Statistical modeling of face morphology for dermatology and plastic surgery

Florent Jousse

UNIVERSITÉ CÔTE D'AZUR

ÉCOLE DOCTORALE
SCIENCES ET TECHNOLOGIES DE L'INFORMATION ET DE LA COMMUNICATION

Thèse de doctorat

Modélisation statistique de la morphologie du visage pour la dermatologie et la chirurgie plastique

Florent JOUSSE

Inria, équipe Epione, Sophia Antipolis, France

Présentée en vue de l'obtention du grade de docteur en Automatique, traitement du signal et des images d'Université Côte d'Azur, co-dirigée par Xavier Pennec et Hervé Delingette, co-encadrée par Matilde Gonzalez.

Soutenue le 1er Septembre 2022

Devant le jury composé de:

Président	Pierre ALLIEZ	Directeur de recherche à l'Inria, Université Côte d'Azur
Rapporteurs	Lionel REVERET	Chargé de recherche à l'Inria, Université Grenoble Alpes
	Rasmus R. PAULSEN	Professeur associé à l'Université technique du Danemark
	Gerard SUBSOL	Chargé de recherche au CNRS, Université de Montpellier
Examinateur	Miguel A. GONZALEZ BALLESTER	Professeur à l'Université Pompeu Fabra, Barcelone
Co-directeurs	Xavier PENNEC	Directeur de recherche à l'Inria, Université Côte d'Azur
	Hervé DELINGETTE	Directeur de recherche à l'Inria, Université Côte d'Azur
Co-encadrant	Matilde GONZALEZ	Directrice R&D de QuantifiCare
Invité	Arnaud BLETTERER	Ingénieur de recherche chez QuantifiCare

Cifre

QuantifiCare
See Differently

Inria

Titre

Modélisation statistique de la morphologie du visage pour la dermatologie et la chirurgie plastique.

Résumé

Les acquisitions de surfaces tridimensionnelles (3d) jouent un rôle important dans l'évaluation et le suivi des traitements dermatologiques et cosmétiques. Ces surfaces 3d permettent par exemple de quantifier, en volumes, l'effet d'injections d'acide hyaluronique sur les patients, ou encore de mesurer les surfaces de lésions cutanées. Malheureusement, les parties du corps acquises peuvent être altérées par des mouvements non désirés des patients. Par exemple, les expressions faciales et les changements de posture causent des erreurs de volumes importantes. Par ailleurs, l'évaluation des traitements et les annotations cliniques des données 3d prennent du temps, et subissent la variabilité entre les évaluateurs. Par exemple, les études nécessitant l'évaluation du vieillissement de la peau ou du relâchement des tissus sont facilement biaisées par les désaccords entre les évaluateurs. Dans cette thèse nous proposons des méthodes de modélisation statistique pour aborder ces problèmes.

Nous abordons tout d'abord le problème de recalage entre les surfaces 3d acquises, qui ont une connectivité de maillage et un nombre de sommets différents. Le cadre des modèles déformables par processus Gaussiens (GPMMs) permet de recaler des acquisitions 3d de visages avec des expressions neutres mais rencontre des difficultés lorsque l'expression faciale change. Nous proposons donc un nouveau noyau pour les GPMMs, basé sur les distances géodésiques mesurées le long de la surface du maillage, qui permet des déformations plus flexibles et plus réalistes. Par ailleurs, notre formulation du recalage fait usage des moindres carrés pondérés pour sélectionner des parties du visage qui ne seront pas recalées comme les cheveux. Les progrès récents dans l'estimation des distances géodésiques rendent le surcoût en temps de calcul presque négligeable dans la plupart des applications.

Ensuite, nous construisons un modèle statistique de forme pour quantifier et simuler le relâchement de la peau du visage. Nous avons utilisé la régression des moindres carrés partiels (PLSR) pour trouver une relation linéaire entre ce score de relâchement et des caractéristiques géométriques sur la surface du visage. Le potentiel d'interprétabilité de ces modèles est important pour l'analyse de forme sur de petits ensembles de données où la généralisation est difficile à mesurer. Nous proposons donc des techniques de visualisation pour interpréter les paramètres du modèle PLSR. Notre modèle prédit un score de relachement en accord avec les annotations cliniques dans 73% des cas, soit le même ordre de grandeur que la variabilité intra- et inter-évaluateur. La visualisation des variables latentes de notre modèle PLSR montre une déformation cohérente avec les échelles de notation ce qui achève de démontrer la pertinence du modèle.

Enfin, nous proposons d'utiliser un modèle de déformations pour neutraliser les expressions faciales des scans 3d. Malheureusement, dans les modèles existants, les déformations d'expression faciales et d'identité ne sont pas indépendantes ce qui limite leur utilisabilité. En pratique, modifier l'expression faciale a pour effet de modifier aussi l'identité de la personne. Pour résoudre ce problème, nous avons ajouté une pénalité dans l'apprentissage pour maximiser l'orthogonalité des sous-espaces d'expression et d'identité. De cette façon, nous obtenons une quasi-orthogonalité, qui améliore significativement la neutralisation des expressions faciales. En moyenne, sur toutes les expressions la neutralisation avec quasi-orthogonalité produit des maillages 20% plus proche de la vérité terrain par rapport à la neutralisation sans orthogonalité. L'effet de la quasi-orthogonalité est encore plus visible sur les expressions de grande amplitude, comme l'expression bouche ouverte. Nos visualisations montrent des neutralisations très convaincante pour chaque expression.

Mots clés

Modélisation statistique de forme, recalage de formes, modélisation du visage, noyau géodésique, démêlage de représentations, moindres carrés partiels.

Title

Statistical modeling of face morphology for dermatology and plastic surgery.

Abstract

Three-dimensional (3d) scans are essential in evaluating and following dermatological and cosmetic treatments. Unfortunately, the scanned body parts may be altered by artifacts from the reconstruction or unsought motions from the patients. For instance, facial expressions make the 3d face scans improper for measurements. Hence, it is essential to develop methods to standardize the 3d scans to improve the measurement accuracy. Besides, evaluating treatments and labeling 3d scans is time-consuming, prone to errors, and undergoes rater's variability. For example, studies requiring the evaluation of skin aging or tissue sagging are easily distorted by inconsistencies between evaluators. Thus, it would be a significant leap for physicians to have data-driven algorithms that automatically evaluate the treatments. In this thesis, we develop statistical modeling methods for the face to address these problems.

The first step is to register the 3d scans, which have different mesh connectivity and a different number of vertices. The Gaussian Process Morphable Models (GPMMs) framework can register 3d face scans with neutral expressions but struggle when facial expressions change. Hence, we propose a new kernel for GPMMs based on geodesic distances that allow more flexible and realistic deformations. Our new kernel allows for fitting the template mesh toward faces with different facial expressions. Also, our registration formulation uses weighted least squares to select areas such as hair that will not be registered. Furthermore, the recent advances in estimating geodesic distances make the extra computation time cost negligible in most applications.

Then, we built a statistical shape model to quantify the skin sagging on the face. We use Partial Least Square Regression (PLSR) to find a linear relationship between the skin sagging score and geometric features on the face surface. The interpretability potential of linear models such as PLSR makes these methods ideal candidates for shape analysis on small data sets where generalization is essential. Furthermore, we propose visualization techniques to interpret the parameters of the PLSR model. Our jawline sagging model and the rater agree on 73% of the time, which is in the same range as the raters' variability. The visualization of the PLSR latent variables shows that the model has captured jawline sagging deformations that are coherent with the jawline sagging illustrated on the scales used by the physicians.

Finally, we propose to use a morphable face model to neutralize facial expressions from 3d face scans. Unfortunately, in existing models, there is an entanglement between the deformations related to identity changes and those from facial expressions. Consequently, modifying a facial expression also modifies the person's identity, limiting such models' usefulness. We added an orthogonality penalization into the training procedure to address this issue, leading to a quasi-orthogonality between the expression and identity sub-spaces. The quasi orthogonality allows for a better disentanglement of facial expression deformations from face morphology. Averaging on all expressions, the neutralization with quasi-orthogonality produces face meshes that are 20% closer to the ground truth meshes. The effect of quasi-orthogonality is more visible on large amplitude facial expressions, such as opening the mouth. Our visualizations show very convincing facial expression neutralizations.

Key words

Statistical shape modeling, shape registration, face modeling, geodesic kernel, disentangled learning, partial least squares.

Acknowledgements

I would like to thank all the people without whom the success of this thesis would not have been possible, and I would like to apologize in advance if I forget someone.

My first thanks are due to my two advisors who taught me how to be a researcher with a great deal of passion and motivation. Both have greatly contributed to the success of this thesis. Thanks to Hervé Delingette for his support throughout this doctoral thesis, for his thorough proofreading, and for his creative ideas. My thanks also to Xavier Pennec for his strong involvement in the supervision of this PhD thesis, for proofreading the manuscript and taking the time to share his knowledge.

Also, I would like to thank my supervisors at Quantificare, Matilde Gonzalez and Arnaud Bletterer, for their assistance and time spent proofreading my thesis. They also offered helpful advice during the entire process.

Then, I would like to thank my reviewers, Lionel Reveret, Gerard Subsol and Rasmus Paulsen, who have provided a considerable amount of work. They have greatly contributed to improving the manuscript. Thanks also to Pierre Alliez, and Miguel Angel Gonzalez Ballester for leading my thesis committee and for the relevance of our exchanges.

Furthermore, I would like to thank Jean-Philippe Thirion, the head of QuantifiCare, for his numerous encouragements, inspiring scientific ideas, and for establishing the partnership with Inria.

I wish to thank the permanent researchers of our team Epione: Nicholas Ayache, Marco Lorenzi, and Maxime Sermesant. But also Isabelle Strobant for her support and kindness.

I am also grateful to my friends and colleagues at QuantifiCare Simon Henri, Arthur Picon, Xavier Cremasci, Philippe Grange, with whom I have worked on the integration of the algorithms. Thank you to Paul Tourniaire, Buntheng Ly, and Rémi Bernarhd for providing advice for the PhD defense.

Thank you also to all my friends and colleagues from Epione (Yann Thanderwas, Nicolas Guigui, Santiago Smith, Dimitri Hamzaoui, Josquin Harrisson Hind Dadoun, Dimbihery Rabenoro) for the memorable moments we were able to share.

I would also like to thank all my family for their support and encouragement. So I would like to thank my parents Arnaud and Anne-Marie, who have always supported me. Thanks also to my sister Roxanne and her husband Charles. Finally, I would want to take this opportunity to thank Sophie, the person I love and with whom I have shared my life for years. Her support has been invaluable to me during these three years of thesis writing.

Contents

CONFIDENTIAL

Chapter 1
Introduction

1.1 General context of the thesis

In *dermatology* and *aesthetic medicine*, it is necessary to develop reliable and accurate metrology tools to evaluate the condition of the skin and of soft tissues. Indeed, measurements such as volumes, areas, and roughness are used to define medical conditions, thus improving the monitoring of treatments and the relevance of clinical trials. For example, dermatologists may evaluate the severity of psoriasis flares by combining area measurements and severity indices based on the appearance of the plaques [2]. In aesthetic medicine, practitioners use *volume measurements* and *rating scales* to evaluate rejuvenation procedures, dermal fillers, and cosmetic creams [3; 4]. Dermatologists may classify nevi (a.k.a. moles) into cancerous or benign tumors by evaluating numerous criteria accounting for asymmetry, borders' appearance, color, diameter, and evolution [5]. From these examples, we can define two types of measurements: On the one hand, there are *quantitative measurements*, such as volumes, lengths, and areas, that are objective but sometimes are not sufficient to define complex symptoms or biomarkers. On the other hand, there are *qualitative measurements*, such as physicians' descriptions and severity scores, that cover more medical conditions but are more subjective. This thesis aims to improve quantitative and qualitative measurements in dermatology and plastic surgery.

Measuring a body part's volume or area may seem simple, but it turns out to be much less so in practice. Indeed, standard *metrology instruments*, such as rulers, protractors, and graduated cylinders, are not designed and adapted to take measurements on anatomic surfaces that are soft and curved. Moreover, the patient's skin can be painful due to wounds or sores, making it difficult to take physical measurements.

A solution to ease the measurement process consists of taking measurements on medical images instead of directly on the patients [6; 7; 8; 9]. In aesthetic medicine and dermatology, the medical images are 3d acquisitions of patients' body parts, acquired with *3d acquisition system* or 2d photographs. For example, QuantifiCare's LifeViz (fig. 1.1b) is a portable three-dimensional (3d) acquisition system used to acquire faces in 3d in chapter 4. With portable acquisition systems, such as QuantifiCare's LifeViz, acquiring 3d surfaces is comparable to taking photos of the patients, as one would do with a DSLR[1] camera. The difference is that QuantifiCare's LifeViz are calibrated stereo cameras that take two photos each time with different points of view. Then, the 3d information is found by stereo-reconstruction algorithms, making use of the stereo-vision principle. Another example of an acquisition system is FaceScape's dome (fig. 1.1a), which uses 52 cameras and photogrammetry algorithms to obtain high-resolution 3d scans. These 3d representations, also coined *3d scans*, can be used to compute any measure definable on a surface, such as distances, curvatures, and areas.

However, using 3d scans to represent a patient's anatomy opens up many more fascinating possibilities for analysis. In addition to making measurements on the surface, we can also perform *statistics* and *analyses* on the whole surface. Considering the 3d scan in its entirety and centralizing many data allows one to model the effect of treatments or pathologies on

[1] Digital single-lens reflex (DSLR)

(a) Placement of Facescape's cameras in 3d around the subject.

(b) LifeViz ®Mini

Figure 1.1: Examples of 3d acquisition systems. FaceScape's acquisition system is represented on the left (photogrammetry), and QuantifiCare's acquisition system(stereo-photogrammetry) is on the right.

anatomical surfaces by searching for correlations in the data. The resulting *statistical models* or machine learning algorithms can estimate the effect of treatments on a new patient making them more specific, minimizing risks, and potentially increasing their outcomes. Moreover, statistical analyses and models allow for finding relevant biological markers (a.k.a. *biomarkers*) [10; 11; 12] for pathologies and aging, whose importance is paramount in modern medicine. For example, lifespan studies use *aging biomarkers*, such as telomere shortening (fig. 1.2), epigenetic alterations, and loss of proteostasis to estimate biological age and cellular senescence [13; 14; 15]. Some other aging biomarkers are related to the phenotype [1] and the shape of organs. For example, statistical methods on brain shapes allowed finding early biomarkers for Alzheimer's disease [16; 17; 18]. Statistical modeling methods for shape analysis, also known as *statistical shape analysis* methods, form a very active research field with its share of open problems. This thesis also studies the methods to implement statistics between 3d shapes and applications in dermatology and cosmetic surgery. Our work takes place in the context of an industrial/academic partnership between QuantifiCare[2] and Inria[3].

1.2 Implications of this thesis in dermatology and cosmetic surgery

This thesis aims to develop statistical methods to analyze the shapes and deformations of human body parts such as faces. We focused on the face that is a prominent area in aesthetic medicine. We propose statistical tools directly valuable to dermatologists and plastic surgeons. In particular, the statistical shape models (a.k.a. *morphable models*) developed in this thesis have several applications in dermatology and plastic surgery:

[1] Observable characteristics of organisms.

[2] QuantifiCare is a contract research organization, providing support and services for clinical trials in dermatology and aesthetic medicine.

[3] Inria is a French research institute in computer science and applied mathematics bringing together more than 3900 researchers and engineers.

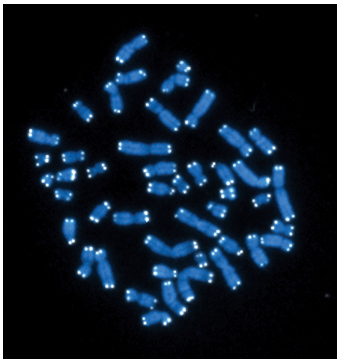


Figure 1.2: The 46 human chromosomes are shown in blue, with the telomeres appearing as white pinpoints. Telomeres shortening is a biomarker for cellular scenescence and organismal aging. Image from national institute of health [19].

The first application of this thesis is the study of *human morphology*. That is to say, the analysis of the variability of body parts (e.g., faces) within a group of people. We can, for example, compute and visualize a mean shape and the most common morphological changes in a dataset. The resulting shape models can be remarkably valuable in developing computerized metrology tools. First, the shape model can be used in *reconstruction algorithms* (e.g., [20]) to improve the robustness to missing parts and reduce the computation time. Secondly, the shape models and registration algorithms developed in this thesis are used by QuantifiCare to automatically define areas of interest and position anatomical landmarks on 3d faces, which eases the workload of technicians in clinical trials.

