge of the curves' parametrizations and rotational alignment, we can extract the equivalent deformation class directly as

$$[G^L] = Rf(\tilde{c}_2^*)f(\tilde{c}_1^*)^{-1}R^{-1} \quad \forall R \in \text{SO}(n)^*, \quad (3.24)$$

so that

$$[G^L]f([\tilde{c}_1^*]) = f([\tilde{c}_2^*]). \quad (3.25)$$

Note that  $f(\tilde{c}_1^*)^{-1}$  is an element wise inversion of the representation, unlike  $f^{-1}(\tilde{c}_1^*)$  which is a map from the representation space to  $\mathcal{C}$ . Since  $\text{SE}(n)^{z-1}/\text{SO}(n)^*$  is a group, the deformation  $[G^L]$  is an element of  $\text{SE}(n)^{z-1}/\text{SO}(n)^*$  acting from the left. Alternatively, a deformation that acts from the right is defined as

$$[G^R] = Rf(\tilde{c}_1^*)^{-1}f(\tilde{c}_2^*)R^{-1} \quad \forall R \in \text{SO}(n)^*. \quad (3.26)$$

The above definitions are for a class of deformations that can map one equivalent class of shapes to another. However, if we want to estimate the deformation between two particular shapes  $f(\tilde{c}_i^*) \in f([\tilde{c}_1^*])$  and  $f(\tilde{c}_j^*) \in f([\tilde{c}_2^*])$ , the deformation that acts from the left is given as

$$G^L = Rf(\tilde{c}_i^*)R^{-1}f(\tilde{c}_j^*)^{-1}, \quad (3.27)$$

### 3. DEFORMATION BASED CURVED SHAPE REPRESENTATION

---

such that  $R = (R_1^*, \dots, R_z^*)$  and

$$R^* = \arg \min_{R_i \in SO(n)} \|R_i \tilde{c}_i^* - \tilde{c}_j^*\|_2. \quad (3.28)$$

The solution for (3.28) can be estimated in a closed form given the curve parametrizations are known, see Appendix B on how to compute optimal rigid transformation matrix between two points or set of points. Henceforth, we will use  $G^L$  and  $G^R$  to describe deformation types of (3.27) that act from the left or from the right, respectively.

**Constraints on  $\mathcal{C}$ :** It is possible to enforce constraints defined in  $\mathcal{C}$  into the representation space. The main types of constraints we consider are based on categorization of curves as either *open curves* or *closed curves* which are defined as follows:

1. *Open curves:* a curve  $\tilde{c}^* = (p^s, \dots, p^z)$  is an open curve if  $p^s \neq p^z$ . The definition is translated into the representation space by including an additional transformation  $g_z$  into the representation such that  $g_z \times p_z = p^s$  and  $g_z \neq e$ .
2. *Closed curves:* a curve  $\tilde{c}^* = (p^s, \dots, p^z)$  is a closed curve if  $p^s = p^z$ . Similar to open curves, this condition is translated to the representation space by forcing the last added transformation matrix to be identity,  $g_z = e$ .

Hence, a representation space that handles both closed and open curves is given as  $SE(n)^z/SO(n)^*$ , which will be used to denote the proposed representation space henceforth.

Moreover, it is possible to add additional and elaborate constraints on the set of curves, e.g., consider only subset of  $\mathcal{C}$  curves that preserve a specific geometric moment. However, it might prove to be difficult to enforce complicated constraints on the representation space directly without compromising computational effectiveness. This is because an enforced constraint will most likely define a submanifold or a subset in the representation  $SE(n)^z/SO(n)^*$  where a closed form geodesic equation might not be possible to define, see Chapter 7 for details.



### 3.3 Metric and geodesic curves

The proposed representation space is a matrix Lie group, which is not a Euclidean space. As a result, the usual definition of distance as a straight line does not generalize to  $SE(n)^z$ . Alternatively, techniques from differential geometry can be used to define distance in a curved space. In this section, we will overview concepts from Lie theory and differential geometry to later develop a distance function in  $SE(n)$ , and its direct product group  $SE(n)^z$ .

A Lie group is a smooth manifold with smooth group operations; that is, the group's binary operator  $(x, y) \mapsto xy^{-1}$  is  $C^\infty$ . The tangent space at the identity of the group  $e$  is an algebra called Lie algebra, which we will denote by  $\mathfrak{g}$ . The smooth and invertible binary operator of a Lie group enables one to define a diffeomorphism onto itself. For instance, consider a left translation of a Lie group defined as  $L_a : G \rightarrow aG, a \in G$ , where  $G$  denotes the Lie group. Meanwhile, the differential structure, due to Lie groups being smooth manifolds, enables one to do calculus on a Lie group. To compute distance, volume and other geometric notions, however, an additional structure called metric is needed. Consequently, a Lie group  $G$  can be complemented with a smoothly varying metric tensor  $\langle \cdot, \cdot \rangle$ , making it a Riemannian manifold. The metric tensor  $\langle \cdot, \cdot \rangle$  is defined at the tangent space  $T_g G$  as  $\langle \cdot, \cdot \rangle_g : T_g G \times T_g G \rightarrow \mathbf{R}$  for every  $g \in G$ , see [MB93, dCV92] for a detailed discussion.

Subsequently, the length of a given curve that is defined as

$$\gamma : [t_0, t_1] \rightarrow G, \tag{3.29}$$

where  $[t_0, t_1] \subset \mathbf{R}$ , is given as

$$\ell(\gamma) = \int_{t_0}^{t_1} \sqrt{\langle \dot{\gamma}(t), \dot{\gamma}(t) \rangle_{\gamma(t)}} dt. \tag{3.30}$$

Note that  $\dot{\gamma}(t) \in T_{\gamma(t)} G$  is the derivative of the curve with respect to  $t$ . There are several possible curves that start  $\gamma(t_0) \in G$  and end at  $\gamma(t_1) \in G$ . Hence, the shortest distance between two group elements  $g_1 = \gamma(t_0)$  and  $g_2 = \gamma(t_1)$  is measured by the connecting curve that has the minimum length, defined as

$$d(\gamma(t_0), \gamma(t_1)) = \min\{\ell(\gamma)\}. \tag{3.31}$$

### 3. DEFORMATION BASED CURVED SHAPE REPRESENTATION

---

Equation (3.31) defines an objective functional in an infinite dimensional space and its solution is estimated by solving for its critical point which is defined by a second-order partial differential equation, generally known as *Euler-Lagrangian* equation.

Alternatively, we can generalize the notation of a straight line as a shortest distance in a vector spaces to the curved space of the group. Subsequently, a curve that defines a parallel vector field along itself is defined to be similar to a straight line in Euclidean space. Hence, such a curve has to satisfy the following

$$\frac{D}{dt} \left( \frac{d\gamma}{dt} \right) = 0. \quad (3.32)$$

Intuitively, (3.32) can be understood as a requirement for the second derivative of the curve  $\gamma$  to be zero or orthogonal to the tangent space of  $T_{\gamma(t)}G$ . Such a curve is known as a geodesic curve. Furthermore, if a curve satisfies (3.32) then the length of  $\dot{\gamma}(t)$  at each point  $\gamma(t)$  stays constant. That is

$$\frac{D}{dt} \langle \dot{\gamma}, \dot{\gamma} \rangle = 2 \left\langle \frac{D}{dt} \left( \frac{d\gamma}{dt} \right), \dot{\gamma} \right\rangle = 0. \quad (3.33)$$

Consequently, a curve that satisfies (3.32) is a locally length minimizing curve. Hence solving for (3.32) is equivalent to solving for (3.31).

In addition to the above definitions, in a Lie group, if a vector field  $\dot{\gamma}(h)$  is left invariant, i.e., if the following is true for  $h \in G$

$$dL_a \dot{\gamma}(h) = \dot{\gamma}(ah) \in T_{ah}G, \quad (3.34)$$

then it defines a parallel vector field with respect to the left translation action. Consequently, the integral curve of a left invariant vector field defines a geodesic curve as  $\gamma(t) = \exp(t\dot{\gamma})$ .

**Metric in SE(n):** A metric in a group is defined by a symmetric positive definite matrix at each point of the group element which we denote by  $\langle \cdot, \cdot \rangle_{g_i}$ . A smoothly varying metric on a Lie group is said to be left invariant if the left translation diffeomorphism is an isometry, i.e., if the following is true

$$\langle x, y \rangle_e = \langle dL_a x, dL_a y \rangle_a, \quad \forall x, y \in \mathfrak{g}, \forall a \in G, \quad (3.35)$$

where  $dL_a$  is the derivative of the left translation  $L_a$  at the identity  $e$ . In other words, a left invariant Riemannian metric can be identified with  $\langle \cdot, \cdot \rangle_e$  through the pullback map,  $dL_a^{-1}$ . One can define different types of metrics on  $\text{SE}(n)$ , however, having a left invariant metric (3.35) plays a crucial role in solving for the geodesic curve objective functional (3.31) analytically. Consequently, in this thesis, we will consider a constant product metric (a metric that remains the same at every tangent space of the group element) of the form defined as follows

$$\langle \cdot, \cdot \rangle_e = \begin{pmatrix} w_1 I_n & 0 \\ 0 & w_2 I_n \end{pmatrix}, \quad (3.36)$$

where  $w_1$  and  $w_2$  are scalar weighting constants.

**Geodesics in  $\text{SE}(n)$ :** Geodesic curves are defined for each group,  $\text{SO}(n)$  and  $\mathbf{R}^n$ , as

$$\varphi_R(t) = R_1(R_1^{-1}R_2)^t \quad (3.37)$$

$$\varphi_v(t) = v_1 + (v_2 - v_1)t. \quad (3.38)$$

Equation (3.38) is clearly a geodesic curve, since it is just a straight line in  $\mathbf{R}$ . Meanwhile, (3.37) has been proven to be geodesic in  $\text{SO}(n)$  [Moa02, Bha09]. Subsequently, a geodesic curve in the semi-direct product space  $\text{SE}(n)$  is given as

$$\varphi(t) = \begin{pmatrix} \varphi_R(t) & \varphi_v(t) \\ 0 & 1 \end{pmatrix}, \quad (3.39)$$

where  $t \in [0, 1]$ . The curve  $\varphi(t)$  defines a vector field along itself which is given as

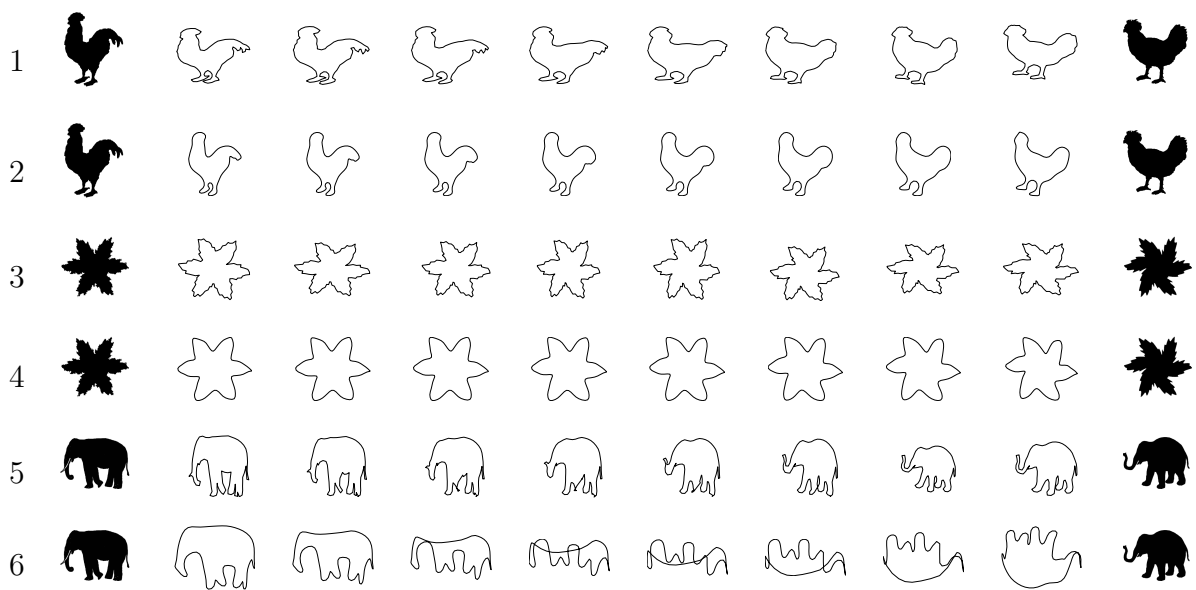
$$\dot{\varphi}(t) = \begin{pmatrix} R_1(R_1^{-1}R_2)^t \log(R_1^{-1}R_2) & (v_2 - v_1) \\ 0 & 0 \end{pmatrix}. \quad (3.40)$$

The defined vector field (3.40) has been shown to be parallel along the curve in [ZKC98, Par95]. Hence, the curve (3.39) is geodesic. Furthermore, the length of the tangent vectors (3.40) remains constant under the metric (3.36). To that end, one can consider (3.39) as a parametrized left translation of the group identity  $e$  that is defined as

$$L_{\varphi(t)} : e \rightarrow \varphi(t) \quad (3.41)$$

### 3. DEFORMATION BASED CURVED SHAPE REPRESENTATION

---



**Figure 3.3:** Shapes along the geodesic path between the initial shape (first column) and target shape (last column). The odd rows show results from our approach while the even rows are results from infinite dimensional based approach called SRV function [SKJJ11]. All shapes are represented by 100 uniformly sampled and normalized points. We note that results from [SKJJ11] are smoothed and loses local features of the shapes.

Subsequently, to show that the lengths of the tangent vectors (3.40) remain constant we need to show that the metric (3.36) is invariant. That is

$$\langle \dot{\varphi}(t), \dot{\varphi}(t) \rangle_{\varphi(t)} = \langle dL_{\varphi(t)}^{-1} \dot{\varphi}(t), dL_{\varphi(t)}^{-1} \dot{\varphi}(t) \rangle_e \quad (3.42)$$

$$= \langle \varphi(t)^{-1} \dot{\varphi}(t), \varphi(t)^{-1} \dot{\varphi}(t) \rangle_e. \quad (3.43)$$

Using matrix algebra we can simplify  $\varphi(t)^{-1} \dot{\varphi}(t)$  at any  $t$  as

$$\varphi(t)^{-1} \dot{\varphi}(t) = \begin{pmatrix} \log(R_1^{-1} R_2) & \varphi_R(t)^{-1}(v_2 - v_1) \\ 0 & 0 \end{pmatrix}. \quad (3.44)$$

Note that  $\varphi_R(t)^{-1}$  is a rotation matrix, hence it preserves the Euclidean distance. As a result, (3.43) remains constant at any point in  $t$  which implies that the metric is preserved. Subsequently, following the definition of geodesic distance (3.30), the distance between  $g_1, g_2 \in \text{SE}(n)$  is given by

$$d(g_1, g_2) = \int_0^1 \sqrt{\langle dL_{\varphi(t)}^{-1} \dot{\varphi}(t), dL_{\varphi(t)}^{-1} \dot{\varphi}(t) \rangle_e} dt, \quad (3.45)$$

and since the lengths of the tangent vectors are preserved at each point we can rewrite the distance function as

$$d(g_1, g_2) = (1 - 0) \times \sqrt{\langle dL_{\varphi(t)}^{-1} \dot{\varphi}(t), dL_{\varphi(t)}^{-1} \dot{\varphi}(t) \rangle_e}, \quad (3.46)$$

which is reduced to the following under (3.36)

$$d(g_1, g_2) = (w_1 \|\log(R_1^T R_2)\|_F^2 + w_2 \|v_2 - v_1\|_2^2)^{1/2}. \quad (3.47)$$

Although one can define different metrics by weighting the rotation and translation distance differently, in this thesis we will set both  $w_1$  and  $w_2$  to 1. Even though it is not particularly important for our discussion, fixing  $w_1$  and  $w_2$  to unit makes sure that an orthonormal basis of the lie algebra  $\mathfrak{g}$  stays orthonormal everywhere [Mil76]. Consequently, the distance in the representation space  $\text{SE}(n)^z$  is given by the product metric of (3.47) as follows

$$\mathfrak{d}(G_1, G_2) = (d(g_1^1, g_1^2)^2 + \dots + d(g_z^1, g_z^2)^2)^{1/2}. \quad (3.48)$$

Similarly, the geodesic curve is given by taking the direct product of (3.39) as

$$\zeta(G_1, G_2) = (\varphi(t)_1, \dots, \varphi(t)_z). \quad (3.49)$$

Note that both (3.48) and (3.49) assume a known parametrization of the curves, see Figure 3.3 to compare geodesic deformations of curves under (3.49) and the SRV framework proposed in [SKJJ11].

### 3. DEFORMATION BASED CURVED SHAPE REPRESENTATION

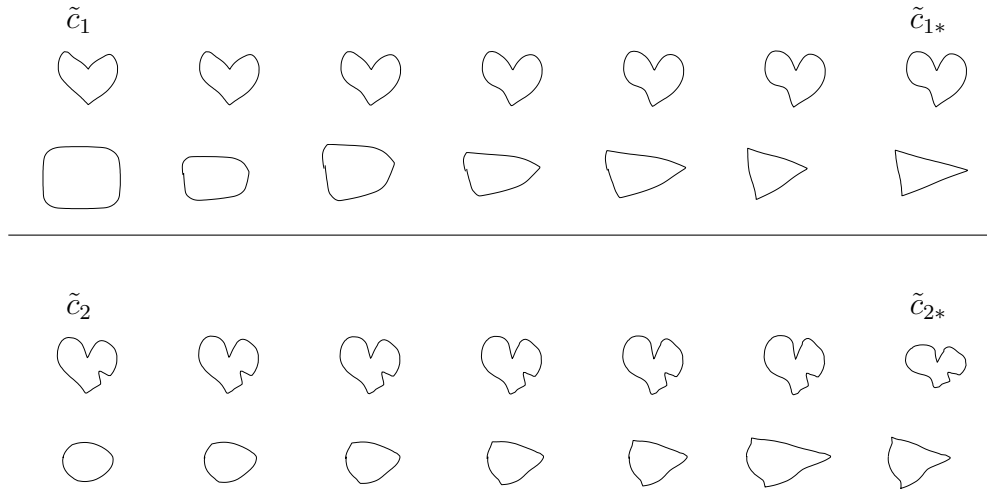
---

## 3.4 Characteristics and interpretation

In this section, we present some of the important properties of the proposed representation and its distance metric followed by a discussion on its interpretation and relationship with earlier works.

### 3.4.1 Properties

1. **Left invariant metric and its application:** As described earlier, the distance metric (3.36) is left invariant. The implication of this is that the distance between two shapes is invariant to a deformation acting on both shapes from the left. More concretely, let  $G' \in \text{SE}(n)^z$  be a deformation acting on the representations of two distinct shapes,  $f(\tilde{c}_1)$  and  $f(\tilde{c}_2)$ , then  $\mathfrak{d}(f(\tilde{c}_1), f(\tilde{c}_2)) = \mathfrak{d}(G'f(\tilde{c}_1), G'f(\tilde{c}_2))$ . This fact can be observed by inserting the action of  $G'$  into (3.48), in which case it will cancel itself out. As discussed in [DAO16, DAO17], this property is particularly important in transporting deformation between two similar shapes. To clarify further, we consider below a deformation transportation problem discussed in [SKJJ11]. Let  $\tilde{c}_1$  and  $\tilde{c}_{1*}$  be shape contours representing exactly the same real world object denoted by  $O_1$ , except that  $\tilde{c}_{1*}$  is deformed under some unknown external factor. For instance,  $\tilde{c}_1$  and  $\tilde{c}_{1*}$  can be contours of  $O_1$  from a different viewpoint. Furthermore, let  $O_2$  denote a similar, but not identical, object to  $O_1$  with  $\tilde{c}_2$  as its shape contour. Subsequently, given  $\tilde{c}_1$ ,  $\tilde{c}_{1*}$ , and  $\tilde{c}_2$  the problem of estimating  $\tilde{c}_{2*}$ , a deformed  $\tilde{c}_2$  under the same external factor, can be framed as a deformation transportation problem. In [SKJJ11], the deformation is estimated by transporting the vector field along the geodesic curve connecting  $\tilde{c}_1$  and  $\tilde{c}_{1*}$ . Alternatively in our framework, the deformation due to an external factor can be factored out as  $G^L = f(\tilde{c}_{1*})f(\tilde{c}_1)^{-1}$ . Consequently, since our metric is left invariant,  $\mathfrak{d}(f(\tilde{c}_1), f(\tilde{c}_2)) = \mathfrak{d}(G^L f(\tilde{c}_1), G^L f(\tilde{c}_2))$ . Thus,  $f(\tilde{c}_{2*}) = G^L f(\tilde{c}_2)$ , see Figure 3.4.
2. **No high order differential quantities:** Most infinite dimensional representations define a similarity metric between curved shapes based on high



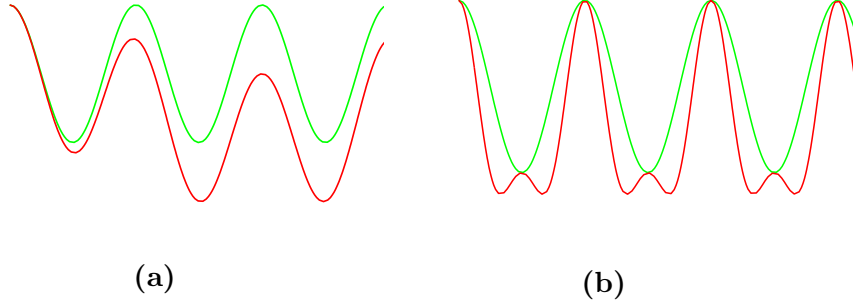
**Figure 3.4:** Deformation transport between curves with known parametrization. The first set of shapes shows two examples where  $\tilde{c}_1$  deforms to  $\tilde{c}_{1*}$ . The second set shows the transported deformation to their similar objects  $\tilde{c}_2$  to give  $\tilde{c}_{2*}$ , respectively.

order differential quantities [SKJJ11, Men13]. However, differential quantities are highly sensitive to noise and local perturbations. Consequently, a pre-smoothing stage is necessary before analysing a shape. On the other hand, pre-smoothing a shape can potentially filter legitimate features, see Figure. 3.3. Although the proposed representation is based on the relative transformation matrices between neighbouring points, it is not as sensitive as curvature is, for example, to local perturbations [MCH<sup>+</sup>06]. Hence, pre-smoothing is not required.

3. **Relative vs absolute deformation of points:** Let  $\tilde{c}_1 = (p^s, \dots, p_z^1)$  and  $\tilde{c}_2 = (p^s, \dots, p_z^2)$  be two shapes that are scaled and aligned, see Figure. 3.5. Subsequently, we define *absolute deformation of points* (AD) as the transformation of every  $p_i^1 \in \tilde{c}_1$  to match its corresponding point  $p_i^2 \in \tilde{c}_2$ . In earlier works, such deformations are modelled by displacement vector fields [Ami94] and in some cases by the action of a matrix Lie group [FB12, GM07]. If we formulate the matching of each point, from  $\tilde{c}_1$

### 3. DEFORMATION BASED CURVED SHAPE REPRESENTATION

---



**Figure 3.5:** Two pairs of curves. Under absolute deformation (AD), the curves in (b) are more similar than the curves in (a), while under relative deformation (RD) the curves in (a) are more similar than the curves in (b).

to  $\tilde{c}_2$ , by the action of rigid transformation matrices then the AD is given by  $G = (T_1, \dots, T_z) \in \text{SE}(n)^z$  such that

$$\tilde{c}_2 = G\tilde{c}_1 = (T_1 p^s, \dots, T_z p_z^1). \quad (3.50)$$

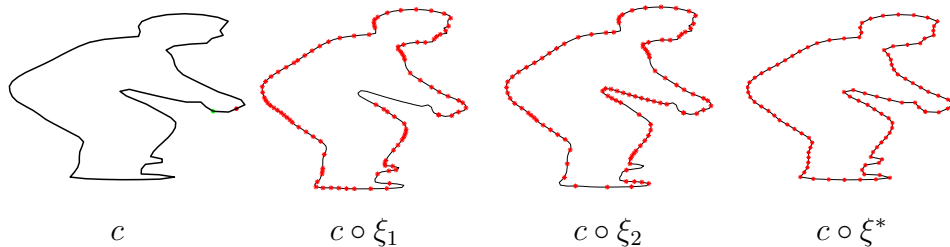
Alternatively, rather than aligning every matching point, as is done with AD, one can align the intrinsic properties of the curves— these are properties of a curve that are mainly invariant to displacement. In this thesis, intrinsic properties of a given curve are approximated by the relative deformation of the curves' points, as defined in (3.14) by  $f(\cdot)$ . Hence, alignment of  $\tilde{c}_1$  and  $\tilde{c}_2$  under *relative deformation of points* (RD) is given as  $f(\tilde{c}_2) = f(\tilde{c}_1)G^R$ , where  $G^R \in \text{SE}(n)^z$ . Subsequently, using (3.48) the cost of deformation under RD is given by

$$\begin{aligned} d_r(\tilde{c}_1, \tilde{c}_2) &= \left( \sum_{i=1}^z d(e, g_i^R) \right)^{1/2} \\ &= \left( \sum_{i=1}^z \left( \|\log((R_i^1)^T R_i^2)\|_F^2 + \|v_i^2 - v_i^1\|_2^2 \right)^{1/2} \right)^{1/2}. \end{aligned} \quad (3.51)$$

While, the cost of deformation under AD is given as

$$\begin{aligned} d_a(\tilde{c}_1, \tilde{c}_2) &= \left( \sum_{i=1}^z d(e, T_i) \right)^{1/2} \\ &= \left( \sum_{i=1}^z \left( \|\log(R_i)\|_F^2 + \|v_i\|_2^2 \right)^{1/2} \right)^{1/2}. \end{aligned} \quad (3.52)$$





**Figure 3.6:** Different samplings of a given continuous curve  $c$ . The red dots denote sampled points. The last sampling  $\xi^*$  is a uniform sampling. Note that  $\xi_1$  and  $\xi_2$  do not preserve shape.

Note that  $d_r$  is the same as (3.48). The main difference between  $d_r$  and  $d_a$  is tolerance to displacement—  $d_r$  emphasizes intrinsic differences between curves, while  $d_a$  emphasizes cost of displacement per point, see Figure 3.5.

### 3.4.2 Interpretation

The proposed distance function, as derived in (3.48), has a direct relationship to what is called *effort functional* [You98, Tro95, You10a]. In general, distance between curved shapes based on effort functional is written as

$$d_s(c_1, c_2) = \min_{\psi \in \Psi} \{d_\Psi(e, \psi) : c_1 = \psi(c_2)\}, \quad (3.53)$$

where  $\Psi$  is the space of allowed deformations for matching  $c_2$  to  $c_1$ ,  $d_s$  and  $d_\Psi$  are distance metrics in the shape and deformation spaces, respectively. Such formulation leads to a natural interpretation of distance as a measure of deformation by least action, hence the term *effort functional*. see [MD10, Tro95] for further mathematical details.

Similarly, a version of (3.53) under the proposed representation can be formulated by explicitly defining the space of deformations  $\Psi$  for matching  $c_1$  and  $c_2$ . To define  $\Psi$  more precisely, we use the arc length sampler  $\xi_i$  defined in (3.2). Given a fixed number of points  $z$ , the samplers have a similar effect as to what are referred to as shape and orientation preserving diffeomorphisms [You10a]. In this case, however, there is no guarantee for an arc length sampler to preserve shape, see Figure 3.6. As such, the sampling of a shape determines the parametrization and thus defines different deformations of a given shape. Subsequently, we define

### 3. DEFORMATION BASED CURVED SHAPE REPRESENTATION

---

a set of possible deformations between continuous curves  $c_1$  and  $c_2$ , for a given  $z$ , as

$$\Psi = \{G^R \mid \exists \xi_i, \xi_j : f(c_1 \circ \xi_i)G^R = f(c_2 \circ \xi_j)\}. \quad (3.54)$$

Consequently, under the framework of the proposed representation, distance between two continuous curves is given as

$$d_s(c_1, c_2) = \min_{G^R \in \Psi} \mathfrak{d}(f(\tilde{c}_1^i), f(\tilde{c}_2^j)), \quad (3.55)$$

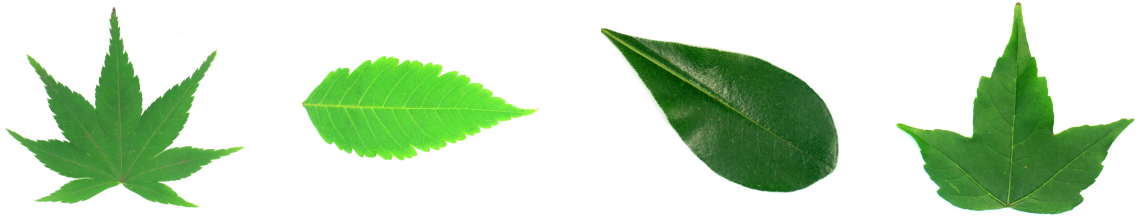
where the  $i$  and  $j$  are indicators of the sampling functions  $\xi$ . Moreover, using the left invariance property of the metric and the definition of  $\Psi$ , (3.55) can be simplified as follows

$$\begin{aligned} d_s(c_1, c_2) &= \min_{G^R \in \Psi} \mathfrak{d}(f(c_1^i), f(c_1^i)G^R) \\ &= \min_{G^R \in \Psi} \mathfrak{d}(e, G^R). \end{aligned} \quad (3.56)$$

Hence (3.56) embodies a version of the effort functional (3.53) in the proposed curve representation space. That is, it characterizes the distance between two curves by the magnitude of the closest possible deformation to the identity, or by the least right action. However, if the parametrizations of the two curves  $\xi_i$  and  $\xi_j$  are given a priori,  $\Psi$  will be a singleton—since  $G^R$  can be computed directly, as described in (3.26). For instance, we have so far assumed a perfect point-to-point correspondence between shapes. Thus,  $\xi_i$  and  $\xi_j$  are assumed to be known. Consequently, the proposed metric (3.48) can be viewed as measuring *the optimal deformation between two shapes given the parametrizations*.

