UNIVERSITÀ DEGLI STUDI DI MILANO

FACOLTÀ DI SCIENZE E TECNOLOGIE

Scuola di Dottorato in Scienze Matematiche

Dipartimento di Matematica Federigo Enriques

Ciclo XXXIV

PhD Thesis

Gaussian Processes for 3D Shape Modelling of Noisy and Incomplete Data. An Application to Human Ears Reconstruction.

Candidate:

Filipa Marreiros Malveiro Valdeira

Supervisor:

Prof. Alessandra Micheletti

Co-supervisors:

Prof. Cláudia Soares

Doctor Ricardo Ferreira

Coordinator:

Prof. Dario Bambusi

Academic Year 2021/2022

Acknowledgments

First and foremost, I would like to express my deepest gratitude to all my supervisors for their genuine dedication, the countless hours we spent discussing and for helping me navigate the uncertainty of research. I am extremely thankful to Professor Alessandra Micheletti for guiding me throughout this long journey and for always stepping up to patiently help me. The support I received was fundamental and I am grateful for the opportunity to work with her and for all the projects we developed together. My deep appreciation also goes to Professor Cláudia Soares, who first introduced me to the academic world and motivated me to do this PhD. I could not have asked for a more supportive and knowledgeable mentor, who constantly inspires me to grow as a researcher and tackle new challenges. Lastly, I sincerely thank Doctor Ricardo Ferreira for his invaluable help and suggestions. I especially thank him for the close guidance during the initial period of the PhD, but also for his large commitment to this project, that went well beyond my stay at μRoboptics. I extend my gratitude to the anonymous referees who read this thesis and provided helpful feedback.

I am extremely grateful for having been part of the BIGMATH project and all the opportunities that it brought along; I would like to thank all of those involved for the many learning opportunities, cooperation and support. A special thank you is due to Luís Tavares, Filipe Jesus and Tiago Cerveira, who, alongside Ricardo, so kindly welcomed me to μRoboptics during several months; I had the opportunity to learn a lot from them and I truly treasure that experience. Finally, I thank Greta Malaspina and Stevo Racković, for all the academic endeavours we took on together; for closely sharing the many highs and lows that come with a PhD; but most of all, for keeping me sane during this entire period.

A heartfelt thank you goes to the friends I have found in Milan and who turned this into an even greater experience. In particular, all of my colleagues at the Mathematics department, who have welcomed me since the very beginning and with whom I shared many unforgettable memories. I must mention those who started this journey with me: Carlo, Emmanuel, Junior, Lorenzo, Yanyan; Ji, with whom I endured many academic and bureaucratic struggles (and successes); and Emanuele, for his infinite patience in answering my questions.

I must also thank all the friends who kept me in touch with the "outside world", while I was submerged in the PhD. Especially Catarina, Adriana, Ana and João, for their lasting friendship and support, regardless of the distance; and Gil, whose daily encouragement is unmeasurable, and without whom I could not have accomplished this.

Lastly, I profoundly thank my family. Especially my parents, whom I could never thank enough for the unconditional support throughout my entire life; and my brother, who I greatly admire as a fellow researcher and as a friend. This thesis is dedicated to them.

Abstract

Mathematical modelling of 3D shapes is a requirement in a vast and increasing number of applications, especially when it comes to the human shape. Despite being an active branch of research, with many proposed solutions, some settings remain particularly challenging. An example of such an open problem is the modelling or reconstruction of human ears from 3D point clouds, the motivation behind this thesis. This specific problem proves to be a particularly challenging case due to the fine and non-linear details to be modelled, together with the extensive presence of missing data and outliers, originated from the data retrieval procedure.

Gaussian Processes were recently proposed as a suitable framework for the formulation of shape modelling problems and have mostly been applied to the human face. We show that this is indeed the most promising setting for our particular kind of data, but it can not be trivially applied from existing solutions. Therefore, we propose a fully unsupervised pipeline from raw 3D scans to a complete ear model, able to cope with the specific challenges encountered.

Nonetheless, the main bottleneck of the pipeline is found on the shape registration part, that hinders the quality of the final results. We address this problem by incorporating the registration task entirely within the Gaussian Processes framework, unlike previous approaches and we provide a method for parameter estimation based on Variational Bayesian Inference. Not only do we achieve better registration results in the presence of large regions of missing data, but we provide a more unified way to deal with shape modelling. This is a step towards a more coherent approach and away from complex pipelines, merging different and often contradictory assumptions.

Keywords: Shape modelling, Registration, Gaussian Processes, Bayesian Inference

Contents

List of Tables

List of Figures

Notation

Linear Algebra

$\text{diag}(A)$ Diagonal vector extracted from matrix A

$\text{diag}(a)$ Diagonal matrix obtained from vector a

$\text{tr}(A)$ Trace of matrix A

$\tilde{A} = A \otimes I_D$ Kronecker product of matrix A with I_D

$\text{vec}(A)$ Vectorized formulation of matrix A

I_D Identity matrix of size $D \times D$

Greek symbols

β Scaling transformation

δ_i Deformation vector to template point t_i

Γ Rotation matrix

Σ Covariance matrix

σ_N^2 Observation noise variance

ς^2 Registration variance

Probability

$\mathbb{E}[x]$ Expectation of x

$\mathcal{N}(\mu, \Sigma)$ Gaussian distribution with mean vector μ and covariance matrix Σ

$\text{cov}(X)$ Covariance of X

$\text{var}(x)$ Variance of x

Roman symbols

D Dimension

$k(x, x')$ Generic kernel function

$k_{Matern}(x, x')$ Matérn kernel

$k_{PDM}(x, x')$ Empirical kernel or Point Distribution Model kernel

$k_{SE}(x, x')$ Squared Exponential kernel

M	Number of shapes in the dataset
N_S	Number of points of the target shape S
N_T	Number of points of the template shape T
$\mathcal{S}$	Dataset of M shapes S^m
$\mathbf{s}$	Vector representation of S
$\mathbf{t}$	Vector representation of T
S^m	Shape matrix (Target) m
s_j^m	Point j of matrix S^m
T	Template matrix
t_i	Point i of matrix T

Subscripts

i	Relative to template point t_i
j	Relative to target point s_j

Superscripts

-1	Inverse
T	Transpose

Glossary

3DMM	3D Morphable Model
BCPD	Bayesian Coherent Point Drift
CPD	Coherent Point Drift
GPR	Gaussian Process Regression
GP	Gaussian Process
GT	Ground Truth
ICP	Iterative Closest Point
NICP	Non-rigid Iterative Closest Point
PCA	Principal Component Analysis
PDM	Point Distribution Model
RQ	Rational Quadratic
SE	Squared Exponential
SFGP	Shape Fitting Gaussian Process
SSM	Statistical Shape Model
VBI	Variational Bayesian Inference

Chapter 1

Introduction

Shape modelling is becoming ever more relevant and complex, as new applications and access to data keep increasing. The particular case of the 3D setting brings with it enormous potential, but not without an equal amount of challenges. The literature on the topic is both large and diverse, connecting different areas of expertise, and has seen considerable developments in recent years. From face recognition [1, 2] to medical applications [3, 4], the relevance of accurate and detailed models has been attested in the past and will very likely continue for the foreseeable future.

The full process of generating a complete and informative 3D model from raw data generally calls for different techniques and is often presented as a pipeline [5–7], composed of sequential separated steps with some level of specificity for the shape and application under study. This by itself is an immediate drawback, since application to new settings requires inevitable tuning and data pre-processing, among other things. Furthermore, some types of data have additional challenges that pre-existing methods may not have taken into account.

We shall broadly formulate the shape modelling problem, resorting to the simple house example in Figure 1.1. As depicted, a *shape* is represented as a set of N points in dimension D; in this case, $D = 2$, although we will focus on the more challenging $3D$ case throughout the thesis. A common assumption is that shapes representing the same object can be obtained by transforming a representative example of that class, denoted as *template*. Ideally, the template is as similar as possible to all the shapes in the same class and can be thought of as being a mean shape of the object at hand (the exact definition of *mean shape* will be clarified later on). In other words, the template should be chosen such that other samples may be obtained by employing as few transformations as possible. Furthermore, a template should be a complete example, i.e., no regions of the shape should be missing; in the same way, there should be no occurrence of points not pertaining to the shape. In Figure 1.1, the template is represented on the left, as the average shape of a house.

Under these assumptions, the task of *shape modelling* consists in describing the transformations applied to the template leading up to the remaining shapes in the class. This description is denoted as *model* and should fulfil three main goals: requiring a reduced number of parameters to describe the transformation (*compactness*), preventing transformations that lead to unreasonable shapes (*specificity*)

and including enough freedom as to represent all possible samples of the class (*generalization*). For instance, a model for the house could include just one parameter for variation of the height, keeping the overall shape unaltered. This would be a compact and specific model, but with reduced generalization capabilities, since most houses could not be represented under this description.

A closely related task is *model fitting*. Given an available shape model and an unobserved new shape of the same class, denoted as *target*, the goal is to find the parameters that lead to the best approximation of the unseen shape under the considered model. In other words, this corresponds to the task of deforming the template in order to obtain the target, when restricted by a pre-existing model. Shape modelling and fitting are closely related since the usefulness of one highly depends on the other.

The two tasks face several challenges in common, one of them being the kind of transformations involved. For the moment, we consider *rigid* and *non-rigid* types. Rigid transformations are those that do not alter the relative position of points within the shape, including both translation and rotation. While this setting already poses considerable challenges, the addition of non-rigid transformations certainly increases the difficulty of the problem. However, this proves necessary for many applications, where rigid deformations are not enough to represent the possible class shapes.

An additional obstacle is the problem of *registration* or *correspondence*. Since shapes are retrieved as unorganized point clouds, there is no established connection between points in different samples. This affects both modelling (as deformations must be learnt from corresponding points) and fitting (where the template must be in correspondence with the target). The registration problem entails two different questions: how to define correspondence between points and how to retrieve it from unorganized point sets. Correspondence can be defined under different criteria, but in general, conveys the idea that corresponding points refer to the same feature. For instance, the top of the roof in the template and target of Figure 1.1 should be taken as matching points. Correspondence retrieval is often achieved by deforming the template, under generic transformation restrictions, and taking the closest point in the target. Naturally, this implies the existence of a model to use in template deformation, hinting at a so-called *chicken-and-egg problem* faced in shape modelling. That is, the registration task calls for a previous model, while modelling requires registered shapes.

Finally, as depicted in Figure 1.1, the point sets of each sample have different sizes when compared to the template and even amongst themselves. Consequently, there are points in the template without correspondence with the target and vice-versa. These are denoted *missing points* and *outliers*, respectively. The absence of a one-to-one correspondence must be accounted for in both model learning and fitting, adding additional challenges to both tasks.

Throughout this thesis, we focus on modelling a shape which is both rare across the literature [8, 9] and offers considerable challenges — the human ear. The work was mainly motivated by the research and technological interests of our industrial partner, the company μRoboptics, which is engaged in the development of prostheses for detailed parts of the human face. The ear is a representative and prime example of this goal. Due to its inherent anatomy and position, the retrieval of 3D data from this region is not a simple task, leading to samples with vast occurrences of both missing data and outliers. Additionally, it entails fine details and non-rigid deformations, thus calling for sufficiently expressive and

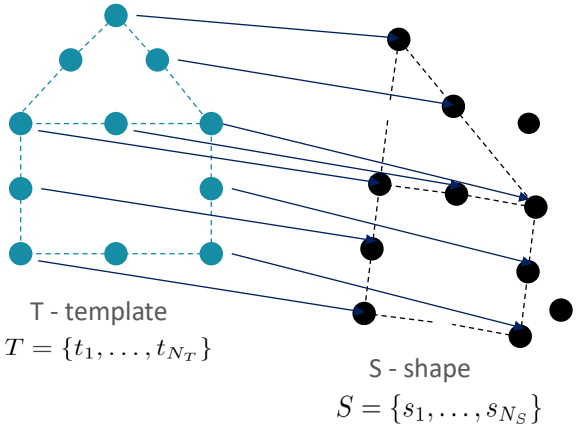


Figure 1.1: Schematic representation of the problem at hand. The template T is represented in blue circles and a generic target shape S in black circles; the dashed lines connecting the circles are added for better visualization of the shape, but they do not provide additional data to the problem. The arrows connecting both shapes establish a semantic corresponding between the point clouds, i.e. two points are connected by an arrow if they represent the same feature. Lastly, it is possible to observe that some points are present in the target, but not in the template (outliers); while others are present in the template but not in the target (missing points).

flexible models. Nonetheless, we strive for generic approaches and despite taking the ear as a case study throughout the work, we do not restrict the application to this particular case.

Existing approaches to the registration and modelling task fail to handle the multiple challenges found in the ear shape, calling for new techniques. The use of Gaussian Processes (GP) for shape modelling is not new and has been around for some time [10, 11]. It is particularly remarkable for the freedom it offers in expressing different prior knowledge with kernels and thus very appealing for challenging scenarios. The particular formulation followed in this thesis [11] is, as the authors themselves point out, an "Open framework". The generality and freedom it entails are both a blessing and a curse. The potential to find solutions for the difficult obstacles encountered is large, but a straightforward and immediate application of previous methods is not feasible. Therefore, we develop new solutions within the GP framework, aimed at modelling detailed shapes, with a high prevalence of missing regions and outliers.

1.1 Outline and Contributions

A major point of focus of this work is a global and contextualized perspective on the topic, in the search for an increasingly unified setting for shape modelling. Often, modelling and registration fields have little concern for each other and do not take full advantage of existing knowledge on the other side. We provide a considerable review of both the modelling and registration worlds and their evolution, taking a high-level approach which allows us to bridge related, but otherwise separated, worlds (Chapter 2). This provides a conceptual justification for our choice of framework and places our work in the literature. This chapter also contains an introduction and description of the datasets used throughout the thesis.

Chapter 3 is devoted to Gaussian Processes (GP) and their use in shape modelling. It includes an initial section on relevant background and previous use in modelling approaches. While the use of GPs for

shape modelling is not a novelty, the focus in past work lies mostly in kernel design, with straightforward techniques for correspondence retrieval and model fitting. Thus, expert knowledge is required to build meaningful kernels, leading to tailored and non-generic methods. Besides, shapes more frequently studied (e.g. human face) benefit from several pre-existing models, with new work built on top of already accurate proposals. Since ears have been scarcely seen in related literature, the available prior knowledge is limited. Therefore, the above-mentioned challenges must be dealt with in different ways.

Our approach is to explicitly formulate the GP shape modelling problem with missing data and outliers, unlike previous work, where this is handled with pre or post-processing steps. The formulation is used throughout the subsequent chapters and provides a more favourable setting for the ear shapes. Thus, the main contribution of Chapter 3 is

- **Formulation of the shape modelling problem in the GP framework with explicit separation of missing data and outliers.** In previous approaches, it is assumed that there is a one-to-one-correspondence between samples. By explicitly considering the existence of non-corresponding points in the target and template, we achieve a more principled approach (Section 3.4), that improves the results in further sections.

This is a minor contribution on its own, but it is the foundation of subsequent chapters. The formulation allows us to efficiently deal with the high ratios of outliers and missing data observed in the ear scans.

The two subsequent chapters contain the extended content of two produced papers (one published and one under submission). In particular, they have somewhat more background and mathematical details, as well as additional results, which were kept out of the papers due to size restrictions. Furthermore, they were reformulated to build a fully coherent and integrated text with consistent notation and where the evolution and connections between different parts can be made more clear.

Chapter 4 provides a complete pipeline from the raw scans of human ears to a set of complete shapes, which can be used to generate a shape model. Traditionally, models are retrieved by employing a state-of-the-art registration method followed by a dimensionality reduction technique. However, this approach fails for shapes with extreme ratios of missing data and outliers, such as the human ears. Both of these factors lead to a reduced overlap between a template and possible targets, such that common registration methods fail to retrieve an accurate correspondence and, consequently, lead to poor models. The alternative could be to include manual pre-processing, but this is not desirable as it restricts the amount of data used to build the model.

Therefore, we propose the replacement of the registration task with three different steps, aimed at producing a training dataset with complete, but diverse shapes. Since the samples are highly incomplete, we resort to shape completion with Gaussian Processes to fill the absent regions. However, this requires the previous identification of non-missing points (and correspondence between them), motivating the two initial steps. A trivial application of a non-rigid registration would lead to poor results, given the extensive regions of outliers extracted alongside the ears. To overcome this, we propose an initial step for outlier removal and rough alignment. This is achieved through a novel rigid registration method, RANSIP, with a cost function adjusted to our setting, by taking into account point cloud normals. Further, the proposed

algorithm employs a randomized approach that helps in overcoming local minima — a common obstacle in rigid registration and amplified by the large outlier ratio.

However, real raw ear scans do not have established correspondence between them. Hence, it is difficult to accurately evaluate the results of registration or modelling. We handle the absence of ground truth by creating a simulated dataset with realistic features. Finally, given that each stage serves different goals, we propose suitable metrics to evaluate each one. The final result is a full pipeline that can be applied to raw shapes. Further, the reasoning for the selection of each step is carefully detailed, so that a similar process could easily be extended to new types of data. Complete shapes obtained through the pipeline are shown to produce a dataset of increased variability compared with existing ones.

The work related to Chapter 4 produced a paper [12] with title *From Noisy Point Clouds to Complete Ear Shapes: Unsupervised Pipeline* published in the IEEE Access Journal (Q1). The main contributions included in this chapter are summarized below.

- **Suitable metrics and datasets for the evaluation of pipelines.** While we apply the pipeline to ear scans, the proposed metrics and use of simulated datasets may be used in similar scenarios when facing the absence of ground truth, high prevalence of missing data and outliers (Section 4.2).

- **Novel registration method, RANSIP.** We propose a new method for rigid registration (Section 4.4), suitable for a low overlap between targets and template. Compared with state-of-the-art registration methods, RANSIP achieves better results in outlier removal (Section 4.5).

- **Gaussian Process for shape completion of challenging shapes.** We propose a novel pipeline that facilitates the use of GP regression for shape completion, thus taking advantage of the large potential of kernels in shape prediction (Section 4.6).

- $3D$ **ear dataset with increased variability.** As an output of the pipeline, we obtain a new complete ear dataset, retrieved from full head scans, with higher variability than previous ones (Section 4.7).

Chapter 5 develops a new probabilistic registration method, Shape Fitting Gaussian Process (SFGP), fully formulated within the GP framework and especially suitable for shapes with extensive missing data. This bridges the gap between the GP setting with increased flexibility on model design, but a hard assignment on model fitting steps, and the probabilistic registration, with known advantages in dealing with outliers and missing data, but often restricted in terms of underlying kernels. Unlike the pipeline in Chapter 4, the registration is fully conducted within the GP setting, resulting in a more principled and unified approach.

We first show how the problem of registration with soft assignment can be formulated within the GP framework as a multi-annotator Gaussian Process Regression. Previous approaches to registration in this framework were restricted to hard assignments, which are known to struggle in the presence of outliers and missing data. The concept of multi-annotation has been widely used in traditional machine learning for cases of uncertain labelling. We show how this idea can be applied to GP registration, to implement the notion of soft assignment.

An important parameter of the multi-annotator formulation is the level of confidence for each annotation, which in practice translates to a variance. In order to estimate this parameter, we formulate the registration problem in the traditional probabilistic setting and solve it with Variational Bayesian Inference. Comparing the update equations with those retrieved from the GP formulation, we are able to obtain an estimate for the variance. Nonetheless, we show that the differences with respect to state-of-the-art probabilistic methods are fundamental when dealing with missing data. The final algorithm is applied to the ear setting (both simulated and real data) and shown to increase performance when compared to benchmarks.

The work related to Chapter 5 has produced a paper [13] with title *Probabilistic Registration for Gaussian Process 3D Shape Modelling in the Presence of Extensive Missing Data*, currently under submission. The main contributions of this chapter are summarized below.

- **Shape registration/model fitting as a multi-annotator GPR.** We show how the problem of registration with soft assignment can be understood within the GP framework as a multi-annotator Gaussian Process Regression (Section 5.2).
- **Parallel between probabilistic registration and our method, SFGP.** We provide a parallel between the traditional probabilistic registration setting and our algorithm, under a few assumptions, which allows us to estimate parameters for SFGP (Section 5.3). We further show how their differences lead to a good performance in the presence of extensive missing data (Section 5.4).
- **Application to a difficult registration problem – 3D ears registration.** We show that our method is suitable for the registration of 3D point clouds with highly non-rigid deformations, high occurrence of missing data and outliers, by performing simulations with 3D point sets of human ears. The results show improvement with respect to state-of-the-art proposals (Section 5.5).

We leave the last chapter for a summary of the main points and achievements, together with a perspective on future advances and questions that remain open (Chapter 6).

The work contained in this thesis was developed in cooperation with our industrial partner, μRoboptics, within the BIGMATH project. The BIGMATH project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No 812912.

Chapter 2

Shape analysis

Shape analysis has seen considerable developments since its initial contributions up to the recent models of detailed 3D shapes with complex deformations. We will start by introducing basic concepts and tools, that have allowed the subsequent development of our approaches (Section 2.1). Next, we will focus on how to describe variations in shape through Statistical Shape Models, together with the main tasks and challenges involved in this procedure (Section 2.2). We will then cover the evolution of the several components, which will help us in positioning our work within the vast literature found for this topic (Section 2.3). After this, we will see how the different segments are usually integrated in pipelines, in order to produce a model from the original input data (Section 2.4). Finally, we will specifically focus on the problem of ear modelling, looking at existing approaches and data. This will allow us to identify the main challenges behind this problem and to confront them with the literature introduced in previous sections. Therefore, at the end, we will have a more defined problem formulation and the motivation for the chosen framework should become clear.

2.1 Background on shape analysis

2.1.1 Kendall shape space

Kendall set the foundation for statistical shape analysis [14], describing *shape* as the geometrical information that remains after removing translation, rotation and scaling. In this framework, a shape is represented by *landmarks*, points obtained from the object boundary, with the same semantic meaning across shapes. For a dimension D and N landmarks, we can define $\left[s_1, \ldots, s_N\right]$ as the set of landmarks, where $s_j \in \mathbb{R}^D$. Landmarks can then be grouped to form a configuration matrix of size $N \times D$

$$S = \begin{bmatrix} s_{11} & \ldots & s_{1D} \\ \vdots & \ldots & \vdots \\ s_{N1} & \ldots & s_{ND} \end{bmatrix}. \tag{2.1}$$

An example of a $2D$ vertebra with 60 landmarks is provided in Figure 2.1, where each point s_i is appropriately labelled.

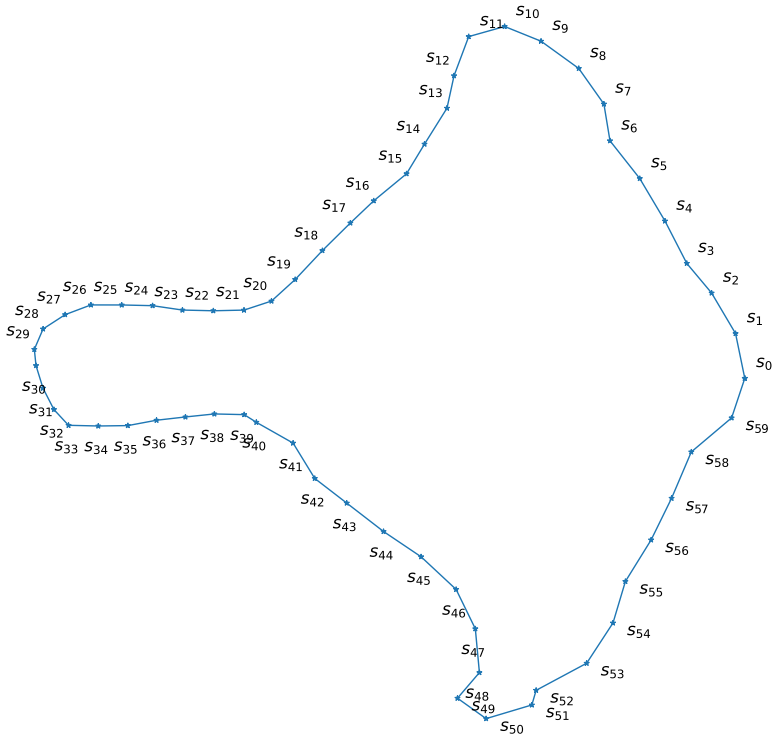


Figure 2.1: Representation of shape landmarks from a dataset in [15]. In this case, there are 60 landmarks in two dimensions, depicting a vertebra. The points in blue represent the landmark coordinates, labelled with the corresponding index. The lines connecting pairs of points are added for visualization purposes only.

The space generated by all landmark coordinates is called *configuration space*, generally considered as $\mathbb{R}^{D\times N}$ (although some particular cases could be removed such as coincident points). By normalizing all shapes with respect to translation and scaling, we obtain the *pre-shape space*, which is an hypersphere of unit radius and has dimension $(N-1)D-1$. In order to remove scaling, it is necessary to choose a size measure, which is often taken to be the centroid size [15], given as

$$g(S) = \sqrt{\sum_{i=1}^{N}\sum_{j=1}^{D}(s_{ij}-\bar{s}_j)^2}, \tag{2.2}$$

where $\bar{s}_j$ is the mean in $j-th$ dimension. The centroid size can also be reformulated as $g(S) = \|CS\|$ where $C = I_N - \frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T$ is the centering matrix and $\|X\| = \sqrt{\text{tr}(X^TX)}$. Translation can be removed in different ways, but here we choose the centred landmark coordinates, $S_C = CS$, as the respective plot will provide the correct visualization of the original shape. Therefore, one can obtain a pre-shape with

$$Z_C = CS/\|CS\|. \tag{2.3}$$

In *pre-shape space* two out of the three initial transformations have been removed, namely translation

and scaling. It is still necessary to remove rotation to achieve the *shape space*, of dimension $ND - D - 1 - \frac{D(D-1)}{2}$. A rotation matrix, $\Gamma \in \mathbb{R}^{D\times D}$, is a special orthogonal matrix, i.e. an orthogonal matrix with determinant $+1$ (see Appendix A.5). Rotation is less straightforward to remove and we shall see later how this can be performed.

However, it is common to study not only the shape but also the size of given samples, in which case it should not be removed. The resulting space, originated by removal of translation and rotation, is denoted as *size-and-shape*.

Finally, we will define *Euclidean similarity transformations* of S as the set of shapes originated by a translation p, scaling β and rotation Γ, i.e. $\{\beta S\Gamma + \mathbf{1}_N p^T : \beta \in \mathbb{R}^+, \Gamma \in SO(D), p \in \mathbb{R}^D\}$. In the same way, we define *rigid body transformations* as the set of shapes originated only by translation and rotation.

2.1.2 Procrustes analysis

Given Kendall's definition of shape, in order to compare the shape of two or more given objects it is necessary to remove any similarity transformations between them. This leads to Procrustes Analysis, which entails the minimization of the squared Euclidean distance between points in shapes. The formulation for two shapes (Ordinary Procrustes Analysis) has a closed form solution for any dimension D, while the generalization for a higher number of shapes (Generalized Procrustes Analysis) only has a closed form up to 2 dimensions. We shall cover both of them in the following sections.

Ordinary Procrustes Analysis

Full Ordinary Procrustes Analysis (OPA) is the least squares matching of two shapes S_1 and S_2, with respect to the similarity transformations β, Γ and p, given as the following optimization problem

$$\underset{\beta,\Gamma,p}{\text{minimize}} \|S_2 - \beta S_1\Gamma - \mathbf{1}_N p^T\|^2. \tag{2.4}$$

The solution to Problem (2.4), assuming that S_1 and S_2 have been centred, is given as

$$\hat{p} = 0, \quad \hat{\Gamma} = UV^T, \quad \hat{\beta} = \frac{\text{tr}(S_2^T S_1 \hat{\Gamma})}{\text{tr}(S_1^T S_1)},$$

where $S_2^T S_1 = \|S_1\|\|S_2\| V\Lambda U^T$, with $U, V \in SO(D)$ [15]. The full Procrustes fit of S_1 onto S_2 is consequently given as

$$S_1^P = \hat{\beta} S_1 \hat{\Gamma} + \mathbf{1}_N \hat{p}^T.$$

Generalized Procrustes analysis

Generalized Procrustes Analysis (GPA) is a generalization of OPA for the case of more than 2 shapes, i.e., when we have a dataset of M shapes $\mathcal{S} = \{S_1, \ldots, S_M\}$. In this case, the problem becomes

$$\underset{\beta_m,\Gamma_m,p_m}{\text{minimize}} \sum_{m=1}^{M} \|(\beta_m S_m \Gamma_m + 1_N p_m^T) - \Upsilon\|^2, \tag{2.5}$$

where Υ is the mean shape of the population. It is further necessary to impose a constraint on the size, so that $\hat{\beta}$ is prevented from going to 0. This can be taken, for instance, as

$$\sum_{m=1}^{M} g^2(\beta_m S_m \Gamma_m + \mathbf{1}_N p_m^T) = \sum_{m=1}^{M} g^2(S_m),$$

where $g(S)$ is a size function such as (2.2). The full Procrustes mean is then given as

$$\bar{S} = \frac{1}{M} \sum_{m=1}^{M} S_m^P,$$

where S_m^P is the full Procrustes fit of the m-th shape given as $S_m^P = \hat{\beta}_m S_m \hat{\Gamma}_m + \mathbf{1}_N \hat{p}_m^T$. Algorithms for the solution of Problem (2.5) can be found in [15], providing the estimation of $\hat{\beta}_m, \hat{\Gamma}_m, \hat{p}_m$.

GPA can be seen not only as a method to align a dataset, but also as a way to fit empirical data to Kendall's shape space. Furthermore, both OPA and GPA can be formulated without minimization of scaling if we intend to keep size differences.

2.1.3 Distances

A point of interest in shape analysis is to study dissimilarities between shapes and for this purpose it is fundamental to define a distance function between two samples. At a first glance it could be tempting to use the minimum of OPA in (2.4), usually denoted as ordinary (Procrustes) sum of squares (OSS), as a distance measure. However, it should be noted that, in general, we have that $OSS(S_1, S_2) \neq OSS(S_2, S_1)$ and so this can not be taken as a distance.

Instead, we can consider a particularization of the Procrustes problem, where both shapes have unit size, i.e., they are already in the pre-shape space. The solution of this problem leads to the Procrustes distance, either partial or full, depending on whether scaling is included or not. The *partial Procrustes distance* is given as

$$d_P(S_1, S_2) = \inf_{\Gamma \in SO(D)} \|Z_2 - Z_1\Gamma\| = \sqrt{2}\Big(1 - \sum_{i=1}^{D} \lambda_i\Big)^{1/2},$$

and the *full Procrustes distance* as

$$d_F(S_1, S_2) = \inf_{\Gamma \in SO(D), \beta \in \mathbb{R}^+} \|Z_2 - \beta Z_1\Gamma\| = \sqrt{1 - \Big(\sum_{i=1}^{D} \lambda_i\Big)^2},$$

where Z_1 and Z_2 are the pre-shapes of S_1 and S_2, and λ_i are square roots of eigenvalues of $Z_1^T Z_2 Z_2^T Z_1$, with $\lambda_1 \geq \cdots \geq \lambda_{m-1} \geq |\lambda_m|$. The optimal rotation is given for both distances as

$$\hat{\Gamma} = UV^T, \tag{2.6}$$

where U and V are obtained from the singular value decomposition of $Z_2^T Z_1 = V\Lambda U^T$, with $U, V \in SO(D)$

and the optimal scaling as

$$\hat{\beta} = \sum_{i=1}^{D} \lambda_i,$$

for $d_F(S_1, S_2)$.

Both of these distances are extrinsic, since they are computed in an embedding of the true space, but an intrinsic distance can also be computed entirely on the pre-shape space: the Riemannian distance. It corresponds to the great circle distance between Z_1 and Z_2 on the pre-shape sphere, given as

$$\rho(S_1, S_2) = \arccos\Big(\sum_{i=1}^{D} \lambda_i\Big),$$

where the eigenvalues remain the same as for the Procrustes distances. However, note that $d_F(S_1, S_2) = \sin\rho$ and $d_P(S_1, S_2) = 2\sin(\rho/2)$, which means that when considering shapes close in the pre-shape space, these distances are very close to each other. For example, when comparing different instances of the same object, one can expect to be in this scenario and so, very often, the choice of distance has little impact.

2.1.4 Tangent space

A good approximation to compute distances between shapes is to use the tangent space to the pre-shape sphere, usually taking the mean shape as the pole. Given the pole Υ and a pre-shape Z, we find the optimal rotation $\bar{\Gamma}$ between them according to (2.6). The tangent coordinates of Z are then given as

$$\nu = (I - \mathrm{vec}(\Upsilon)\,\mathrm{vec}(\Upsilon)^T)\,\mathrm{vec}(Z\bar{\Gamma}),$$

where $\mathrm{vec}(\Upsilon)$ is the vectorized formulation of matrix Υ. We note that the Procrustes residuals given as $\hat{\beta}_m S_m \hat{\Gamma}_m - \Upsilon$ are approximate tangent coordinates (they are not orthogonal to the pole), but if the shape is close to the pole they are good approximations [15]. The importance of the tangent space to the mean shape of a given dataset will become clear throughout the next sections.

2.2 Statistical shape models

In the previous section we saw how to remove similarity transformations from a dataset, so that differences between the resulting objects are due to shape. However, this does not provide the tools to study or represent the observed differences, which requires a posterior analysis. In particular, we can consider *shape modelling* as the task of describing the variability in shapes, usually for a given object of interest. A typical way to achieve this is to build a Statistical Shape Model (SSM). Particularly for the human shape and shapes of interest in biology, this is the main reference [16] and has been used in countless applications. Statistical Shape models are composed of a *shape space*, together with a *prior* defined on it. The former provides a limited space where we can find instances of the shape of interest (and consequently excludes unreasonable shapes), while the latter provides information on how likely a given instance is to be observed. Note that under this context the shape space is generally much

smaller than the one defined by Kendall. Kendall's definition includes all possible configurations of the landmarks (except singularities) when disregarding Euclidean similarity transformations. For the SSM we are typically interested in a particular class of shapes and so the shape space is ideally much more restricted.

Using Kendall's representation with landmarks, the standard SSMs are called Point Distribution Models (PDM), first proposed in 1995 [17] and the basis for most models observed in state-of-the-art approaches. Their core idea is to model shape variation by statistically analysing and representing the variation of each landmark observed on a training dataset. Other shape representations are possible and have been proposed, such as planar curves [18, 19] or surfaces [20]. Nonetheless, landmark based ones remain the most popular [21].

2.2.1 Point distribution models

The assumptions behind a PDM are that we have a set of M different instances of our shape of interest, constituting the training dataset. Furthermore, all shapes have the same number of points N and the points in each shape are ordered in such a way that point i of any shape in the dataset has the same semantic meaning, according to the assumptions on (2.1). Under these assumptions, GPA can be applied as detailed in Section 2.1.2 to remove the effects of similarity transformations and we assume it was already employed.

Remark on notation. From this point onwards we have the need to distinguish between indices relative to a shape within a dataset and to a point within a shape. Therefore, we will use superscripts to indicate different shapes of a dataset: $S^m \in \mathbb{R}^{N\times D}$ is shape m of the dataset $\mathcal{S}$with M shapes, $\mathcal{S} = \{S^1, \ldots, S^M\}$. On the other hand, $s_j^m \in \mathbb{R}^D$ refers to point j of shape S^m. We further need to represent the vectorized form of a shape, taken as $\mathbf{s}^m = ((s_1^m)^T, ..., (s_N^m)^T)^T \in \mathbb{R}^{ND}$ and, consequently, $\mathbf{s}^m$ is the vectorized representation of S^m.

The authors in [17] proposed to obtain a space shape and a prior through Principal Component Analysis (PCA) [22]. The main idea behind PCA is to represent a given dataset in a space of lower dimension than its original one, while keeping most of the observed variance, where each direction in the new space is uncorrelated. Below we describe PCA under the shape modelling context.

PCA for shape modelling

Under the previous assumptions and considering the vectorized shape space $\mathbb{R}^{ND}$, we obtain the mean shape $\bar{\mathbf{s}} \in \mathbb{R}^{DN}$ as

$$\bar{\mathbf{s}} = \frac{1}{M}\sum_{m=1}^{M} \mathbf{s}^m$$

and the sample covariance matrix $\Sigma \in \mathbb{R}^{DN\times DN}$ as

$$\Sigma = \frac{1}{M-1}\sum_{m=1}^{M} (\mathbf{s}^m - \bar{\mathbf{s}})(\mathbf{s}^m - \bar{\mathbf{s}})^T.$$

Given the spectral decomposition of the covariance matrix, $\Sigma = U\Lambda U^T$, where Λ is a diagonal matrix

of eigenvalues of Σ $\{\lambda_1, \ldots, \lambda_{DN}\}$ and U the orthonormal matrix with columns corresponding to the eigenvectors $\{u_1, \ldots, u_{DN}\}$, ordered by their eigenvalue size, the directions of maximum variance (and minimum projection error) are given by the columns of U. Consequently, a shape $\mathbf{s}$ can be represented in this model as

$$\mathbf{s}(\alpha) = U\alpha + \bar{\mathbf{s}}, \tag{2.7}$$

where α are the coordinates of shape in the new space and can be obtained as $\alpha^m = U^T(\mathbf{s}^m - \bar{\mathbf{s}})$ for the m-th shape of the dataset. The former corresponds to a rotation of the data points, but PCA also entails the projection to a lower dimensional space. Furthermore, we note that we lie in the case of high dimensionality, where the dimensions of shapes are usually much higher than the number of samples we have access to, $DN \gg M$. This means that at least $DN - M + 1$ eigenvalues will be zero. Therefore, we keep only the eigenvectors of U corresponding to the P largest eigenvalues, so that we refine $U \in \mathbb{R}^{DN \times P}$ and $\alpha \in \mathbb{R}^P$.

While the eigenvectors define the shape space, in order to get a SSM we still need some way to represent how likely a shape is to be observed. In [17] a Bayesian approach is taken by enforcing a prior on the parameter α. The authors assume that the set of shapes follow a normal distribution $\mathcal{N}(\bar{\mathbf{s}}, \Sigma)$, in which case the components will have a distribution $\mathcal{N}(0, \Lambda)$, such that

$$p(\alpha) = \prod_{i=1}^{DN} \frac{1}{(2\pi\lambda_i)^2} \exp\left\{ -\frac{\alpha_i^2}{2\lambda_i} \right\}, \tag{2.8}$$

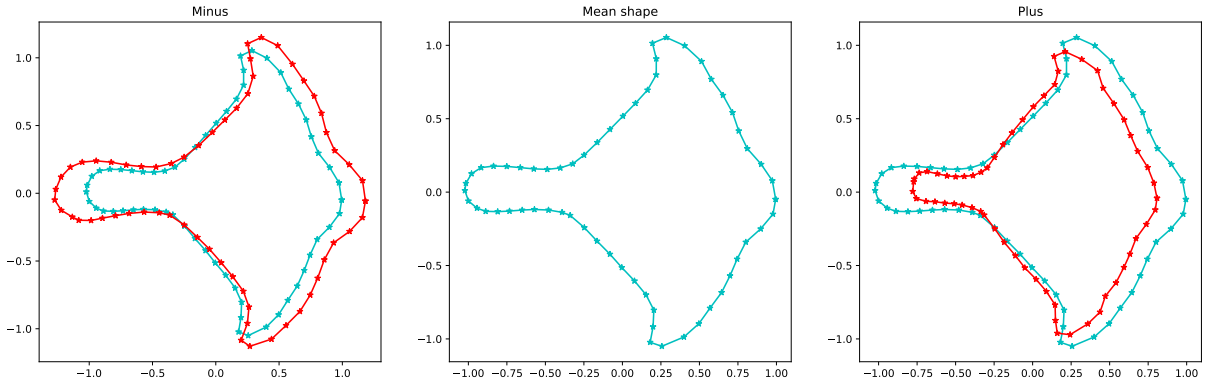
thus defining a prior on the new shape space.

Figure 2.2 exemplifies a PDM model obtained from the vertebra dataset in [15]. In particular, it depicts both the mean shape and the first two modes of variations, i.e. the shape alterations caused by the first two principal components. The first component mostly controls the elongation of the left portion of the vertebra and the prominence of the distinctive features on the right portion. The second component appears to control the direction of the inclination on the left part.

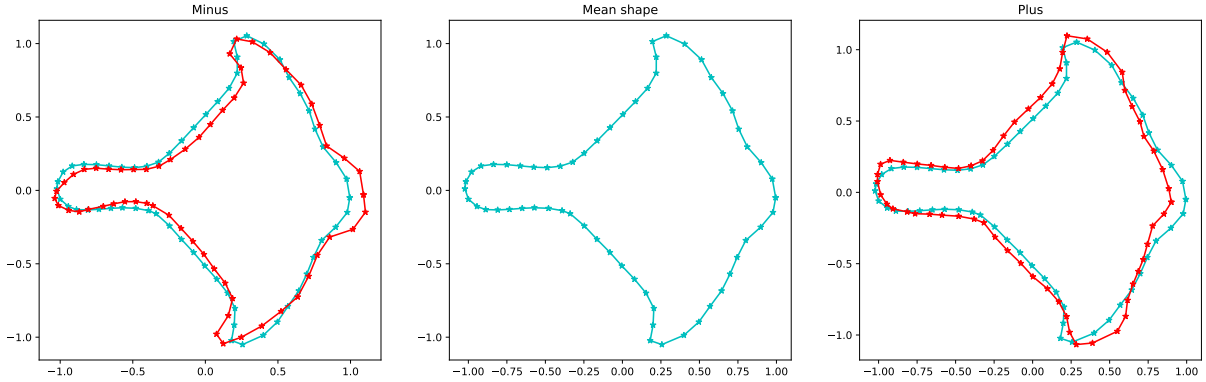
2.2.2 3D setting

These models were originally applied for 2D data, but, as access to 3D data increased, the focus has shifted to the three dimensional setting. This kind of data brought several new challenges by breaking the assumptions previously considered. While the approach in [17] remained the basis for the majority of subsequent models, extending it to this new setting and corresponding applications is not straightforward.

The most relevant difference is perhaps that the previous assumption on labelling is no longer applicable. Data obtained through 3D scans or similar procedures typically consists of hundreds or thousands of points presented in an unorganized manner, which largely contrasts with the small amount of organized landmarks assumed in Kendall's work and in [17]. While that setting allowed for manual labelling of each sample, doing so for the 3 dimensional scenario, would be not only extremely costly and time-consuming, but also severely prone to errors. However, the absence of correspondence between shapes prevents the application of PCA as described before.



(a) First mode



(b) Second mode

Figure 2.2: First two modes of the PDM obtained from the vertebra dataset [15]. The mean shape is represented in blue and the respective modes in red. They are obtained by taking the respective principal component with a weight of $-3\sqrt{\lambda_i}$ (on the left) and $+3\sqrt{\lambda_i}$ (on the right).