The second application of this thesis is the automatic quantification of *morphological biomarkers*[1], which helps evaluate the effectiveness of treatments in dermatology and cosmetic surgery. Our work goes in the direction of discovering non-invasive biomarkers of aging that can be used in studies as objective metrics of biological age. For example, we study the jawline sagging that appears with aging. The *jawline* sagging is a morphological aging biomarker used to assess the effect of dermal fillers on patients. The 2021 report on biological markers [21], co-written by Longevity International, Aging Analytics Agency, and Deep Knowledge Group, provides a broad view of the use of biomarkers in medicine and the impact of AI in their development. Most of the time, the evaluation of biomarkers is done by physicians, which is not always reproducible [22; 23; 24]. In contrast, using a statistical model reduces the variability and provides a more consistent evaluation of these morphological biomarkers.

The third application brought by this thesis is the standardization of facial expressions across 3d face scans to make the measurements between 3d scans more precise and relevant. In particular, we use a statistical shape model of facial expressions and postural changes to estimate a patient's neutral face. Here, the motivation is to standardize acquisitions before performing measurements between different acquisitions. For example, the measurement of volume changes between 3d scans is exposed to facial expressions, and it would be a radical advancement to make the measurements invariant to changes of expressions.

[1] morphological biomarkers are observable properties of an organism's morphology. For example, body weight, wrinkles, abdominal fat, or soft tissue ptosis.

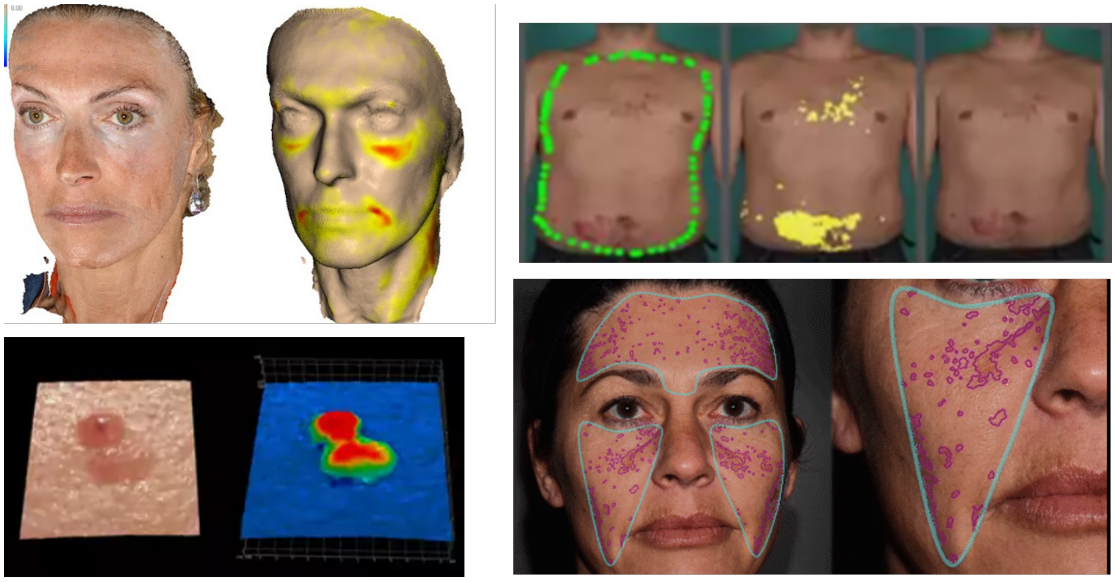


Figure 1.3: Examples of measurements and visualization on 3d scanned shapes (left) and standardized images (rights). On the left: volumes quantification. On the right: area measurements and lesion counts.

1.3 Main research problems

Correspondence problem on QuantifiCare's data QuantifiCare provided 3d face scan datasets that can be used to learn the face morphology and some morphological biomarkers of the face. Unfortunately, these 3d face scans are not in *correspondence*. This means that the number of vertices and the mesh connectivity encoding the surfaces is different across the 3d face scans. The first problem is to solve the correspondence problem. In addition, existing methods such as GPMMs [25] are not adapted to QuantifiCare's data, where the 3d scans can contain hair and clothes. The first question is: *How can we filter out or weight areas that must not be registered?* Moreover, current GPMM kernels does not allow large geometrical changes such as hole openings and closing. Thus, the framework struggles to register facial expressions with mouth openings. So the second question is: *Why GPMMs prevent large deformations such as hole-openings and closings?* and *How to relax the constraint on the deformations?*

Evaluation of morphological biomarkers on 3d scans This thesis also aims to model the interactions between the patient's shape and clinical parameters such as morphological biomarkers. With these models, we can simulate and evaluate *morphological deformations*. For example, we use such models to evaluate the skin sagging on patients' faces, that is a morphological biomarker of aging. A common problem when there is little data available or when the data is annotated by a single rater is that the ground truth is small, not necessarily representative, nor entirely reliable. In most methods, the number of features representing a face is incomparably higher than the number of faces that are available. To reduce the number of *dimensions* representing the data, it is a common practice to use Principal Component Analysis (PCA) [25; 26; 27], and then to perform statistics on the reduced data. With PCA, the latent variables are chosen to retain a maximum of information, and these latent variables can be interpreted

because the transformation between the features and latent variables is linear. However, PCA's latent variables do not maximize the prediction quality. *Is there a method to reduce the number of dimensions that is interpretable and more discriminative for the skin sagging?* Besides, we asked the questions *How much variability is due to inconsistencies between evaluators in skin sagging studies?* and *How to interpret the model's parameters to validate the method?*.

Standardization of 3d face scans In some instances, the facial expression of the person can change between two 3d face scans, which causes errors in QuantifiCare's measurements. We propose to use a face morphable model of expressions and identities to *neutralize* the facial expressions. However, in existing face database models such as Basel Face Model (BFM) [25], Facescape [26], and Facewarehouse [27], there is an entanglement between the subspaces modeling the expressions and the identities. In other words, the two subspaces are not orthogonal. This entanglement prevents from using the models to neutralize facial expressions because modifying expression also modifies the identities. Therefore, the main questions are *Why is there an entanglement between expressions and identities?* and *How to disentangle identities from expressions?*

1.4 Overview of the thesis

In chapter 2: We propose a solution to solve the correspondence problem, which is required to build statistical shape models. Our solution consists in fitting a template mesh to 3d scanned meshes. The fitted template mesh provides coherent features among 3d faces. Some state-of-the-art algorithms to fit a template mesh, such as Non-Rigid Iterative Closest Point (NICP), use specific functions coined kernels to parameterize the allowed template deformations. In this chapter, we propose a new kernel based on *geodesic distances* that can be used in NICP. Our kernel allows more complex deformations of the template, such as widening holes and changing the curvature. We also propose a new formulation of the optimization problem in NICP that makes the template fitting procedure more flexible. We can choose, for example, not to fit the template mesh to hair and clothes.

This work has been published in an international conference FG2021 [28]. All the concepts described in this chapter are implemented by the author in QuantifiCare's C++ library. The non-rigid mesh registration library developed is used in a production setting by QuantifiCare and in this thesis to compute point correspondences between meshes.

In chapter 3: We propose a method to learn the relationship between morphological biomarkers and the shape of a patient face. We applied our methodology to jawline sagging scores used in rejuvenation studies to evaluate the impact of dermal filler. Before our work, the evaluation of jawline sagging was only possible visually. In practice, the physicians compared the patients to a rating scale (e.g., the Merz scale [29]) to give them a score. Our model is currently used as an inference tool to estimate a score of jawline sagging from the patient's 3d scan on a first clinical trial. To a certain extent, this method can reproduce the behavior of an expert and reduce the inherent variability in the given scores.

This work has not been published and is currently under commercial embargo. However, the author implemented the methodology as a C++/Python library and command-line programs used in a clinical trial to assess the effect of dermal fillers on the jawline of two hundred patients.

Except for the graphical interface, the author has written all the code. Other clinical trials based on this work are currently under preparation.

In chapter 4: We study the impact of facial expressions on longitudinal volume measurements, which are also used in clinical trials. First, we show that facial expressions have a substantial impact on the measurement of volume changes[1]. Then, we propose a methodology to neutralize facial expressions from 3d scanned faces using a descriptive model of face morphology and expression. We show that this neutralization is valuable on some facial expressions, such as opening the mouth or smiling. In comparison with state-of-the-art, our method produces *quasi-orthogonal* deformation bases to better separate facial expressions from the person's identity in the shape model. In other words, we can modify facial expressions with less impact on the person's morphology identity than previous methods.

Our methodology to obtain quasi-orthogonal expressions and identity bases in face morphable models will be submitted for publication soon. Currently, the quasi-orthogonalization is implemented in Python, still at the prototypical stage, but will be implemented in QuantifiCare's librairies after this thesis.

[1] QuantifiCare's tools can be used for example to measure volumes changes before and after treatment.

Chapter 2
Solving the correspondence problem

Summary

BUILDING statistical shape models require that the mesh connectivity[1] and the number of vertices remain the same from one 3d scan to another. It is said that the shapes must be in correspondence. A standard approach to solve the correspondence problem is to fit a template mesh to the 3d scans and use the new topology of the template instead of the 3d scan itself. The template fitting can be done using a registration algorithm such as Non-rigid Iterative Closest Point (NICP) and a framework to parameterize the allowed deformations, such as Gaussian Process Morphable Models (GPMMs). In this chapter, we propose a new kernel for the GPMMs to perform better when the topology of the template and the 3d scans change significantly and modifications in NICP to make it more robust to outliers. These adaptations were needed to register QuantifiCare's scans containing outliers such as hair on the reconstructed models.

First, we propose a new kernel based on geodesic distances for the Gaussian Process Morphable Models (GPMMs) framework used in NICP to constrain the allowed deformations. Using geodesic distances allows us to consider the topological and geometric characteristics of the surface (holes, curvature), leading for example to better modeling of facial expressions. Next, we learned the kernel hyperparameters for registering two selected expressions and tested the proposed kernel on the FaceWarehouse dataset. We show that the proposed kernel performs significantly better than Euclidean-based kernels for the task of face registration on all 20 expressions of the dataset.

Second, we propose a modification of the loss function used in the non-rigid ICP registration that allows the weighting of the correspondences according to their confidence. Finally, as a use case, we show that we can make the registration more robust to outliers in the target model, such as non-skin parts.

This work has been published in FG2021 [28]. The template fitting methodology described in this chapter has been implemented as an object-oriented library with both a C++ API and a command-line argument API. This library is used to solve the correspondence problem in QuantifiCare's applications[2] and to build the statistical shape models in the following chapters[3].

2.1 Introduction

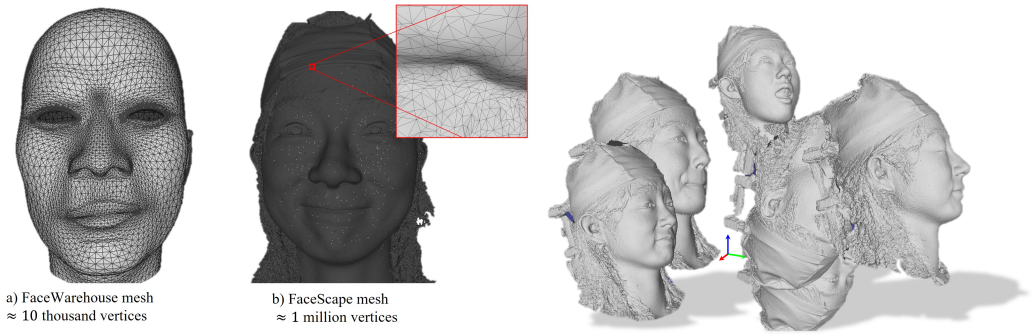
2.1.1 The correspondence problem

Statistical shape models have various applications, such as the synthesis of realistic faces, the modeling of skeletons in anthropology, or the creation of medical atlases. To build such models,

[1] Vertices ids and how the triangles are arranged.

[2] In Quantificare's application, the correspondences between the 3d scanned meshes are used to propagate landmarks and contours from one mesh to another

[3] Building statistical shape models require computing first point correspondences between the 3d scanned meshes.



(a) Different mesh density and mesh connectivity. **(b)** Non spatially registered

Figure 2.1: Examples meshes that are not in correspondence. The meshes may have different topology (number of vertices, mesh connectivity) as in the left figure 2.1a. Or may be misaligned in the space as in the right figure 2.1b. The spatial alignment as well as the mesh connectivity has to be consistent to allow comparison between the shapes.

it is necessary to identify and compare the shapes' characteristics (a.k.a *features*). A simple feature could be the position of points measured on the shape surface. In this case, the feature $\mathbf{f}_i \in \mathbb{R}^3$, is a three-dimensional vector. To implement shape statistics, combining the features into a vector living in a higher dimension space is convenient. If n is the number of features that we have, we can write this high dimensional feature vector $\mathbf{x}_i = (\mathbf{f}_0^\intercal \,|\, \mathbf{f}_1^\intercal \,|\, ... \,|\, \mathbf{f}_n^\intercal)^\intercal$, $\mathbf{x}_i \in \mathbb{R}^{3n}$. Typically in this thesis, the order of magnitude of n will be tens of thousands.

To make the comparison between the features relevant, we must ensure that points chosen on different shapes correspond to the same anatomical location on every shape. For example, a point chosen at the tip of the nose on one face must correspond to a point on the tip of the nose on another face. Because we are looking for features that correspond between the shapes, this problem is called the *correspondence problem*. In figure 2.1, we show examples of 3d scanned meshes from public datasets whose mesh structure (i.e., number of vertices, connectivity) does not allow for establishing correspondences between the vertices. One can notice that the accuracy with which the features are placed on the shapes has an important impact on the relevance of the statistics we can make and, therefore, on the quality of the statistical models we want to build. We consider a dataset of m shapes (i.e., 3d scanned meshes), leading to m feature vectors. We usually encode all these feature vectors into a *data matrix*, that we note $\mathbf{X} \in \mathbb{R}^{3n \times m}$.

The methodology to solve the correspondence problem depends on the nature of the shapes that we observe. The general approach when the shapes are 3d scans [25; 27; 30; 31] is first to fit a reference mesh, commonly referred to as the *template mesh*, towards the 3d scans. Then, use the fitted template mesh instead of the reconstructed meshes to compute features and statistics. The template allows for representing all the shapes with a shared 3d mesh structure, thus solving the problem of correspondences between the features. This chapter discusses how we can fit a face template mesh to a face scanned mesh with the Non-rigid Iterative Closest Point (NICP) algorithm and Gaussian Process Morphable Models (GPMMs) framework. Then, we propose improvements to tackle identified limitations.

2.1.2 State of the art to fit a template mesh

2.1.2.1 Iterative Closest Point based methods

The standard approach to fit a template mesh is derived from the ICP (Iterative Closest Point) algorithm. This algorithm has been initially designed to find a rigid transformation between two point clouds (i.e., fitting a point cloud to another) and a mapping between the two-point sets. The idea of ICP is to make an alternated optimization between the search of the closest points between the point clouds and a minimization of the distance between the estimated corresponding points. The algorithm alternates between transforming the reference point cloud (that reduces the distance) and updating the point correspondences. The problem of finding the transformation minimizing the distance between these two point clouds is called the *registration problem*. It is a well-known problem equivalent to a least-squares problem for rigid transformations.

Many variations of ICP have been proposed to extend the algorithm to more complex shapes (such as meshes) and non-rigid deformations. Early versions [32; 33; 34; 35; 36; 37] allowed only rigid deformations between shapes, which was not sufficient for shape modeling purposes. Therefore, Feldmard et al. [38] proposed to combine affine and locally affine transformations to allow non-rigid motions, with promising results on a wide range of shapes: teeth, faces, skulls, brains, and hearts. Subsol et al. [39; 40] adapted the ICP procedure to compute non-rigid deformations between 3d lines, improving the registration of skulls and brains where crest lines are identifiable. After that, Amberg et al. [41] added a stiffness term to the loss function being optimized that penalizes differences between the locally affine transformation matrices assigned to neighboring vertices.

To make the registration procedure more flexible, Luethi et al. [25; 42] proposed using Gaussian processes to model ICP shape deformations. Their framework, coined Gaussian Process Morphable Models (GPMMs), uses kernels to add priors to the allowed deformations. Thus, one can make the algorithm specific to the characteristics of the surface being registered. This allows for using a specific kernel for different surfaces (solids, soft tissues, bones) and possibly improving the registration. This method has been used, for example, to build the Basel Face Model [25] or the FaceScape model [30], that are generative models of face morphologies.

Non-rigid ICP limitations and solutions The main limitation of the GPMMs is that they may fail when the topology of the 3d scans changes too much. For example, this could be the case when we try to fit a template mesh toward face meshes with different facial expressions introducing holes in the reconstructed model. We hypothesize that the proposed kernels are causing the problem because they are functions of the Euclidean distance between the vertices. Indeed, the template is not a flat surface, which causes the Euclidean distance to be a bad approximation of the actual distance along the surface. Consequently, we propose to use geodesic distances in the kernels to make them more specific to the template's geometrical and topological characteristics.