## 3.5 Evaluation

In this section we present evaluation of the proposed shape representation and the distance metric. As discussed in Chapter 1, however, the parametrization of shapes is generally not known in practical applications. Consequently, to evaluate the proposed distance metric we used uniform sampling based parameter estimation which will be discussed later in Chapter 4, Section 4.3. Subsequently,



**Figure 3.7:** Example leaves from different categories of the Flavia leaf dataset [WBX<sup>+</sup>07].

in Section 3.5.1 we evaluate the representativeness of the approach on two different datasets. Later in Section 4.5.2, we evaluate the robustness of the proposed approach with respect to local noise.

### 3.5.1 Shape retrieval

Given a query shape, a shape retrieval system tries to rank shapes in a given database according to their similarity to the query. Usually, such systems are solely based on a distance metric rather than a mathematical model of shape categories. Hence, the performance of a shape retrieval system highlights the representativeness of the used distance metric. As a result, we present the evaluation of a shape retrieval system that is based on the proposed distance metric (3.48) on different datasets.

**Flavia leaf dataset** [WBX<sup>+</sup>07]: The dataset contains 32 types of leaf species with a total of 1907 examples, see Figure. 3.7. In [LKSM14] a leave-one-out test scenario was performed on the dataset to evaluate an the elastic similarity metric derived from SRV-framework [SKJJ11]. Leave-one-out is a setup where every leaf is used as a query against the rest of the dataset. To compare our approach with other methods, we also replicate the leave-one-out scenario with Mean Average Precision (MAP) used as a performance measure. For this experiment, every leaf shape is represented by  $z = 200$  points that are uniformly sampled from its contour. Table 3.1 summarizes the result of our approach and results reported in [LKSM14] and [MYVB13]. Although our method achieved high MAP, it is not necessarily inclusive of all relevant information; precision drops as recall goes to

### 3. DEFORMATION BASED CURVED SHAPE REPRESENTATION

---

Methods	MAP
Angle function [KSMJ04]	45.87
Shape context [BMP02]	47.00
TSLA [MYVB13]	69.93
Elastic metric with 200 points [LKSM14]	81.86
Gaussian elastic metric with 200 points [LKSM14]	92.37
<b>Our method with 200 points</b>	<b>94.11</b>

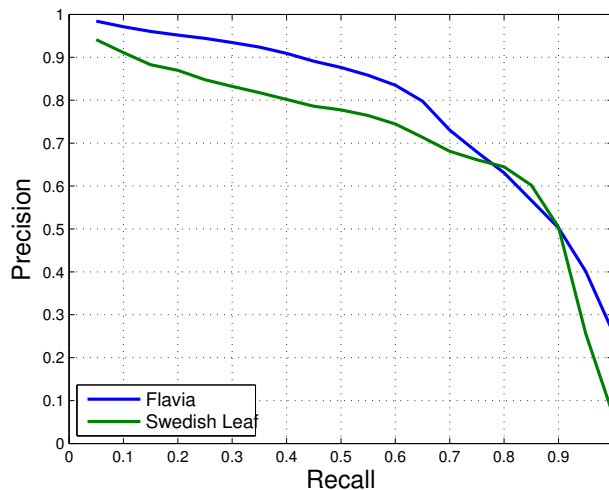
**Table 3.1:** Mean average precision (MAP) on the Flavia leaf dataset. Our result is highlighted at the bottom.

1, see Figure 3.9. Nonetheless, it outperformed the elastic shape metric and the Gaussian elastic metric, discussed in [LKSM14], in terms of MAP. One possible reason for this is that we do not pre-smooth the data and thus local details are more likely to be captured with our method.

**Swedish leaf [Söd01]:** The Swedish leaf dataset contains 1125 example leaves from 15 leaf types. The example shapes are distributed uniformly; there are 75 example leaves from each species. To compare our approach with other methods, we follow the same experimental scenario discussed in [MYVB13, LKSM14], which is nearest neighbour classification. We randomly select 25 leaf shapes from each type and use the left out 50 for testing. We repeat the experiment 100 times and take the average classification rate. In all of the 100 experiments, the shapes are represented by  $z = 200$  uniformly sampled points. Our method achieved an average classification rate of 99.50 with 0.01287 standard deviation. Table 3.2 shows the comparison of our result with other shape matching methods. We note that performance of a nearest neighbour classifier is not an explicit measure of a distance metric’s performance, as the classification is based on the best result among elements of a group. Consequently, in Figure 3.9 we show the precision-recall (PR) curve of a retrieval result obtained by performing the leave-one-out experimental scenario on the data set. Note that, the area under the PR curve is smaller in the Swedish leaf case as compared to the area under the PR curve for the Flavia leaf data set. In general, we observe that the inter-class similarity in the Flavia leaf data set is high as compared to the Swedish leaf data set making



**Figure 3.8:** Examples from different categories of the Swedish leaf dataset [Söd01].



**Figure 3.9:** Precision-Recall curves on the Flavia and the Swedish leaf dataset.

discrimination in Flavia difficult, see Figure 3.7 and Figure 3.8. However, the intra-class variation in the Swedish data set is mainly due to nonlinear elastic deformations, whereas in the Flavia it is mainly due to a combination of rotation and scaling. Hence, distance metric based on uniform sampling, (4.9), does not perform well in the Swedish leaf data set as compared to the Flavia leaf data set, in terms of precision and recall.

### 3.5.2 Robustness

In this section, we test the robustness of the proposed similarity metric to local noise. To that end, we use shapes from the fighter jets dataset [TGJ07],

**Local shape perturbations:** In general, local perturbation of a curved shape

### 3. DEFORMATION BASED CURVED SHAPE REPRESENTATION

---

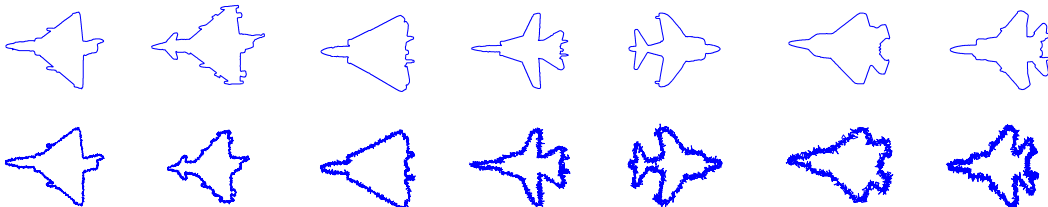
Methods	Recognition rate
Fourier descriptors [LJ07]	89.60
Shape-Tree [FS07]	96.28
TSLA [MYVB13]	96.53
Elastic metric with 200 points [LKSM14]	99.18
<b>Our approach with 200 points</b>	<b>99.50</b>

**Table 3.2:** Nearest-neighbour recognition rate on the Swedish leaf dataset. Our result is highlighted at the bottom.

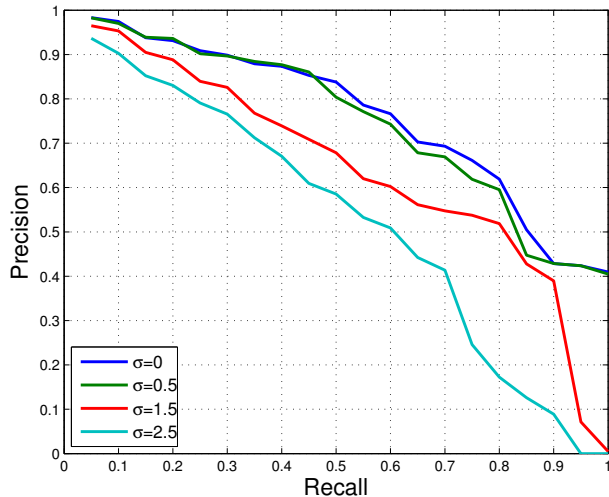
$\sigma$	0	0.5	1.5	2.5
MAP	97.11	96.72	89.95	83.27

**Table 3.3:** MAP on the fighter jets dataset with a Gaussian noise of different standard deviations.

with a noise that does not alter the shape degrades the performance of a shape alignment and retrieval system. Consequently, we evaluate the tolerance of the proposed approach to local perturbations on *fighter jets* dataset [TGJ07]. The dataset contains 7 types of fighter jets each with 30 examples. The main cause of intra-class variation is deforming parts of the plane and rotation. We begin our experiment by introducing an additive white Gaussian noise to the shapes of the fighter jets. We denote the standard deviation of the noise by  $\sigma$ . For all subsequent experiments, the contour of each shape is approximated by uniformly sampled  $z = 200$  points. Next, we do a leave-one-out test scenario where the original (noise free) dataset is queried by every shape from datasets corrupted



**Figure 3.10:** The first row shows the 7 types of fighter jets from [TGJ07]. The second row shows examples from the corrupted shapes with  $\sigma = 2.5$ .



**Figure 3.11:** Precision-Recall curves on the fighter jets dataset for different noise magnitudes quantified by the standard deviation ( $\sigma$ ).

by noise with different distributions, see Figure 3.10. Table 3.3 summarizes the computed MAP values and Figure 3.11 shows their respective precision-recall curve. In general, the proposed similarity metric is tolerant to local perturbations that do not alter the shape significantly.

### 3.5.3 Computational cost

The proposed approach is implemented in MATLAB, R2017a version, running on an Intel core i7-3540M with 3.0 GHz $\times$ 4 processing speed and 7.7 GB RAM with Ubuntu 14.04 64-bit operating system. Henceforth, the reported computational cost is for shapes approximated by  $z = 100$  sampled points. Subsequently, computing the geodesic distance (3.48), assuming the preprocessing is already performed and a known point correspondence is established, takes 0.028397 seconds. Meanwhile, geodesic distance computation in SRV framework [SKJJ11, LKSM14], with a similar preprocessing assumption, takes 0.002974 seconds, running on the same machine. Since the SRV framework is the most widely used infinite dimensional curved shape representations, we have only provided a comparison with it; except the SRV framework, most of the methods reported in Table 3.2 and

### 3. DEFORMATION BASED CURVED SHAPE REPRESENTATION

---

Table 3.1 are based on “shape descriptors”. We observe that, in this particular case, the geodesic computation is slower than SRV.

## 3.6 Conclusion

We have introduced a shape representation framework that leads to closed form geodesic distance and curve equations. Distinctively, the defined geodesic distance emphasizes intrinsic deformation rather than per point displacement. Although the distance function is computed in a closed form similar to landmark-based representations, the data structuring of the curves is different from landmark-based approaches. The proposed representation neither assigns special meaning to the discrete points nor assumes manually annotated points. Instead, the points are treated as a discrete approximations of the continuous curve which ultimately determine optimal curve deformation between two given curves. Nevertheless, if the parametrization of the curves is known a priori the distance metric measures the optimal deformation between two given points. As a result, geodesic deformation between two curves in the representation space can be interpreted as the solution to what is defined as *effort functional*, given the priori knowledge of the curve parametrizations. This formulation makes the representation flexible enough to estimate optimal curve parametrization for a given point set size,  $z$ , whenever the parametrizations of the curves are not given; we will cover optimal parameter estimation in Chapter 4. Moreover, the proposed metric depends only on the first derivative of the curves. This is different from distance metrics in infinite dimensional space which are generally defined using higher order derivatives of the curves and often require piecewise differentiability, hence tend to be sensitive to local shape perturbations. On the contrary, the proposed approach is relatively tolerant to small local shape variations. Additionally, the group structure and the defined left translation invariant metric are used to define deformation transporting between two curved shapes in a closed form for a known parametrizations. The representativeness of the proposed similarity metric is tested on leaf shape retrieval problems and outperformed both infinite dimensional representation and other shape descriptor approaches.



# Chapter 4

## Optimal parametrization

### 4.1 Motivation

The proposed curved shape representation framework, Chapter 3, developed a distance metric and geodesic deformation based on an assumed knowledge of the curve parametrizations. In such a case, the curve sampling can be done consistently by correcting the rate of change and starting and ending points, see Section 3.7. In most computer vision problems, however, the parametrization of curves is not known a priori, and has to be estimated— in computer vision, the problem of parameter estimation is known as point correspondence estimation, henceforth we will use these terms interchangeably. In this chapter, we will develop two basic approaches for point correspondence estimation:

1. Linear reparametrizations: Given two curves  $c_1$  and  $c_2$ , we assume the parametrization of  $c_2$  is a linear reparametrization of  $c_1$ 's parameter space.
2. Geometric moment preserving reparametrizations: In this case, we expand the linear reparametrization notion to a larger class of reparametrizations; reparametrizations that preserve a predefined geometric moment of the shape arguments.

Subsequently, we will compare the computational cost and performance of each approach in estimating point correspondence in cases of occlusion, missing parts or nonlinear deformations.

## 4. OPTIMAL PARAMETRIZATION

---

The chapter is organized as follows: in Section 4.2, we present related works and formulate the problem of optimal parameter estimation. In Section 4.3, a solution to special case of the parameter estimation problem is presented. Later in Section 4.4, a solution to the general case is discussed. Experimental results and comparisons are presented in Section 4.5. Finally, Section 4.6 presents conclusion of the chapter.

### 4.2 Problem definition

In general, point correspondence estimation is given by the solution of the following

$$\arg \min_{\psi \in \Psi} \{d_s(\psi(c_1), c_2) + E(\psi)\}, \quad (4.1)$$

where  $\Psi$  denotes a deformation space,  $d_s$  is a dissimilarity metric, and  $E(\cdot)$  is an energy term that measures the cost of the deformation. The objective of (4.1) is to solve not only for optimal point matching but for smooth deformations as well. There are several works in the formulation and optimization of both  $d_s$  and  $E(\cdot)$ . In the deformation based approaches, the energy term is usually described in terms of the effort functional given in (3.53) [You98]. However, there are several elaborate formulations of the energy term based on elasticity theory [You10a, MD10]. Similarly, extensive work has been done in formulating the dissimilarity function  $d_s$ . Particular to shape matching, in [CAS92] a dissimilarity function that is based on curvature is formulated. Later, in [Tag99] and [SKK03], a symmetric version of what was proposed in [CAS92] was presented and solved with dynamic programming. Regardless of the dissimilarity function, in [SN06] it is shown that restricting the matching solutions to those that preserve order of points leads to a more accurate result than the unrestricted case.

The general form of shape matching cost (4.1) is given as a measure of shape similarity and deformation cost. However, as shown in (3.56), the proposed similarity metric is equivalent to effort functional. Hence, it measures the cost of deformation as a dissimilarity between shapes. Subsequently, since

### 4.3 Special case: linear reparametrizations

---

$G^R = f(c_1 \circ \xi_i)^{-1} f(c_2 \circ \xi_j)$ , we describe (3.56) in terms of the sampling functions as

$$\arg \min_{\xi_i, \xi_j} \mathfrak{d}(f(c_1 \circ \xi_i), f(c_2 \circ \xi_j)) = \arg \min_{\xi_i, \xi_j} \mathfrak{d}(e, G^R), \quad (4.2)$$

where  $\xi_i$  and  $\xi_j$  are as defined in (3.2). Hence, we will consider the point correspondence estimation problem as the estimation of the sampling functions  $\xi_i$  and  $\xi_j$ . In Section 4.3 and Section 4.4, we will first address a specialized case of (4.2) and later we will provide a solution for the general case.

### 4.3 Special case: linear reparametrizations

Let  $I$  be the parameter space of all continuous curves, and let all possible reparametrizations of  $I$  take the following form

$$\beta = Kt + a \quad t \in I, \quad (4.3)$$

where  $K = \pm 1$  and  $a$  is any real number. Now, consider two continuous curves  $c_1$  and  $c_2$  with unknown parametrization of type (4.3) and approximated by a uniform sampler, as defined in (3.4). In such a case, the optimization problem defined in (4.2) is rewritten as

$$\arg \min_{\beta_1, \beta_2} \mathfrak{d}(f(c_1 \circ \xi_1^*), f(c_2 \circ \xi_2^*)), \quad (4.4)$$

where the  $*$  is used to identify a uniform sampler. The samplers are related with parametrizations of type (4.3) through the arc length function which can be defined for each curve as

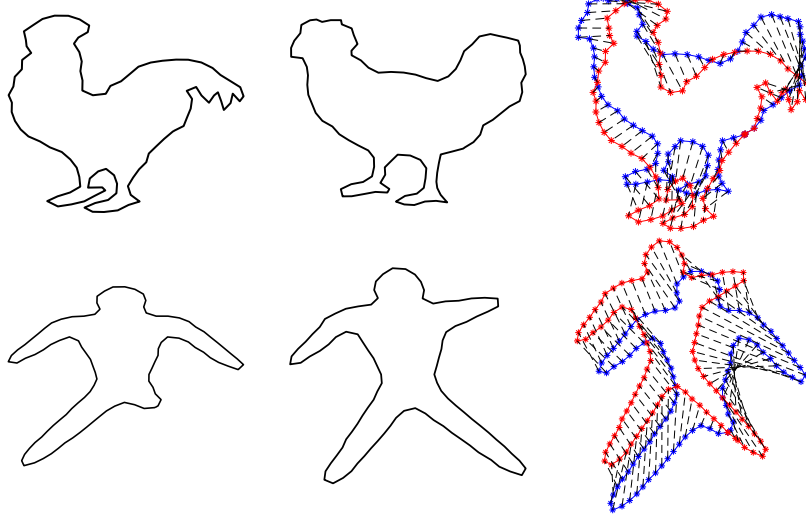
$$s_i(t) = \int_{t_0}^t \|\dot{c}_i\| dt = \int_{\beta_i^{-1}(t_0)}^{\beta_i^{-1}(t)} \|\dot{c}_i \circ \beta_i\| \frac{d\beta_i}{dt} dt = s_i \circ \beta_i(t), \quad (4.5)$$

where the index  $i$  is used as a variable to identify the curves along with their parametrizations. However, since  $\beta_i$  is assumed to be of the form of (4.3), the above formulation is reduced to

$$s_i \circ \beta_i(t) = \pm \int_{\beta_i^{-1}(t_0)}^{\beta_i^{-1}(t)} \|\dot{c}_i \circ \beta_i\| dt. \quad (4.6)$$

#### 4. OPTIMAL PARAMETRIZATION

---



**Figure 4.1:** Point correspondence estimation based on linear parametrization assumption: The first and second columns show the curves to be matched, the last column shows the matching result. Note that in the case of the first row, the estimated point correspondences are reasonably accurate than the second row where a nonlinear local deformation is observed.

Hence, reparametrizations of the form (4.3) define a family of uniform samplers that vary from one to another by orientation, sign of (4.6), and starting and ending points, boundaries of (4.6). We formalize this by defining a family of all possible samplers for a uniformly sampled curve  $\tilde{c}_i^*$  as follows

$$\Xi^* = \{\xi_j^* \mid \forall i \in [0, z] : \tilde{c}_i^* \ni p_k = p_{(k+j) \bmod z} \in \tilde{c}_j^*\}, \quad (4.7)$$

where mod represents the modulo operation. Subsequently, let  $f(\tilde{c}_i^*) = (g_1, \dots, g_z)$ , then  $f(\tilde{c}_i^*)$  under  $\xi_j^*$  sampler is given as  $(g_{(z+j) \bmod z}, \dots, g_{(z+j-1) \bmod z})$  in case of closed curves. Thus, one can understand (4.3) as defining a  $z$ -cyclic permutations of a curve's representation. Consequently, the cost of estimating an optimal, orientation preserving, parametrization of  $\tilde{c}_2^*$  with respect to  $\tilde{c}_1^*$  is given as

$$\mathcal{J}(f(\tilde{c}_1^*), f(\tilde{c}_2^*)) = \min_{\xi_j^* \in \Xi^*} \mathfrak{d}(f(\tilde{c}_1^*), f(\tilde{c}_2^* \circ \xi_j^*)). \quad (4.8)$$

Meanwhile, orientation reversing parametrizations are estimated by generalizing (4.8) to the following objective functional

$$\arg \min(\mathcal{J}(f(\tilde{c}_1^*), f(\overrightarrow{\tilde{c}_2^*})), \mathcal{J}(f(\tilde{c}_1^*), f(\overleftarrow{\tilde{c}_2^*}))), \quad (4.9)$$

where  $\rightarrow$  and  $\leftarrow$ , represent ordering of points in clock-wise and anti-clockwise direction, respectively. Note that if the curves are open curves, then optimal parametrization can be estimated by testing only the different end points as a starting point. In this thesis, we estimate the solution of (4.9) by iteratively testing every starting point in a nested loop. Thus, the time complexity of aligning closed curves is  $\mathcal{O}(z^2)$ . Moreover, it is worth noting that if  $f(\vec{c}_2^*) = (g_1, \dots, g_z)$  then  $f(\overleftarrow{c}_2^*) = (g_z^{-1}, \dots, g_1^{-1})$ .

Nevertheless, (4.9) is based on the assumption that the unknown parametrizations of the curves are of the type (4.3). However, occlusion and large nonlinear deformation of curved shapes can not be captured by linear reparametrizations alone, see Figure 4.1. Subsection 4.4 discusses a generalization of (4.9) to nonlinear reparametrizations.

## 4.4 General case

Contrary to (4.3), consider parametrization of  $I$  by any nonlinear parametrization  $\beta$ . In general, the family of unrestricted nonlinear parametrizations is inclusive of the ones that define curve samplers that do not preserve shape, see Figure 3.6. Consequently, in this thesis, instead of carefully defining nonlinear parametrizations that will lead to samplers that preserve shape, we indirectly enforce a constraint by defining the family of samplers due to nonlinear parametrizations as

$$\Xi^A = \{\xi_i \mid A(c_i \circ \xi_i) = A(c_i \circ \xi^*)\}, \quad (4.10)$$

where  $A(\cdot)$  is function that computes the area of a given curve in  $\mathbf{R}^2$ . In general, the area of a closed region  $c_1$  is described, using Green's theorem, as

$$\iint_{c_1} dx dy = \oint_{c_1} P dx + Q dy, \quad (4.11)$$

such that  $\frac{\partial Q}{\partial x} - \frac{\partial P}{\partial y} = 1$ . Hence, for a continuous curve  $c_i$  approximated by  $\tilde{c}_i = (p_1, \dots, p_z)$  we define its area, by setting  $\frac{\partial P}{\partial y} = 0$ , as

$$\oint_{c_1} x dy \sim A(\tilde{c}_i) = \frac{1}{2} \sum_{j=1}^z (p_j^y - p_{(i+1) \bmod z}^y)(p_j^x - p_{(j+1) \bmod z}^x), \quad (4.12)$$

## 4. OPTIMAL PARAMETRIZATION

---

where  $p_j^x$  and  $p_j^y$  denote the  $x$  and  $y$  coordinate components of the point  $p_j$ . Note that, to be invariant to the orientation of the curve we will always consider the absolute value of (4.12). In case of curves in  $\mathbf{R}^{n>2}$ , alternative geometric moments can be adopted to define (4.10) instead of area, e.g., length of the curves. However, preserving length of a curve is not as strong of a constraint as preserving area of a curve to enforce shape preservation.

Subsequently, the alteration of a given shape  $c_1$ , when sampled by  $\xi_i$ , from a uniformly sampled one is quantified as

$$\begin{aligned} \rho_{c_1}(\xi^*, \xi_i) &= |A(c_1 \circ \xi^*) - A(c_1 \circ \xi^i)| \\ &= \left| \sum_{j=2}^z A_{c_1}(p_{j-1}^*, p_j^*) - \sum_{j=2}^z A_{c_1}(p_{j-1}^i, p_j^i) \right|, \end{aligned} \quad (4.13)$$

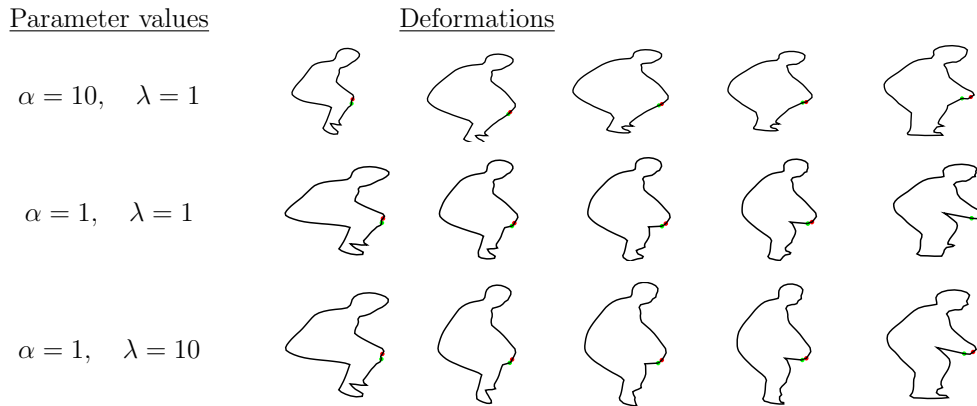
where  $|\cdot|$  denotes the absolute value. Note that  $A_{c_1}(p_{j-1}^*, p_j^*)$  is evaluating (4.12) per sequential points. Hence  $\rho_{c_1}(\cdot, \cdot)$  defines a curve specific function that takes a given sampler  $\xi$  and computes its corresponding penalty term. In other words, (4.13) measures the deviation of a given sampler  $\xi$  from the set  $\Xi^A$  for a specific curve. Finally, we redefine (4.2) with the penalty term as follows

$$\min_{\xi_i, \xi_j} \{ \alpha \times \mathfrak{d}(e, f(c_1 \circ \xi_i)^{-1} f(c_2 \circ \xi_j)) + \lambda \times (\rho_{c_1}(\xi_*, \xi_i) + \rho_{c_2}(\xi_*, \xi_j)) \}. \quad (4.14)$$

We have added weighting factors  $\alpha$  and  $\lambda$  to emphasize the effect of one over the other— assigning a large value for  $\lambda$  leads to an objective functional that favours samplers that preserve area even with a high deformation cost and vice versa for a small  $\lambda$  value, see Figure 4.2. The objective functional given in (4.14) is symmetric. That is, if we switch the arguments ( $c_1$  and  $c_2$ ) the solution will not change. Moreover, its solution can be estimated via dynamic programming in polynomial time, since it can be decomposed into linearly sequential sub-problems [BD15]. In the next Subsection, we will discuss the optimization of (4.14) with dynamic programming.

### 4.4.1 Optimal sampling

We herein describe a dynamic programming based solution for estimating orientation preserving shape parameter for open curves. Later, we will address the case of closed curves.



**Figure 4.2:** Geodesic paths for different values of  $\alpha$  and  $\lambda$ . Note the impact of a large  $\lambda$  value on the deformation (geodesic) and the final result.

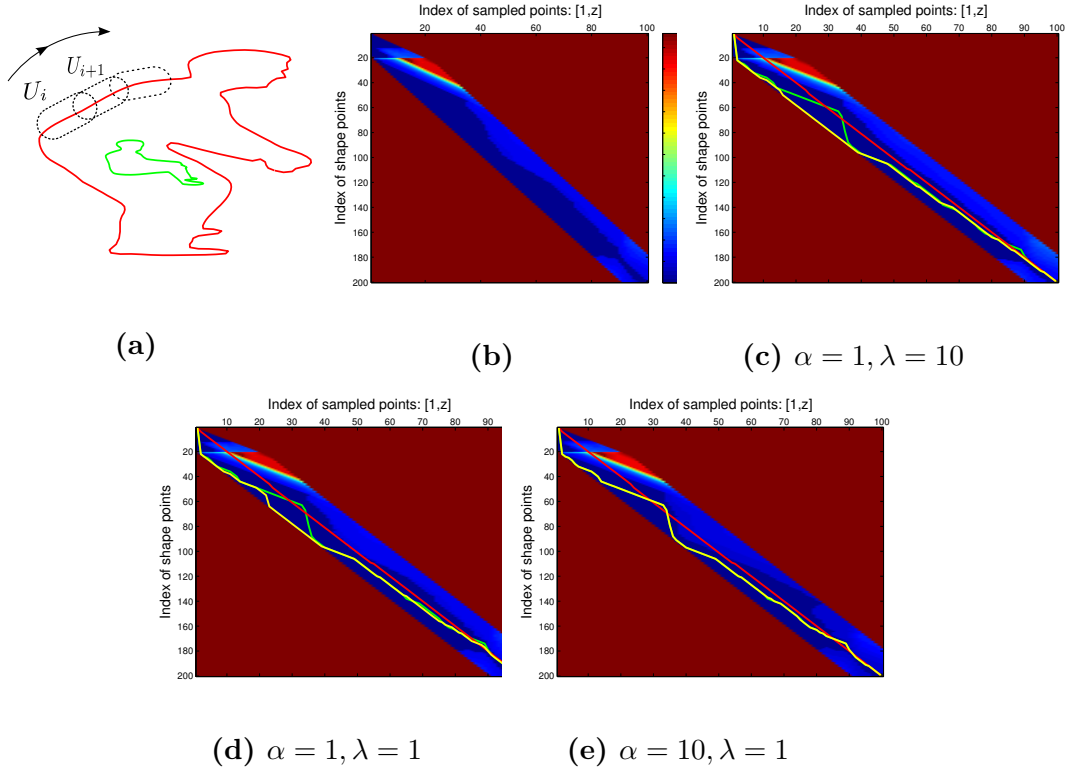
As indicated earlier, the objective functional given in (4.14) can be solved with dynamic programming in a polynomial time. To show this fact, we first describe  $z$  sample points of a continuous shape in a more general form as

$$c_k \circ \xi_j = (x_1, x_2, \dots, x_z) : \quad x_i \in U_i, \quad (4.15)$$

where  $x_{i=1, \dots, z}$  are variables that can take points as a value from their respective domain  $U_{i=1, \dots, z}$ , while the  $U_i \subset c_k$  are compatible charts<sup>1</sup> of the curve  $c_k$ . Hence, the size (cardinality) of  $U_i$  introduces a restriction on the search space of the corresponding  $x_i$ . In this work, we define the charts  $U_{i=1, \dots, z}$  by sliding a pre-defined window size to cover the whole curve, see Figure 4.3a. The ratio of the window sizes, defined on two given shapes, determines the constraint on the elasticity of a deformation from one to another, we denote this ratio by  $\eta$ . For instance, if  $w_1$  and  $w_2$  are the window sizes defined for  $c_1$  and  $c_2$ , respectively, then we approximate the elasticity constraint by  $\eta = w_2/w_1$ , see Figure. 4.4. Subsequently, we first formulate the problem of optimal parameter estimation for one of the argument curves,  $c_2$ , while fixing the sampling of  $c_1$  to uniform point sampling; later, we will described solving (4.14) in its entirety. Although fixing one of the samplings converts what was a symmetric objective functional (4.14) to an asymmetric one, it has its own advantages. The advantages are:































<sup>1</sup>Sets with smooth intersection maps.