This problem was tackled in the 3D Morphable Models (3DMM) [5], that can be considered as an extension of [17] to the 3D universe. They propose an initial step for establishing correspondence between all the shapes in the dataset and a *shape of reference* or *template*. This term refers to an arbitrary shape, ideally as close as possible to all the remaining shapes, to which all the others can be matched. This task, denoted as *shape correspondence* problem [23], is not exclusive of the shape modelling context and, consequently, has its own branch of literature, independently of the application. At the time, the correspondence methods were mostly focused on 2D data and so the authors in [5] first project the 3D coordinates to a 2D setting and then apply an existing gradient-based optical flow algorithm where variations in shape and texture are matched. For regions with non distinctive features (e.g. forehead, cheeks), where the algorithm falls short, the authors use smooth interpolation. After the correspondence is achieved, the previous formulation of PCA is applicable and a shape model can be obtained. Nonetheless, correspondence is a crucial step and limits the accuracy of any model [16]. In fact, there is a considerable loss on information in performing this step in 2D.

A related problem is that the data can no longer be assumed to be noiseless and complete. Retrieval of 3D data not only produces samples without correspondences, but also provides no assurance that it is possible to find a one-to-one correspondence between them. We consider the template shape T with N_T points to be the complete shape. An arbitrary shape in the dataset S^m will in general have $N_S^m \neq N_T$

points, such that some points in S^m have no correspondence in T — *outliers* — and some points in T will have no correspondence in N_S — *missing data*. Furthermore, the capturing devices will necessarily impose some noise on the data.

2.2.3 Applications

The models previously described allow us to represent a given shape in a lower dimensional space, with a prior on the components given by (2.8). While this could be the endpoint if the goal is to merely study shape variability, the models usually have some other application in mind, which are in general tasks where it is required to ensure that only likely shapes are considered. For instance, the Active Shape Model in [17] was developed to locate shapes in images, while the 3DMM were originally proposed for inferring a full 3D facial surface from a single image of a person.

Despite having been applied to different kinds of shapes, including objects [24] or animals [25], without a doubt their major application, and in particular for the 3D case, has been found in modelling human shapes. We do not necessarily restrict this overview to solutions under this setting, but it is an area where a large portion of the literature lies, some of them being generic enough for an application to an arbitrary shape [26], others including specific assumptions on the task at hand.

An important line of research is the modelling of the full body [26–29], where not only the shape is of concern, but also variations in pose, which calls for an accurate model of articulated behaviour [27, 28] or handling of extreme non-rigid deformations [26, 29]. This is then used in applications such as prediction and tracking of human movement, or in the fashion industry. We note that the particular study of articulated movement is not a point of study of this thesis, and we limit our study to static shapes.

Consequently, we consider models of singular parts of the human body. In general, two main areas of focus can be pointed out: portions of medical interest and the face/head [30]. Naturally, the former may include the latter, but due to its complexity and potential of applications, there is a large segment of literature specifically focused on modelling the human face. Under medical applications, we find applications such as anomaly detection [3], development of prosthesis [4, 31], or segmentation[1] [32–34]. Instead, models of the human face can be used for face recognition [1, 2, 35, 36], augmented reality and gaming [37], face reconstruction [38, 39], facial motion sequence tracking [37] or expression recognition, amongst others.

Other then their specific tasks, at some point of the pipeline most of the mentioned practical applications lead to a more abstract problem: *model fitting*. Usually, the difference across applications is seen on the type of data to be fitted to the 3D model, but we can roughly express it as finding a shape representation within the defined shape space (i.e. finding an appropriate vector α), taking into account the prior. The original 3DMM was designed for the reconstruction of 3D faces from example images and the fitting followed the Bayesian approach as described before, where the likelihood was given by the similarity between the produced and observed image, with prior given by (2.8).

The increasing complexity of the applications and of the shapes to be represented also found limitations

[1] localizing a given shape in a 2D or 3D medical image

on the initially proposed PCA-model. The latter is not able to represent shapes with fine details or with a large difference with respect to the training set, thus calling for further solutions.

2.2.4 Challenges in statistical shape modelling

We are now in a position to identify the main challenges involved in 3D modelling as

- **correspondence** across a dataset of training shapes;
- **modelling** increasingly complex and detailed shapes, given a limited dataset;
- **fitting** an existing model to unseen instances of the shape;
- **robustness** to noise, missing data and outliers in all of the previous tasks.

2.3 Evolution of shape modelling

As an answer to the previous challenges, several improvements have been suggested since the proposal of 3DMM in 2002, in the field of (3D) statistical shape learning, although they are considered the foundation for most approaches. As we have seen, the process of going from raw input data to a full and expressive model, generally includes a pipeline with sequential steps regarding **correspondence**, **modelling/training** and **fitting**. We will first consider the main concepts and research directions proposed for each of these components, as this will allow us to localize our problem in the context of the literature (respectively Section 2.3.1, 2.3.2 and 2.3.4). Finally, we will address full pipelines and how the previous steps are usually integrated. We also include an initial section on the types of data, which deeply determine any method and has been overlooked in the review up to this point.

2.3.1 Data

Data used to train (and fit) the model may be obtained in different formats. The most common representation for 3D shapes are *triangle meshes*, composed of a set of vertices and a set of triangles. Vertices are sets of vectors in $\mathbb{R}^3$, where each vector component corresponds to a point coordinate in space; triangles are a set of vectors in $\mathbb{R}^3$, where each component is a nonnegative integer which corresponds to the index of a vertex — one triangle connects neighbour vertices. Although mesh information may be used for any of the modelling steps, here the focus is on modelling with *point clouds*, i.e. exclusively the vertices of the mesh, which correspond to an unorganized version of S as defined in (2.1). In general, the triangulation is kept constant and can be used to represent in 3D a given point cloud for visualization purposes.

Additionally, *appearance* information may also be provided (e.g. texture, albedo) and modelled. In fact, the original 3DMM models both shape and appearance with separate PCA models and several subsequent approaches also include this component [40]. This is of particular importance for applications concerning images, such as image synthesis or its inverse, reconstruction of 3D shape from a 2D input. In this thesis, image data is not used, and the focus lies specifically on modelling shape/geometry.

Nonetheless, multiple approaches in literature involving an appearance component can be applicable to the point cloud case, as the two kinds of information are dealt with separately; other, however, specifically require appearance information. Despite disregarding non-geometry input, point cloud models are widely used in non-image applications [41–44], since they have the advantage of being invariant to pose and illumination.

Some methods also resort to training directly from 2D images when building a 3D model, either exclusively [45] or as a mixture of 3D and 2D [46]. While training from 2D images is much more accessible and one can get considerably larger datasets, it loses direct 3D information. We focus entirely on building models from 3D point clouds.

2.3.2 Modelling

We saw that for a statistical model we wish to find both a shape space and prior on the model parameters. The shape space must be restrictive enough not to allow for unreasonable shapes, but also large enough to (ideally) represent any possible shape. PCA-based models as proposed in [5] have strong assumptions behind. In particular, the assumption that any shape can be represented by a linear combination of the learnt vectors, which may not be true, specially when considering non-rigid deformations. Even if this is not the case, usually the training set size (in the order of tens or hundreds) is much smaller than the dimension to learn (thousands of points in two or three dimensions) and so it is likely that the space learnt from PCA is not representative enough to handle unseen shapes with larger differences to the ones in the training set. This is particularly true when it comes to small shape details, as the main directions which are kept express global variations and the local ones are usually lost.

Increasing training data

The last observation suggests one simple approach to improve the model quality: increase the number of training samples and their diversity. This has been seen for the case of human faces with the recent model [6] learnt from 10,000 examples, highly diverse in terms of age, gender and ethnicity. As expected, this has a positive influence on the model performance. Nonetheless, acquiring 3D data (and correctly processing it) is not a straightforward task and with the exception of [6], most models are obtained from a few hundreds or tenths of samples [7, 9]. Given the existence of several small datasets, some work has also been done in bringing them together to produce a single model [30], which is extremely challenging as they usually present drastically different characteristics, such as pose or sampling, making the correspondence task increasingly difficult.

Localized deformations

Another option, proposed even in the initial work of [5], is to split the shape into several regions and retrieve a PCA-model separately for each of them. In this way, the size of training data becomes closer to the dimension to be represented and local variations are better represented. The main obstacles are the segmentation which needs to be done specifically for the shape at hand and how to smoothly merge different parts in the end. Nonetheless, proposals in this direction [1, 47] have shown good results — for

instance, particular advantage is seen on the mouth region in [47]. Automatic segmentation has also been studied [41] to mitigate the drawbacks of this approach.

Following the idea of part-based models, a more general and formal solution is to use wavelet basis functions [32–34, 48]. In this setting, each mesh is represented as a linear combination of different scaling terms (corresponding to progressive subdivisions of the mesh) and wavelets at different positions. PCA can then be performed on the new shape coordinates.

Another way to steer away for the global property of PCA, is to introduce sparsity terms in the objective function, thus leading to more localized deformations [49]. This has been recently explored for PCA in different spaces [50, 51].

Expressions

We note that, for the specific case of the human face, a considerable obstacle is found when dealing with expressions. They can introduce considerably larger deformations with respect to a neutral face, than the variations found between two different individuals with a neutral expression. Popular approaches include decoupling identity and expression using multilinear models [35, 52], multilinear wavelet models [37] and neural networks [53–55] or learning deformations with respect to a neutral face and transferring them [2]. Although this specific topic does not pertain to our scope, it is still related, as facial expressions may be considered large non-linear deformations of the face — this is particularly relevant when dealing with the correspondence problem. I.e., similar obstacles are faced when retrieving correspondences between shapes with generic non-linear deformations and faces with different expressions.

PCA on different spaces

While PCA-models assume that the manifold on which the data lie is an hyperplane, with axes following directions of maximal variance, other manifolds where to perform PCA can be considered [29, 44, 56–58]. More in line with Kendall's original work, the authors in [44] propose to perform GPA after correspondence was obtained, so that the shapes are within Kendall's shape space. Afterwords, they project them to the tangent space to the Procrustes mean and perform PCA in this space.

In [56] they use a medial representation for shapes, which focus on describing thickness, bending and widening of the object. The authors then propose Principal Geodesic Analysis, a generalization of PCA to describe the variability of data on a manifold. This has later been used in full body representations to predict human movement [29]. An extension of PCA has also been proposed to the Shell space [57, 58], where shapes are modelled by non-linear thin-shells subjected to tangential and bending distortions. They observe improvements in modelling articulated body pose data and complex deformations arising from expressions.

Neural networks

Given the current interest on neural networks and their great performance observed in image data, it is only natural to propose their use for shape modelling, which indeed has been observed in recent years [24, 30, 45, 53–55, 59, 60]. This can be seen as yet another way to introduce non-linearity in the models.

Different architectures have been employed, including Convolutional Neural Networks (CNN) [39, 45, 60] a popular approach for computer vision tasks, Generative Adversarial Networks (GAN) [53, 54, 59] due to their proven ability in generating realistic visual data and their unsupervised nature, or Variational Auto-Encoders (VAE) [26].

Given that most approaches were initially developed for image data, their application in 3D shape modelling often involves some step of conversion of 3D meshes to 2D [53, 59, 60] or learning entirely from 2D images [39, 45]. However, some recent proposals have seen working directly on 3D data [26, 54, 55], although the absence of Ground Truth (GT) and limited data for training remains a challenge. Neural networks-based models have mostly been used for shape synthesis [54, 59], expression recognition [53] and monocular reconstruction [30, 38, 60], but they still show considerable limitations in generalization capabilities [61].

2.3.3 Correspondence

So far we have assumed that we had some way of establishing correspondence between all shapes in a dataset, but this is not usually the fact and it is an extremely challenging point. It is known that, regardless of the model being used, registration has a great impact on the final performance [16]. We note that this is not only a problem for training where we generally need to have all shapes in correspondence to build a model, but also for most approaches proposed in model fitting. Besides, if one has all shapes registered it is also easy to remove Euclidean similarity transformations with GPA, but usually these two processes are connected.

First, we note that the initial assumption on labelled shapes as we have seen for Kendall, can be considered as *sparse correspondence*. A few distinctive points are marked on the boundary of the shape based on their semantic meaning or geometric properties. Naturally, this can be extended to 3D as well either manually or automatically [62]. The former has evident drawbacks concerning human error and time consumption, but even the latter can be difficult. It is not clear what does it mean for two points to be correspondent in different shapes. Even ignoring these challenges (and given that some methods exist with good performance to detect landmarks in the human face, for instance), a model obtained from such a limited shape information would not be able to produce realistic and detailed shapes for most applications in the 3D setting.

Hence, most of the models we covered, as well as 3DMMS, use *dense correspondence*, where a much larger number of points are put into correspondence. Although more challenging to compute, dense correspondences are able to express more detailed structures. Nonetheless, the landmarks[2] of sparse correspondence are often used as an auxiliary tool for dense registration [6, 44, 63, 64], even if there is an effort to stay away from this approach [42, 61, 65] given the limitations previously mentioned.

In general, one can think of the correspondence problem between two shapes S_1 and S_2 as finding the reparametrization of a shape point set S_1, such that it minimizes some function of dissimilarity between the ordered shape and S_2. The most common approach to put a full dataset in correspondence is to select

[2]Note that in this context the term *landmark* gains a slight different meaning, as they no longer fully describe the shape, although they are coherent with the landmark definition of Kendall

a *reference shape* or *template*, which is put into correspondence with all remaining shapes, so that the shapes are also in correspondence between themselves [30, 66]. However, some methods avoid selecting a template (as this may lead to different results depending on the chosen shape) and simultaneously register all shapes to each other [67, 68].

As previously introduced, the initial approach for 3DMMs proposed to retrieved correspondences within a 2D setting, which inevitably leads to a considerable loss of information when compared to a fully 3D contained approach. Although the 2D setting has not been dropped during the years [63], using the original 3D data is highly appealing and has been widely considered in the literature [42, 66]. Besides, when converting data between the two different dimensional embeddings, appearance information is of fundamental importance to preserve shape characteristics. Given our restriction to point clouds, remaining in the 3 dimensional setting becomes increasingly relevant. Therefore, the review is dedicated to the subset of 3D point cloud registration.

Evidently, this would be straightforward if the two samples had the same shape, in which case we would be in presence of two different permutations of the same shape, where an exact reordering of points would be achievable. However, two different shapes will necessarily have transformations between them and, since at this point there is no access to a shape model (or this can not be used since no correspondence has been established), we have no prior information on the allowable deformations. A way to circumvent this, is to use a similarity based method, where points are matched according to selected features ([64] and [23] for a review). However, this has limitations, for example on regions of no distinctive features, which occurs very often in dense correspondence of biological shapes; they can lead to inconsistent correspondences on a global level; they struggle with only partial overlapping. Therefore, most statistical shape analysis use *registration* methods. Although sometimes used in an interchangeable way (specially in the modelling setting), registration is a type of correspondence where the template is deformed to fit the target in order to minimize a distance function between them, with the deformations restricted by a generic (non shape specific) model.

Even just considering similarity transformations, the solution is not easy, but extending it to non-rigid (as it is often necessary for a good correspondence) increases the problem complexity. Thus it makes sense that there is an entire branch of literature dedicated to this problem [69–71], where different assumptions on transformations are taken. We will cover this in more detail in Chapter 4. For now, we will just consider that there are several methods which can take two point clouds and put them into correspondence with some assumptions on the allowed transformations.

Overall, shape modelling pipelines tend to follow one of these options: assume the training shapes are already in correspondence, obtained by an unspecified registration method [37, 41]; chose a suitable available registration method so that the pipeline starts from the raw scans [39, 55, 59]; suggest a new and often specific registration method for the shape in question [42, 44, 66]. Note that statistical shape models can also be used in registration [50] but in a first instance, one does not have access to them — this is often called data-driven registration.

2.3.4 Fitting

The model fitting problem can generally be translated as follows. Given an arbitrary model $\mathcal{M}(\alpha)$ and a new target shape S, what are the parameters α which minimize some distance function $d(\hat{S}, S)$ between the model reconstructed shape $\hat{S}$ and true shape S. The translation to a less abstract problem is highly dependent on the type of data and model chosen, and is connected to the application in view. Nonetheless, the goal is usually to obtain a complete and clean 3 dimensional shape from an input, which for some reason does not present these characteristics. This is also denoted as shape reconstruction.

As we have seen, shape reconstruction from one [38, 39, 60, 72, 73] or more images [48], is a common application of 3D shape models. Therefore, several sources propose fitting of a 3D model to an image, and while the general concept is similar, the approaches are usually not applicable. On the other hand, methods depending on appearance information often include separate procedures for shape and appearance, and the former can be considered independently, although it will in principle decrease the expected performance. As stated before, we focus on fitting 3D point clouds.

The simplest version of the fitting problem is found under the assumption that the target to be fitted S is in correspondence with the model $\mathcal{M}(\alpha)$ and that rigid transformations have been removed. In this case a *maximum a posteriori* (MAP) approach is usually employed [74]. Under a Gaussian assumption, the likelihood is given as

$$P(S|\hat{S}) = \prod_{j=1}^{N} P(s_j|\hat{s}_j) = \prod_{j=1}^{N} C \exp\left\{ -\frac{1}{2\sigma_N^2}\|\hat{s}_j - s_j\| \right\} = C^N \exp\left\{ -\frac{1}{2\sigma_N^2}\|\hat{\mathbf{s}} - \mathbf{s}\| \right\} \tag{2.9}$$

and in terms of model parameters

$$P(S|\alpha) = C^N \exp\left\{ \frac{1}{2\sigma_N^2}\|U\alpha - \mathbf{s}\| \right\},$$

where C is a normalization constant. With the prior as given by (2.8), we have the posterior

$$p(\alpha|\mathbf{s}) = C_2 \exp\left\{ -\frac{1}{2\sigma_N^2}\|U\alpha - \mathbf{s}\| \right\} \exp\left\{ -\frac{1}{2}\|\alpha\|^2 \right\}.$$

Maximizing the posterior is equivalent to minimizing

$$E = \frac{1}{\sigma_N^2}\|U\alpha - \mathbf{s}\| + \|\alpha\|^2.$$

Other priors for α have been proposed. For face modelling, authors in [44] have pointed out that the length of α follows a chi-square distribution with P degrees of freedom, which has a mean P, contrasting with the assumption that the expected length will be 0 in the former formulation. For this reason, they include a constraint based on the Mahalanobis distance on the minimization problem, which in practice means modelling the shape space as a manifold that is at a constant Mahalanobis distance from the mean. Later, in the review paper [16], it is pointed out that this leads to highly unrealistic shapes along the directions of principal components. To solve this, they propose to constrain the shapes to an hyper-box proportional to the standard deviation of the data along each axis, translated to an equivalent prior to

be introduced in the previous problem.

In applications such as face recognition, it is fundamental to handle different poses of the shape of interest and this cannot be assumed previously done as in [16]. I.e., the removal of similarity transformations must be included in the fitting process, since faces are expected to appear in any position and should be detected regardless of this factor. Therefore, authors in [44, 74] further apply similarity transformations to $\hat{s}$, leading to additional parameters to be estimated. The resulting problem is challenging: in [74] it is solved by using small angle approximations (valid for small variations in position), while in [44] the non-linear problem is directly solved with Levenberg-Marquardt optimization.

Generally, for reconstruction from sparse points (as considered in [44, 74]) it is reasonable to assume known correspondences, but fitting to a dense set of points obviously has advantages in introducing more detail on the target shape. In this case the correspondence must be obtained in some way. Several proposals assume that a registration method was previously employed, in the same way as explained for training. Others, however, try to include correspondence in the fitting problem. An example is found in [43] where they simultaneously estimate pose/scale, model parameters and correspondences, assuming only rough alignment at the beginning. The closest point in the target is taken as correspondence and then the fitting to the similarity transformations and model parameters is done, with a first order approximation of the cost function. Since the template is being transformed to the target and correspondences are being estimated during the process, this can be seen as non-rigid registration with a shape prior. Other methods follow a similar approach [35, 47, 75]. Similarly to the registration, fitting methods can also employ landmarks [2, 37] or not [43].

An additional consideration is the existence of missing regions on the target shapes. When these are relatively small, most methods are able to overcome it naturally (if the sampling is dense enough) or the meshes are previously pre-processed with hole filling methods. However, if large portions of the shape are missing this is not enough, as the correct shape must be predicted. This problem is usually denoted as *shape completion*. In general, this setting is not approached during modelling, as it is assumed that the samples used in training are close to complete and therefore most methods assume that a previous model already exists and can be used to predict the missing parts.

2.4 The pipelines

In the previous sections, we have looked at each step of the shape modelling task separately, but in a real scenario they must be combined in order to go from raw 3D point clouds to a final model. Part of the literature focuses on proposing full pipelines [5–7, 44]. Generally, they start with a registration method, followed by PGA to remove similarity transformations. Then a model is obtained (PCA is used in most of the full pipelines found) and a fitting method is proposed, given the application in mind.

It is also common to assume that one or more steps have been previously taken care of, starting the pipeline from different points. A common assumption is that shapes are already in correspondence by a previous method and only modelling and fitting are proposed [35, 37, 41, 50, 51, 57], since they tend to be more dependent on each other. However, some focus specifically on modelling [33, 41], remaining generic

enough for any possible application, or on fitting [43, 76–78], assuming a model is already provided.

Either way, this segmented approach leads to sequential approaches to the same problem in different formulations. From the previous section, we have seen that registration can be used with data priors in which case they equate to a shape fitting problem, and in the same way, shape fitting may include correspondence estimation, thus leading to a registration method. Evidently, they can be generalized by the same problem, as we detail below.

2.4.1 General formulation

Under the assumption that all shapes are represented as 3D point clouds, we will formulate both registration and model fitting. Given a model $\mathcal{M}$ with parameters α, a target shape S, a permutation matrix P and similarity transformations given as a function, $\rho(T) = \beta\Gamma T + p$, of the template, T, we formulate it as

$$\underset{\alpha,P,(\beta,\Gamma,p)}{\text{minimize}}\, d(\hat{S}, S) = d\Big(PR(\mathcal{M}(\alpha)), S\Big) + f(\alpha) + g_L, \tag{2.10}$$

where $f(\alpha)$ represents a constraint on α and g_L a constraint from landmarks. We saw that the main difference between registration methods and model fitting methods is that the former take generic rules for the constraint $f(\alpha)$, while the latter observe them from training data. On the model side, P is usually assumed known [44, 74] and often R as well [16]. Even in this scenario, the choice of a constraint for α is not trivial, as we saw. On the registration side, often only the transformations given by ρ are considered (α is removed), denoted as *rigid registration*, which is already a challenging problem. The full problem, *non-rigid registration*, requires a definition of the constraint on α, even more challenging in this case, since the shape space is much larger.

It is then clear why the generic problem is often tackle in simplified reformulations — each of the sub-problems contains enough complexity and challenges involved. However, a single approach would be perhaps advantageous, since solving each partial problem provides additional information to the others, and so looking at it as a whole can improve the final result. An initial step in this direction [5, 9], and possibly the most commonly used across shape modelling pipelines, are boostrap based techniques where several iterations of registration, training and model fitting are performed, so that the model is progressively more accurate. Although this improves the final results, it still has some drawbacks, other than not being a principled approach. A reduced performance on the initial step of registration may not be recoverable further down in the pipeline and different optimality criteria are usually employed for different segments [21].

This has motivated approaches in a more unified way, in different degrees. The first idea is to introduce constraints on α both from the data and from generic knowledge on shape deformation [47, 79]. This restricts the shape space when compared to registration methods (avoiding unreasonable shapes), but provides more freedom when the model does not have enough variability. Although, it still assumes the existence of a previous model, this can be a rough one. So we can understand this as a more principled version of boostrap approaches. For instance, in [79], this is achieved by mixing a training covariance with a smooth one and then perform fitting in the Bayesian setting. A more general formulation of this,

is the Gaussian Process framework [11, 80] where the covariance can be expressed with more freedom. The fact that the prior knowledge can be adjusted within the same formulation, means that potentially the same cost function can be used for both the initial registration and shape fitting. Finally, some go even further and propose simultaneous learning of the model and correspondence [30, 81], so that no previous knowledge is required.

2.5 Ear modelling

The previous background was rather generic, but some considerations are due regarding the specific shape under consideration — the human ear. First, we will cover existing models of this shape in the literature and how they have been produced (Section 2.5.1). We will then introduce the datasets to be used throughout this thesis (Section 2.5.2). This will allow us to finally define the specific goals and challenges of our work (Section 2.5.3).

2.5.1 Literature

Given the increased interest in human face models and availability of 3D data, this area has seen several developments in the recent years, leading to overall high quality full face and full head models [6, 82]. However, it is known that such models present poor quality or variability in more detailed parts, in particular the ears [83]. Reasons for such performance degradation are: the variability of the whole head shape surpasses the one of these smaller parts and data occlusion in these regions, due the anatomical characteristic of this region preventing a good performance on data retrieval. Therefore, the majority of existing full head models are not enough for fine applications such as prosthesis design. In fact most face models do not include the ears [30]. An exception is [83], in the Gaussian Processes framework, where they focus on augmenting a head model with an ear one. However, this still requires the identification of 50 manual landmarks for registration.

Consequently, models focusing exclusively on the ear region have been proposed. To the extent of our knowledge there are two existing 3DMM of the ear in [8] and [9]. However, only the latter final model, that is the mean shape and principal components, is made publicly available.

The work in [8] makes use of the large deformation diffeomorphic metric mapping (LDDMM) framework to produce the model. Under this setting, the authors model the deformation of one shape into the other as a flow of diffeomorphisms. This model is then simplified with a kernel based PCA, thus obtaining a morphable model. To build the model they use the SYMARE database [84], composed of 58 ears.

In [9], the authors start with a limited dataset of 20 samples from 10 subjects, building an initial model with Coherent Point Drift [85], a non-rigid registration method, and PCA. However, given its reduced variability, the authors propose to use the initial model along with an existing 2D ear database already labelled with landmarks to produce a larger augmented dataset. First, they find the parameters of the model leading to a shape that, when projected to 2D, is the most similar to the 2D image in the dataset. Then, they deform the mean shape of the initial model to match each of the ears, with a variation of CPD.

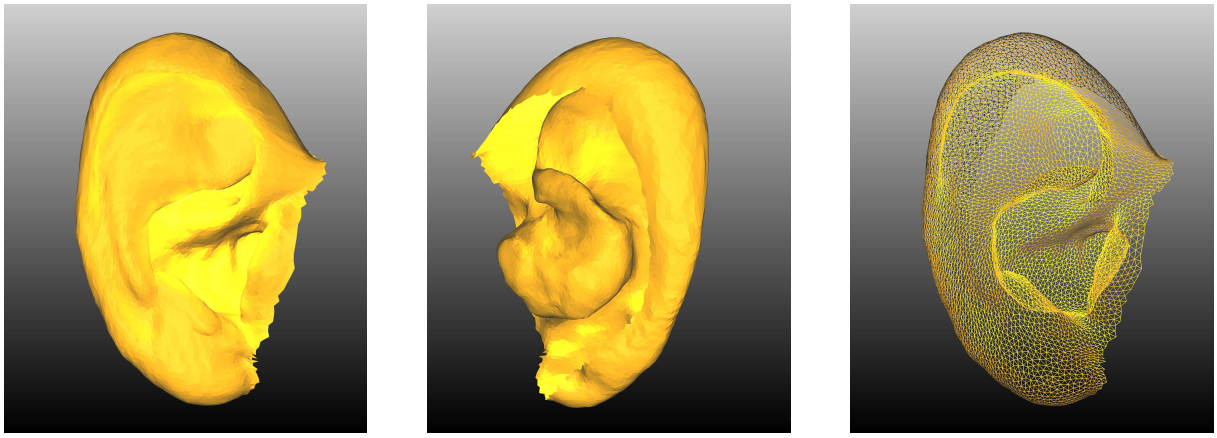
The final model is obtained from this dataset in a straightforward manner with Generalized Procrustes Analysis and PCA. The model is incrementally improved by iterating over the data augmentation and model production step.

2.5.2 Datasets

Ear dataset

Given that the dataset in [9] is the largest one available, we begin by looking into it. We shall denote it as *Ear Dataset* from now on. The dataset is composed of 500 meshes of ears, similar to the one depicted in Figure 2.3. Each sample is composed of 7111 3D points and correspondence has been previously established. That is, the i-th point of each shape represents the same feature and there is a one-to-one correspondence between all samples of the dataset. Under this setting it is possible to immediately apply PGA and obtain the mean shape, depicted in Figure 2.4. In the same Figure, we see the most different shape with respect to the mean, where the dissimilarity was computed as the average distance between each correspondent point.

We also present other measures of variability of the dataset in Figure 2.5. For each shape we compute the distance between each of its points and the corresponding one on the mean shape. The figures depict both the mean and standard deviation for all points. From these two considerations, it is clear that the dataset does not present very high variability, as it was indeed expected due to the augmentation process used to create it.

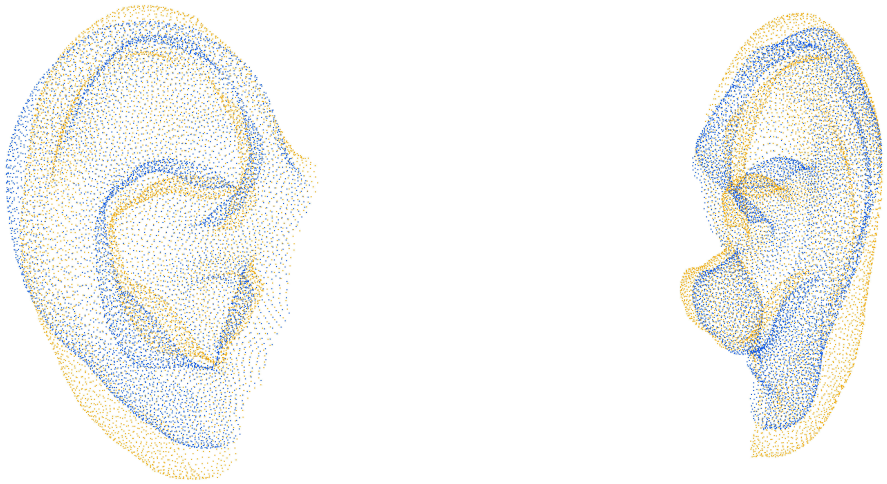


(a) Front view (b) Back view (c) Triangulated view

Figure 2.3: Sample from the *Ear Dataset*, with different views.

Head dataset

However, for the protheses scenario we need to establish a relationship between the ear and the remaining part of the head. This means that we should be able to obtain samples from full head scans. We take the *Headspace Dataset* from [83], of which an example is provided in Figure 2.6. This contains 3D scans of 1519 individuals with different genders, age and ethnicity; although the distribution is not balanced in the last two categories, with a predominance of white young adults. Each sample



(a) Front view (b) Back view

Figure 2.4: Relevant samples from the *Ear Dataset*. Then mean shape is depicted in orange and most different shape (according to the average Euclidean distance) in blue. Despite being slightly shorter and presenting a moderately different curvature of top, the shape in blue is still similar to the mean shape. This hints at the lack of variability existing in this dataset.

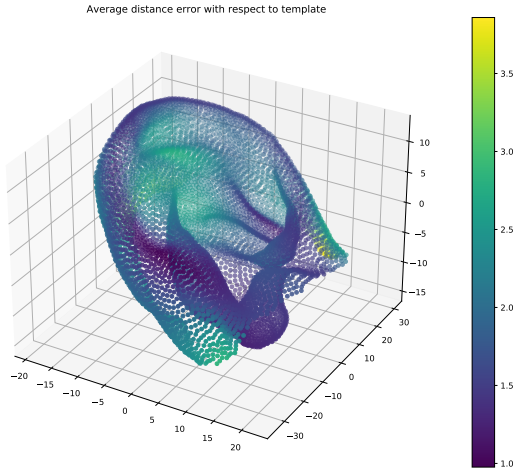
contains around 180,000 vertices, but correspondence is not provided, i.e., the scans do have the same number of vertices and there is no information on the correspondence between points of different samples. Additionally, mesh and texture data is included for each individual. We use the former exclusively for visualization purposes and completely disregard the latter. Thus, for each scan we exclusively extract the vertex and triangle information.

Furthermore, each sample contains a set of 35 landmarks over the face region, that allow for the use of GPA to roughly remove large components of translation and rotation. Upon this pre-processing step, it is possible to extract a region around the ear location in order to obtain a dataset of ears from the original dataset (Figure 2.6). We denote this new dataset as *Head Dataset* which, despite the name, refers only to the ear portion of the scans.

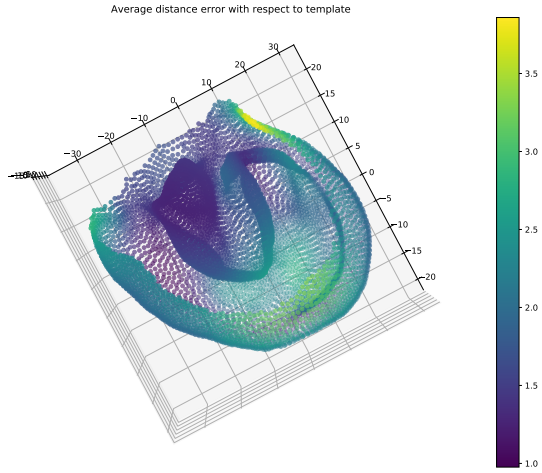
However, to ensure that the full ear is completely captured for any head, it is necessary to define a rather large section, leading to extensive regions of outliers for most cases. The alternative approach would be to register the full heads and retrieve the corresponding points, but we have seen that most methods will fail on this region. Looking at the retrieved point cloud, we also note that a large portion of the shape is missing. This is a challenge both for the registration and training.

2.5.3 Challenges

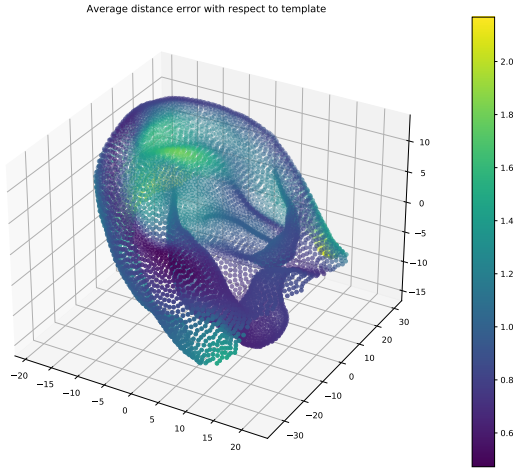
The goal of this project is to build a model for the ears retrieved from the *Head dataset*. A standard approach with a registration method followed by PCA is not applicable, due to the extensive presence of outliers and missing data. Not only do they hinder the results of generic registration methods, but they mean that a large percentage of points will not find correspondence. This prevents the application of PCA as we are not able to retrieve deformations for all points of the template. In fact, most of the



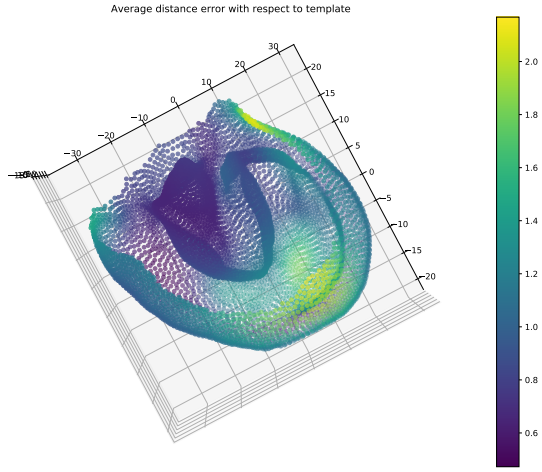
(a) Average distance error — front view



(b) Average distance error — back view



(c) Standard deviation— front view

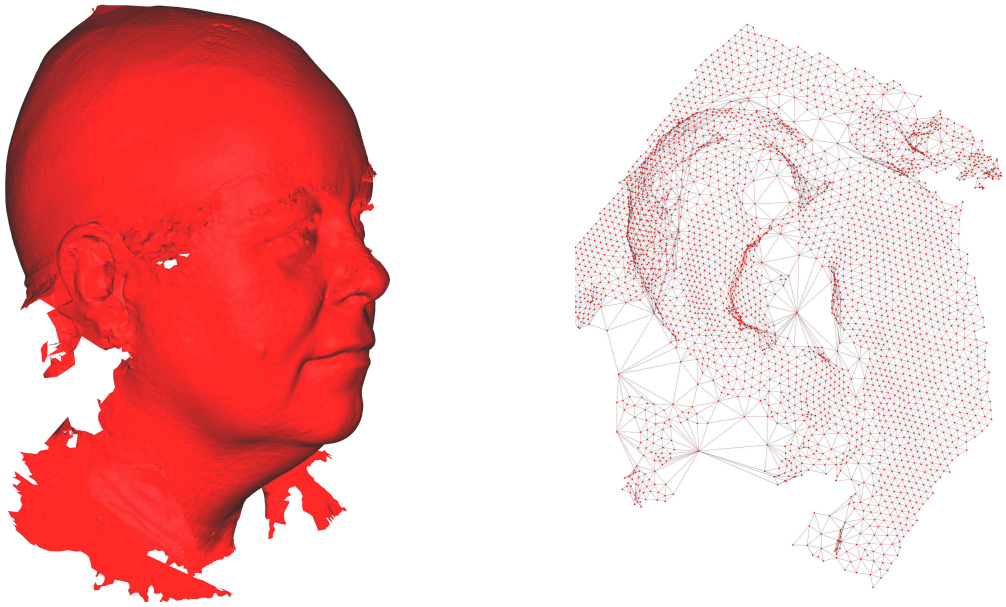


(d) Standard deviation — back view

Figure 2.5: Variability of the *Ear dataset*. The top figures represent the average distance error for each point on the template over the entire dataset. The bottom figures represent the standard deviation under the same setting. Both values are expressed as a color map.

proposed approaches for modelling assume that it is possible to get a one-to-one correspondence between a template and the samples.

Given the existence of the *Ear Dataset*, one could think of fitting the available model to each sample, in a data driven registration manner. However, the limited variability observed in the data would likely not be able to adequately capture the diversity of the *Head Dataset*. This points to the use of unified approaches, for a balanced prior of data learnt deformations and generalization. The Gaussian Processes are a principled and promising framework where to develop this and they will be introduced in the next chapter.



(a) Full head

(b) Ear portion

Figure 2.6: Example of a sample from the original *Headspace Dataset* [83] and the correspondent ear region in the *Head Dataset*, as obtained from the full head. The extensive occurrence of missing data and outliers for this region is easily noticeable and it is mostly caused by ear shape, preventing an adequate capture during the scanning process.

Chapter 3

Gaussian Processes for shape modelling

In this chapter we cover the background on Gaussian Processes (GP), necessary to understand the subsequent chapters. We will mainly focus on Gaussian Process Regression (GPR), introduced in Section 3.2, as it is an important tool in shape modelling. In a GP framework the model is defined by the kernel of a GP, so we will devote Section 3.3 to this topic. The former introduction is completely generic and can be applied to a variety of problems. In Section 3.4, we formulate the shape modelling under this setting and set the notation for the following chapters.

3.1 Introduction

Definition 3.1 (Gaussian Process)**.** *A Gaussian Process (GP) is an infinite collection of random variables, any finite number of which have a joint Gaussian distribution. [86]*

Therefore, a GP can be seen as an infinite-dimensional generalization of the multivariate Gaussian distribution or, in other words, a probability distribution over functions. It is a non-parametric modelling approach and consequentially powerful for scenarios which are difficult to model explicitly.

GPs have been widely applied in supervised learning for regression [87–89] and classification problems [90, 91]. In this context, we are fundamentally interested in the former, i.e. the prediction of continuous output variables given a certain input observed data, which in this case is denoted as Gaussian Process Regression (GPR). GPs are appealing in the regression setting as they provide a structured and unified way to model with covariance kernels, which can easily encode a wide variety of prior information, and they naturally lead to a measure of uncertainty on the predictions.

We can at this point have a hint of why GPR may be of interest in shape modelling. On one hand, the possibility of modelling with covariance functions is appealing as it allows to express a multitude of shape characteristics, without restriction to a linear model for example. On the other hand, in the presence of extensive missing data, the possibility to predict any unseen value given some observed ones is of utmost importance. Therefore, throughout this chapter we will focus only on the regression task.

Up to the last section, we will use the notation commonly found in machine learning settings. Later, we will explain how to relate each concept to our shape modelling problem. So, we consider a dataset $\mathcal{D} = \{(x_i, y_i) \quad | i = 1 \dots N\}$ of noisy observations $y_i \in \mathbb{R}^1$ for points $x_i \in \mathbb{R}^D$. In a matrix notation, we write $\mathcal{D} = \{X, y\}$, where matrix $X = [x_1, \dots, x_N]$ and vector $y = [y_1, \dots, y_N]^T$. In the regression problem, y_i can take any real value (differently from classification) and we aim at predicting the output for unobserved values $X_* = [x_1^*, \dots, x_P^*]$.

3.2 Gaussian Process Regression

The two main views over GPR are the *weight-space* and *function-space* — these are two possible interpretations of the GPR model that are equivalent. They put emphasis on different features of the GPs and are both relevant, not only to understand the future formulation, but also to establish a relationship to other existing approaches. Therefore, we provide an overview of both points of view.

3.2.1 Weight-space view

In this setting, GPs are understood as an extension of the Bayesian formulation of linear regression [92] to a higher dimensional feature space. We start by briefly introducing Bayesian linear regression and then present the extension leading to GPs.

Bayesian linear regression model

The model assumes that the output $f(x)$ is a linear combination of the input x, with given weights w, so that $f(x) = x^T w$. Besides, the observations are taken as the true function values with additive noise $y_i = f(x_i) + \epsilon_i$, where ϵ_i is assumed to follow a Gaussian distribution with zero mean and variance σ_N^2.

Under these assumptions, we have that the *likelihood* of the observations, given the weights, and assuming that the training samples are i.i.d., is

$$p(y|X, w) = \prod_{i=1}^{N} p(y_i|x_i, w) = \mathcal{N}(X^T w, \sigma_N^2 I).$$

Then, a prior is defined over the weights w, i.e. our belief of what w should look like, before considering any observations. We assume this prior also follows a normal distribution with zero mean and covariance given by Σ, so $w \sim \mathcal{N}(0, \Sigma)$. By Bayes rule, we obtain the posterior as

$$p(w|y, X) = \frac{p(y|X, w)p(w)}{p(y|X)},$$

where the denominator is a normalization constant, denoted as marginal likelihood, that evaluates $p(y|X, w)$ over all possible values of w, and is therefore independent of the weights. More precisely, it is given as

$$p(y|X) = \int p(y|X, w)p(w)dw.$$

Theorem 3.1. *Given the regression problem* $y_i = x_i^T + \epsilon_i \ i = 1, \ldots, N$ *with* $x_i, w \in \mathbb{R}^D$, $\epsilon_i \sim \mathcal{N}(0, \sigma_N^2)$, *if we assume that* w *has a prior distribution* $\mathcal{N}(0, \Sigma)$, *where* Σ *is a positive definite covariance matrix, then the posterior also follows a Gaussian distribution and can be written as*

$$p(w|y, X) \sim \mathcal{N}\left(\frac{1}{\sigma_N^2} A^{-1} X y, A^{-1}\right),$$

where $A = \sigma_N^{-2} X X^T + \Sigma^{-1}$.

For the proof see [86].