Another challenge with non-rigid ICP (NICP) methods is to avoid any over-fitting of the template. Indeed, the closest point heuristic guiding the matching of points is not always valid, and the template mesh may easily fit unnaturally on the target mesh. A common practice is to remove the point correspondences according to specific criteria to make the registration more robust. Classic filtering criteria are based on geometric conditions, such as removing the point correspondences whose distances are too large. Instead of removing the point correspondences,

we propose to assign weights to them to allow for more flexible adjustments. For this purpose, we have modified the regression energy function of the NICP algorithm to incorporate the weights, and we describe how the solution can be computed analytically and efficiently. For example, we show that we can use these weights combined with a skin detection algorithm to avoid the template to fit clothes and the hair, which is useful for QuantifiCare's.

2.1.2.2 Alternatives to non-rigid ICP

Other solutions can be considered to find point correspondences between 3d scans. For example, some recent surveys (2013 and 2019) can be found in Tam et al. [43] and Egger et al.[44]. The last [44] distinguishes correspondence computation methods that are done on a subset of landmarks (sparse methods) from those done on the whole surface (called dense methods).

Optical flow methods The first category of methods uses 2d parameterizations of the 3d surface. The idea is to turn the problem of finding corresponding points in 3d into a 2d image matching problem. For example, Blanz et al. [45] proposed to parameterize their 3d meshes by tuples 2d images (depth and color). Then, they used an optical flow estimation algorithm to estimate correspondences between the 2d projections. Such optical flow algorithms are described in [46; 47]. In a more recent context, Cao et al. [27] used a similar registration technique to build the FaceWarehouse morphable model. Indeed, their 3d scanner (the Kinect v2) produces a tuple of a depth map and a colored image. Therefore, the registration in 2d was straightforward. Optical flow (OF) methods are particularly well suited to match two instances of the same person (e.g., with different facial expressions) because the difference is small enough to be found by the OF algorithm.

Spectral methods Some other methods [48] are known as spectral correspondences. The idea is to find correspondences between 3d scans in their frequency domain. First, the scans are transformed into their frequency domain using the spectral decomposition of the Laplace operator. Then, correspondences are found as the closest point on different frequency levels. However, spectral methods are susceptible to missing parts and noise on the mesh as this can produce drastically different spectrums. Thus, spectral methods are not well suited to 3d scans where hair, lights, and clothes will produce holes and other artifacts on the reconstructed surface. For these reasons, spectral methods are generally not used to register 3d face scans or other non-synthetic scans. Nevertheless, in slightly different contexts, Jain et al. [48] showed promising results of non-rigid registration of synthetic meshes of hands, humanoids, and birds.

2.1.3 Chapter organization

In this chapter, we describe the building block algorithms required to fit a template mesh to a 3d scan. In section 2.2 we describe how we can build a deformation basis for the template using Gaussian Process Morphable models. In section 2.3, we introduce the kernels proposed in [42], our new geodesic squared exponential kernel, and its computation. We propose a strategy to optimize the kernel hyperparameters for a specific task. Also, we describe our experiments done on the FaceWarehouse dataset to optimize and test the kernel. In section 2.4 we present the template fitting problem with known point correspondences, which is a simplified version of our problem. We formulated the problem as an optimization problem, and we added weights to the

function being optimized. These weights are used to penalize some point correspondences based on some criterion. Then, we derive the analytical solution to this regression problem. In section 2.5 we describe the problem of fitting a template toward a 3d scan and our solution using a Non-rigid Iterative Closest Point Algorithm (NICP). In section 2.7, we show some registration results achieved with the different kernels. Then, we show a use case of the weighted loss function. Results are finally discussed in section 2.8.

2.2 Building a deformation basis for the template mesh

We have seen in the introduction that the correspondence problem can be solved by fitting a template mesh toward the 3d scans. To do so, we first have to develop a mesh deformation method. More formally, if $\mathbf{x} \in \mathbb{R}^{3n}$ is the vector containing the position of the template's vertices, we want to deform the template by changing the position of every vertex of the template. In our case, a mesh deformation is a linear application $\mathbf{f} : \mathbb{R}^{3n} \to \mathbb{R}^{3n}$ that maps the positions of template's vertices $\mathbf{x} \in \mathbb{R}^{3n}$ to new positions $\mathbf{x} + \mathbf{u}$, where $\mathbf{u} \in \mathbb{R}^{3n}$ is a vector called the *displacement vector*. Note that the displacement vector (i.e., $\mathbf{u}$) has a size $3n$ because we model each dimension in a separate random variable. The main difficulty is that we can not allow the displacement vector $\mathbf{u} \in \mathbb{R}^{3n}$ to be any vector of the Euclidean space $\mathbb{R}^{3n}$ because it would result in unrealistic template deformations.

We can decompose this displacement vector as: $\mathbf{u} = \mathbf{A}\boldsymbol{\alpha} + \boldsymbol{\mu}$, where $\boldsymbol{\mu} \in \mathbb{R}^{3n}$ is an offset and $\mathbf{A} \in \mathbb{R}^{3n \times l}$ is a basis matrix, that spans the space of realistic/possible vertices displacements. Typically, the number of basis vectors l will be much less than the number of vertices on the template $3n$. Therefore, the possible vectors $\mathbf{u}$ are the elements of an affine subspace (a.k.a. linear manifold): $\mathcal{U} = \{\mathbf{A}\boldsymbol{\alpha} + \boldsymbol{\mu} \mid \alpha \in \mathbb{R}^l\}$ called the *deformation space*. In this section, we propose solutions to model the deformation space by adopting a probabilistic point of view.

2.2.1 Morphable models or Point Distribution Models

A possible choice to model the deformation space $\mathcal{U}$ is to assume that the possible vectors $\mathbf{u} \in \mathcal{U}$ are generated by a joint normal distribution: $\mathbf{u} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, where $\boldsymbol{\Sigma} \in \mathbb{R}^{3n \times 3n}$ and $\boldsymbol{\mu} \in \mathbb{R}^{3n}$ are respectively the sample covariance and sample mean estimated from shape examples. This method may be referred as morphable models or Point Distribution Models (PDM). Formally:

> $\mathbf{u} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \iff \exists \boldsymbol{\mu} \in \mathbb{R}^{3n}, \mathbf{A} \in \mathbb{R}^{3n \times 3n}$ such that $\mathbf{u} = \mathbf{A}\boldsymbol{\alpha} + \boldsymbol{\mu}$ and $\boldsymbol{\alpha} \sim \mathcal{N}(\mathbf{0}_{3n}, \mathbf{I}_{3n})$,
> where the covariance matrix is $\boldsymbol{\Sigma} = \mathbf{A}\mathbf{A}^\intercal$.

The non-degenerate case ($m > 3n$): The mean vector $\boldsymbol{\mu}$ and the covariance matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{3n \times 3n}$ can be estimated from the observation of several deformed templates. Given a set of m shapes with matched vertices $\{\mathbf{x}_i\}_{1 \leq i \leq m}$, the sample mean $\boldsymbol{\mu}$ is:

$$\boldsymbol{\mu} = \frac{1}{m} \sum_{i=1}^{m} \mathbf{x}_i, \tag{2.1}$$

and the sample covariance $\mathbf{\Sigma} \in \mathbb{R}^{3n \times 3n}$ is:

$$\mathbf{\Sigma} = \frac{1}{m-1} \sum_{i=1}^{m} (\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^\intercal. \tag{2.2}$$

This formulation allows for generating new shapes by sampling from the distribution with the following steps:

1. Find a matrix $\mathbf{A} \in \mathbb{R}^{3n \times 3n}$ such that $\mathbf{\Sigma} = \mathbf{A}\mathbf{A}^\intercal$. This operation is called the Cholesky decomposition.
2. Let $\boldsymbol{\alpha} = (\alpha_1, ..., \alpha_{3n})^\intercal$ be a random vector, $\boldsymbol{\alpha} \sim \mathcal{N}(\mathbf{0}_{3n}, \mathbf{I}_{3n})$.
3. Let $\mathbf{u} = \boldsymbol{\mu} + \mathbf{A}\boldsymbol{\alpha}$ be a sample from the displacement distribution. Then, the deformed template (i.e. a new shape) is simply the sum $\mathbf{x}+\mathbf{u}$. Some examples of randomly generated samples using different kernels can be seen in figure 2.2.

Intuitively, this means that the distribution of $\mathbf{u} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{\Sigma})$ is the transformed equivalent of the standard normal distribution $\boldsymbol{\alpha} \sim \mathcal{N}(\mathbf{0}_{3n}, \mathbf{I}_{3n})$. In other words, our data has a standard normal distribution in another space.

The degenerate case ($m < 3n$) In the degenerate case, the covariance matrix $\mathbf{\Sigma}$ is singular (because its rank is $m < 3n$), which means that the Cholesky decomposition $\mathbf{\Sigma} = \mathbf{A}\mathbf{A}^\intercal$ is not unique. In this case, it has been proposed in [45] to use the Singular Value Decomposition (SVD), which in this case is simply: $\mathbf{\Sigma} = \mathbf{V}\mathbf{\Delta}\mathbf{V}^\intercal$ because $\mathbf{\Sigma}$ is a symmetric matrix. Here, there will be m non zero singular values, we call $\bar{\mathbf{V}} \in \mathbb{R}^{3n \times m}$ the slice of $\mathbf{V} \in \mathbb{R}^{3n \times 3n}$ corresponding to the non negative zero singular values (and similarly $\bar{\mathbf{\Delta}} \in \mathbb{R}^{m \times m}$ is the subset of the diagonal matrix $\mathbf{\Delta} \in \mathbb{R}^{3n \times 3n}$ with non-zero diagonal entries). Then, we write the distribution as: $\mathbf{u} = \mathbf{V}\mathbf{\Delta}^{\frac{1}{2}}\boldsymbol{\alpha}+\boldsymbol{\mu}$, where $\boldsymbol{\alpha} \sim \mathcal{N}(\mathbf{0}_m, \mathbf{I}_m)$.

Why we cannot use Point distribution models? To use Point distribution models, one must already have a dataset of deformed templates (i.e., meshes in correspondence) to estimate the mean and the covariance, which is not our case. However, we will see in the next section that Gaussian Process Morphable Models offer a solution to the absence of data.

2.2.2 Gaussian Process Morphable Models (GPMMs)

Definition 1. *A Gaussian process (GP) is a stochastic process, i.e. an infinite collection of random variables indexed by time or space. Any finite collection of those random variables must follow a multivariate normal distribution.*

The idea of GPMMs [42] is to define a distribution over the function $\mathbf{u} : \mathbb{R}^3 \to \mathbb{R}^3$ that takes as input the position of a vertex on the template and returns a new position. We denote this using the notation $\mathbf{u}(\mathbf{x}) \sim \mathcal{GP}(\boldsymbol{\mu}(\mathbf{x}), \mathbf{k}(\mathbf{x}_i, \mathbf{x}_j))$, where $\boldsymbol{\mu} : \mathbb{R}^3 \to \mathbb{R}^3$ is a vector valued function, that returns the average displacement at vertex $\mathbf{x}$ and $\mathbf{k} : \mathbb{R}^3 \times \mathbb{R}^3 \to \mathbb{R}^{3\times3}$ is a matrix valued kernel, that encodes the similarity between the displacement of two vertices. The Gaussian process is defined by its *mean function* $\boldsymbol{\mu}(\mathbf{x})$ and *kernel* $\mathbf{k}(\mathbf{x}_i, \mathbf{x}_j)$. Without information from any data, a good choice for the mean function would be the zero function $\boldsymbol{\mu}(\mathbf{x}) = (0, 0, 0)^\intercal$, which represents the idea that, on average, the vertices do not move around the template mesh.

Some assumptions about the distribution of the displacements can be encoded into the kernel. For example, we can integrate spatial correlations between the displacements of neighbouring vertices. Similarly to what is done in the previous section 2.2.1, we can build a set of basis functions from the Gaussian process and generate new shapes with the following steps:

1. First, find the eigenvalues/eigenfunctions λ_i and ϕ_i of the integral operator τ_k evaluated on the domain Ω (i.e: the mesh vertices): $\tau_k f(.) := \int_\Omega k(x,.)f(x)d\rho(x)$. More details about the computation of the eigenfunctions using the Nyström method can be found in section 2.2.2.1 and in Rasmussen's textbook [49].

2. Let $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, ...)^\intercal$ be an infinite random vector, $\boldsymbol{\alpha} \sim \mathcal{N}(0,1)$.

3. Let $\mathbf{u}(\mathbf{x}) = \boldsymbol{\mu}(\mathbf{x}) + \sum_{i=1}^{\infty} \alpha_i \sqrt{\lambda_i}\phi_i(\mathbf{x})$ be a sample from the distribution.

The infinite sum in step 3 can be truncated realizing the following low-rank approximation: $\mathbf{u}(\mathbf{x}) = \boldsymbol{\mu}(\mathbf{x}) + \sum_{i=1}^{\infty} \alpha_i \sqrt{\lambda_i}\phi_i(\mathbf{x}) \approx \sum_{i=1}^{l} \alpha_i \sqrt{\lambda_i}\phi_i(\mathbf{x})$. The low-rank approximation works well when the eigenvalues values are decreasing rapidly since the approximation error is equal to the tail of the sum:$\sum_{i=r+1}^{\infty} \lambda_i$. This formulation provides a model of vertex displacement parameterized by the $\boldsymbol{\alpha} \in \mathbb{R}^L$, which can be used to deform the template mesh.

We can rewrite this parametric model in matrix form, which allows an efficient implementation:

$$\mathbf{u} = \boldsymbol{\mu} + \mathbf{A}\boldsymbol{\alpha}, \tag{2.3}$$

where $\mathbf{A} \in \mathbb{R}^{3n \times l}$ is the matrix containing the evaluation of the eigenfunctions at mesh vertices and form a deformation basis for our template:

$$\mathbf{A} = \begin{pmatrix} \sqrt{\lambda_1}\phi_1(\mathbf{x}_1) & \dots & \sqrt{\lambda_1}\phi_1(\mathbf{x}_N) \\ \vdots & \ddots & \vdots \\ \sqrt{\lambda_r}\phi_r(\mathbf{x}_1) & \dots & \sqrt{\lambda_r}\phi_r(\mathbf{x}_N) \end{pmatrix}, \tag{2.4}$$

$\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_r)^\intercal$ is the vector of parameters and $\boldsymbol{\mu} = [\boldsymbol{\mu}(\mathbf{x_1})^\intercal, \dots, \boldsymbol{\mu}(\mathbf{x_1})^\intercal]^\intercal$

2.2.2.1 Estimation of the eigenfunctions (Nystrom method):

Computing the eigenfunctions directly on all the template vertices may be not computationally tractable due to the high dimension of the matrices involved. Indeed, the Gram matrix $\mathbf{K} \in \mathbb{R}^{3n \times 3n}$ containing the evaluation of the kernel at mesh vertices may be too big to fit in computer's memory. For example a matrix with ten thousands rows and columns, takes up eight gigabytes of memory. Last but not least the eigendecomposition is an $\mathcal{O}(n^3)$ operation, which is computationally expensive.

Thankfully, the computation of the eigenfunctions of the integral operator τ_k can be approximated by the Nyström [50] method by evaluating the kernel on a sufficient number of sample vertices. The idea consists in approximating the integral by a weighted average evaluated at some sample vertices (see equation 2.5). As suggested in [42], we find that sampling a thousand randomly selected vertices on the template (around 1k vertices) leads to a good approximation of the eigenfunctions.

Using l samples vertices $\kappa = (x_1, ..., x_l)$ and replacing the integral by a finite sum, we end up

with the Karhunen–Loève expansion:

$$\lambda_i \phi_i(x') = \int k(x, x')\phi_i(x) d\mu(x) \approx \frac{1}{n} \sum_{l=1}^{n} k(x_l, x')\phi_i(x_l), \tag{2.5}$$

which solution is :

$$\phi_i(x) \approx \frac{\sqrt{n}}{\lambda_i} k_l(x)^\intercal u_i. \tag{2.6}$$

Here u_i is the i-th eigenvector of a matrix $\mathbf{K} \in \mathbb{R}^{3l \times 3l}$ called the Gram matrix, constructed as the following:

$$\underset{N \times N}{\mathbf{k}} = \begin{pmatrix} k(x_0, x_0) & \cdots & k(x_0, x_j) \\ \vdots & \ddots & \vdots \\ k(x_i, x_0) & \cdots & k(x_i, x_j) \end{pmatrix}, \tag{2.7}$$

$$\underset{3n \times 3n}{\mathbf{K}} = \underset{n \times n}{\mathbf{k}} \otimes \underset{3 \times 3}{\mathbf{I}}. \tag{2.8}$$

Here $k_l(x)$ is the covariance matrix between x and all the control points in κ and $\otimes$ denotes the Kronecker product.