## 4. OPTIMAL PARAMETRIZATION



**Figure 4.3:** Intermediate values of dynamic programming. (a) shows two input shapes.  $\tilde{c}_1^*$ , coloured in green, is uniformly sampled and  $c_2$ , coloured in red, is to be sampled optimally. (b) shows the search space, defined by the charts  $U_i$ , and the cost of selecting a point from  $c_2$  for the  $i^{\text{th}}$  position with the color coding. (c-e) shows three optimal sampling paths based on (4.16) for different values of  $\alpha$  and  $\lambda$ . The green path is an optimal for  $\alpha = 1$  and  $\lambda = 0$ , the red path is an optimal for  $\alpha = 0$  and  $\lambda = 1$ , and finally the yellow path is an optimal sampling for  $\alpha$  and  $\lambda$  values shown below the figures.



$\alpha$	$\lambda$	$\eta$	Geodesic				
<b>1</b>	<b>0</b>	20.9151					
		3.6601					
		2.0915					
<b>1</b>	<b>10</b>	20.9151					
		3.6601					
		2.0915					

**Figure 4.4:** Geodesic paths between two shapes under different elasticity constraint  $\eta$ . Note that an objective functional with an appropriate  $\lambda$  value gives a consistent result regardless of the elasticity constraint.

#### 4. OPTIMAL PARAMETRIZATION

---

- Computational efficiency: Solving (4.14) as it is requires  $\mathcal{O}(b^4z)$  time, while fixing one of the arguments require  $\mathcal{O}(b^2z)$  time, where  $b$  is the cardinality of the charts denoted as  $b = |U_i|$ .
- Computational efficiency in processing a group of shapes: When aligning a set of shapes, one often fixes a template shape to which the rest is going to be aligned. In such a case, aligning  $n$  shapes to a fixed reference shape requires  $n - 1$  alignment operations, whereas a direct one-to-one alignment requires  $(n - 1)!$  alignment operations.

Subsequently, let the representation of a uniformly sampled reference curve  $\tilde{c}_1^*$  be  $f(\tilde{c}_1^*) = (g_1^*, \dots, g_z^*)$ . The estimate of the optimal sampler for a given curve  $c_2$ , with respect to  $\tilde{c}_1^*$ , is give by optimizing (4.14), which is rewritten as

$$\arg \min_{\xi_j} \sum_{i=2}^z \phi_i(x_{i-1}, x_i), \quad (4.16)$$

where  $\phi_i$  is defined as

$$\begin{aligned} \phi_i(x_{i-1}, x_i) &= \alpha \times d(g_i^*, g_i)^2 \\ &+ \lambda \times |(A_{c_2}(p_{i-1}^*, p_i^*) - A_{c_2}(x_{i-1}, x_i))|, \end{aligned} \quad (4.17)$$

where  $x_{i-1}$  and  $x_i$  are sampled points of  $c_2$  using a candidate sampler for the  $(i - 1)^{th}$  and  $i^{th}$  position, such that  $g_i x_{i-1} = x_i$ , and  $p_{i-1}^*, p_i^*$  are the sampled points of  $c_2$  for the  $(i - 1)^{th}$  and  $i^{th}$  position when a uniform sampler  $\xi^*$  is used. Moreover, only the penalty term of  $c_2$  is added in (4.17). Since the sampling of  $c_1$  is fixed to uniform sampling, its penalty is zero. Subsequently, (4.16) is identified as the composition of linearly sequential sub-problems. As a result, its solution is estimated by computing the minimum and the minimizer of (4.16). Nevertheless, the solution due to (4.16) is based on the assumption that the starting point and orientation of the curve is already estimated. In case, where neither the starting point nor the orientation of the points are available one can adopt either one of the two possibilities:

1. Solve (4.16) for every possible starting point in both orientations (clockwise and anti-clockwise direction)– in case of open curves possible starting points

are only the two end points, while in closed curves every point is a candidate starting point. Similarly to (4.9) the final solution will be the one that minimizes the cost. The main advantage of this approach is that it gives an optimal result upto the weighting parameters and the size of the search space (feasibility set). However, it is computationally expensive. For  $z$  points it requires  $\mathcal{O}(2s^2z^2)$  for a curve with a fixed template.

2. Estimate the starting point and orientation with linear parametrization assumption (4.9). Once the starting point and orientation of points are estimated, nonlinear parametrizations can be estimated using (4.16). One of the main advantages of this approach is that it is computationally efficient in comparison to the earlier one, requires  $\mathcal{O}(2z + s^2z)$  time. However, the solution will be highly dependent on the optimality of the starting point and orientation estimation. In this thesis, we will mainly follow this approach to estimate nonlinear parametrization of curves with respect to one another.

#### 4.4.2 Dynamic programming

In dynamic programming, a solution for a given cost functional is computed by solving for the minimum and the minimizer of the cost functional's sub-problems, sequentially. In our case, the algorithm begins by initializing the cost of matching the starting points to zero; since we have assumed the starting point and orientation to be already estimated. We denote this as

$$C_1(x_1 = p^s) = 0. \tag{4.18}$$

In other words, the above initialization forces the starting point of the sampling to be fixed to  $p^s$ . Subsequently, the solution space of the cost functional, given in (4.16), is traversed sequentially by exploiting the linearly separable substructure as follows

$$\begin{aligned} C_j(x_j) &= \min_{x_2, \dots, x_j} \sum_{i=2}^j \phi_i(x_{i-1}, x_i), \quad \forall x_j \in U_j \\ &= \min_{x_{j-1}} C_{j-1}(x_{j-1}) + \phi_j(x_{j-1}, x_j), \quad j = 2, \dots, z \end{aligned}$$

## 4. OPTIMAL PARAMETRIZATION

---



---

**Algorithm 1:** Fixing one curve and optimally sampling another

---

**Data:**  $c_1 \circ \xi^*, c_2$

**Initialization:**  $s = |U|, z = |c_1 \circ \xi^*|, m = |c_2|, i = 2,$

$O_1(1) = p_1, O_z(z) = p_z;$

**for**  $i \leq z$  **do**

**for**  $x_j \subset U_j$  **do**

$U_{j-1} = \{p_k \mid \forall p_k : (\ell|_{[0,k]} < \ell|_{[0,j]})\};$

**for**  $x_{j-1} \subset U_{j-1}$  **do**

$C_j(x_j) = \phi_i(x_{j-1}, x_j) + C_{j-1}(x_{j-1});$

**end**

**end**

$C_i(x_j) = \min_{U_j} C_j(x_j);$

$O_i(x_j) = \arg \min_{U_j} C_j(x_j);$

**end**

$k = z - 1;$

**for**  $k > 1$  **do**

$x_k = O_{k+1}(x_{k+1});$

**end**

**Result:**  $c_2 \circ \xi_j = \{x_1, \dots, x_z\}$

---

where  $\phi_i(\cdot, \cdot)$  is as defined in (4.17). Subsequently, the optimizers of the cost functional are given as

$$O_j(x_j) = \arg \min_{x_j} C_j(x_j). \quad (4.19)$$

Furthermore, in order to get a monotonic and injective mapping, we impose the following restriction on the solution space.

- Let  $\ell|_{[0,n]}$  denote the length of a curve segment up to  $p_n$ . Subsequently, if  $x_j = p_n$  then we will only consider  $x_{j-1} = p_k : \ell|_{[0,k]} < \ell|_{[0,n]}$ .

As a result, the solution space is restricted to samplings that are monotonic with a strictly positive derivative with respect to the arc length. Hence the condition ensures an injective mapping. In effect, the point matching is always one-to-one. Note that, in most infinite dimensional optimal parameter estimation there

is a possibility of getting a one-to-many point correspondence, especially if the matching is based on  $L_2$  metric and if there is a high nonlinear local deformation [BBM16]. Moreover, continuity can be formalized in topological terms using the charts of the curve to be sampled. In this work, however, we simply bound the search space of  $x_j$  by a predefined window size. Subsequently, by traversing the solution space and saving the cost of selecting all possible points for the positions along with the optimizers it is possible to build the landscape of the optimization problem, see Figure 4.5. The Final optimal solution, however, is estimated by backtracking the optimal path from the solution of  $x_z$  to the beginning position  $x_1$ , see Algorithm 1.

The pseudocode given in Algorithm 1 can directly be generalized to solve the symmetric version of (4.16) which is given in (4.14). The main advantage of solving (4.16) over (4.14) is that it is a symmetric objective functional, which, in practical terms, means there is no need for the user to choose a shape argument to fix to a uniform sampling. In fact, as described in (4.14) the resulting solution is a pair of optimal samplings for both shape arguments, see Figure. 4.5– to the best of our knowledge, joint optimal parameter estimation is only discussed in [LRK15]. On the contrary, the time complexity of (4.14) is  $\mathcal{O}(s^4z)$  while (4.16) has a  $\mathcal{O}(s^2z)$  time complexity.

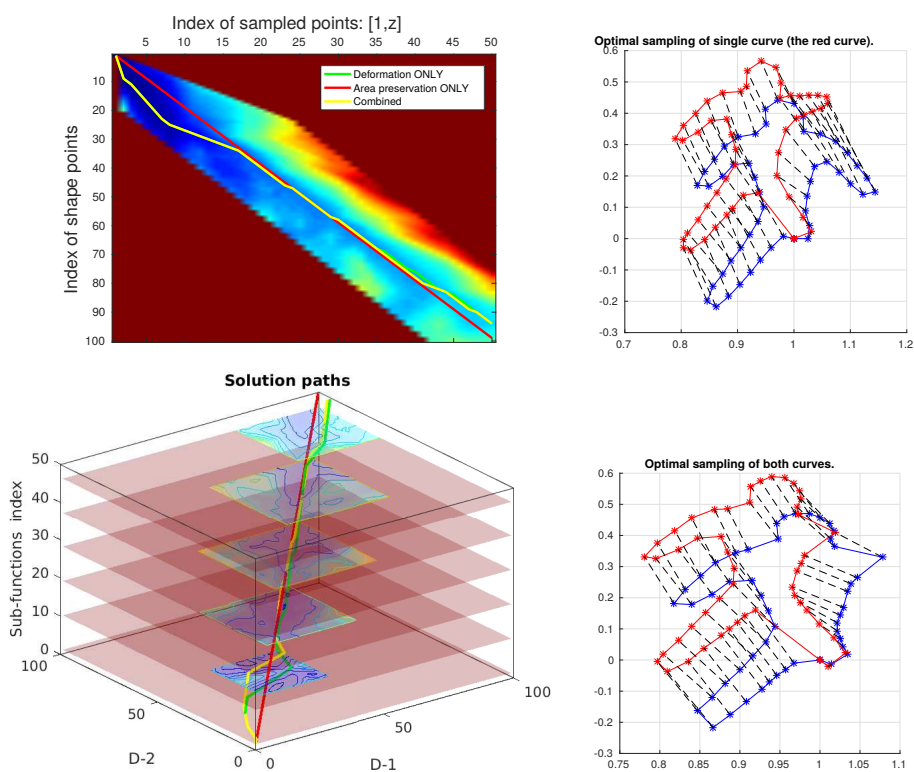
## 4.5 Evaluation

In this section, we will describe the evaluation of the proposed parameter estimation techniques and compare optimal parameter estimation based on linearity assumption with parameter estimation based on area preserving reparametrizations.

### 4.5.1 Shape retrieval

In Chapter 3, we have evaluated the distance metric on a fairly simple dataset where occlusion and nonlinear deformations are minimal. Here, we replicate the shape retrieval experimental scenario with datasets that are composed of shapes with large occlusion, missing parts and nonlinear deformations.

## 4. OPTIMAL PARAMETRIZATION



**Figure 4.5:** Illustration of the symmetric and the asymmetric solutions. Each row shows the solution sampling along with the problem space and the solution paths. The first row shows the solution due to (4.16) where the blue curve is uniformly sampled with  $z = 100$ . The second row shows the optimal sampling of both curves with (4.14) where  $z = 50$ .

**Kimia99, Kimia216, and MPEG-7** [SKK04, LLE00]: Kimia99 is composed of 9 shape classes with 11 examples each, Kimia216 contains 18 class of shapes with 12 examples, and the MPEG-7 dataset contains 70 shape classes each containing 20 elements. All datasets are composed of shape categories with variations due to different viewpoints, occlusions, and/or large deformations. As a result, these datasets pose a significantly more challenging problem than either the Flavia or the Swedish leaf datasets do. We evaluate our approach, on all datasets, by replicating the experimental scenario described in [BL08] and [WBY<sup>+</sup>12], which is similar to the leave-one-out. In the case of Kimia99 and Kimia216 retrieval accuracy is measured by counting the overall results of the top 10 (in Kimia99) and top 11 (in Kimia216) retrievals from the same class, excluding the query shape [WBY<sup>+</sup>12]. Meanwhile, retrieval accuracy on the MPEG-7 dataset is measured based on what is called “Bull’s eye score” – Bull’s eye score takes the overall percentage of retrieval results, among the first 40, that belong to the same class as the query [WBY<sup>+</sup>12]. In all datasets, we evaluate our approach for  $z = 100$  using both linear and area preserving parameter estimations. More importantly, in both cases, matching starting points and shape orientation are estimated based on the linear parameter space estimation (4.9). Furthermore,  $\lambda = 0.1$ ,  $\alpha = 1$ , and  $\eta = 20$  in all of the area preserving parameter estimations. In Table 4.1 and Table 4.2 we compare our result with other approaches evaluated on Kimia99 and Kimia216. On MPEG-7, retrieval based on the uniform sampling approach achieved a bull’s eye score of 68.02%, while retrieval based on the optimal sampling approach achieved 84.17%.

We note that retrieval results based on area preserving reparametrizations achieve a better score than results based on linear parameter estimations, in all three data sets. In general, given a large  $\eta$  and a small  $\lambda$  (relative to  $\alpha$ ) retrieval results based on area preserving parameter estimation should perform better than retrieval based on a uniform sampling. Nevertheless, in this case the reported area-preserving parameter estimations are based on estimates of the starting point and the shape orientation alignment using (4.9). In most cases, e.g., Kimia99 and Kimia216, such an approach is adequate for reasonable point matching. However, in cases where the inter and inner class variations are both due to large nonlinear deformations the linear parameter estimation is not

## 4. OPTIMAL PARAMETRIZATION

---

Methods	1st	2nd	3rd	4th	5th	6th	7th	8th	9th	10th
SC [BMP02]	97	91	88	85	84	77	75	66	56	37
Shock graph [SKK04]	99	99	99	98	98	97	96	95	93	82
Height functions [WBY+12]	99	99	99	99	98	99	99	96	95	88
Symbolic representation [DT08]	99	99	99	98	99	98	98	95	96	94
<b>Our method with optimal sampling</b>	99	99	97	97	97	98	93	90	79	53
<b>Our method with uniform sampling</b>	91	81	73	75	63	57	51	44	35	30

**Table 4.1:** Retrieval results on Kimia99 shape dataset.

adequate for starting point and orientation estimation, e.g., MPEG-7 data set. Hence, the area preserving parametrization inherits misalignment errors from the linear parameter estimation, especially in cases of large deformations, which lead to performance degradation.

The estimation of both linear and area preserving curve parameters is performed using MATLAB, on a hardware platform as described in Section 3.5.3. Subsequently, for curved shapes that are approximated by  $z = 100$ , the linear parameter estimation takes 3.579600 seconds. Meanwhile the area preserving computation, including the starting point estimation, takes 6.946829 seconds when one curve is fixed as a template and the other is sampled—sampling both curves, simultaneously, takes 3.4480 minutes. In both cases of the area preserving sampling  $\eta = 10$ . Note that, in all of the above cases, the computational time includes rotational alignment but excludes the computation involved with representation of the curves, (3.10). The same matching problem is estimated with a C++ implementation of SRV framework [SKJJ11] running on the same machine in 6.894110 seconds. Hence, the matching speed of the SRV based framework is comparable with the linear parameter estimation, while the area preserving sampling is computational expensive in comparison with both SRV framework and linear parameter estimation.

### 4.5.2 Robustness

In this subsection, we compare the linear parameter estimation presented in Section 4.3, and the area preserving parametrizations given in Section 4.4, qualitatively. To that end, in addition to the already mentioned datasets, we use



## 4.5 Evaluation

Methods	1st	2nd	3rd	4th	5th	6th	7th	8th	9th	10th	11th
SC [BMP02]	214	209	205	197	191	178	161	144	131	101	78
Shock graph [SKK04]	216	216	216	215	210	210	207	204	200	187	163
Skeleton graph matching [BL08]	216	216	215	216	213	210	210	207	205	191	177
<b>Our method with optimal sampling</b>	216	216	208	205	203	193	192	178	169	162	123
<b>Our method with uniform sampling</b>	212	210	188	181	174	165	159	151	141	132	120

**Table 4.2:** Retrieval results on Kimia216 shape dataset.

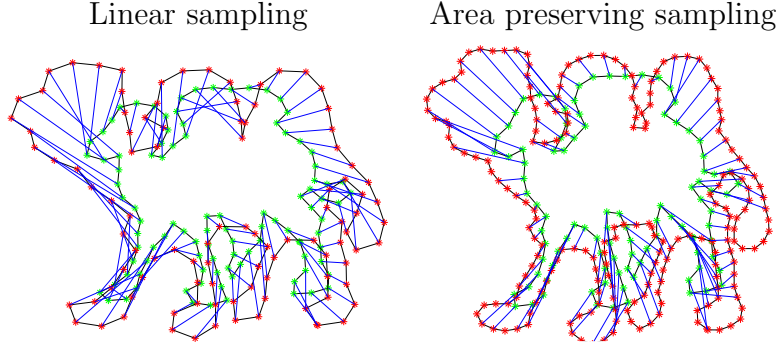
Mythological creatures [BBBK08], and 1070-Shape dataset [K10].

**Occlusion:** Shapes can exhibit partial similarity due to occlusions or nonlinear local deformations, see Figure. 4.9. Thus, a point correspondence estimation method has to be able to handle nonlinear local deformations. For instance, the linear parametrization, discussed in Subsection 4.3, is constrained to linear reparametrization, thus fails to match shapes that exhibit nonlinear local deformations, see Figure 4.6. Alternatively, the area preserving parametrization can be tuned to perform well by adjusting  $\lambda$  in (4.16). To illustrate this, we select example shapes with partial similarity from *Mythological creatures* and *1070-shape* datasets. Subsequently, in Figure. 4.9 we show matching results for different values of  $\lambda$ . In all the matching scenarios, the elasticity constraint  $\eta$  is held constant to 37.46. Hence, the solution search space is fixed. Nevertheless, in case of occlusion or partial matching, we note that forcing the sampler to preserve area when that is clearly not the case leads to point mismatching, hence, a small value should be assigned to  $\lambda$ .

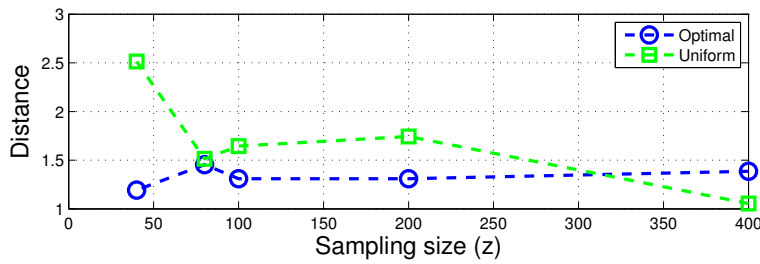
**Effects of sampling size:** The proposed approach casts the point correspondence estimation problem as a sampling function estimation problem, hence the sampling function is optimized and not given. However, the number of points  $z$  is selected a priori. Here, we investigate the impact of having different values of  $z$  on the optimal point correspondence estimation and its accuracy. In general, approximating a shape, especially one with intricate structure, by a small number of points leads to a less detailed result. Nevertheless, it is desirable to have a consistent distance for different sample sizes. To that end, Figure 4.7 shows how the distance between two open curves, shown in Figure 4.8, varies under differ-

## 4. OPTIMAL PARAMETRIZATION

---



**Figure 4.6:** Linear vs area preserving sampling. For the area preserving case, the green curve is sampled uniformly, and the curve in red is sampled with area preserving parametrization. Note that the area preserving sampling adjusts the sampling rate while the linear parameter sampling does not.

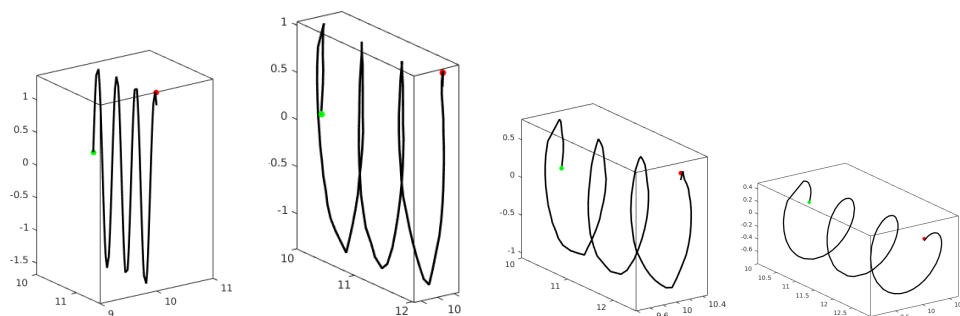


**Figure 4.7:** Impact of sampling size under the optimal and the uniform sampling based point correspondence estimation.

ent sampling sizes for both the linear and area preserving parameter estimations. The result is obtained by varying  $z$  while the rest of the parameters are fixed  $\alpha = 1$ ,  $\lambda = 10$ . We observe that the distance under the optimal sampling is more consistent relative to the uniform sampling based distance.

## 4.6 Conclusion

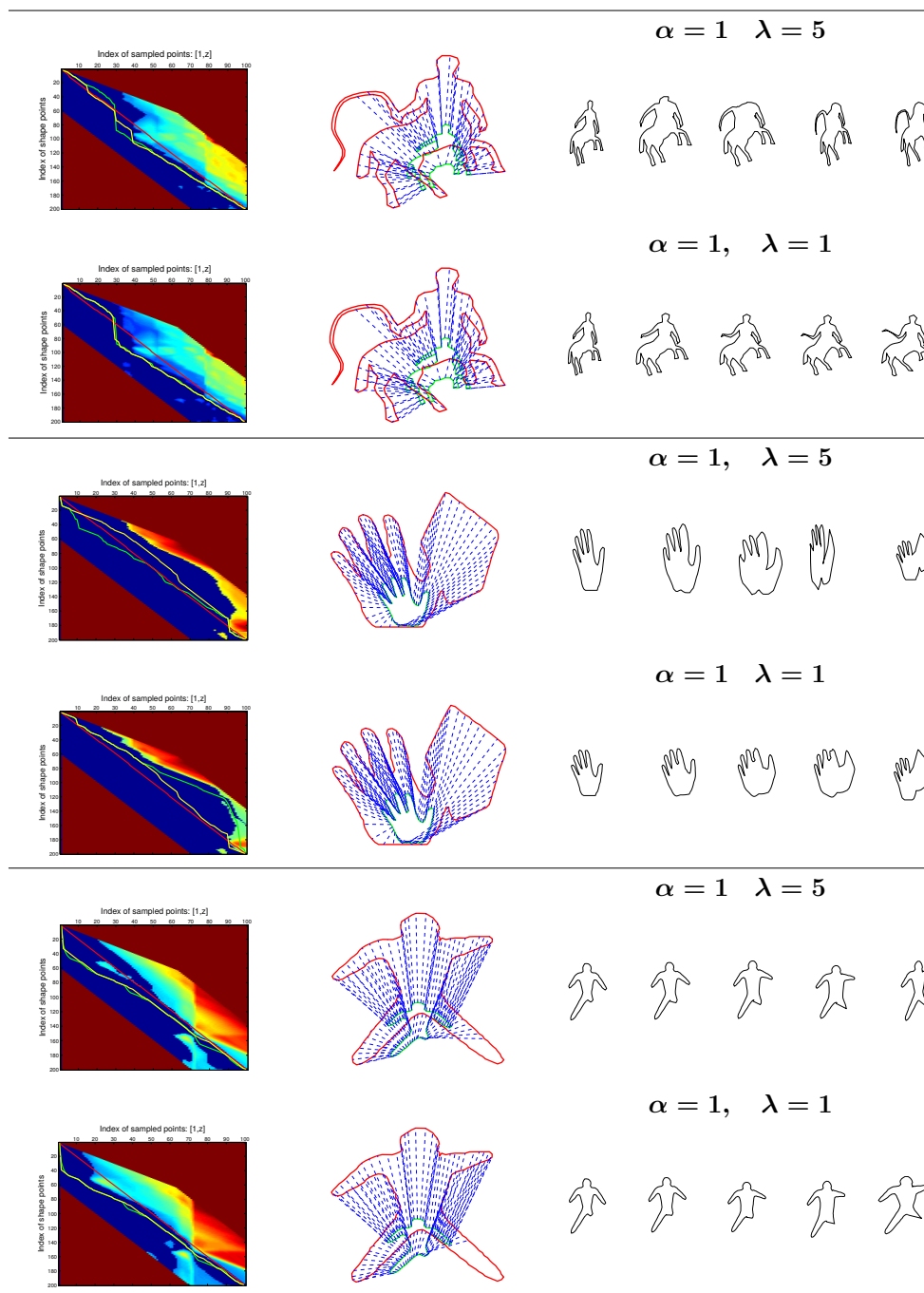
In Chapter 4, we proposed two approaches for the estimation of curved shape parametrization or point correspondence. The first approach assumes a linear reparametrization of a given curve with respect to another. As a result, the family of possible parametrizations is defined as  $z$ -cyclic permutation of the curved shape



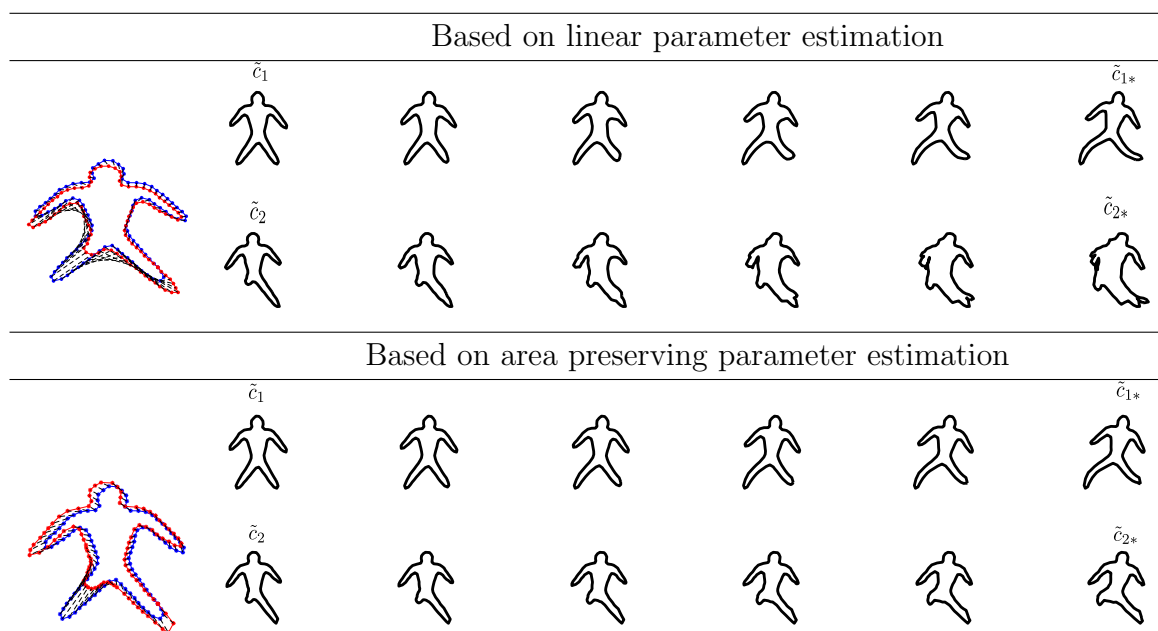
**Figure 4.8:** A geodesic path from the first to the last curve. These curves are used in estimating the effects of sample size.

representations. Although computationally appealing, linear reparametrizations are inadequate when it comes to nonlinear reparametrizations which are manifested as occlusion or local elastic deformation of curved shapes. As a result, we proposed an optimal parameter estimation approach that considers a larger family of reparametrizations than linear–area preserving reparametrizations. Subsequently, we have shown, both qualitatively and quantitatively, that area preserving reparametrizations provide as good or better solution in most cases than linear reparametrizations. Furthermore, the estimated point correspondence are one-to-one in both cases mainly because of the data structuring which insists on the samplers to be injective and monotonic with strictly positive derivative.