Finally, the predictive distribution is found by averaging all of the possible models weighted by the posterior $p(w|y, X)$. Consequently, we have

$$p(f_*|x_*, y, X) = \int p(f_*|x_*, w) p(w|y, X) dw = \mathcal{N}\left(\frac{1}{\sigma_N^2} x_*^T A^{-1} X y, x_*^T A^{-1} x_*\right).$$

Gaussian Processes

Given the limitations of having a linear model, one may extend this by projecting the inputs onto higher dimensional space, through a given set of basis functions and then consider the linear combination on that space. For this, we take the function $\phi(x)$, mapping the input in D dimensions to an M dimensional one, with $M > D$. The difference to the linear model is simply that the weight vector w is now M dimensional as well, and instead of the input x or X, we shall have the transformed one, $\phi(x)$ or $\Phi(X)$, where X is the data matrix, and $\Phi(X)$ is the matrix of the transformed data.

Replacing x by $\phi(x)$ in all the above equations, the predictive equation becomes

$$p(f_*|x_*, y, X) = \mathcal{N}\left(\frac{1}{\sigma_N^2} \phi(x_*)^T A^{-1} \Phi(X) y, \quad \phi(x_*)^T A^{-1} \phi(x_*)\right),$$

where $A = \sigma_N^{-2} \Phi(X) \Phi(X)^T + \Sigma^{-1}$ Note that the dimension M of the new feature space is expected to be considerably large and the previous equation entails the inversion of matrix of size $M \times M$. The way to overcome this limitation is the known *kernel trick*. First, we observe that the previous equation can be written as

$$\begin{aligned} p(f_*|x_*, y, X) = \mathcal{N}\Big(&\phi_*^T \Sigma \Phi (\Phi^T \Sigma \Phi + \sigma_N^2 I)^{-1} y, \\ &\phi_*^T \Sigma \phi_* - \phi_* \Sigma \Phi (\Phi^T \Sigma \Phi + \sigma_N^2 I)^{-1} \Phi^T \Sigma \phi_*\Big), \end{aligned} \tag{3.1}$$

where the dependence on x as been omitted, so that $\Phi = \Phi(X)$ and $\phi_* = \phi(x_*)$. Note that the feature space always appears in terms of the structure $\phi^T \Sigma \phi$, with different inputs on ϕ. We then define

$$k(x, x') = \phi(x)^T \Sigma \phi(x'), \tag{3.2}$$

denoted as *covariance function* or *kernel*, and see that $\phi^T \Sigma \phi$ is an inner product with respect to Σ. Every occurrence of $\phi^T \Sigma \phi$ can be replaced by $k(x, x')$, to avoid computing the feature space. The formulation

can therefore be done entirely within a function space, without resorting to a parametrization with weights, as evidenced in the next section.

3.2.2 Function-view

The function space view is generally the one used when applying GPs in practice. Here, we interpret the GP as a distribution over functions, where realizations of random variables in the definition of a GP correspond to the values of a function $f(x)$. In this context, we can reformulate the definition of a GP as follows:

Definition 3.2. *A Gaussian Process is a probability distribution over functions $y(x)$ such that the set of values of $y(x)$ evaluated at an arbitrary set of points $x_1, \ldots, x_N$ jointly have a Gaussian distribution.[92]*

A GP $f(x)$ is fully specified by its mean $\mu(x)$ and covariance function $k(x, x')$ defined as

$$\begin{aligned} \mu(x) &= \mathbb{E}[f(x)] \\ k(x, x') &= \mathbb{E}[(f(x) - \mu(x))(f(x') - \mu(x'))] \end{aligned}$$

and usually written as

$$f(x) \sim \mathcal{GP}(\mu(x), k(x, x')).$$

We will consider $\mu(x) = 0$ as it is commonly done, although this can easily be generalized for any $\mu(x)$. Nonetheless, a simple pre-processing of the data allows the assumption of zero mean.

An important aspect of GP is the marginalization property (also known as consistency), ensuring that when the GP specifies $(X_1, X_2) \sim \mathcal{N}(\mu, \Sigma)$, it must also specify $X_1 \sim (\mu_1, \Sigma_1)$, with Σ_1 the respective submatrix of Σ. This property makes it possible to handle GPs despite their infinite characteristic, by marginalizing over the infinitely many variables outside of the scope of interest.

Once again, we assume the additive noise model, $y_i = f(x_i) + \epsilon$, where $\epsilon \sim \mathcal{N}(0, \sigma_N^2)$. Since by definition any finite number of points follows a multivariate Gaussian distribution, the joint distribution of the training and test points is given as

$$\begin{bmatrix} y \\ f_* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} K_{XX} + \sigma_N^2 I_N & K_{XX^*} \\ K_{X^*X} & K_{X^*X^*} \end{bmatrix}\right)$$

where $K_{XX} = [k(x_i, x_i)]_{i=1}^N \in \mathbb{R}^{N \times N}$ is the kernel matrix of the observed points, $K_{X^*X^*} = [k(x_i^*, x_i^*)]_{i=1}^P \in \mathbb{R}^{P \times P}$ is the kernel matrix of the predicted points, $K_{X^*X} = [k(x_j^*, x_i)]_{j,i=1}^{P,N} \in \mathbb{R}^{P \times N}$ is the kernel matrix of the predicted and observed points, $K_{XX^*} = K_{X^*X}^T$ its transpose and I_N is the identity matrix of size N.

Given the joint distribution, we obtain the predictive equations by taking the conditional distribution $f_*|x_*, y, X$, using (A.4). This leads to another Gaussian distribution given as

$$f_*|X, y, X_* \sim \mathcal{N}(\bar{f}_*, cov(f_*)), \tag{3.3}$$

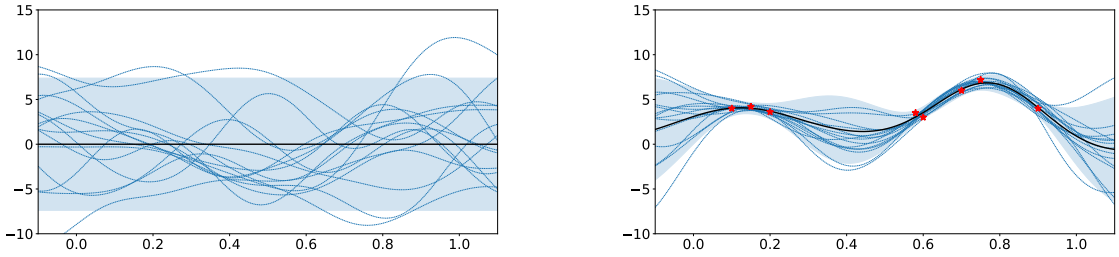
where

$$\begin{aligned} \bar{f}_* &= K_{X^*X}\Big[K_{XX} + \sigma_n^2 I_N\Big]^{-1} y \\ cov(f_*) &= K_{X^*X^*} - K_{X^*X}\Big[K_{XX} + \sigma_n^2 I_N\Big]^{-1} K_{XX^*}. \end{aligned} \tag{3.4}$$

These equations were obtained under the assumption of Gaussian likelihood which allows for the closed form solution presented above. We note that for non-Gaussian likelihoods this is no longer possible and it is necessary to resort to approximations.

Finally, we present a visual representation of GPR under the function-space view in Figure 3.1, where we show the prior defined by the GP and the posterior after observing some training points. In this setting we can interpret the prior as a set of permissible functions and the posterior as a further restriction on these functions, as they should pass close to the training data points. The notion of "how close" is expressed through the observation noise σ_N^2, whose effect on the posterior is demonstrated in Figure 3.2.

It is easy to see that by replacing the kernel in (3.1) as (3.2) we obtain the same distribution as in (3.4), thus showing that the two formulations are equivalent. While in the weight-space we formulate the prior on some parameters w, here we place it directly on the function without parameters. For instance under this view the prior for Bayesian regression would be $\mathcal{GP} \sim (0, \phi(x)^T \Sigma \phi(x)')$.



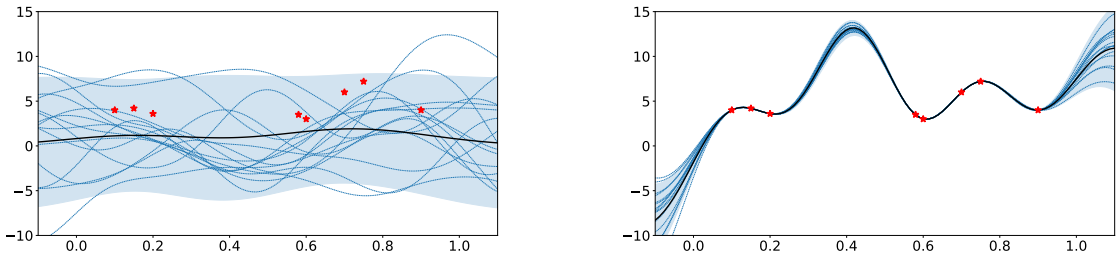
(a) Prior

(b) Posterior

Figure 3.1: Visual representation of GPR. On the left we see the prior, i.e the admissible functions according to our model, represented with light blue dashed lines. We notice that the GP has a zero mean, indicated by a black line. The shaded area represents the 95% confidence interval. On the right, we see the training points depicted with red stars. The posterior now contains functions which pass close to the observed points. Therefore, the mean has changed to indicate this and we see that the variance is much smaller near the points, increasing as we get away from them.

3.2.3 Decision theory

The posterior distribution is not enough to obtain a prediction, we need to define an optimal criterion for a single point estimate of $f(x^*)$. This means defining a loss function $\mathcal{L}(y^*_{\text{true}}, y^*_{\text{pred}})$, where y^*_{true} is the true value for y and y^*_{pred} the one predicted by us. In a Bayesian setting the choice of prediction values is separated from the inference, i.e. first the distribution is computed and then a decision is taken depending on the choice of a cost function, that determines how incorrect predictions are penalized.



(a) $\sigma_N^2 = 100$ (b) $\sigma_N^2 = 10^{-4}$

Figure 3.2: Impact of observation noise. All the plot elements have the same meaning as in Figure 3.1. Both figures represent the posterior of a GP after observing the training points, with same parameters except σ_N^2 which is taken to extreme values. For high values of σ_N^2 the confidence in the observations is smaller and the prior remains almost unchanged, while for low values of σ_N^2 there is almost no uncertainty near the observed regions.

Given a loss function, the expected loss (risk) is given as

$$R_{\mathcal{L}} = \int \mathcal{L}(y^*_{\text{true}}, y^*_{\text{pred}}) p(y^*_{\text{true}} | x^*, X, y) dy^*_{\text{true}}, \tag{3.5}$$

and therefore we want to find the value of y^*_{pred} minimizing the loss, so

$$y^*_{\text{optimal}} = \underset{y^*_{\text{pred}}}{\text{argmin}}\, R_{\mathcal{L}}. \tag{3.6}$$

Common loss functions are the absolute error $|y^*_{\text{true}-y^*_{\text{pred}}|}$ and the squared error $(y^*_{\text{true}} - y^*_{\text{pred}})^2$. For these two losses y^*_{optimal} is given by the median and mean of the posterior, respectively [86].

3.3 Kernels

From the previous introduction we can see that the kernel is a critical element of the GP and its choice should entail information on the problem at hand. Here we introduce the basic concepts on this topic (Section 3.3.1), the most commonly used kernels (Section 3.3.2) and how they can be represented through eigenfunctions and eigenvalues (Section3.3.3).

3.3.1 Basic concepts

In general, a kernel [86] is a function that maps a pair of inputs $x, x' \in \mathcal{X}$ to $\mathbb{R}$ and its associated Gram matrix, given an input $x = \{x_1, \ldots, x_n\}$, is the $n \times n$ matrix, with $G_{ij} = k(x_i, x_j)$. However, not all kernels are valid covariance functions. For a kernel to be a valid covariance function it should be positive definite, that is

$$\sum_i^N \sum_j^N k(x_i, x_j) c_i c_j \geq 0,$$

for any $n \in \mathbb{N}$, $x_1, \ldots, x_N \in \mathbb{R}$ and $c_1, \ldots, c_n \in \mathbb{R}$ or equivalently the respective Gram matrix should be positive semi-definite (for any given input), in which case, it is called covariance matrix.

Several types of valid kernels have been proposed in the literature. They can be split into two main classes: *stationary* and *non-stationary*. If k is a function of $x - x'$, we denote it as *stationary*, meaning that it is invariant to simultaneous translation of its variables, i.e. $k(x, x') = k(x + p, x' + p)$. On the other hand, for a non-stationary kernel the correlation between data points will depend on their location. For stationary kernels, if k is in particular a function of $\|x - x'\|$, we say it is *isotropic* and otherwise *anisotropic*.

3.3.2 Examples

We will now look at a few commonly used kernels, together with their relevant properties and behaviour. This is far from being an extensive review and aimed mostly at revealing the meaning behind kernel parameters and how they may differ, especially focusing on the ones that we will mostly use for shape modelling. Therefore, we will cover the Squared Exponential, Matérn and Rational Quadratic kernels, all of them being stationary isotropic. Other kernels may be found in [86] and for instance, non-stationary kernels, not covered here, can be found in [93].

Squared Exponential kernel

The *Squared Exponential* (SE) kernel (also known as *Radial Basis Function* (RBF) or *Gaussian* kernel) is without doubt the most used in GP applications and it is defined as

$$k_{SE}(x, x') = \sigma_{SE}^2 \exp -\frac{\|x - x'\|^2}{2l_{SE}^2}, \tag{3.7}$$

where σ_{SE}^2 is the kernel variance and l_{SE} the characteristic lengthscale, the scale on which the process is expected to present significant changes. We provide an intuitive idea of what these two parameters represent in Figure 3.3. This function is infinitely differentiable and has been argued to be too smooth to model some processes, motivating the Matérn kernels that follow. Another important consideration is that the SE kernel is equivalent to a representation of X in a feature space defined by Gaussian-shaped basis functions.

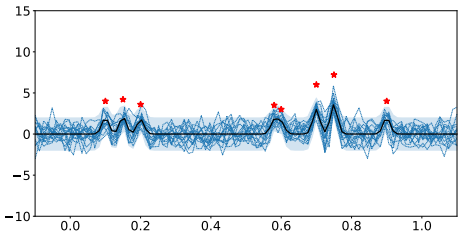
Matérn class of kernels

With the intent of producing a less smooth kernel than the SE, the Matérn kernels have been proposed as

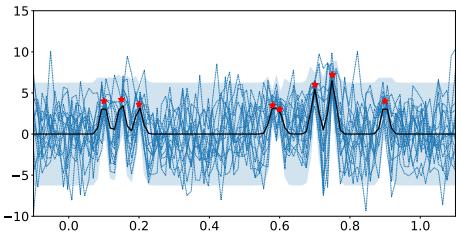
$$k_{Matern}(x, x') = \sigma_{Matern}^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \Big(\frac{\sqrt{2\nu}\|x - x'\|}{l_{Matern}} \Big)^{\nu} K_{\nu} \Big(\frac{\sqrt{2\nu}\|x - x'\|}{l_{Matern}} \Big), \tag{3.8}$$

where $\sigma_{Matern}^2, \nu_{Matern}, l_{Matern} > 0$, $\Gamma(x)$ is the Gamma function and K_{ν} is a modified Bessel function [94] (note that for $\nu \to \infty$ we get the SE kernel). This class is specially appealing for $\nu = p + 1/2$, where p is a non-negative integer, as the function can be expressed as a product of an exponential and polynomial. In particular, for machine learning, the common choices are $\nu = 3/2$ and $\nu = 5/2$. [1] In this

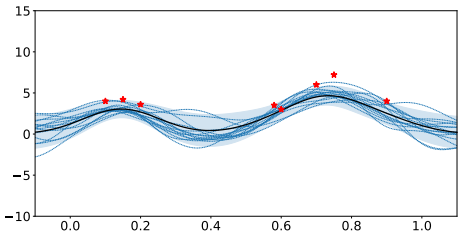
[1] For $p = 0$ the process is too rough and for $p \geq 3$ too similar to the SE kernel



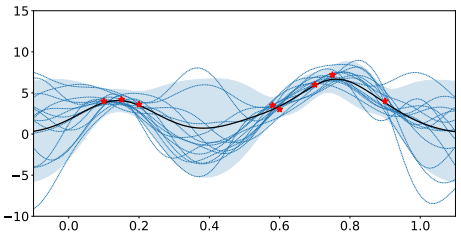
(a) $\sigma^2_{SE} = 1, l_{SE} = 0.01$



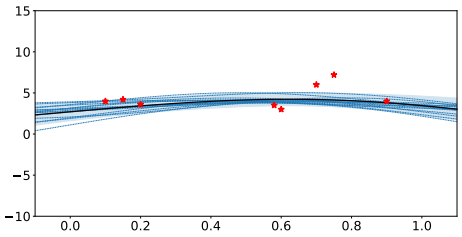
(b) $\sigma^2_{SE} = 10, l_{SE} = 0.01$



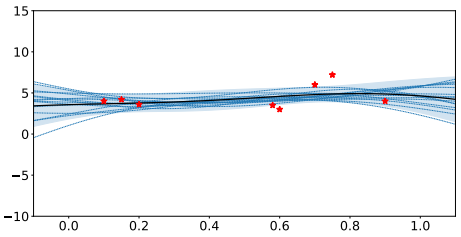
(c) $\sigma^2_{SE} = 1, l_{SE} = 0.1$



(d) $\sigma^2_{SE} = 10, l_{SE} = 0.1$



(e) $\sigma^2_{SE} = 1, l_{SE} = 0.5$



(f) $\sigma^2_{SE} = 10, l_{SE} = 0.5$

Figure 3.3: Variation of SE kernel parameters. All the plot elements have the same meaning as in Figure 3.1. Both figures represent the posterior of a GP after observing the same training points (red stars), with the same observation noise $\sigma^2_N = 1$. On the rows the kernel variation is seen for values of 1 and 10 for σ^2_{SE}, essentially increasing the amplitude of the function distance from the mean. On each column, the lengthscale increases from 0.01 to 0.5. This determines "how fast the functions are allowed to change" — for lower values there is a quick variation of the function and for higher ones the function becomes smoother.

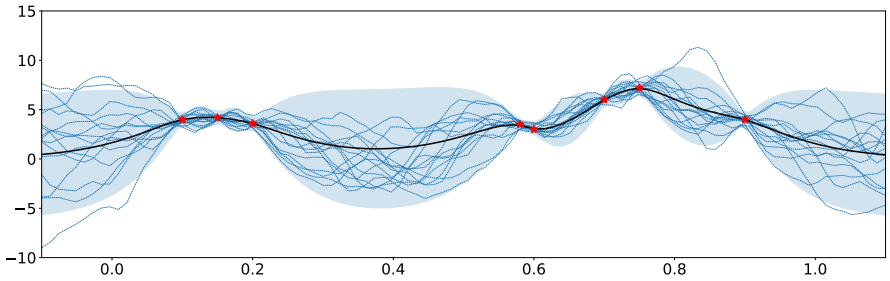
case the kernels can be respectively expressed as

$$k_{M3/2}(x, x') = \sigma^2_{Matern}\Big(1 + \frac{\sqrt{3}\|x - x'\|}{l_{Matern}}\Big) \exp\Big(-\frac{\sqrt{3}\|x - x'\|}{l_{Matern}}\Big), \tag{3.9}$$

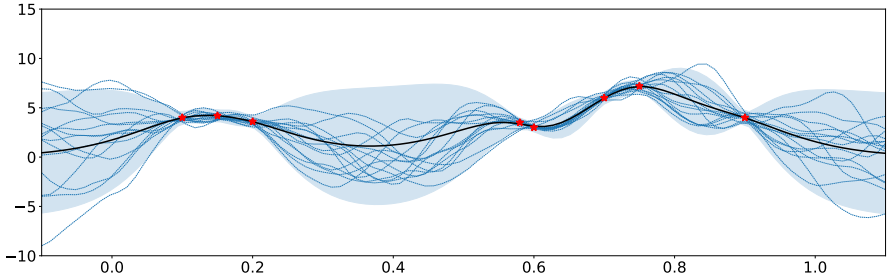
and

$$k_{M5/2}(x, x') = \sigma^2_{Matern}\Big(1 + \frac{\sqrt{5}\|x - x'\|}{l_{Matern}} + \frac{5\|x - x'\|^2}{3l^2_{Matern}}\Big) \exp\Big(-\frac{\sqrt{5}\|x - x'\|}{l_{Matern}}\Big). \tag{3.10}$$

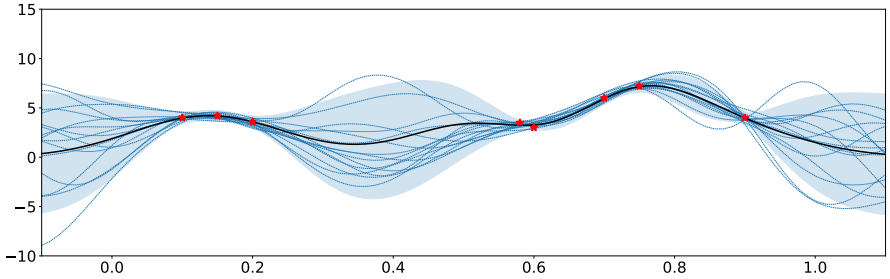
Essentially, the Matern kernels behave as a less smooth version of the SE, as demonstrated by Figure 3.4.



(a) Matern kernel 3/2



(b) Matern kernel 5/2



(c) SE kernel

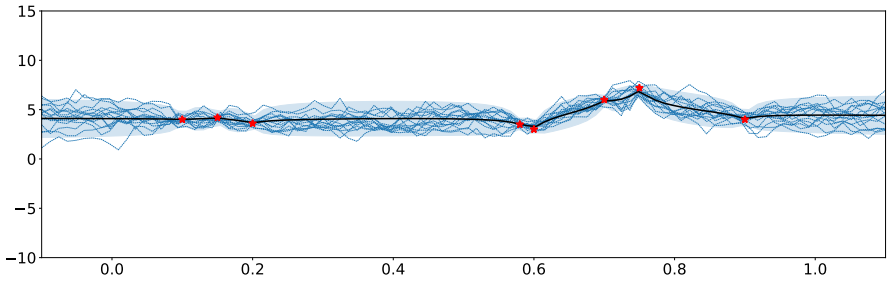
Figure 3.4: Comparison between SE and Matern kernel. Posterior of a GP after observing the same training points (black), with the same parameters $\sigma_N^2 = 0.1$, $\sigma_{SE}^2 = \sigma_{Matern}^2 = 10$, $l_{SE} = l_{Matern} = 0.1$, for 3 different kernels. Both Matern kernels are less smooth than the SE, although they still resemble it and their parameters have the same meaning. Besides, as we increase ν, we progressively get closer to the SE kernel.

Rational Quadratic kernel

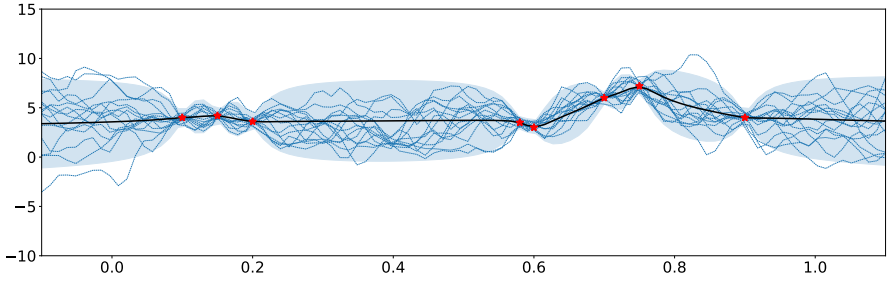
The Rational Quadratic (RQ) kernel is defined as

$$k_{RQ}(x, x') = \sigma_{RQ}^2 \Big(1 + \frac{\|x - x'\|^2}{2\alpha l_{RQ}^2}\Big)^{-\alpha}, \tag{3.11}$$

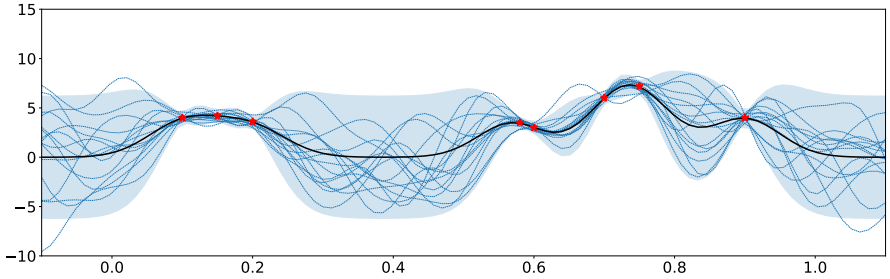
where $\sigma_{RQ}^2, \alpha, l_{RQ} > 0$. It corresponds to an infinite sum of SE kernels with different characteristic lengthscales (note that also for this kernel when $\alpha \to \infty$ we get the SE kernel). The parameter α determines the weighting between the different lengthscales and its effect is illustrated in Figure 3.5.



(a) RQ kernel $\alpha = 0.001$



(b) RQ kernel $\alpha = 0.01$



(c) SE kernel

Figure 3.5: Comparison between SE and RQ kernel. Posterior of a GP after observing the same training points (black), with the same parameters $\sigma_N^2 = 0.1$, $\sigma_{SE}^2 = \sigma_{Matern}^2 = 10$, $l_{SE} = l_{Matern} = 0.005$, for 3 different kernels. The RQ kernel allows for localized deformations with smaller lengthscale, while for regions without observations the original lengthscale is used. Besides, as we increase α, we progressively get closer to the SE kernel.

Combining kernels

Although it might be challenging to prove the validity of a new kernel, it is particularly easy to obtain more complex kernels from existing ones, as their space is closed under many operations. The following are valid kernels, given $k_1(x, x')$ and $k_2(x, x')$ also valid,

- $k(x, x') = k_1(x, x') + k_2(x, x')$
- $k(x, x') = k_1(x, x')k_2(x, x')$
- $k(x, x') = g(x)k_1(x, x')g(x')$ for any function $g(x)$
- $k(x, x') = x^T A x'$ for A symmetric positive definite matrix.

Some intuition is due regarding the first two operations, to be used later in this work. Multiplication enforces an **AND** operation between outputs, and so two points will be highly correlated only if they are for both kernels. Summation translates to an **OR** operation, so that a pair of points will be highly correlated in either one of them. Further insight on kernel operations can be found in [95].

3.3.3 Mercer's theorem

A kernel can be expressed in terms of its eigenvalues and eigenfunctions [86]. An eigenfunction of a kernel k is a function $\phi(x)$ that obeys the following equation

$$\int k(x, x')\phi(x)d\mu(x) = \lambda\phi(x') \tag{3.12}$$

with respect to a measure μ, where λ is an eigenvalue. There are an infinite number of eigenfunctions $\phi_1(x), \phi_2(x), \ldots$, assumed ordered so that the eigenvalues follow $\lambda_1 \geq \lambda_2 \geq ....$ We can now express the kernel in terms of ϕ_i and λ_i as demonstrated by Mercer's theorem below.

Theorem 3.2 (Mercer's theorem)**.** *Let $(\mathcal{X}, \mu)$ be a finite measure space and $k \in L_\infty(\mathcal{X}^2, \mu^2)$ be a kernel such that $T_k : L_2(\mathcal{X}, \mu) \to L_2(\mathcal{X}, \mu)$ is positive definite. Let $\phi_i \in L_2(\mathcal{X}, \mu)$ be the normalized eigenfunctions of T_k associated with the eigenvalues $\lambda_i \geq 0$. Then,*

1. *the eigenvalues $\{\lambda_i\}_{i=1}^{\infty}$ are absolutely summable*
2. *$k(x, x') = \sum_{i=1}^{\infty} \lambda_i \phi_i(x)\phi^*(x')$ holds μ^2 almost everywhere, where the series converges absolutely and uniformly μ^2 almost everywhere.*

Mercer's theorem allows to build the Karhunen-Loève expansion of a zero-mean GP as

$$\int u_k(x) = \sum_{v=1}^{k} u_v \phi_v(x), \tag{3.13}$$

where u_v are independent variables following $\mathcal{N}(0, \lambda_v)$. Since λ_v decays to zero, then we can approximate the GP by a finite sum. The eigenvalues and eigenfunctions of (3.12) are usually numerically computed trough the Nyström method [96].

3.4 Formulation for shape modelling

The use of GP to model shape deformations has been proposed several times in the literature [10, 11], since its probabilistic nature is an appealing setting for the definition of a statistical shape model. Furthermore, as we have seen, it is possible to express a multitude of priors with kernels in a principled and straightforward manner, thus expressing knowledge on the shape characteristics. The large freedom of kernels, means that GPs can often be found to be generalizations of many other approaches in the Statistical Shape Model literature. For instance, the large deformation diffeomorphic metric mapping framework is closely related, but it generally entails limitations on the kernels to be used. In order to extend it to new kernels, further work is necessary [97], unlike GPs where this arises naturally. Therefore,

GPs can be seen as an extremely general way to add more flexibility to the original Point Distribution Models (PDM) [11].

Besides, being defined on a continuous domain, GPs also prove advantageous within shape modelling settings. Although we will often work in the discrete domain — since deformations are defined on a template point cloud with a defined number of points — it is usually necessary to perform a previous subsampling of the samples. Working on a continuous setting means that, at any given point, we can trivially compute the shape for any desired point and, for instance, create a denser point cloud.

We follow the GP formulation in [11], with many subsequent approaches, especially focused on the human head [82, 83, 98]. This line of work puts special emphasis on achieving a unified setting for the shape modelling problem, thus building a more principled approach.

3.4.1 Problem formulation

We consider that any shape $S \in \mathbb{R}^{N_S \times D}$, with elements $s_j \in \mathbb{R}^D$, can be obtained from a template shape $T \in \mathbb{R}^{N_T \times D}$, with elements $t_i \in \mathbb{R}^D$, by adding a deformation $u(t_i) \in \mathbb{R}^D$, also denoted as u_i, to the template points. For the purpose of this introduction, let us temporarily assume that $N_S = N_T$ and point s_j of S is in correspondence with point t_i of T for $j = i$, i.e., shapes are already registered.

In this case, we have that $s_j = t_i + u_i$, according to our model. The deformations are then modelled by a Gaussian Process, such that $u(t) \sim GP(\mu(t), k(t, t'))$ with mean function $\mu : \Omega \to \mathbb{R}^d$ and a kernel $k : \Omega \times \Omega \to \mathbb{R}^{D \times D}$, where $\Omega \subset \mathbb{R}^D$ such that $T \subseteq \Omega$. Note that, because T is a discrete point set, we need to define the Gaussian Process on the continuous Ω, which in practice should correspond to the entire surface generated by T. In practice, we will usually remain in the discrete set T, but this detail is important given the definition of a GP and the need to consider a non-discrete setting in later stages of the shape modelling pipeline.

Under these assumptions, we can see that we lie in the GP framework introduced in Section 3.2, where the template points t_i correspond to the input points x_i and the deformations $u(t_i)$ to the function $f(x_i)$, the GP. Furthermore, we can observe noisy occurrences of the deformations, denoted as $\delta(t_i)$ or δ_i, corresponding to the outputs y_i in the previous formulation. Figure 3.6 provides a visual representation of this setting.

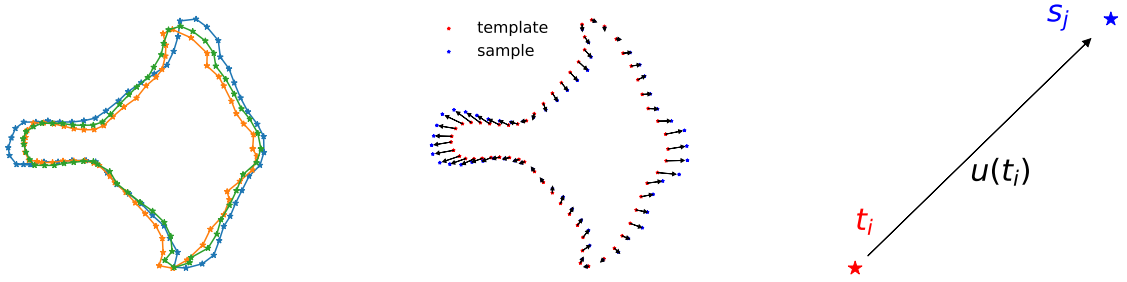
This model can be used to generate likely shapes by sampling the prior. In can also be used as a standard PCA shape model, resorting to Mercer's theorem, meaning that any fitting algorithm proposed for those models may be applied. As stated in [11], u can be approximated as

$$u(x) \approx \mu + \sum_{i=1}^{r} \alpha_i \sqrt{\lambda_i} \phi_i(x), \tag{3.14}$$

where λ_i and $\phi_i(x)$ are the eigenvalues and eigenfunctions in (3.12) and $\alpha_i \in \mathcal{N}(0, 1)$.

3.4.2 Multi-output

In the introduction we saw that GPs are defined for $y \in \mathbb{R}$, so for a scalar output. However, in order to model a deformation in higher dimensions $u(x) \in \mathbb{R}^D$, with $D > 1$, we need to formulate GP for multiple



(a) Vertebra dataset (b) Deformations (c) Notation

Figure 3.6: Representation of GP framework for shape modelling. We use the vertebra dataset from [15], containing shapes similar to the ones depicted in (a). In (b) we can see an example of how a target (in blue) is created from a template (in red) by adding deformation vectors (in black). A zoomed version is found in (c) with the respective notation.

outputs. This is known as Multioutput GP (MOGP), a topic with its origins in geostatistics, although under the term cokriging (GPR is also known as kriging). It has later been used in the machine learning community for different types of multi-task learning. In the general setting, this a studied scenario with several different possible categories. We can have different inputs for different outputs (heterotopic data) or the same (isotopic data) [99]. Here we are in the second case, since the different outputs correspond to different coordinates of a point (the former is generally used when the different outputs correspond to different tasks which, in turn, may be related to different inputs).

There are multiple approaches to extend GP to the multi output case [99–102]. We follow [99], where the output is considered to be a vector-valued function. Thus, instead of considering a single process $f(x)$, we consider several $\{f_d(x)\}_{d=1}^{D}$, one for each dimension of the output. We take the vector valued function f which follows $f \sim \mathcal{GP}(\mu, K)$, where μ is the concatenation of μ_d of each process, so $\{\mu_d(x)\}_{d=1}^{D}$ and $K(x, x')$ is a matrix-valued function with entries $k(x, x')_{d,d'}$ expressing the covariance between dimension d of input x and dimension d' of input x'. Therefore, for the discrete case, given some training data, we will have the mean as a stacked vector of each dimension and the covariance as a block partitioned matrix. For this reason, we will generally work with the vector representation of shapes, s and t.

A useful class to build valid kernels for the multi-output case are separable and sum of separable kernels [99]. They can be expressed as

$$K(x, x') = k(x, x')B,$$

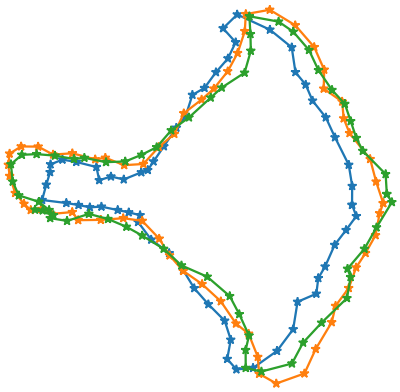
where $B \in \mathbb{R}^{D \times D}$ must be a positive semi-definite matrix. The name separable comes from the fact that input influence is considered in $k(x, x')$ and the interactions between the outputs are considered separately through B. Then, since the sum of kernels is still a valid kernel, we can obtain the class Sum of Separable Kernels as

$$K(x, x') = \sum_{q=1}^{Q} k_q(x, x')B_q.$$

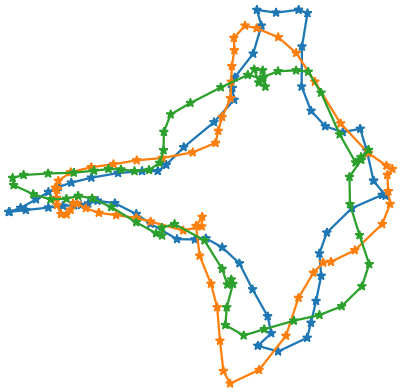
By choosing different B_q and k_q we can build different kernels from the ones defined for a scalar output. A simple choice for B is the identity matrix I_D, though it implies that there is no relation between the outputs.

3.4.3 Useful kernels

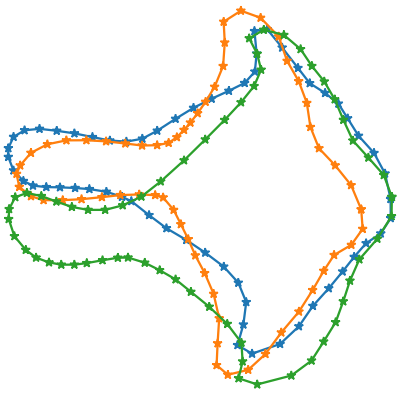
It is now convenient to interpret the previously considered kernels in the context of shape modelling. Figure 3.7 accompanies the following description, by visually depicting the effect of each kernel.



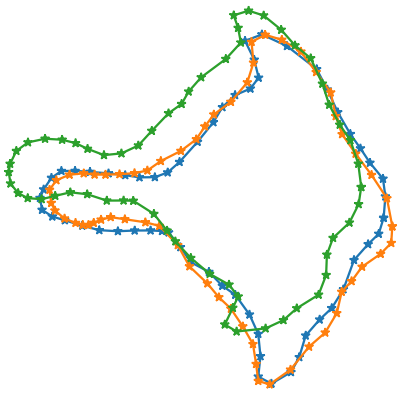
(a) Empirical kernel



(b) SE kernel with small lengthscale



(c) SE kernel with large lengthscale



(d) RQ kernel

Figure 3.7: Samples from the posterior of four different kernels. The Empirical kernel originates realistic shapes, but with reduced variability, i.e., it is restricted to the deformations observed in the training data. The two SE kernels enforce similar deformations to neighbour points (with respect to the Euclidean distance): a smaller lengthscale leads to unreasonable local deformations, while a large one approximates a rigid transformation. Finally, the RQ kernel is able to express the local deformations observed in (b), while producing reasonable shapes.

Empirical kernel. This is the extension of the Point Distribution Models (PDM) to the continuous setting and it is obtained in a similar way. So, we have that

$$\mu_{PDM}(x) = \frac{1}{M}\sum_{m=1}^{M} u^m(x), \tag{3.15}$$

where $u^m(x) \in \mathbb{R}^D$ is the deformation in shape m at x, according to the previous notation. The covariance function is then

$$K_{PDM}(x,x') = \frac{1}{M-1}\sum_{m=1}^{M}(u^m(x)-\mu_{PDM}(x))(u^m(x')-\mu_{PDM}(x'))^T, \tag{3.16}$$

with $K_{PDM}(x,x') \in \mathbb{R}^{D\times D}$. This kernel is often denoted as *PDM kernel*, *Empirical kernel* or *PCA kernel*. The continuous formulation is usually obtained from discrete training data by interpolation.

Squared Exponential and Rational Quadratic kernel The SE kernel enforces the notion of smoothness, i.e. the idea that nearby points should suffer similar deformations. Unless otherwise stated, we will build the matrix valued kernel as

$$K_{SE}(x,x') = k_{SE}(x,x')I_D = \begin{bmatrix} k_{SE}(x,x') & 0 & 0 \\ 0 & k_{SE}(x,x') & 0 \\ 0 & 0 & k_{SE}(x,x') \end{bmatrix}, \tag{3.17}$$

where we consider no correlation between the outputs. In the same way we will take $K_{RQ}(x,x') = k_{RQ}(x,x')I_D$, where the RQ kernel expresses an important concept of multiscaling: it allows for global deformations on a higher lengthscale, while also handling local ones, with lower values of l.

3.4.4 GPR in shape modelling

Following Section 3.2 we will now apply the GPR equations to the shape modelling case, with the purpose of interpreting some elements in this context and pointing out relevant remarks. Let us say we observe only a partial target, i.e., a subset of S. Given this setting, only C points in the template will have a correspondence, while the remaining ones are considered missing points. We denote the observed points as $T_{\text{corr}} \in \mathbb{R}^{C\times D}$, a subset of T , with elements t_{Ci} and the missing points as $T_{\text{miss}} \in \mathbb{R}^{(N_T-C)\times D}$, with elements t_{Mi}. Both sets are disjoint and their union leads to the full template, such that $T = \left[T_{\text{corr}}, T_{\text{miss}}\right]$.

For the points with correspondence we obtain noisy observations $\delta_i = \delta(t_i)$ of the true deformations $u_i = u(t_i)$. A representation of this setting is found in Figure 3.8. Note that the template points with observations t_{Ci} correspond to the training points x, while δ_i are the respective labels, y. We are now interested in predicting the deformations δ^* for the entire template shape T, corresponding to the testing points x^*. We are in the position to apply GPR as described in Section 3.2.2.

Restating the noise assumption of GPs, we have that $\delta(t) = u(t) + \epsilon$, where $\epsilon \sim \mathcal{N}(0,\sigma_n^2)$. We will

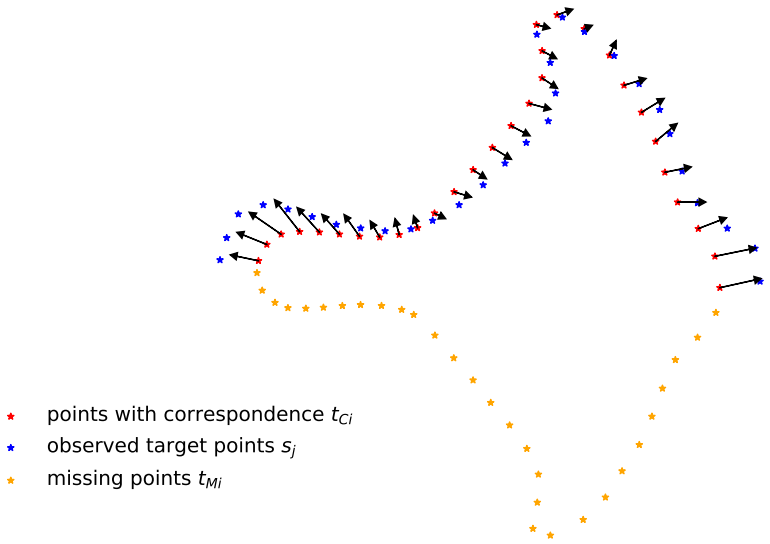


Figure 3.8: Example of a starting point for shape GPR, when only part of the target (points in blue) is seen. The points of the template are then split into point with and without correspondence. For the points with correspondence (red) we observe the deformations (black arrows), which are noisy vectors going from a point t_i to a point s_j in the target.

formulate the GPR equations (3.4) for the posterior as

$$u(t)|T_{corr}, \delta \sim \mathcal{GP}(\mu_p(t), k_p(t,t')),$$

where

$$\begin{aligned}\mu_p(t) &= K_{T_{\text{corr}}}(t)^T (K_{T_{\text{corr}}T_{\text{corr}}} + \sigma^2 I_{CD})^{-1}\delta \\ k_p(t,t') &= k(t,t') - K_{T_{\text{corr}}}(t)^T (K_{T_{\text{corr}}T_{\text{corr}}} + \sigma^2 I_{CD})^{-1} K_{T_{\text{corr}}}(t').\end{aligned}$$

We further have that $K_{T_{\text{corr}}T_{\text{corr}}} = (k(t_{Ci}, t_{Cj}))_{i,j=1}^{C} \in \mathbb{R}^{CD\times CD}$, that is

$$K_{T_{\text{corr}}T_{\text{corr}}} = \begin{bmatrix} k(t_{C1},t_{C1}) & k(t_{C1},t_{C2}) & \dots & k(t_{C1},t_{CC}) \\ k(t_{C2},t_{C1}) & k(t_{C2},t_{C2}) & \dots & k(t_{C2},t_{CC}) \\ \vdots & \vdots & & \vdots \\ k(t_{Cc},t_{C1}) & k(t_{Cc},t_{C2}) & \dots & k(t_{Cc},t_{CC}) \end{bmatrix}$$

and $K_{T_{\text{corr}}}(t) = (k(t, t_{C_i}))_{i=1}^{C} \in \mathbb{R}^{CD\times D}$, that is

$$K_{T_{\text{corr}}}(t) = \begin{bmatrix} k(t_{C1},t) \\ k(t_{C2},t) \\ \vdots \\ k(t_{CC},t) \end{bmatrix}.$$

The deformations are the concatenation vector $\delta = \{\delta_i\}_{i=1}^{C} \in \mathbb{R}^{CD}$.