Nystrom method with matrix notations The Nystrom method can be seen from a discrete perspective, which is more convenient to implement. In this case, we want to approximate a matrix $\mathbf{K} \in \mathbb{R}^{3n \times 3n}$ by a linear combination of $m \ll 3n$ of its columns. Let $\mathbf{C} \in \mathbb{R}^{n \times m}$ be the sampled columns from $\mathbf{K}$ and rewrite the matrices as the following block matrices:

$$\mathbf{C} = \begin{bmatrix} \mathbf{W} \\ \mathbf{S} \end{bmatrix}, \ \mathbf{K} = \begin{bmatrix} \mathbf{W} & \mathbf{S}^\intercal \\ \mathbf{S} & \mathbf{B} \end{bmatrix}, \tag{2.9}$$

where $\mathbf{W} \in \mathbb{R}^{m \times m}$, $\mathbf{S} \in \mathbb{R}^{(n-m) \times m}$, $\mathbf{B} \in \mathbb{R}^{(n-m) \times (n-m)}$. From the Nystrom method (see [51] section 2.1 or [50] section 1.2) we can write :

$$\tilde{\mathbf{K}} = \mathbf{C}\mathbf{W}^{-1}\mathbf{C}^\intercal \tag{2.10}$$

$$\tilde{\mathbf{K}} = \underbrace{\mathbf{C}\mathbf{V}}_{\mathbf{P}} \underbrace{\mathbf{\Sigma}^{-\frac{1}{2}}}_{\mathbf{\Lambda}} \underbrace{\mathbf{V}^\intercal \mathbf{C}^\intercal}_{\mathbf{P}^\intercal}, \tag{2.11}$$

if $\mathbf{W}$ is invertible. This means that the eigendecomposition of $\mathbf{G}$ can be reduced to the eigendecomposition of a smaller matrix $\mathbf{W}$ and a matrix product, which computation time is $\mathcal{O}(m^3 + mnl)$

A trick to speed-up the diagonalization From the definition of the Kronecker product we can see that the eigenvalues/eigenvectors can be computed on the small matrix $\mathbf{k}$ instead of $\mathbf{K}$ and then we retrieve the original eigenvalues/eigenvectors as follows:

$$\begin{aligned} \mathbf{K} &= \mathbf{k} \otimes \mathbf{I}_3 \\ &= (\mathbf{U}_k \mathbf{\Sigma}_k \mathbf{U}_k^*) \otimes (\mathbf{U}_I \mathbf{\Sigma}_I \mathbf{U}_I^*) \\ &= (\mathbf{U}_k \otimes \mathbf{U}_I)((\mathbf{\Sigma}_k \mathbf{U}_k^*) \otimes (\mathbf{\Sigma}_I \mathbf{U}_I^*)) \\ &= (\mathbf{U}_k \otimes \mathbf{U}_I)(\mathbf{\Sigma}_k \otimes \mathbf{\Sigma}_I)(\mathbf{U}_k^* \otimes \mathbf{U}_I^*) \\ &= \underbrace{(\mathbf{U}_k \otimes \mathbf{U}_I)}_{\mathbf{U}} \underbrace{(\mathbf{\Sigma}_k \otimes \mathbf{\Sigma}_I)}_{\mathbf{\Sigma}} \underbrace{(\mathbf{U}_k \otimes \mathbf{U}_I)^*}_{\mathbf{U}^*}, \end{aligned} \tag{2.12}$$

where $\otimes$ denotes the kronecker product.

2.2.3 Posterior model

We have defined a model of vertices displacements in equation (2.3) that follows a Gaussian process. We call this model the prior model because it is built only using our prior assumptions about the template deformations. In particular, no data is used to build the model. Thanks to the Bayesian formalism, this prior model can be improved by combining it with observed displacements between the template and the 3d scanned mesh. For example, if we knew the eyes' position on the 3D mesh, we would know where the template's eyes should be deformed. This information can be combined with the prior model into a new model called the posterior model.

For a set of h observed landmarks, $\mathbf{y} \in \mathbb{R}^{3h}$ is the column vector containing the displacements of the landmarks between the template and the target. Let $k_*(x)$ be the vector containing the kernel prior values between the test point $\mathbf{x}$ and the h training points. Using theses notations we can write the posterior equations:

$$\boldsymbol{\mu}_p(\mathbf{x}) = \mathbf{k}_*(\mathbf{x})^\intercal [\mathbf{K} + \sigma_n^2 \mathbf{I}]^{-1} \mathbf{y}, \tag{2.13}$$

$$\begin{aligned} \mathbf{k}_p(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{k}(\mathbf{x}_i, \mathbf{x}_j) \\ - \mathbf{k}_*(\mathbf{x}_i)^\intercal [\mathbf{K} + \sigma_n^2 \mathbf{I}]^{-1} \mathbf{k}_*(\mathbf{x}_j). \end{aligned} \tag{2.14}$$

This posterior model can be used in place of the prior model to perform the registration of the template.

2.2.3.1 Drawback of using a posterior model

There is a drawback to using a posterior model. Indeed, the deformation model becomes specific to the target mesh, which makes impossible any factorization of the computations for other template registrations. In contrast, if the deformation model does not depend on the target mesh on which we want to fit the template, we can compute the deformation basis $\mathbf{A} \in \mathbb{R}^{l \times 3n}$ once and reuse it for other template fittings. In practice, we do not use the posterior model for this reason.

2.3 The choice of kernel for GPMMs

2.3.1 B-spline kernels

In the previous section, we have seen how we can turn a Gaussian process into a parametric model and use it to deform a mesh. Moreover, we have seen that the kernel specifies our prior for the distribution of vertices displacements. In the field of face registration, Gerig et al. [25] proposed to use the B-Spline kernel of [52]. This kernel can be written as a sum over the B-spline support parameterized by σ:

$$k_{Sp}(\mathbf{x}_i, \mathbf{x}_j) = \sum_{k \in \mathbb{Z}^d} 2^{2-\sigma} \zeta(2^\sigma \mathbf{x}_i - k) \zeta(2^\sigma \mathbf{x}_j - k), \tag{2.15}$$

where function ζ is built with third order univariate B-spline as: $\zeta(x) = b_3(x_1)b_3(x_2)b_3(x_3)$ and σ is an hyperparameter of the splines which corresponds to the support of the spline. The univariate spline function b_3 is a function with compact support defined as:

$$b_3(x) = \begin{cases} \frac{2}{3} - |x|^2 + \frac{1}{2}|x|^3 & \text{if} \quad 0 \leq |x| < 1, \\ \frac{(2-|x|)^3}{6} & \text{if} \quad 1 \leq |x| < 2, \\ 0 & \text{else.} \end{cases} \tag{2.16}$$

The measure of similarity between the displacement of vertices is controlled by σ, such that increasing σ results in smoother deformations and conversely.

Summing B-splines kernels Several B-splines kernels can be summed together with different values of hyperparameter σ and weight s to create a multiscale kernel:

$$\mathbf{k}_{BSp}(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{I}_3 \sum_{l \in L} s_l k_{Sp}(\mathbf{x}_i, \mathbf{x}_j). \tag{2.17}$$

In [25] the authors proposed to use 4 levels but to the best of our knowledge the weights have not been made public. We will study the impact of multiscale B-spline kernels compared to single scale in 2.6.1

Despite its multiscale property, this kernel may not be used for shape registration when the topology of the meshes changes too much, for instance, during the registration of facial expressions. Indeed, the measure of similarity between the vertices given by this kernel is a function of the Euclidean distance and does not consider the mesh surface's specificities. In other words, this kernel ensures that vertices that are close in Euclidean space $\mathbb{R}^3$ follow similar deformations. Thus, it does not allow the mouth or the eyes to open and close properly. In [25], they worked around the problem of facial expressions by creating facial expressions deformations from examples of manually registered face meshes.

2.3.2 Euclidean squared exponential kernel

Another standard kernel that has been used for face registration and extensively on many other applications involving Gaussian processes is the squared exponential kernel (a.k.a. Gaussian kernel, radial basis function). In our 3d dimensional context, it has the form:

$$\mathbf{k}_{ese}(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{I}_3 \cdot s \cdot \exp\left(-\frac{||\mathbf{x}_i, \mathbf{x}_j||^2}{2\sigma^2}\right), \tag{2.18}$$

where $s \in \mathbb{R}$ is a scaling parameter accounting for the average deviation from the mean (i.e., the standard deviation), and σ^2 is the variance homogeneous to a square length that determines a scale [1] from which the vertices stop behaving similarly.

This kernel has a behavior that is very close to the B-spline kernel. We can see that the kernel's value (i.e., the similarity) decreases exponentially with respect to the distance between $\mathbf{x}_i$ and $\mathbf{x}_j$. Since that distance is Euclidean, the kernel does not account for the surface's singularities (holes, bumps), which causes the kernel to perform poorly when trying to fit a

[1] To be a bit more precise it is an inflection point from which the similarity value starts to drop abruptly.

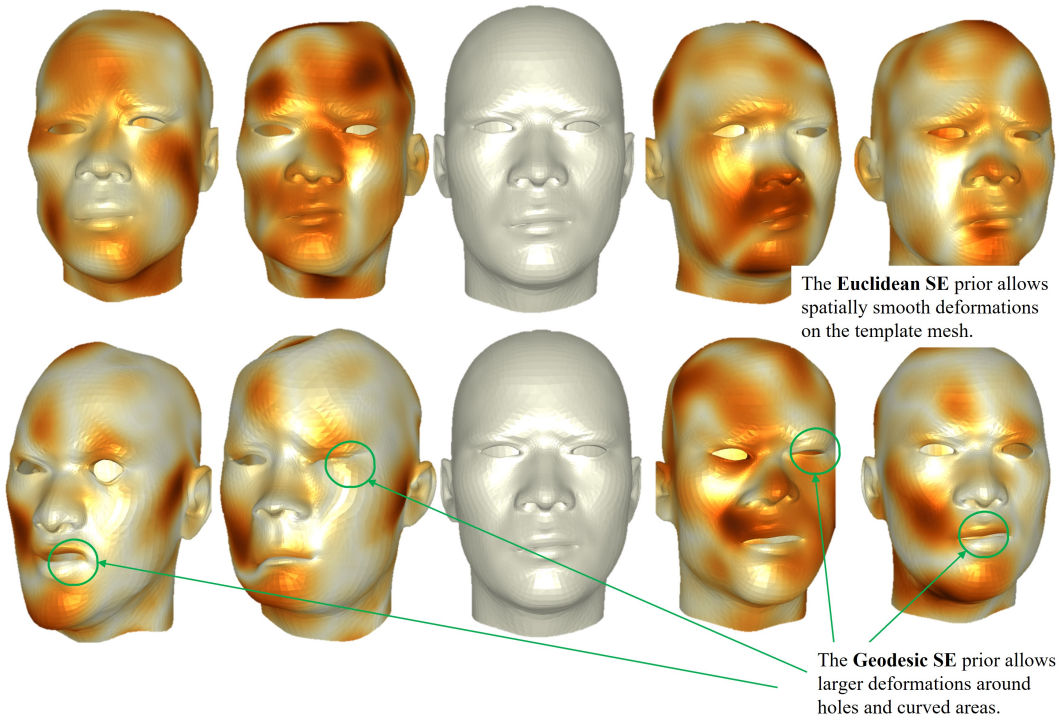


Figure 2.2: Random samples from the Gaussian Process Morphable Model showing more flexible deformations around the eyes and the mouth.

template mesh toward scans with facial expressions, such as opening/closing the eyes or the mouth.

2.3.3 Geodesic squared exponential kernel

As we have seen, the main problem with the kernels proposed with GPMMs (B-spline kernels or Euclidean squared exponential kernel) is that the behavior of vertices displacements is a function of the Euclidean distance. Thus, vertices at equal distance in $\mathbb{R}^3$ behave similarly without any concern about the specificity of the mesh surface.

We propose another kernel based on geodesic distances (i.e., the shortest distance along the mesh surface) to solve this issue. It can be used to improve the results of facial expression registration because it allows mouth openings and closings. Furthermore, this kernel will produce more realistic deformations around holes and curved areas on the template in a more general context. As we can see from Figure 2.3, using geodesic distances reduces the similarity between vertices separated by holes or in high curvature areas. For example, vertices at the top and the bottom of the mouth are more distant with the geodesic distance than with the Euclidean distance. Thus, the kernel becomes more specific to the surface. We will show that this leads to better face registration and more realistic deformations in the section 2.7.

Similarly to the Euclidean squared exponential kernel (Euclidean SE kernel), we can write

the geodesic squared exponential kernel (geodesic SE kernel) as follows:

$$\mathbf{k}_{gse}(x_i, x_j) = \mathbf{I}_3 \cdot s \cdot \exp\left(-\frac{\gamma(\mathbf{x}_i, \mathbf{x}_j)^2}{2\sigma^2}\right), \tag{2.19}$$

with γ being the geodesic distance on the mesh. The kernel has the same parameters as the Euclidean SE kernel. In addition, it is parameterized by the length scale $\sigma \in \mathbb{R}$ which determines the shape of the Gaussian (i.e. larger σ induce smoother deformations) and a scaling factor $s \in \mathbb{R}$. Note that while the support of a B-spline kernel is compact, that is not the case for squared exponential kernels. Furthermore, here the parameter σ corresponds to the variance (as for the SE kernel) and not the support (as for the B-spline kernel). Thus, we expect to use a larger sigma value with B-spline kernels than with squared exponential kernels. Following the definition of multiscale kernels in [25], several geodesic kernels can also be summed together with varying σ to build a multiscale kernel.

Use of geodesic distance based kernels in the literature In the field of surface reconstruction, Del Castillo et al. [53] propose to use a Gaussian process to infer the surface that passes through a point cloud. They have shown that using geodesic distances in the kernel yields better results than using the Euclidean distance. They propose first to search an iso-parameterization $p : S \in \mathbb{R}^3 \longrightarrow T \in \mathbb{R}^2$ of the point cloud and then to compute the geodesic distances as "straight lines" in the new parameterization. Then the surface reconstruction problem is expressed as a kriging problem. The limitations that apply to surface parameterization algorithms are also valid for this method. For example, the algorithm may have trouble with severe curvature areas and sharp edges. Besides, the iso-parameterization does not always exist (for example, for a spherical mesh), while the geodesic distance does. Apart from this, the use of geodesic SE kernels in a Gaussian process has remained limited, and to our knowledge, it has not been used for shape registration.

2.3.3.1 Geodesic distance computation

The geodesic distance can be computed efficiently via the heat method proposed in [54]. The method involves only the resolution of two partial differential equations which is computationally efficient in practice:

1. The first step is to integrate the heat equation for a fixed time t, which means that we are looking for the function $u : \mathbb{R}^3 \to \mathbb{R}$ that is a solution of the following differential equation:

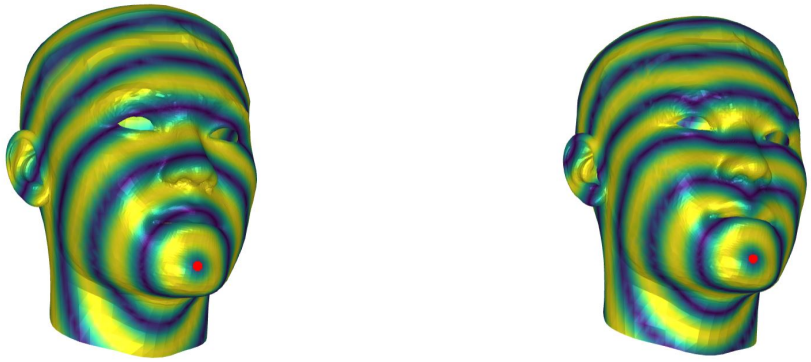
$$\frac{\delta u}{\delta t} = \Delta u, \text{ where } \Delta \text{ is the Laplace operator.} \tag{2.20}$$

2. The second step is to evaluate $\mathbf{X}$ the normalized gradient of our function u:

$$\mathbf{X} = -\frac{\nabla u}{|\nabla u|}. \tag{2.21}$$

3. Finally a Poisson equation is solved to retrieve the geodesic distance:

$$\Delta\gamma = \nabla \cdot \mathbf{X}. \tag{2.22}$$



(a) Euclidean distance **(b)** Geodesic distance

Figure 2.3: This figure present the isolines of distances from a point on the mesh. The points that are on the same isoline are at equal distance from the red point. This shows that holes in the mesh (eg. mouth and eyes) are increasing the distance on some part of the mesh specifically where the isolines bend which means also lower correlation.

Practically, the heat method works on any surface discretization with proper definitions of gradient (∇), divergence ($\nabla\cdot$) and Laplacian (Δ) (see [54] for definitions and computation on simplicial meshes). The geodesic SE kernel can be evaluated on the same domain as the geodesic distances.

Notice that the computation time of geodesic distances is not a limitation because the geodesic distances can be computed once and reused for further registration with the same template. We use an open source implementation that can be found in the IGL library [55].

2.3.3.2 Positive definiteness of geodesic exponential kernels

Many properties, such as Mercer's theorem, are based on the positivity of the kernel. The geodesic SE kernel that we proposed to use is a particular case of the kernels studied in Feragen et al. [56] referred as the geodesic Gaussian kernel (i.e. in our case $q = 2$ in equation (1) of [56]). For this particular case, it has been shown that the kernel is positive definite for all $\sigma > 0$ if and only if the space is flat. Non-positive kernels can produce negative eigenvalues, whose square root leads to complex values in the Karhunen-Loève decomposition (Equation (2.5)). However, negative eigenvalues can be discarded by setting an appropriate L. In practice, no negative eigenvalues have been encountered in our experiments with $L < 1000$ and $10^0 < \sigma < 10^4$.