## 4. OPTIMAL PARAMETRIZATION



**Figure 4.9:** Impact of  $\lambda$  on partial shape matching. In all of the above examples,  $\eta = 37.46$ . Each row shows optimal sampling of a shape, shown in red, to match the uniformly sampled shape shown in green. Note that, a large value of  $\lambda$  leads to a matching that favours area preservation.



**Figure 4.10:** Illustration of deformation transporting in linear vs area preserving parameter estimation.

#### 4. OPTIMAL PARAMETRIZATION

---

# Chapter 5

## Statistical curved shape analysis

### 5.1 Motivation

One of the main advantages of a stable and efficient representation is that it lays the foundation to explain a shape dataset with statistical models. In particular, a computationally efficient distance function is very important to compute statistical moments of a shape distribution. In Chapter 5, we will adapt different statistical models to estimate distributions of a shape dataset represented with the proposed approach (3.14) in Chapter 3.

We start by formalizing the notion of a random variable and a probability distribution in  $SE(n)$  and its product group  $SE(n)^z$  in Section 5.2. In Section 5.3, we formalize the notion of expectation and covariance of a shape distribution. Section 5.4 presents different parametric models for estimating the distribution of a shape dataset represented with the proposed approach. In Section 5.5 models with latent variables are discussed. Several of the discussed parametric models with or without latent variables are evaluated for object in object category recognition in Section 5.6. The chapter concludes with a summary in Section 5.7.

### 5.2 Background

In general, a random variable can be defined as a function on some data generating space  $\Omega$  that takes its values from a sample space  $M$ , which formally is defined

## 5. STATISTICAL CURVED SHAPE ANALYSIS

---

as

$$X : \Omega \rightarrow M. \quad (5.1)$$

Different random variables can be defined on the same data generating space with potentially different sample spaces. Such random variables can be aggregated, using direct product, into a single function known as random vector which we denote with a bold font  $\mathbf{X}$ . A  $z$ -dimensional random vector is defined as

$$\mathbf{X} : \Omega \rightarrow \prod_{i=1}^z M_i. \quad (5.2)$$

The notion of measure (distance, volume, area, probability, etc.) on both  $\Omega$  and  $M$  can be formalized by defining an admissible subsets of both spaces; such a subset is known as  $\sigma$ -field [Dur10]. Consequently, a given space with a  $\sigma$ -field  $(\Omega, \mathcal{F})$  is called measurable space. In this thesis, we are interested in random variables and vectors that take value in  $SE(n)$  and  $SE(n)^z$ , respectively. As a result, we define our  $\sigma$ -field based on the topology of  $SE(n)$ , and we use (3.47) as a measure on the  $\sigma$ -field of  $SE(n)$ — if a give measure in  $(\Omega, \mathcal{F})$  is finite then it can be made into probability measure by normalizing it. In such a case,  $(\Omega, \mathcal{F})$ , with the probability measure, is called probability measure space and elements of  $\mathcal{F}$  are called events. Subsequently, the probability distribution of a random variable that take its value in  $SE(n)$  is defined as follows.

**Definition 2.** *Let  $(\Omega, \mathcal{F}, \varepsilon)$  be a data generating probability measure space, let  $X$  be a random variable taking its value from a measure space  $(SE(n), \mathcal{B}, m)$ , then the probability distribution of  $X$  is defined as  $P_X(x \in B) = \varepsilon(X^{-1}(x))$ .*

Note that  $m$  represents the distance function we defined earlier on the tangent space (3.47). Furthermore, if  $P_X$  is absolutely continuous with respect to  $m$ , i.e.,  $P_X \ll m$ , then there exists an integrable function  $p$  such that

$$P_X(x \in B) = \int_x p \, dm. \quad (5.3)$$

We will refer to  $p$  as a probability density function (pdf) of the random variable  $X$ — it is sometimes referred to as Radon-Nikodym derivative of  $P_X$  with respect to  $m$ . Now that we have defined random variables and vectors, we can interpret



the representation of a given curve  $f(\tilde{c}_1)$  as an instance of a random vector  $\mathbf{X}$  defined as

$$\mathbf{X} : (\Omega, \mathcal{F}, \varepsilon) \rightarrow \prod_{i=1}^z (\text{SE}(n), \mathcal{B}, m)_i, \quad (5.4)$$

where  $\varepsilon$  is a probability measure in the data generating space. The distribution of (5.4) is given as

$$\int \cdots \int_{\mathbf{x}} p(\mathbf{x}) dm \cdots dm. \quad (5.5)$$

Chapter 5 is dedicated to the modelling and estimation of (5.5) type probability distributions from sample shape data set  $\{f(\tilde{c}_1), \cdots, f(\tilde{c}_q)\}$ . In thesis, the sample shapes are assumed to be identically distributed and independently generated (i.i.d). Although we do not discuss interpretation and modelling philosophy, most of our modelling approaches are based on what is called frequentist interpretation, for alternative modelling approach see [Jay03].

## 5.3 Statistics of shapes

Let  $\mathbf{X}$  be a random vector of the type defined in (5.4). We can then compute what we expect to see from  $\mathbf{X}$  as

$$E(\mathbf{X}) = \int \mathbf{X} d\varepsilon = \int \mathbf{X} p dm, \quad (5.6)$$

where  $p$  is the pdf of  $\mathbf{X}$ . The degree of variations from the expected value is quantified by the covariance matrix, which is written as

$$\text{cov}(\mathbf{X}) = E(\mathbf{X} - E(\mathbf{X}))^2. \quad (5.7)$$

However, in most cases there is no way to know the analytical form of  $\mathbf{X}$ , hence both (5.6) and (5.7) are estimated from sampled data points of  $\mathbf{X}$ . To that end, consider a set of  $q$  sample shape representations  $\{f(\tilde{c}_1), \cdots, f(\tilde{c}_q)\}$  that are i.i.d and generated by  $\mathbf{X}$ , where  $f(\cdot)$  is as defined in (3.14) in Chapter 3. Below we define the mean and covariance of a shape distribution.

## 5. STATISTICAL CURVED SHAPE ANALYSIS

---

**Mean shape:** Given the above sample shape points, we can estimate (5.6) as the sample mean

$$E(\mathbf{X}) \sim \sum_{i=1}^q \frac{1}{q} f(\tilde{c}_i) \quad (5.8)$$

The estimator (5.8) is unbiased since it converges to the true expectation as the number of sample shape points goes to infinity, due to law of large numbers [Dur10]. However, we can not directly use (5.8) since our representation space  $\text{SE}(n)^z$  is not a vector space and is not closed under addition. Instead, we replace (5.8) by what is known as Fréchet mean [Pen06, Kar77]. Fréchet mean is a generalization of a distribution's centroid in any metric space  $\mathcal{U}$ ; it is defined as

$$\min_{\tilde{x} \in \mathcal{U}} \sum_{y \in \mathcal{U}} \delta(\tilde{x}, y)^2, \quad (5.9)$$

Equation (5.9) translates to our representation space as

$$\bar{f}(\tilde{c}) = \arg \min_{\bar{f}(\tilde{c})} \sum_{i=1}^q \mathfrak{d}(f(\tilde{c}_i), \bar{f}(\tilde{c}))^2 \quad (5.10)$$

$$= \arg \min_{\bar{f}(\tilde{c})} \sum_{i=1}^q \sum_{j=1}^z \|\log(\bar{R}_j^T R_{ji})\|_F^2 + \sum_{i=1}^q \sum_{j=1}^z \|v_{ji} - \bar{v}_j\|_2^2, \quad (5.11)$$

where  $\bar{f}(\tilde{c})$  denotes the mean shape representation. The second term of (5.11) can be satisfied by taking the arithmetic mean of the translation vectors since the vectors are elements of a linear space. The first term, however, involves an optimization in the  $\text{SO}(n)$  group which is not a linear space. Consequently, over the past years, considerable work has been done on characterizing the mean of several matrices [ALM04, BMP10, HCSV13]. In this thesis, we will consider the local optimizers of rotation matrices known as Karcher mean [Kar77]. The Karcher mean  $\bar{R}$  of a given  $q$  rotation matrices is characterized by a zero gradient of (5.9) with respect to  $\bar{R}$ – it is written as

$$\sum_{i=1}^q w_i \bar{R} \log(R_i^T \bar{R}) = 0, \quad (5.12)$$

where  $w_i$  are probabilities of each observation. Since we do not know the analytical form of the data generator we will assign  $1/q$  for each  $w_i$ . The gradient given by (5.12) is locally well defined and is unique for localized enough sample points, see [Ken90]. Hence, given an initial guess of the Karcher mean we can iteratively follow the opposite direction of (5.12) until we reach a critical point. For alternative optimization approaches see [AMS09]. To that end, we initialize the iterative process using the approach described in [HCSV13], which is given by the following iterative equations

$$\begin{aligned}\bar{R}_1 &= R_1 \\ \bar{R}_{i+1} &= \bar{R}_i(\bar{R}_i^{-1}R_{i+1})^{1/(i+1)}.\end{aligned}\tag{5.13}$$

Subsequently, the final estimate of the mean rotation matrix, for i.i.d. sam-

---

**Algorithm 2:** Estimation of the Karcher mean in  $SO(n)$

---

**Data:**  $\{R_1, \dots, R_m\}$

**Initialization:**  $thresh = val_1, Lrate = val_2, \bar{R} = (5.13),$

$\nabla_{\bar{R}} = \sum_{i=1}^m w_i \bar{R} \log(R_i^T \bar{R});$

**while**  $\|\nabla_{\bar{R}}\| > thresh$  **do**

$\bar{R} = \bar{R} \exp(-Lrate \times \nabla_{\bar{R}})$   
 $\nabla_{\bar{R}} = \sum_{i=1}^m w_i \bar{R} \log(R_i^T \bar{R})$

**end**

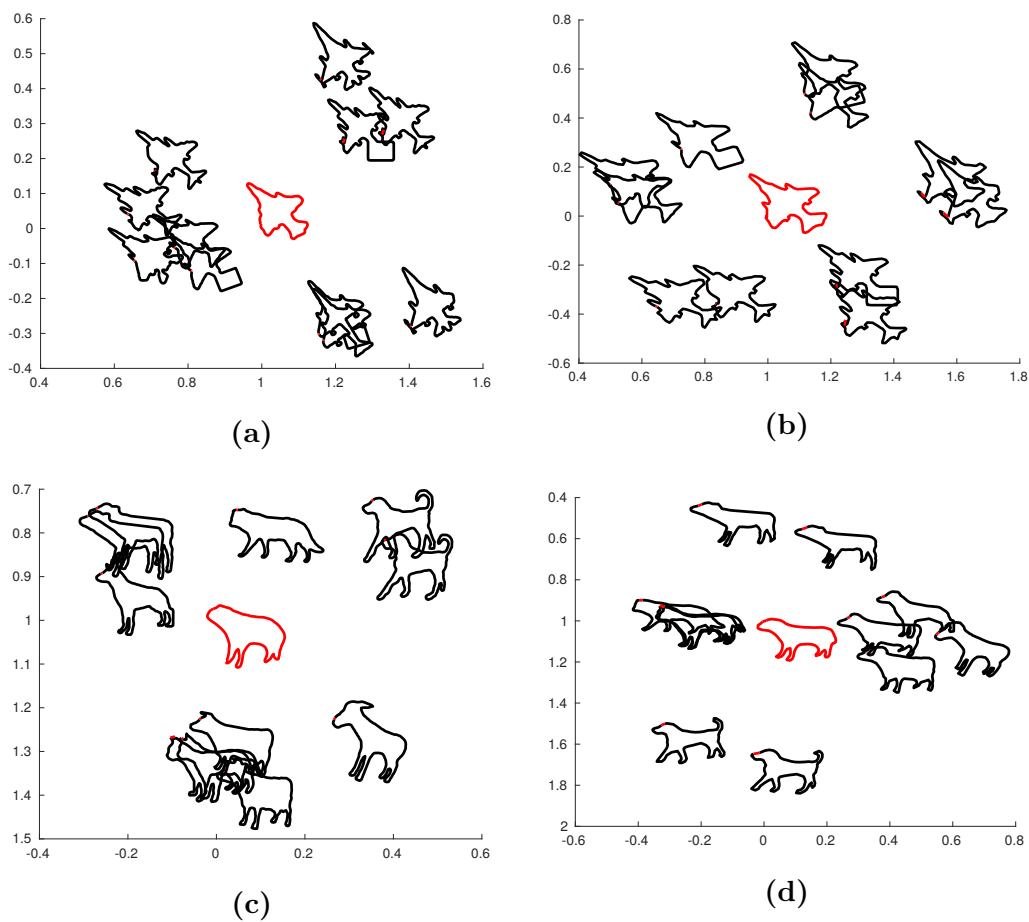
---

ple points of a random variable, is estimated iteratively from the initial estimate (5.13). Algorithm 2 summarizes the procedure. Subsequently, a simple generalization, a direct product of Algorithm 2, is used to estimate the mean shape representation of a given shape sample dataset  $\{f(\tilde{c}_1), \dots, f(\tilde{c}_q)\}$ . Note that the mean shape estimate is based on the assumption that variation due to rotation, translation, and parametrization are filtered out from the shape data set. See Figure 5.1.

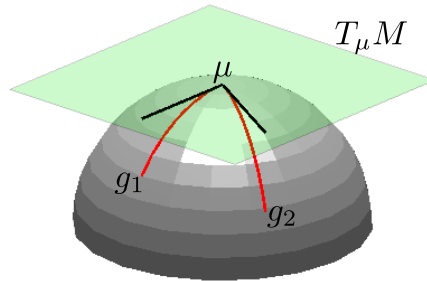
**Covariance:** Similar to the mean shape estimation, we can estimate the covari-

## 5. STATISTICAL CURVED SHAPE ANALYSIS

---



**Figure 5.1:** Mean computation of shapes visualized using Multidimensional scaling [BG05]. The shapes depicted in red are the computed mean of each dataset. (a) and (c) are mean shapes when the point correspondence estimation is based on linear parameter estimation. (b) and (d) are the mean shapes computed from shapes dataset with area preserving parameter estimation, respectively.



**Figure 5.2:** Illustration of the tangent space at the mean shape, where the covariance matrix of a linearized data points are computed.

ance of a shape representation dataset as

$$\begin{aligned} cov(\mathbf{X}) &\sim \frac{1}{q} \sum_{i=1}^z (\bar{f}(\tilde{c}) - f(\tilde{c}_i))^2 \\ &\quad \frac{1}{q} \sum_{i=1}^z \mathfrak{d}(\bar{f}(\tilde{c}), f(\tilde{c}_i))^2, \end{aligned} \tag{5.14}$$

where  $f(\cdot)$  is as given in (3.14) and  $\mathfrak{d}(\cdot, \cdot)$  as (3.48) in Chapter 3. The above equation for the covariance matrix is the same as the (5.11) except that in (5.14) the mean is already estimated. Figure (5.2) shows an illustration of the covariance matrix computation at the tangent space of the mean.

## 5.4 Modelling shape distribution

There are several ways one could approach the estimation of a probability distribution for a random vector of type (5.4). The approaches are mainly guided by the assumptions imposed on the distributions:

1. Parametric: Given a sample dataset, parametric approach assumes the distribution of a random vector  $\mathbf{X}$  to take the form of a parametrized function, e.g., exponential distribution families. As such, in parametric distribution the effort is to correctly estimate the parameters of the chosen distribution form. We herein explore Gaussian and mixture of Gaussians for modelling shape distributions.

## 5. STATISTICAL CURVED SHAPE ANALYSIS

---

2. Non-parametric: An alternative to parametric modelling is to directly attempt to estimate the analytical form of the distribution from the dataset points. Such kind of approaches are called non-parametric estimation, e.g., histograms, Parzen-window (Kernel) density estimation, neural networks, etc. Non-parametric approaches are usually effective when the available dataset is large.

In this section, we will mainly cover parametric density estimation approaches for a data set of shape samples represented with the proposed method (3.14).

### 5.4.1 Parametric density estimation

Most existing statistical shape analysis approaches are based on parametric density estimation or latent variable modelling [DM98]. In particular, in [DAO15] an inhomogeneous time Markov process is used to capture the statistical properties of a deformable shape represented in a matrix Lie group. Similarly, here we consider a curve representation  $f(\bar{c})$  as a realization of a random vector  $\mathbf{X}$ .

**Gaussian distribution:** A Gaussian distribution of a random vector  $\mathbf{X}$  is completely determined by two parameters, *mean*  $\mu$  and *covariance*  $\Sigma$ . Hence the evaluation of a  $D$ -dimensional observation  $\mathbf{X} = \mathbf{x}$  with Gaussian pdf is written as

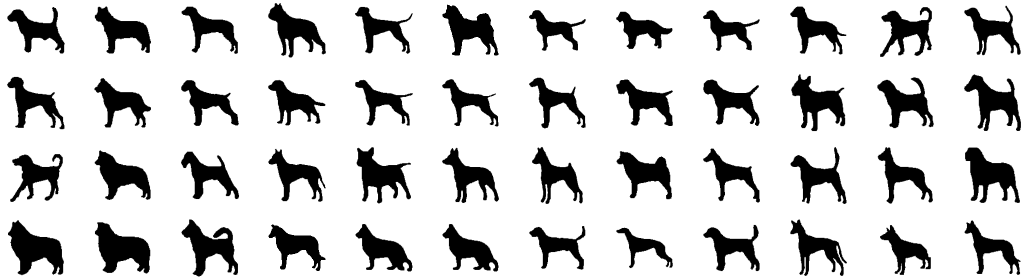
$$\mathcal{N}(\mathbf{x}; \mu, \Sigma) = \frac{1}{(2\pi)^{D/2} |\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x} - \mu)^T \Sigma^{-1}(\mathbf{x} - \mu)\right). \quad (5.15)$$

Refer to [GN99] for further details in matrix valued Gaussian random variables. In our case, however, if we assume that our shape dataset follows (5.15), the density estimation simply means the correct estimation of (5.11) and (5.14). Hence (5.15) amounts to the following in the defined representation space

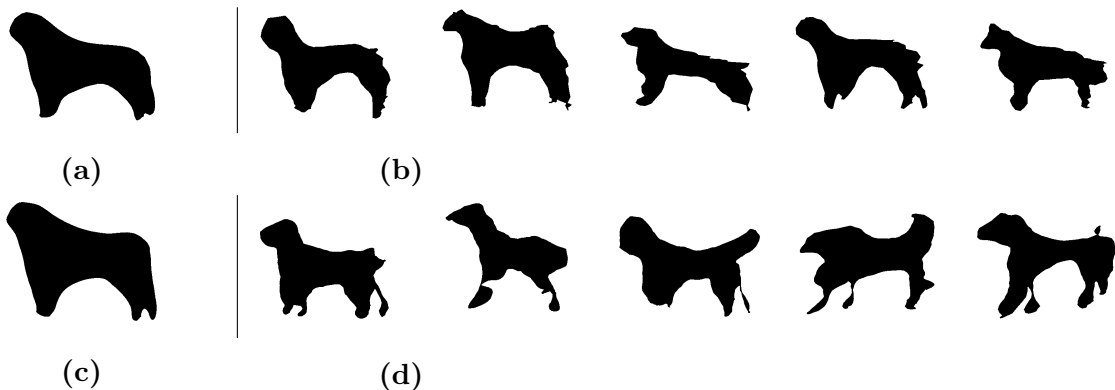
$$\frac{1}{(2\pi)^{D/2} |\Sigma|^{1/2}} \exp\left(-\frac{1}{2}\mathbf{v}^T \Sigma^{-1} \mathbf{v}\right), \quad (5.16)$$

where  $\mathbf{v}$  is the vectorized form of an observation at tangent space of the mean. We write  $\mathbf{v}$ , using  $vec(\cdot)$  operator to vectorize a matrix, as follows

$$\mathbf{v} = \prod_{i=1}^z vec \begin{pmatrix} \log(\bar{R}_i^T R_i) & \bar{R}_i^T \bar{v}_i - v_i \\ 0 & 0 \end{pmatrix}. \quad (5.17)$$



**Figure 5.3:** Example dataset of dogs from Kimia 1070-shape dataset.



**Figure 5.4:** Mean shapes of the dogs dataset along with a randomly sampled shapes. (a) Mean shape based on area preserving parameter estimation, (b) randomly sampled shapes from a Gaussian distribution of shapes with area preserving parameter estimation. (c) Mean shape based on linear parameter estimation, (d) randomly sampled shapes from a Gaussian distribution of shapes with linear parameter estimation.

One can further sample the estimated distribution using one of the well studied random samplers [Mad02]. However, since the sampled vector is going to be an element of the tangent vector it requires exponentiation to finally get the randomly sampled shape representation, see Appendix A. Hence, for a given Gaussian distribution sampler  $S(\mu, \Sigma)$ , we can generate a random shape as

$$f_r = \bar{f}(\tilde{c}) \exp_{\bar{f}(\tilde{c})}(S(0, \Sigma)), \quad (5.18)$$

where  $f_r$  represents the randomly sampled representation, see Figure 5.4.

**Shape as a stochastic process:** In cases where a spatial relationship among

## 5. STATISTICAL CURVED SHAPE ANALYSIS

---

random variables exists, a much more natural approach is to consider graphical models for modelling a distribution [Bes75]. As such, a given random vector  $\mathbf{X}$  can be seen as spatially ordered set of random variables. Let the probability of a given random vector  $\mathbf{X} = \{X_1, \dots, X_z\}$  to take the value  $f(\tilde{c}) = \{g_1, \dots, g_z\}$  to be expressed as

$$p(X_1 = g_1, \dots, X_z = g_z), \quad (5.19)$$

Regardless of what form  $p(\cdot)$  takes, one can factor (5.19) using the chain rule, which follows directly from the axioms of a probability measure, as follows

$$p(g_1, \dots, g_z) = p(g_z | g_{z-1}, \dots, g_1) \cdots p(g_2 | g_1) p(g_1). \quad (5.20)$$

Since the observed transformation matrices are indexed in an order, one can interpret (5.20) as a model of a stochastic process. Moreover, additional assumptions can be imposed on the dependency of the sequential transformation matrices. For instance, assume that (5.20) is a Markov chain<sup>1</sup> then (5.20) is simplified as

$$p(g_1, \dots, g_z) = \prod_{i=1}^{z-1} p(g_{i+1} | g_i). \quad (5.21)$$

In such a case, we model (5.21) as a non-stationary Markov process with the following transition rule

$$g_{i+1} = \hat{g}_{i+1} \times g_i. \quad (5.22)$$

Note that the  $\hat{\cdot}$  denotes the observed transition matrix. Hence, the probability of a given shape representation  $f(\tilde{c}_j) = \{g_1, \dots, g_z\}$  is as follows

$$p(g_1, \dots, g_z) = \prod_{i=1}^{z-1} p(g_{i+1} | g_i) = \prod_{i=1}^{z-1} p(\hat{g}_i) \quad (5.23)$$

Subsequently, it is possible to model the distribution of the transition matrices with a Gaussian distribution.

---

<sup>1</sup>A stochastic process that depends only on the current state to predict the future.



## 5.5 Latent variable estimation

As described in Section 5.4, one can estimate the distribution of a random vector  $\mathbf{X}$  that takes a shape representation as its value. However, in cases where the distribution of the dataset can not be adequately explained with simple distribution forms, it is practical to estimate the distribution by introducing unobserved variable, which is referred as latent variable in the machine learning literature [Bis06, ZJ09, TB99]. Section 5.5 details different latent variable models in the proposed shape representation space.

### 5.5.1 Dimension reduction

Consider a class specific shape representation dataset  $Q = \{f(\tilde{c}_1), \dots, f(\tilde{c}_q)\}$ . As described in Chapter 3, the deformation space is the whole group  $SE(n)^z/SO(n)$ . However, we expect the admissible variations in  $Q$  to be a submanifold of the original space if not a subgroup, since the shapes in  $Q$  are from the same category. In such a case, we estimate the unobserved submanifold or subspace from  $Q$  by estimating the latent variables that describe the subspace or submanifold. In the last two decades, several methods that employ both linear and nonlinear dimension reduction have been introduced [TDSL00, RS00, HSH14]. In this subsection, we will depend on the geodesic curves (3.49) to linearize the dataset and estimate a linear subspace using Principal Component Analysis (PCA) [Jol86].

**PCA model:** The main idea behind PCA is to describe an observed dataset with a linear combination of uncorrelated vectors such that the variance of the dataset in each of the vector direction is maximized. More concretely, we are optimizing the following objective function

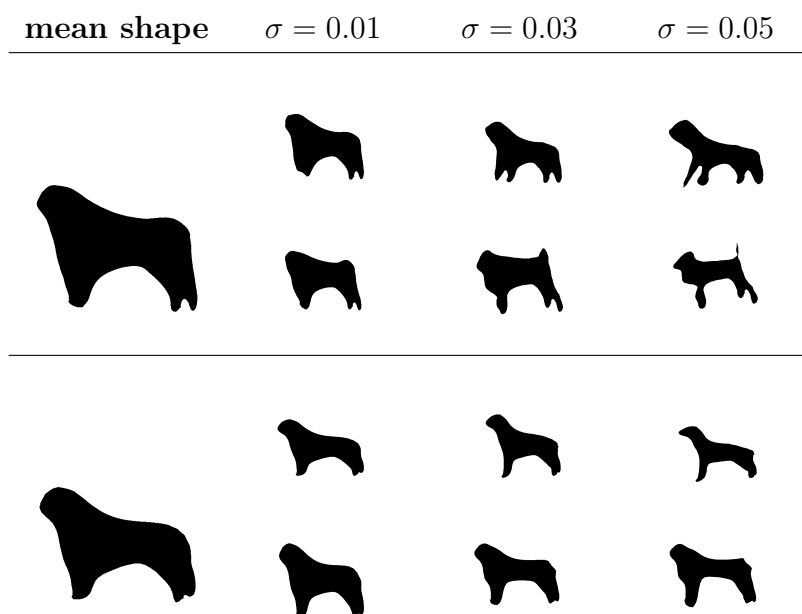
$$\max_{\mathbf{v}} E(Q\mathbf{v} - E(Q\mathbf{v}))^2, \quad (5.24)$$

where  $Q$  is the dataset and  $\mathbf{v}$  is the sought after vector direction that maximizes the variation of the dataset. If we assume the dataset has zero as its mean then (5.24) becomes

$$\max_{\mathbf{v}} E(Q\mathbf{v})^2. \quad (5.25)$$

## 5. STATISTICAL CURVED SHAPE ANALYSIS

---



**Figure 5.5:** Shapes in two principal directions on the dogs dataset. The first column shows the mean shape, computed from the dataset with parameter estimation based on linearity assumption and area preserving assumption, respectively. The last three columns show the shapes in the first two principal directions with scaling factor given by  $\sigma$  when the dataset is aligned with linear and area preserving parameter estimations, respectively.

In our case, a centred dataset is computed as (5.17) at the tangent space of the estimated mean. Hence, the geodesic connection (3.49) is used to linearise the data set  $Q$  to its vector form  $\mathbf{Q}$ . The solution for (5.25) is estimated using Singular Value Decomposition (SVD), since the objective function can be represented as follows

$$\begin{aligned} \max_{\mathbf{v}} E(\mathbf{Q}\mathbf{v})^2 \\ \max_{\mathbf{v}} \mathbf{v}^T \mathbf{Q}^T \mathbf{Q} \mathbf{v}. \end{aligned} \quad (5.26)$$

Hence,  $v$  is maximized by setting it to the eigenvector of the covariance matrix with the highest eigenvalue. Note that  $\mathbf{Q}^T \mathbf{Q}$  is the covariance matrix of the dataset. Subsequently, one can select as many eigenvectors sequentially as needed to reduce the dimensionality of the representation, the selected vectors are referred as principal components. The results in the reduced dimensional space can be exponentiated to the group representation as follows, with  $\mathbf{P}$  representing the selected eigenvectors

$$\bar{f}(\tilde{c}) \exp_{\bar{f}(\tilde{c})}(\mathbf{Q}_i^r \mathbf{P}^{-1}), \quad (5.27)$$

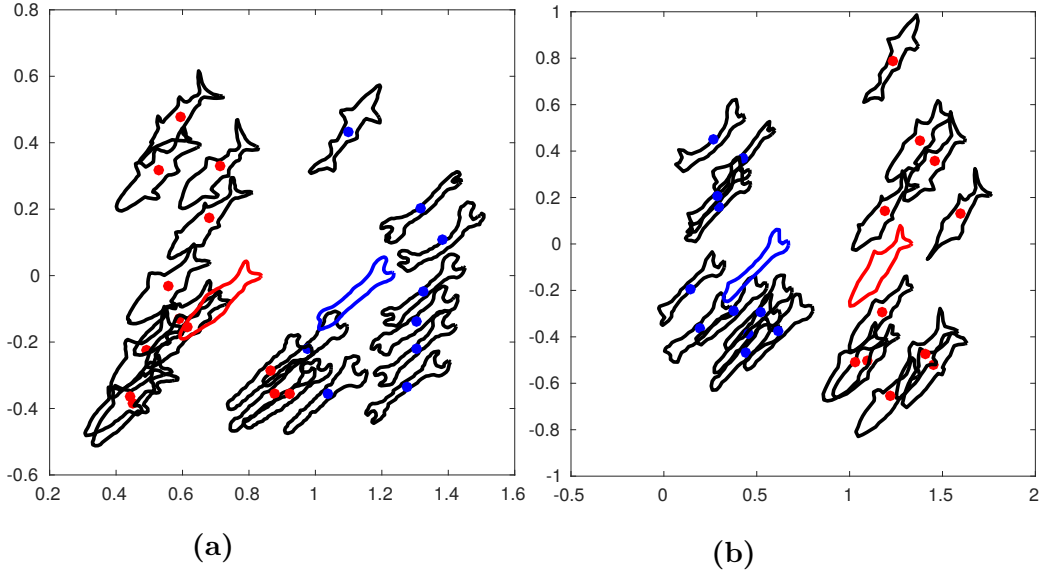
where  $Q^r$  represents some data point in the estimated subspace defined by  $\mathbf{P}$ , see Figure 5.5.