We are usually interested in predicting the deformations at all the template points t_i, in which case the predictive equations become

$$\delta^* | T, T_{corr}, \delta \sim \mathcal{N}(\mu_p, D_{\sigma_P^2}),$$

where $\mu_p \in \mathbb{R}^{N_T D}$, $D_{\sigma_P^2} \in \mathbb{R}^{N_T D \times N_T D}$ are given as

$$\begin{aligned} \mu_p &= K_{T_{corr}T}^T (K_{T_{corr}T_{corr}} + \sigma^2 I_{CD})^{-1} \delta \\ D_{\sigma_P^2} &= K_{TT} - K_{T_{corr}T}^T (K_{T_{corr}T_{corr}} + \sigma^2 I_{CD})^{-1} K_{T_{corr}T}. \end{aligned} \tag{3.18}$$

The deformed template is then obtained as $\bar{\mathbf{t}} = \mathbf{t} + \mu_p$, where we take the mean of the posterior, thus following the squared error loss function. The new kernel matrices are given as $K_{T_{\text{corr}}T} \in \mathbb{R}^{CD \times N_T D}$, that is

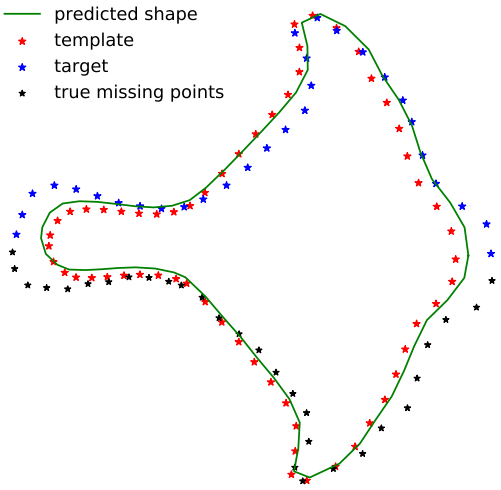
$$K_{T_{\text{corr}}T} = \begin{bmatrix} k(t_{C1}, t_1) & k(t_{C1}, t_2) & \dots & k(t_{C1}, t_{N_T}) \\ k(t_{C2}, t_1) & k(t_{C2}, t_2) & \dots & k(t_{C2}, t_{N_T}) \\ \vdots & \vdots & & \vdots \\ k(t_{CC}, t_1) & k(t_{CC}, t_2) & \dots & k(t_{CC}, t_{N_T}) \end{bmatrix}$$

and $K_{TT} \in \mathbb{R}^{N_T D \times N_T D}$, that is

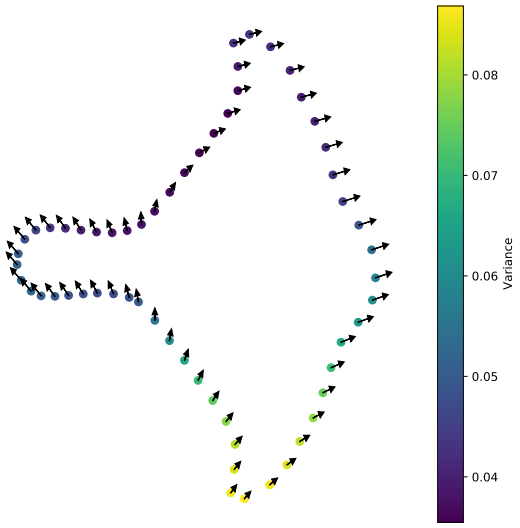
$$K_{TT} = \begin{bmatrix} k(t_1, t_1) & k(t_1, t_2) & \dots & k(t_1, t_{N_T}) \\ k(t_2, t_1) & k(t_2, t_2) & \dots & k(t_2, t_{N_T}) \\ \vdots & \vdots & \dots & \vdots \\ k(t_{N_T}, t_1) & k(t_{N_T}, t_2) & \dots & k(t_{N_T}, t_{N_T}) \end{bmatrix}.$$

Note that, if the template points remain the same (as it is usually the case), K_{TT} remains unaltered and all the other kernel matrices are sub-matrices of this one.

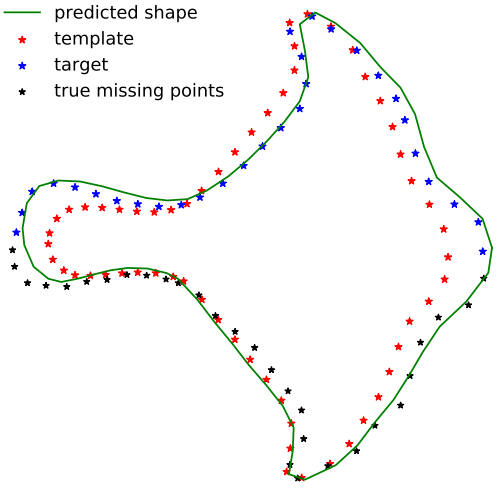
For a better understanding of regression and the posterior distribution we present Figure 3.9 and Figure 3.10. The former is the result of GPR with a SE kernel for two different σ_N^2, while the latter uses a PDM kernel. The different behaviours are evidenced not only in the final deformed template, i.e., in the mean of the posterior, but also in its variance (see the captions on both figures for a more detailed explanation of the differences). This will be particularly relevant in Chapter 5.



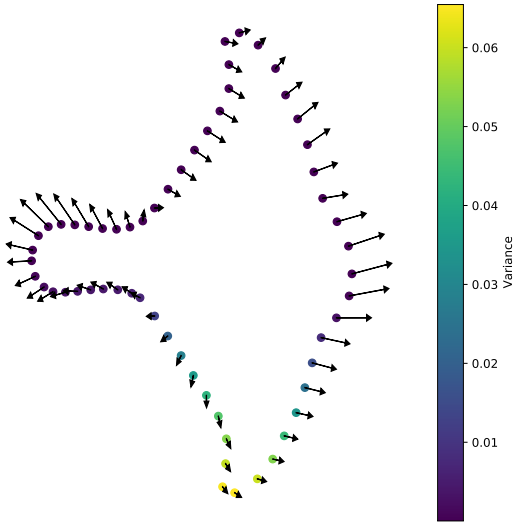
(a) Target, template and prediction for k_{SE} with high σ_N^2



(b) Posterior variance and mean for k_{SE} with high σ_N^2

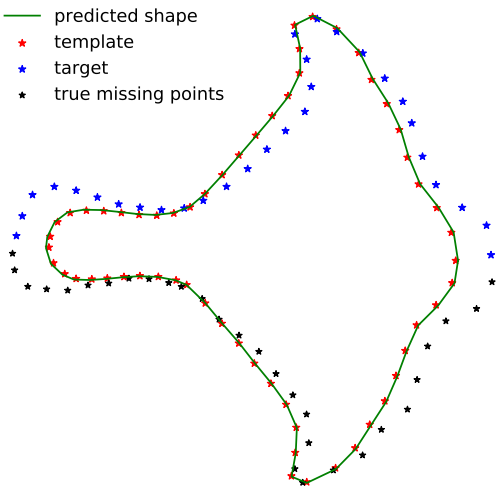


(c) Target, template and prediction for k_{SE} with low σ_N^2

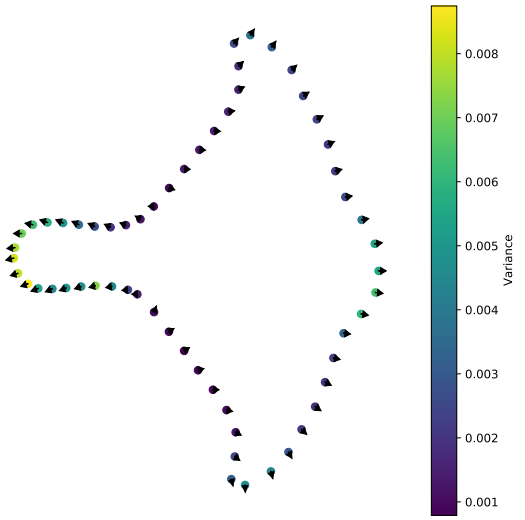


(d) Posterior variance and mean for k_{SE} with low σ_N^2

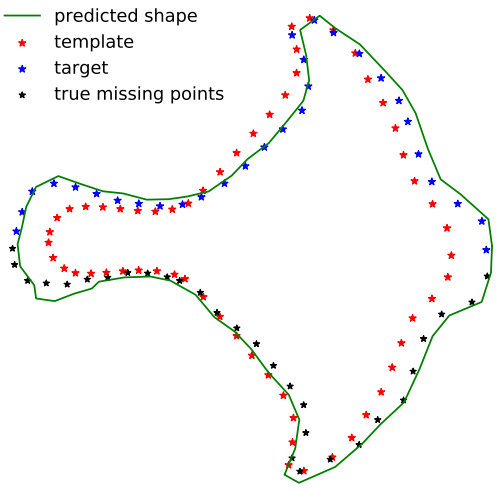
Figure 3.9: GPR for a SE kernel with the vertebra dataset. On the right column we see the points used as training (in blue) which were used to predict the full shape (blue plus black). The template is represented in red and the predicted shape from GPR in green. The two rows correspond to two different noise levels on the observations. On the right column, we find the posterior variance of each template point as a colormap, together with a vector representing the mean of the posterior. First, note that, as expected, for a high σ_N^2 we obtain a shape almost similar to the original template, with very slight deformations. It is also evident that the variance is highest in the region of missing points, progressively decreasing as we approach the location with observations.



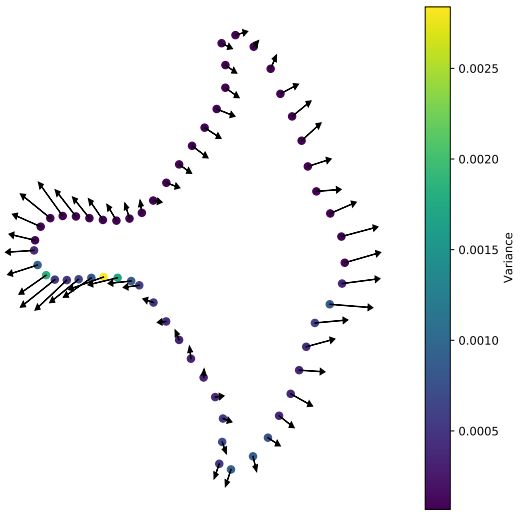
(a) Target, template and prediction for k_{PCA} with high σ_N^2



(b) Posterior variance and mean for k_{PCA} with high σ_N^2



(c) Target, template and prediction for k_{PCA} with low σ_N^2



(d) Posterior variance and mean for k_{PCA} with low σ_N^2

Figure 3.10: GPR for a PDM kernel with the vertebra dataset. On the right column we see the points used as training (in blue) which were used to predict the full shape (blue plus black). The template is represented in red and the predicted shape from GPR in green. The two rows correspond to two different noise levels on the observations. On the right column, we find the posterior variance of each template point as a colormap, together with a vector representing the mean of the posterior. We remark that the target shape used in this example was also used in the training of the PDM, which explains the almost perfect prediction, in practice this should not happen. The main interest is to compare this Figure with Figure 3.9. In particular, note that the posterior variance does vary exclusively with distance to the observations. In the top row, it should resemble almost exactly the original variance, so the regions with warmer colors are the ones with more variation in the dataset. On the bottom, we naturally see the added effect of the observations, changing the initial belief.

Chapter 4

From noisy point clouds to complete ear shapes

4.1 Introduction

As expressed in Chapter 2 the task of building a model from the *Head Dataset* brings with it a considerable amount of challenges. We have set the motivation for the use of Gaussian Processes (GP) as a preferred tool and introduced the necessary knowledge in Chapter 3. Nonetheless, two main questions remain open when it comes to this choice. First, despite having covered the modelling component to a great extent, we have overlooked the registration part within this framework. Furthermore, we have yet to show evidence that indeed GPs are necessary and an adequate choice for our problem, and how they can be applied. This chapter will answer both questions. We will mostly focus on rigid registration and how to produce a desired input so that a GP formulation may be applied. By addressing this problem we will produce a pipeline taking raw input point clouds from the *Head Dataset* as input and developing complete shapes, which can be used to produce an improved model with respect to the pre-existing *Ear Dataset*.

Most applications of GP to shape modelling assume correspondences were previously obtained by a state-of-the-art registration method [11, 80], from which Nonrigid Iterative Closest Point (NICP) is probably the most used. However, we have mentioned that this is a generic and unified framework, where the registration problem is naturally formulated as model fitting. Consequently, the immediate approach would be to remain entirely within this setting and take advantage of existing methods. However, existent proposals for registration with GP assume not only that the shapes have been previously roughly aligned, i.e., that rigid transformations were removed, but also that there is a one-to-one correspondence between both shapes, i.e., there is a low occurrence of missing data and outliers [11]. While this is a reasonable assumption for several scenarios (data of human heads, bones), it is far from being acceptable in this case. In the absence of landmarks, it is not straightforward to find an initial alignment (even a rough one), when the point clouds present the problems we have identified in Chapter 2. Even if in presence of an alignment, the quantity of missing data in the ears is much higher than usual and the one-to-one

assumption leads to results as depicted in Figure 4.1 (obtained by manually pre-aligning the shapes and naively applying the registration proposed in [11]). Given that GP provides freedom to build complex kernels, we must assume that better results could be achieved by improving the underlying model while maintaining the same method. However, we should keep in mind that our initial goal is to obtain a flexible model of the ears, since we have only access to one with limited variability (*Ears Dataset*). We seek to find a principled approach not dependent on trial-and-error strategies.

Consequently, we opt for a structured approach. Given that the main limitation are the extensive regions of missing data, we focus on completing those sections, ideally with previous knowledge from the *Ear Dataset*, while keeping the information from the existing data, as much as possible. In order to do so, it is necessary to have previous identification of which points are missing, hinting at the use of a generic registration method. It will become clear throughout the chapter that the trivial application of standard methods to our problem is not fruitful, motivating our proposal of a new complete pipeline summarized in Figure 4.2 and briefly described in the following paragraphs.

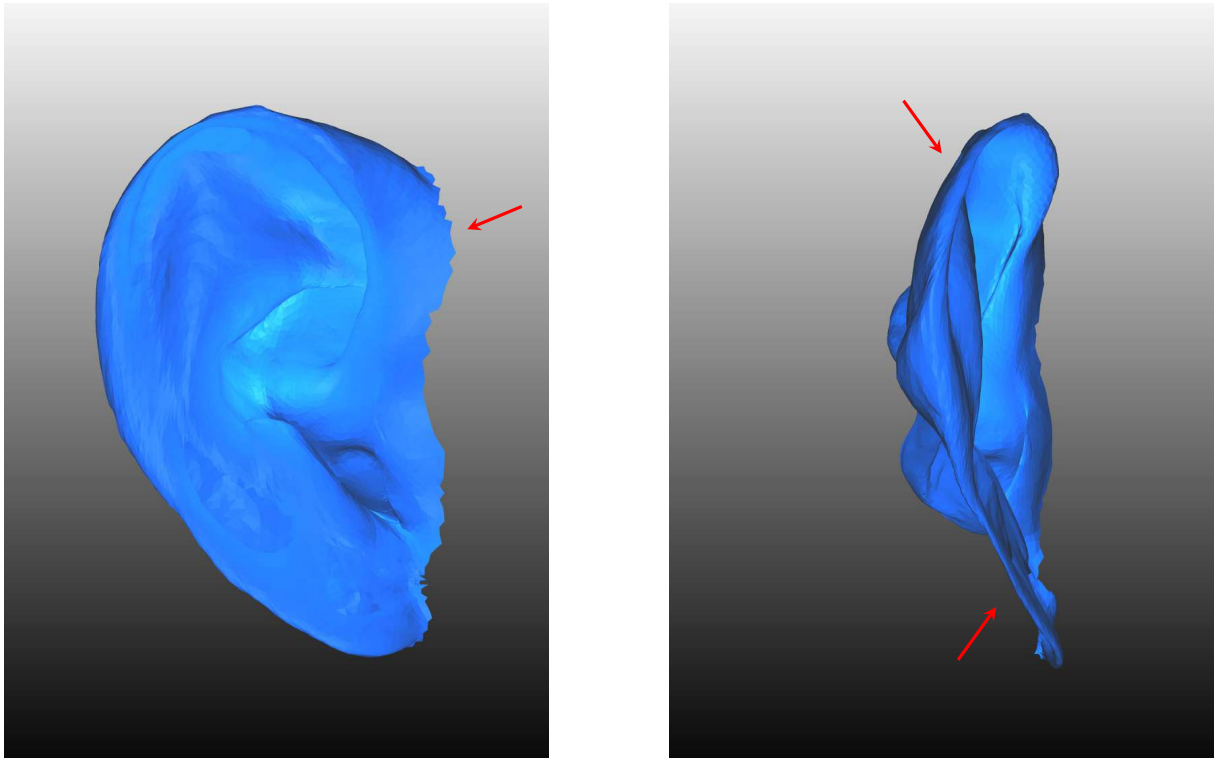
To register the samples amongst themselves, we need a template, which should be a good representation of the ear shape without any missing data or outliers. Therefore, we use the mean shape of the ear model available from [103], already introduced in Chapter 2.

The problem is then to register the template to the ear region obtained from the *Head Dataset*. However, in order to choose the best method, we need to compute evaluation metrics and so we require a ground truth, that is, we need to know the correspondence a priori. This is evidently not the case of our target data, as this is precisely the problem we are trying to solve, motivating us to use the entire *Ear Dataset* for testing purposes, where all the shapes already have the same number of vertices and are in correspondence. The obstacle is that there are no missing points, outliers or noise in any of the samples, while registration of real head scans will entail all these problems. The solution is then to introduce all these problems in the original data, in order to replicate as much as possible the difficulties of real scans, while still having the knowledge of the true correspondences. We explain this process in Section 4.2.

As seen in Figure 4.2, we propose two different registration steps. The first should place the template in rough alignment with respect to the target, mostly in terms of rotation and translation, and should identify the most distant outliers. This is the case of the skin area around the ear. After this, it is expected that non-rigid methods are able to perform better, thus refining the correspondences between the template and target. We cover the state-of-the-art literature on registration methods (both rigid and non-rigid) in Section 4.3, followed by our proposal for a rigid registration method tailored to the challenges in our data in Section 4.4. The main methods are compared throughout Section 4.5, leading to the two-step approach visible in Figure 4.2.

Given an accurate identification of correspondences to the template, we focus on predicting the missing data regions, by searching for the deformed template that most resembles each target. We compare three different approaches in Section 4.6.

After selecting the pipeline steps with the *Simulated Dataset*, we test the approach on real data, i.e., the ear regions of the *Head Dataset*. The dataset obtained as an output from the pipeline is denoted as *Reconstructed Dataset*. The results and discussion can be found in Section 4.7.



(a) Front view

(b) Side view

Figure 4.1: Example of GP registration. The red arrows indicate problematic regions in the deformed template. On the left, we a have a front view which presents overall acceptable results, except for the top part where the template deforms too much due to the outliers. On the right, we see the template collapsing both on the top and bottom of the ear due to the lack of posterior points on the target.

4.2 Metrics and datasets

4.2.1 Datasets

Missing data

The obtained scans of the *Head Dataset* have missing points, not only uniformly spread, but also concentrated in particular regions of the ear which are more difficult to capture by the scanning process. Therefore, in the *Ear Dataset* we introduce

- Uniform missing data: by randomly removing data points from the entire shape, with a ratio of 0.2, i.e. we uniformly remove 20% of points in the whole shape.
- Structured missing data: by randomly removing data points from a particular area, with a higher ratio of 0.8. This consists of a single region, manually defined on the inner and bottom part of the ear and identified by observation of the *Head Dataset*, aiming at replicating it as close as possible.

Outliers

The ear region of the *Head Dataset* also contains outliers (points with no correspondence in the template), both uniformly spread and distributed in a structured manner. In particular, the structured outliers come from the fact that when we cut the ear portion from the entire head of the scan we do not

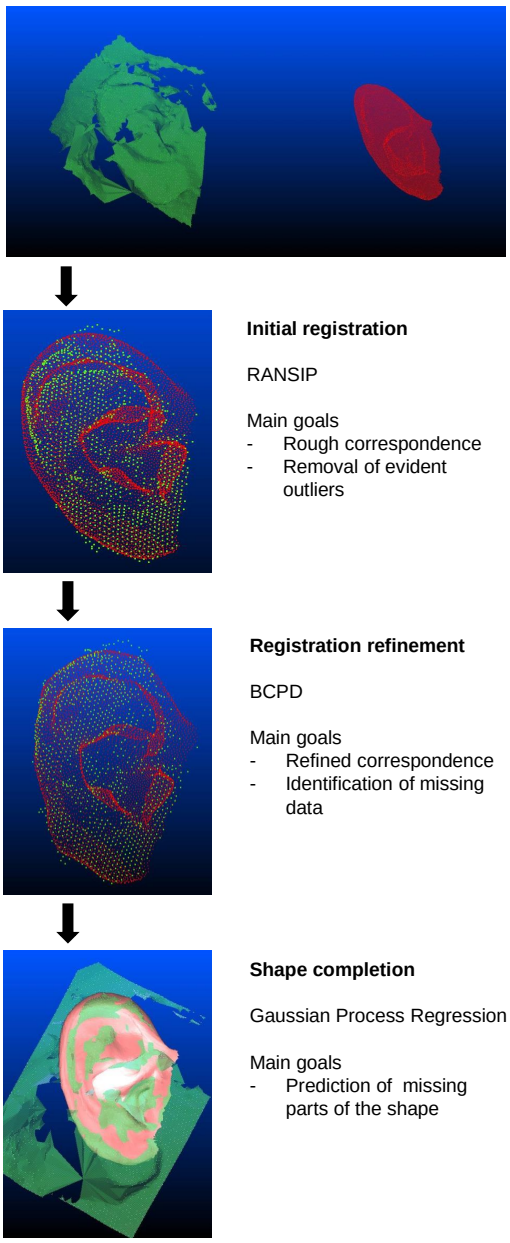


Figure 4.2: Pipeline scheme. In red we observe the template being registered to the target shape in green. The top image shows the initial position of both point clouds, where we can see that there is considerable rotation and translation between them. The first step is the initial registration which should place both shapes in a reasonable aligment and remove the most clear outliers, in general the points of the skin around the ear. The result of this step is depicted in the second image, where the green points are the inliers kept from the initial target. Afterwards an improved correspondence is achieved by allowing non-rigid deformations with BCPD. Finally, the points of the template without correspondence are predicted with Gaussian Process Regression, producing the red shape in the final picture, thus originating the *Reconstructed Dataset.*

know exactly which points belong to the ear, and consequently include some extra points, as one can see in Figure 4.3(b). Therefore, in the *Ear Dataset* we add

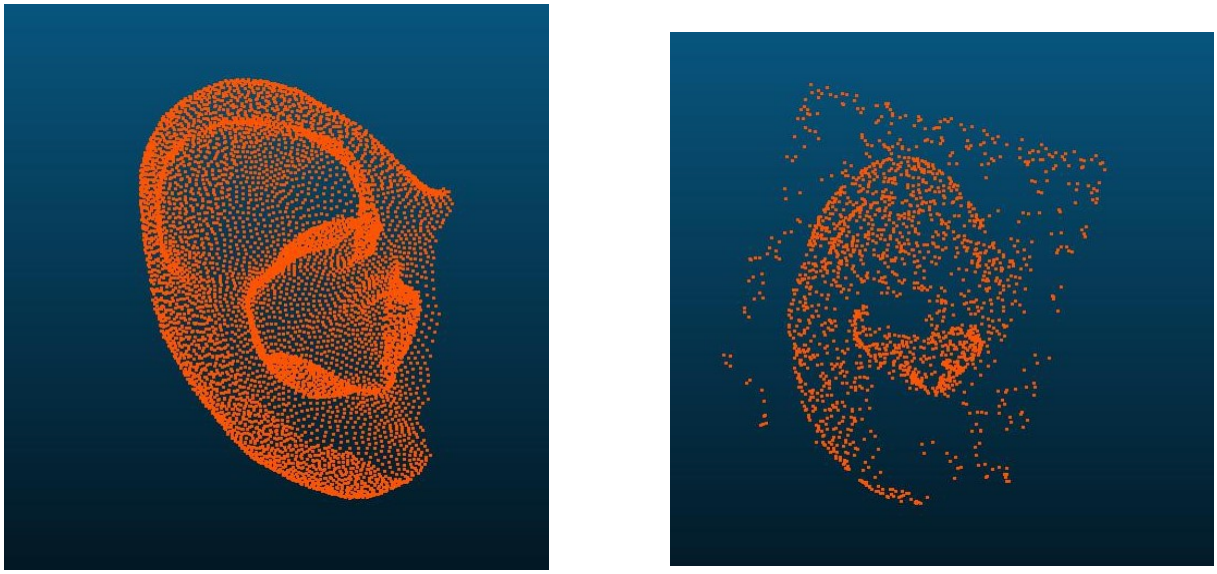
- Uniform outliers: by introducing additional points over a bounding box containing the entire shape, with a ratio of 0.1.
- Structured outliers: by introducing additional points over a particular area of the shape, with a higher ratio of 0.4. This regions were manually defined in order to cover the skin around the ear found in most scans, as observed in the example shape of Figure 4.3.

Measurement Noise

For each point in the *Ear Dataset* we introduce Gaussian noise with zero mean and a chosen variance, so that they are slightly displaced, to simulate the lack of complete accuracy in the screening process.

Simulated dataset

Altering the original data of the *Ear Dataset* with the previously stated processes, we obtain a noisy dataset (denoted as *Simulated Dataset*) that should replicate the point clouds obtained from the real head scans. Figure 4.3 exemplifies this process, showing the mean shape of the ear model (left) and the tampered ear produced from the original data (right), attempting to replicate the problems.



(a) Mean shape of the *Ear Dataset*

(b) Shape from the *Simulated Dataset*

Figure 4.3: Representation of the *Simulated Dataset*, obtained from the *Ear Dataset* as described in Section 4.2.1. The mean ear of the latter is represented to provide a visual comparison. The most noticeable features are the outliers surrounding the main shape and the extensive missing region on the bottom part.

4.2.2 Registration metrics

Often the registration methods focus on minizing the distance between both shapes. However, this may be misleading in some cases, especially when dealing with extensive missing points and outliers.

We propose to simultaneously look at four different metrics, which together allow us to get a good picture of the results and ensure that one of them is not minimized (or maximized) at the cost of others. Furthermore, we will focus on different metrics at different points of the pipeline.

Metrics for registration focusing on measurement noise

We consider two different metrics to evaluate the registration results :

Fraction of correspondences. Computed as the number of correspondences found over the total number of points in the template for which a correspondence exists. We want this to be as close to 1 as possible. A common metric for similar cases would be the Jaccard Index, where given two sets (A and B) we compute the similarity between them as $\frac{|A \cap B|}{|A \cup B|}$. In our case, if A is the set of points of the template and B the set of the target, our metric could be expressed in this notation as $\frac{|A \cap B|}{|A|}$. We use the latter since the template corresponds to the ground truth and we want to relate all samples to it, regardless of the number of points of each target.

Distance error. For each point in the template we identify the true correspondence in the target and the registered point by the method, computing the Euclidean distance between them. The true correspondence is possible to retrieve since the original *Ear Dataset* is registered. This metric then expresses the average distance for all points in the shape. So, given the template point cloud with N_T points, the original target point corresponding to the i-th point given as $s_i \in S$ and the registered target point given as $\tilde{s}_i \in S$, this error is expressed as

$$D = \frac{1}{N_T} \sum_{i=1}^{N_T} \|s_i - \tilde{s}_i\|.$$

The two previous metrics must be evaluated simultaneously since we want to identify as many points of the template as possible (fraction of correspondences), but we also require them to be correctly matched (distance error).

Metrics for registration focusing on outliers and missing data

We need to specifically evaluate the performance of the methods when it comes to identifying outliers and missing data, given that these are the two main challenges in our data. For this purpose, we will use metrics typically used in classification problems, since we are essentially classifying each point as either inlier, outlier or missing. Therefore, taking as an example the outliers, we define

- True Positives (TP) as the outliers identified as such;
- True Negatives (TN) as the non outliers identified as such;
- False Positives (FP) as the points incorrectly classified as outliers;
- False Negatives (FN) as the outliers not identified as such.

Considering this, we define the following metrics

- *Precision* (or Specificity) = $\frac{TN}{TN+FP}$ (expresses the ability to correctly identify non outlier);
- *Recall* (or Sensitivity) = $\frac{TP}{TP+FN}$ (expresses the ability to correctly identify outliers).

In the same way, we consider these metrics for missing points.

4.3 Registration methods

In this section we will review the most relevant types of registration methods, as we search for one fitting the ears characteristics. On one hand, this is meant to explain the essence of each method to be used in the following sections and motivating future choices, so we will detail generic formulations of the algorithms. On the other, we will focus on the most used algorithms in shape modelling pipelines. Naturally, there are many variations of each presented method, which we will not cover in this introduction. More thorough expositions of registration methods can be found in survey papers [69–71].

In the same way as before we consider the template $T \in \mathbb{R}^{N_T D}$,to be registered to the target shape $S \in \mathbb{R}^{N_S D}$. As briefly introduced in Chapter 2, the registration problem consists of simultaneously finding the deformation and permutation of the points of T that minimize a cost function expressing the shape difference between T and S.

Usually, registration methods are split into three different categories: *Iterative Closest Point* based methods, *Probabilistic methods* and *Graph methods*, as they approach the problem in different settings. We shall follow this division. However, we focus more on the first group as they closely relate to the method that we proposed for the initial registration.

4.3.1 Iterative Closest Point based methods

One of the most popular registration method class is the Iterative Closest Point (ICP), first proposed in 1992 [104], which has since then originated multiple variations [105, 106]. The main idea behind this approach is to iteratively select a point match for each template point and, given that selection, compute the optimal transformation of the template to minimize the distance function.

In its original formulation, ICP restricts the deformation of the template to rotations Γ and translations p and minimizes the sum of the squared distances between corresponding points. Therefore, it is expressed as

$$\min_{\Gamma \in \mathcal{SO}(3), p \in \mathbb{R}^3} \sum_{i=1}^{N_s} \|\Gamma t_i + p - s_i\|^2, \tag{4.1}$$

where

$$s_i = \underset{s_k \in S}{\operatorname{argmin}} \|\Gamma t_i + p - s_k\|. \tag{4.2}$$

The solution is obtained by solving (4.1) for fixed s_i and (4.2) for fixed Γ and p, iteratively. Both problems can be easily solved. The former has a closed form solution, as it corresponds to the Orthogonal Procrustes problem in (2.4) and can be obtained with SVD, for instance. The closest points in (4.2) are usually

obtained with k-D trees [107] for an efficient search. Nonetheless, ICP presents two main drawbacks. On one hand, it is non-convex and has been found to get easily trapped in local minima, thus being rather sensitive to initialization. On the other hand, the L2 cost function used by ICP is known to struggle in the presence of outliers, a common occurrence in registration problems. Therefore, one can generally say that ICP will perform well when the two point sets have a reasonable initial alignment and each point in T has a valid correspondence. Unfortunately, this is not the case for many problems, hence the need for variations.

One option is to consider different functions, i.e. we can formulate ICP more generically as

$$\min_{\Gamma \in \mathcal{SO}(3), p \in \mathbb{R}^3} \sum_{i=1}^{N_s} \rho(\|\Gamma t_i + p - s_i\|), \tag{4.3}$$

and consider different loss functions, given as ρ. This can be interpreted as attributing different weights to each pair of points. In that light, we have hard rejection of outliers, where ρ has a binary assignment, or soft rejection, where ρ is a continuous variable. Hard assignment may be based on a maximum distance between paired points, an exclusion of pairs on a pre-determined lower percentile, a pair containing a boundary point, amongst many others [105]. Soft rejection has been implemented with the Huber function, for instance. An interesting comparison between different cost functions is found in [108].

An important variation to the error metric is *point-to-plane* ICP [109] (as opposed to *point-to-point* in (4.1)) where the cost does not correspond to the distance between two points, but rather to a point and the tangent plane of the corresponding point. The problem becomes

$$\min_{R \in \mathcal{SO}(3), p \in \mathbb{R}^3} \sum_{i=1}^{N_s} \|(\Gamma t_i + p - s_i) \cdot n_{s_i}\|^2,$$

where n_{s_i} is the normal vector computed at the target point s_i. The method has been found to present faster convergence, but to have a smaller basin of convergence [106], thus being more sensitive to the initial shapes position.

Since the previous methods find a local solution, they often assume an initial coarse alignment by a different method [110]. However, there have been attempts to find global solutions. A popular method is Go-ICP [111] where the authors use a branch and bound strategy, to minimize the original ICP cost function with the L2-norm. The method can be extended to have robustness to outliers based on trimming and the authors state that it can be extended to other cost-functions.

Another strategy proven to cope well with outliers is the use of Random Sample Consensus (RANSAC) [112] based methods [113, 114]. When applied to the registration problem, it entails repetitively sampling both point clouds and estimating the transformation parameters based on those subsets. The main idea is that there is a certain probability of picking all inliers and after a certain number of repetitions, the correct transformation will have been observed. The critical point is in how to sample points, so that fewer iterations are necessary. Note that to reach a good transformation we need to find inliers for both clouds and they must form a match, this often means that many iterations are required. The number of iterations may be reduced but this is at the cost of more complex sampling strategies which increase

the computation time of each repetition. A recent and state-of-the-art RANSAC method is Semidefinite-Based Randomized Approach (SDRSAC) [114], where sampling is based on graph matching applied to subsets of points, generally enforcing the idea that matching pairs of points in both clouds must have similar distances between them.

Non-rigid variation

While the original formulation is restricted to rigid transformations, ICP can be extended to non-rigid ones, as done in the optimal step non-rigid ICP (NICP) proposed in [115]. It is assumed that each point t_i in the template may undergo a local affine transformation represented by a transformation matrix $A_i \in \mathbb{R}^{D\times(D+1)}$, so that the parameters are contained in a matrix $A = [A_1, \ldots, A_{N_T}]^T \in \mathbb{R}^{(D+1)N_T\times D}$. The cost function is then defined as

$$E(A) = E_d(A) + \alpha E_s(A) + \beta E_l(A), \tag{4.4}$$

where α, β are parameters and $E_d(A)$, $E_s(A)$ and $E_l(A)$ are three different cost terms to be defined below.

The first term of (4.4) minimizes the weighted squared distance between the deformed template and the target,

$$E_d(A) = \sum_{i=1}^{N_T} w_i \|A_i t_i - s_i\|^2,$$

where s_i is obtained with (4.2) in the usual way. A correspondence is dropped, i.e. w_i is set to zero, if the target is on a border of the mesh, the angle between normals is larger than a fixed threshold or the line between the transformed point and the correspondent target point intersects the deformed reference shape. Consequently, this uses the L2-norm of the original ICP, except that each point can move independently and there are many more parameters to be estimated. Therefore, a restriction to the deformations is necessary and implemented with the regularization term $E_s(A)$. It penalises the weighted difference of the transformations of neighbouring vertices, with a parameter balancing the rotational and skew transformations against translation.

$$E_s(A) = \sum_{\{i,j\}\in\mathcal{E}} \|(A_i - A_j)G\|_F^2,$$

where $\|\cdot\|_F$ is the Frobenius norm, $\mathcal{E}$ are the edges connecting neighbouring vertices and G is a diagonal matrix $\mathrm{diag}(\mathbb{1}_D^T, \gamma)$, with γ controlling the translation versus rotational and skew deformations. The final term guides the registration with landmarks as

$$E_l(A) = \sum_{l=1}^{L} \|A_l t_l^* - s_l^*\|^2,$$

where $\{(t_1^*, s_1^*), \ldots, (t_L^*, s_L^*)\}$ are a set of L landmarks.

Over the iterations, NICP takes a sequence of decreasing α in (4.4), such that as the position of the

template improves, the points have increased freedom to move independently.

4.3.2 Probabilistic methods

A different approach to the registration problem is a probabilistic formulation, where the most representative method is the Coherent Point Drift (CPD) [85]. The template points, upon an appropriate transformation, are assumed to be centroids of a Gaussian Mixture Model (Appendix A.2), generating the target points. The probability density function for this scenario is therefore given as

$$p(s_j) = \sum_{i=1}^{N_T} P(t_i) p(s_j|t_i),$$

where $p(s_j|t_i) = \frac{1}{(2\pi\sigma^2)^{D/2}} \exp - \frac{\|s_j - \mathcal{T}(t_i)\|^2}{2\sigma^2}$, with σ^2 the noise variance and $\mathcal{T}(t_i)$, the i-th template point after a transformation. Point $\mathcal{T}(t_i)$ can be obtained through rigid, affine or non-rigid deformations, leading to different CPD algorithms. It is assumed that the probability of each point s_j belonging to centroid i is equal, so $P(t_i)$ is set to $1/N_T$. Outliers and noise are accounted for with an additional uniform distribution, such that the final density function is

$$p(s_j) = w\frac{1}{N_S} + (1-w)\sum_{i=1}^{N_T} \frac{1}{N_T} p(s_j|t_i),$$

where $w \in (0, 1)$ is the prior probability that point s_j is an outlier. With an i.i.d. assumption, the likelihood is given as

$$\mathcal{L}(\theta) = \prod_{j=1}^{N_S} \left[w\frac{1}{N_S} + (1-w)\sum_{i=1}^{N_T} \frac{1}{N_T} p(s_j|t_i) \right],$$

where θ includes the transformation parameters to compute $\mathcal{T}(t_i)$ and σ^2.

The parameters are found with the Expectation-Maximization algorithm [116]. In practice, this corresponds to an E-step where we compute the posterior, which can be understood as finding the correspondence in ICP, but in a probabilistic way. Followed by the M-step where we compute the transformation parameters given the correspondences, which can be understood as a weighted ICP.

Regarding the non-rigid CPD formulation, the transformation is given as

$$\mathcal{T}(t_i) = t_i + \delta_i, \tag{4.5}$$

where δ_i is a displacement vector. The regularization on deformations is enforced with a Gaussian kernel, meaning that the centroids move coherently, i.e. nearby points observe similar deformation vectors. The degree of smoothness enforced is controlled by additional parameters.

While still considered state-of-the-art, CPD does not handle particularly well outliers, missing data and different number of points between both point clouds. Consequently, variants of CPD have been developed in recent years to attempt to deal with such drawback [117–121], some restricted to rigid registration [122–124]. Examples of variations include enforcing preservation of local structure through k-connected neighbours [125] or assignment of different membership probabilities [119].

A state-of-the-art variation of CPD is the Bayesian CPD (BCPD) [126]. Unlike CPD, the transformation is defined as $\mathcal{T}(t_i) = \beta\Gamma(t_i + \delta_i) + p$, where β is a scale factor, Γ a rotation matrix, p a translation vector and δ_i the displacement vector representing a non-rigid deformation. They formulate a joint distribution for the target points, but also for explicit correspondence vectors between the two shapes. Besides, the motion coherence is expressed as a prior distribution on the displacement vectors instead of a regularization term and the optimization is not based on EM algorithm, but rather on Variational Bayesian inference (VBI). Further insight and mathematical details are left for Chapter 5, where this will be covered in detail.

4.3.3 Graph based methods

Correspondence can also be found through graph matching where generally each vertex represents a point in the point cloud. Graph matching methods can be of first, second or higher order. First order methods, using only information about each vertex, have been replaced by higher order methods and are not commonly used at the moment. Second order methods [127, 128] try to match both vertices and edges, while higher order methods [129, 130] include extra information such as angles between vertices and have the advantage of being invariant to scale and affine changes. Both can be formulated as a Quadratic Assignment Problem (QAP), which is NP-hard. Thus, most graph matching approaches are limited to a small number of nodes (at most in the order of hundreds) and are therefore not suitable for this application, since our point clouds have thousands of points.

4.4 RANSIP

The initial registration is a preliminary step which should place the template in an approximated position, i.e. remove most of the effects of translation and rotation, so that we can use the non-rigid registration algorithms. The main problem, other than missing data, is the extensive presence of outliers. These two factors make the problem extremely difficult. We propose an heuristic method, based on some considerations for our specific data, which will be seen to work well when compared to the approaches previously described.

Despite the many variations of ICP, it is quite challenging to determine the most appropriate choice in each scenario and their performance is highly dependent on the particular characteristics of each dataset. For our setting, we have found that the best outlier rejection policy is based on a maximum distance and the angle between normals of corresponding points, as proposed in [131]. To deal with local minima we employ a randomized approach, which can be considered as a simplified version of RANSAC methods [114]. Throughout this section we shall motivate our choices and describe the detail of our heuristic method, which we call RANSIP.

We propose ICP according to (4.3), with the following cost function

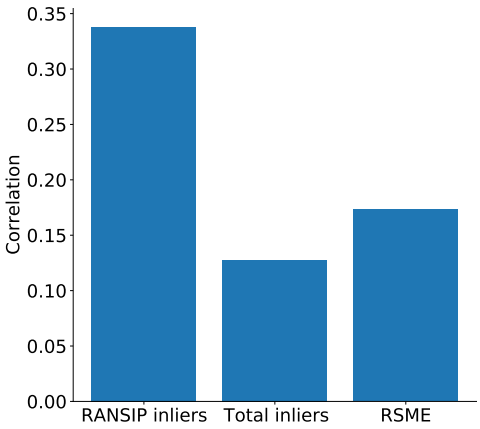
$$\rho(x_i) = \begin{cases} x_i^2 & \text{if} \quad x_i < d_{\max} \text{ and } \sin(\theta_i) < \sin(\pi/4) \\ 0 & \text{otherwise} \end{cases} \tag{4.6}$$

where θ_i is the angle between the normal vector n_{t_i} at t_i and the normal vector at s_i, n_{s_i}, so that

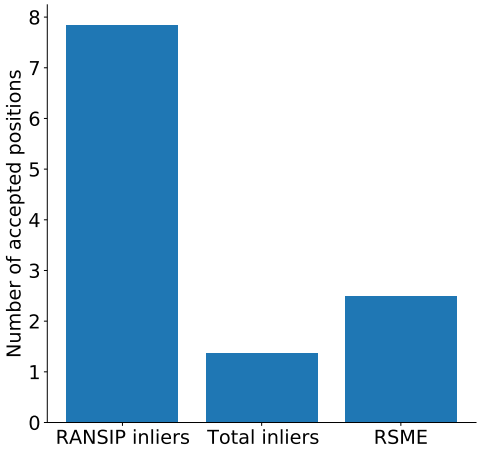
$$\sin(\theta_i) = \frac{\|n_{t_i} \times n_{s_i}\|}{\|n_{t_i}\|\|n_{s_i}\|},$$

where $x \times y$ is the outer product between vectors x and y. Because this is a hard rejection criterion, there is still a closed-form solution for (4.1), so that our optimization problem can be easily solved with any state-of-the-art implementation of ICP. Note that both (4.1) and (4.2) keep their original formulation, with the difference that some pairs will be excluded from (4.1), upon application of the rejection policy given by (4.6). Normals are computed with the Open3D library [132], by taking adjacent points and calculating their principal directions, through covariance analysis.