In NICP, the predictive equations for Gaussian process regression are used to combine the prior kernel with the known displacements between corresponding landmarks. These equations require the inversion of the Gram matrix (see [49] chapter 2). Therefore, in the general case the kernel must have non-zero eigenvalues. But in practice, the observations (i.e. displacements between landmarks) are always noisy. Taking that noise into account in the predictive equations involves adding a constant to the diagonal of the Gram matrix. This addition ensures that we have non-null eigenvalues and thus that the matrix is invertible. See equation (2.13) and section 2.2.2.

2.4 Fitting the template mesh with known point correspondences

2.4.1 Template fitting as a linear regression problem

In the previous sections, we have described how to build a parametric model of deformations, whose allowed deformations are parameterized by choice of a kernel function. We have proposed to use the Geodesic squared exponential kernel and have defined a basis $\mathbf{A} \in \mathbb{R}^{3n \times l}$ for these deformations, where l is the number of modes. Also, we have seen that a template mesh can be encoded as a vector of point positions $\mathbf{x} \in \mathbb{R}^{3n}$.

In this section, we place ourselves in the context where the correspondences between the shapes are known, and we are looking for the transformation that deforms one shape to another. In other words, we search the transformation between two states $\mathbf{x}_1$ and $\mathbf{x}_2$ of the same template mesh. This fitting problem is typically a regression problem, where we search the parameters $\boldsymbol{\alpha} \in \mathbb{R}^l$ of the deformation that minimizes the distance between the two states of the template:

$$\hat{\boldsymbol{\alpha}} = \arg\min_{\boldsymbol{\alpha}} ||\mathbf{x}_2 - \mathbf{x}_1 + \mathbf{A}\boldsymbol{\alpha}||^2. \tag{2.23}$$

The function being optimized is called the regression energy, noted $E : \mathbb{R}^l \to \mathbb{R}$:

$$\mathbf{E}(\boldsymbol{\alpha}) = ||\mathbf{x}_2 - \mathbf{x}_1 + \mathbf{A}\boldsymbol{\alpha}||^2. \tag{2.24}$$

Instead of deriving the solution to this problem, we go straight to a regularized version.

Adding regularization to the regression problem We call regularization the fact of penalizing large values for the deformation's parameters $\boldsymbol{\alpha} \in \mathbb{R}^l$. This penalization, coined ridge regression, gives us a parameter to favor solutions that do not involve large deformations. The regularized optimization writes:

$$\hat{\boldsymbol{\alpha}} = \arg\min_{\boldsymbol{\alpha}} ||\mathbf{x}_2 - \mathbf{x}_1 + \mathbf{A}\boldsymbol{\alpha}||^2 + \rho||\boldsymbol{\alpha}||^2, \tag{2.25}$$

and the regularized energy function:

$$\mathbf{E}(\boldsymbol{\alpha}) = ||\mathbf{x}_2 - \mathbf{x}_1 + \mathbf{A}\boldsymbol{\alpha}||^2 + \rho||\boldsymbol{\alpha}||^2. \tag{2.26}$$

A solution to this optimization problem satisfies a linear system involving the gradient of $\mathbf{E}$:

$$\nabla_\alpha \mathbf{E}(\boldsymbol{\alpha}) = \mathbf{0}. \tag{2.27}$$

We compute the gradient of $\mathbf{E}$ in appendix C, which gives us:

$$\begin{aligned} & \nabla_{\boldsymbol{\alpha}} \mathbf{E}(\boldsymbol{\alpha}) = 0 \\ \Leftrightarrow \quad & 2\mathbf{A}^\intercal\mathbf{y} - 2\mathbf{A}^\intercal\mathbf{A}\boldsymbol{\alpha} - 2\rho\boldsymbol{\alpha} = 0 \\ \Leftrightarrow \quad & 2\mathbf{A}^\intercal\mathbf{A}\boldsymbol{\alpha} + 2\rho\boldsymbol{\alpha} = 2\mathbf{A}^\intercal\mathbf{y} \\ \Leftrightarrow \quad & (\mathbf{A}^\intercal\mathbf{A} + \rho\mathbf{I}_l)\boldsymbol{\alpha} = \mathbf{A}^\intercal\mathbf{y}. \end{aligned} \tag{2.28}$$

We know that $\mathbf{A}$ is orthogonal by construction thus $\mathbf{A}^\intercal\mathbf{A} = \mathbf{I}_l$ and we can simplify the system into:

$$\frac{3n\rho}{l}\mathbf{I}_l\boldsymbol{\alpha} = \mathbf{A}^\intercal\mathbf{y}. \tag{2.29}$$

Finally we can obtain $\boldsymbol{\alpha} \in \mathbb{R}^l$ by simply computing a matrix product.

2.4.2 Penalizing point correspondences in the fitting process

In some cases, we can use information about the quality of point correspondences into the regression problem. Indeed, there are regions of the 3d scans that are more likely to be noisy (e.g. hair, borders, ...) and these regions are not of interest to model the face shape. We propose to use the confidence between the point correspondences such that the more uncertain point correspondences have less impact in the optimization.

Adding weights to the regression problem The confidence weights can be added as a diagonal matrix $\mathbf{W} \in \mathbb{R}^{3n\times 3n}_{>0}$ with strictly positive values, penalizing some rows of the system :

$$\hat{\boldsymbol{\alpha}} = \arg\min_{\boldsymbol{\alpha}} ||\mathbf{W}(\mathbf{x}_2 - \mathbf{x}_1 + \mathbf{A}\boldsymbol{\alpha})||^2 + \rho||\boldsymbol{\alpha}||^2. \tag{2.30}$$

Using again the notation $\mathbf{y} = \mathbf{x_2} - \mathbf{x_1}$ and multiplying by $\frac{1}{2}$, we rewrite the energy function as:

$$\mathbf{E}(\boldsymbol{\alpha}) = \frac{1}{2}||\mathbf{W}(\mathbf{y} - \mathbf{A}\boldsymbol{\alpha})||^2 + \frac{\rho}{2}||\boldsymbol{\alpha}||^2. \tag{2.31}$$

Then we replace the norm with outer products and simplify the energy function as:

$$\begin{aligned}
2\mathbf{E}(\boldsymbol{\alpha}) &= (\mathbf{W}(\mathbf{y} - \mathbf{A}\boldsymbol{\alpha}))^\intercal(\mathbf{W}(\mathbf{y} - \mathbf{A}\boldsymbol{\alpha})) + \rho\boldsymbol{\alpha}^\intercal\boldsymbol{\alpha} \\
&= (\mathbf{W}\mathbf{y} - \mathbf{W}\mathbf{A}\boldsymbol{\alpha})^\intercal(\mathbf{W}\mathbf{y} - \mathbf{W}\mathbf{A}\boldsymbol{\alpha}) + \rho\boldsymbol{\alpha}^\intercal\boldsymbol{\alpha} \\
&= \mathbf{y}^\intercal\mathbf{W}^\intercal\mathbf{W}\mathbf{y} - \mathbf{y}^\intercal\mathbf{W}^\intercal\mathbf{W}\mathbf{A}\boldsymbol{\alpha} - (\mathbf{W}\mathbf{A}\boldsymbol{\alpha})^\intercal\mathbf{W}\mathbf{y} + (\mathbf{W}\mathbf{A}\boldsymbol{\alpha})^\intercal\mathbf{W}\mathbf{A}\boldsymbol{\alpha} + \rho\boldsymbol{\alpha}^\intercal\boldsymbol{\alpha} \\
&= \mathbf{y}^\intercal\mathbf{W}^\intercal\mathbf{W}\mathbf{y} - \mathbf{y}^\intercal\mathbf{W}^\intercal\mathbf{W}\mathbf{A}\boldsymbol{\alpha} - (\boldsymbol{\alpha}^\intercal\mathbf{A}^\intercal\mathbf{W}^\intercal\mathbf{W}\mathbf{y})^\intercal + \boldsymbol{\alpha}^\intercal\mathbf{A}^\intercal\mathbf{W}^\intercal\mathbf{W}\mathbf{A}\boldsymbol{\alpha} + \rho\boldsymbol{\alpha}^\intercal\boldsymbol{\alpha} \\
&= \mathbf{y}^\intercal\mathbf{W}^\intercal\mathbf{W}\mathbf{y} - \mathbf{y}^\intercal\mathbf{W}^\intercal\mathbf{W}\mathbf{A}\boldsymbol{\alpha} - \mathbf{y}^\intercal\mathbf{W}^\intercal\mathbf{W}\mathbf{A}\boldsymbol{\alpha} + \boldsymbol{\alpha}^\intercal\mathbf{A}^\intercal\mathbf{W}^\intercal\mathbf{W}\mathbf{A}\boldsymbol{\alpha} + \rho\boldsymbol{\alpha}^\intercal\boldsymbol{\alpha} \\
&= \mathbf{y}^\intercal\mathbf{W}^\intercal\mathbf{W}\mathbf{y} - 2\mathbf{y}^\intercal\mathbf{W}^\intercal\mathbf{W}\mathbf{A}\boldsymbol{\alpha} + \boldsymbol{\alpha}^\intercal\mathbf{A}^\intercal\mathbf{W}^\intercal\mathbf{W}\mathbf{A}\boldsymbol{\alpha} + \rho\boldsymbol{\alpha}^\intercal\boldsymbol{\alpha}
\end{aligned}$$

Since $\mathbf{W}$ is a diagonal matrix, we use the notation $\hat{\mathbf{W}} = \mathbf{W}^\intercal\mathbf{W}$ that is a also a diagonal matrix whose coefficients are the squared of $\mathbf{W}$'s coefficients:

$$2\mathbf{E}(\boldsymbol{\alpha}) = \mathbf{y}^\intercal\hat{\mathbf{W}}\mathbf{y} - 2\mathbf{y}^\intercal\hat{\mathbf{W}}\mathbf{A}\boldsymbol{\alpha} + \boldsymbol{\alpha}^\intercal\mathbf{A}^\intercal\hat{\mathbf{W}}\mathbf{A}\boldsymbol{\alpha} + \rho\boldsymbol{\alpha}^\intercal\boldsymbol{\alpha}. \tag{2.32}$$

We derive against $\boldsymbol{\alpha}$ and set the gradient to zero :

$$\begin{aligned}
& & \nabla_{\boldsymbol{\alpha}}\mathbf{E}(\boldsymbol{\alpha}) &= 0 \\
&\Leftrightarrow\quad & -\mathbf{y}^\intercal\hat{\mathbf{W}}\mathbf{A} + \boldsymbol{\alpha}^\intercal\mathbf{A}^\intercal\hat{\mathbf{W}}\mathbf{A} + \rho\boldsymbol{\alpha} &= 0 \\
&\Leftrightarrow\quad & \mathbf{A}^\intercal\hat{\mathbf{W}}\mathbf{A}\boldsymbol{\alpha} + \rho\boldsymbol{\alpha} &= \mathbf{A}^\intercal\hat{\mathbf{W}}\mathbf{y} \\
&\Leftrightarrow\quad & (\mathbf{A}^\intercal\hat{\mathbf{W}}\mathbf{A} + \rho\mathbf{I}_l)\boldsymbol{\alpha} &= \mathbf{A}^\intercal\hat{\mathbf{W}}\mathbf{y}.
\end{aligned} \tag{2.33}$$

Finally, the vector of parameter $\boldsymbol{\alpha}$ can be computed by inverting a matrix of size $(l \times l)$:

$$\boldsymbol{\alpha} = (\mathbf{A}^\intercal \hat{\mathbf{W}} \mathbf{A} + \rho \mathbf{I}_l)^{-1} \mathbf{A}^\intercal \hat{\mathbf{W}} \mathbf{y}, \tag{2.34}$$

or by direcly solving the linear system (2.33) using a linear solver such as Eigen [57]. In practice, this modified loss function allows to filter the point correspondences based on some criterion. These weights can be used for example to penalize the point correspondences that are not on the skin and avoid the template to fit clothes and hair. For this, during the registration, we assign to each point pair a weight $w \in \mathbb{R}$ whose value is the probability of being on the skin. To compute such probability we can use the work of Kolkur et al. [58]. We show the effect of weighting the point correspondences in the result section.

Note about gradient descent based solutions In the method [25] to which we compare ourselves, the energy function is optimized through gradient descent (LGBFS [59]) but we decided to solve directly the least square problem analytically. As we have seen, our formulation allows also to regularize the optimization and weighting the correspondences. Using the least squares formulation ensures that the solution is optimal at each iteration.

2.5 Fitting the template mesh toward a 3d scan

2.5.1 Non-rigid Iterative Closest Point (NICP) solves an optimization problem

In the previous section, we have seen how to fit a template mesh toward another state of the same template mesh. In this section, we describe how to fit a template mesh to another mesh with a different mesh topology, such as a 3d scanned mesh. This problem is again an optimization problem, but in this case the dependent variable $\mathbf{y} = \mathbf{x_2} - \mathbf{x_1} \in \mathbb{R}^{3n}$ is also optimized.

Formally we can write the optimization problem:

$$\hat{\boldsymbol{\alpha}}, \hat{\mathbf{y}} = \underset{\boldsymbol{\alpha}, \mathbf{y}}{\arg\min} \overbrace{\frac{1}{2} ||\mathbf{W}(\mathbf{y} - \mathbf{A}\boldsymbol{\alpha})||^2}^{\text{Distance}} + \overbrace{\frac{\rho}{2} ||\boldsymbol{\alpha}||^2}^{\text{Regularization}} \quad . \tag{2.35}$$

y is also refined

The regression energy function being optimized includes the distance between the point correspondences and a regularization term similarly to the function that we have defined previously (equation (2.31)). This problem can be optimized by an alternated optimization called Non-rigid Iterative Closest Point (NICP) [25] where $\mathbf{y} \in \mathbb{R}^{3n}$ and $\boldsymbol{\alpha} \in \mathbb{R}^l$ are alternatively fixed.

- We have already discussed the optimization of $\boldsymbol{\alpha}$ when $\mathbf{y}$ is fixed in the section 2.4.
- On the other hand, optimizing $\mathbf{y}$ when $\boldsymbol{\alpha}$ is fixed is not a well posed problem. Indeed, they are many differences between the shapes that makes the perfect assignment impossible. To quote just a few examples: first, the template mesh has no hair while the 3d scan does, second the 3d scans may have holes or noisy artefacts because of reconstruction problems. The heuristic to solve this part of the optimization problem is to take the

closest points to be the corresponding points. We will see that this heuristic has to be used with precautionary measures.

To sum up, NICP can be described as an iterative procedure that contains the following steps: Firstly, we search corresponding points as the closest points between the reference mesh and the target mesh. Then, the reference mesh is fitted to the target mesh by minimizing a regression energy function which includes the distance between the point correspondences and a regularization term. The iterative procedure is stopped when the regression energy stops to change significantly. It is useful to define this stop criteria as a percentage of energy change from the previous iteration.

One key element of NICP is the search of closest points between the template mesh and the target mesh surface. To speed-up the nearest neighbors search task a commonly used strategy is to partition the 3d search space into a tree-based data structure such as octrees, where each internal node has 8 children. We use a fast implementation to build the octrees that has been proposed by Behley et al. [60] and the computation of nearest neighbors between octrees can be found [61].

Thanks to the octrees we are able to get for each vertex on the template mesh its closest vertex on the target mesh rapidly. However, a better match can be chosen to be the closest point from template vertices to the target mesh surface. That is to say, inside a mesh triangle. To do so, we first get the adjacent faces to the closest vertex. Then we pick the nearest point corresponding to the orthogonal projection of the template vertices to the adjacent faces. This projected point gives us the closest point on the surface of the target mesh rather than the closest vertex.

2.5.2 The elements that makes NICP robust

The described NICP procedure works well if the 3d scans are acquired in a controlled environment but may fail in certain conditions. In fact, NICP is very sensitive to local minima problems. The closest point heuristic is not sufficient by itself to solve the correspondence problem properly and therefore many adjustment has to be performed.

2.5.2.1 Initializing NICP with a similarity transformation

The first element of our NICP algorithm consists in aligning the template mesh toward the target. Some references may call this alignment the Procrustes analysis. The performances of NICP depend on the initial alignment of the meshes because if the two meshes are too far away the optimization may lead to a local minimum which does not correspond to a satisfying registration. This first alignment consists in a similarity transformation (rotation, translation, homogeneous scaling) that minimizes the distance between a set of n pairs of landmarks. Let $\boldsymbol{V}_i \in \mathbb{R}^{3\times n}$ and $\boldsymbol{V}_j \in \mathbb{R}^{3\times n}$ be the 3d positions of the landmarks. We are solving the following optimization problem:

$$\mathbf{R}, \mathbf{t}, s = \arg\min_{\mathbf{R},\mathbf{t},s} ||\mathbf{V}_i - s(\mathbf{R}\mathbf{V}_j + \mathbf{t})||^2, \tag{2.36}$$

where $\mathbf{R} \in \mathbb{R}^{3\times 3}$ is a rotation matrix, $\mathbf{t} \in \mathbb{R}^3$ is a translation vector and $s \in \mathbb{R}$ a scaling factor.