### 5.5.2 Clustering

Grouping a collection of data into distinct clusters is an alternative data analysis technique that is extensively studied in a wide range of problems. Apart from the well known k-means algorithm, there are several clustering algorithms emphasizing different aspects of the problem [NJW<sup>+</sup>02, WS06]. However, the performance of a clustering algorithm depends on a goal specific similarity metric, among other factors. In that regard, we propose (4.9) or (4.14), depending on the reparametrization assumption, as a similarity metric to cluster shapes. To that end, we discuss modelling of a shape class, represented in  $\text{SE}(n)^z$ , by k-means clustering.

In k-means clustering, the central idea is to estimate  $k$  points such that the data points around these  $k$  centroids are more similar with each other under

## 5. STATISTICAL CURVED SHAPE ANALYSIS



**Figure 5.6:** Examples of clustering results for  $k=2$ . Two different clustering results for different point correspondence estimations. (a) Point correspondence estimation based on linear parametrization. (b) Point correspondence estimation based on area preserving parametrization.

the metric being considered. Moreover, the  $k$  points are thought of as the mean points of the clusters [M<sup>+</sup>67]. However, the number of  $k$  is not known prior to the modelling task. Let us assume that we have selected the number of clusters  $k$  to categorize the data into. In such a case,  $k$ -means of shape clusters are estimated by optimizing the following

$$\min \sum_{j=1}^q \sum_{i=1}^k r_{ij} \mathfrak{d}(\bar{f}(\tilde{c}_i), f(\tilde{c}_j))^2, \quad (5.28)$$

where  $\bar{f}(\tilde{c}_i)$  is the representation of the estimated mean shape of the  $i^{\text{th}}$  cluster, and  $r_{ij}$  is 1 if and only if the  $j^{\text{th}}$  shape is assigned to the  $i^{\text{th}}$  cluster otherwise is 0. Note that the latent variables in (5.28) are the  $r_{ij}$ . Subsequently, the optimization of (5.28) can be done through Expectation-Maximization (EM) algorithm [NH98]. The general concept of an EM algorithm is to alternate between the following two steps:

1. **E-step:** Estimate the distribution of the latent variable from an initial parameter guess.

2. **M-step**: Estimate the parameters, which are the means in k-means, that maximize the likelihood of the dataset based on the latent variable distribution estimate.

Subsequently, in k-means clustering the **E-step** would be to estimate the  $r_{ij}$  for each data point, given initial mean guesses. In the **M-step** we minimize (5.28) with respect to the mean by setting its gradient to zero

$$\frac{\partial \sum_{i=1}^q r_{ij} \mathfrak{d}(\bar{f}(\tilde{c}_i), f(\tilde{c}_j))^2}{\partial \bar{f}(\tilde{c}_i)} = 0. \quad (5.29)$$

Note that the gradient of (5.29) is exactly the same as what is called the Karcher equation given in (5.12), since minimizing (5.28) for a given mean is equivalent to solving for Karcher mean. Consequently, it can be optimized using Algorithm 2, only in this case, we do not assign equal weights for the whole dataset, we have an estimate of the parameters  $r_{ij}$  as

$$r_{ij} = \frac{r_{ij}}{\sum_{i=1}^q r_{ij}}. \quad (5.30)$$

See Figure 5.6 for an illustration of a k-means result.

In addition to estimating the  $k$  clusters, the algorithm can be extended to guess the optimal number of  $k$  by introducing an additional assumption on the data distribution. In [HE04], each cluster is assumed to have a Gaussian distribution and clusters that do not satisfy the statistical test of a Gaussian distribution, which is based on Anderson-Darling test, are further split into two clusters. The procedure continues until all clusters confirm to Gaussian distribution. Nevertheless, note that as  $k \rightarrow q$ , the model becomes more like a non-parametric model. In fact, if  $k = q$  the model is memorization of the whole dataset.

### 5.5.3 Mixture of Gaussians

Mixture of Gaussians is a model based on Gaussian components in a superposition as described below

$$p(\mathbf{X}) = \sum_{i=1}^k \pi_i \mathcal{N}(\mathbf{X} | \mu_i, \Sigma_i), \quad (5.31)$$

## 5. STATISTICAL CURVED SHAPE ANALYSIS

---

where the  $\pi_i$  quantify the prior of each component while satisfying  $\sum_{i=1}^k \pi_i = 1$ . Each component is parametrized with its own covariance  $\Sigma_i$  and mean  $\mu_i$ . Given the number of components  $k$  is known, we will introduce a random latent variable  $\delta = \{l_1 \cdots l_k\}$  such that  $l_i \in \{0, 1\}$  and  $\sum_{i=1}^k l_i = 1$ . The introduction of the latent variable enables us to workout the responsibility of each Gaussian component in explaining the dataset. For instance, the posterior of component  $l_i$  for data point  $x_j$  is given as

$$\begin{aligned} p(l_i = 1|x_j) &= \frac{p(l_i)p(x_j|l_i)}{\int_l p(x_j, l)} \\ &= \frac{\pi_i \mathcal{N}(x_j|\mu_i, \Sigma_i)}{\sum_{i=1}^k \pi_i \mathcal{N}(x_j|\mu_i, \Sigma_i)}. \end{aligned} \quad (5.32)$$

Following the above formulation, the set of parameters for the mixture of Gaussians is defined  $(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \{(\pi_1, \mu_1, \Sigma_1), \cdots, (\pi_k, \mu_k, \Sigma_k)\}$ . The parameters are estimated from a given dataset  $\mathbf{Q} = \{x_1 \cdots x_q\}$ . Assuming the dataset  $\mathbf{Q}$  is i.i.d and the number of components  $k$  is known, parameter estimation is done by maximizing the log-likelihood which is given as

$$\ln p(\mathbf{Q}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{j=1}^q \ln \left\{ \sum_{i=1}^k \pi_i \mathcal{N}(x_j|\mu_i, \Sigma_i) \right\}. \quad (5.33)$$

**Mixture of Gaussian in SE(n):** Given a dataset  $\mathbf{x} \in \mathbf{R}^n$ , fitting a Gaussian mixture is a direct implementation of EM-algorithm as given in [Bis06]. Consequently, we need to derive a procedure to maximize the log-likelihood for the contribution parameters, the means and covariance matrices for (5.33). Particular to our case, which involves rotation matrices  $R$  and translation vector  $v$ , an observation  $x_j = \begin{pmatrix} R_j & v_j \\ 0 & 1 \end{pmatrix}$  is represented at the tangent space of the mean transformation matrix  $\bar{g}_i = \begin{pmatrix} \bar{R} & \bar{v} \\ 0 & 1 \end{pmatrix}$  as (5.17), which we write again in what follows.

$$\mathbf{v} = \begin{pmatrix} \log \bar{R}^T \bar{R}_j & \bar{R}^T (\bar{v} - v_j) \\ 0 & 0 \end{pmatrix}$$

Subsequently, the derivative of (5.33) with respect to one of its means,  $\mu_i$ , is as follows

$$\frac{\partial \ln p(\mathbf{x}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})}{\partial \mu_i} = - \sum_{j=1}^q \frac{\pi_i \mathcal{N}(x_j|\mu_i, \Sigma_i)}{\underbrace{\sum_{i=1}^k \pi_i \mathcal{N}(x_j|\mu_i, \Sigma_i)}_{\alpha_i}} \times \frac{\partial \mathbf{v}^T \boldsymbol{\Sigma}_i^{-1} \mathbf{v}}{\partial \mu_i}. \quad (5.34)$$

Since  $\boldsymbol{\Sigma}_i^{-1}$  is not a function of the mean  $\mu_i$ , using rules of matrix derivative [Ber05], we can simplify (5.34) as follows

$$- \sum_{j=1}^q \alpha_i \times \frac{\partial \mathbf{v}^T}{\partial \mu_i} \mathbf{v} \boldsymbol{\Sigma}_i^{-1}. \quad (5.35)$$

The optimizer of (5.35) is the solution of

$$0 = - \sum_{j=1}^q \alpha_i \times \frac{\partial \mathbf{v}^T}{\partial \mu_i} \mathbf{v} \boldsymbol{\Sigma}_i^{-1}. \quad (5.36)$$

Subsequently, we can eliminate the negative sign from (5.36) and turn the maximization problem into minimization. In that case, (5.36) is the same as the Karcher equation (5.12). Although we can safely eliminate the constant covariance matrix by multiplying both sides with  $\boldsymbol{\Sigma}_i$ , the weights  $\alpha_i$  do not add up to one. Since we are treating the mean as an expectation we multiply both sides by  $1/\sum_{j=1}^q \alpha_{ij}$ ; the parameters do not depend on the mean that is being optimized, thus will not affect our result. Hence, equation (5.36) can be optimized by solving for

$$0 = \sum_{j=1}^q \frac{\alpha_i}{\sum_{j=1}^q \alpha_{ij}} \times \frac{\partial \mathbf{v}^T}{\partial \mu_i} \mathbf{v}. \quad (5.37)$$

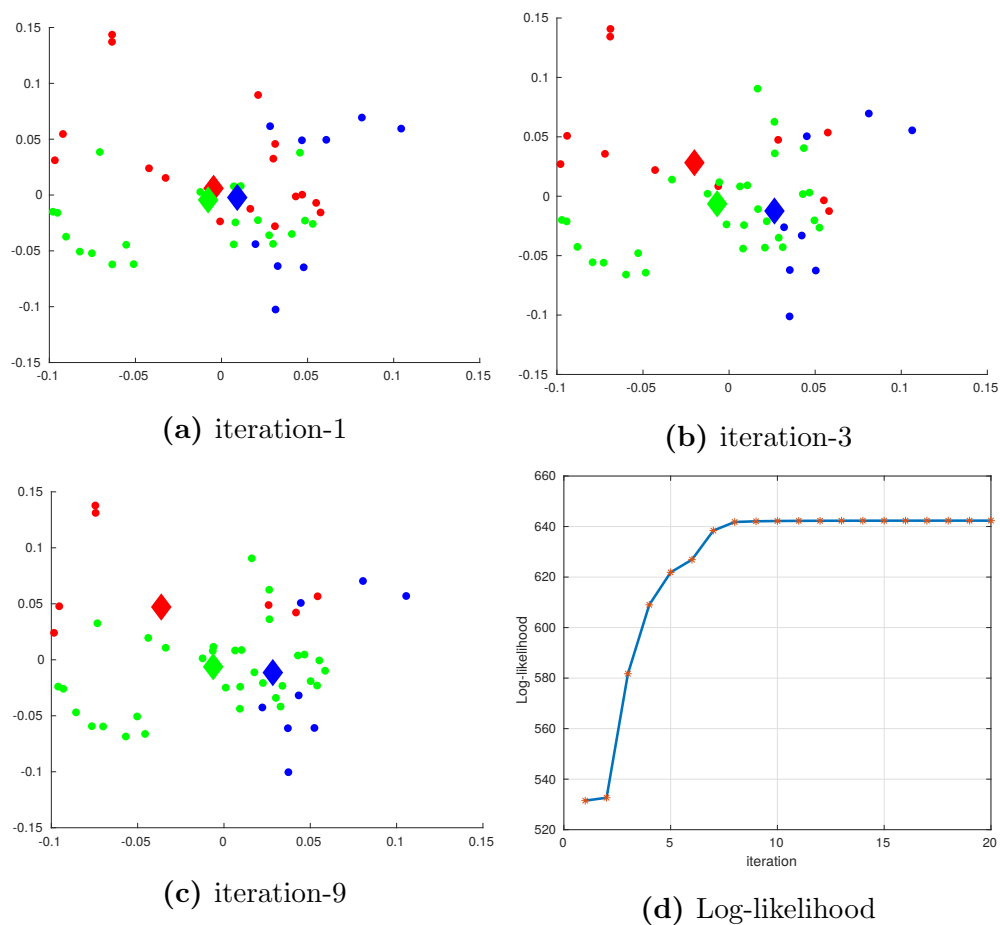
Consequently, we can use Algorithm 2 to solve for (5.37) while weighting each term by  $\alpha_i/\sum_{j=1}^q \alpha_{ij}$ .

Meanwhile, the covariance matrix can be estimated on the tangent space of the estimated mean with a closed form. Consequently, the optimal covariance matrix, as described in [MN95], is given as follows by using  $\text{vec}(\cdot)$  operation to vectorize a matrix.

$$\hat{\boldsymbol{\Sigma}}_i = \sum_{j=1}^q \frac{\alpha_i}{\sum_{j=1}^q \alpha_{ij}} \text{vec}(\mathbf{v}) \text{vec}(\mathbf{v})^T. \quad (5.38)$$

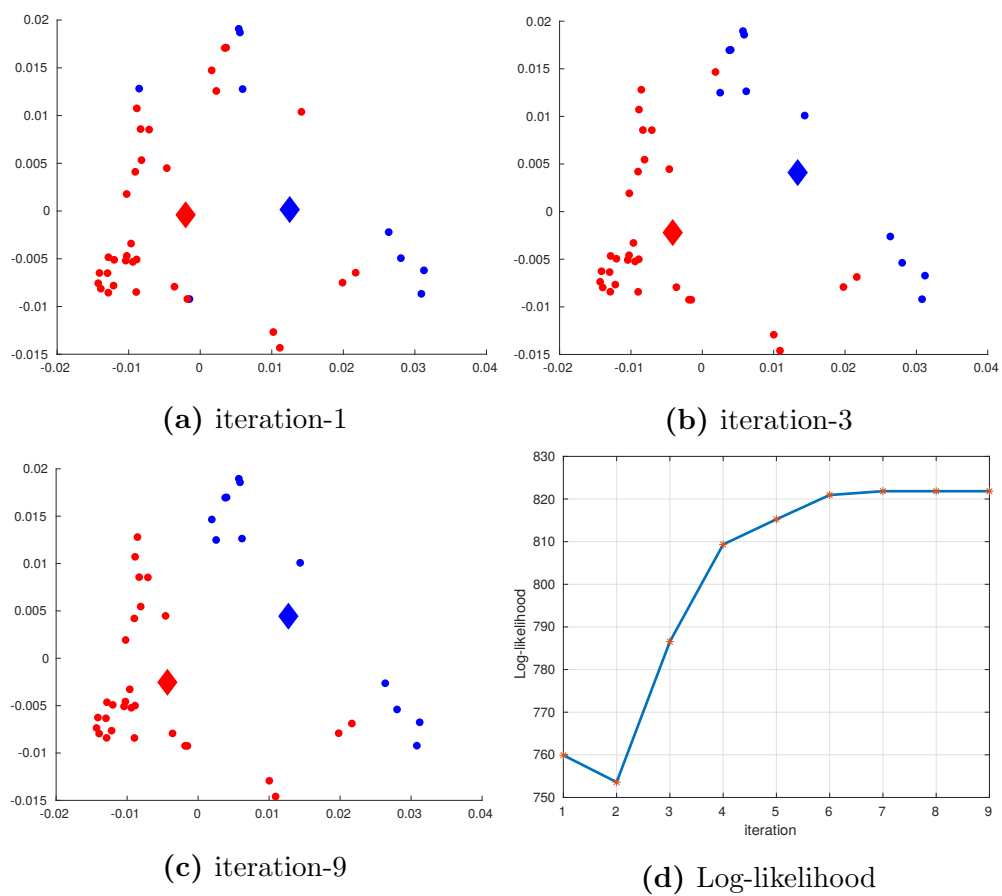
## 5. STATISTICAL CURVED SHAPE ANALYSIS

---



**Figure 5.7:** Mixture of Gaussian result on the dogs dataset with ( $k = 3$ ), for visualization the dataset is plotted with coordinates estimated with Multidimensional scaling. The diamond shapes show the position of the means. The color of each data point is assigned to the Gaussian that maximizes its posterior (5.32). (d) Shows the Log-likelihood after each iteration.





**Figure 5.8:** Mixture of Gaussians ( $k = 2$ ) result on a car dataset [LS03] where shape variation is only due to change of viewpoint.

## 5. STATISTICAL CURVED SHAPE ANALYSIS

---



**Figure 5.9:** Examples from ETH80 dataset [LS03].

Similarly, the composition weights  $\pi_i$  are estimated as given in [Bis06].

$$\hat{\pi}_i = \frac{\alpha_i}{q} \quad (5.39)$$

Finally, the EM-algorithm for estimating the mixture of Gaussians parameters for random variables that take their value in  $SE(n)$  is given as follows:

**E-step:** Compute equation (5.32), given the parameter estimates,  $\lambda$ .

**M-step:** Given the estimate of equation (5.32), maximize for the parameters. That is solve (5.37), (5.38), and (5.39). Note that, if the estimated mean  $\hat{\mu}$  happens to be one of the data points one has to reinitialize and look for alternative estimate. Otherwise, the log-likelihood will increase exponentially by collapsing on to a single data regardless of the other data points contribution, see [Bis06] for a details.

## 5.6 Evaluation

Section 5.6 details the evaluation of the statistical shape models discussed earlier.

**Dataset:** The ETH80 dataset [LS03] contains 8 object categories with 10 different object instances per category. Each object instance of the category is captured from 41 different viewpoints or poses. Moreover, each image of an object instance

Algorithms	Recognition accuracy (%)
PCA mask [LS03]	83.41%
Shape context + DP [LS03]	86.40%
IDSC + DP [LJ07]	88.11%
IDSC + Morphological strategy [HJZG12]	88.04%
Height function [WBY <sup>+</sup> 12]	88.72%
Robust symbolic [DT08]	90.28%
Kernel-edit [DT10]	91.33%
Bag of contour fragments [WFB <sup>+</sup> 14]	91.49%
<b>Our approach</b>	
Gaussian	90.67%
Mixture of Gaussians (k=2)	93.18%
<b>k-means (k=2)</b>	<b>95.06%</b>
k-means (k=3)	94.33%
k-means (k=4)	94.42%

**Table 5.1:** Average accuracy comparison between models based on the proposed approach and earlier shape feature based classifiers on ETH80 dataset. The best performing algorithm is highlighted with bold font.

is labelled with its pose, and the silhouette of the objects are given to enable the evaluation of shape based approaches. In total, the dataset contains 410 images per category, see Figure 5.9 for illustrative examples of the dataset. Since our approach is based on shape, we only consider the contour of the objects.

**Data processing and experimental scenario:** To compare our approach with methods based on shape features we replicate the leave-one-out scenario [LS03]. Consequently, out of the 80 object sets (10 object instance per 8 object categories) we use 79 to train a model for each object category and use the last one for testing. The procedure is repeated 80 times, thus each object instance is used once as a test set. Note that an object instance has 41 images from different viewpoint. Subsequently, the overall average accuracy is used as an evaluation metric for a given modelling procedure. A model for each object category is trained on shapes that are rotational aligned and optimal parameter estimated with respect to a template shape; template shape for every shape category is selected as the one viewed from an azimuth and elevation value  $(0^\circ, 90^\circ)$ . This

## 5. STATISTICAL CURVED SHAPE ANALYSIS

---

is mainly because in  $(0^\circ, 90^\circ)$  most of the object's distinctive features are visible. For optimal parameter estimation we set  $\alpha = 1$ , and  $\lambda = 0.2$ . Subsequently, the aligned training dataset is used for estimating a statistical model for each shape category.

**Classification:** We train three different kinds of models for the recognition task, k-means clustering, Gaussian, and mixture of Gaussians. In the k-means clustering, a given shape is assigned to a cluster if the cluster is the closest to the query shape than the others. note that, each query shape has to be rotationally aligned and its optimal parameter has to be estimated with respect to the mean shape using  $\alpha = 1$  and  $\lambda = 0.2$ . Subsequently, for a given query shape  $f(\tilde{c}_q)$  its label is estimated as

$$\hat{y} = \arg \min_{y \in Y} \{ \arg \min_{k \in K} \mathfrak{d}(f(\tilde{c}_{kl}), f(\tilde{c}_q)) \}, \quad (5.40)$$

where  $Y$  denotes the set of possible data labels. Similar to the k-means, in the Gaussian and mixture of Gaussians models a query shape is categorised into a given label if it maximizes the posterior or equivalently the likelihood function. That is, for a Gaussian model the posterior is given as

$$p(y|\mathbf{v}) \sim p(\mathbf{v}|y)p(y), \quad (5.41)$$

where  $\mathbf{v}$  is a tangent vector at the estimated mean shape. Since we do not have any prior over the models, maximizing the posterior is equivalent to maximizing the likelihood. Hence, the final label is given as

$$\hat{y} = \arg \max_{y \in Y} p(\mathbf{v}|y). \quad (5.42)$$

In this particular dataset, the best performance is achieved using k-means clustering with  $k=2$ , see Table 5.1. Naturally, one would expect the mixture of Gaussians to perform as well or better than k-means clustering. However, the log-likelihood (5.33) objective function is not convex and has many local minima and potential saddle points. Consequently, one has to start the optimization with appropriate initializations or at least iterate for a sufficiently enough time to escape a potential saddle point [LSJR16]. Here, we have iterated for ten times to

optimize the log-likelihood in all training scenarios. As a result, in practice, if the dataset's distribution resemble clusters of Gaussian distributions with diagonal covariance matrix then k-means clustering is the computationally cheapest model with reasonable performance. Nevertheless, all the models (Gaussian, Mixture of Gaussians, and k-means) performed comparably or better with respect to the other shape descriptor based classifiers.

## 5.7 Conclusion

In Chapter 5, we discussed the computation of basic statistics from a collection of shapes that are represented with the proposed curved shape representation in Chapter 3. Furthermore, we have presented the adaptation of several conventional statistical models to the proposed shape representation space. The statistical models are adapted to the proposed representation by linearising observed shape variations to the tangent space of the mean shape. Such models are particularly useful for classification and simulation of class specific shape categories. We have shown that for a shape based classification problem statistical models, built on the proposed representation, perform as well or better when compared to a shape descriptor based object recognition algorithms. Although all of the discussed models are based on frequentist approach, Bayesian statistical models can also be applied to the proposed curved shape representation by defining an a priori distribution over the parameters. For parameters that take value in  $\mathbf{R}^n$ , one can define a priori distribution as is done conventionally, e.g., conjugate prior. Meanwhile, for parameters that take value in  $SE(n)$ , e.g., mean shape, the priori distribution can be defined in the Lie algebra of the representation space. Subsequently, a given observation can be evaluated at the Lie algebra using the exponential maps, see Appendix A.

## 5. STATISTICAL CURVED SHAPE ANALYSIS

---

# Chapter 6

## 3D facial expression analysis

### 6.1 Motivation

Chapter 6 presents an application of the proposed curved shape representation and modelling approaches for facial expression analysis from 3D point cloud. Representing facial expressions is an integral part of what is called affective computing [Pan09, ZPRH09]. Facial representations are used in combination with a classifier to recognize facial expressions and infer affect. Many other disciplines such as human computer interaction, computer graphics, health monitoring benefit from the ability to model facial expressions for the purpose of analysis, animation, and recognition.

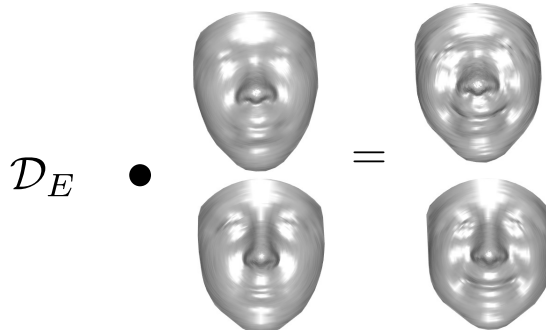
The chapter is organized as follows: in Section 6.2 we present earlier works in facial expression representation and modelling. In Section 6.3, we present preprocessing approaches that are used to decompose 3D facial surface into a set of 3D curved shapes referred to as *facial curves*. Subsequently, a direct product of the proposed curved shape representation, Chapter 3, is used to capture facial surface and their deformations. Section 6.4 formulates the problem of curve correspondence estimation and proposes a dynamic programming based solution. In Section 6.5, we present a general modelling approach using exponential maps. Experimental results and comparisons with earlier approaches are presented in Section 6.6. The chapter ends with concluding remarks in Section 6.7.

### 6.2 Related works

Over the last decades, several methods were proposed for facial expression representation. In general, these methods can be categorized into two main categories; 1) methods based on feature descriptors, and 2) methods based on a generic expression space learning.

In the first category, features are labelled, according to their type, as either geometric or appearance. Subsequently, both feature types are further labelled, according to their construction, as predefined [PB07], directly learned from a training data [RSMH11, LHMT14], or a hybrid thereof [ZJ05]. Approaches based on predefined features aim to detect facial action units (AU) defined in [EF77], and defer the task of expression labelling to a higher level processing, while in learned features, the attempt is to learn a descriptive representation directly from a training data set. In [Pan09, BVS<sup>+</sup>96], methods that detect facial AU are argued to be comprehensive and robust to subjective labelling of expressions – since the detection of AU is decoupled from expression detection, a new facial expression can be discovered by combining AUs. On the other hand, systems that integrate feature learning with expression labelling have been shown to benefit from the supervised learning of strong discriminative features [LHMT14, TYRW14]. In general, mapping functions from the raw data space to the feature space are not necessarily bijective, hence inversion of features is numerically approximated [MV15, VKMT13]. Consequently, it is not straightforward to translate a linear combination or scalar multiplication of features to the raw data space. Moreover, most feature based approaches depend on pre-annotated/estimated landmark points [FZO<sup>+</sup>11]. In the second category of facial expression representations, a general space of faces and their expressions is estimated. In [HL07], an  $L_2$  norm is defined on the displacement field of faces to learn an embedding of the expression space. In [CVTV05], a 3D template face is used to match all faces and learn a generalized expression space rather than a per subject learning. However, empirically estimating the space of facial expressions from a small data set in high dimensional space is difficult and most of the time only a small portion of the space ends up being estimated, e.g., expression space of a given





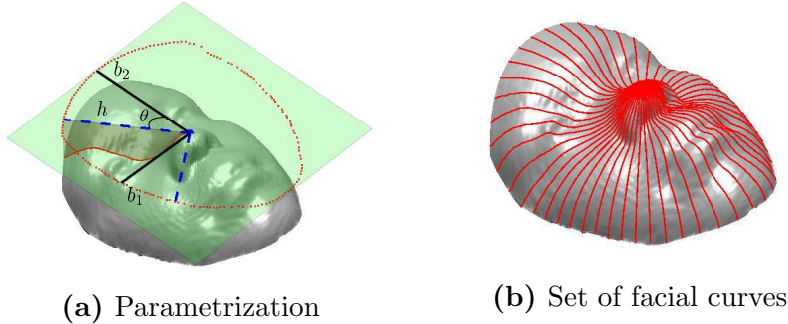
**Figure 6.1:** Proposed 3D facial expression representation. A smile is captured by the deformation  $\mathcal{D}_E$  and applied to two different subjects, preserving their specific shapes.

subject. As such, the estimated expression space is not guaranteed to be complete or connected [BCV13]. One consequence of this is the need for search based computational schemes, e.g., the geodesic distance between expressions is computed using graph-based shortest path algorithms [CHFT06, HL07]. Meanwhile, in [MMS08, TF00, AOBM09], a linearity prior is introduced in the estimation of the expression space. In [TF00, MMS08], an observed facial data is formulated as a bilinear function of face and expression. Thus, expressions are modelled as linearly separable objects. In a similar spirit to our approach, in [AOBM09] the point-to-point difference between a neutral face and a face with an expression is taken as expression residue and the expression space is estimated with PCA (principal component analysis). However, expression space is a nonlinear space and linear models fail to discover the underlying non-linear manifold [WHL<sup>+</sup>04]. Alternatively, in [DBAS<sup>+</sup>13, SSD06, SSDK09, DADS10] the full facial surface is considered as a geometric object and analysed by decomposing it into a set of curves called *facial curves*. Face and facial expression is analysed using the SRV framework [SKJJ11] on these curves.

Here, we propose a similar surface decomposition except instead of the SRV framework we will use the proposed curve representation to represent facial curves. Subsequently, by taking a direct product of the facial curve representations, a mapping function that identifies a set of facial curves with an element of a high dimensional matrix Lie group is defined. Given such a mapping function, facial

## 6. 3D FACIAL EXPRESSION ANALYSIS

---



**Figure 6.2:** Facial curve extraction.

expressions are then represented by a left action of the group on a neutral face representation, see Figure 6.1. Although the idea of decoupling facial expression from a neutral face is presented in [AOBM09], the representation space presented here, Lie group, models expression space as a nonlinear space. Consequently, captures nonlinear variation of facial expressions. Meanwhile, the proposed facial expression representation can be locally linearised by mapping the expression representations from the Lie group to the Lie algebra. Hence, conventional linear models, like SVM (support vector machine), can directly be trained on the representation.

### 6.3 Surface representation

We decompose a segmented, hole-free, facial surface into a family of curves that start from a given reference point, see Figure. 6.2b. This decomposition allows to view a facial surface as an ordered set of curves and a facial expression as its deformation. In what follows, we cover the data preprocessing and facial curve extraction stage. As discussed earlier, the main causes of variation in a shape data set, with respect to a given fixed coordinate system, are deformation, scaling, translation and rotation. Among those, the most informative variation is the one due to deformation as the rest does not change the nature of the shape. Consequently, in the preprocessing stage we filter out shape preserving transformations with respect to a given fixed world coordinate system.

**Removing location and scale:** A given facial point cloud  $\Gamma = \{p_1, \dots, p_n\}$ , with  $p_i \in \mathbb{R}^3$ , is centred to zero mean and unit norm as follows:

$$\Gamma = \left\{ \frac{p_1 - \bar{p}}{h}, \dots, \frac{p_n - \bar{p}}{h} \right\}, \quad (6.1)$$

where  $\bar{p} = \frac{1}{n} \sum_i^n p_i$ , and  $h = \sqrt{\sum_i^n \|p_i - \bar{p}\|_2^2}$ .