We then run ICP multiple times, with different initial positions for the template. The translation is taken as the difference between the center of mass of both shapes, but the rotation matrix is picked randomly. We take the result with the highest number of inliers, where inliers are counted according to (4.6). One will wonder whether a more elaborated strategy would benefit the search and reduce the number of iterations, as in usual RANSAC methods. However, the overlap ratio between our shapes is extremely low and the probability of picking only inliers which correspond would be very low. Therefore, it pays-off to avoid the complex and long computations for point selection, by doing several ICP iterations, which are considerably cheap in terms of computational time, since we keep the closed-form. This approach also has the advantage that the initial position of the shapes is irrelevant to the outcome.



(a) Correlation between metrics and true distance error.



(b) Number of iterations with distance error smaller than 15mm out of the top 10 values per shape.

Figure 4.4: Different criteria to evaluate the final position of the template shape. For this experiment, ICP is run for a fixed number of times, with random initialization on the rotation. Average of results for 50 different shapes in the *Simulated Dataset*. **RANSIP inliers** are the inliers respecting both distance and normals alignment criteria, while **Total inliers** only refers to the former; **RMSE** is the root mean squared error.

In Figure 4.4 we show a motivating example for our choice of outlier cost and why a more trivial one is not suitable for this kind of data. For this setting, we run ICP several times, with only a maximum distance criterion for inliers and the random initialization as previously described. The results (i.e., the final transformation of the template) are then compared to the true (completed) target. This evidences

why a straightforward application of methods with the L2-norm standard criterion are not applicable.

Having established a good correlation between our metric and the correct template positioning, we need to determine how many iterations are necessary for such configuration to appear, given our random initialization. In resemblance to RANSAC methods, we want to find the parameters leading to the maximum number of points conforming to our model, but we can not apply the same stopping criterion. In that case, the number of necessary iterations is computed according to the probability of selecting inliers given the current estimate, i.e., best value achieved. However, we do not subsample the shape at any point, so the reasoning becomes meaningless. Instead, we compute the cumulative average of inliers found up to the current iteration, as well as the cumulative standard deviation (**std**). We assume that, upon a stabilization of those statistics, we will be able to identify the best configurations as the ones lying above a certain ratio of **std** with respect to the average. Figure 4.5 provides a visual explanation of the stopping criterion and the pseudo-code for RANSIP is found in Algorithm 4.4.

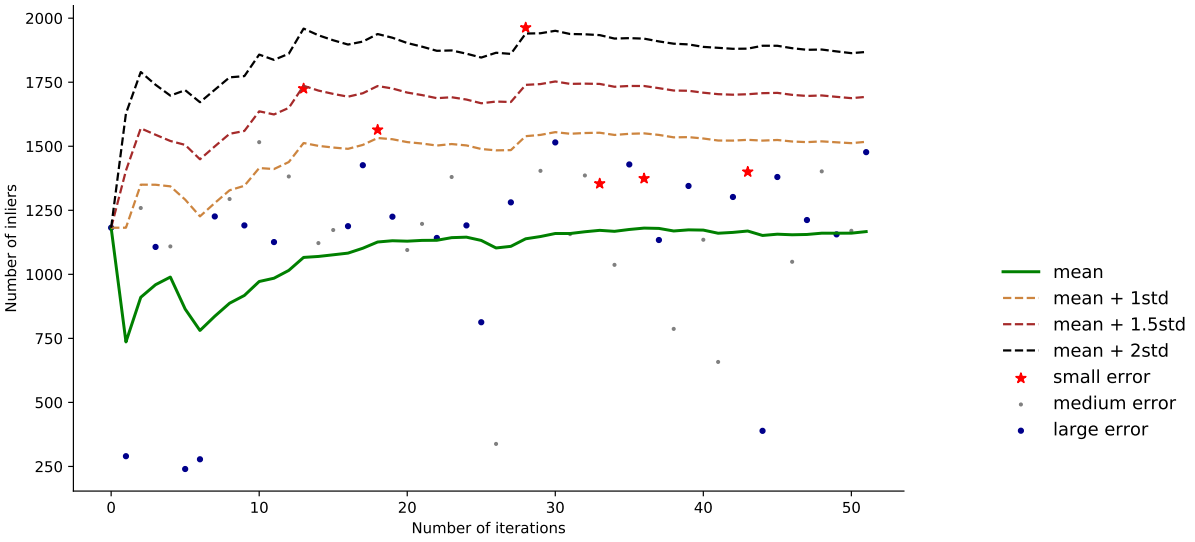


Figure 4.5: RANSIP stopping criteria for an average shape. We represent the evolution of the cumulative average of inliers found (solid green line), to which different ratios of the correspondent **std** were added (dashed lines). The scattered points represent the number of inliers in each individual iteration, classified according to thresholds on the true distance error. We note that although some transformations with small error are not identified by our metric, the ones above a certain level of **std** (in this case 1.5) always depict good positioning of the template. In this case, the number of iterations corresponds to the minimum, as the inlier metrics had already stabilized. Naturally, more favourable and unfavourable scenarios exist where the classifications are, respectively, more and less separable.

4.5 Registration results

4.5.1 Initial registration

In this section, we apply different state-of-the-art registration methods to the *Simulated Dataset*. The goal is to roughly register the samples, ideally keeping all the points belonging to the ear and identifying the outliers. We choose methods from different areas, both rigid and non-rigid, to understand which ones

Algorithm 1 RANSIP algorithm

Input: $S, T, d_{\max}$

Compute normals for n_S for target shape
Compute translation $p_{CM} = S_{CM} - T_{CM}$
iter = 0, stop = False
while iter $<$ iter$_{MAX}$ or stop == False **do**
    Get random rotation matrix Γ_R
    Compute transformed template $\tilde{T}$ with p_{CM} and Γ_R
    Compute normals for n_T for current template
    Perform ICP with cost function in (4.6)
    Compute current number of inliers N_I
    Update cumulative inliers mean and variance: $\bar{N_I}$ and var(N_I)
    if $N_I > N_I^{\text{best}}$ **then**
        $N_I^{\text{best}} = N_I$
    end if
    if iter $>$ iter$_{MIN}$ and $N_I^{\text{best}} > \bar{N_I} + \varepsilon \text{var}(N_I)$ **then**
        stop = True
    end if
end while

are more suited to overcome our data problems. Two fundamental methods are ICP [104] for the rigid registration and CPD [85] for the non-rigid counterpart. These methods are considered state-of-the-art but not particularly suited to the data problems we find in this situation. Therefore, we further consider more recent variations for both cases: NICP[115] (the non-rigid version of ICP) and BCPD [126] (the Bayesian formulation of CPD). Note that BCPD incudes both rigid and non-rigid explicit transformations. Given the susceptibility of ICP-like method to local optima, we also test two randomized methods: the SDRSAC [114] and our proposed approach RANSIP.

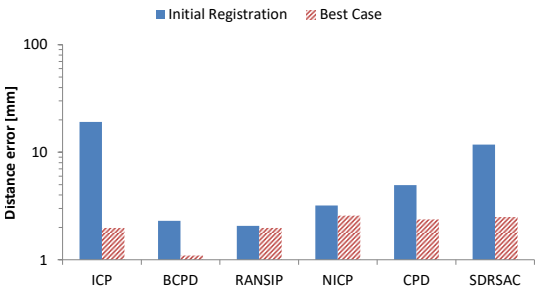
The registration methods take as input the template (i.e., the mean shape of the *Ear Dataset*) and a target shape from the *Simulated Dataset*, providing the deformed template that best fits the target. CPD, BCDP and SDRSAC explicitly provide the correspondence for each template and, consequently, the classification of outliers. For the remaining methods, correspondence is retrieved by taking the closest point, within a maximum distance threshold. For each of the algorithms and scenarios, an extensive study was performed in order to find the optimal parameters, including the distance threshold. The results presented bellow correspond to the parameter combination leading to the best performance in each step.

Furthermore, we compare the results with a so-called *best case scenario*. This consists of registering the original *Ear Dataset* without missing data, outliers or noise, with the template appropriately positioned with respect to the shapes in the dataset. Note that under this setting, there should be a one-to-one correspondence for every point and the global optimum should be close to the initial position.

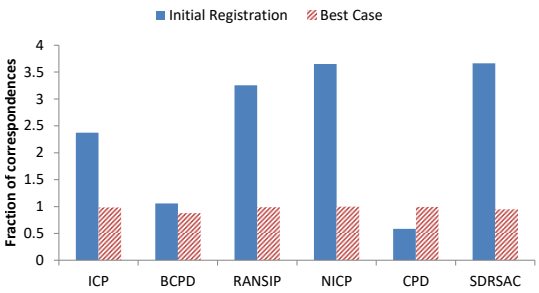
Figure 4.6 provides the general metrics for this setup, comparing all the methods both for the *real* and *best case* scenarios. Figure 4.7 depicts the outlier and missing data metrics only for the real case, as they are not relevant otherwise.

ICP and NICP

It is immediately evident that ICP does not cope well with the real case scenario. This is natural, since ICP can easily be trapped in local optima and the prevalence of data problems creates more local

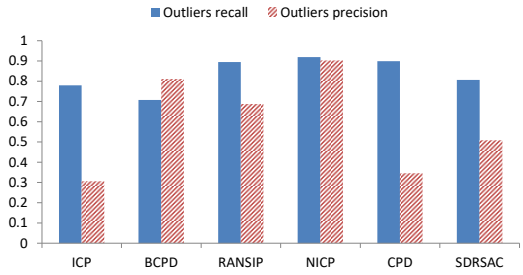


(a) Distance error

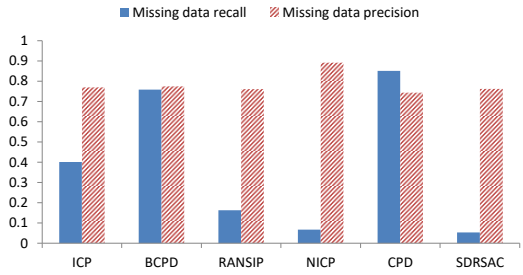


(b) Correspondences

Figure 4.6: Distance error for the initial registration with each of the different methods, for the realistic and best case scenarios.



(a) Outlier metrics



(b) Missing data metrics

Figure 4.7: Precision and recall for outliers and missing data, for each method considered in step 1. In general, missing precision is high and similar for all methods, meaning that not a lot of inliers are being discarded as missing. However, several methods, have a low recall meaning that they are not correctly identifying missing regions and are establishing matches for points of the template that do not have it. Outliers recall is generally high, meaning that most outlier points are detected but often many others are incorrectly considered (low precision).

optima. That is, with ICP the template easily falls into positions far away from the ground truth.

However, for the best case scenario, where the template is in the correct initial position and there is a one-to-one correspondence we expect ICP to perform well, and that is indeed the case. In fact, the distance error for this case (around 2mm) is an important value as it tells us that, limited to rigid deformations, this is the best error we can expect to find and it is due to the shape differences between the template and targets.

Regarding the correspondences fraction, it is noted that the original version of ICP (used here) allows for a point of the target to be associated with more than one point of the template. Consequently, in the best case, where both point sets have the same number of points, the fraction is around 1. In the real case, if a point in the template does not have a correspondence on the target (missing data), it will be associated to another point as long as is stays within the defined threshold, even if the target point was already associated to another template point. This means we will find more correspondences than we should and explains the value over 1. The latter conclusion is further evidenced by the missing data recall of ICP in Figure 4.7. The low value for this metrics expresses the restricted ability in detecting missing points.

NICP performs considerably better than ICP, but it should be noted that it receives a mesh instead of a point cloud, thus making it an unfair comparison with respect to other methods. The distance error is only surpassed by BCPD and RANSIP, although the fraction of correspondences is considerably away from the desired value 1, which is naturally related to the low missing data precision. Nonetheless, the outliers precision and recall achieve the best values out of all the considered methods.

SDRSAC and RANSIP

Both SDRSAC and RANSIP are composed of several runs of ICP. Therefore, without any data problems they should have a similar performance as the latter, as indeed is evidenced by Figure 4.6 for the modified version. The original one has a higher distance error and this could be explained by the fact that we are introducing a random initialization when the original one was already good. Meaning that the cost considered by the SDRSAC is not the most adequate for our data, as it does not detect the best transformation as such.

Looking at the real case, the fact that ICP is run several times helps in avoiding the local optima trapping, leading to an increased performance for the initial registration, as shown by the distance error of RANSIP. The correspondence fraction is still considerably above 1, for the reasons already stated with respect to ICP. When it comes to SDRSAC the performance is still better than the ICP, but again it is evident that the cost is not the most adequate for this situation.

Regarding Figure 4.7, we notice that both outlier metrics and missing data precision are close to 1 (being this the desired value), while missing data recall is very low. This is again related to the correspondence fraction as explained for ICP. However, since the goal of this first step is to remove the majority of outliers, this low value does not prevent the use of the method.

CPD and BCPD

For the best case scenario, CPD performs slightly worse than rigid methods such as ICP or SDRSAC. In this case, the non-rigid deformation does not seem to help in the registration, which could be due to the fact that the dataset does not have enough variability. That is, as the data was sampled from the model, the shapes are considerably similar amongst themselves and rigid deformations to the template are enough for a small distance error.

CPD does not handle well the real data. Despite the low value of distance error when compared with ICP, we see that this is achieved due to the low fraction of correspondences found (Figure 4.6). This is in accordance with the low outlier precision in Figure 4.7, showing that a high number of non-outliers is being identified as outliers.

Unlike CPD, BCPD seems to improve the distance error when compared to the non-rigid methods, which would disprove the justification given for the CPD increase. However, the correspondence fraction for BCPD is lower than for the others (around 90%), which means it is wrongly discarding points as outliers, thus obtaining a lower distance error.

When it comes to the real case, BCPD performs well, achieving a distance error slightly above the best case of the rigid methods and even below the best case of CPD and NICP. This does not come at

the cost of the correspondence fraction as this value stays close to 1 (Figure 4.7).

4.5.2 Registration refinement

The output from the first step allows us to remove the majority of the outliers and only keep the non-outliers identified by each method. Given the previous results, we take only the BCPD and RANSIP outputs. Despite having superior outlier metrics, NICP has a considerably higher distance error, entailing that the initial position of the template would be further from the optimal one. Upon (effective) outlier removal, all the methods are expected to increase their performance, when compared to the initial scenario (Section 4.5.1).

Therefore, we test each of the previous approaches for both BCPD and RANSIP outputs. The template consists of the mean shape from the *Ear Dataset* rotated and translated, according to the transformations estimated in Section 4.5.1, for each shape of the dataset. The transformed template is then registered to the shapes in the *Simulated Dataset*, where outliers have been removed, according to the classification retrieved in Section 4.5.1. Figure 4.8 shows the distance error and the fraction of correspondences and Figure 4.9 the outlier and missing data metrics, respectively.

Regarding ICP, we note that even with the removal of the outliers, the remaining data problems still prevent the method from achieving an acceptable performance. CPD, on the other hand, benefits from this removal and is able to express lower values for the distance error, although it has a correspondence fraction considerably above 1.

For almost any metric and any subsequent method, we notice that RANSIP in the first step produces the best performance. Consequently, this is selected as the method for the first step of the pipeline.

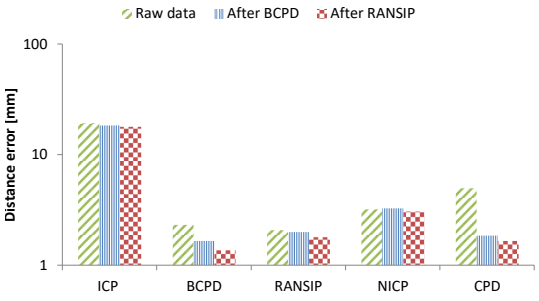
Regarding the choice for the second step, we are particularly interested in achieving good performance on the missing data metrics, as it will allow us to correctly identify where to do shape completion in the subsequent step. Looking at Figure 4.9, we conclude that the precision is similar across the different strategies, meaning that non-missing points are correctly identified as such, so we are not losing important information. However, when it comes to the ability to detect the true missing points, expressed by the recall, most approaches present low values, with the exception of ICP and BCPD.

The comparison between CPD and BCP is pertinent. After the outlier removal, CPD manages to achieve competitive values of distance error with respect to BCPD. However, it is not able to cope well with missing data and ends up attributing correspondences to template points that should not have one. This increases the fraction of correspondences and decreases the recall for missing data.

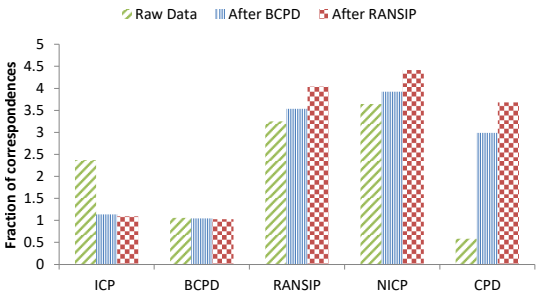
4.6 Shape completion

4.6.1 Approaches

Given the amount of missing data in our scenario, after the registration step a large percentage of points in the template will have no correspondence in the target. In order to relate the complete shape of the ear with the head, we would like to first complete the missing data. Of course the completion can

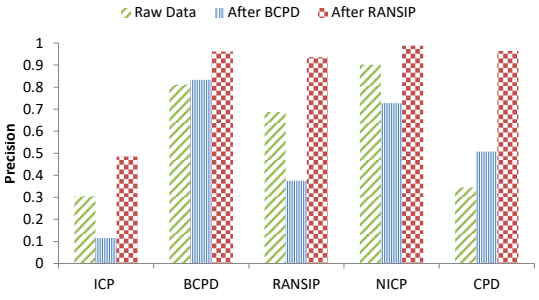


(a) Distance error

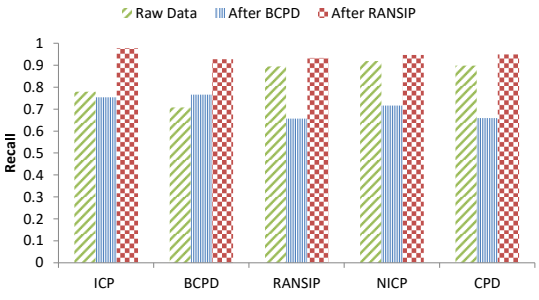


(b) Correspondences

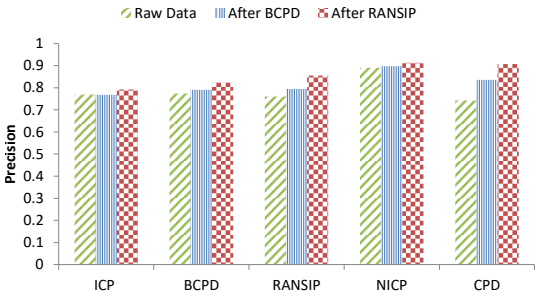
Figure 4.8: Distance error and fraction of correspondences metrics for the registration refinement with each of the different methods. Registration refinement refers to application of each method upon outlier removal and template re-positioning with either BCPD (blue) ro RANSIP (red). A comparison with registration from the original raw data (green) is also provided as baseline.



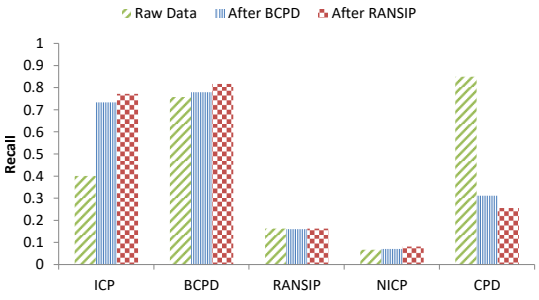
(a) Outliers precision



(b) Outliers recall



(c) Missing data precision



(d) Missing data recall

Figure 4.9: Outlier and missing data metrics, for the registration refinement with each of the different methods. Registration refinement refers to application of each method upon outlier removal and template re-positioning with either BCPD (blue) ro RANSIP (red). A comparison with registration from the original raw data (green) is also provided as baseline.

be helped by some information on the shape but this would require a previous model, taking us back to the chicken-and-egg problem mentioned before. Therefore, we consider three alternatives which entail different levels of prior information on the shape.

Deformed template from the registration method

Under this option, we take the transformed template from the registration method and see how well it resembles the original target. The main advantage is that this uses the generic transformation model defined by the registration method and does not require any previous knowledge on the shape. On the other hand, this lack of information is expected to lead to shapes less similar to ears.

Probabilistic PCA

An option to counteract this problem is to use a previous model of the ear shape. After we establish correspondence for the existing points, it is possible to predict the remaining ones with Probabilistic PCA (PPCA) [133] as suggested in [134]. This of course assumes a previous model for which we will use the PCA model obtained from the *Ear Dataset*. The PPCA model follows a similar structure to Equation (2.7), but it has a probabilistic derivation, so that

$$\mathbf{s}(\alpha) = W\alpha + \bar{\mathbf{s}} + \epsilon, \tag{4.7}$$

where the additive noise follows $\epsilon \sim \mathcal{N}(0, \sigma^2 I_{N_T D})$. The model components are found with maximum likelihood estimation and the results are very similar to the ones described in Section 2.2.1 (see [134] for further details).

After registration we can split the target point set into $S = [S_{corr}, S_{out}]$, where S_{corr} are the points with correspondence in the template and S_{out} the remaining ones. In the same way we split the template into $T = [T_{corr}, T_{miss}]$ and the model matrices into $W = [W_{corr}, W_{miss}]$ and $\bar{\mathbf{s}} = [\bar{\mathbf{s}}_{corr}, \bar{\mathbf{s}}_{miss}]$. Naturally, the dimensions of the matrices with the same subscript are coherent, in particular, $T_{corr} \in \mathbb{R}^{C \times D}$, $W_{corr} \in \mathbb{R}^{CN_T \times P}$ (recall P to be the number of components kept from the PCA model) and $\mu_{corr} \in \mathbb{R}^{CN_T}$. The goal is to obtain the full shape S from S_{corr}.

We can express the distribution of the target shape points as

$$\begin{aligned} p(S) &= p\big(S_{miss}, S_{corr}\big) \\ &= \mathcal{N}\left(\begin{bmatrix} \bar{\mathbf{s}}_{miss} \\ \bar{\mathbf{s}}_{corr} \end{bmatrix}, \begin{bmatrix} W_{miss}W_{miss}^T & W_{miss}W_{corr}^T \\ W_{corr}W_{miss}^T & W_{corr}W_{corr}^T + \sigma I_{N_T D} \end{bmatrix} \right) \end{aligned}$$

and obtain the distribution of α given the known points S_{corr} as

$$p(\alpha | S_{corr}) = \mathcal{N}\big(M^{-1}W_{corr}^T \sigma{-}2(S_{corr} - \mu_{corr}), M^{-1}\big), \tag{4.8}$$

where $M = \sigma{-}2W_{corr}^T W_{corr} + I_P$. The prediction for the complete target shape is obtained by inserting the mean from (4.8) into (4.7).

Gaussian Process framework

The disadvantage of the previous option is that the initial PCA model limits the shape space. It would be ideal to include some shape information but still allow for some freedom in shape matching.

We will apply GPR to predict the full shape, as introduced in Section 3.4.4.

We use the Empirical kernel $K_{PDM}(x, x')$ introduced in (3.16) and computed with the *Ear Dataset*, together with the SE kernel $K_{SE}(x, x')$ introduced in (3.17). We build the additive kernel

$$K_{model}(x, x') = K_{PDM}(x, x') + K_{SE}(x, x'), \tag{4.9}$$

which will enforce both the data learnt deformations and the coherent movement of nearby points, thus increasing the flexibility of the initial sample kernel.

4.6.2 Results

For this step, we take approximately 10% of the *Simulated Dataset* as shapes to be predicted and use the remaining shapes in the *Ear Dataset* to build the models. Points identified in Section 4.5.2 as having a correspondence are taken as observations, while the remaining points are missing data to be predicted according to the approaches described above. The necessary parameters for each method were tuned by grid search.

Figure 4.10 shows the reconstruction error between the complete targets after prediction and the original shapes from the *Ear Dataset*, for each of the three alternatives. We consider as baseline the mean shape used for the reconstruction of each shape. We can see that the only option with lower error with respect to the baseline is the PPCA. The fact is that given the way the dataset was built, the samples do not present an increased variability. This also explains why the GP approach performs so poorly, since we increased the variability of the model with the Gaussian kernel but we are predicting shapes which are very similar to our PCA model. However, this does not mean that in the real case the GP will perform worse, as we expect the ear from the *Head Dataset* to present more variability. For this reason, in the next section we apply the whole pipeline to the latter dataset.

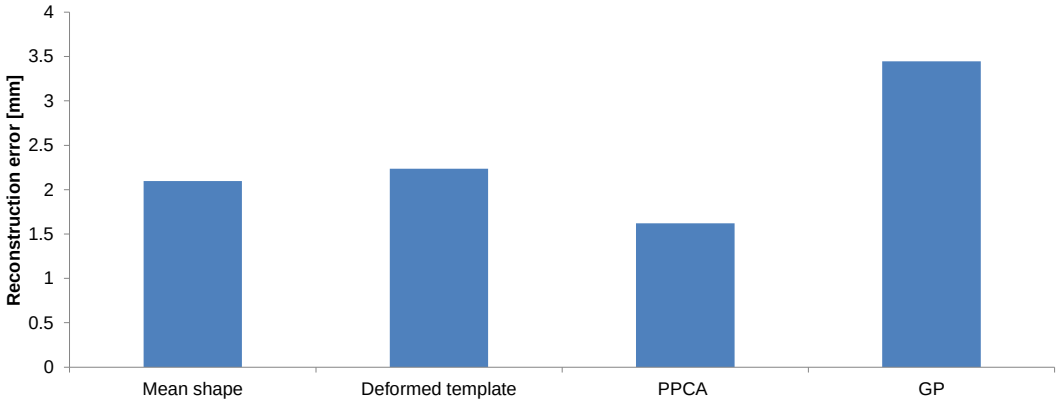
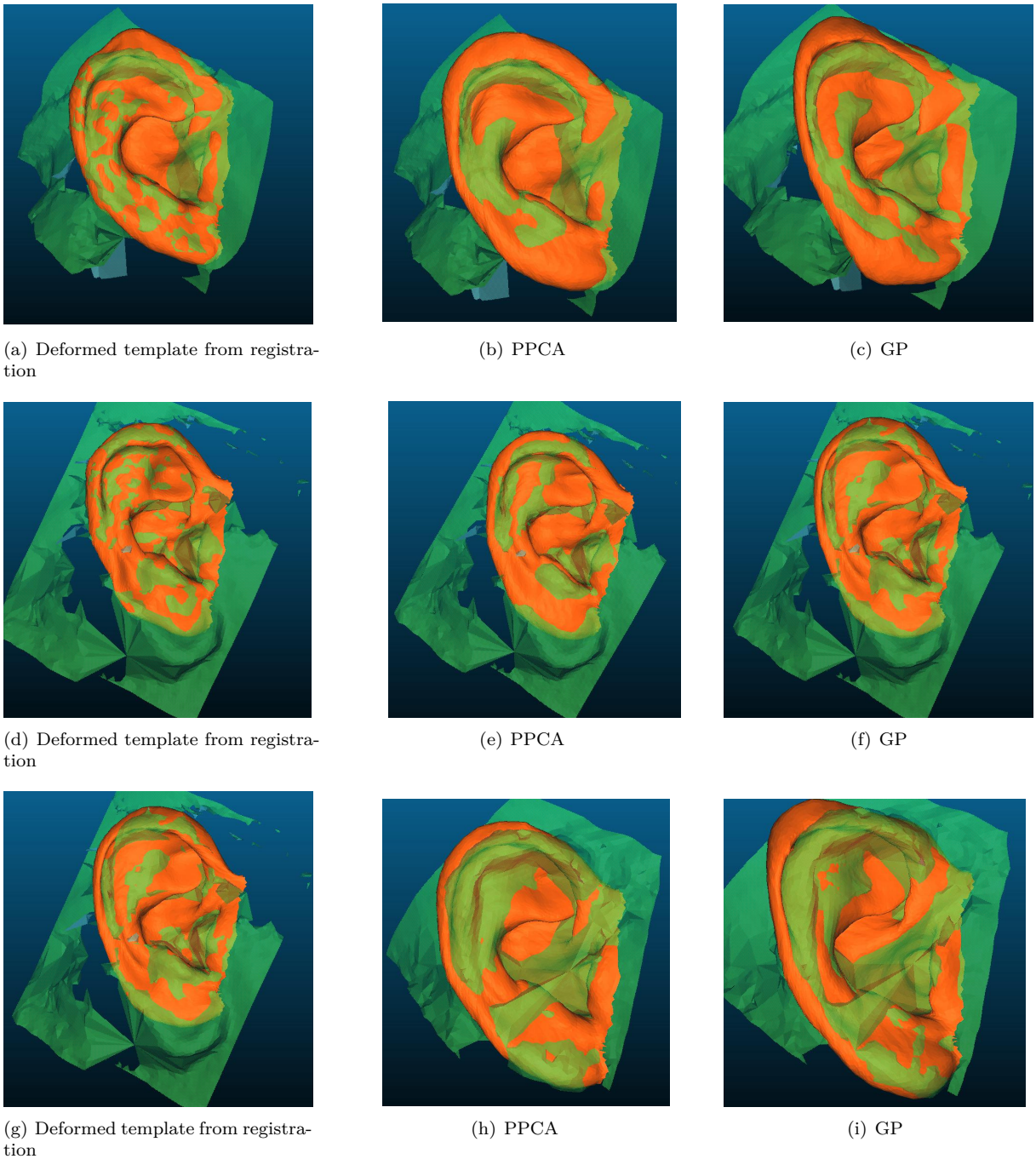


Figure 4.10: Comparison of the reconstruction error with the different shape completion options. The error with respect to the mean shape is used as baseline.

4.7 Full pipeline

Finally, we apply the complete pipeline to real ears from the *Head Dataset*. Evidently there is no ground truth, so it is not possible to obtain quantitative metrics. Therefore, we evaluate the results by empirical observation in Figure 4.11.



(a) Deformed template from registration (b) PPCA (c) GP

(d) Deformed template from registration (e) PPCA (f) GP

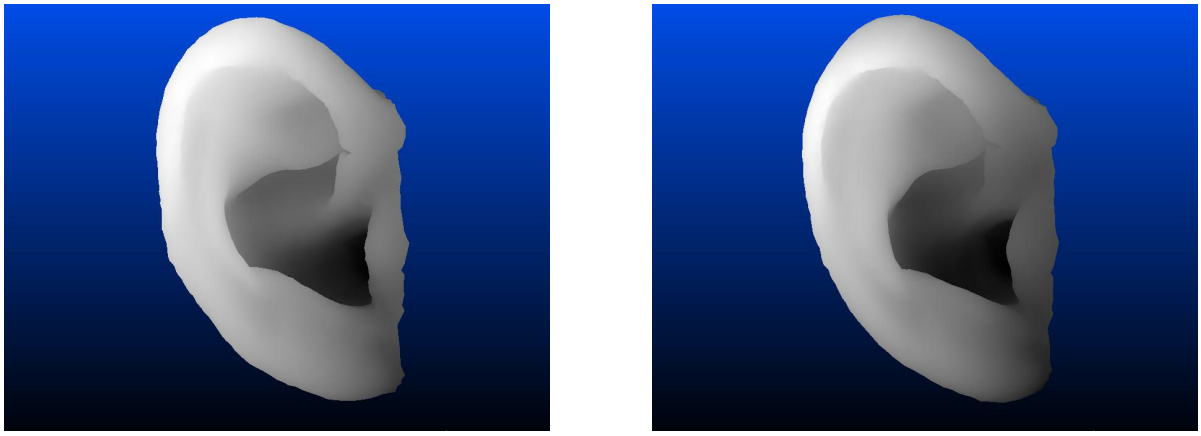
(g) Deformed template from registration (h) PPCA (i) GP

Figure 4.11: Example of application of the full pipeline to three real scans of the ear. Each rows depicts the results of a different scan with the shape completion by deformed template from the registration (left column), PPCA (middle column) and GP regression (right column). In each picture, the green shape corresponds to the original scan and the orange one to the shape predicted by the respective approach.

As expected, the PPCA approach guarantees to produce a smooth ear shape, but the variability in the model is not enough to correctly complete the shape when the latter is too different from the

dataset samples. The deformed template from the registration solves this problem in part but, since it has no further information on the shape, suffers too much deformation. The GP framework is a good compromise between the two previous options, as it includes a prior on shape, while the augmentation through the Gaussian kernel provides additional flexibility. Consequently, unlike the PPCA approach, it is able to reach deformations not previously seen on the training dataset, such as the detail presented in Figure 4.13.

However, there are still shortcomings on the final shape, as in some parts it is not adequately fitted to the target. This suggests that further work can be done in finding a more appropriate kernel or including further steps on either the registration or shape completion steps.



(a) PPCA (b) GP

Figure 4.12: Mean shape of *Reconstructed Dataset* with PPCA and GP. The visual differences are not extensive and found mostly on the inner region of the ear. The overall shape is also more elongated in the PPCA result and presents a different curvature on top. Nonetheless, both shapes are reasonable examples of a human ear, as desired.

After this visual comparison, we analyse the full produced datasets for both GP and PPCA shape completion. The primary goal is to compare both approaches and confront them with the equivalent metrics on the *Ear Dataset*. We apply Generalized Procrustes Analysis (GPA) to all of the datasets in order to study only shape differences. After this, we compute the mean shape of each one (Figure 4.12) and the deformation each point suffers with respect to it, i.e., the Euclidean distance between the correspondent point of the mean shape and each sample of the dataset (Figure 4.14). We observe the average deformation over the full shape and over the same point on each shape of the dataset, represented in the boxplots of Figure 4.14. We also represent the latter over a 3D plot of the mean shape of the *Ear Dataset* (Figure 4.15), so that we can visualize where the deformations occur. Looking at the average shape deformation we see that all the three datasets present similar values, with the Ear one having a slightly increased average. On the other hand, the GP one has fewer shapes similar to the mean one, while the PPCA one has a similar distribution with respect to the Ear, except it has less different shapes in general.

At this point, we may ask whether the overall shape differences in the *Reconstructed Dataset* are caused by a particular region which is not correctly fitted or registered, since we have observed that this is still a limitation of the pipeline. Observing the average deformation over point (Figure 4.14 on the

right), we observe that this is not the case given that even the regions with lower average deformation have an acceptable value and are not being compensated by regions of increased one. Looking at Figure 4.15, while the PPCA and *Ear Dataset* have a small region of very high deformation, this is not present in the GP, indicating that possibly we need to allow for more deformation on the GP prior. On the other hand, we see that the latter has higher large deformations on a large region of the bottom of the ear. This is a positive outcome, as one of the main obstacles we have found was the fitting of such area.

(a) PPCA (b) GP

Figure 4.13: Detail of shape completion step with PPCA and GP for the same sample. The green shapes correspond to the original scan and the orange ones to the shape predicted by each approach. Looking at the small protuberance on the bottom of the ear, it is clear that the PPCA model does not have enough variability to properly deform the template. On the other hand, the augmented kernel of the GP is able to provide the necessary variability, when given the same correspondences as the PPCA model.

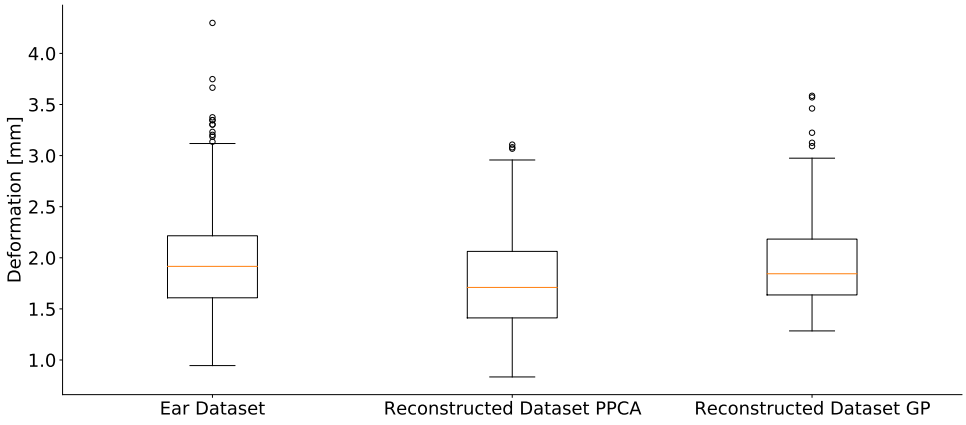
We also look at the most different shapes of the *Reconstructed Datasets* on Figure 4.12, that is the shape with higher average deformation with respect to the mean shape. As expected, while the PPCA has no unexpected deformations, the GP has a problematic upper area. This is observed rather frequently and calls for an improvement on the registration procedure with GP, since the problem lies in the fitting of the posterior part of the ear to the skin, incorrectly identified as inlier. This also means that part of the variability observed in the region on Figure 4.15 is due to the incorrect registration and will decrease. However, an improved process will allow for more deformation of the template, leading to complete fitting to the target shape, which is still a limitation, as seen in Figure 4.11.

4.8 Concluding remarks

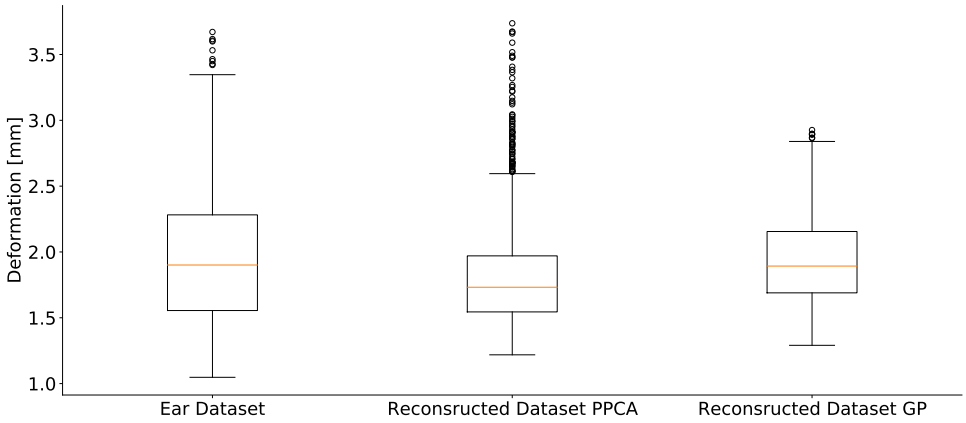
Taking the previous results into account, we propose the final pipeline composed by an initial registration with RANSIP, a registration refinement with BCPD and GP regression for shape completion, with an empirical and SE kernel. The proposed approach is able to complete the ears from the *Head Dataset*, although it still lacks some accuracy in matching the final shape. We expect the pipeline to extend to other datasets with similar limitations, and although not considered in our work, it would be a natural next step.

A particular point of remark is the confirmation of our initial assumption that GP constitutes a promising framework to deal with the extreme challenges of the ear point clouds. Despite still showing some limitations, this serves as a starting point to test further improvements on either the registration/fitting or modelling, given that the GP shape formulation is highly flexible. The registration could without a doubt benefit from a more integrated approach, which we will explore over the next chapter.

Furthermore, from the output *Head Dataset* with the complete ears, it is also possible to obtain a



(a) Mean deformation averaged over shape



(b) Mean deformation averaged over point

Figure 4.14: Box plot of the deformation of each point with respect to the mean shape, averaged over the shape or the same point of all shapes. The former metric expresses how different are on average the reconstructed shapes, while the latter indicates how this variation is spread across the shape points. From the first plot it is evident that PPCA leads to a dataset with reduced variation compared to the GP. In particular, the shapes with inferior average deformation lie near the first quartile of the PPCA ones. The second plot assures that the variability comes from different points of the shape and not a particular region of increased variation. However, it should be noted that PPCA reaches a level of deformation never attained by GP, albeit for a small set of points.

relationship between the head shape and the ear region. This can help in building an improved ear model, which in turn may increase the performance of the current pipeline. Additionally, the complete dataset may be used to build new informed kernels in a guided manner, able to identify and capture the main characteristics of the ear shape, without explicitly using the empirical kernel.

It is also worth noting that the comparison amongst several kinds of registration methods with the large variety of considered metrics offers an interesting insight on their behaviour and capabilities. Further studies could be conducted under a similar setting, involving an increased pool of methods and more diverse datasets.