We tested different combinations of landmarks for this alignment. We found that using four landmarks placed at the irises and the mouth corners is the minimum to help NICP not fall into a local minimum. To detect these four landmarks, we used an algorithm implemented in the

QuantifiCare's applications, which detects the eyes by searching for specular reflections of the flash and detects the mouth by fitting an ellipsoid to the mouth in the image domain. Besides, more landmarks can be detected on the face, which may improve the initialization of NICP. For example, Paulsen et al. [62] proposed to use a deep neural network combining multiple modalities (i.e., depth maps, Gaussian curvature, RGB images, mesh renderings) and viewpoints to detect a set of 73 landmarks and pseudo-landmarks. Pseudo-landmarks are points whose position is determined by the position of landmarks. For example, points equally spaced along the jawline are pseudo-landmarks because their positions depend on landmarks at the ear.

2.5.2.2 Coarse-to-fine strategy

In [25], the authors proposed to achieve a coarse-to-fine registration by doing several registrations with different regularization weights. Again, this helps not to fall into a local minimum which does not correspond to a satisfying registration. We achieve similar results by increasing the number of modes that we use in the registration gradually. This allows to avoid the re-computation of Gaussian process (diagonalization of $\mathbf{K}$, model construction, etc.) and makes the optimization faster in the first iterations of NICP because the optimization is done with less parameters. In practice, this coarse-to-fine strategy corresponds to solving several time the optimization problem defined in (2.35). The coarse-to-fine algorithm turns out to:

$$\begin{array}{ll} (\text{iteration} \quad 1 \mid k = 1) & \arg\min_{\boldsymbol{\alpha},\mathbf{y}} \frac{1}{2}||\mathbf{W}(\mathbf{y} - (\boldsymbol{A_{ij}})_{\substack{\mathbf{1}\leq \boldsymbol{i}\leq \boldsymbol{k} \\ \boldsymbol{j}\in\text{columns}}} \boldsymbol{\alpha})||^2 + \frac{\rho}{2}||\boldsymbol{\alpha}||^2, \\ \vdots & \vdots \\ & k := k+1 \\ (\text{iteration} \quad n \mid k = n) & \arg\min_{\boldsymbol{\alpha},\mathbf{y}} \frac{1}{2}||\mathbf{W}(\mathbf{y} - (\boldsymbol{A_{ij}})_{\substack{\mathbf{1}\leq \boldsymbol{i}\leq \boldsymbol{k} \\ \boldsymbol{j}\in\text{columns}}} \boldsymbol{\alpha})||^2 + \frac{\rho}{2}||\boldsymbol{\alpha}||^2, \end{array} \tag{2.37}$$

where $(\boldsymbol{A_{ij}})_{\substack{\mathbf{1}\leq \boldsymbol{i}\leq \boldsymbol{k} \\ \boldsymbol{j}\in\text{columns}}}$ denotes the slice of the matrix $\mathbf{A}$ corresponding to the k first deformation modes.

2.5.2.3 Filtering correspondences

The closest point heuristic may lead to wrong point correspondences. Fortunately, the optimization scheme that we have proposed allows to weight point correspondences and to prevent this kind of problems. The details about the optimization are given in section 2.4. Several strategies have been implemented:

Filtering point correspondences on mesh borders

Figure 2.4: Example of scanned mesh with a hole caused by light reflections on the beard.

In some cases, there may be missing data on the target mesh, for example a hole in the surface. This can happen with poor reconstruction conditions (e.g. poor lighting) as shown on figure 2.4 The point correspondences in this area will tend to stretch the template mesh toward the borders of the holes which is not satisfactory. To make it more robust against missing parts of the surface we added a constraint which consists in removing the point correspondences if it has one end on a border of the mesh. An illustrative example can be found in the figure 2.5.

Filtering point correspondences based on normal orientation Another typical problem is a misalignment of the surfaces after the initialization of NICP. When registering a template to 3d face scans a consequence could be that the two surfaces are not perfectly aligned together (for example noses not being superimposed). In this situation, NICP is likely to find a local minimum where the template nose flatten onto a wrong area such as the jaw. A solution that we used is to remove the point correspondences whose normals are not pointing toward the same direction (see figure 2.6 for an example). In our experiments, we achieve good results with a maximum angle of 25 degrees between the normals.

Mutual neighbors filtering In some cases, the template mesh may have more information than the target mesh. For example, the template mesh may contain bones or inner anatomy while the target mesh does not. Therefore, some vertices on the template mesh will found wrong matches on the target side. To make the correspondences more robust against this problem we ensure that the correspondences are symmetric (a, b) (i.e: if $a \in V$ is the closest point from $b \in W$, then $b \in W$ has to be the closest point from $a \in V$). See figure 2.7 for an illustration.

Skin filtering When fitting a template mesh toward 3d face scans, an important issue can be that the template will fit outside the skin area such as hair and clothes. The high variability in shape and quality in these area makes the modelling impossible. To avoid point correspondences outside the skin area we compute a skin mask on the texture images, then we assign a weight to each point correspondences corresponding to its probability of being on the skin (see [58] for skin detection algorithms). These weights are used in the optimization process through our modified loss function.

Soft assignment In other contexts, such as the registration of noisy point clouds soft assignment has been proposed. The idea is that instead of matching one vertex to an unique match (i.e. hard assignment), we can match a vertex to a few vertices and give to them a probability.

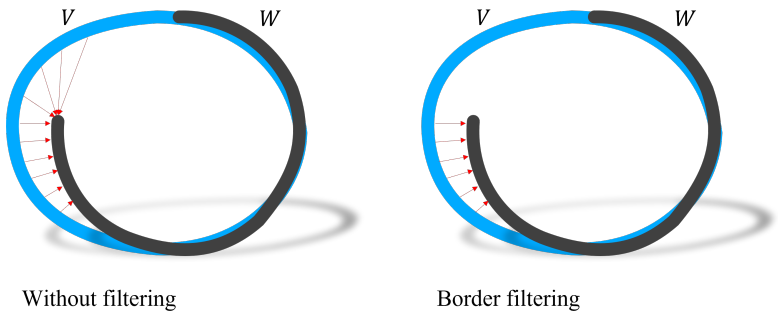


Figure 2.5: This figure shows the effect of filtering the point correspondences that have one end on the border of the mesh. This strategy is suited in the case where the target mesh has missing information such as holes. The red arrows indicates possible point correspondences.

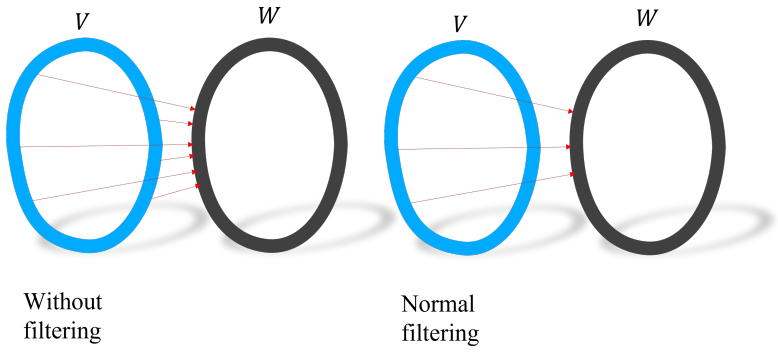


Figure 2.6: This figure shows the effect of adding the filtering of point correspondences based on their normals. This may help if the template V is not well aligned to remove incorrect matchings. Red arrows indicate possible matchings.

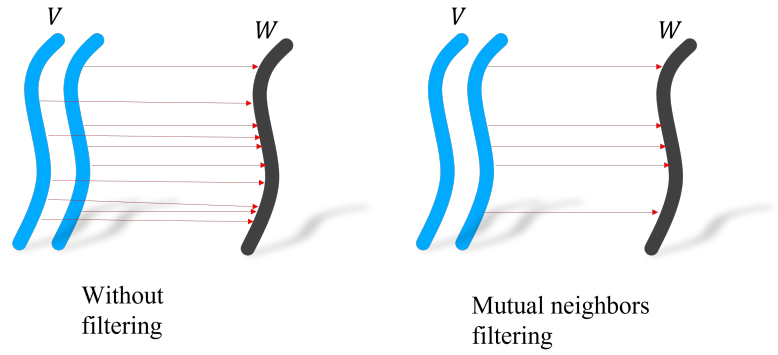
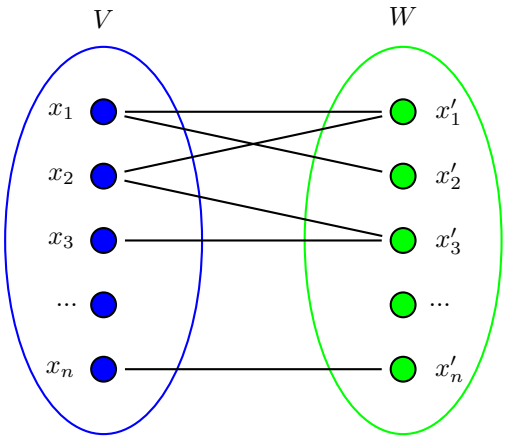


Figure 2.7: This figure shows the effect of adding the mutual neighbors filtering. With Mutual neighbors the closest point has to be the same in the two directions. This is particularly suited when the template mesh V has more information than the target mesh W. Red arrows indicate possible matchings.



Bipartite graph representing the matchings of vertices between the template V in blue and the target W in green.

	x'_1	x'_2	x'_3	...	x'_n	$\sum$
x_1	0.7	0.3	0	⋮	0	1
x_2	0.5	0	0.5	⋮	0	1
x_3	0	0	1	⋮	0	1
⋮	...	...	...	⋱	0	1
x_n	0	0	0	0	1	1
$\sum$	1	1	1	1	1	

Adjacency matrix corresponding to the bipartite graph. The sum of the weight on a row or on a column is equal to 1.

Figure 2.8: This figure shows an example of soft-assignment strategy. One vertex on the template mesh can be associated to several vertices on the target mesh with different probabilities.

If we look at the problem from the graph side this is equivalent to finding a bipartite graph where each side of the graph is a mesh. The weights on the graph are then the probability of the correspondence and the sum of the probabilities or weights on a line of the adjacency matrix has to be equal to 1 (see figure 2.8.).

Soft assignment is well designed when we work with two point clouds and may help with noisy data. For example, it allows to assign a vertex of one side to the n closest vertices on the other side which reduces the probability of finding an outlier. Despite that, this strategy is not really meaningful in our case because we are assigning point clouds to surfaces.

2.6 Experimentation of template fitting with known point correspondences

2.6.1 Tuning of kernel's hyperparameters for face registration

We have seen how we can fit a template mesh to a target mesh using NICP algorithm and the GPMMs framework. We also proposed a new kernel based on geodesic distances that can be used as a prior for face registration. Before the validation of this choice of kernel we propose to optimize the kernels parameters using a few meshes from the FaceWarehouse dataset [27]. This dataset contains three thousand face meshes with 20 different facial expressions. Each face is represented with a template mesh fitted to the 3d scans, thus all the meshes are sharing the same connectivity. In this experiment we assume that the point correspondences between the template mesh and the 3d scan are known. This allows to compare which kernel is a better prior for the registration of faces with expressions without taking into account the assignment part.

The kernels (Geodesic SE kernel, Euclidean SE Kernel and B-spline kernel) described previously have hyperparameters. We do not consider the scale hyperparameter s for SE kernels

because it has no influence on the registration. For all kernels, there is an hyperparameter σ, which determines the deformations. We propose to optimize σ to minimize the regression energy for the fitting of the template mesh toward two selected face meshes from the FaceWarehouse dataset. This optimization can be achieved with the following steps:

1. We compute the mean template $\bar{\mathbf{x}} \in \mathbb{R}^{3n}$ by averaging vertices positions of the 150 faces with neutral expression: $\bar{\mathbf{x}} = \frac{1}{150}\sum_{j=1}^{150} \mathbf{x_j}$.

2. We choose two face meshes with different expression and we build $\mathbf{y}_1 = \mathbf{x}_1 - \bar{\mathbf{x}}$ and $\mathbf{y}_2 = \mathbf{x}_2 - \bar{\mathbf{x}}$, where $\mathbf{x}_1 \in \mathbb{R}^{3n}$ is the vector of vertices position for a face mesh with a neutral expression and $\mathbf{x_2} \in \mathbb{R}^{3n}$ is the vector for a face mesh with the mouth open.

3. We fit the mean template whose vertices positions are $\bar{\mathbf{x}} \in \mathbb{R}^{3n}$ toward the two meshes by solving two optimization problems:

$$\boldsymbol{\alpha_1} = \arg\min_{\alpha} \frac{1}{2}||\mathbf{W}(\mathbf{y}_1 - \mathbf{A}\boldsymbol{\alpha})||^2 + \frac{\rho}{2}||\boldsymbol{\alpha}||^2 \tag{2.38}$$

$$\boldsymbol{\alpha_2} = \arg\min_{\alpha} \frac{1}{2}||\mathbf{W}(\mathbf{y}_2 - \mathbf{A}\boldsymbol{\alpha})||^2 + \frac{\rho}{2}||\boldsymbol{\alpha}||^2, \tag{2.39}$$

   where the deformation basis $\mathbf{A} \in \mathbb{R}^{L\times 3n}$ has been computed with a specific kernel hyperparameter σ. The regularization parameter is set to $\rho = 5.00$.

4. We compute the residual norm $\epsilon_1 = ||\mathbf{y}_1 - \mathbf{A}\boldsymbol{\alpha_1}||^2$, $\epsilon_2 = ||\mathbf{y}_1 - \mathbf{A}\boldsymbol{\alpha_1}||^2$ and $\epsilon = \frac{1}{2}(\epsilon_1 + \epsilon_2)$

5. Then, we iterate over the steps 3 and 4 to find the optimum σ. At each iteration we modify the kernel parameter and recompute a new basis matrix $\mathbf{A}$ for the deformations. Since there is one hyperparameter to be optimised for each kernel a grid search is efficient enough.

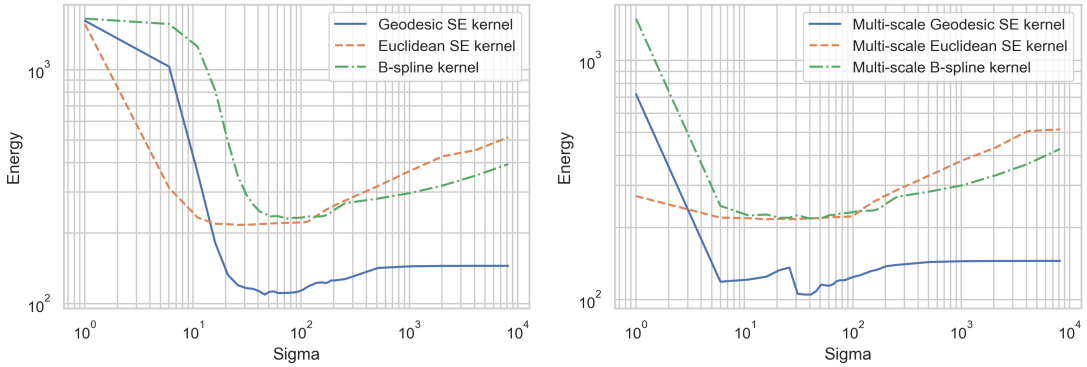
In equation (2.38), the matrix $\mathbf{A} \in \mathbb{R}^{l\times 3n}$ contains the basis vectors[1] of the GPMMs (see equation (2.3)), thus changing the kernel's hyperparameters also changes the matrix $\mathbf{A}$ and therefore the value of the energy after optimization.

To sum up, we searched for each kernel the hyperparameter σ that minimizes the average regression energy ϵ over the registration of two meshes. A mesh with neutral expression and another with mouth open has been used, so that the change of expression is important. For the SE kernels, σ refers to the lengthscale which controls the inflexion point of the Gaussian and for the spline kernels σ is the width of the spline support. Thus, in any case the hyperparameter σ has an impact on the similarity between vertices displacements.

To make the comparison with GPMMs [25] we constructed multiscale kernels by summing several kernels with increasing σ (i.e. $\sigma_n = 2 \times \sigma_{n-1}$). The weights of the kernels are built in the same manner.

Regression scores plots in function of kernels and hyperparameters Our first experiment has been to tune the hyperparameters of each described kernel (SE kernels, B-spline kernels and multiscale kernels). Figure 2.9 shows the plots of energy R against the hyperparameter σ for all kernels. The optimal hyperparameters and their associated energies are:

[1]The number of regressors n is determined by the size of the low rank approximation described in equation (2.5). For the experiments, the bounds of the sum are fixed to $l = 1000$ to keep a fast computation time.