**Filtering rotation:** The head orientation of a normalized facial point cloud  $\Gamma$  is aligned with an arbitrarily selected reference face, using the Iterative Closest Point (ICP) algorithm. We note that ICP gives a reasonable result only when the point cloud is a segmented and hole free facial data. Subsequently, singular value decomposition (SVD) is used for estimating the coordinate orientation of the reference face and the given face to further refine the alignment. We again stress that the data should be described from a fixed reference coordinate system for the representation to be meaningful.

**Facial surface decomposition:** We first select the starting point of the facial curves  $p^s$  as the tip of the nose; this is mainly because it is relatively easier to estimate. The estimation is done by selecting the point with a maximum component in the direction of the eigenvector with the smallest eigenvalue. As such the estimated nose tip point is given as

$$p^s = \arg \max_{\forall p \in \Gamma} \langle \mathbf{P}_j, p \rangle, \quad (6.2)$$

where  $\mathbf{P}_j$  denotes the eigenvector with the smallest eigenvalue while; the eigenvectors and eigenvalues are computed using SVD on the normalized facial data. Subsequently, let  $P_t$  be the tangent plane at  $p^s$  with  $b_1$  and  $b_2$  as its orthonormal basis, see Figure 6.2a. Then for some  $\theta \in [0, 2\pi]$ , a facial curve on the surface  $\Gamma$  is given as follows

$$\Gamma(\theta) = \{p \in \Gamma \mid (R(\theta) \times b_1) \cdot (p - p^s) = 0 \wedge ((R(\theta) \times b_2) \cdot (p - p^s) > 0)\}, \quad (6.3)$$

where  $R$  is a rotation matrix about the normal vector of  $P_t$ , see Figure 6.2a. Furthermore, a given facial curve can be parametrized by  $r \in [0, h]$ , where  $h$

## 6. 3D FACIAL EXPRESSION ANALYSIS

---

is the mean radius of the surface projection on to the rotation plane. Consequently, the parametrization of the full facial surface is given by  $\Gamma(r, \theta)$  such that  $\Gamma(0, \theta) = p_r, \forall \theta$ . Note that we can further define,  $\xi$  on  $r$  to further approximate each curve by a set of  $z$  discrete points, see Chapter 3. The parametrization can then be used to extract a set of curves by defining the values of  $\theta$  and  $\xi$ .

In Subsection 6.3.1, we propose to represent face and facial expressions as a set of curves represented with the proposed method, and facial expression as the left action of a direct product Lie group on the surface.

### 6.3.1 Face representation

Let  $\Upsilon^i = \{\tilde{c}_1, \dots, \tilde{c}_k\}$  be a set of  $k$  facial curves sampled from a facial surface  $\Gamma$  as described in Section 6.3. Let  $(\mathcal{C}_z)^k$  be the set of all possible  $k$  facial curves approximated  $z$  points. Subsequently, given the curve representation discussed in Chapter 3, a straightforward representation of  $\Upsilon^i$  is to take the direct product of the proposed representation (3.14) which defines the following mapping from  $(\mathcal{C}_z)^k$  to a high dimensional Lie group as

$$\mathcal{F} : (\mathcal{C}_z)^k \rightarrow (\text{SE}(3)^z)^k \quad (6.4)$$

such that

$$\mathcal{D} = \mathcal{F}(\Psi^i) = (f(\tilde{c}_1), \dots, f(\tilde{c}_k)) = (G_1, \dots, G_k). \quad (6.5)$$

The inverse of the mapping function is given as

$$\mathcal{F}^{-1}(\mathcal{D}) = \{f^{-1}(G_1), f^{-1}(G_2), \dots, f^{-1}(G_k)\}, \quad (6.6)$$

where  $f^{-1}(\cdot)$  is as defined in (3.20). The geodesic path  $\Lambda(t)$  and distance  $d_F$  between two facial representations,  $\mathcal{F}(\Upsilon^1)$  and  $\mathcal{F}(\Upsilon^2)$ , are given by direct products using (3.48) and (3.49), respectively. Specifically,

$$\Lambda(t) = (\zeta_1(t), \dots, \zeta_k(t)), \quad (6.7)$$

see Figure 6.4, and the distance  $d_F$  is defined as

$$d_F(\mathcal{F}(\Upsilon^1), \mathcal{F}(\Upsilon^2)) = (\mathfrak{d}(G_1^1, G_1^2)^2 + \dots + \mathfrak{d}(G_k^1, G_k^2)^2)^{1/2}. \quad (6.8)$$

### 6.3.2 Expression representation

Similar to the single curve deformation, a deformation between two facial representations  $\mathcal{F}(\Upsilon^1)$  and  $\mathcal{F}(\Upsilon^2)$  is given by the action of the group on itself. A deformation that acts from the left is

$$\mathcal{D}^L = \mathcal{F}(\Upsilon^2)\mathcal{F}(\Upsilon^1)^{-1}. \quad (6.9)$$

Alternatively, a deformation that acts from the right is

$$\mathcal{D}^R = \mathcal{F}(\Upsilon^1)^{-1}\mathcal{F}(\Upsilon^2), \quad (6.10)$$

where  $\mathcal{F}(\Upsilon)^{-1} = (f(\tilde{c}_1)^{-1}, \dots, f(\tilde{c}_k)^{-1})$ ,  $f(\cdot)^{-1}$  is element wise matrix inversion as is given in (3.20) in Chapter 3.

In general, a left action of a given group  $G$  on a set  $Y$  is given as  $G \times Y \mapsto Y$ . The action is said to be *regular* if for every  $x, y \in Y$  there exists exactly one  $g \in G$  such that  $gx = y$ . In our case,  $Y$  is the group  $G$  itself, hence (6.9) is a regular action. To see this fact, consider  $\mathcal{F}(\Upsilon^1), \mathcal{F}(\Upsilon^2) \in (\text{SE}(3)^z)^k$  such that

$$\mathcal{D}^L\mathcal{F}(\Upsilon^1) = \mathcal{F}(\Upsilon^2). \quad (6.11)$$

Subsequently, let us assume that there is another  $\mathcal{D}_2^L \neq \mathcal{D}^L$  that satisfies the condition  $\mathcal{D}_2^L\mathcal{F}(\Upsilon^1) = \mathcal{F}(\Upsilon^2)$ . In that case,

$$\mathcal{D}^L\mathcal{F}(\Upsilon^1) = \mathcal{D}_2^L\mathcal{F}(\Upsilon^1), \quad (6.12)$$

since  $(\text{SE}(3)^z)^k$  is a group it implies  $\mathcal{D}^L = \mathcal{D}_2^L$ . Hence, given representation of a face with a neutral expression  $\mathcal{F}(\Upsilon_N)$  and a non-neutral expression  $\mathcal{F}(\Upsilon_E)$ , we can uniquely identify a deformation due to the non-neutral expression as

$$\mathcal{D}_E = \mathcal{F}(\Upsilon_E)\mathcal{F}(\Upsilon_N)^{-1}. \quad (6.13)$$

Thus, we use  $\mathcal{D}_E$  to represent a facial expression irrespective of subject specific facial shapes. Consequently, the proposed facial expression representation is complete, i.e., a given facial deformation can only be identified by a unique group element, see Chapter 2. However, the uniqueness of the facial expression is conditioned on the knowledge of the surface parameters,  $(\xi, \theta)$ . Nevertheless, similar to curved shapes the parametrization of a facial surface is not usually known a priori, unless there exists manually annotation of the facial points. As a result, in Section 6.4 we discuss a restricted version of the parameter estimation problem for facial surfaces.

## 6. 3D FACIAL EXPRESSION ANALYSIS

---

$$\begin{array}{ccc}
 f(\bar{\psi}_j) & \xrightarrow{m} & f(\bar{\psi}_{j+1}) \\
 \mathcal{D}_j^R \downarrow & & \downarrow \mathcal{D}_{j+1}^R = m^{-1}\mathcal{D}_j^R m^* \\
 f(x_j) & \xrightarrow{m^*} & f(x_{j+1})
 \end{array}$$

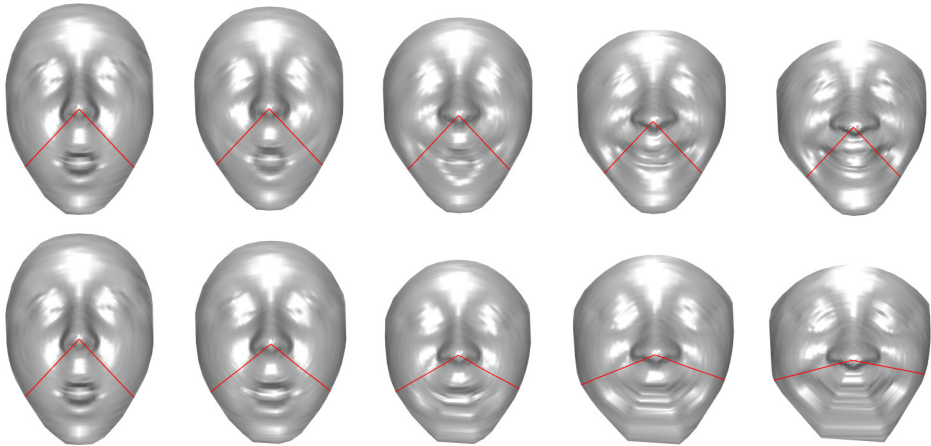
**Figure 6.3:** Commutative diagram. Deformation between two curves is represented in terms of the deformation between previously selected curves. This formulation is used to estimate curve correspondence via dynamic programming.

### 6.4 Curve correspondence

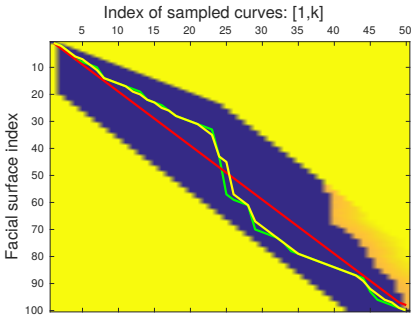
The proposed facial expression representation assumes the parametrization of two different faces to be optimal when factoring deformations (6.13). In practical terms, this means the index of a curve that passes through a particular region of the mouth in  $\mathcal{F}(\Upsilon_N)$  is assumed to correspond with the index of a curve that covers the same mouth region in  $\mathcal{F}(\Upsilon_E)$ . In such a case, the factored expression reflects the deformation of a curve due to the observed expression. Such an assumption, however, is violated when there is a significantly large nonlinear deformation between faces, see Figure 6.4. Consequently, in case of large nonlinear deformations, the factored expression includes deformations that reflect the curve mismatch rather than the observed expression. In this section, we present a cost functional that emulates the area preserving curve parametrization (4.16) for matching facial curves optimally and discuss a dynamic programming based solution.

#### 6.4.1 Cost of mismatching curves

Let  $\theta : [0, k] \rightarrow [0, 2\pi]$ , where  $k$  is the number of curves. Similar to the curve sampling functions  $\xi$ , see Chapter 3, we insist on the angular sampling  $\theta$  to be injective, monotonic, and we restrict its initial and last values to  $\theta(0) = 0$  and  $\theta(k) = 2\pi$ . Consequently,  $\theta$  will define the angles used to sample facial curves from a facial surface  $\Gamma$ . That is, for a given facial surface  $\Gamma$ , different families of



(a)



(b)

**Figure 6.4:** Curve matching between two different faces with “surprise”, on the far left, and “happy”, on the far right, expressions. The first row shows the geodesic deformation between the faces when both faces are represented by 50 uniformly sampled curves. The red curves are tracked along the deformation to illustrate the mismatch. In the second row, the “surprise” face is represented by 50 uniformly sampled curves while the “happy” faces is optimally sampled via dynamic programming, see Section 6.4. The cost matrix shows three solutions for different weighting factors—the blue region is the feasible set defined by the sectors size. The red curve is the optimal solution for  $\alpha = 0$  and  $\lambda = 1$ , the green path is optimal for  $\alpha = 1$  and  $\lambda = 0$ , finally the yellow path is optimal for  $\alpha = 1$  and  $\lambda = 1$ , the second deformation is based on the yellow path.

## 6. 3D FACIAL EXPRESSION ANALYSIS

---

curves  $\Gamma(\xi_j, \theta_i)$  are obtained for different values of  $i$  and  $j$ . Subsequently, for a fixed arc length based uniform curve parametrization  $\xi^*$ , we define the space of deformations between two facial surfaces  $\Gamma^1$  and  $\Gamma^2$  as

$$\Omega = \{\mathcal{D}^R | \exists \theta_i, \theta_j : \mathcal{F}(\Gamma^1(\xi^*, \theta_i))\mathcal{D}^R = \mathcal{F}(\Gamma^2(\xi^*, \theta_j))\}. \quad (6.14)$$

Subsequently, the distance between two facial surface representations can be written in terms of the parameters  $(\theta_i, \theta_j)$  as

$$\min_{\theta_i, \theta_j} d_F(\mathcal{F}(\Gamma^1(\xi^*, \theta_i)), \mathcal{F}(\Gamma^2(\xi^*, \theta_j))). \quad (6.15)$$

Next, using the left invariance property of the distance metric between curves, see Section 3.4, we write (6.15) as

$$\min_{\theta_i, \theta_j} d_F(\mathcal{F}(\Gamma^1(\xi^*, \theta_i)), \mathcal{F}(\Gamma^2(\xi^*, \theta_j))\mathcal{D}^R) = \min_{\theta_i, \theta_j} d_F(e, \mathcal{D}^R), \quad (6.16)$$

where  $\mathcal{D}^R = \mathcal{F}(\Gamma^1(\xi^*, \theta_i))^{-1}\mathcal{F}(\Gamma^2(\xi^*, \theta_j))$  and  $e$  is the identity in  $(\text{SE}(3)^z)^k$ . Intuitively, the functional given in (6.16) attempts to find the least costly deformation from  $\Omega$ , see (6.14). In effect, optimal curve parametrizations of the faces,  $\theta_i$  and  $\theta_j$ . However, our definition of  $\Omega$  permits angular samplings that does not preserve geometric properties, e.g., volume of the face with respect to its support plane. Thus, a solution parametrization might give the least costly deformation, according to (6.16), but can deviate from the target shape, see Figure 6.4. To address this problem, we add a term that penalizes parametrizations that do not preserve a particular geometric property; in our case volume.

The enclosed volume of a face  $\Gamma$ , parametrized with  $r$  and  $\theta$ , with respect to the rotation plane is given as

$$\text{Vol}(\Gamma(r, \theta)) = \int_0^{2\pi} \int_0^h r \Gamma(r, \theta) dr d\theta. \quad (6.17)$$

Subsequently, for a given  $\theta$  and uniform sampler  $\xi^*$  the volume is approximated as

$$\text{Vol}(\Gamma(\xi^*, \theta)) \approx \sum_{j=1}^{k-1} \Delta\theta(j) \sum_{i=1}^{z-1} h \Delta\Gamma(\theta(j), \xi^*(r(i))) \Delta r(i), \quad (6.18)$$



where  $\Delta$  denotes the forward difference. As a result, the approximate volume given in (6.18) depends on  $\theta$ , since  $\xi^*$  is fixed for simplicity, see (6.14). Assuming a uniform angular sampler  $\theta^*$  preserves volume, the objective functional given in (6.16) is penalized by the difference between the volume of the face due to a candidate solution sampler and a uniform sampler, which is formulated as

$$\begin{aligned} \arg \min_{\theta_i, \theta_j} \{ & \alpha \times d_F(e, \mathcal{D}^R) \\ & + \lambda \times |(\text{Vol}(\Gamma^1(r, \theta_i)) - \text{Vol}(\Gamma^1(r, \theta_*)))| \\ & + \lambda \times |(\text{Vol}(\Gamma^2(r, \theta_j)) - \text{Vol}(\Gamma^2(r, \theta_*)))|\}, \end{aligned} \quad (6.19)$$

with  $\lambda$  and  $\alpha$  as scalar weighting terms. Thus, a large value of  $\lambda$  encourages volume preserving solutions, while a large value of  $\alpha$  encourages deformation optimizing solutions.

### 6.4.2 Dynamic programming based solution

In Subection 6.4.2, we reformulate (6.19) as a dynamic optimization problem. To simplify the computational cost, we fix the angular sampler  $\theta$  of the surface  $\Gamma^1$  to a uniform sampler and optimize for the sampler of the other surface  $\Gamma^2$ . As a result, (6.19) is simplified to

$$\begin{aligned} \arg \min_{\theta_j} \{ & \alpha \times d_F(e, \mathcal{D}^R) \\ & + \lambda \times (\text{Vol}(\Gamma^2(\xi^*, \theta_j)) - \text{Vol}(\Gamma^2(\xi^*, \theta^*)))\}, \end{aligned} \quad (6.20)$$

Next, we write (6.20) as a recursive function to estimate its solution via dynamic programming. To elaborate, we first write a general form of a facial surface decomposition as

$$\Gamma^2(\xi^*, \theta) = \{x_1, \dots, x_k\} : \quad x_j \subset U_j. \quad (6.21)$$

The  $U_j \subset \Gamma^2$  are sectors of the face, i.e., subsets of the facial surface from which  $x_j$  can take its values. Hence, the sector size introduces a constraint in the search space of each  $x_j$ . In this work, we approximate the sectors  $U_j$  by a fixed

## 6. 3D FACIAL EXPRESSION ANALYSIS

---

size sliding window. Subsequently, we can rewrite the first term of (6.20) as a dynamic optimization problem as

$$d_F(e, \mathcal{D}^R)^2 = \alpha \times \sum_{j=1}^{k-1} \phi_j(x_j, x_{j+1}). \quad (6.22)$$

Given two successive uniformly sampled curves of  $\Gamma$ ,  $\tilde{c}_j^*$  and  $\tilde{c}_{j+1}^*$ , we define  $\phi_j$  in (6.22) as

$$\phi_j(x_j, x_{j+1}) = \mathfrak{d}(e, m^{-1}(\mathcal{D}_j^R)m^*)^2, \quad (6.23)$$

such that

$$m^{-1} = f(\tilde{c}_{j+1}^*)^{-1}f(\tilde{c}_j^*), \quad (6.24)$$

and

$$m^* = f(x_j)^{-1}f(x_{j+1}). \quad (6.25)$$

Hence,  $\mathcal{D}_{j+1}^R = m^{-1}(\mathcal{D}_j^R)m^*$ , see Figure 6.3. Similarly, the remaining term of (6.20) can be written as a recursive function with optimal substructure as

$$\lambda \times \sum_{j=1}^{k-1} (\text{Vol}(\Gamma^2(\xi^*, \theta))|_{x_j}^{x_{j+1}} - \text{Vol}(\Gamma^2(\xi^*, \theta^*))|_{\tilde{c}_j^*}^{\tilde{c}_{j+1}^*}). \quad (6.26)$$

Consequently, by writing (6.20) in terms of (6.22) and (6.26) we formulate the objective functional as a dynamic optimization problem and estimate the solution via dynamic programming, see Figure 6.4. To that end, we assume that  $\Gamma^1$  is approximated by  $k$  uniformly sampled curves, while  $\Gamma^2$  is approximated by  $K \gg k$  curves. In effect, the goal is to sample  $k$  curves from  $\Gamma^2$  such that the deformation from  $\Gamma^1$  to  $\Gamma^2$  is least costly, according to (6.20), while preserving the volume of  $\Gamma^2$ . In order to get a monotonic and injective sampler, we impose further restriction on the values  $x_j$  can take. As such, let  $\text{Vol}|_{[0,n]}$  be the volume of the facial surface sector up to  $\tilde{c}_n^2$ , where  $\tilde{c}_n^2$  is a facial curve of  $\Gamma^2$ . Subsequently, if  $x_{j+1} = \tilde{c}_n^2$  then we will only consider  $x_j = \tilde{c}_a^2 \in U_j : \text{Vol}|_{[0,a]} < \text{Vol}|_{[0,n]}$ . The proposed curve correspondence estimation is summarized in Algorithm 3. Note that the cost of selecting a curve for the  $j^{\text{th}}$  position and its optimizer are denoted

---

**Algorithm 3:** Optimal sampling of curves from  $\Gamma^2$ 


---

**Data:**  $\Gamma(\theta_*)$ ,  $\Gamma^2$ 
Initialization:  $s = |U|$ ,  $k = |\Gamma(\theta_*)|$ ,  $j = 1$ ,  $O_1(1) = \tilde{c}_1^2$ ;

**for**  $j < k$  **do**

    **for**  $x_{j+1} \subset U_{j+1}$  **do**

         $U = \{\tilde{c}_a^* \in U_j : (\text{Vol}|_{[0,a]} < \text{Vol}|_{[0,j+1]})\}$ ;

        **for**  $x_j \subset U$  **do**

             $C_{j+1}(x_{j+1}) = \rho_i(x_j, x_{j+1}) + C_j(x_j)$ ;

        **end**

         $C_{j+1}(x_{j+1}) = \min_{x_j \in U_j} C_{j+1}(x_{j+1})$ ;

         $O_{j+1}(x_{j+1}) = \arg \min_{x_j \in U_j} C_{j+1}(x_{j+1})$ ;

    **end**
**end**
 $q = k - 1$ ;

**for**  $q > 1$  **do**

     $\tilde{c}_q^2 = O_{q+1}(x_{q+1})$ ;

**end**
**Result:**  $\Gamma^* = \{\tilde{c}_1^2, \dots, \tilde{c}_k^2\}$ 


---

## 6. 3D FACIAL EXPRESSION ANALYSIS

---

by  $C_j(x_j)$  and  $O_j(x_j)$  as is discussed in Subsection 4.4.2 in Chapter 4. For a fixed sector size  $s$ , the time complexity of the estimation is  $\mathcal{O}(s^2k)$ .

**Limitations:** The proposed curve correspondence estimation is done in a restricted way, i.e., we are optimally sampling curves and leaving the point sampling to uniform sampling,  $\xi^*$ . Hence the solution to the objective functional is a curve in the search space. We believe a conjugate optimization of both curve and point sampling is more reliable and complete than optimal curve sampling, which would give a surface as a solution. However, it is highly taxing in computational time, see Figure 6.5b. Moreover, the weighting term  $\lambda$  on the constraint of the objective functional is tuned and not estimated, see Figure 6.6 and Figure 6.7. One possible solution for this is to consider a combination of dynamic programming and Lagrangian multiplier methods [Bel56].

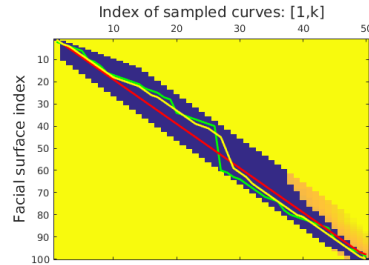
### 6.5 Modelling expressions

The presented approach represents facial expressions in a nonlinear space, a matrix Lie group. However, one can easily linearize a representation of a given expression  $\mathcal{D}_E$  by projecting it to the group's Lie algebra, which is a vector space. To define the projection we first define the mapping of a matrix  $g \in \text{SE}(3)$  to its Lie algebra  $\mathfrak{se}(3)$  by a matrix logarithm as

$$\log : \text{SE}(3) \mapsto \mathfrak{se}(3), \quad (6.27)$$

see Appendix A for further details. Subsequently, by taking the direct product of (6.27) we define the mapping of an expression  $\mathcal{D}_E$  to the Lie algebra as  $\mathcal{L}(\mathcal{D}_E) = ((\log)^z)^k(\mathcal{D}_E)$ . Under such linearization, an expression will be  $(3 + 3) \times (\#\text{points} - 1) \times (\#\text{curves})$  dimensional vector, where the threes are counts of the independent components of the skew-symmetric matrix and the translation. Alternatively, an expression represented on the Lie algebra can be mapped back to the Lie group using a direct product of matrix exponentials defined as

$$\exp : \mathfrak{se}(3) \mapsto \text{SE}(3), \quad (6.28)$$



(a) Cost Matrix

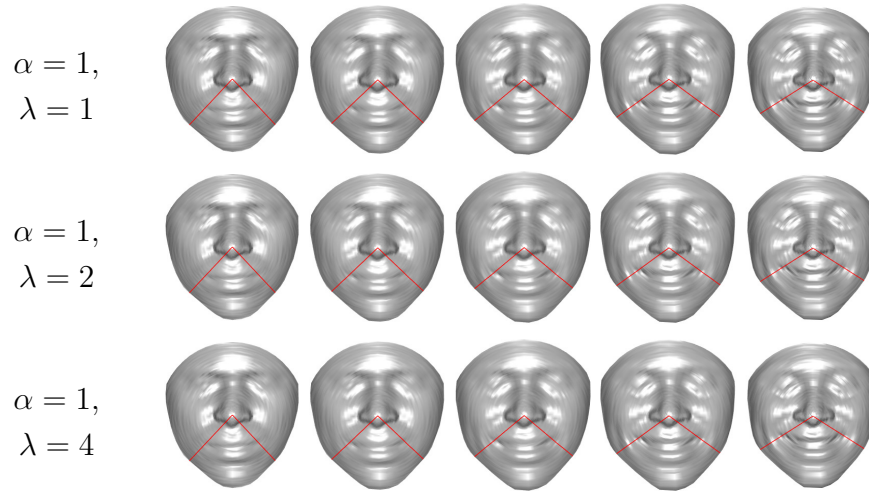


(b) Geodesic deformation

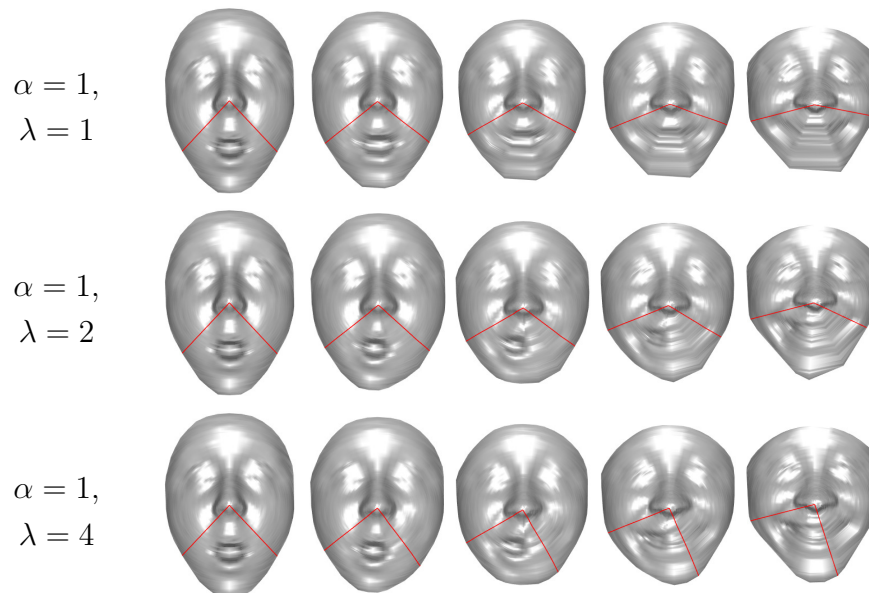
**Figure 6.5:** Curve matching. In (a) a cost matrix with three solutions is shown that correspond to the three deformations in (b). In all the deformations, the first faces are approximated by 50 uniformly sampled curves, i.e., they are  $\Gamma$ , while the last faces are sampled optimally, i.e., they are  $\Gamma^*$ . Pair of facial curves are highlighted in red in all of the deformations to illustrate the impact of the matching. The deformation in the first row is according to the red path in the cost matrix which is the solution when  $\alpha = 0$  and  $\lambda = 1$ , that is when both faces are sampled uniformly. The deformation in the second row is based on the green path which is the solution when  $\alpha = 1$  and  $\lambda = 0$ , that is when there is no volume based constraint. The last deformation is based on the yellow path which is computed for  $\alpha = 1$  and  $\lambda = 4$ . In this particular example, regardless of the good curve matching solution given by the yellow path, the deformation of the lower lip is not smooth. *This is mainly because we are only matching curves and disregarding point matching.*

## 6. 3D FACIAL EXPRESSION ANALYSIS

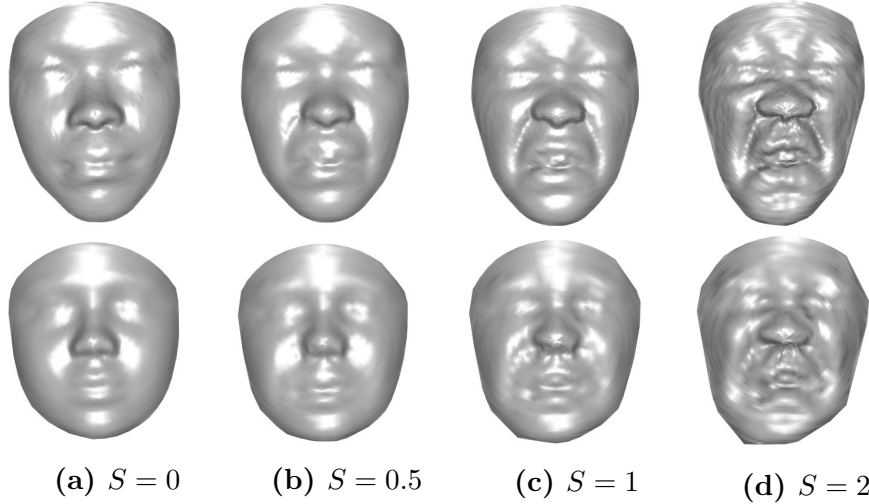
---



**Figure 6.6:** Curve matching for a relatively small deformation under three different  $\lambda$  values. Note that, the impact of  $\lambda$  is minimal on the matching results since the deformation is relatively small.