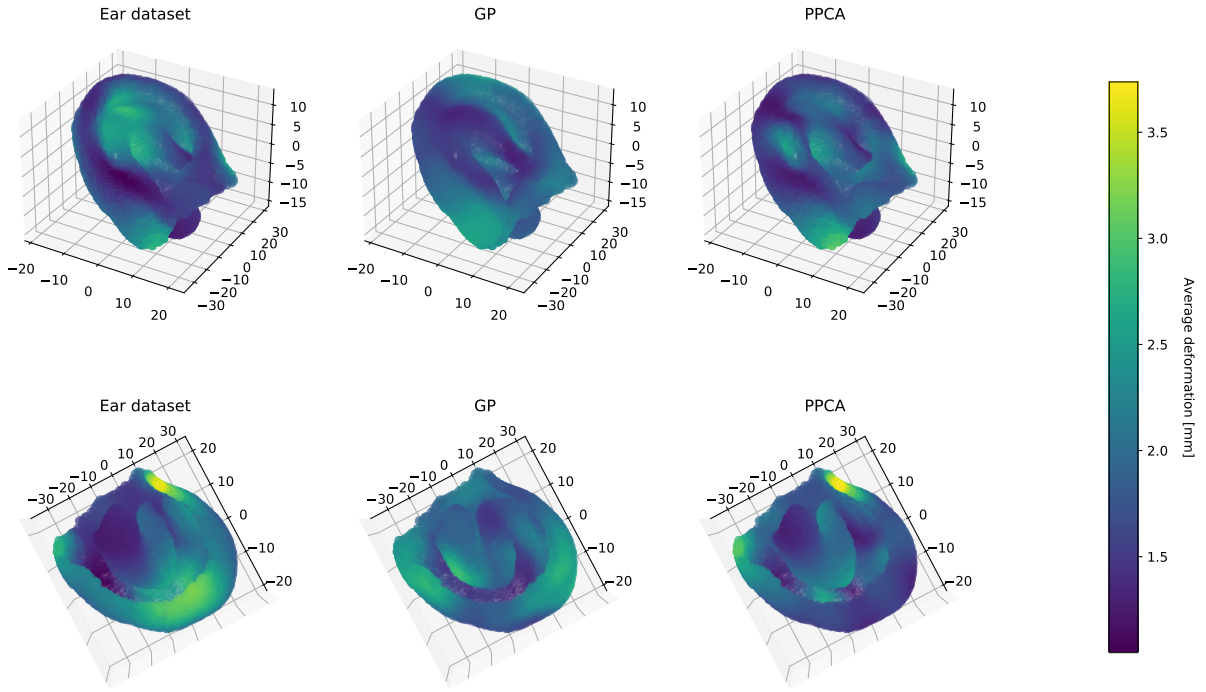
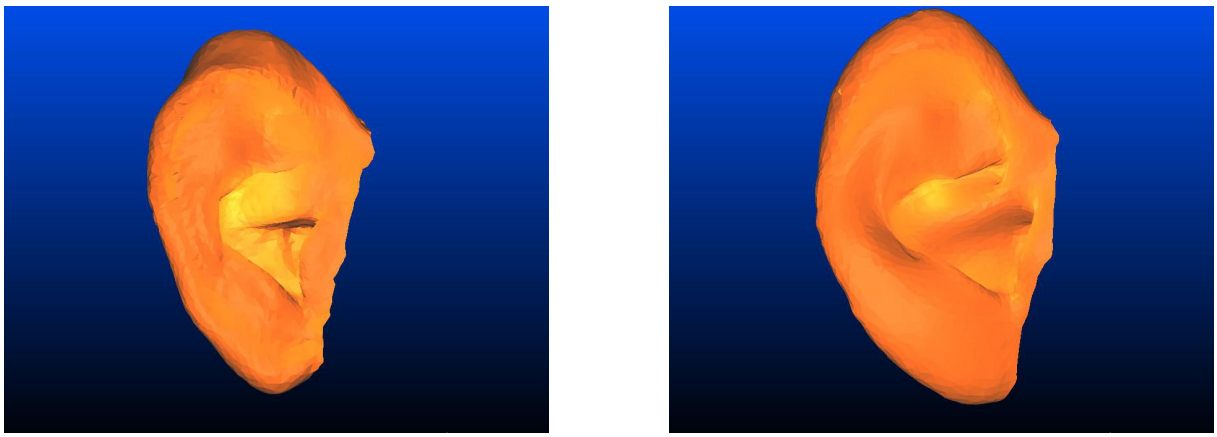


Figure 4.15: Average deformation of each point (with respect to the mean shape) over all the shapes, for each dataset: Ear (on the left), Reconstructed with GP (middle) and Reconstructed with PPCA (right). The first row provides an upper view of the ear, while the second provides a bottom one, in order to facilitate visualization. Given that the PPCA approach is built on top of the *Ear Dataset* it is natural that both present a similar pattern of deformation. In contrast, GP does not have the same regions with the extremes of low and high deformation, having more moderate values over the entire shape.



(a) GP

(b) PPCA

Figure 4.16: Most different shape in the *Reconstructed Dataset* with GP (left) and PPCA (right). From each dataset we compute the average Euclidean distance with respect to its mean shape, in order to access how reasonable are the completed shapes. While the PPCA result has no unexpected deformations, the GP sample reflects the problems encountered in the registration on the top part. Nonetheless, it is still a reasonable result and does not exclude GP as strong candidate for shape completion, taking into account that the model parameters were not tuned beforehand.

Chapter 5

Probabilistic Registration/Fitting

5.1 Introduction

The work developed in the previous chapter has evidenced the advantage and potential of working within the Gaussian Process (GP) framework to overcome some of the challenges involved in ear modelling. However, a detailed fitting is still not achieved for part of the samples. Besides, the proposed approach is still fairly empirical and does not take full advantage of the unified setting that GP offers. In this chapter, we assume that shapes were subjected to an initial rough alignment, for instance with RANSIP, and that most of the effects of translations and rotations were previously removed. However, unlike the previous proposal, we do not remove any outliers, to avoid removing possibly helpful information. We then wish to merge the two last steps of the pipeline to solve Problem (2.10) in a fully unified manner.

The advantages of this global approach are multiple. While previously each of the obstacles was tackled individually (we first handled the outliers, then missing data and finally used previous shape information for completion), here they are considered simultaneously. Although this translates to a more complex problem, we can benefit from more information at the same time — for instance, a data prior can improve registration by restricting deformations based on a observed samples of the shape. Besides, a more generic approach is expected to handle different datasets and scenarios with more ease, thus increasing the potential of applications. Finally, and perhaps more importantly, the same underlying model is used for all tasks. Not only is this a more principled approach, but it also means that different models can be tested in a more structured way.

The idea of bringing together data-driven and motion coherence approaches is evidently not new. As hinted in Chapter 2 it is often viewed as data-driven registration (on one side) or model fitting with regularization (on the other). Most of the approaches build on an ICP-like formulation, where the Euclidean transformations are replaced with the Principal Component Analysis (PCA) model previously obtained [47, 135, 136]. Then additional terms are added to constrain the deformations and this is where the variations are usually found. Motion coherence can be enforced exactly as in Nonrigid Iterative Closest Point (NICP) [47, 135] or by simply penalizing the distance between neighbouring deformations [136]. Other terms are often added such as an orthogonality term to prevent large distortions in [136].

Initial rigid alignment is usually assumed, as we shall do in this chapter. An exception is [47], where rigid transformations are applied to the template and estimated, culminating in an ICP problem with additional model parameters α to be estimated. The method has shown particular improvements on fitting the mouth region of faces, but we do not except extreme amounts of non-rigid deformation, given a reasonable initial alignment.

Another common aspect is the closest-point approach for the correspondence step. An exception is [137], which is perhaps the closest work to the one here presented. Unlike previous methods, they use a probabilistic assignment, formulating the problem in the Coherent Point Drift (CPD) framework. However, their proposal presents two considerable differences with respect to standard CPD. On one hand, they use the empirical covariance instead of the Gram matrix of the Squared Exponential (SE) kernel, thus including prior data knowledge. On the other, they consider anisotropic Gaussian Mixture Models (GMMs) oriented according to surface normals. Their goal is slightly different than ours since they consider shape reconstruction from sparse points. I.e., large missing regions are still expected but dense regions do not exist, while we face regions of high density and others completely devoid of point data.

We are now in the position to understand the points made in favour of the GP formulation. Kernels are (as introduced in Chapter 3) extremely powerful tools and the few limitations on a valid kernel function mean that most relationships between data points can be formulated through a kernel. In fact the combination of a PCA model with motion coherence can be easily encoded with an Empirical and SE kernel, as was done in Chapter 4. Consequently, it is very likely that most of the terms used for the shape fitting/registration process can be transformed into a single covariance function with appropriate parameters and operations between the kernels.

By expressing all the constraints in a single structure — the kernel — we achieve two fundamental goals. On one hand, this allows us to apply different constraints within the same registration/fitting method, and more clearly separate the effect of the two components. On the other, it means that new constraints can be immediately included without changing the method at all, so that the optimization can be independent from the chosen model.

In the prior work devloped for GP shape modelling, we can find proposals for the fitting/registration. One simple option is to tackle the problem with a non-rigid application of the ICP [104], where the transformation is obtained through Gaussian Process Regression (GPR). This means that to each point in the template we attribute the closest target point, based on their Euclidean distance. These correspondences are then taken as observations and GPR is used to compute deformations for the entire shape (the mean of the posterior is the template used in the next iteration). So, it resembles [47, 135, 136] except that the model is built with a GP prior. Our approach also relates to this method in the formulation of the problem, but not in the way the correspondences are retrieved. For ear shapes, given the large regions of missing data and the highly non-rigid deformations, the closest point approach often leads to undesirable results [12].

In another proposed approach in [98], the authors formulate the registration problem/model fitting for both surfaces and images. Here, we focus on the surface formulation. First, the authors do a

low-rank approximation, obtaining a parametric approximation of the original kernel. The problem is then posed as a Maximum a Posteriori (MAP) estimation problem, where the likelihood expresses some distance measure between the target and template shapes, and the prior is given by the GP. The authors chose the mean squared Euclidean distance from the reference to the closest target point and solve the problem with an L-BFGS optimizer [138]. Therefore, the restriction of hard assignment when choosing correspondences is maintained in this approach, leading to similar problems as faced in the previous one.

By modelling registration as a GP multi-annotator problem, we will show that we can perform probabilistic registration within the GP framework, which allows us to benefit both from an expressive prior through the kernels and from the nice properties of probabilistic assignment, particularly when dealing with outliers and noise.

5.1.1 Our method

From the previous introduction we conclude that proposals for registration within the GP framework target hard assignment and assume a one-to-one correspondence between the target and template, thus motivating their extension with a probabilistic assignment. Several proposals for probabilistic methods exist in the registration area (see Section 4.3.2 for a more detailed review), but they do not consider extensive prior knowledge on the shape, other than imposing some regularization of the deformations, which usually can be translated to a Gaussian kernel. This observation motivated us to develop Shape Fitting Gaussian Process (SFGP): a probabilistic shape fitting/registration method within the GP framework, where we can benefit both from a complex kernel prior and a soft assignment in the correspondences.

Given the previous considerations we propose a new method, that bridges the gap between these two formulations. Our main contributions are:

- **Shape registration/model fitting as a multi-annotator GPR.** We show how the problem of registration with soft assignment can be understood within the GP framework as a multi-annotator Gaussian Process Regression (Section 5.2).

- **Parallel between probabilistic registration and our method, SFGP.** We provide a parallel between BCPD and our algorithm, under a few assumptions, which allows us to take advantage of the positive aspects of the probabilistic setting (Section 5.3). We further show how their differences lead to a good performance in the presence of extensive missing data (Section 5.4).

- **Application to a difficult registration problem – 3D human ears registration.** We show that our method is suitable for the registration of 3D point clouds with highly non-rigid deformations, high occurrence of missing data and outliers, by performing simulations with 3D point sets of human ears. The results show improvement with respect to state-of-the-art registration methods and current GP registration proposals (Section 5.4).

We start by providing background on the probabilistic approach followed here, Bayesian Coherent Point Drift (BCPD) (Section 5.1.2). Then, we formulate the registration/fitting in the GP framework where the soft assignment is expressed with multi-annotators in Section 5.2. This still leaves an open

question on how to set one of the variables. We solve this by finding a parallel with BCPD, i.e., by formulating our own problem within VBI setting and then retrieving the corresponding values (Section 5.3). The proposed method SFGP is tested on simulated 2D and 3D datasets (Section 5.4) and on real ears from the *Head Dataset* (Section 5.5).

5.1.2 Probabilistic shape registration

We will cover the Bayesian formulation of the probabilistic registration as described in BCPD [126]. Our focus is mostly on the assumptions employed and how they are encoded, to allow us to confront our approach with [126] in later stages of the work, as they are shown to be closely related. In the same way as CPD, the template points (T) are taken as centroids of GMMs and target points (S) as data generated by the centroids. Further, a point s_j can be an outlier with probability ω, in which case it is generated from an outlier probability distribution $p_{out}(s_j)$. If s_j is not an outlier, then it corresponds to a point t_i with probability α_i (membership probability).

In [126] the authors consider explicit similarity transformations and non-rigid ones, such that a point i of the transformed template is given as

$$\mathcal{T}(t_i) = \beta\Gamma(t_i + \delta_i) + p, \tag{5.1}$$

where t_i is the original template point, $\beta \in \mathbb{R}^+$ is a scale factor, $\Gamma \in SO(D)$ is a rotation matrix, $p \in \mathbb{R}^D$ is a translation vector and $\delta_i \in \mathbb{R}^D$ is a displacement vector for non-rigid transformations. According to a GMM, after model selection, the generation of a target point s_j follows a multivariate normal distribution with mean $\mathcal{T}(t_i)$ – the transformed template point – and covariance matrix $\varsigma^2 I_D$, so the probability distribution to generate s_j starting from t_i is

$$\phi_{ij}(s_j; \mathcal{T}(t_i), \varsigma^2) = \frac{1}{(\varsigma\sqrt{2\pi})^D} \exp\left\{ -\frac{\|s_j - \mathcal{T}(t_i)\|^2}{2\varsigma^2} \right\}. \tag{5.2}$$

In order to explicitly introduce correspondences, two additional variables are added: $c \in \{0,1\}^{N_S}$, an indicator variable that takes value of 1 for c_j if point s_j is an outlier, and $e \in \{1, \ldots, N_T\}^{N_S}$, where $e_j = i$ if the j-th target point corresponds to the i-th template point. Taking the assumptions regarding outliers into account, we have the joint distribution for (s_j, e_j, c_j) as

$$p(s_j, e_j, c_j | T, \delta, \varsigma^2, \alpha) = \{w p_{out}(s_j)\}^{1-c_j} \left\{ (1-w) \prod_{i=1}^{N_T} (\alpha_i \phi_{ij})^{\gamma_i(e_j)} \right\}^{c_j}, \tag{5.3}$$

where γ_i is an indicator function, taking a value of 1 if $e_j = i$ and 0 otherwise, and α_i is the probability that $e_j = i$, with $\sum_{i=1}^{N_T} \alpha_i = 1$. The authors take $p(\alpha)$ as a Dirichlet distribution and set a prior on the deformations as

$$p(\delta|T) = \phi(\delta; 0, \lambda^{-1} G \otimes I_D),$$

where $G = (g_{ii'}) \in \mathbb{R}^{N_T \times N_T}$, with $g_{ii'} = k(i, i')$, and λ is a positive constant.

Finally, the full joint distribution is given as

$$p(S,T,\theta) \propto p(\delta|T)p(\alpha)\prod_{j=1}^{N_S} p(s_j,e_j,c_j|T,\delta,\varsigma^2,\alpha,\rho), \tag{5.4}$$

where $\theta = (v,\varsigma^2,\alpha,\rho,c,e)$ and $\rho = (\beta,\Gamma,p)$.

5.2 Multi-annotators formulation

In this section we shall formulate the GP shape fitting model as a multi-annotator GPR problem. To do so, we will first introduce the GPR with multiple annotators for a generic case (Section 5.2.1) and then apply it to our context (Section 5.2.2).

5.2.1 GPR with multiple annotators

In learning from data it is often seen that there are no given exact labels for the data (e.g., the attribution of a label is subjective and so there is no exact ground truth). A common approach is to obtain a collection of labels for each data point, provided by different annotators, with different levels of confidence [139, 140]. The multiple labels should then be weighted according to defined criteria, in order to retrieve a ground truth for each data point. This allows, for instance, to relate annotators to difference degrees of confidence on the provided labels. A proposal for GPR with multiple annotators providing labels is given in [141].

We shall provide a generic introduction on this topic and as such the standard machine learning notation used throughout most of Chapter 3is recovered, hoping to make the connection between the two settings easier. This will then be reformulated with the GP shape modelling notation for a smooth transition.

Instead of $\mathcal{D} = \{(x_i,y_i)|i = 1,\dots,N\}$, in a multi-annotator setting we consider a collection $\mathcal{D}_m = \{(x_i^m,y_i^m)|i = 1,...,N_m\} = (X_m,y_m)$ of the m-th annotator. We further define $X = \cup_{m=1}^M X_m$ and $Y = \{y_1,\dots,y_M\}$ and N is the number of unique inputs in X (each annotator does not necessarily provide a label for all unique input points). For each unique data point $x_i \in X$, there will be at most M labels and we define the set of existing labels for point i as $\mathcal{L}_i$, containing the indices of all annotators who provided a label.

So, for each unique $x_i \in X$, we compute the associated variance $\hat{\sigma}_i^2$ and label $\hat{y}_i$ as a combination of all annotators in the following way

$$\frac{1}{\hat{\sigma}_i^2} = \sum_{m\in\mathcal{L}_i}\frac{1}{\sigma_m^2}, \quad \hat{y}_i = \hat{\sigma}_i^2\sum_{m\in\mathcal{L}_i}\frac{y_i^m}{\sigma_m^2}, \tag{5.5}$$

where σ_m^2 is the variance associated to the m-th annotator.

Assuming annotators provide the labels independently from each other then

$$
\begin{aligned}
p(Y|f) &= \prod_{i=1}^{N} \prod_{m \in \mathcal{L}_i} \mathcal{N}\big(y_i^m | f_i, \sigma_m^2\big) \propto \exp\left\{ -\frac{1}{2} \sum_{i=1}^{N} \sum_{m \in \mathcal{L}_i} \frac{(y_i^m - f_i)^2}{\sigma_m^2} \right\} \\
&= \exp\left\{ -\frac{1}{2} \sum_{i=1}^{N} \frac{(\hat{y}_i - f_i)^2}{\hat{\sigma}_i^2} - \frac{1}{2} \sum_{i=1}^{N} \sum_{m \in \mathcal{L}_i} \frac{(y_i^m)^2}{\sigma_m^2} + \frac{1}{2} \sum_i \frac{\hat{y}_i^2}{\hat{\sigma}_i^2} \right\} \\
&= \exp\left\{ -\frac{1}{2} \sum_{i=1}^{N} \frac{(\hat{y}_i - f_i)^2}{\hat{\sigma}_i^2} + B \right\},
\end{aligned}
$$

where B is independent of f. So the posterior $p(f|Y)$ follows

$$
p(f|Y) \propto p(f)p(Y|f) \propto \exp\left\{ -\frac{1}{2} f^T K^{-1} f - \frac{1}{2} (\hat{Y} - f)^T \hat{\Sigma}^{-1} (\hat{Y} - f) \right\}
$$

which naturally corresponds to a Gaussian distribution $\mathcal{N}(\mu_p, \Sigma_p)$, with mean and covariance

$$
\begin{aligned}
\mu_p &= (K^{-1} + \hat{\Sigma}^{-1})^{-1} \hat{\Sigma}^{-1} \hat{Y} \\
\Sigma_p &= (K^{-1} + \hat{\Sigma}^{-1})^{-1},
\end{aligned}
$$

where $\hat{\Sigma} = \text{diag}(\hat{\sigma}_1^2, ..., \hat{\sigma}_N^2)$ and $\hat{Y} = [\hat{y}_1, \ldots, \hat{y}_N]^T$, where $\hat{\sigma}_i^2$ and $\hat{y}_i$ are given by (5.5). Therefore, the predictive equations can be obtained as

$$
\begin{aligned}
\bar{f}_* &= K_{X^*X} (K_{XX} + \hat{\Sigma})^{-1} \hat{Y} \\
\text{cov}(f_*) &= K_{X^*X^*} - K_{X^*X} (K_{XX} + \hat{\Sigma})^{-1} K_{XX^*},
\end{aligned}
\tag{5.6}
$$

where the kernel matrices are defined in (3.4). We can see that these equations differ from (3.4) only on the noise covariance matrix $\hat{\Sigma}$ and label vector $\hat{Y}$, otherwise exhibiting the same structure.

5.2.2 Shape modelling

We shall now formulate the GP shape model fitting, expanding the formulation introduced in Chapter 2. Recall that the template points t_i correspond to the points x_i, the observed deformations $\delta(t_i)$ correspond to the observations y_i and the true deformations $\mu(t_i)$ to the function $f(x_i)$, the GP. Furthermore, we can get noisy observations of the deformations $\delta(t) = u(t) + \epsilon$, where $\epsilon \sim \mathcal{N}(0, \sigma_n^2)$.

Modelling missing data and outliers In [80] the authors assume a one-to-one correspondence between target and template exists. Therefore, they considered the assumption in Section 3.4.1 that it is possible to register T to S, such that for each t_i there exist a point s_j with the same semantic meaning. However, this is often not the case for real data and, in particular, not the case for the ears in the *Head Dataset*.

Therefore, we consider that there are both missing data and outliers in our target shapes and so there are points of the template without correspondence in the target and vice-versa. We formulate this assumption by splitting the template T into two subsets $\left[T_{corr}, T_{miss}\right]$ (as introduced in Section 3.4.4),

where the former set contains the points with correspondence and the latter the ones with missing data. Recall that $T_{\text{corr}} \in \mathbb{R}^{C\times D}$ with with elements t_{Ci} and $T_{\text{miss}} \in \mathbb{R}^{(N_T-C)\times D}$, with elements t_{Mi}. Both sets are disjoint and their union leads to the full template.

We further split S into points with correspondence in T, S_{corr}, and outliers, S_{out}. However, S_{corr} may have repeated points, since multiple points in the template may be matched with the same point in the target. Therefore, we first define the deformations for all observed template points as

$$\delta = \begin{bmatrix} \delta_{C1} \\ \vdots \\ \delta_{CC} \end{bmatrix} = \begin{bmatrix} s_{t_{C1}} - t_{C1} \\ \vdots \\ s_{t_{CC}} - t_{CC} \end{bmatrix}, \tag{5.7}$$

where $s_{t_{Ci}}$ is the target point corresponding to the template point t_{Ci}. Then, $S_{\text{corr}} = [s_{t_{C1}}, \ldots, s_{t_{CC}}]^T \in \mathbb{R}^{C\times D}$ and S_{out} is the set of all points in S not contained in S_{corr} in no particular order.

Fitting problem with hard assignment Under this model, and because of the gaussianity of the distributions, for which the mode and mean coincide, the shape fitting and registration problem is usually formulated as a MAP problem [11], that is

$$\max_u p(u|T, S).$$

In particular, for our case, it depends on the observed points and deformations, so we have

$$\max_u p(u|T_{corr}, \delta).$$

However, we do not know the correspondence beforehand, so T_{corr}, δ are not known and they also depend on u, leading to

$$\operatorname*{argmax}_u p(u|T_{corr}(u), \delta(u)).$$

In an ICP-like approach we split our problem into two sub-problems, where we keep u fixed in the inner maximization and T_{corr},δ fixed in the outer. So, our final formulation is

$$\max_u \left\{ \max_{T_{corr},\delta} p(u|T_{corr}(u), \delta(u)) \right\}. \tag{5.8}$$

In the outer optimization we are computing the transformation given the current correspondences and in the inner cycle we are computing the correspondences given the current deformation of the template. The strategy proposed in [11] is to retrieve the correspondences through a closest point strategy and then compute the transformations with standard GPR equations, as this corresponds to the solution of the outer problem, given T_{corr},δ. In the following section, we formulate a soft-assignment to replace the closest point strategy.

Soft assignment Introducing a soft-assignment means to attribute different possible target points to each template point, each with a different probability of correspondence. We can interpret this as each template point having several different possible labels (deformations) given by annotators with different levels of confidence (probability of correspondence). Therefore, one point t_i will have N_S labels corresponding to each point in the target that could be a possible match. A schematic representation is found in Figure 5.1.

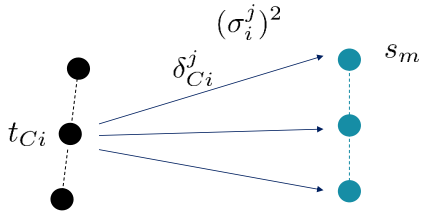


Figure 5.1: Interpretation of multi-annotators in shape modelling context. Each template point t_i gets a label, i.e., a deformation from each target point s_m. To each label, a level of confidence is attributed through a variance $(\sigma_i^j)^2$. In general, this variance should have an inverse relationship to the probability of s_m being the corresponding point to t_i.

We are now in the setting of GP with multi-annotators, introduced in Section 5.2.1. As we have seen, the predictive equations for GPR remain the same with a different noise covariance matrix and observations vector, as these are built from the multiple labels, according to (5.5). In the context of our problem, the equations become

$$\begin{aligned} \mu_p &= K_{T_{corr}T}^T (K_{T_{corr}T_{corr}} + \tilde{D}_{\sigma_n^2})^{-1} \delta \\ D_{\sigma_P^2} &= K_{TT} - K_{T_{corr}T}^T (K_{T_{corr}T_{corr}} + \tilde{D}_{\sigma_n^2})^{-1} K_{T_{corr}T}, \end{aligned} \tag{5.9}$$

where $D_{\sigma_n^2} = \text{diag}(\sigma_{C1}^2, \ldots, \sigma_{CC}^2)$, $\tilde{D}_{\sigma_n^2} = D_{\sigma_n^2} \otimes I_D$ and $\delta = \left[\delta_{C1}^T, \ldots, \delta_{CC}^T\right]^T \in \mathbb{R}^{N_T D}$. The components of $\tilde{D}_{\sigma_n^2}$ and δ are obtained through (5.5) as

$$\frac{1}{\sigma_{Ci}^2} = \sum_{j \in \mathcal{C}_i} \frac{1}{(\sigma_{Ci}^j)^2}, \qquad \delta_{Ci} = \sigma_{Ci}^2 \sum_{j \in \mathcal{C}_i} \frac{\delta_{Ci}^j}{(\sigma_{Ci}^j)^2}, \tag{5.10}$$

where $\delta_{Ci}^j = s_j - t_{Ci}$. The variance $(\sigma_{Ci}^j)^2$, corresponding to the annotator's confidence in the original formulation, represents here the probability of correspondence between point C_i and s_j. Note that while the variance of annotator j was the same across the data points i in the original formulation, here we consider that annotator j of point C_i is not necessarily the same as annotator j of point C_k, i.e. there can be at most $N_T \times N_S$ annotators. In order to complete our algorithm, two points need to be addressed: how to define the correspondence set T_{corr} and how to compute $(\sigma_{Ci}^j)^2$. The computation of this value will be detailed in the next section.

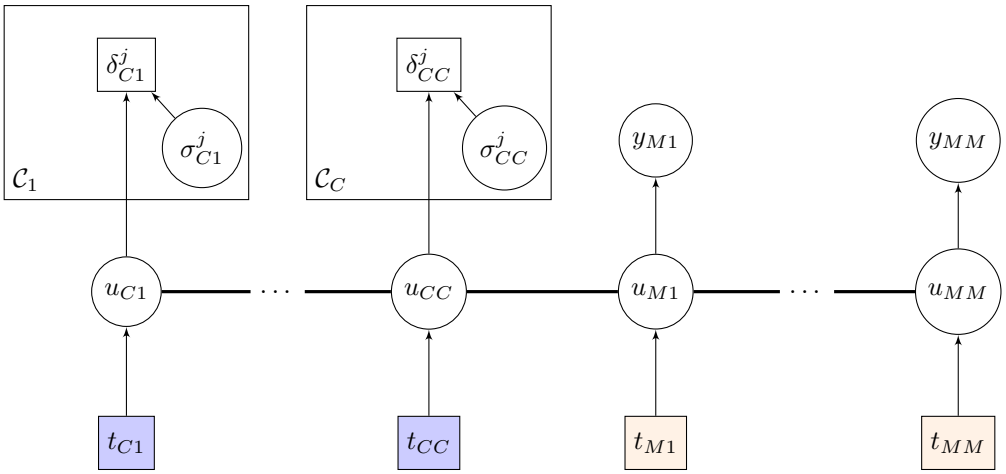


Figure 5.2: Graphical model for the GP with multiple annotators in the context of shape modelling. The model follows the notation proposed in [86] for GP graphical models, where the horizontal bold line represents a set of fully connected nodes. Squared nodes represent observed variables and circles are latent ones. Templates points with correspondence are identified with blue, while missing points are identified with orange to facilitate visual recognition. The plates represent repetition of contained variables and, in this case, refer to the multiple annotations j of a given template point.

5.3 Variance

In this section we will detail how $(\sigma^j_{Ci})^2$ is to be computed. For this purpose, we need to reformulate our problem under the usual probabilistic approach, from which we chose the BCPD formulation with Variational Bayesian Inference (VBI), for being a state-of-the-art approach with several advantages with respect to CPD. We provide background on VBI (Section 5.3.1), followed by the derivation of the equations for our problem (Section 5.3.2) and finally we establish the parallel with our initial formulation (Section 5.3.3).

5.3.1 Variational Bayesian Inference

Variational inference [142, 143] is a family of techniques used to approximate a posterior distribution when the exact computation is not possible. This could happen because the latent space has a very high dimensionality or the computations involved are not computationally tractable, for instance. They are a type of deterministic approximations, contrasting with stochastic ones, of which the Markov chain Monte Carlo is likely the most popular. In our case, we take a joint distribution similar to (5.4) and wish to find the posterior $p(\theta|S,T)$. Through this section we shall introduce VBI applied to this posterior, thus keeping the same notation.

The idea behind VBI is to use a tractable distribution $q(\theta)$ to approximate $p(\theta|S,T)$ and then minimize the Kullback–Leibler (KL) divergence (Appendix A.4) between them

$$\begin{aligned} q^*(\theta) &= \underset{q(\theta \in Q)}{\operatorname{argmin}} \quad KL(q(\theta) \;||\; p(\theta|S,T)) \\ &= \underset{q(\theta \in Q)}{\operatorname{argmin}} \, \mathbb{E}[\log q(\theta)] - \mathbb{E}[\log p(\theta,T,S)] + \log p(S,T), \end{aligned}$$

where Q is a predefined set of distribution families to which q belongs. However, as the $\log p(S,T)$ may not be computable, the evidence lower bound (ELBO) is maximized instead

$$ELBO(q) = \mathbb{E}[\log p(\theta, T, S)] - \mathbb{E}[\log q(\theta)].$$

The ELBO is equivalent to the negative KL divergence up to a constant, and therefore maximizing the former is equivalent to minimizing the latter. One possible family from which we can choose q is the mean-field variational family, where the latent variables are mutually independent and governed by different factors. It is the one we pick here, so $q(\theta) = \prod_{i=1}^{M} q_i(\theta_i)$, where $q_i(\theta_i)$ is the distribution for the variable θ_i. In particular, we will consider

$$q(\theta) = q_1(\delta)q_2(c,e)q_{31}(\varsigma_1^2)...q_{3i}(\varsigma_i^2)...q_{3N_T}(\varsigma_{N_T}^2).$$

A standard method to maximize the ELBO, and the one followed here, is the Coordinate Ascent Mean-field VI. If we fix all other q_j, then we know that the optimal q_i is

$$q_i(\theta_i)^* \propto \exp\{\mathbb{E}_{-i}[\log p(\theta_i|\theta_{-i}, S, T)]\} \propto \exp\{\mathbb{E}_{-i}[\log p(\theta_i, \theta_{-i}, S, T)]\}, \tag{5.11}$$

where $\mathbb{E}_{-i}[\log p(\theta, S, T)]$ is the expectation of the joint probability with respect to the remaining $q_{j\neq i}$ and θ_{-i} corresponds to all parameters in θ except θ_i. So, each q_i is updated iteratively by computing $\mathbb{E}_{-i}[\log p(\theta, S, T)]$, until convergence is reached. Note that at each step we will need the expected value of the remaining variables. Further, at each step we are updating the distribution parameters of q_i.

5.3.2 Derivation

We will briefly detail the assumptions and problem formulation used in this approach, by deriving equivalents of equation (5.1) through (5.4) according to our assumptions. This will lead to the final expression for the joint distribution in (5.15).

Transformation model We do not consider similarity transformations, so the transformation acting on the template is merely given by a displacement vector δ_i

$$\mathcal{T}(t_i) = t_i + \delta_i. \tag{5.12}$$

Gaussian mixture model According to the GMM we obtain a similar expression to (5.2)

$$\phi_{ij}(s_j; \bar{t}_i, \varsigma_i^2) = \frac{1}{(\varsigma_i\sqrt{2\pi})^D} \exp\Big(-\frac{\|s_j - \mathcal{T}(t_i)\|^2}{2\varsigma_i^2}\Big), \tag{5.13}$$

but we introduce an individual variance for each template point given as ς_i^2. As we will show later, this has a positive impact when dealing with large regions of missing data. Regarding the outliers, we follow the same assumptions as in BCPD, but for simplification we take $p_{out}(s_j) = 1/N_S$, as was previously taken in CPD. Besides, we take equal membership probabilities $\alpha_i = 1/N_T$, meaning that a point in the

target is expected to be associated to any point of the template with equal probability. This is also the assumption in CPD and is here taken for simplification on this first formulation of the framework. Thus, we obtain the joint distribution

$$p(s_j, e_j, c_j|T, \delta, \varsigma^2) = \left\{ \frac{w}{N_S} \right\}^{1-c_j} \left\{ \frac{(1-w)}{N_T} \prod_{i=1}^{N_T} (\phi_{ij})^{\gamma_i(e_j)} \right\}^{c_j}. \tag{5.14}$$

Prior distributions The prior on the deformations is expressed with the previously defined kernel K, but we take $\lambda = 1$, as we can usually include this parameter as part of the kernel estimation. We then have the prior given as $p(\delta|t) = \mathcal{N}(0, K)$.

Full joint distribution

Finally, the full joint distribution is given as

$$p(s, t, \theta) \propto p(\delta|t) \prod_{j=1}^{N_S} p(s_j, e_j, c_j|T, \delta, \varsigma^2), \tag{5.15}$$

where $\theta = (\delta, \varsigma^2, c, e)$ are the parameters to be estimated.

We present the updates for each component in Proposition 5.1, Proposition 5.2 and Proposition 5.3. The proofs follow the ones in [126, 142] and can be found in Appendix B.1. In general, the final equations exhibit a similar structure (to those in [126]) except that ς^2, taken as a scalar in [126], is replaced by the diagonal matrix D_{ς^2}.

For ease of notation, and in preparation for the subsequent equations, we define $p_{ij} = \mathcal{E}[c_j \gamma_i(e_j)]$ as the probability of correspondence between template point i and target point j, with the respective probability matrix $P = [p_{ij}]_{i,j=1}^{N_T, N_S} \in [0,1]^{N_T \times N_S}$. We further define $\nu_i = \sum_{j=1}^{N_S} p_{ij}$, representing the expected number of target points corresponding with t_i, as well as $\nu = P1_{N_S}$, the corresponding vector.

Proposition 5.1. *The update equations for the expected value and covariance of δ are*

$$\begin{aligned} \mathbb{E}[\delta] &= \Sigma \tilde{D}_\nu \tilde{D}_{\varsigma^2}^{-1} (\tilde{D}_\nu^{-1} \tilde{P} \boldsymbol{s} - \boldsymbol{t}) \\ \mathrm{cov}(\delta) &= \Sigma = (K^{-1} + \tilde{D}_\nu \tilde{D}_{\varsigma^2}^{-1})^{-1}, \end{aligned} \tag{5.16}$$

where $D_{\varsigma^2} = \mathrm{diag}(\varsigma_1^2, \ldots, \varsigma_{N_T}^2)$ and $D_\nu = \mathrm{diag}(\nu_1, \ldots, \nu_{N_T})$.

Proposition 5.2. *The update for the correspondence probability is*

$$p_{ij} = \frac{(1-w)\langle \phi_{ij} \rangle}{\frac{N_T}{N_S} w + (1-w) \sum_{i'=1}^{N_T} \langle \phi_{i'j} \rangle}, \tag{5.17}$$

where $\langle \phi_{ij} \rangle = \phi_{ij}(s_j; \bar{t}_i, \varsigma_i^2) \exp\left\{ -\frac{\mathrm{tr}(\mathrm{cov}(\delta_i))}{2\varsigma_i^2} \right\}$, $\bar{t}_i = \mathbb{E}[\mathcal{T}_i] = t_i + \mathbb{E}[\delta_i]$, $\mathrm{cov}(\delta_i)$ is the submatrix of $\mathrm{cov}(\delta)$ defined in (5.1) related to the δ_i component and $\mathrm{tr}(\cdot)$ is the trace of the matrix.

Proposition 5.3. *The update for each variance term ς_i^2 is given as*

$$\varsigma_i^2 = \frac{1}{d}\Big(\frac{[\tilde{P}\text{diag}(\boldsymbol{s})\boldsymbol{s}]_i - 2\bar{t}_i^T[\tilde{P}\boldsymbol{s}]_i}{\nu_i} + \|\bar{t}_i\|^2 + \text{tr}(\text{cov}(\delta_i))\Big). \tag{5.18}$$

5.3.3 Parallel

If we assume that there are no missing points, then it is possible to establish a parallel between the previous formulation and the GP framework. With this aim, we reformulate our expressions in order to obtain a similar structure to the ones in the previous section.

Proposition 5.4. *Considering no missing points, i.e. $T_{corr} = T$, and if the variance $(\sigma_{Ci}^j)^2$ in Equation* (5.10) *is taken as*

$$(\sigma_{Ci}^j)^2 = \frac{\varsigma_i^2}{p_{ij}}, \tag{5.19}$$

where p_{ij} is given by (5.17)*, then an equivalent exists between the update equations in Proposition 5.1, Proposition 5.2, Proposition 5.3 and the update equations for multi-annotator GPR in* (5.6)*.*

Proof. When $T_{corr} = T$, and since $K^T = K$, the posterior mean and covariance in (5.9) become

$$\mu_p = K(K + \tilde{D}_{\sigma_n^2})^{-1}\delta$$
$$D_{\sigma_P^2} = K - K(K + \tilde{D}_{\sigma_n^2})^{-1}K,$$

where $D_{\sigma_n^2} = \text{diag}(\sigma_1^2, \ldots, \sigma_{N_T}^2)$, with σ_i^2 and δ_i given by (5.10) and here restated without the notation for correspondences

$$\frac{1}{\sigma_i^2} = \sum_{j \in N_S} \frac{1}{(\sigma_i^j)^2}, \quad \delta_i = \sigma_i^2 \sum_{j \in N_S} \frac{\delta_i^j}{(\sigma_i^j)^2}.$$

Taking the variance as in (5.19), we can write the previous equations as

$$\frac{1}{\sigma_i^2} = \frac{P_i 1_{N_s}}{\varsigma_i^2} = \frac{\nu_i}{\varsigma_i^2}$$
$$\delta_i = \nu_i^{-1}\sum_j p_{ij}(s_j - t_i) = \nu_i^{-1}\tilde{P}_i s - \nu_i^{-1}\sum_j p_{ij}t_i = \nu_i^{-1}\tilde{P}_i s - t_i,$$

where P_i refers to the $i-th$ row of matrix P. Therefore, we have that $D_{\sigma_n^2} = D_{\varsigma^2}D_\nu^{-1}$ and $\delta = \tilde{D}_\nu^{-1}\tilde{P}s - t$. The posterior deformations in (5.3.3) can then be written as

$$\begin{aligned}\mu_p &= K(K + \tilde{D}_{\varsigma^2}\tilde{D}_\nu^{-1})^{-1}\delta \\ &= KK^{-1}\Big[1 + \tilde{D}_{\varsigma^2}\tilde{D}_\nu^{-1}K^{-1}\Big]^{-1}\delta \\ &= \Big[K^{-1} + \tilde{D}_\nu\tilde{D}_{\varsigma^2}^{-1}\Big]^{-1}\tilde{D}_\nu\tilde{D}_{\varsigma^2}^{-1}(\tilde{D}_\nu^{-1}\tilde{P}\mathbf{s} - t)\end{aligned}$$

and the covariance as

$$\begin{aligned} D_{\sigma_P^2} &= K - K(K + \tilde{D}_{\varsigma^2}\tilde{D}_{\nu}^{-1})^{-1}K \\ &= K - \left[K^{-1} + \tilde{D}_{\nu}\tilde{D}_{\varsigma^2}^{-1}\right]^{-1}\tilde{D}_{\nu}\tilde{D}_{\varsigma^2}^{-1}K \\ &= \left[K^{-1} + D_{\nu}D_{\varsigma^2}^{-1}\right]^{-1}\left[K(K^{-1} + \tilde{D}_{\nu}\tilde{D}_{\varsigma^2}^{-1}) - \tilde{D}_{\nu}\tilde{D}_{\varsigma^2}^{-1}K\right] \\ &= \left[K^{-1} + \tilde{D}_{\nu}\tilde{D}_{\varsigma^2}^{-1}\right]^{-1}, \end{aligned}$$

thus being equivalent to the expressions for $\mathbb{E}[\delta]$ and $\text{cov}(\delta)$ in (5.16). So if we follow the update for p_{ij} and ς_i^2 according to Proposition 5.2 and Proposition 5.3, we have a full parallel in the update steps. □

Although this is established for the case of no missing data only, we take (5.19) as a reasonable update for the annotators variance, together with the necessary updates for p_{ij} and ς_i^2.

5.3.4 Final algorithm

At this point, it is necessary to define a criterion for the classification of missing points and inliers. Given the probability matrix P, we apply a predefined threshold P_{MIN}, such that pairings with a lower value than P_{MIN} are identified as non-corresponding. So, for each point t_i, the considered correspondences to the target are $\mathcal{C}_i = \{j : s_j \in S, p_{ij} > P_{MIN}\}$. Then if a point t_i has no elements in $\mathcal{C}_i$, it is considered a missing point, meaning that $T_{miss} = \{t_i : t_i \in T, |\mathcal{C}_i| = 0\}$ and $T_{corr} = \{t_i : t_i \in T, |\mathcal{C}_i| > 0\}$.

The pseudo-code for our method is found in Algorithm 2 and Algorithm 3, where the former contains the main outer steps and the latter details the computation for the correspondence part.

Algorithm 2 SFGP

Input: $t, s, K, D_{\varsigma_0^2}, \omega, P_{MIN}$

1: $\bar{t} = t$, $D_{\sigma_P^2} = \mathbf{0}$
2: **while** some stopping criterion is not met **do**
3: $\quad T_{corr}, \hat{\delta}, D_{\sigma_n^2} = \text{get\_correspondences}(r, s, \bar{r}, D_{\sigma_P^2}, D_{\varsigma^2}, \omega, P_{MIN})$
4: $\quad \mu_p = K_{T_{corr}T}^T(K_{T_{corr}T_{corr}} + \tilde{D}_{\sigma_n^2})^{-1}\hat{\delta}$
5: $\quad D_{\sigma_P^2} = K_{TT} - K_{T_{corr}R}^T(K_{T_{corr}T_{corr}} + \tilde{D}_{\sigma_n^2})^{-1}K_{T_{corr}T}$
6: $\quad \bar{t} = t + \mu_p$
7: $\quad \varsigma_i^2 = \frac{1}{D}\left(\frac{[\tilde{P}\text{diag}(s)s]_i - 2\bar{r}_i^T[\tilde{P}s]_i}{\nu_i} + \|\bar{t}_i\|^2 + \text{Tr}(D_{\sigma_P^2})\right)$
8: **end while**

5.4 Analysis and results

In this section we present the results of experiments with both 2D (Section 5.4.2) and 3D data (Section 5.4.3), with the respective discussion. For each subsection, we first describe the datasets and settings, following with an analysis of the results. We start with a toy example to explain the behaviour of the algorithm (Section 5.4.1).