(a) Minimum is reached for $\sigma = 70$ with the Spline kernel, $\sigma = 25$ with the Euclidean kernel and $\sigma = 45$ with the geodesic kernel.

(b) In this plot σ_1 is the lengthscale of the lower level, 4 kernels are summed with σ multiplied by two each time. Minimum is reached for $\sigma_1 = 40$ with the Spline kernel, $\sigma_1 = 11$ with the Euclidean kernel and $\sigma_1 = 35$ with the geodesic kernel.

Figure 2.9: These plots compare the energy of registration in function of the hyperparameter sigma for the 3 kernels discussed (with one scale or mutliple scales). The energy has been computed for two facial expressions (neutral and mouth open). The geodesics squared exponential kernel has an energy two times lower in both cases which indicates a better fit of the surface.

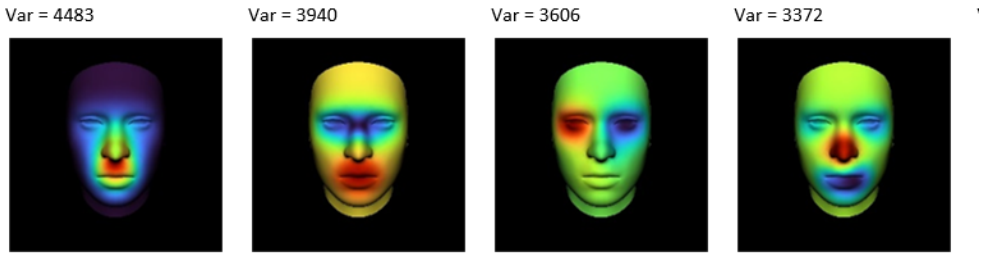
$\sigma = 25, R = 216$ for the Euclidean SE kernel ; $\sigma_1 = 11, R = 216$ for the multiscale Euclidean SE kernel ; $\sigma = 45, R = 109$ for the geodesic SE kernel ; $\boldsymbol{\sigma_1 = 35, R = 104}$ for the geodesic SE kernel ; $\sigma = 70, R = 230$ for the B-spline kernel ; and $\sigma_1 = 40, R = 217$ for the B-spline kernel. The energy of regression is not really meaningful if we look at it in an absolute way. But, it is important to see that the use of geodesic distances reduces the regression energy by a factor of two and that other kernels have similar energy values. The results are discussed in a more thorough way in the section 2.8.

Visualization of the Gram matrix spectrum The visualization of the spectrum of the Gram matrix $\mathbf{K} \in \mathbf{R}^{3n \times 3n}$ whose entries are $K_{ij} = k(x_i, x_j)$ shows the impact of the hyperparameter. For this, we compute the eigenvectors of $\mathbf{K}$ and then we map the coefficients of the eigenvectors to RGB colors that can be displayed on the 3d mesh surface. In this way, two vertices with the same displacement will be displayed with the same color. Figure 2.10 shows that reducing σ leads to more localized deformations on the template mesh.

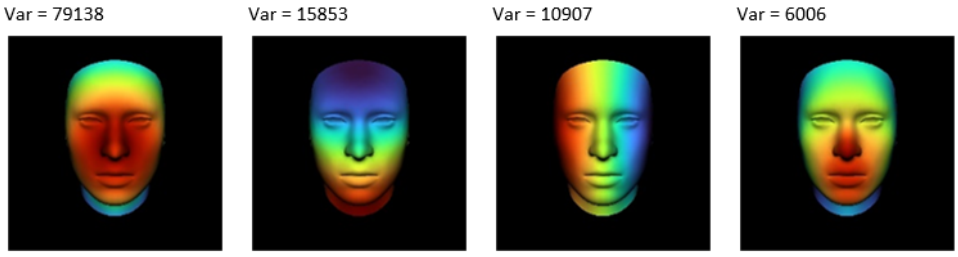
2.6.2 Validation of the geodesic SE kernel on FaceWarehouse dataset

Thereafter, we validated the geodesic SE kernel with the tuned hyperparameters on the whole 3000 samples of the FaceWarehouse dataset. Indeed, it is necessary to validate that the performance improvement provided by using the geodesic SE kernel extends to all faces from the dataset and is not specific to the meshes used to learn the hyperparameter.

For this purpose, we computed for each of the 20 facial expression (called pose in the dataset) the mean and standard deviation of the residuals ϵ of the regression energy over the 150 subject faces. This process is done with the hyperparameters found previously, section 2.6.1. Then, we compared for each facial expression which kernel performs better by looking at the average and



(a) $\sigma = 19$



(b) $\sigma = 150$

Figure 2.10: This figure shows a visualization of the eigenfunctions of the Euclidean squared exponential kernel as a function of the hyperparameter σ. Lower σ leads to more localized eigenfunctions, also the variance associated to the eigenfunction is lower. The color scale shows the displacements such that two vertices with the same color move in the same direction.

standard deviation of the energy ϵ over the 3000 face meshes.

Qualitative comparison of the template registration The qualitative registration results obtained with the geodesic SE kernel, Euclidean SE and the B-spline kernel (single scales) are shown on figure 2.11. The colorscale indicates the error in millimeters. We observe a larger registration error around the mouth and the eyes with the B-spline kernel and the Euclidean SE kernel than with the geodesic SE kernel. The hyperparameters used to generate these results are those obtain using the optimization described in section 2.6.1.

Comparing regressions energy in function of the facial expression Then, we compared the registration results on the whole FaceWarehouse dataset. The mean and standard deviation of registration energy for each facial expression are shown in table 2.12. In these results, the kernel hyperparameters are fixed to the values found in previous section. After averaging on all expressions, the Euclidean SE kernel (single scale) obtain a minimum energy $R = 77(\pm 45)$, the geodesic SE kernel (single scale) obtain a minimum energy $\boldsymbol{R = 54(\pm 5)}$, and the B-spline kernel (single scale) obtain a minimum energy $R = 79(\pm 45)$. We can see that the standard deviation of regression energy is lower with geodesic SE kernel which means more robustness. For example, for the face registration with smile expression, we obtain an energy of $R = 59$ with the geodesic distance compared to $R = 245$ with the Euclidean distance and $R = 245$ with the B-spline kernel.

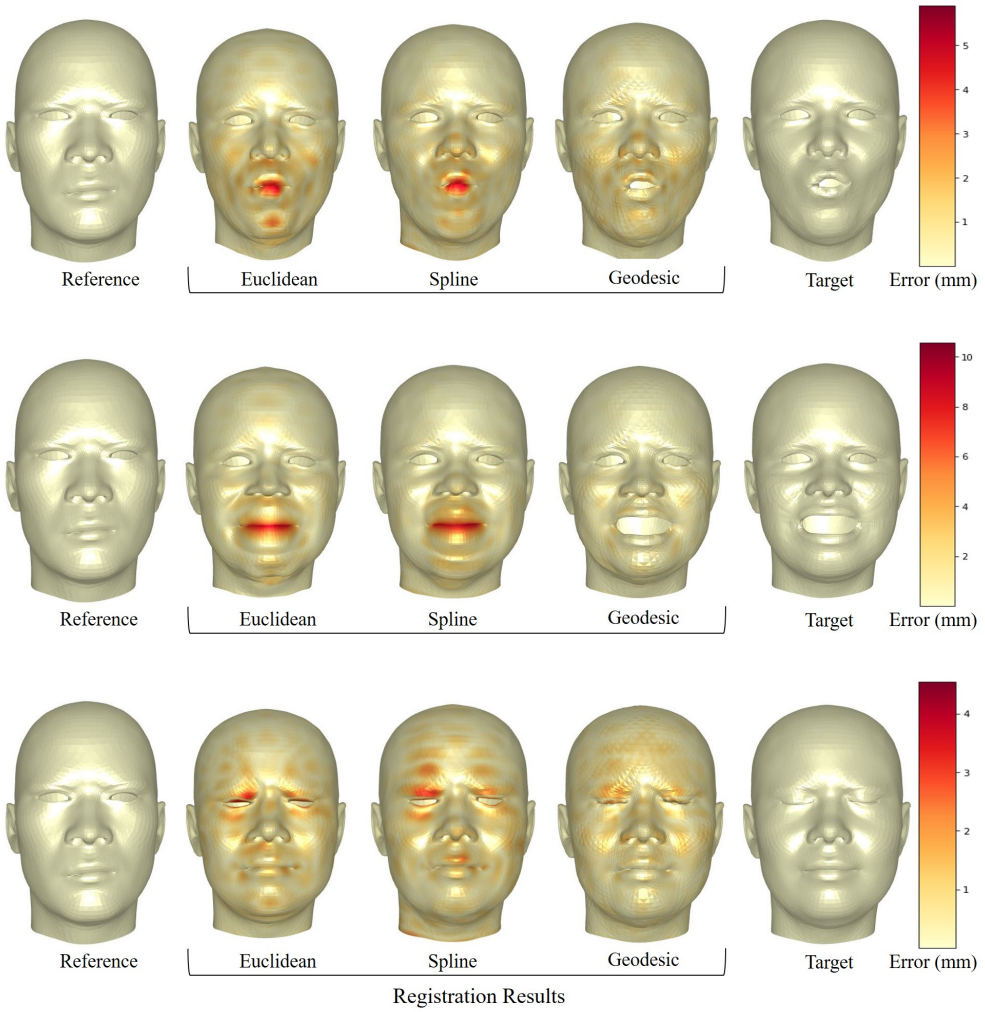


Figure 2.11: This figure shows the registration result of fitting a template (left column) to a target mesh (right column) for 3 faces with expressions. The meshes in the central columns are the results of the registration (i.e. the deformed template) that are obtained with the different kernels. It should be noted that using the geodesics squared exponential kernel results in a better fit (lower error) around the eyes and the mouth. B-spline kernel and Euclidean squared exponential kernel gives similar results.

2.7 Experimentation without knowing the point correspondences

We have already seen that using the geodesic SE kernel yields to better template deformations around the mouth and eyes. We have demonstrated this effect on the problem of fitting a template mesh to another mesh where the point correspondences are known. In some extent, these point correspondences were used as a ground truth for our deformations. In the case where the correspondences are not known it is hard to show quantitative improvements because this

	Euclidean		Geodesic		Spline	
	mean	std	mean	std	mean	std
Pose						
neutral	44.7	7.3	**41.3**	7.7	45.4	7.9
mouth stretch	244.6	39.0	**58.9**	12.7	244.9	39.3
smile	62.8	13.5	**55.9**	13.3	63.5	13.9
brow lower	55.3	12.8	**51.5**	12.7	55.9	13.8
brow raiser	56.5	13.8	**55.5**	14.4	57.4	14.7
anger	57.4	13.6	**53.8**	13.5	57.2	14.7
jaw left	69.0	16.1	**51.6**	13.1	71.7	16.5
jaw right	71.6	19.2	**52.5**	15.5	74.1	19.8
jaw forward	59.8	15.4	**50.3**	14.4	61.2	16.1
mouth left	62.7	12.9	**54.8**	12.1	65.0	13.6
mouth right	64.3	13.7	**54.5**	12.8	65.4	14.5
dimpler	58.9	13.2	**53.7**	12.9	61.5	13.6
chin raiser	58.4	14.1	**50.0**	14.0	60.4	14.8
lip puckerer	66.6	13.2	**55.0**	12.5	70.2	14.6
lip funneler	90.4	18.1	**61.0**	12.6	96.5	18.5
sadness	58.7	13.2	**53.6**	12.5	60.4	13.8
lip roll	65.3	13.0	**64.6**	12.0	74.1	12.8
grin	154.9	26.3	**63.0**	13.7	155.5	26.2
cheek blowing	58.8	16.0	**50.9**	15.8	59.8	16.5
eyes closed	77.4	12.7	**53.8**	13.4	76.0	13.2

Figure 2.12: This table shows the average and standard deviation of registration energy R results for the 20 facial expression of the FaceWarehouse dataset using different kernels. The results have been computed with single scale kernels and tuned hyperparameters. The geodesic SE kernel perform the best in all cases, best results are shown in bold font.

ground truth for the template deformation does not exist anymore. However, we can show some qualitative results of registration. This will allow us to evaluate the impact of using weights in the normal equations (2.33).

A qualitative registration result of our template on a 3d scan is shown in figure 2.14. We used as a target a 3d scan acquired with QuantifiCare LifeViz ® mini. NICP as it is described in the previous section has been used. The left column shows the results of NICP without weighting the point correspondences. The right column shows the result of NICP where the point correspondences are weighted according to a confidence map. The confidence map is shown in figure 2.13 and corresponds to the skin area. The detection of skin area is described in [58]. As it is expected, the registration quality is better when considering weighted point correspondences. For instance, the registration artifacts around the ears have disappeared.

Evaluation of different coarse-to-fine strategies We have seen that the performances of NICP can be improved by adopting a coarse-to-fine registration strategy. This may helps NICP not to fall into a bad local minimum. A coarse-to-fine strategy can be implemented by performing gradual increments of the number of deformation modes used in the registration process (see equation (2.37) for more details). We tested 3 strategies to increment the number of modes used in NICP: linear, exponential and cubic growth. In our case, we expect that increasing gradually the number of modes will improve the robustness of NICP to a bad initial alignment of the

Figure 2.13: This figure shows an example of confidence map generated that can be generated using a skin detection algorithm. The confidence map on the right can be used to weight the point correspondences in NICP. In this case the values are binaries.

template mesh to the target mesh.

To test this hypothesis we fitted a template mesh using 3 different coarse-to-fine strategies toward a set of 3d scans acquired with a QuantifiCare LifeViz ® mini camera. To compare the registration results, we placed manually a set of 14 corresponding landmarks on the template mesh and on our dataset of 3d scans, which allows us to know where the template mesh is supposed to fit the target meshes (the 3d scans). In practice, this allows to compute an error that is the distance between the corresponding landmarks on the template and the target mesh. Of course, the 14 landmarks are not used in NICP in any way. We recall that in this experiment the point correspondences between the template mesh and 3d scans are not known. The template mesh is approximately aligned to the target mesh using their barycenters.

We can see from figure 2.15 that the coarse-to-fines strategies (exponential and cubic) leads to significantly better registration results than using the same number of modes during the whole NICP process (called constant in the figure). More precisely, the registration error is on average two times lower. However, the linear strategy (i.e. constant increment of number of modes) has approximately the same error as when we use the same number of modes during the whole NICP process.

An advantageous side effect of coarse-to-fine strategies is that the computation time is lower, because on the first iterations of NICP the least square problem contains less equations.

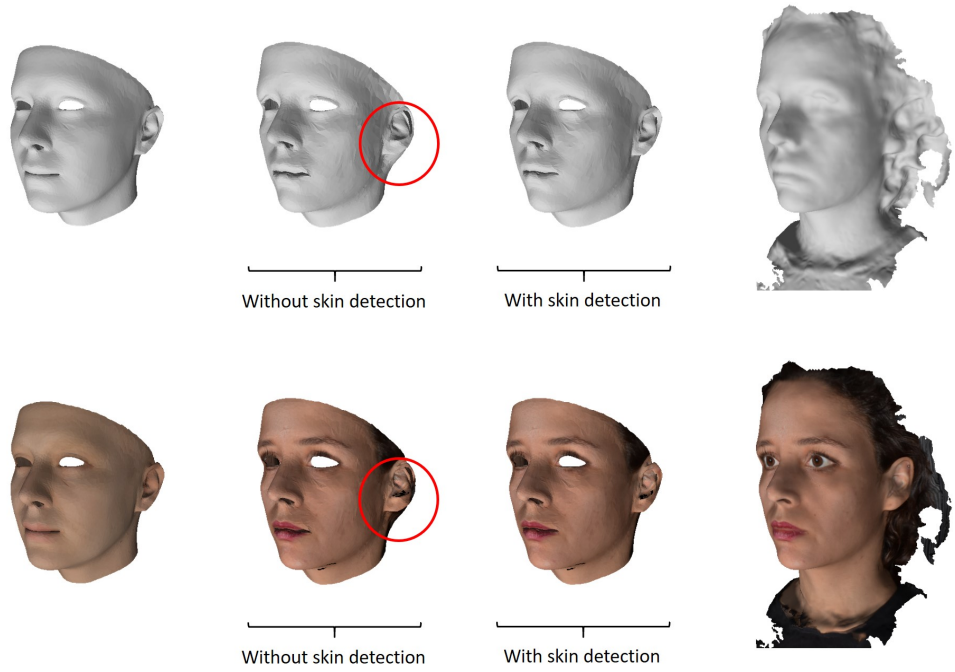


Figure 2.14: This figure shows the registration result of fitting a template (left column) to a 3d scan (right column). The meshes in the central columns are the results of the registration (i.e. the deformed template) that are obtained with and without the weights in the normal equations (2.33). In this case the weights allow to penalize point correspondences that are not on the skin. Without the skin detection and the weights we can see that the template fits the hair in the red area. The 3d scans have been acquired with the QuantifiCare's LifeViz ® mini.