**Figure 6.7:** Curve matching for a relatively large deformation under three different  $\lambda$  values. Contrary to Figure 6.6, the value of  $\lambda$  has a significant impact on the matching result since the deformation is relatively large.



**Figure 6.8:** Given a vector form of an expression  $\mathcal{L}(\mathcal{D}_E)$ , the figure shows the action of  $\mathcal{E}(S \times \mathcal{L}(\mathcal{D}_E))$  on two different neutral faces for different scales  $S$ . In (a) the scale  $S = 0$ , thus the face remains neutral.

we use  $\mathcal{E}$  to denote the direct product of (6.28). Thus  $\mathcal{E}(\mathcal{L}(\mathcal{D}_E)) = \mathcal{D}_E$ . Consequently, using  $\mathcal{L}$  one can train linear discriminate models, e.g., SVM, on the Lie algebra. Alternatively, a linear combination or scaling of a linearized expression can be mapped back to the Lie group using  $\mathcal{E}$ , see Figure. 6.8.

## 6.6 Experiments

In Section 6.6, we evaluate the proposed approach on the BU-3DFE dataset [YWS<sup>+</sup>06] for facial expression recognition. The dataset contains a neutral face and 6 different expressions, *anger* (AN), *happiness* (HA), *surprise* (SU), *fear* (FE), *sadness* (SA), and *disgust* (DI). The expressions are collected from 100 subjects (56 female and 44 male) of different race and age. Each expression has different levels of intensity, ranging from 1 to 4; the most intense one is labelled 4. In what follows, we detail the experimental setup along with results and comparisons.

## 6. 3D FACIAL EXPRESSION ANALYSIS

---

### 6.6.1 Experimental setup

There are usually three main experimental scenarios that are performed using BU-3DFE dataset for evaluating a modelling approach in expression recognition. The scenarios are summarized as follows:

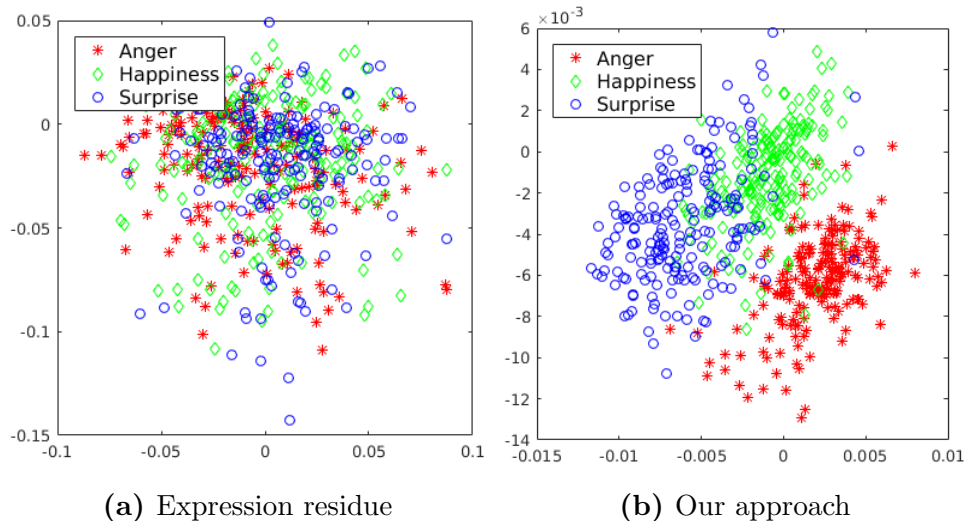
- i. Select 60 subjects, split the dataset into 10 sections and use the 54 out of 60 for training and the remaining 6 for testing.
- ii. Select 60 subjects, split the dataset into 10 sections and use the 54 out of 60 for training and the remaining 6 for testing. Repeat the experiment 100 times.
- iii. Randomly select 60 subjects, split the dataset into 10 sections and use the 54 out of 60 for training and the remaining 6 for testing. Repeat the experiment 100 times. Except in this case the subjects are randomly select in each round.

In this thesis, we are conducting experiment type (iii) 20 times instead of 100. Consequently, we will mainly compare our approach with methods evaluated with experimental type (ii) and experimental type (iii).

Given neutral faces of subjects, we duplicate the experimental scenario discussed in [BADDB11], experimental scenario (iii). We select 60 subjects randomly with all the 6 expressions in two intensities (3 and 4). Out of the 60 subjects 54 subjects are selected as a training data and the left out 6 are reserved for testing. The process is repeated 10 times by dissecting the 60 subjects into different training and testing groups, similar to 10 fold cross validation. Next, the overall process is repeated 100 times. In our case, we repeat the overall process 20 times. Hence, we perform, in total, 200 times training and testing.

Once we randomly select 60 subjects for 20 times, we prepare the selected dataset under different parameters and curve sampling settings. In all facial surface representations, we fix the number of points representing curves  $z = 50$ . However, we prepare all facial surfaces for two different numbers of curve values, i.e., for  $k = 50$  and  $k = 100$ . We call these datasets D-1 and D-2, respectively. Subsequently, for both datasets, D-1 and D-2, we select the  $k$  curves optimally,





**Figure 6.9:** Two-dimensional expression space computed from three expressions with PCA. (a) PCA on expressions extracted as expression residues. (b) PCA on the Lie algebra of expressions extracted with our approach.

as described in Section 6.4, and uniformly as described in Section 6.3. Hence in total, we prepare 4 datasets, i.e., D-1-U, and D-1-O (D-1 with uniform and optimal curve sampling, respectively), and D-2-U and D-2-O (D-2 with uniform and optimal curve sampling, respectively). In the optimal sampling case, all neutral faces are optimally sampled to  $k$  curves with respect to a randomly selected and uniformly sampled neutral reference face. Next, faces with expression are optimally sampled with respect to their respective optimally sampled neutral faces. In all of the optimal sampling  $\alpha = 1$  and  $\lambda = 0.7$ . For every training and testing phase, SVM with linear kernel is trained on the Lie algebra, see Section 6.5. Classification is done on the Lie algebra in one-vs-all classification scenario.

### 6.6.2 Linear versus the proposed expression space

Most methods, evaluated on the BU-3DFE dataset, are based on features that are extracted from annotated/estimated landmark regions. As a result, to demonstrate the representativeness of our approach, in comparison with landmark-free linear expression representation, we replicate a simple PCA based expression representation as described in [AOBM09]. To build a PCA based expression space,

## 6. 3D FACIAL EXPRESSION ANALYSIS

---

we take facial surfaces that are aligned and parametrized as described in Section 6.3. Facial expressions are then decoupled from neutral faces by taking the point-to-point difference, which gives the expression residue. That is for a neutral face  $\Gamma_N$  and a face with an expression  $\Gamma_E$ , the expression residue  $E_{res} = \Gamma_E - \Gamma_N$ . Subsequently, the expression space is estimated with a subspace spanned by 30 principal components computed from the expression residues with PCA. All expression residues are then projected on to the expression space where SVM is used, with linear kernel, to train and classify expressions in a experimental scenario (iii) which is described in Subsection 6.6.1. Note that the expression space estimation is done in each training and testing phase. Recognition based on expression residue scored 43.79 % average accuracy. In comparison, our approach performs much better, see Table 6.1. The main reason for such a large difference in performance is the Lie group based expression representation which disentangles different expressions when mapped to the Lie algebra. To illustrate this, we compute a two-dimensional, with PCA, expression space of *anger*, *happiness*, and *sadness* from the whole BU-3DFE dataset with 3 and 4 intensities. As shown in Figure 6.9, expressions extracted with our approach are more separately clustered as compared to expression residues.

### 6.6.3 Results

In Subsection 6.6.3, we present results of our approach and existing expression recognition methods that follow the same experimental scenario as [BADDB11]. As shown in Table 6.1, our approach outperformed state of the art method [BADDB11], when an SVM is trained on expressions extracted from D-2-O dataset, see Subsection 6.6.1. The lowest accuracy rate of our approach is on expressions extracted from D-1-U. However, accuracy rate improves as the number of curves is increased, see Table 6.2 and Table 6.4. This is mainly because a dense set of curves approximate the facial surface more closely. Consequently, subtle details of an expression are more likely to be captured from a dense set of curves than sparse. Additionally, optimal sampling of the curves improves performance, regardless of the curve number, see Table 6.3 and Table 6.5. As argued in Section 6.4, expressions extracted from optimally sampled curves are more representative than

## 6.7 Conclusion

Methods	Landmarks	Expression features	Classifier	Performance		
				i	ii	iii
[WYWS06]	Yes	Primitive surface features	LDA	83.60%	61.79%	-
[SD07]	Yes	Distance of landmarks	MLP	91.30%	67.52%	-
[GWLT09]	No	Region based depth difference	SVM	-	76.22%	-
[BDBP <sup>+</sup> 10]	Yes	SIFT features	SVM	-	-	77.54%
[BADDDB11]	Yes	SIFT features	SVM	-	-	78.43%
[ZLC <sup>+</sup> 13]	Yes	Mean curvature + Conformal factor	SRC	-	-	70.93%
Ours on D-1-U	No	No features	SVM	-	-	76.45%
Ours on D-1-O	No	No features	SVM	-	-	77.91%
Ours on D-2-U	No	No features	SVM	-	-	78.14%
<b>Ours on D-2-O</b>	No	No features	SVM	-	-	<b>79.16%</b>

**Table 6.1:** Comparison between performances of the proposed approach and recent results. We highlight the top score at the bottom.

uniformly sampled ones. Compare Table 6.2 against Table 6.3, and Table 6.4 against Table 6.5. Nevertheless, in all our experiments, *fear* is largely confused with *happiness*.

## 6.7 Conclusion

In Chapter 6, we introduced a new deformation-based facial expression representation. The representation is based on a mapping function that identifies a set of facial curves with an element of a high dimensional matrix Lie group. Furthermore, an algorithm for facial curve correspondence estimation is proposed. To validate the proposed representation, SVM is trained on the Lie algebra of the expression representation space. The results outperformed state of the art methods evaluated on the BU-3DFE dataset. Nevertheless, there are areas where the approach can be improved. First, accurate estimation of world coordinate system (alignment) impacts the performance of the proposed representation, hence improving the coordinate alignment of a dataset is important, e.g., semi-rigid ICP. Second, improving the time complexity of the correspondence estimation algorithm is very important so that correspondence between faces can be computed without restriction, i.e., correspondence estimation for both curves and points. Such a solution can, theoretically, lead to a dense point correspondence

## 6. 3D FACIAL EXPRESSION ANALYSIS

---

%	AN	HA	SU	FE	SA	DI
<b>AN</b>	<b>67.96</b>	1.17	0.58	1.50	18.67	10.12
<b>HA</b>	0.04	<b>95.92</b>	1.08	2.71	0.00	0.25
<b>SU</b>	0.00	1.04	<b>95.83</b>	1.25	1.42	0.46
<b>FE</b>	5.42	22.21	10.25	<b>47.75</b>	8.21	6.17
<b>SA</b>	12.21	2.04	2.58	3.79	<b>78.83</b>	0.54
<b>DI</b>	6.58	6.21	6.46	4.92	3.42	<b>72.42</b>
Average= 76.45%						

**Table 6.2:** Confusion matrix on expressions extracted from faces with uniformly sampled 50 curves (D-1-U).

%	AN	HA	SU	FE	SA	DI
<b>AN</b>	<b>65.83</b>	0.83	0.00	3.33	19.17	10.83
<b>HA</b>	0.00	<b>95.00</b>	2.50	0.83	0.00	1.67
<b>SU</b>	0.00	0.83	<b>95.00</b>	1.67	1.67	0.83
<b>FE</b>	3.33	15.00	10.83	<b>56.67</b>	9.17	5.00
<b>SA</b>	13.33	0.83	2.50	3.33	<b>80.00</b>	0.00
<b>DI</b>	5.00	8.33	6.67	3.33	1.67	<b>75.00</b>
Average= 77.91%						

**Table 6.3:** Confusion matrix on expressions extracted from faces with optimally sampled 50 curves (D-1-O).

%	AN	HA	SU	FE	SA	DI
AN	<b>65.83</b>	0.83	0.00	0.83	22.50	10.00
HA	0.00	<b>97.50</b>	1.67	0.83	0.00	0.00
SU	0.00	0.83	<b>94.17</b>	1.67	2.50	0.83
FE	5.83	19.24	10.83	<b>54.64</b>	4.17	5.00
SA	15.83	0.83	1.67	3.33	<b>77.50</b>	0.83
DI	4.17	7.50	3.33	3.33	2.50	<b>79.17</b>
Average= 78,14%						

**Table 6.4:** Confusion matrix on expressions extracted from faces with uniformly sampled 100 curves (D-2-U).

%	AN	HA	SU	FE	SA	DI
AN	<b>72.50</b>	0.00	0.83	4.17	16.67	5.83
HA	0	<b>93.33</b>	0.83	5.00	0.00	0.83
SU	0.00	0.83	<b>95.83</b>	2.50	0.83	0.00
FE	5.00	23.33	5.83	<b>50.83</b>	7.50	7.50
SA	13.33	1.67	2.50	1.67	<b>80.83</b>	0.00
DI	3.33	5.00	5.83	1.67	2.50	<b>81.67</b>
Average= 79,16%						

**Table 6.5:** Confusion matrix on expressions extracted from faces with optimally sampled 100 curves (D-2-O).

## 6. 3D FACIAL EXPRESSION ANALYSIS

---

estimation between faces without the need for training dataset.

# Chapter 7

## Conclusion and outlook

### 7.1 Conclusion

In this thesis, we have introduced a novel shape representation framework that leads to closed form geodesic distance and geodesic deformation equations, see Chapter 3. We have formalized optimal parameter estimation in terms of optimal point sampling. Subsequently, we have introduced a linearity assumption into the possible reparameterization functions and estimated the optimal parameters using a nested loop brute force search. Later on, a more general class of reparameterizations are introduced by insisting on the preservation of a geometric quantity, area, and the solutions are estimated using dynamic programming. In both cases, we ensure a one-to-one point correspondence by restricting the search space of optimal parameters to only injective and monotonic functions, which is not the case in other infinite dimensional settings. Furthermore, we have introduced a symmetric optimal parameter estimation by optimally sampling both curve arguments with respect to one another instead of fixing one curve and sampling the other, see Chapter 4.

In addition to providing a similarity measure between curved shapes, in Chapter 5 we have shown that several parametric statistical models can readily be adopted to the introduced curved shape representation space. Thereby, opening the possibility of modelling the distribution of shape categories with a wide range of variability. Such models can be used to explicitly model a specific shape

## 7. CONCLUSION AND OUTLOOK

---

category to later be used for recognition and simulation tasks. Finally, an application of the introduced curved shape representation, apart from contour based object recognition, is presented in face and facial expression analysis from 3D point cloud data in Chapter 6.

Nevertheless, there are several potential directions to improve and generalize the proposed curved shape representation approach and its application domains. In Section 7.2, we present possible future work directions. Although several application domains can be suggested without a substantial extension of what is proposed here, we focus on those that we believe will have a significant impact and require additional research.

### 7.2 Outlook

Although the proposed representation demonstrates reasonable accuracy in most contour based object recognition problems, one can potentially generalize the approach to handle a wide range of problems that are defined by a given objective functional or by a given mathematical condition. Herein, we describe potential directions to extend the proposed approach and provide a general sketch on how the extension ideas can be incorporated with the proposed curve representation.

**Constrained shape space and deformation:** As indicated in Chapter 3, constraints can be introduced into the curve representation space  $SE(n)^z/SO^*(n)$ . For instance, consider a set of closed curves that have a constant enclosed area, i.e.  $\mathcal{C}^A = \{\tilde{c} \in \mathcal{C} : A(\tilde{c}) = K\}$ , where  $K$  is a given scalar constant and  $A(\cdot)$  is an area function as defined in (4.12) in Chapter 4. Subsequently, one would like to study the geodesic deformation and distance between shapes in  $\mathcal{C}^A$  in such a way that the geodesic deformation preserves the defined constant area constraint. Since  $\mathcal{C}^A$  is a subset of the set  $\mathcal{C}$ , a geodesic equation and distance that preserve the constraint are assumed to be more descriptive of the subset as opposed to a the geodesic equation and distance functions defined in  $\mathcal{C}$ .

In effect, the representation of the shapes in  $\mathcal{C}^A$  is a subset of the original representation space  $SE(n)^z/SO^*(n)$ , if not a subgroup. As a result, solving for a



geodesic curve and geodesic distance in the newly defined constrained representation space might not be as simple as it was in  $\text{SE}(\mathfrak{n})^z/\text{SO}^*(\mathfrak{n})$ . Even worse, the constrained representation space might not be connected. However, transferring constraints defined in  $\mathcal{C}$  to the proposed curve representation space opens up the possibility of studying different shape sets, defined by different constraints, exclusively. Hence, the introduction of a constrained representation and deformation are desirable properties, even if they might introduce computational overhead and theoretical difficulties. Although the existence of a geodesic curve depends on the particular structure of the constrained representation space, we can devise an optimization problem to approximate the characteristics of a geodesic curve in a constrained representation space. To that end, consider the following objective functional for a constrained representation space defined by  $\mathcal{C}^A$ ,

$$\arg \min_{\Phi} \int \sqrt{\langle \dot{\Phi}(t), \dot{\Phi}(t) \rangle_{\Phi(t)}} + \lambda \left| \frac{dA(f^{-1}(\Phi(t)))}{dt} \right| dt, \quad (7.1)$$

where  $\lambda$  is a scalar regularisation term, and  $f^{-1}(\cdot)$  is as defined in (3.20) in Chapter 3. The sought after deformation equation is denoted by  $\Phi(\cdot)$  such that  $f^{-1}(\Phi(0)) = \tilde{c}_1$  and  $f^{-1}(\Phi(t)) = \tilde{c}_2$  for some  $\tilde{c}_1, \tilde{c}_2 \in \mathcal{C}^A$ . Note that,  $A(f^{-1}(\Phi(t)))$  might not be equal to  $K$ ,  $\forall t \in [0, 1]$ , since the solution is an approximation. The first term of (7.1) penalizes the cost of the deformation which is the same as the optimization problem defined in (3.31), while the second term insists on keeping the area of the curve  $\tilde{c}_1$  fixed while deforming. The solution for (7.1) can be estimated iteratively, unlike the geodesic equation in  $\text{SE}(\mathfrak{n})^z/\text{SO}^*(\mathfrak{n})$  which has a closed form solution. Nevertheless, since (7.1) is being solved in the original representation space  $\text{SE}(\mathfrak{n})^z/\text{SO}^*(\mathfrak{n})$  we can initialize its optimization by (3.49).

In summary, one of the main potential future directions of the proposed approach is to study its generalization to various kinds of constrained shape sets and their constrained representation space. Although computational taxing, we expect a constrained representation to be geometrically true and descriptive of the shape category in contrast to using a general representation space for describing a constrained family of curves. To the best of our knowledge, the problem of representing and deforming constrained shapes has not been explored in any of the earlier works.

## 7. CONCLUSION AND OUTLOOK

---

**Optimal metric:** The cost function for estimating optimal parameters, which is defined in (4.14) in Chapter 4, is controlled by three parameters  $\alpha$ ,  $\lambda$ , and  $\eta$ . In all of the experimental evaluations, the values of these free parameters are estimated manually. However, one would assume that for different shape categories the values of the parameters are different. For instance, one could interpret the parameter  $\eta$  as elasticity constraint and since different shape categories exhibit different elasticity characteristic,  $\eta$  can be seen as a category specific parameter. Hence, these parameters should be estimated per shape category instead of manual tuning. One possible solution for the estimation of category specific parameters is to discretize the possible values of the parameters and estimate the optimal value by a brute force searching. Since the free parameters are only three, a discrete brute force might not be as computationally taxing as in high dimensional space. Alternatively, one can see (4.14) as defining a family of metrics that vary by their parameter values. Subsequently, the problem of parameter estimation can be framed in terms of category specific metric learning.

In general, we believe that estimating the free parameters from a dataset will lead to a performance boost in shape based object modelling and object recognition. Furthermore, given the estimation of the free parameters, a semantic association can be made between shapes that fail to achieve optimality under the learned parameter values and unnatural deformations of the shape category. For instance, one could associate suboptimal solutions to be caused by unnatural deformations of the shape category there by detecting a natural shape deformation, e.g., articulation, from a deformation due to artefacts, e.g. occlusion. This, of course, is assuming that the free parameters are estimated from an occlusion free dataset.

**Segmentation:** Most of the applications discussed in this thesis are concerned with object recognition from its contour or silhouette. However, the most ubiquitous visual data is in the form of an image. Hence, one assumes a prior segmentation or object contour detection procedure while designing a learning algorithm based on the proposed representation framework. Nevertheless, the problem of detecting the object's silhouette from an image, which is broadly known as segmentation, can potentially be framed as a problem of estimating an evolution

equation of a curve represented by the proposed framework. Consequently, one possible future research direction is to consider the segmentation of a given image into two connected regions using the defined curved shape representation. To that end, it is possible to adapt an optimization based approach that evolves a given curve representation such that the curve divides an image into two connected regions that exhibit a smooth color variation. In fact, there are several kinds of energy functionals that are designed to estimate an image segmentation into different regions, e.g., Geometric/Sobolev active contour models [You10a].

Nevertheless, an evolution equation of a curve, represented by the proposed approach, has to allow global linear transformations as well as local. That is, the starting point of a curve should not necessarily be fixed; it should be able to translate and rotate. For instance, a potential evolution equations can take the following form

$$f(\tilde{c})_{i+1} = (G_i^L \cdot \mathbb{G}_i) f(\tilde{c})_i, \quad (7.2)$$

where  $\mathbb{G} \in \text{SE}(n)^z$  such that  $\mathbb{G}_i \mathbb{G}_j^{-1} = e, \forall i, j \in [1, z]$ . In other words,  $\mathbb{G}$  represents the global linear transformation of the shape. Subsequently, it is possible to design a functional, that takes a solution of the form (7.2), such that a suitable energy functional, e.g., Mumford-Shah functional [TYW01, MS89], is minimized.

**Pose estimation:** Apart from deformation transporting and facial expression representation, a factored deformation can be used to model one of the main important variations of objects; variation due to pose. Although there is no reason to expect object variation, due to pose, to form a subgroup in the representation space, given a template shape it is possible to linearize its variations and model it as a subspace or set of subspaces at the tangent space of the template shape. For example, PCA can be used to model such variations, see Chapter 5. Such a subspace can be used to simulate and recognize object variation due to pose using subspace clustering algorithms [Vid11].

More interestingly, however, a linear predictor can be learned to estimate the pose of an object, given its deformation from a template shape, and vice-versa, estimate its deformation given a pose. To that end, let  $T_{f(\tilde{c}_i)} \mathbf{V}$  be the estimated

## 7. CONCLUSION AND OUTLOOK

---

subspace from a dataset that contains variation of the template shape  $f(\tilde{c}_t)$  due to different poses. Subsequently, if each of the template shape variations are labelled with azimuth and elevation  $(\theta, \phi)$  then the label space is a sphere  $\mathbf{S}^2$ . Furthermore, the labels can be linearized using the exponential maps, see Appendix A, to the tangent space of the template label  $T_{(\theta_t, \phi_t)}\mathbf{S}^2$ . Finally, if we assume that the data generating function, the function that maps pose to a shape deformation, is an immersion defined as

$$\mathbf{F} : \mathbf{S}^2 \rightarrow \text{SE}(\mathbf{n})^z / \text{SO}^*(\mathbf{n}), \quad (7.3)$$

then its Jacobian at  $(\theta_t, \phi_t)$  is an injective linear map defined as

$$D_{(\theta_t, \phi_t)}\mathbf{F} : T_{(\theta_t, \phi_t)}\mathbf{S}^2 \rightarrow T_{f(\tilde{c}_t)}\mathbf{V}. \quad (7.4)$$

Hence, the problem of deformation estimation from the pose of the object is formulated as estimation of (7.4) from the linearized dataset and pose labels. Additionally, the estimation of (7.4) can be tied with the subspace  $T_{f(\tilde{c}_t)}\mathbf{V}$  estimation using what are known as supervised dimension reduction methods [RGC<sup>+</sup>08]. Furthermore, we believe (7.4) can provide a reasonably accurate result, assuming the deformations of the template shape are factored from a dataset that is optimally matched. Such optimal matching is often difficult in application domains where an image might be composed of more than one object often with occlusion. Consequently, the main work, here, is to consider simple object pose models like (7.4) in conjugation with additional information like colour.

# Appendix A

## Exponential maps

Here we briefly cover the notion of exponential map in Lie groups. In all subsequent discussions  $G$  denotes a given matrix Lie group and  $\mathfrak{g}$  denotes its Lie algebra.

Consider a given vector  $V$  that is an element of a Lie algebra  $\mathfrak{g}$ . Subsequently, we want to compute a geodesic curve that goes through the group identity,  $e$ , and follows the direction of  $V$ ; the scale of  $V$  determines the length of the geodesic curve with its respective metric. Such a curve is given by what is known as exponential map and is written as

$$\exp(tV) = g(t), \quad \text{s.t. } g(t) \in G, \quad (\text{A.1})$$

where  $t \in \mathbf{R}$  denotes the scale of the vector  $V$ . The exponential map is not necessarily defined over all of the Lie algebra, unless the group is compact or the algebra is metrically complete. Moreover, the notion of exponential maps is not limited to Lie groups, it can be defined in any Riemannian manifold with a given connection. However, in case of matrix Lie groups we can use the left or right translation invariance as a connection. For instance, a left translation invariant vector field is defined as

$$\frac{dg(t)}{dt} = D_e L_{g(t)} V, \quad (\text{A.2})$$

where  $D_e L_{g(t)}$  denotes the derivative of the left translation map at  $e$ —the left translation is defined as  $L_{g(t)} : g \mapsto g(t)g, \forall g \in G$ . As a result, exponential

## A. EXPONENTIAL MAPS

---

maps defined using left translation invariance also define what are known as 1-parameter subgroups. That is

$$\exp((t_1 + t_2)V) = \exp(t_1V) \exp(t_2V). \quad (\text{A.3})$$

Note that  $\exp(-tV) = \exp(tV)^{-1}$ . In a matrix Lie group, maps of the form (A.1) coincide with the exponentiation of a matrix defined as

$$\exp(V) = \sum_{i=0}^{\infty} \frac{V^i}{i!}. \quad (\text{A.4})$$

Furthermore, since matrix exponentiation is an invertible map one can define a map from the 1-parameter subgroups back to the Lie algebra  $\mathfrak{g}$  as follows

$$\log(\exp(tV)) = tV. \quad (\text{A.5})$$

**Lie bracket:** Most matrix Lie groups do not commute, i.e.  $gh \neq hg$ . A measure of the noncommutativity is defined by what is known as commutator:  $[g_1, g_2] = g_1^{-1}g_2^{-1}g_1g_2$ , for some  $g_1, g_2 \in G$ . A similar notion can be defined for vector fields  $X, Y$  that are defined as

$$X : G \rightarrow \bigsqcup_{\forall g \in G} T_g G, \quad (\text{A.6})$$

where  $\bigsqcup$  denotes a disjoint union. In general, vector fields can be understood as differential operators that takes a smooth function  $f$ , defined on  $G$  and takes values in  $\mathbf{R}$ , and gives its derivative. In such a case, the measure of the commutativity is given as

$$[X, Y](f) = X(Y(f)) - Y(X(f)), \quad (\text{A.7})$$

and is known as Lie bracket of the vector fields evaluated for  $f$ . For instance, consider the vector field defined by  $\exp(tX)$  and  $\exp(tY)$ . The Lie bracket of the vector fields is defined by the following at  $t = 0$  or at the identity.

$$[X, Y] = \frac{d^2}{dt^2} \exp(tX) \exp(tY) \exp(-tX) \exp(-tY), \quad (\text{A.8})$$

observe the striking similarity of the Lie brackets with how the commutator is defined.

## Appendix B

# Computing optimal rigid transformation

The optimal rotation matrix, i.e., the one closest to the identity amongst all possible rotations, between two vectors  $p_1, p_2 \in \mathbf{R}^2$  can be computed by minimizing

$$\min_{R \in SO(2)} \|Rp_1 - p_2\|_2^2. \quad (\text{B.1})$$

The solution for (B.1) is unique unless the points are collinear [Kab76]. Furthermore, in  $\mathbf{R}^2$  the rotation plane is known and is fixed. Subsequently, the solution of (B.1) is given as  $R = VU^T$  such that the covariance of the points is  $C = p_1^T p_2 = U\Sigma V^T$ . However, the solution might include a reflection and needs to be rectified, see [Kab76, Kan94]. Nevertheless, in a high dimensional space, rotation is not the same as coordinate orientation. Moreover, the rotation plane between two points is not known. As a result, (B.1) does not necessarily give a high-dimensional optimal rotation matrix that preserves the coordinate orientation.

Alternatively, we can compute the rotation between  $p_1^n, p_2^n \in \mathbf{R}^n$  by estimating the rotation plane first and then using (B.1). The rotation plane is estimated by computing the main principal components, denoted by  $B$ , of the data set ( $p_1^n$  and  $p_2^n$ ) using SVD (singular value decomposition)– the plane spanned by the two eigenvectors with the largest eigenvalues is taken as the rotation plane. Next, using (B.1) optimal rotation  $R$  from  $p_1$  to  $p_2$ – these are the orthogonal

## B. COMPUTING OPTIMAL RIGID TRANSFORMATION

---

projections of  $p_1^n$  and  $p_2^n$  onto the estimated rotation plane– is computed. Finally, we represent the computed  $R \in \text{SO}(2)$  in homogeneous coordinates as  $R_h \in \text{SO}(n)$  and conjugate it with the basis transformation to get its equivalent in our original coordinate frame as  $\mathcal{R} = BR_hB^T$ . Hence,  $\mathcal{R}$  gives us coordinate preserving optimal rotation matrix.