Algorithm 3 get_correspondences

Input: $\mathbf{t}, \mathbf{s}, \bar{t}, D_{\sigma_P^2}, D_{\varsigma^2}, \omega, P_{MIN}$

Output: $D_{\sigma_n^2} = \text{diag}(\sigma_{C1}^2, \ldots, \sigma_{CC}^2), \hat{\delta} = (\hat{\delta}_{C1}^T, \ldots, \hat{\delta}_{CC}^T)^T, T_{corr}$

1: **for** $(i, j) \leftarrow (1, 1)$ to (N_T, N_S) **do**

2: $\quad \phi_{ij}(s_j; \bar{r}_i, \varsigma^2) = \frac{1}{(\varsigma_i\sqrt{2\pi})^d} \exp\Big(- \frac{\|s_j - \bar{r}_i\|^2}{2\varsigma_i^2} \Big)$

3: $\quad \langle \phi_{ij} \rangle = \phi_{ij} \exp\Big\{ - \frac{1}{2\varsigma^2} \text{Tr}(\sigma_{P_i}^2 I_d) \Big\}$

4: $\quad p_{ij} = \frac{(1-w)\langle \phi_{ij} \rangle}{\frac{N_R}{N_S} w + (1-w) \sum_{i'=1}^{N_T} \langle \phi_{i'j} \rangle}$

5: $\quad (\sigma_i^j)^2 = \frac{\varsigma_i^2}{p_{ij}}$

6: **end for**

7: $\mathcal{C}_i = \{j : s_j \in S, p_{ij} > P_{MIN}\}$
$\quad T_{corr} = \{t_i : t_i \in T, |\mathcal{C}_i| > 0\}$

8: **for** $i \in T_C$ **do**

9: $\quad \frac{1}{\sigma_{Ci}^2} = \sum_{j \in \mathcal{C}_i} \frac{1}{(\sigma_{Ci}^j)^2}$

10: $\quad \hat{\delta}_{Ci} = \sigma_{Ci}^2 \sum_{j \in \mathcal{C}_i} \frac{\delta_{Ci}^j}{(\sigma_{Ci}^j)^2}$

11: **end for**

5.4.1 Toy example

We have identified the bottom part of the ear as a problematic area for registration, since usually there is a collapse from the back segment to the front. We developed a toy example with similar characteristics to provide some insight on the behaviour of our method. The template and target used are represented in Figure 5.3, together with the resulting deformed template at representative iterations. Figure 5.4 represents the deformations and correspondence variance for each template point, as provided to the GPR step (regardless of missing points).

The variance is initialized with the same value for every point in T, so the first iterations have almost an effect of rigid transformations, moving the template to the right, closer to the target. Eventually, the top part and a small segment at the bottom find correspondences with high confidence, meaning that their variance goes to low values. These parts then remain matched for the rest of the process while the points with larger variance deform and adjust to the target.

So, the general idea is that the parts with increased confidence will be attached first and the others will then slowly adjust, a behaviour we desire. Of course, a natural consequence is that if the initial attachment is incorrect a recovery will probably not happen, as the variance will remain low. Hence the need for an initial assumption on rough initialization.

5.4.2 2D data

Dataset

As 2D data we take the *Fish Dataset* [144], where the template is a 2D fish with 98 points. The target point sets are then generated by applying different kinds of alterations to the data. Non-rigid deformations are generated by warping the template points with a Gaussian radial basis function. The dataset has four other variations considering outliers, missing data, rotation and noise, all of them with

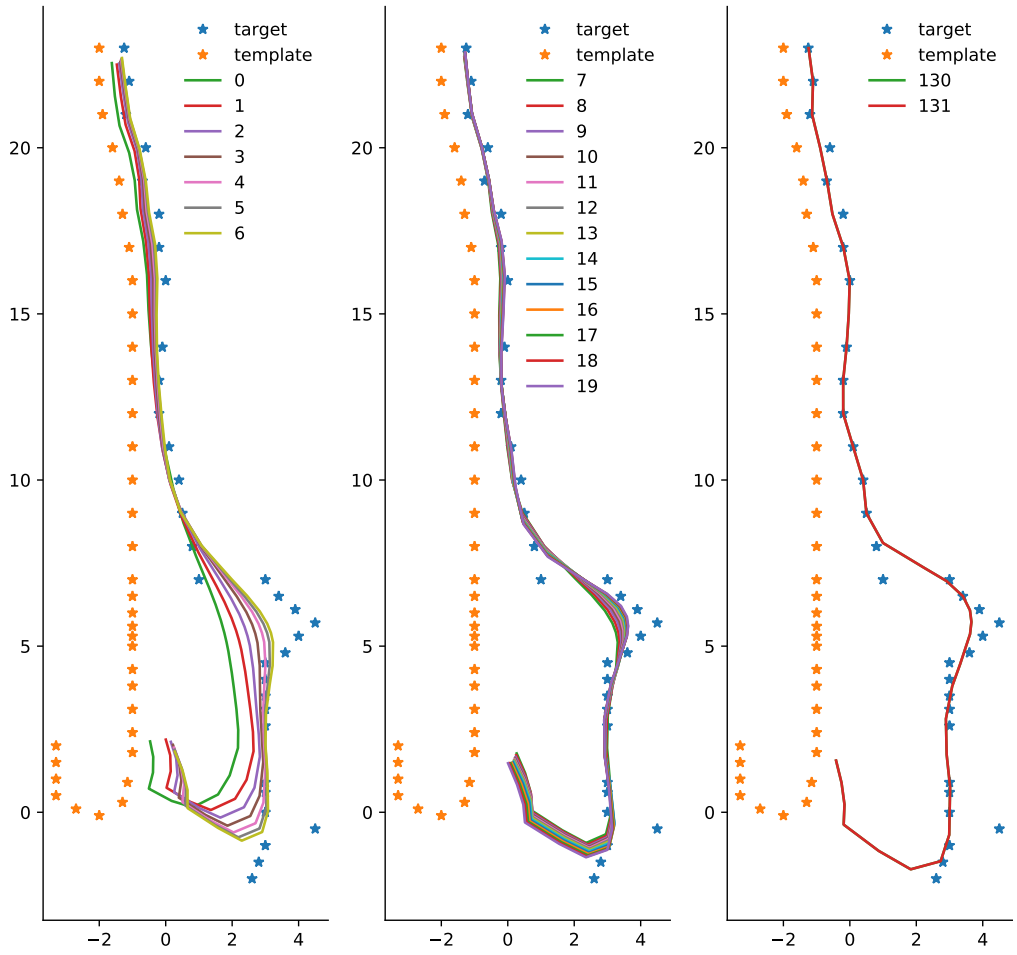


Figure 5.3: Evolution of deformed template over representative iterations of SFGP. The target and template are represented with blue and orange stars, respectively. The different iterations are represented with solid lines with different colours as indicated in the legend. Only a subset of the total iterations are represent to exemplify relevant behaviour at different times. During the first iterations (on the left), the deformed template has presents considerable differences between consecutive iterations for the majority of the shape. After this initial behaviour, a large region of the template remains unchanged as smaller parts are slowly adjusted and fitted to the target (middle plot). Finally, the last iterations present convergence of the method, where there difference between iterations is smaller than a given threshold (on the right).

a moderate level of deformations included[1]. In order to accurately replicate the ear data challenges, we further create a new dataset, based on the noise level 2 of the *Fish Dataset*. Here, we introduce structured missing data in the following way: we choose one point of the template as centre and increasingly set the width of a squared bounding box around this point — all the points within the box are removed.

Setting

We consider different variations of our method, as well as different variations of BCPD, in order to show the relevance of the introduced modifications. Their description can be found in Table 5.1. To fairly compare our method with BCPD, we set their parameters in the same way whenever possible — consequently, we will use the Squared Exponential kernel for our model. The remaining parameters are tuned with the deformation level 1 for both methods, by grid search.

[1] A more detailed description of the dataset can be found in [144]

Name	Description
SFGP_Full	SFGP in its complete version
SFGP_bcpdReg	SFGP where the registration variance ς^2 is taken as a scalar instead of a vector, computed according to BCPD equations
GPReg_noTresh	SFGP without the threshold for missing points P_{MIN}
GPClosestPnt	registration with GPR, but where the correspondence part is achieved by taking the closest point, i.e. not considering multi-annotators
BCPD_Standard	BCPD method with the standard parameters
BCPD_Opt_Norm	BCPD method with optimized parameters for the *Fish Dataset*, with normalization of both shapes
BCPD_Opt_noNorm	BCPD method with optimized parameters for the *Fish Dataset*, without normalization of both shapes, since this is not used in our method and could potentially benefit it in some cases.

Table 5.1: Brief description of the different methods used in the experiments. Note that, for BCPD, by no normalization we mean that shapes maintain their relative size. They are actually both normalized with respect to the target shape size, as recommended by BCPD authors.

All methods consisting of GPSF variations (i.e. *GPSF_Full*, *GPSF_bcpdReg*, *GPSF_noTresh*) use a Squared Exponential Kernel, with a variance of 0.05 and a lengthscale of 1.5. The outlier probability is set to 0.1 and $P_{MIN} = 0.01$. The initial value for the registration variance is $\varsigma^2 = 1$. *GPClosestPnt* uses the same kernel. The variance for the observations (constant over the iterations) is set to 0.1 and the maximum distance for the closest point attribution is 0.15. The parameters for BCPD can be found in Table 5.2, where we keep the notation used in the original paper and in the authors implementation. The parameters for all methods were optimized on the *Fish Dataset* with deformation level 2, by grid search. They are kept constant throughout the experiments, except for the variation of ω when pointed out.

	ω	λ	β	γ	normalization
BCPD_Standard	0.1	2	2	3	e
BCPD_OPT_Norm	0.1	1	1.5	2	e
BCPD_OPT_noNorm	0.1	1	10	0.1	x

Table 5.2: Parameters for BCPD experiments.

Metrics

For the evaluation of results we mainly look at the Euclidean distance error between corresponding deformed template $\bar{t}_i$ points and the ground truth s_i^*, i.e. the complete and deformed target shape without noise, averaged over the shape, so $d(s,t) = \frac{1}{N_T}\sum_{i=1}^{N_T} \|s_i^* - \bar{t}_i\|_2^2$. This is then averaged over the entire dataset, consisting of 100 samples. However, it should be noted that BCPD will occasionally not lead to a successful registration, in which case it does not produce an output or does not produce correspondence for any point. Since this result will not be taken into account for the distance metric and often occurs for the most challenging settings, we also present the fraction of successful registration. Our method does not consider a failed registration unless there are no deformations found in the first iteration.

Discussion

The results for all the considered methods and data variations can be found in Figure 5.5. Our main focus is the dataset with increasing level of missing region (Figure 5.5(a)), as this closely replicates the challenges in the ear reconstruction problem. While for the lowest level it is evident that BCPD (when optimized) performs better, as we increase the missing area, our method presents a progressive advantage. Comparing *SFGP_Full*, *SFGP_bcpdReg* and *SFGP_noTresh* it becomes clear why those modifications are advantageous when facing extensive missing regions. It is also evident that the closest point approach has the poorest performance overall.

It is also interesting to look at the results in the presence of outliers (Figure 5.5(c) and Figure 5.5(d)), for which we tested all methods with $\omega = 0.1$ and $\omega = 0.3$ (with the exception of *GPClosestPnt* where this is not applicable), since this parameter reflects the expected outlier probability. While BCPD performs better if this parameter is adequately adjusted to the real outlier occurrence, we note that our method presents almost the same results for both settings. Therefore, if one does not have any prior knowledge on this quantity, our method is more suitable.

Looking at the variation of noise (Figure 5.5(e)) and deformations (Figure 5.5(b)), we see that overall an adequately fitted BCPD outperforms our method and is able to achieve lower errors, even when both parameters are previously tuned. We also note that the two variations of our method always perform better than the full proposal for these scenarios — the proposed alterations do not bring an advantage when we are not dealing with structured missing data. However, this decrease in performance is deemed acceptable given the gain it provides in Figure 5.5(a) and when compared with the Closest Point proposal always leads to lower error.

An intuition of why our method is able to perform well for extensive missing data is offered in Figure 5.6, where we compare the fitting results from BCPD with different levels of deformation and our proposed method. We can see that the main challenge in achieving an adequate fitting with BCPD is that if we allow for a high level of deformation then the missing region collapses, while in the opposite case the deformations are not enough to fit all details on the non-missing parts. With our approach we are able to avoid collapsing, while allowing enough deformation to fit the fine details.

Furthermore, in Figure 5.7 we present additional metrics for the missing region version of this dataset. The high performance on both recall and precision for our method tells us that the lower distance error observed before is in fact coming from an accurate identification of missing points. We notice that the GP with closest point presents a very high precision but at a cost of a low recall.

5.4.3 3D Ear simulated data

Dataset

In this section we test our method with 3D ear data, the main goal of our work. For this we take the *Simulated Dataset* as described in Chapter 4.2. However, we consider here the fitting of a subset of shapes, as the dataset is composed from 500 samples, but it does not present much variability across them. Therefore, we consider the shapes with more deformation with respect to the template (measured

as the average of Euclidean distance between corresponding shape points). The template was chosen as the first shape of the dataset. Usually, one tries to achieve a template as close as possible to all shapes in the dataset (e.g. mean shape) in order to improve shape fitting. However, since we face a problem of lack of variability in our dataset, we opt for this to have a larger difference between the template and targets (naturally, this shape is not included in the possible targets).

Setting

We compare GPSF and BCPD for the *Simulated Dataset*. Since initial rough alignment is assumed, the template is manually pre-positioned with respect to one of the samples; this leaves only minor rotation and translation effects for the remaining shapes. Both methods take as input the selected template and the target, providing as output the deformed template that best fits each sample. The latter is used to evaluate the results, by comparison with the real (unmodified) target shape, i.e., the samples in the *Ear Dataset*.

The parameters for both methods were individually tuned by grid search with a sample from the *Simulated Dataset* not included in the test set. For GPSF the Squared Exponential Kernel, has a variance of 10 and a lengthscale of 10. The initial value for the registration variance is $\varsigma^2 = 5$. The outlier probability and missing threshold remain unchanged from the previous setting, i.e., they are set to $\omega = 0.1$ and $P_{MIN} = 0.01$. The parameters for BCPD are found to be $\lambda = 10$, $\beta = 1$, $\omega = 0.3$, $\gamma = 0.1$ and normalization set to 'x'.

Discussion

Figure 5.8 presents the fitting results with both BCPD and SFGP, for the *Simulated Dataset*. For fairness, we consider the SFGP with a Squared Exponential kernel, even if the final goal is to include other information regarding the shape characteristics. The metric in place remains the distance error between the deformed template and target, although we separately look at missing and non-missing regions. The boxplots depict the error for the points in all shapes of the test dataset.

The error for the non-missing points is fairly similar between SFGP and BCPD, with approximately the same median value. However, SFGP presents a smaller interquartile range. Thus, BCPD exhibits more variability in distance error and will more often present higher and lower values from the median. In practice, this means that there is a level of fitting that SFGP is not able to achieve when compared to BCPD, but it also means that SFGP is less prone to bad fitting.

Nonetheless, the most considerable difference in behaviour is in the fitting of missing data. In fact, the first quartile of BCPD lies above the third quartile of SFGP, when it comes to distance error of these regions. It should be noted that failures in missing regions will generally lead to considerably higher distance errors. Points will often fit the nearest observed shape feature, that could lie at a relatively high distance. This contrasts with non-missing regions, where errors in fitting will generally cause a correspondence to nearby points. Thus, it is imperative to prevent failure within missing regions.

The fact that SFGP is less susceptible to extensive missing data is an advantage on its own. However, it also instils confidence on its potential when further prior information is added to the kernel. This

additional knowledge is expected to help on both the identification of absent points and the correct prediction of the deformation, upon such identification.

A remark on computational time is also due. While BCPD takes around 10 seconds per shape with acceleration and 20 minutes without, SFGP currently requires around 100 minutes. However, this is an unoptimized version and does not employ any method to deal with high dimensionality of the data, to which GPR is sensible. On one hand, the same acceleration used to obtain P in BCPD, will reduce computational time. On the other, several GPR tools for large data setting already exist and could be included to reduce time complexity. Both of these should be contemplated in future work.

5.5 Application to the real dataset

After evaluating quantitative results with simulated data, we move to the *Head dataset*, where we conduct a qualitative evaluation, in the absence of GT. This section serves both as demonstration of SFGP capabilities to overcome several issues with the ears, but also as guide to apply our method in other contexts.

For this experiment, we take the mean shape from the *Ear dataset* as a template and fit it to a selected target shape from the *Head dataset*. The target was selected in order to exhibit the most commonly encountered challenges, i.e., extensive missing data and outliers, together with considerable deformations with respect to the target. SFGP was applied with the parameters specified in Section 5.4.3, except for the kernel type and its corresponding subparameters.

5.5.1 Kernel variation

While the previous simulations were conducted with an SE kernel for comparisons with BCPD, the main goal is to use more powerful kernels in order to increase the performance of SFGP. In general, we will build a model kernel by combining pre-existing kernels through allowed operations. Here, we restrict the choices to weighted sums, i.e.

$$K_{model}(x, x') = \sum_{i=1}^{N_K} \theta_i K_i(x, x'),$$

where $K_i(x, x')$ is a valid kernel and θ_i its respective weight. In particular, we have considered N_K to be at most 2, in which case $K_1(x, x') = K_{PDM}(x, x')$ and $K_2(x, x')$ is taken as the SE, RQ or Matérn kernel. While $K_{PDM}(x, x')$ does not have parameters, all the other kernels do.

Within the GP literature hyperparameter selection and model selection are generally conducted through MLE [86], but the existence of GT is assumed. In the *Head dataset*, it is possible to use the *Ear Dataset* for this purpose but its usefulness is rather limited. Nonetheless, this method was used to estimate reasonable values for the lengthscale and variance, as $\hat{l} = 10$ and $\hat{\sigma}^2 = 10$. However, when trying to include K_{PDM} in combination with different kernels the process is meaningless, as the kernel was built from that same dataset and leads to erroneous results.

Estimation of said parameters and weights from a dataset without GT is something we have approached during our work, but still without conclusive results. Therefore, we follow with a qualitative evaluation of the result of different combinations of kernels and parameters, depicted in Figure 5.9. For this comparison the parameters were set as follows.

First, note that the kernel variance will control the amplitude of deformations suffered by the template. The estimated value $\hat{\sigma}^2$ is a reasonable estimate but when adding K_{PDM} is should be adjusted accordingly. Therefore, the final kernel is built as

$$K_{model}(x, x') = \theta_{PDM} K_{PDM}(x, x') + (1 - \theta_{PDM}) K_2(x, x'),$$

where θ_{PDM} controls the relative weight of the empirical kernel and was set to 0.8, unless otherwise specified. The variance σ_2^2 for $K_2(x, x')$ is set as $\hat{\sigma}^2(1 - \theta_{PDM})$, in order to keep the overall magnitude of deformations close to the estimated level. The lengthscale is set to the estimated value $\hat{l}$, unless otherwise specified.

Discussion

Unsurprisingly, the combination with better visual results is the RQ kernel with the PDM kernel in 5.9(e). It provides the benefits of the empirical covariance with the additional freedom of the RQ kernel. At a first glance, the addition of the RQ kernel could seem unnecessary, from comparison with 5.9(b). However, looking at the shape details, it becomes clear that the RQ kernel is fundamental for the fine deformations.

Further, it is evident that the inclusion of the K_{PDM}, regardless of the other kernel components in place, prevents unwanted behaviour on the outlier region. This was a limitation found in the previous probabilistic approach (corresponding to 5.9 here), that the empirical kernels helps to overcome. The prior knowledge on the ear shape reduces the chances of unlikely deformations on the top region, as these are not observed in the training dataset.

We compare the influence of adding different smooth kernels to K_{PDM}, by taking into account 5.9(d), 5.9(e) and 5.9(f). The Matérn kernel leads to closer fitting on fine details, but this increased non-rigid deformation extends to unwanted regions on the top and right parts of the shape. On the other hand, the extreme smoothness of the SE kernels does not offer the same close fitting as the RQ and Matérn. The RQ kernel expresses a particularly important concept for shape deformation, by considering different levels of deformations on different lengthscales. This is reflected on the observed results.

However, one could wonder whether a different lengthscale for the SE kernel could allow for finer deformations and surpass the output of th RQ kernel. In 5.9(h) we present the results for a smaller lengthscale. As expected, the accuracy of fitting in detailed regions is increased, but this additional flexibility causes a collapse on other problematic regions, that the empirical kernel can no longer prevent. Lastly, the importance of the weighting parameter between different kernel components is illustrated in 5.9(g), where θ_{PDM} is set to a lower level. Naturally, this reduces the influence of the empirical kernel and the corresponding advantages.

This is by no means an exhaustive comparison of parameters and kernel combinations. It merely aims at illustrating the most relevant behaviours of commonly used kernels and the interpretation of their effects. It also corroborates the previous statement of the potential of SFGP under more complex prior information.

5.5.2 Limitations

Despite exhibiting an improved behaviour with respect to the existing approaches, the application of SFGP with the selected kernel from the previous section (K_{RQ} and K_{PDM}) still presents some limitations under particularly difficult scenarios. Two representative examples are found in Figure 5.10. This experiment follows the same setting as the previous section, though depicting different target shapes.

While we focused on the collapse of the bottom section and were able to overcome it, the outliers on the top part remain a problem in some cases. This is usually observed when there is almost no difference in curvature between the border of the ear and the skin around it, as seen in Figure 5.10(a). Another obstacle are regions that remain almost disconnected from the main portion of the shape and are initially far away from the template (Figure 5.10(b)). Because the algorithm progressively advances along the shape, if it fails to attach to any part of an almost disconnected region, it will not be able to later recover from it.

5.6 Concluding remarks

We developed a method that bridges the gap between the Gaussian Process framework used in 3D Morphable Models and the probabilistic registration methods, by formulating the shape fitting problem in a GPR multi-annotator setting. This allows us to benefit from advantages on both sides and obtain a method particularly suited for shape fitting in the presence of extensive missing data — a useful tool for challenging shapes such as the human ear.

Naturally, even if the missing points are correctly identified, the shape prediction in those regions will be as good as the prior model. Therefore, it is beneficial to have a more complex and accurate model, able to express more knowledge on the particular shape. We have seen that a sum of a RQ kernel with an Empirical kernel produces good results, but a more exhaustive study would be necessary in order to reach more objective conclusions.

In particular, it would be interesting to have a fairly automated way to achieve an adequate model. The results achieved so far may be a good starting point as they lead to complete shapes, without the collapses observed in the first part of the thesis.

It would also be pertinent to study how the parallel with the probabilistic registration holds when we introduce the missing point set and the threshold, to have a more theoretical insight on the properties of our method.

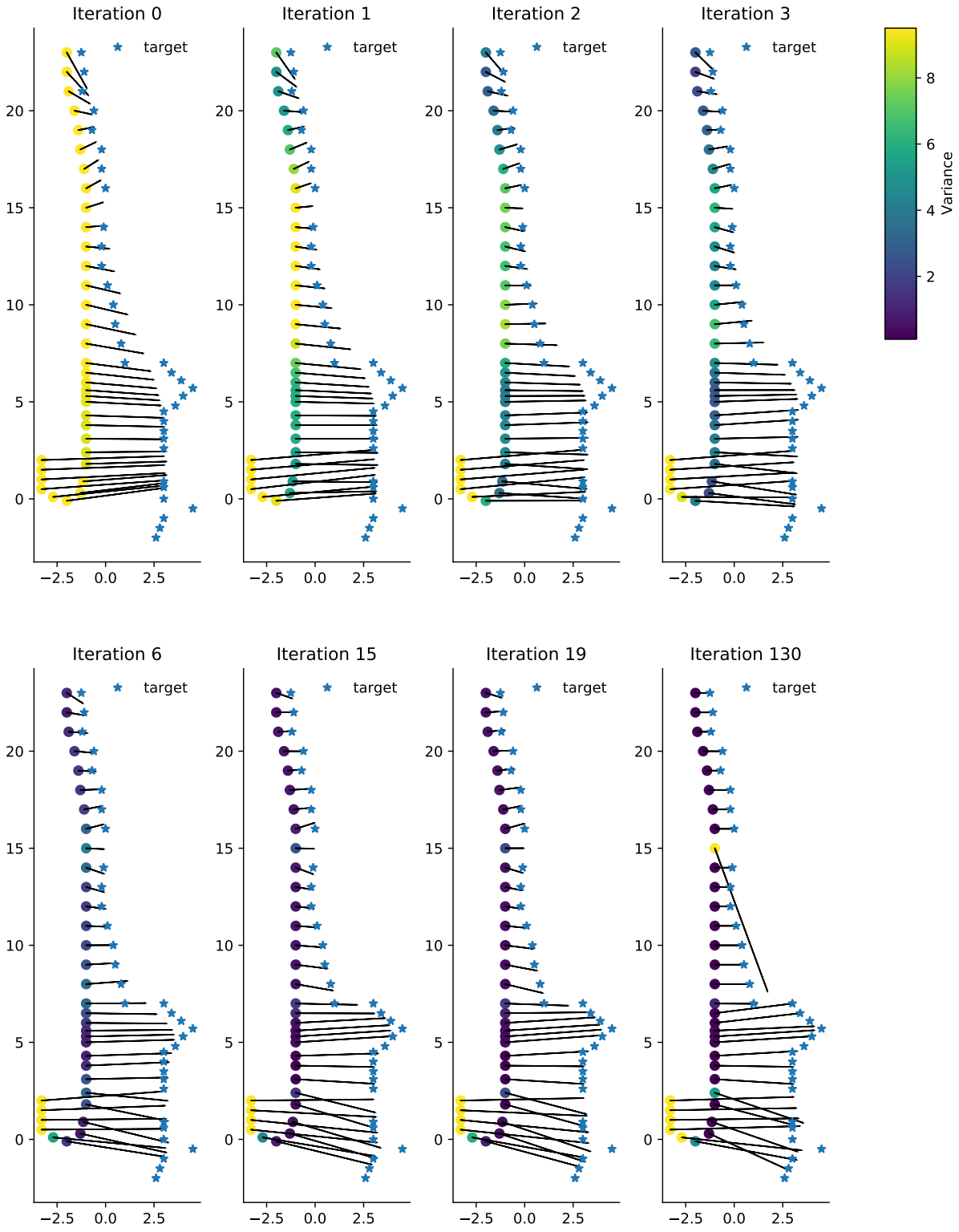
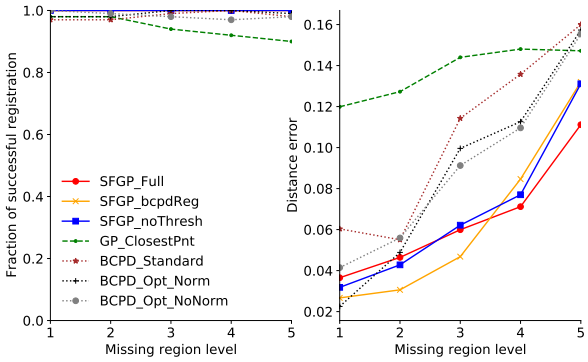
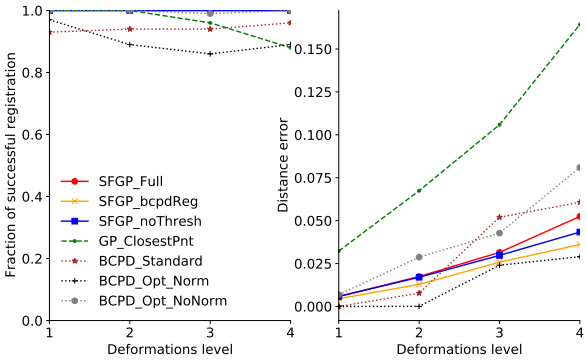


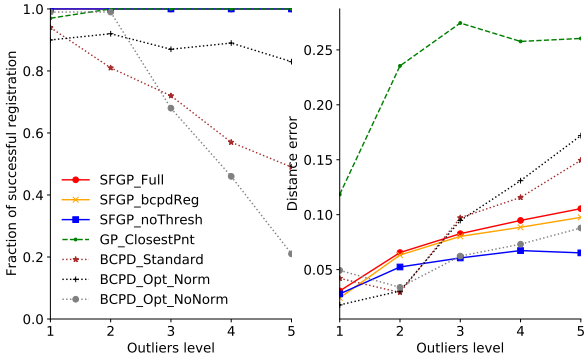
Figure 5.4: Representation of deformations and variance of each point of the template for specific iterations of SFGP. Deformation vectors are represented with a black line originating at the template point and variance of each point by color gradient. All template points are initialized with the same variance (Iteration 0) and, since they are at similar distances to several target points, the deformations follow the general direction of T to S, with low deformation on the y-axis. The variance is then progressively reduced as the points reach correspondences with a high level f confidence. At the end, (Iteration 130) points in T_{corr} will have very low variance, while missing points will present high values for this variable.



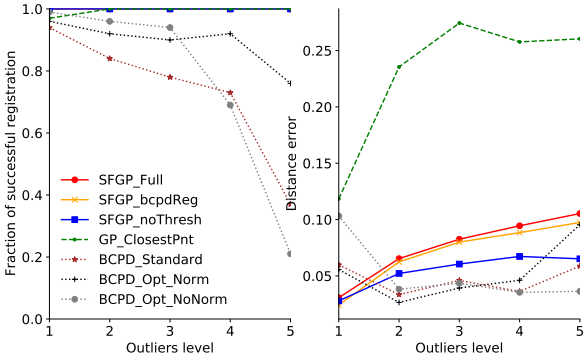
(a) Increasing level of missing region



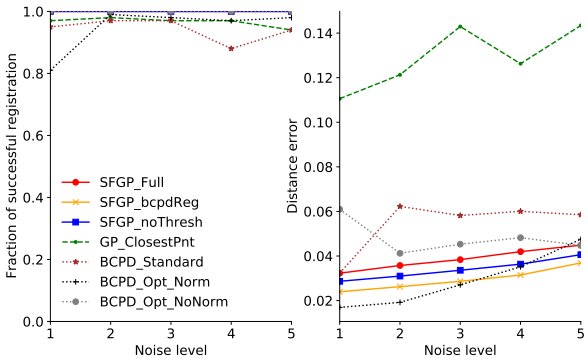
(b) Increasing level of deformations



(c) Increasing fraction of outliers, for $\omega = 0.1$

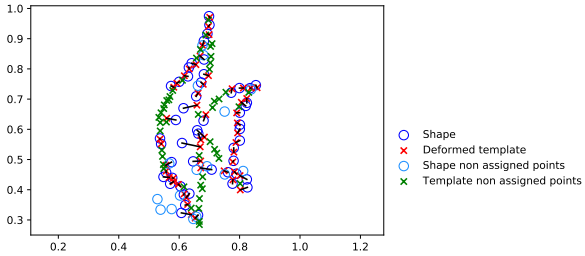


(d) Increasing fraction of outliers, for $\omega = 0.3$

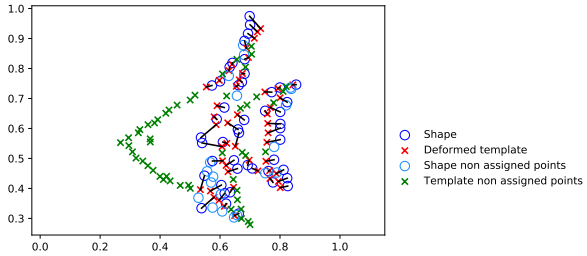


(e) Increasing level of noise

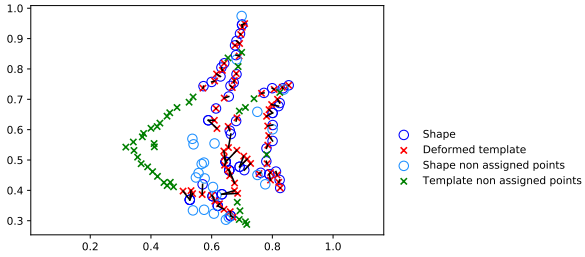
Figure 5.5: Results for *Fish Dataset* with different types of modifications. The x-axis always depicts an increase in a given data modification, while the y-axis provides the ratio of successful items registered and the average distance error. Variants of SFGP are depicted with full lines, variants of BCPD are depicted with thin dashed lines and GP with closest point with a thick dashed line.



(a) BCPD registration with low λ



(b) BCPD registration with high λ



(c) Our registration

Figure 5.6: Example for missing data performance with *Fish Dataset*. The target can be seen in blue circles, while the deformed template after registration is represented with red and green crosses. Points with and without correspondence are identified both for the template and target shape. The two results for BCPD are run with different values of parameter λ responsible for controlling the expected length of the deformation — small values of λ allow for more deformation and vice-versa. With BCPD we can either get enough non-rigidity to fit the existing points, at cost of collapsing the missing region, or preserve the shape of this segment at the cost of a rigid transformation that can not appropriately fit the observed points. With our approach we are able to allow a level of non-rigid deformations which fits the shape details, while correctly identifying the missing regions.

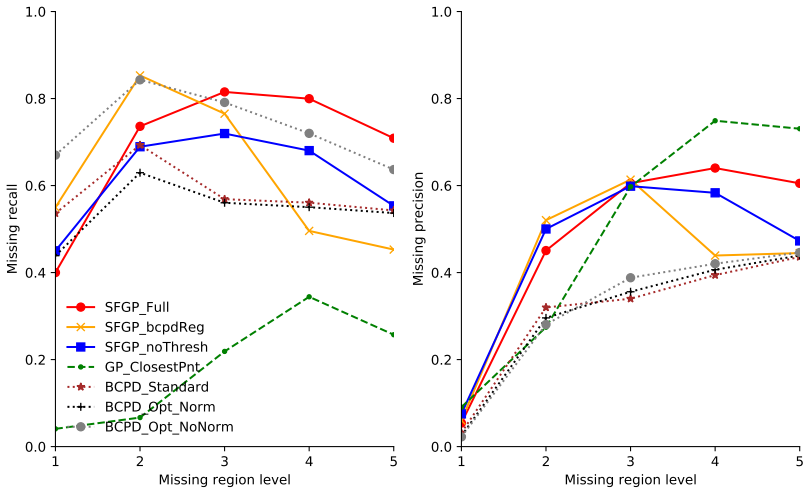


Figure 5.7: Recall and precision for increasing missing region. A high recall indicates that the method is able to identify most of the missing points, while a high precision means that most of the points identified as missing are in fact missing.

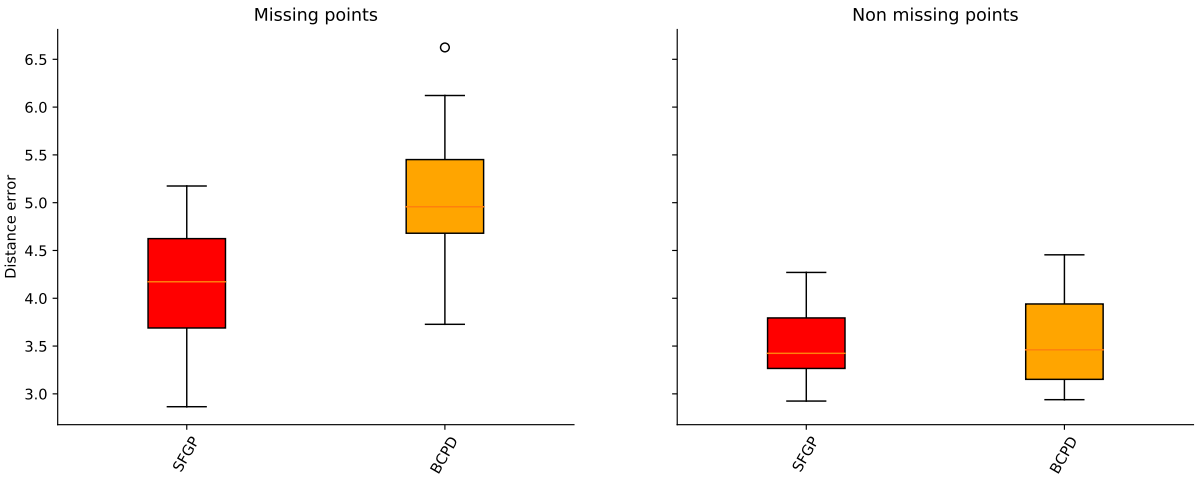
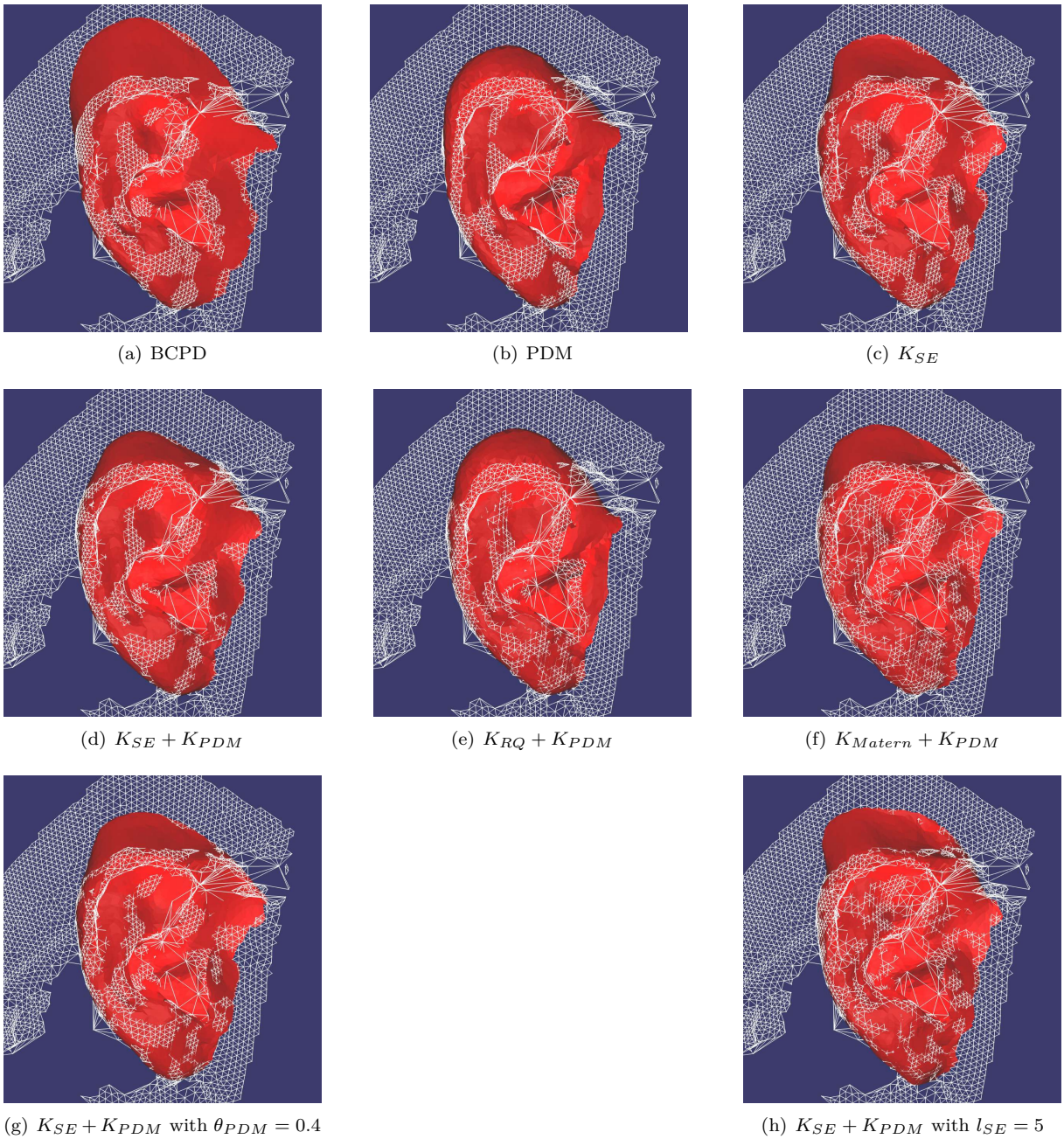


Figure 5.8: Results for the *Simulated Dataset* with SFGP and BCPD. The boxplot is obtained from the mean Euclidean distance between the true and deformed template point sets, for all shapes considered. On the left we consider only the subset of points that are missing, while on the right only the existing ones are considered.

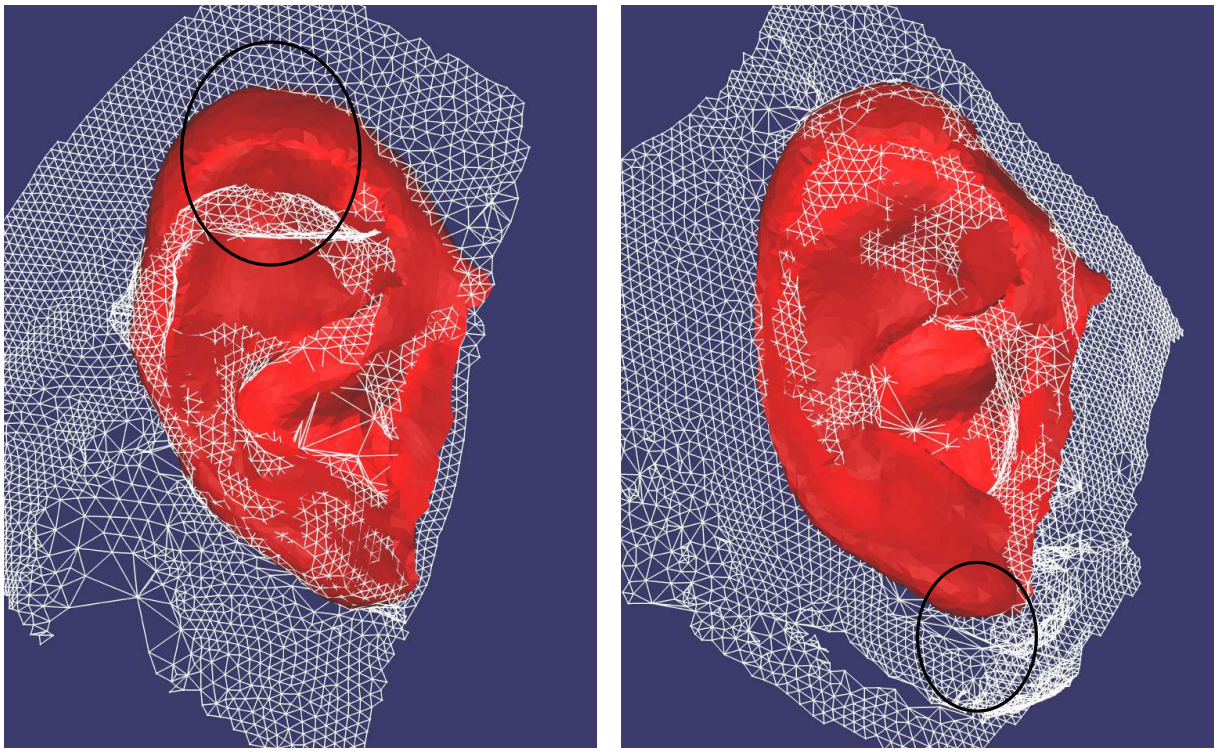


(a) BCPD (b) PDM (c) K_{SE}

(d) $K_{SE} + K_{PDM}$ (e) $K_{RQ} + K_{PDM}$ (f) $K_{Matern} + K_{PDM}$

(g) $K_{SE} + K_{PDM}$ with $\theta_{PDM} = 0.4$ (h) $K_{SE} + K_{PDM}$ with $l_{SE} = 5$

Figure 5.9: Example of SFGP fitting with different choices of kernel. The target shape is depicted as a white triangulated mesh, while the final deformed template is represented as an red mesh. SFGP was run under the same conditions for all settings, except for the kernel used. Unless otherwise specified, the kernel lengthscale is $l = 10$ and $\theta_{PDM} = 0.8$.



(a) Case 1 (b) Case 2

Figure 5.10: Limitations of SFGP. The target shape is represented as a white triangulated mesh and the final deformed template as a red mesh. Black circles indicate the areas of interest, i.e., where SFGP fails. On the left, the template is not able to correctly differentiate the outliers from the ear shape, while on the right, due to an almost unconnected region on the bottom, the template is not able to reach the full target shape.

Chapter 6

Discussion and Future Work

On the particular case of 3D ear shape modelling, we have produced a representative *Simulated Dataset* to be used for evaluation of any desired metrics. Besides, we have obtained complete and accurate shapes from a noisy and incomplete *Head dataset*, which can be used in the future for development of increasingly accurate models[1]. This constitutes an improvement on publicly available pre-existing data and models of 3D human ears, with reduced representation across the literature.

Furthermore, we have proposed an unsupervised pipeline, able to produce complete meaningful shapes from point clouds with extensive presence of outliers, missing data and noise. The pipeline is composed of two registration steps (initial and refinement) and a shape completion task. Not only did we provide a selection of specific methods to be used, but the approach taken to reach the final pipeline, together with the multiple metrics involved is one that could assist in the selection of upcoming pipelines in guided and structured approach. This contrasts with a vast majority of existing methods, where alternative choices are often overlooked *a priori*, with unsubstantiated motivation.

Within this context, we studied a particular instance of ICP, which we denoted RANSIP, particularly suited to the characteristics of our dataset. It is found to be a simple and yet effective approach, competing with related and more complex randomized approaches, such as SDRSAC. In the future, further comparison with ICP methods and alternative datasets would be beneficial, in order to better understand its potential outside the scope of our pipeline.

We then moved to a unified approach of the last pipeline steps, assuming an initial rough alignment of the dataset. This is a common assumption for many shape modelling tasks, so the applicability is not limited. The unified line of work aims at bringing together the flexibility of the GP framework in model building and the advantages of probabilistic registration when dealing with outliers and missing data. Therefore, we formulated the shape fitting problem within the GP framework with soft-assignment, as a multi-annotator Gaussian Process Regression.

In order to have a principled computation of the annotators variance, we established a parallel with BCPD, a probabilistic registration method, under the assumption of no missing points. Despite their similarities, we further evidenced how the differences are fundamental in handling challenging shapes,

[1] the dataset will be made publicly available upon acceptance of the second paper

especially the ones with extensive missing data.

This led to the formulation of SFGP, a registration and fitting algorithm, tested with different kinds of data, both on the 2D and 3D setting. For simulated data, we observed an improvement with respect to state-of-the-art registration methods and current GP registration proposals, in particular when considering the distance error for missing points. We finally applied SFGP to the real 3D ears of the *Head dataset* with different choices of kernels, achieving considerable visual improvements on the registration results.

6.1 Future Work

Given the extensive number of existing methods on shape fitting, modelling and registration, even review papers often focus on a particular segment of the literature and the connection between them is usually overlooked. Chapter 5 is a step further in a unified formulation of registration and shape fitting methods. The literature review done throughout the thesis, and in particular over that chapter, provides a hint on how the unified point of view applies to related approaches, other than BCPD and GP. However, this is done in an informal manner and it would be interesting to determine exactly which methods are amenable to a formulation in the same general framework. Most regularization terms can easily be related to an existing kernel and the remaining ones could point to new kernels, relevant to shape analysis. Setting parameters on alternative methods can also be facilitated if such connections are established.

Nonetheless, it can be rightfully argued that different formulations can lead to optimization problems more amenable to large data techniques, or generally faster and efficient algorithms. However, there has been a growing number of large data solutions for GP in recent years [145], as their use much overpasses the application to shape modelling. This is an avenue that we started to explore, but still requires a more extensive study, albeit extremely promising in increasing the efficiency of our algorithm SFGP. On a similar note, the particular setting of shape modelling, where part of the matrices involved remain constant or are subsets of pre-existing ones, hint at more efficient formulations of the update equations. This could potentially speed up the process without resorting to GP approximation techniques.

The inclusion of explicit rigid transformations would undoubtedly be the next step for a generic formulation of our problem. However, it is an open question whether this would provide an advantage by compensating mistakes in the initial alignment or degrade the current performance observed in the experiments. Given the close relationship with BCPD, and since the latter includes rigid transformations, it could be a line worth pursuing. However, the formulation entirely within a GP context is not trivial.

The freedom in model selection through kernel design means that there are countless possible models to choose from. On a first instance by determining the combination of kernels to use and, on a second, be tuning their hyper parameters. Both these steps would ideally be done in an automated way. Model selection in a Bayesian context has pre-existing proposed solutions, but they often struggle in the GP setting due to numerical problems, amongst others. It is also not evident how one could take advantage of an existing limited dataset to guide kernel modelling, except for the current empirical kernel. We have

also pursued this topic, but with no definitive conclusion on how to include this in our formulation.

Finally, criteria to evaluate the shape modelling results in real data is a problem that remains open and of crucial importance for the development of further work. The probabilistic nature of GP might offer a possible solution, since it provides a measure of confidence in the output.

The work here presented has advanced shape modelling literature, both regarding the case study of the ear (our initial motivation) and alternative shapes with similar challenges. The pipeline in Chapter 4 is expected ot impact tasks such as prothesis production, by providing a solution to retrieve full models from limited data and in the presence of reduced pre-existing datasets. The method developed in Chapter 5 could improve existing approaches on both shape fitting and registration fields, and was found to adequately handle samples with extensive regions of missing data. Furthermore, this formulation shows high potential for subsequent improvement both on a computational and modelling point of view.

Bibliography

[1] I. Kakadiaris, G. Passalis, G. Toderici, M. N. Murtuza, Y. Lu, N. Karampatziakis, and T. Theoharis. Three-dimensional face recognition in the presence of facial expressions: An annotated deformable model approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29:640–649, 2007.

[2] X. Lu and A. Jain. Deformation modeling for robust 3D face matching. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 1377–1383, 2006.

[3] D. Rahbani, A. Morel-Forster, D. Madsen, J. Aellen, and T. Vetter. Sequential Gaussian process regression for simultaneous pathology detection and shape reconstruction. In *Medical Image Computing and Computer Assisted Intervention*, page 429–438, 2021.

[4] B. Vafaeian, H. Tajmir Riahi, H. Amoushahi, N. Jomha, and S. Adeeb. A feature-based statistical shape model for geometric analysis of the human talus and development of universal talar prostheses. *Journal of Anatomy*, 240, 09 2021.

[5] V. Blanz and T. Vetter. A morphable model for the synthesis of 3D faces. *26th Conference on Computer Graphics and Interactive Techniques*, 09 2002.

[6] J. Booth, A. Roussos, S. Zafeiriou, A. Ponniahy, and D. Dunaway. A 3D morphable model learnt from 10,000 faces. In *2016 IEEE Conference on Computer Vision and Pattern Recognition*, pages 5543–5552, June 2016.

[7] P. Paysan, R. Knothe, B. Amberg, S. Romdhani, and T. Vetter. A 3D face model for pose and illumination invariant face recognition. In *2009 Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance*, pages 296–301, Sep. 2009.

[8] R. Zolfaghari, N. Epain, C. T. Jin, J. Glaunès, and A. Tew. Generating a morphable model of ears. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1771–1775, March 2016.

[9] H. Dai, N. Pears, and W. Smith. A data-augmented 3D morphable model of the ear. In *2018 13th IEEE International Conference on Automatic Face Gesture Recognition*, pages 404–408, May 2018.

[10] J. Zhu, S. C. Hoi, and M. R. Lyu. Nonrigid shape recovery by Gaussian process regression. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1319–1326, 2009.

[11] M. Lüthi, T. Gerig, C. Jud, and T. Vetter. Gaussian process morphable models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(8):1860–1873, 2018.

[12] F. Valdeira, R. Ferreira, A. Micheletti, and C. Soares. From noisy point clouds to complete ear shapes: Unsupervised pipeline. *IEEE Access*, 9:127720–127734, 09 2021.

[13] F. Valdeira, R. Ferreira, A. Micheletti, and C. Soares. Probabilistic registration for Gaussian process 3D shape modelling in the presence of extensive missing data, 2022. arXiv:2203.14113.

[14] D. G. Kendall. The diffusion of shape. *Advances in Applied Probability*, 9(3):428–430, 1977.

[15] I. L. Dryden and K. V. Mardia. *Statistical Shape Analysis with Applications in R. Second Edition.* John Wiley and Sons, 2016.

[16] A. Brunton, A. Salazar, T. Bolkart, and S. Wuhrer. Review of statistical shape spaces for 3D data with comparative analysis for human faces. *Computer Vision and Image Understanding*, 128:1–17, 2014.

[17] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Active shape models—their training and application. *Computer Vision and Image Understanding*, 61(1):38–59, jan 1995.

[18] E. Klassen, A. Srivastava, M. Mio, and S. Joshi. Analysis of planar shapes using geodesic paths on shape spaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(3):372–383, 2004.

[19] P. Michor, D. Mumford, J. Shah, and L. Younes. A metric on shape space with explicit geodesics. *Atti Accad. Naz. Lincei Cl. Sci. Fis. Mat. Natur. Rend. Lincei (9) Mat. Appl.*, 19, 07 2007.

[20] A. Elad and R. Kimmel. On bending invariant signatures for surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(10):1285–1295, 2003.

[21] H. Laga. A survey on nonrigid 3D shape analysis. *Academic Press Library in Signal Processing*, pages 261–304, 01 2018.

[22] I. Jolliffe. *Principal Components Analysis for special types of data.* Springer New York, NY, 2002.

[23] Y. Sahillioğlu. Recent advances in shape correspondence. *The Visual Computer*, 36, 08 2020.

[24] J. Zhang, X. Chen, Z. Cai, L. Pan, H. Zhao, S. Yi, C. Yeo, B. Dai, and C. Loy. Unsupervised 3D shape completion through gan inversion. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1768–1777, jun 2021.

[25] T. Cashman and A. Fitzgibbon. What shape are dolphins? building 3D morphable models from 2d images. *IEEE transactions on pattern analysis and machine intelligence*, 35, 02 2012.

[26] O. Litany, A. Bronstein, M. Bronstein, and A. Makadia. Deformable shape completion with graph convolutional autoencoders. pages 1886–1895, 06 2018.

[27] D. Anguelov, P. Srinivasan, D. Koller, S. Thrun, J. Rodgers, and J. Davis. Scape: Shape completion and animation of people. *ACM Transactions on Graphics*, 24:408–416, 07 2005.

[28] L. Pishchulin, S. Wuhrer, T. Helten, C. Theobalt, and B. Schiele. Building statistical shape spaces for 3D human modeling. *Pattern Recognition*, 67:276–286, 2017.

[29] O. Freifeld and M. Black. Lie bodies: A manifold representation of 3D human shape. In *European Conference on Computer Vision*, pages 1–14, 10 2012.

[30] F. Liu, T. Luan, and X. Liu. 3D face modeling from diverse raw scan data. In *2019 IEEE/CVF International Conference on Computer Vision*, pages 9407–9417, 02 2019.

[31] A. Mueller, P. Paysan, R. Schumacher, H.-F. Zeilhofer, B.-I. Berg-Boerner, J. Maurer, T. Vetter, E. Schkommodau, P. Juergens, and K. Schwenzer-Zimmerer. Missing facial parts computed by a morphable model and transferred directly to a polyamide laser-sintered prosthesis: an innovation study. *British Journal of Oral and Maxillofacial Surgery*, 49(8):e67–e71, 2011.

[32] Y. Li, T.-S. Tan, I. Volkau, and W. L. Nowinski. Model-guided segmentation of 3D neuroradiological image using statistical surface wavelet model. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–7, 2007.

[33] C. Davatzikos, X. Tao, and D. Shen. Hierarchical active shape models, using the wavelet transform. *IEEE Transactions on Medical Imaging*, 22(3):414–423, 2003.

[34] D. Nain, S. Haker, A. Bobick, and A. Tannenbaum. Shape-driven 3D segmentation using spherical wavelets. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2006*, pages 66–74, 2006.

[35] B. Amberg, R. Knothe, and T. Vetter. Expression invariant 3D face recognition with a morphable model. In *International Conference on Automatic Face and Gesture Recognition*, pages 1–6, 10 2008.

[36] F. Haar and R. Veltkamp. 3D face model fitting for recognition. In *2008 European Conference on Computer Vision*, volume 5305, pages 652–664, 10 2008.

[37] A. Brunton, T. Bolkart, and S. Wuhrer. Multilinear wavelets: A statistical shape space for human faces. In *2014 European Conference on Computer Vision*, volume 8689, pages 297–312, 01 2014.

[38] T. Luan, F. Liu, and X. Liu. Towards high-fidelity nonlinear 3D face morphable model. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1126–1135, 06 2019.

[39] X. Zhu, Y. Fan, D. Huang, C. Yu, H. Wang, J. Guo, Z. Lei, and S. Li. *Beyond 3DMM Space: Towards Fine-Grained 3D Face Reconstruction*, pages 343–358. 11 2020.

[40] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. *IEEE Transactions on pattern analysis and machine intelligence*, 23(6):681–685, 2001.

[41] M. Smet and L. Van Gool. Optimal regions for linear model-based 3D face reconstruction. volume 6494, pages 276–289, 11 2010.

[42] Z. Fan, X. Hu, C. Chen, and S. Peng. Dense semantic and topological correspondence of 3D faces without landmarks. In *European Conference on Computer Vision*, 2018.

[43] D. C. Schneider and P. Eisert. Fast nonrigid mesh registration with a data-driven deformation prior. In *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*, pages 304–311, 2009.

[44] A. Patel and W. A. P. Smith. 3D morphable face models revisited. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1327–1334, 2009.

[45] L. Tran and X. Liu. Nonlinear 3D face morphable model. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7346–7355, 04 2018.

[46] D. Jiang, Y. Jin, F. Zhang, Z. Zhu, Y. Zhang, R. Tong, and M. Tang. Sphere face model: A 3D morphable model with hypersphere manifold latent space, 2021.

[47] S. Cheng, I. Marras, S. Zafeiriou, and M. Pantic. Statistical non-rigid icp algorithm and its application to 3D face alignment. *Image and Vision Computing*, 58, 11 2016.

[48] A. Brunton, J. Lang, E. Dubois, and C. Shu. Wavelet model-based stereo for fast, robust face reconstruction. In *2011 Canadian Conference on Computer and Robot Vision*, pages 347–354, 2011.

[49] T. Neumann, K. Varanasi, S. Wenger, M. Wacker, M. Magnor, and C. Theobalt. Sparse localized deformation components. *ACM Transactions on Graphics*, 32(6), nov 2013.

[50] C. Ferrari, S. Berretti, P. Pala, and A. del Bimbo. A sparse and locally coherent morphable face model for dense semantic correspondence across heterogeneous 3D faces. *IEEE transactions on pattern analysis and machine intelligence*, PP, 2021.

[51] J. Sassen, K. Hildebrandt, and M. Rumpf. Nonlinear deformation synthesis via sparse principal geodesic analysis. *Computer Graphics Forum*, 39:119–132, 08 2020.

[52] T. Bolkart and S. Wuhrer. A robust multilinear model learning framework for 3D faces. In *2016 IEEE Conference on Computer Vision and Pattern Recognition*, pages 4911–4919, 2016.

[53] V. Abrevaya, A. Boukhayma, S. Wuhrer, and E. Boyer. A decoupled 3D facial shape model by adversarial training. pages 9418–9427, 10 2019.

[54] S. Cheng, M. Bronstein, Y. Zhou, I. Kotsia, M. Pantic, and S. Zafeiriou. MeshGAN: Non-linear 3d morphable models of faces. 03 2019.

[55] A. Ranjan, T. Bolkart, S. Sanyal, and M. Black. *Generating 3D Faces Using Convolutional Mesh Autoencoders*, pages 725–741. 09 2018.

[56] P. Fletcher, C. Lu, S. Pizer, and S. Joshi. Principal geodesic analysis for the study of nonlinear statistics of shape. *IEEE Transactions on Medical Imaging*, 23(8):995–1005, 2004.

[57] C. Zhang, B. Heeren, M. Rumpf, and W. A. P. Smith. Shell pca: Statistical shape modelling in shell space. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1671–1679, 2015.

[58] B. Heeren, C. Zhang, M. Rumpf, and W. Smith. Principal geodesic analysis in the space of discrete shells. *Computer graphics forum*, 37, 06 2018.

[59] R. Slossberg, G. Shamai, and R. Kimmel. *High Quality Facial Surface and Texture Synthesis via Generative Adversarial Networks*, pages 498–513. 01 2019.

[60] Y. Feng, F. Wu, X. Shao, Y. Wang, and X. Zhou. Joint 3D face reconstruction and dense alignment with position map regression network. In *European Conference on Computer Vision*, pages 523–539, 2018.

[61] C. Ferrari, S. Berretti, P. Pala, and A. del Bimbo. A sparse and locally coherent morphable face model for dense semantic correspondence across heterogeneous 3D faces. *IEEE transactions on pattern analysis and machine intelligence*, PP, 2021.

[62] P. Perakis, G. Passalis, T. Theoharis, and I. A. Kakadiaris. 3D facial landmark detection under large yaw and expression variations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(7):1552–1564, 2013.

[63] Z. Fan, X. Hu, C. Chen, and S. Peng. Boosting local shape matching for dense 3d face correspondence. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10936–10946, 2019.

[64] Z. Fan, S. Peng, and S. Xia. Towards fine-grained 3D face dense registration: An optimal dividing and diffusing method. 09 2021.

[65] J. Klatzow, G. Dalmasso, N. Martínez-Abadías, J. Sharpe, and V. Uhlmann. μmatch: 3D shape correspondence for biological image data. *Frontiers in Computer Science*, page 7, 2022.

[66] P. Shen, H. Wei, and Y. Wang. Efficient 3D scan registration for hand modeling and reconstruction. In *2018 Chinese Control And Decision Conference (CCDC)*, pages 5612–5616, 2018.

[67] C. Zhang, W. A. P. Smith, A. Dessein, N. Pears, and H. Dai. Functional faces: Groupwise dense correspondence using functional maps. In *2016 IEEE Conference on Computer Vision and Pattern Recognition*, pages 5033–5041, 2016.

[68] H. Zhu, B. Guo, K. Zou, Y. Li, K.-V. Yuen, L. Mihaylova, and H. Leung. A review of point set registration: From pairwise registration to groupwise registration. *Sensors*, 19:1191, 03 2019.

[69] G. K. Tam, Z.-Q. Cheng, Y.-K. Lai, F. C. Langbein, Y. Liu, D. Marshall, R. R. Martin, X.-F. Sun, and P. L. Rosin. Registration of 3D point clouds and meshes: A survey from rigid to nonrigid. *IEEE Transactions on Visualization and Computer Graphics*, 19(7):1199–1217, 2013.

[70] B. Deng, Y. Yao, R. Dyke, and J. Zhang. A survey of non-rigid 3D registration. 03 2022.

[71] B. Maiseli, Y. Gu, and H. Gao. Recent developments and trends in point set registration methods. *Journal of Visual Communication and Image Representation*, 46:95–106, 03 2017.

[72] G. Hu, F. Yan, J. Kittler, W. Christmas, C. H. Chan, Z. Feng, and P. Huber. Efficient 3D morphable face model fitting. *Pattern Recognition*, 67:366–379, 2017.

[73] X. Zhu, J. Yan, D. Yi, Z. Lei, and S. Li. Discriminative 3D morphable model fitting. *2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition*, 1:1–8, 2015.

[74] V. Blanz, A. Mehl, T. Vetter, and H.-P. Seidel. A statistical method for robust 3D surface reconstruction from sparse data. In *2nd International Symposium on 3D Data Processing, Visualization and Transmission, 2004. 3DPVT 2004.*, pages 293–300, 2004.

[75] G. Pan, X. Zhang, Y. Wang, Z. Hu, X. Zheng, and Z. Wu. Establishing point correspondence of 3D faces via sparse facial deformable model. *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society*, 22, 06 2013.

[76] P. Huber, Z.-H. Feng, J. Kittler, and M. Rätsch. Fitting 3D morphable models using local features. In *2015 IEEE international conference on image processing*, pages 1195–1199, 09 2015.

[77] C. Basso and A. Verri. Fitting 3D morphable models using implicit representations. *Journal of Virtual Reality and Broadcasting*, 4(18), February 2007.

[78] A. Patel and W. Smith. Exploring the identity manifold: Constrained operations in face space. In *2010 European Conference on Computer Vision*, pages 112–125, 09 2010.

[79] Y. Wang and L. Staib. Boundary finding with prior shape and smoothness models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(7):738–743, 2000.

[80] M. Lüthi, C. Jud, and T. Vetter. A unified approach to shape model fitting and non-rigid registration. In *Machine Learning in Medical Imaging*, pages 66–73, 2013.

[81] D. Hirshberg, M. Loper, E. Rachlin, and M. Black. Coregistration: Simultaneous alignment and modeling of articulated 3D shape. volume 7577, pages 242–255, 10 2012.

[82] S. Ploumpis, H. Wang, N. Pears, W. A. P. Smith, and S. Zafeiriou. Combining 3D morphable models: A large scale face-and-head model. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[83] S. Ploumpis, E. Ververas, E. O. Sullivan, S. Moschoglou, H. Wang, N. Pears, W. A. P. Smith, B. Gecer, and S. Zafeiriou. Towards a complete 3D morphable model of the human head. *IEEE transactions on pattern analysis and machine intelligence*, 2020.

[84] C. T. Jin, P. Guillon, N. Epain, R. Zolfaghari, A. van Schaik, A. I. Tew, C. Hetherington, and J. Thorpe. Creating the Sydney York morphological and acoustic recordings of ears database. *IEEE Transactions on Multimedia*, 16(1):37–46, 2014.

[85] A. Myronenko and X. Song. Point set registration: Coherent point drift. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(12):2262–2275, Dec 2010.

[86] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005.

[87] R. R. Richardson, M. A. Osborne, and D. A. Howey. Gaussian process regression for forecasting battery state of health. *Journal of Power Sources*, 357:209–219, 2017.

[88] M. Mahdaviara, A. Rostami, F. Keivanimehr, and K. Shahbazi. Accurate determination of permeability in carbonate reservoirs using Gaussian process regression. *Journal of Petroleum Science and Engineering*, 196:107807, 2021.

[89] K. Thormann, M. Baum, and J. Honer. Extended target tracking using gaussian processes with high-resolution automotive radar. In *2018 21st International Conference on Information Fusion*, pages 1764–1770, 2018.

[90] S. Sun, P. Zhong, H. Xiao, and R. Wang. Active learning with Gaussian process classifier for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 53(4): 1746–1760, 2015.

[91] Y. Li, S. Liu, and L. Shu. Wind turbine fault diagnosis based on Gaussian process classifiers applied to operational data. *Renewable Energy*, 134:357–366, 2019.

[92] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, 2006.

[93] C. Paciorek and M. Schervish. Nonstationary covariance functions for Gaussian process regression. In *Advances in Neural Information Processing Systems*, volume 16. MIT Press, 2003.

[94] J. J. Florentin, M. Abramowitz, and I. A. Stegun. Handbook of mathematical functions. *American Mathematical Monthly*, 73:1143, 1966.

[95] D. Duvenaud. *Automatic model construction with Gaussian processes*. PhD thesis, 11 2014.

[96] C. T. Baker. The numerical treatment of integral equations. 1977.

[97] S. Sommer, F. Lauze, M. Nielsen, and X. Pennec. Sparse multi-scale diffeomorphic registration: The kernel bundle framework. *Journal of Mathematical Imaging and Vision*, 46, 07 2012.

[98] M. Lüthi, C. Jud, and T. Vetter. Using landmarks as a deformation prior for hybrid image registration. In *Pattern Recognition*, pages 196–205, 2011.

[99] M. A. Alvarez, L. Rosasco, N. D. Lawrence, et al. Kernels for vector-valued functions: A review. *Foundations and Trends® in Machine Learning*, 4(3):195–266, 2012.

[100] Z. Chen, J. Fan, and K. Wang. Remarks on multivariate gaussian process. *arXiv preprint arXiv:2010.09830*, 2020.

[101] Z. Chen, B. Wang, and A. Gorban. Multivariate Gaussian and Student$-t$ process regression for multi-output prediction. *Neural Computing and Applications*, 04 2020.

[102] F. Berkenkamp, M. Turchetta, A. P. Schoellig, and A. Krause. Safe model-based reinforcement learning with stability guarantees. In *31st International Conference on Neural Information Processing Systems*, page 908–919, 2017.

[103] H. Dai, N. Pears, and W. Smith. Augmenting a 3D morphable model of the human head with high resolution ears. *Pattern Recognition Letters*, 128:378–384, 2019.

[104] P. J. Besl and N. D. McKay. A method for registration of 3D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.

[105] S. Rusinkiewicz and M. Levoy. Efficient variants of the icp algorithm. In *3rd International Conference on 3-D Digital Imaging and Modeling*, pages 145–152, 2001.

[106] F. Wang and Z. Zhao. A survey of iterative closest point algorithm. In *2017 Chinese Automation Congress (CAC)*, pages 4395–4399, 2017.

[107] J. L. Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9):509–517, sep 1975.

[108] P. Babin, P. Giguère, and F. Pomerleau. Analysis of robust functions for registration algorithms. In *2019 International Conference on Robotics and Automation*, pages 1451–1457, 05 2019.

[109] Y. Chen and G. Medioni. Object modeling by registration of multiple range images. In *1991 IEEE International Conference on Robotics and Automation*, pages 2724–2729 vol.3, 1991.

[110] A. Makadia, A. Patterson, and K. Daniilidis. Fully automatic registration of 3D point clouds. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 1297–1304, 2006.

[111] J. Yang, H. Li, D. Campbell, and Y. Jia. Go-ICP: A globally optimal solution to 3D ICP point-set registration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(11):2241–2254, Nov 2016.

[112] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24:381–395, 1981.

[113] D. Aiger and D. Cohen-Or. 4-points congruent sets for robust pairwise surface registration. *35th International Conference on Computer Graphics and Interactive Techniques*, 27, 08 2008.

[114] H. Le, T.-T. Do, T. Hoang, and N.-M. Cheung. SDRSAC: Semidefinite-based randomized approach for robust point cloud registration without correspondences. 2019.

[115] B. Amberg, S. Romdhani, and T. Vetter. Optimal step nonrigid icp algorithms for surface registration. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, June 2007.

[116] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, 1977.

[117] V. Golyanik, B. Taetz, G. Reis, and D. Stricker. In *Extended Coherent Point Drift Algorithm with Correspondence Priors and Optimal Subsampling*, 03 2016.

[118] G. Yang, R. Li, Y. Liu, and J. Wang. A unified framework for nonrigid point set registration via coregularized least squares. *IEEE Access*, 8:130263–130280, 2020.

[119] J. Ma, J. Zhao, and A. L. Yuille. Non-rigid point set registration by preserving global and local structures. *IEEE Transactions on Image Processing*, 25(1):53–64, Jan 2016.

[120] H. Dai, N. Pears, and W. Smith. Non-rigid 3D shape registration using an adaptive template. pages 48–63, 01 2019.

[121] L. Fang, Z. Sun, and K. Lam. An effective membership probability representation for point set registration. *IEEE Access*, 8:9347–9357, 2020.

[122] Z. Min, J. Liu, L. Liu, and M. Q.-H. Meng. Generalized coherent point drift with multi-variate Gaussian distribution and Watson distribution. *IEEE Robotics and Automation Letters*, 6(4):6749–6756, 2021.

[123] W. Liu, H. Wu, and G. S. Chirikjian. LSG-CPD: Coherent point drift with local surface geometry for point cloud registration. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15273–15282, 2021.

[124] A. Zhang, Z. Min, J. Pan, and M. Q.-H. Meng. Robust and accurate point set registration with generalized Bayesian coherent point drift. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 516–523, 2021.

[125] L. Bai, X. Yang, and H. Gao. Nonrigid point set registration by preserving local connectivity. *IEEE Transactions on Cybernetics*, 48(3):826–835, 2018.

[126] O. Hirose. A Bayesian formulation of coherent point drift. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(7):2269–2286, 2021.

[127] T. Cour, P. Srinivasan, and J. Shi. Balanced graph matching. In B. Schölkopf, J. C. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 313–320. MIT Press, 2007.

[128] C. Schellewald and C. Schnörr. Probabilistic subgraph matching based on convex relaxation. In *Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 171–186, 11 2005.

[129] J. Lee, M. Cho, and K. Lee. Hyper-graph matching via reweighted random walks. *2011 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1633–1640, 06 2011.

[130] Q. Nguyen, A. Gautier, and M. Hein. A flexible tensor block coordinate ascent scheme for hypergraph matching. 04 2015.

[131] K. Pulli. Multiview registration for large data sets. In *Second International Conference on 3-D Digital Imaging and Modeling (Cat. No.PR00062)*, pages 160–168, 1999.

[132] Q.-Y. Zhou, J. Park, and V. Koltun. Open3D: A modern library for 3D data processing. *arXiv:1801.09847*, 2018.

[133] M. E. Tipping and C. M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(3):611–622, 1999.

[134] M. Lüthi, T. Albrecht, and T. Vetter. Probabilistic modeling and visualization of the flexibility in morphable models. In *Mathematics of Surfaces XIII*, pages 251–264, 09 2009.

[135] H. Hontani, T. Matsuno, and Y. Sawada. Robust nonrigid icp using outlier-sparsity regularization. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 174–181, 2012.

[136] K. Li, J. Yang, Y.-K. Lai, and D. Guo. Robust non-rigid registration with reweighted position and transformation sparsity. *IEEE Transactions on Visualization and Computer Graphics*, 25(6): 2255–2269, 2019.

[137] F. Bernard, L. Salamanca, J. Thunberg, A. Tack, D. Jentsch, H. Lamecker, S. Zachow, F. Hertel, J. Goncalves, and P. Gemmar. Shape-aware surface reconstruction from sparse 3D point-clouds. *Medical Image Analysis*, 38, 05 2017.

[138] D. C. Liu and J. Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 45:503–528, 1989.

[139] N. Parde and R. Nielsen. Finding patterns in noisy crowds: Regression-based annotation aggregation for crowdsourced data. In *2017 Conference on Empirical Methods in Natural Language Processing*, pages 1907–1912, 01 2017.

[140] Y. Yan, R. Rosales, G. Fung, R. Subramanian, and J. Dy. Learning from multiple annotators with varying expertise. *Machine learning*, 95(3):291–327, 2014.

[141] P. Groot, A. Birlutiu, and T. Heskes. Learning from multiple annotators with Gaussian processes. In *International Conference on Artificial Neural Networks*, pages 159–164, 06 2011.

[142] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, Apr 2017.

[143] M. Wainwright and M. Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1:1–305, 01 2008.

[144] H. Chui and A. Rangarajan. A new algorithm for non-rigid point matching. In *2000 IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 44–51 vol.2, 2000.

[145] H. Liu, Y.-S. Ong, X. Shen, and J. Cai. When Gaussian process meets big data: a review of scalable gps. *IEEE Transactions on Neural Networks and Learning Systems*, 31(11):4405–4423, Jan. 2020.

Appendix A

Probability and linear algebra

A.1 Gaussian distribution

A Gaussian distribution of a D dimensional vector x is given as

$$\mathcal{N}(x|\boldsymbol{\mu}, \Sigma^2) = \frac{1}{(2\pi)^{D/2}\|\Sigma\|^{1/2}} \exp\{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\} \tag{A.1}$$

where $\mu \in \mathbb{R}^D$ is the mean, $\Sigma \in \mathbb{R}^{D\times D}$ the covariance and $|\Sigma|$ the determinant of Σ.

A.1.1 Marginal and conditional distributions

If $x \sim \mathcal{N}(x|\mu, \Sigma)$ is partitioned as $x = [x_1, x_2]^T$, then

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix} \right) \tag{A.2}$$

and the marginal distributions are

$$x_1 \sim \mathcal{N}(x_1|\mu_1, \Sigma_1)x_1 \sim_2 \mathcal{N}(x_2|\mu_2, \Sigma_2). \tag{A.3}$$

The conditional distributions are

$$x_1|x_2 \sim \mathcal{N}(x_1|\mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(x_2-\mu_2), \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21})x_1 \sim_2 \mathcal{N}(x_2|\mu_2, \Sigma_2). \tag{A.4}$$

A.2 Gaussian Mixture Model

A Gaussian mixture model is a weighted sum of M component Gaussian densities with mean μ_m and covariance Σ_m, given as

$$p(x|\theta) = \sum_{m=1}^{M} \pi_m \mathcal{N}(x|\mu_m, \Sigma_m),$$

where π_m are non-negative weights such that $\sum_m^M \pi_m = 1$.

A.3 Dirac delta function

The Dirac delta function, usually denoted as $\delta(x)$, is given as

$$\delta(x) = \begin{cases} \infty & \text{if } x = 0 \\ 0 & \text{otherwise} \end{cases} \tag{A.5}$$

such that

$$\int_{-\infty}^{+\infty} \delta(x)dx = 1. \tag{A.6}$$

A.4 Kullback-Leibner divergence

Given two distributions $q(x)$ and $p(x)$ the Kullback-Leibner divergence is given as

$$KL(p\|q) = . \int p(x) \ln \left\{ \frac{q(x)}{p(x)} \right\} dx. \tag{A.7}$$

A.5 Relevant matrix groups

Definition A.1 (Orthogonal group)**.** *Let $A \in \mathbb{R}^{D\times D}$. We say that A is orthogonal if $A^T A = I_D$, that is $A^{-1} = A^T$. The set of all orthogonal matrices is called orthogonal group of dimension D and denoted as $O(D)$.*

Definition A.2 (Special Orthogonal group)**.** *Let $A \in O(D)$. We say that A is special orthogonal if $(A) = +1$. The set of all special orthogonal matrices is called special orthogonal group and denoted as $SO(D)$.*

Definition A.3 (Special Euclidean group)**.** *We consider a transformation matrix given as*

$$T = \begin{bmatrix} \Gamma & p \\ 0 & 1 \end{bmatrix}, \tag{A.8}$$

where $\Gamma \in SO(D)$ is a rotation matrix and $p \in \mathbb{R}^D$ is a translation vector. The Special Euclidean group is the set of all transformation groups and denoted as $SE(D)$.

Appendix B

Registration

B.1 Proof for update equations

Below follow the derivations of the update equations in Proposition 5.1, Proposition 5.2 and Proposition 5.3.

B.1.1 Proposition 5.1

With $p(s,t,\theta)$ as defined in (5.15) we take the expectation with respect to $q_2(c,e)$ and $q_{3i}(\varsigma_i)$ and drop the terms non dependent on δ, as they are included in a constant term.

$$\mathbb{E}_{-1}[\log p(x,y,\theta)] = -\frac{1}{2}\delta^T K^{-1}\delta - \frac{1}{2}\sum_{i=1}^{N_T}\frac{1}{\varsigma_i^2}\sum_{j=1}^{N_S}\|s_j - \mathcal{T}_i\|^2 p_{ij} + \mathrm{C}.$$

Manipulating the second term, we are able to obtain

$$\begin{aligned}\sum_{i=1}^{N_T}\frac{1}{\varsigma_i^2}\sum_{j=1}^{N_S}\|s_j - \mathcal{T}_i\|^2 p_{ij} &= \sum_{i=1}^{N_T}\frac{1}{\varsigma_i^2}\|t_i+\delta_i\|^2\nu_i - 2\sum_{j=1}^{N_S}\sum_{i=1}^{N_T}\frac{1}{\varsigma_i^2}p_{ij}s_j^T(t_i+\delta_i) + C\\ &= (\mathbf{t}+\delta)^T\tilde{D}_\nu\tilde{D}_{\varsigma^2}^{-1}(\mathbf{t}+\delta) - 2\mathbf{s}^T\tilde{P}^T\tilde{D}_{\varsigma^2}^{-1}(\mathbf{t}+\delta) + C\\ &= \delta^T\tilde{D}_\nu\tilde{D}_{\varsigma^2}^{-1}\delta - 2\Big(\tilde{D}_\nu^{-1}\tilde{P}\mathbf{s}-\mathbf{t}\Big)^T\tilde{D}_\nu\tilde{D}_{\varsigma^2}^{-1}\delta + C.\end{aligned}$$

Therefore, the expectation becomes

$$\begin{aligned}\mathbb{E}_{-1}[\log p(x,y,\theta)] &= -\frac{1}{2}\delta^T K^{-1}\delta - \frac{1}{2}\delta^T\tilde{D}_\nu\tilde{D}_{\varsigma^2}^{-1}\delta + \Big(\tilde{D}_\nu^{-1}\tilde{P}s - t\Big)^T\tilde{D}_\nu\tilde{D}_{\varsigma^2}^{-1}\delta + C\\ &= -\frac{1}{2}\delta^T(K^{-1}+\tilde{D}_\nu\tilde{D}_{\varsigma^2}^{-1})\delta + \Big(\tilde{D}_\nu^{-1}\tilde{P}\mathbf{s}-\mathbf{t}\Big)^T\tilde{D}_\nu\tilde{D}_{\varsigma^2}^{-1}\delta + C\\ &= -\frac{1}{2}\delta^T\Sigma^{-1}\delta + \Big(\hat{\mathbf{s}}-\mathbf{t}\Big)^T\tilde{D}_\nu\tilde{D}_{\varsigma^2}^{-1}\delta + C,\end{aligned}$$

with $\Sigma = (K^{-1}+\tilde{D}_\nu\tilde{D}_{\varsigma^2}^{-1})^{-1}$ and $\hat{\mathbf{s}} = \tilde{D}_\nu^{-1}\tilde{P}\mathbf{s}$. Finally, we can obtain

$$\mathbb{E}_{-1}[\log p(x,y,\theta)] = -\frac{1}{2}\Big\{\delta - \Sigma\tilde{D}_\nu\tilde{D}_{\varsigma^2}^{-1}(\hat{\mathbf{s}}-\mathbf{t})\Big\}^T\Sigma^{-1}\Big\{\delta - \Sigma\tilde{D}_\nu\tilde{D}_{\varsigma^2}^{-1}(\hat{\mathbf{s}}-\mathbf{t})\Big\} + \mathrm{C}_1,$$

from which we have that $q_1^*(\delta)$ follows a multivariate normal distribution with the following parameters

$$q_1(\delta)^* = \phi(\delta; \Sigma\tilde{D}_\nu\tilde{D}_{\varsigma^2}^{-1}(\hat{S} - T), \Sigma),$$

meaning that

$$\mathbb{E}[\delta] = \Sigma D_\nu D_{\varsigma^2}^{-1}(\tilde{D}_\nu^{-1}\tilde{P}\mathbf{s} - T)$$
$$\text{cov}(\delta) = \Sigma = (K^{-1} + \tilde{D}_\nu\tilde{D}_{\varsigma^2}^{-1})^{-1}.$$

B.1.2 Proposition 5.2

We will take the expectation of the joint with respect to $q_1(\delta)$, $q_{3i}(\varsigma_i)$ and drop the terms not dependent on c and e, obtaining

$$\mathbb{E}_{-2}[\log p(x, y, \theta)] = \sum_{j=1}^{N_S}\left[\log\left(\{wp_{out}(s_j)\}^{1-c_j}\left\{\frac{1-w}{N_T}\right\}^{c_j}\right) + \sum_{i=1}^{N_T}\log\{\langle\phi_{ij}\rangle\}^{c_j\gamma_i(e_j)}\right] + C,$$

where we defined $\langle\phi_{ij}\rangle = \exp E[\log(\phi_{ij})]$, computed from the other steps and assumed known at this point. As the variance is only included in $\langle\phi_{ij}\rangle$, the computation of p_{ij} remains as in [126]

$$p_{ij} = \frac{(1-w)\langle\phi_{ij}\rangle}{N_T w p_{out}(s_j) + (1-w)\sum_{i'=1}^{N_T}\langle\phi_{i'j}\rangle}$$

and we omit the derivation. Finally, we have that

$$\begin{aligned}\langle\phi_{ij}\rangle = \exp\mathbb{E}[\log(\phi_{ij})] &= \exp\left\{\mathbb{E}\left[\log\left(\frac{1}{(\varsigma_i\sqrt{2\pi})^D}\exp\left\{-\frac{\|s_j - \bar{t}_i\|^2}{2\varsigma_i^2}\right\}\right)\right]\right\} \\ &= \frac{1}{(\varsigma_i\sqrt{2\pi})^D}\exp\left\{-\frac{\|s_j - E[\bar{t}_i]\|^2 + \text{tr}(\text{cov}(\delta_i))}{2\varsigma_i^2}\right\} \\ &= \phi_{ij}(s_j; \mathbb{E}[\bar{t}_i], \varsigma_i^2)\exp\left\{-\frac{\text{tr}(\text{cov}(\delta_i))}{2\varsigma_i^2}\right\}.\end{aligned}$$

B.1.3 Proposition 5.3

Similarly to [126] we take $q_{3i}(\varsigma_i)$ as a Delta Dirac function, meaning that we can maximize the lower bound directly in the same way. Taking the expectation w.r.t. $q_1(\delta)$ and $q_2(c, e)$ and drop terms not dependent on ς_i^2, we have

$$\begin{aligned}\mathbb{E}[\log p(x, y, \theta)] &\propto -\frac{D}{2}\sum_{j=1}^{N_S}\sum_{i=1}^{N_T}p_{ij}\log\varsigma_i^2 - \frac{1}{2}\sum_{i=1}^{N_T}\frac{1}{\varsigma_i^2}\sum_{j=1}^{N_S}\Big[\|s_j - E[\bar{t}_i]\|^2 + \text{tr}(\text{cov}(\delta_i))\Big]p_{ij} \\ &\propto -\frac{1}{2}\sum_{i=1}^{N_T}\Big(D\nu_i\log\varsigma_i^2 + \frac{1}{\varsigma_i^2}\beta_i\Big)\end{aligned}$$

where we have defined $\beta_i = \sum_{j=1}^{N_S} p_{ij}\big(\|s_j - \hat{t}_i\|^2 + \text{tr}(\text{cov}(t_i))\big)$, assumed known at this step. In order to maximize the ELBO we can minimize

$$\sum_{i=1}^{N_T}\Big(D\nu_i \log \varsigma_i^2 + \frac{1}{\varsigma_i^2}\beta_i\Big),$$

so we can minimize each of the terms, such that

$$(\varsigma_i^2)^* = \underset{\varsigma_i^2}{\text{argmin}}\, D\log(\varsigma_i^2)\nu_i + \frac{1}{\varsigma_i^2}\beta_i.$$

Taking the derivative and equating to zero we get

$$(\varsigma_i^2)^* = \frac{\beta_i}{D\nu_i}.$$

We can now reformulate this expression, noting that β_i can be written as

$$\begin{aligned}\beta_i &= \sum_{j=1}^{N_S} p_{ij}s_j^T s_j - 2\sum_{j=1}^{N_S} p_{ij}s_j^T\mathbb{E}[\bar{t}_i] + \sum_{j=1}^{N_S} p_{ij}\mathbb{E}[\bar{t}_i]^T\mathbb{E}[\bar{t}_i] + \sum_{j=1}^{N_S} p_{ij}\text{tr}(\text{cov}(t_i))\\ &= [P\text{diag}(S)S]_i - 2\mathbb{E}[\bar{t}_i]^T[PS]_i + \nu_i\mathbb{E}[\bar{t}_i]^T\mathbb{E}[\bar{t}_i] + \nu_i\text{tr}(\text{cov}(t_i))\end{aligned}$$

and so

$$\varsigma_i^2 = \frac{1}{D}\left[\frac{[P\text{diag}(S)S]_i - 2\mathbb{E}[\bar{t}_i]^T[PS]_i}{\nu_i} + \|\mathbb{E}[\bar{t}_i]\|^2 + \text{tr}(\text{cov}(\delta_i))\right].$$