Once the optimal rotation matrix from  $p_1^n$  to  $p_2^n$  is computed, the optimal translation vector is computed directly as

$$t = p_2^n - \mathcal{R}p_1^n. \tag{B.2}$$



# Bibliography

- [AGP91] Yali Amit, Ulf Grenander, and Mauro Piccioni. Structural image restoration through deformable templates. *Journal of the American Statistical Association*, 86(414):376–387, 1991. 24
- [AK10] Djamila Aouada and Hamid Krim. Squigraphs for fine and compact modeling of 3D shapes. *IEEE Transactions on Image Processing*, 19(2):306–321, 2010. 3
- [ALM04] T Ando, Chi-Kwong Li, and Roy Mathias. Geometric means. *Linear algebra and its applications*, 385:305–334, 2004. 78
- [Ami94] Yali Amit. A nonlinear variational problem for image matching. *SIAM Journal on Scientific Computing*, 15(1):207–224, 1994. 43
- [AMS09] P-A Absil, Robert Mahony, and Rodolphe Sepulchre. *Optimization algorithms on matrix manifolds*. Princeton University Press, 2009. 79
- [AOBM09] F Al-Osaimi, Mohammed Bennamoun, and Ajmal Mian. An expression deformation approach to non-rigid 3D face recognition. *International Journal of Computer Vision*, 81(3):302–316, 2009. 101, 102, 117
- [BADDDB11] Stefano Berretti, Boulbaba Ben Amor, Mohamed Daoudi, and Alberto Del Bimbo. 3D facial expression recognition using SIFT descriptors of automatically detected keypoints. *The Visual Computer*, 27(11):1021–1036, 2011. 116, 118, 119

## BIBLIOGRAPHY

---

- [BBBK08] Alexander M Bronstein, Michael M Bronstein, Alfred M Bruckstein, and Ron Kimmel. Analysis of two-dimensional non-rigid shapes. *International Journal of Computer Vision*, 78(1):67–88, 2008. 69
- [BBM16] Martin Bauer, Martins Bruveris, and Peter W Michor. Why use sobolev metrics on the space of curves. In *Riemannian computing in computer vision*, pages 233–255. Springer, 2016. 4, 9, 20, 21, 65
- [BBMM14] Martin Bauer, Martins Bruveris, Stephen Marsland, and Peter W Michor. Constructing reparameterization invariant metrics on spaces of plane curves. *Differential Geometry and its Applications*, 34:139–165, 2014. 20, 21
- [BCV13] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013. 101
- [BD15] Richard E Bellman and Stuart E Dreyfus. *Applied dynamic programming*. Princeton university press, 2015. 58
- [BDBP<sup>+</sup>10] Stefano Berretti, Alberto Del Bimbo, Pietro Pala, Boulbaba Ben Amor, and Mohamed Daoudi. A set of selected SIFT features for 3D facial expression recognition. In *Pattern Recognition (ICPR), 2010 20th International Conference on*, pages 4125–4128. IEEE, 2010. 119
- [Bel56] Richard Bellman. Dynamic programming and lagrange multipliers. *Proceedings of the National Academy of Sciences*, 42(10):767–769, 1956. 112
- [Ber05] Dennis S Bernstein. *Matrix mathematics: Theory, facts, and formulas with application to linear systems theory*, volume 41. Princeton University Press Princeton, 2005. 91
- [Bes75] Julian Besag. Statistical analysis of non-lattice data. *The statistician*, pages 179–195, 1975. 84

- [BG05] Ingwer Borg and Patrick JF Groenen. *Modern multidimensional scaling: Theory and applications*. Springer Science & Business Media, 2005. [xii](#), [80](#)
- [Bha09] Rajendra Bhatia. *Positive definite matrices*. Princeton University Press, 2009. [39](#)
- [Bis06] Christopher M Bishop. *Pattern recognition and machine learning*. Springer, 2006. [85](#), [90](#), [94](#)
- [BL08] Xiang Bai and Longin Jan Latecki. Path similarity skeleton graph matching. *IEEE transactions on pattern analysis and machine intelligence*, 30(7):1282–1292, 2008. [67](#), [69](#)
- [BMP02] Serge Belongie, Jitendra Malik, and Jan Puzicha. Shape matching and object recognition using shape contexts. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(4):509–522, 2002. [14](#), [48](#), [68](#), [69](#)
- [BMP10] Dario Bini, Beatrice Meini, and Federico Poloni. An effective matrix geometric mean satisfying the Ando-Li-Mathias properties. *Mathematics of Computation*, 79(269):437–452, 2010. [78](#)
- [BMTY05] M Faisal Beg, Michael I Miller, Alain Trouvé, and Laurent Younes. Computing large deformation metric mappings via geodesic flows of diffeomorphisms. *International journal of computer vision*, 61(2):139–157, 2005. [22](#)
- [Boo84] Fred L Bookstein. A statistical method for biological shape comparisons. *Journal of Theoretical Biology*, 107(3):475–520, 1984. [14](#)
- [Boo86] Fred L Bookstein. Size and shape spaces for landmark data in two dimensions. *Statistical Science*, pages 181–222, 1986. [14](#)
- [Boo94] Fred L Bookstein. The morphometric synthesis: a brief intellectual history. *Frontiers in Mathematical Biology*, pages 212–237, 1994. [14](#)

## BIBLIOGRAPHY

---

- [BR<sup>+</sup>71] Robert E Blackith, Richard Arthur Reyment, et al. *Multivariate morphometrics*. 1971. 14
- [BVS<sup>+</sup>96] Marian Stewart Bartlett, Paul A Viola, Terrence J Sejnowski, Beatrice A Golomb, Jan Larsen, Joseph C Hager, and Paul Ekman. Classifying facial action. *Advances in neural information processing systems*, pages 823–829, 1996. 100
- [CAS92] Isaac Cohen, Nicholas Ayache, and Patrick Sulger. Tracking points on deformable objects using curvature information. In *Computer Vision ECCV'92*, pages 458–466. Springer, 1992. 54
- [CHFT06] Ya Chang, Changbo Hu, Rogerio Feris, and Matthew Turk. Manifold based analysis of facial expression. *Image and Vision Computing*, 24(6):605–614, 2006. 101
- [CTCG95] Timothy F Cootes, Christopher J Taylor, David H Cooper, and Jim Graham. Active shape models-their training and application. *Computer vision and image understanding*, 61(1):38–59, 1995. 23
- [CVTV05] Ya Chang, Marcelo Vieira, Matthew Turk, and Luiz Velho. Automatic 3D facial expression analysis in videos. In *International Workshop on Analysis and Modeling of Faces and Gestures*, pages 293–307. Springer, 2005. 100
- [DADS10] Hassen Drira, Boulbaba Ben Amor, Mohamed Daoudi, and Anuj Srivastava. Pose and expression-invariant 3D face recognition using elastic radial curves. In *British machine vision conference*, pages 1–11, 2010. 3, 101
- [DAO15] Girum Demisse, Djamila Aouada, and Björn Ottersten. Template-based statistical shape modelling on deformation space. In *22nd IEEE International Conference on Image Processing*, 2015. 82

- [DAO16] Girum Demisse, Djamila Aouada, and Björn Ottersten. Similarity metric for curved shapes in Euclidean space. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 42
- [DAO17] Girum G Demisse, Djamila Aouada, and Bjorn Ottersten. Deformation based curved shape representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017. 42
- [DBAS<sup>+</sup>13] Hassen Drira, Boulbaba Ben Amor, Anurag Srivastava, Meroua Daoudi, and Rim Slama. 3D face recognition under expressions, occlusions, and pose variations. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(9):2270–2283, 2013. 101
- [dCV92] Manfredo Perdigao do Carmo Valero. *Riemannian geometry*. 1992. 37
- [DM98] Ian L Dryden and Kanti V Mardia. *Statistical shape analysis*, volume 4. Wiley Chichester, 1998. 3, 26, 82
- [DT08] Mohammad Reza Daliri and Vincent Torre. Robust symbolic representation for shape recognition and retrieval. *Pattern Recognition*, 41(5):1782–1798, 2008. 68, 95
- [DT10] Mohammad Reza Daliri and Vincent Torre. Shape recognition based on kernel-edit distance. *Computer Vision and Image Understanding*, 114(10):1097–1103, 2010. 95
- [Dur10] Rick Durrett. *Probability: theory and examples*. Cambridge university press, 2010. 76, 78
- [D92] W Thomspou Darcy. On growth and form: The complete revised edition, 1992. 22
- [EF77] Paul Ekman and Wallace V Friesen. Facial action coding system. 1977. 100

## BIBLIOGRAPHY

---

- [FB12] Oren Freifeld and Michael J Black. Lie bodies: A manifold representation of 3D human shape. In *Computer Vision–ECCV 2012*, pages 1–14. Springer, 2012. 43
- [FFFP07] Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories. *Computer vision and Image understanding*, 106(1):59–70, 2007. ix, 2
- [FS07] Pedro F Felzenszwalb and Joshua D Schwartz. Hierarchical matching of deformable shapes. In *Computer Vision and Pattern Recognition, 2007. CVPR’07. IEEE Conference on*, pages 1–8. IEEE, 2007. 50
- [FZO<sup>+</sup>11] Tianhong Fang, Xi Zhao, Omar Ocegueda, Shishir K Shah, and Ioannis A Kakadiaris. 3D facial expression recognition: A perspective on promises and challenges. In *Automatic Face & Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on*, pages 603–610. IEEE, 2011. 100
- [GK93] Ulf Grenander and Daniel Macrae Keenan. On the shape of plane images. *SIAM Journal on Applied Mathematics*, 53(4):1072–1094, 1993. 24
- [GM94] Ulf Grenander and Michael I Miller. Representations of knowledge in complex systems. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 549–603, 1994. 22, 24
- [GM07] Ulf Grenander and Michael I Miller. *Pattern theory: from representation to inference*. Oxford University Press, 2007. 22, 43
- [GN99] Arjun K Gupta and Daya K Nagar. *Matrix variate distributions*, volume 104. CRC Press, 1999. 82
- [Gre97] Ulf Grenander. Geometries of knowledge. *Proceedings of the National Academy of Sciences*, 94(3):783–789, 1997. 22, 24

- [GWLT09] Boqing Gong, Yueming Wang, Jianzhuang Liu, and Xiaoou Tang. Automatic facial expression recognition on a single 3D face by exploring shape deformation. In *Proceedings of the 17th ACM international conference on Multimedia*, pages 569–572. ACM, 2009. 119
- [HCSV13] Jeffrey Ho, Guang Cheng, Hesamoddin Salehian, and Baba Vemuri. Recursive Karcher expectation estimators and geometric law of large numbers. In *Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics*, pages 325–332, 2013. 78, 79
- [HE04] Greg Hamerly and Charles Elkan. Learning the k in k means. *Advances in neural information processing systems*, 16:281, 2004. 89
- [Hel62] Sigurdur Helgason. *Differential geometry and symmetric spaces*, volume 12. Academic press, 1962. 17
- [HJZG12] Rong-Xiang Hu, Wei Jia, Yang Zhao, and Jie Gui. Perceptually motivated morphological strategies for shape retrieval. *Pattern Recognition*, 45(9):3222–3230, 2012. 3, 95
- [HL07] Jihun Ham and Daniel D Lee. Separating pose and expression in face images: a manifold learning approach. *Neural Information Processing-Letters and Reviews*, 11(4):91–100, 2007. 100, 101
- [HSH14] Mehrtash T Harandi, Mathieu Salzmann, and Richard Hartley. From manifold to manifold: Geometry-aware dimensionality reduction for SPD matrices. In *European conference on computer vision*, pages 17–32. Springer, 2014. 85
- [Jay03] Edwin T Jaynes. *Probability theory: The logic of science*. Cambridge university press, 2003. 77
- [Jol86] Ian T Jolliffe. Principal component analysis and factor analysis. In *Principal component analysis*, pages 115–128. Springer, 1986. 85
- [K10] Kimia 1070-dataset. <http://vision.lems.brown.edu/content/available-software-and-databases>. Accessed: 2017-09-30. 69

## BIBLIOGRAPHY

---

- [Kab76] Wolfgang Kabsch. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography*, 32(5):922–923, 1976. [131](#)
- [Kan94] Ken-ichi Kanatani. Analysis of 3-d rotation fitting. *IEEE Transactions on pattern analysis and machine intelligence*, 16(5):543–549, 1994. [131](#)
- [Kar77] Hermann Karcher. Riemannian center of mass and mollifier smoothing. *Communications on pure and applied mathematics*, 30(5):509–541, 1977. [78](#)
- [Ken84] David G Kendall. Shape manifolds, procrustean metrics, and complex projective spaces. *Bulletin of the London Mathematical Society*, 16(2):81–121, 1984. [4](#), [5](#), [14](#)
- [Ken90] Wilfrid S Kendall. Probability, convexity, and harmonic maps with small image I: uniqueness and fine existence. *Proceedings of the London Mathematical Society*, 3(2):371–406, 1990. [79](#)
- [KMG98] N Khaneja, Michael I. Miller, and Ulf Grenander. Dynamic programming generation of curves on brain surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(11):1260–1265, 1998. [22](#), [24](#), [28](#)
- [KSMJ04] Eric Klassen, Anuj Srivastava, Washington Mio, and Shantanu H Joshi. Analysis of planar shapes using geodesic paths on shape spaces. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(3):372–383, 2004. [48](#)
- [LHMT14] Ping Liu, Shizhong Han, Zibo Meng, and Yan Tong. Facial expression recognition via a boosted deep belief network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1805–1812, 2014. [100](#)



- [LJ07] Haibin Ling and David W Jacobs. Shape classification using the inner-distance. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(2):286–299, 2007. 50, 95
- [LKSM14] Hamid Laga, Sebastian Kurtek, Anuj Srivastava, and Stanley J Miklavcic. Landmark-free statistical analysis of the shape of plant leaves. *Journal of theoretical biology*, 363:41–52, 2014. 3, 26, 47, 48, 50, 51
- [LLE00] Longin Jan Latecki, Rolf Lakamper, and T Eckhardt. Shape descriptors for non-rigid shapes with a single closed contour. In *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, volume 1, pages 424–429. IEEE, 2000. 67
- [LRK15] Sayani Lahiri, Daniel Robinson, and Eric Klassen. Precise matching of PL curves in  $\mathbf{R}^N$  in the square root velocity framework. *arXiv preprint arXiv:1501.00577*, 2015. 65
- [LS03] Bastian Leibe and Bernt Schiele. Analyzing appearance and contour based methods for object categorization. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 2, pages II–409. IEEE, 2003. xiii, 93, 94, 95
- [LSJR16] Jason D Lee, Max Simchowitz, Michael I Jordan, and Benjamin Recht. Gradient descent only converges to minimizers. In *Conference on Learning Theory*, pages 1246–1257, 2016. 96
- [LSZ10] Wei Liu, Anuj Srivastava, and Jinfeng Zhang. Protein structure alignment using elastic shape analysis. In *Proceedings of the First ACM International Conference on Bioinformatics and Computational Biology*, pages 62–70. ACM, 2010. 26
- [M<sup>+</sup>67] James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA., 1967. 88

## BIBLIOGRAPHY

---

- [Mad02] Neal Noah Madras. *Lectures on monte carlo methods*, volume 16. American Mathematical Soc., 2002. 83
- [MB93] Frank Morgan and James F Brecht. *Riemannian geometry: a beginner's guide*. Jones and Bartlett, 1993. 37
- [MCH<sup>+</sup>06] Siddharth Manay, Daniel Cremers, Byung-Woo Hong, Anthony J Yezzi, and Stefano Soatto. Integral invariants for shape matching. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(10):1602–1618, 2006. 43
- [MD10] David Mumford and Agnès Desolneux. *Pattern theory: the stochastic analysis of real-world signals*. CRC Press, 2010. 45, 54
- [Men13] Andrea CG Mennucci. Metrics of curves in shape optimization and analysis. In *Level Set and PDE Based Reconstruction Methods in Imaging*, pages 205–319. Springer, 2013. 14, 18, 43
- [Mil76] John Milnor. Curvatures of left invariant metrics on lie groups. *Advances in mathematics*, 21(3):293–329, 1976. 41
- [Mil97] John Willard Milnor. *Topology from the differentiable viewpoint*. Princeton University Press, 1997. 1
- [MM03] Peter W Michor and David Mumford. Riemannian geometries on spaces of plane curves. *arXiv preprint math/0312384*, 2003. 4, 5, 18, 20, 21
- [MM05] Peter W Michor and David Mumford. Vanishing geodesic distance on spaces of submanifolds and diffeomorphisms. *Doc. Math*, 10:217–245, 2005. 17, 21
- [MM07] Peter W Michor and David Mumford. An overview of the Riemannian metrics on spaces of curves using the Hamiltonian approach. *Applied and Computational Harmonic Analysis*, 23(1):74–113, 2007. 18, 20

- [MMS08] Iordanis Mpipieris, Sotiris Malassiotis, and Michael G Strintzis. Bilinear elastically deformable models with application to 3D face and facial expression recognition. In *Automatic Face & Gesture Recognition, 2008. FG'08. 8th IEEE International Conference on*, pages 1–8. IEEE, 2008. 101
- [MMSY07] Peter W Michor, David Mumford, Jayant Shah, and Laurent Younes. A metric on shape space with explicit geodesics. *arXiv preprint arXiv:0706.4299*, 2007. 18, 21, 26
- [MN95] Jan R Magnus and Heinz Neudecker. Matrix differential calculus with applications in statistics and econometrics. 1995. 91
- [Moa02] Maher Moakher. Means and averaging in the group of rotations. *SIAM journal on matrix analysis and applications*, 24(1):1–16, 2002. 39
- [MS89] David Mumford and Jayant Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Communications on pure and applied mathematics*, 42(5):577–685, 1989. 127
- [MSJ07] Washington Mio, Anuj Srivastava, and Shantanu Joshi. On shape of plane elastic curves. *International Journal of Computer Vision*, 73(3):307–324, 2007. 21, 22
- [Mum87] David Mumford. *The problem of robust shape descriptors*. Center for Intelligent Control Systems, 1987. 4
- [Mum91] David Mumford. Mathematical theories of shape: Do they model perception? In *San Diego, '91, San Diego, CA*, pages 2–10. International Society for Optics and Photonics, 1991. 4
- [MV15] Aravindh Mahendran and Andrea Vedaldi. Understanding deep image representations by inverting them. In *2015 IEEE conference on computer vision and pattern recognition (CVPR)*, pages 5188–5196. IEEE, 2015. 100

## BIBLIOGRAPHY

---

- [MYVB13] Sofiene Mouine, Itheri Yahiaoui, and Anne Verroust-Blondet. A shape-based approach for leaf classification using multiscale triangular representation. In *Proceedings of the 3rd ACM conference on International conference on multimedia retrieval*, pages 127–134. ACM, 2013. 47, 48, 50
- [NH98] Radford M Neal and Geoffrey E Hinton. A view of the EM algorithm that justifies incremental, sparse, and other variants. In *Learning in graphical models*, pages 355–368. Springer, 1998. 88
- [NJW<sup>+</sup>02] Andrew Y Ng, Michael I Jordan, Yair Weiss, et al. On spectral clustering: Analysis and an algorithm. *Advances in neural information processing systems*, 2:849–856, 2002. 87
- [Pan09] Maja Pantic. Machine analysis of facial behaviour: Naturalistic and dynamic behaviour. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, 364(1535):3505–3513, 2009. 99, 100
- [Par95] Frank C Park. Distance metrics on the rigid-body motions with applications to mechanism design. *Journal of Mechanical Design*, 117(1):48–54, 1995. 39
- [PB07] Maja Pantic and Marian Stewart Bartlett. *Machine analysis of facial expressions*. I-Tech Education and Publishing, 2007. 100
- [Pen06] Xavier Pennec. Intrinsic statistics on Riemannian manifolds: Basic tools for geometric measurements. *Journal of Mathematical Imaging and Vision*, 25(1):127–154, 2006. 78
- [Pen09] Xavier Pennec. Statistical computing on manifolds: from Riemannian geometry to computational anatomy. In *Emerging Trends in Visual Computing*, pages 347–386. Springer, 2009. 3
- [RGC<sup>+</sup>08] Irina Rish, Genady Grabarnik, Guillermo Cecchi, Francisco Pereira, and Geoffrey J Gordon. Closed-form supervised dimensionality reduction with generalized linear models. In *Proceedings of the 25th*

- international conference on Machine learning*, pages 832–839. ACM, 2008. 128
- [RS00] Sam T Roweis and Lawrence K Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000. 85
- [RSMH11] Marc Aurelio Ranzato, Joshua Susskind, Volodymyr Mnih, and Geoffrey Hinton. On deep generative models with applications to recognition. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 2857–2864. IEEE, 2011. 100
- [SD07] Hamit Soyel and Hasan Demirel. Facial expression recognition using 3D facial feature distances. In *International Conference Image Analysis and Recognition*, pages 831–838. Springer, 2007. 119
- [SKJJ11] Anuj Srivastava, Eric Klassen, Shantanu H Joshi, and Ian H Jermyn. Shape analysis of elastic curves in euclidean spaces. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(7):1415–1428, 2011. ix, 4, 18, 20, 21, 22, 26, 40, 41, 42, 43, 47, 51, 68, 101
- [SKK03] Thomas B Sebastian, Philip N Klein, and Benjamin B Kimia. On aligning curves. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(1):116–125, 2003. 54
- [SKK04] Thomas B Sebastian, Philip N Klein, and Benjamin B Kimia. Recognition of shapes by editing their shock graphs. *IEEE Transactions on pattern analysis and machine intelligence*, 26(5):550–571, 2004. 67, 68, 69
- [SLM<sup>+</sup>16] Sumit Srivastava, Shashi Bhushan Lal, DC Mishra, UB Angadi, KK Chaturvedi, Shesh N Rai, and Anil Rai. An efficient algorithm for protein structure comparison using elastic shape analysis. *Algorithms for Molecular Biology*, 11(1):27, 2016. 3

## BIBLIOGRAPHY

---

- [SM06] Eitan Sharon and David Mumford. 2d-shape analysis using conformal mapping. *International Journal of Computer Vision*, 70(1):55–75, 2006. 5
- [Sma96] Christopher G Small. *The Statistical Theory of Shape*, Springer Series in Statistics. New York: Springer-Verlag, 1996. 14, 15, 17, 32
- [SN06] Clayton Scott and Robert Nowak. Robust contour matching via the order-preserving assignment problem. *IEEE Transactions on Image Processing*, 15(7):1831–1838, 2006. 54
- [Söd01] Oskar Söderkvist. Computer vision classification of leaves from Swedish trees. Master’s thesis, Linköping University, Sweden, 2001. x, 48, 49
- [SSD06] Chafik Samir, Anuj Srivastava, and Mohamed Daoudi. Three-dimensional face recognition using shapes of facial curves. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(11):1858–1863, 2006. 101
- [SSDK09] Chafik Samir, Anuj Srivastava, Mohamed Daoudi, and Eric Klassen. An intrinsic framework for analysis of facial surfaces. *International Journal of Computer Vision*, 82(1):80–95, 2009. 101
- [SW12] Fabian A Soto and Edward A Wasserman. Visual object categorization in birds and primates: Integrating behavioral, neurobiological, and computational evidence within a general process framework. *Cognitive, Affective, & Behavioral Neuroscience*, 12(1):220–240, 2012. 3
- [SYM07] Ganesh Sundaramoorthi, Anthony Yezzi, and Andrea C Mennucci. Sobolev active contours. *International Journal of Computer Vision*, 73(3):345–366, 2007. 20

- [Tag99] Hemant D Tagare. Shape-based nonrigid correspondence with application to heart motion analysis. *Medical Imaging, IEEE Transactions on*, 18(7):570–579, 1999. 54
- [TB99] Michael E Tipping and Christopher M Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(3):611–622, 1999. 85
- [TDSL00] Joshua B Tenenbaum, Vin De Silva, and John C Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000. 85
- [TF00] Joshua B Tenenbaum and William T Freeman. Separating style and content with bilinear models. *Neural computation*, 12(6):1247–1283, 2000. 101
- [TGJ07] Ninad Thakoor, Jean Gao, and Sungyong Jung. Hidden markov model-based weighted likelihood discriminant for 2-d shape classification. *Image Processing, IEEE Transactions on*, 16(11):2707–2719, 2007. x, 49, 50
- [Tro95] Alain Trouvé. An infinite dimensional group approach for physics based models in pattern recognition. *preprint*, 1995. 22, 24, 45
- [TYRW14] Yaniv Taigman, Ming Yang, Marc’Aurelio Ranzato, and Lior Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1701–1708, 2014. 100
- [TYW01] Andy Tsai, Anthony Yezzi, and Alan S Willsky. Curve evolution implementation of the Mumford-Shah functional for image segmentation, denoising, interpolation, and magnification. *IEEE transactions on Image Processing*, 10(8):1169–1186, 2001. 127
- [Vid11] René Vidal. Subspace clustering. *IEEE Signal Processing Magazine*, 28(2):52–68, 2011. 127

## BIBLIOGRAPHY

---

- [VKMT13] Carl Vondrick, Aditya Khosla, Tomasz Malisiewicz, and Antonio Torralba. Hoggles: Visualizing object detection features. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1–8, 2013. 100
- [WBR<sup>+</sup>07] Lei Wang, Faisal Beg, Tilak Ratnanather, Can Ceritoglu, Laurent Younes, John C Morris, John G Csernansky, and Michael I Miller. Large deformation diffeomorphism and momentum based hippocampal shape discrimination in dementia of the Alzheimer type. *IEEE transactions on medical imaging*, 26(4):462–470, 2007. 3
- [WBX<sup>+</sup>07] Stephen Gang Wu, Forrest Sheng Bao, Eric You Xu, Yu-Xuan Wang, Yi-Fan Chang, and Qiao-Liang Xiang. A leaf recognition algorithm for plant classification using probabilistic neural network. In *Signal Processing and Information Technology, 2007 IEEE International Symposium on*, pages 11–16. IEEE, 2007. x, 47
- [WBY<sup>+</sup>12] Junwei Wang, Xiang Bai, Xinge You, Wenyu Liu, and Longin Jan Latecki. Shape matching and classification using height functions. *Pattern Recognition Letters*, 33(2):134–143, 2012. 3, 67, 68, 95
- [WFB<sup>+</sup>14] Xinggang Wang, Bin Feng, Xiang Bai, Wenyu Liu, and Longin Jan Latecki. Bag of contour fragments for robust shape classification. *Pattern Recognition*, 47(6):2116–2125, 2014. 3, 95
- [WHL<sup>+</sup>04] Yang Wang, Xiaolei Huang, Chan-Su Lee, Song Zhang, Zhiguo Li, Dimitris Samaras, Dimitris Metaxas, Ahmed Elgammal, and Peisen Huang. High resolution acquisition, learning and transfer of dynamic 3D facial expressions. In *Computer Graphics Forum*, volume 23, pages 677–686. Wiley Online Library, 2004. 101
- [WS06] Mingrui Wu and Bernhard Schölkopf. A local learning approach for clustering. In *Advances in neural information processing systems*, pages 1529–1536, 2006. 87



- [WYWS06] Jun Wang, Lijun Yin, Xiaozhou Wei, and Yi Sun. 3D facial expression recognition based on primitive surface feature distribution. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1399–1406. IEEE, 2006. 119
- [YKR<sup>+</sup>08] Mingqiang Yang, Kidiyo Kpalma, Joseph Ronsin, et al. A survey of shape feature extraction techniques. *Pattern recognition*, pages 43–90, 2008. 4, 13, 14
- [You98] Laurent Younes. Computable elastic distances between shapes. *SIAM Journal on Applied Mathematics*, 58(2):565–586, 1998. 7, 22, 24, 25, 45, 54
- [You99] Laurent Younes. Optimal matching between shapes via elastic deformations. *Image and Vision Computing*, 17(5):381–389, 1999. 5
- [You10a] Laurent Younes. *Parametrized Plane Curves*. Springer, 2010. 25, 45, 54, 127
- [You10b] Laurent Younes. *Shapes and diffeomorphisms*, volume 171. Springer Science & Business Media, 2010. 22
- [You12] Laurent Younes. Spaces and manifolds of shapes in computer vision: An overview. *Image and Vision Computing*, 30(6):389–397, 2012. 13
- [YWS<sup>+</sup>06] Lijun Yin, Xiaozhou Wei, Yi Sun, Jun Wang, and Matthew J Rosato. A 3D facial expression database for facial behavior research. In *7th international conference on automatic face and gesture recognition (FGR06)*, pages 211–216. IEEE, 2006. 115
- [ZJ05] Yongmian Zhang and Qiang Ji. Active and dynamic information fusion for facial expression understanding from image sequences. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(5):699–714, 2005. 100

## BIBLIOGRAPHY

---

- [ZJ09] Zhihua Zhang and Michael Jordan. Latent variable models for dimensionality reduction. In *Artificial Intelligence and Statistics*, pages 655–662, 2009. 85
- [ZKC98] Milos Zefran, Vijay Kumar, and Christopher B Croke. On the generation of smooth three-dimensional rigid body motions. *Robotics and Automation, IEEE Transactions on*, 14(4):576–589, 1998. 39
- [ZLC<sup>+</sup>13] Wei Zeng, Huibin Li, Liming Chen, Jean-Marie Morvan, and Xianfeng David Gu. An automatic 3D expression recognition framework based on sparse representation of conformal images. In *Automatic Face and Gesture Recognition (FG), 2013 10th IEEE International Conference and Workshops on*, pages 1–8. IEEE, 2013. 119
- [ZPRH09] Zhihong Zeng, Maja Pantic, Glenn I Roisman, and Thomas S Huang. A survey of affect recognition methods: Audio, visual, and spontaneous expressions. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(1):39–58, 2009. 99