2.8 Further discussions on results

A first observation is that summing multiple kernels with varying variance (i.e. the hyper-parameter sigma) does not improve so much the results of registration. On these data, the multiscale kernels do not demonstrate any real advantage (see figure 2.9) against single scale kernels. Indeed, the kernel shape is mostly determined by the smaller variances and summing kernels has the effect of making the resulting distribution more heavy tailed.

Secondly, B-spline and Euclidean SE kernels tend to have similar regression energy profiles but their energy minima are not reached with the same σ value. The reason is that σ defines the B-spline finite support for B-Splines and the standard deviation for the SE kernel. Therefore, we found smaller σ values for the Euclidean SE kernel.

As expected, the geodesic SE kernel has a much lower registration energy when fitting facial expressions with the mouth opened or the eyes closed. For other facial expressions the geodesic SE kernel performs slightly better than other kernels. There is therefore a real benefit in using geodesic distances in the kernel. The registration energy has an impact on qualitative results. Figure 2.11 shows that the kernel with a geodesic metric is the only one that properly fit the mouth and the eyes (note that the color scale indicates errors in millimeters).

Furthermore, we remark that with small variances ($\sigma < 20$) the error increases sharply.

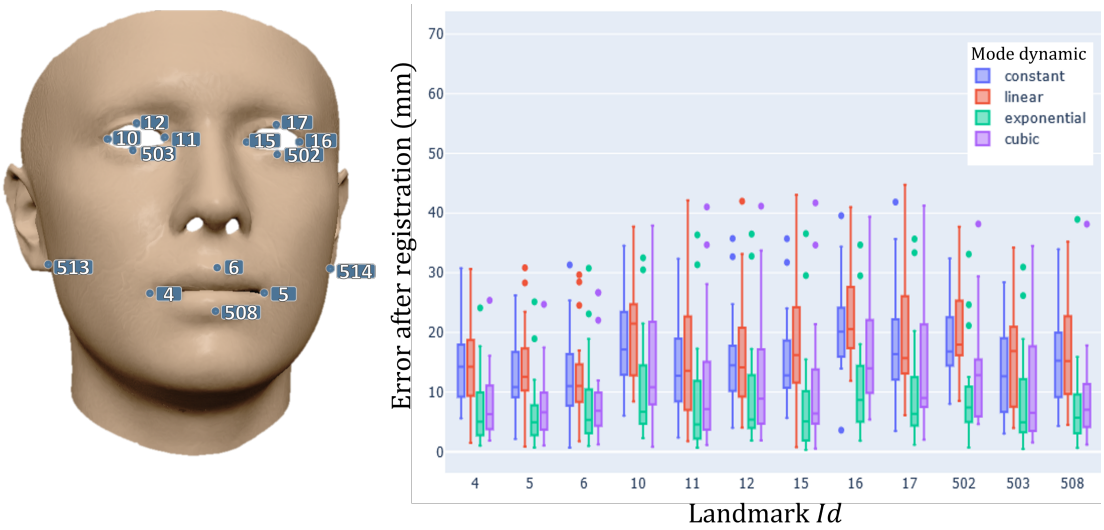


Figure 2.15: In this figure, we evaluate the impact of adopting a coarse-to-fine strategy into NICP. 3 different strategies of mode increments are tested (linear, exponential and cubic) and compared to using all the modes from the first iteration (constant). The registration error is given for 14 different landmarks, that have been placed manually on the meshes of our test set. These landmarks allows us to know where the template mesh is supposed to fit to the target mesh. We observe that two coarse-to-fines strategies (exponential and cubic) have significantly smaller errors registration errors than other strategies (constant and linear).

Indeed, as σ values decrease, the approximation error increases since the number of basis vectors is fixed to a thousand (and the eigenvalues are slowly decreasing with small σ.) Additionally, the energy remains low for a wide range of variance. The approximation error has a direct impact on the face expressivity and thus on the energy.

Finally, the standard deviation of registration errors through the various facial expressions is 45 for the Euclidean SE kernel against 6 for the geodesic SE kernel (see Table 2.12) which means that using the geodesic SE kernels makes the registration more robust with respect to facial expressions.

2.9 Conclusions and future works

The registration of a template mesh towards 3d face scans is still the standard approach to solve the registration problem. We extended the GPMMs framework by proposing a new kernel that takes into account the geodesic distance between mesh points to create a deformation prior. We demonstrated that the GPMMs framework and NICP can now be used to fit a template mesh to 3d face scans with different facial expressions. We also learned the kernel's hyperparameters so that they can be reused for the registration of additional face mesh datasets. We tested the proposed geodesic kernel and the learned hyperparameters for the purpose of registration on the whole FaceWarehouse dataset. Despite the fact that the geodesic SE kernel is not guaranteed to be positive definite, we have shown that it has practical implications and performs significantly better than kernels using Euclidean distances.

Future Work There is certainly a theoretical work to be done on the positive definiteness of geodesic SE kernels. Indeed, it has been shown that in the general case the positive definiteness does not hold and yet geodesic kernels are specifically useful for modelling facial expressions. There is a background work to identify the cases in which we can use rigorously geodesic kernels.

One aspect that has not been studied here is that B-spline kernels generate sparse Gram matrices which is potentially useful for further algorithmic optimizations.

Finally, one may use Laplacian kernels (i.e. the case $q = 1$ in equation (1) of [56]) instead of Gaussian kernels. Indeed, Laplacian kernels are heavy tailed distributions which may replace advantageously the sum of multiple Gaussian kernels.

2.9.1 Summary of contributions

• We introduced a new kernel that can be used in GPMM : the geodesic kernel. This kernel allows more deformations around holes and bumps on the surface. Used in the context of face registration the kernel produce much more realistic deformations around the mouth and the eyes. With this kernel the GPMM framework is now able to handle properly mouth opening and closing.

• We proposed to solve the NICP optimization by solving weighted normal equations, this allows for example to penalize point correspondences which are not on the skin.

• On the industrial side, this work has resulted in a robust mesh deformation library that allows to fit a template mesh toward 3D scanned meshes and compute mesh correspondences. The library has a C++ and command line API. It is now used in QuantifiCare's products and in clinical studies to compute points correspondences between meshes.

Confidential

Chapter 4
Learning shape deformations from longitudinal data

Summary

In aesthetic medicine, measurements on 3d face scans can be used to evaluate patients' responses to medical procedures. For example, we can measure the volume differences between several 3d scans of the same patient before and after injections to evaluate their effects [3; 29; 82; 83]. The problem is that irrelevant factors such as postural movements and facial expressions influence the volume measurements. Therefore, it would be greatly beneficial to minimize them. To this end, the idea is to train a model of face deformation made of two deformation subspaces, representing possible face morphologies and possible facial expressions. In this way, we can either generate synthetic 3d faces or decompose existing ones as the combination of a mean face template with morphological deformations and facial expressions. Once we find the decomposition of a 3d face scan, we can remove the facial expression part to obtain a face with a neutral expression and potentially improve the measurements. This is called expression neutralization.

The objective of our model is to learn deformation spaces encoding the displacements of the template mesh vertices. The deformation space is in our case a linear vector space. A first subspace encodes the possible identity changes while the other encodes the possible facial expressions. It has been proposed in [25; 84] to use two Principal Component Analyses in a joint manner (JPCA). This allows finding a set of basis vectors pointing in the direction of maximum variance when the input shapes are transformed in these low-dimensional subspaces. However, it has been shown [85] that the expression and identity subspaces spanned by these bases are not independent. Consequently, modifying facial expressions will also modify the person's identity. We hypothesized that the independence problem comes from a lack of orthogonality between the two bases. Then, we expressed the learning problem as an optimization problem, also maximizing the orthogonality of the two bases. To solve this problem, we propose an iterative procedure to obtain a quasi-orthogonality between the two subspaces and, thus, a better disentanglement of identities and expressions. Our algorithm, coined Quasi Orthogonal Joint PCA (QOJPCA), solves alternately two eigenvalue/eigenvector problems to obtain a solution, usually converging in a few iterations. Moreover, we provide proof of convergence for QOJPCA.

On the faceScape dataset [26], we show that JPCA produces deformation subspaces (expression and identity) that are not orthogonal. The non-orthogonality increases rapidly with the number of latent variables[1]. Consequently, when we modify the facial expression of a person, it will also modify its identity. In contrast, we can produce quasi-orthogonal deformation subspace with QOJPCA up to 100 latent variables. To demonstrate the usefulness of subspaces orthogonality, we used both models (JPCA and QOJPCA) to perform neutralization experiments. We computed the distance between the face with a neutral expression and the neutralized face on 4000 3d face scans from the FaceScape dataset. On average, we obtain a mean error of 2mm instead of 1.4mm, i.e., a diminution of 30% of the error with QOJPCA. Ensuring better orthog-

[1] Subspace dimension

onality between the expression and identity bases leads to 3d face models that better disentangle expressions from identity.

4.1 Introduction

4.1.1 Context: longitudinal volume measurements

It is a common practice to study volumes in clinical trials. For example, Miehle et al. [86] studied the effect of metrelepting (i.e., a medical drug to cure diabetes) on lipodystrophy, which is a disorder involving an abnormal distribution of fat and some degenerative conditions of adipose tissue. The volume changes measurements were used to assess the effect of the treatment on the adipose tissue distribution. In this study, the authors measured volume changes from **4mL** to **114mL** on the face of patients using 3d cameras. In another study, Mailey et al. [4] studied the facial volume changes after rejuvenation surgeries on nine patients (facelift and fat grafting). The reported volume changes are: $\mathbf{-1mL}\,(\pm\mathbf{0.37})$ on the Nasolabial folds zone, $\mathbf{1.4mL}\,(\pm\mathbf{1.6})$ on the cheeks zone and $\mathbf{0.9mL}\,(\pm\mathbf{4.3})$ on the whole face. These two studies show that in hormonal treatments or surgeries, the volume change measured varies a lot and can be important (up to a hundred milliliters before and after treatment).

For non-invasive procedures, such as the restoration of facial volumes using dermal fillers, the reported volume changes are smaller (in the range of 1 to 5 milliliters [3; 82]). For example, in Kerscher et al. [87] the authors compared the effect of two dermal fillers (hyaluronic acid) on the restoration of cheeks volumes. They used a rating scale (the Merz scale [29]) and volume measurements on 3d face scans. They reported volume changes from **2.52mL** to **3.44mL** before and after the injection, the area of interest is not specified. In another study, Park et al. [83] evaluated the action of injecting **1.0mL** of acid fillers for mid-face volume augmentations.

4.1.2 Impact of facial expressions on volume measurements

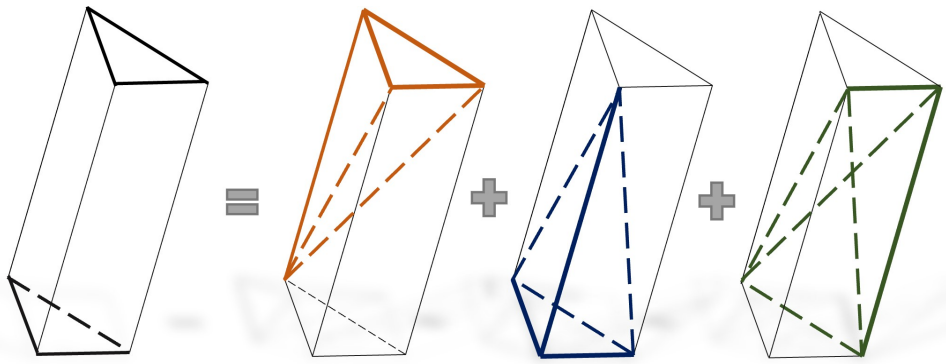


Figure 4.1: The volume between two triangles can be computed by summing the volumes of three tetrahedra.

To evaluate the patients' response to treatments, one can acquire 3d scans of patients at different times with a portable 3d scanner such as LifeViz® cameras. Then, compute the volume between the surface of the 3d scans at different times. To do so, we construct polyhedra between matched points on the 3d scans, which can be found using the Iterative closest point algorithm

(ICP) [32]. The volume measurement accuracy relies, therefore, on the performance of ICP. However, we will see that facial expressions can produce severe volume changes on the face, altering the measurements' values. It can be difficult for a patient to reproduce and maintain a neutral facial expression on different 3d scan acquisitions. Furthermore, we know that postural changes between different 3d acquisitions, even if moderate, impact the volume measurements. However, in this thesis, we focus only on the neutralization of facial expressions because the error produced by postural changes can be partially overcome by registering the faces on fixed areas, such as the forehead. This is not a perfect solution, though, because by registering the faces on the forehead, we shift measurement error from the face area to other areas, such as the neck.

In a preliminary experiment, we estimated the volume changes caused by facial expressions on a public dataset of 3d face scans. To do so, we used the 3d scans of 50 identities with 20 different facial expressions from the FaceScape dataset [26]. A template mesh represents each 3d scan. Consequently, the connectivity and the number of vertices are the same across the 3d scans of the dataset. To compute the volume between two meshes, we used the algorithm implemented in the QuantifiCare applications. This algorithm constructs a set of tetrahedra between the triangles on the 3d scans that "fills" the volume between the 3d scans. As we can see in figure 4.1 the volume of each polyhedron can be computed by summing the volume[1] of 3 tetrahedra, whose volume is known in closed form. We found that this method leads to small errors when the two triangles are intersecting. However, we did not propose any other algorithm because the intersections are made at places where the volume is almost zero, thus generating negligible errors compared to the errors caused by the facial expressions. We show the volume errors caused by the facial expressions of the 50 identities in the following table 4.1. We computed these volumes on several areas of interest on the face. Volumes are in milliliters.

If we consider the whole face area, the volume errors caused by the facial expressions range from 30mL to 144mL, which is way higher than the volumes reported in the studies previously mentioned [3; 29; 82; 83]. Therefore, facial expressions in 3d face scans make the data unsuitable for any volume measurements, at least on the face area. In the mouth area, the errors range from 1.9mL to 14.61mL.

4.1.3 State of the art for facial expression neutralization

In the previous chapters, we have seen that a face template mesh, suitably deformed, can represent any face. Using a template allows for sharing a common mesh structure for all the faces in the dataset and solves the correspondence problem. In particular, we can use a single matrix to encode the mesh connectivity [2] for all the faces in the dataset. Then, we represent the faces as the vector of displacements between the positions of the vertices of the template and the observed face. We note $\bar{\mathbf{v}} \in \mathbb{R}^{3n}$ the vector encoding the vertex positions of the template mesh. Thus, a face is symply encoded by: $\mathbf{u} = \mathbf{v} - \bar{\mathbf{v}}$.

A big challenge with this representation is that the faces are encoded in a space of high dimensions, whose size is three times larger than the number of vertices on the mesh (= 26317 for Facescape meshes). This, combined with the small amount of available data, leads to sparse data points in the ambient space and makes it hard to produce reliable statistics. For this reason, most methods for disentangling facial expressions from facial attributes perform some

[1] The volume is not signed.

[2] The connectivity indicates how vertices are paired to form the mesh triangles.

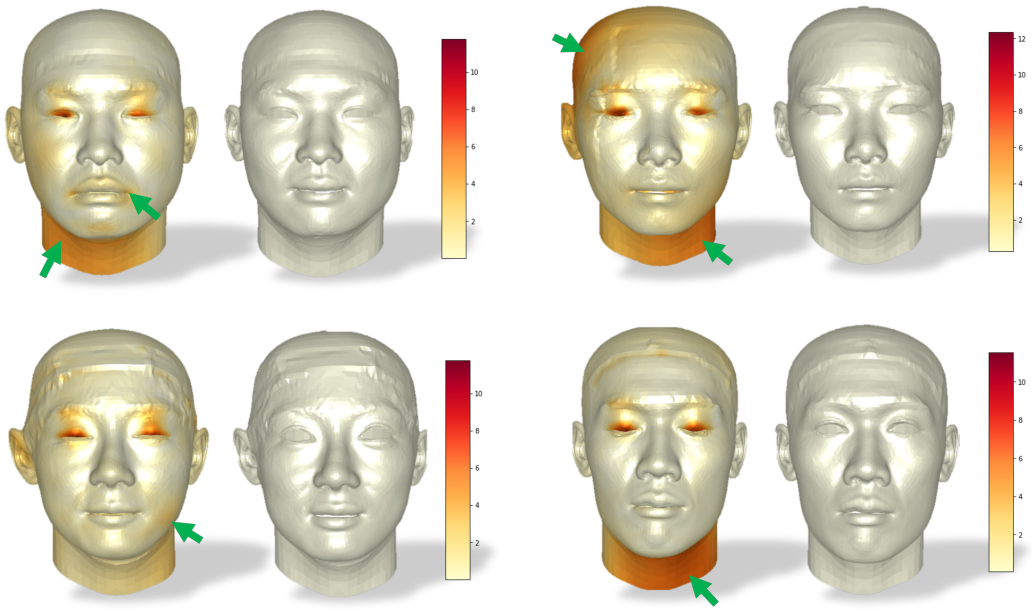


Figure 4.2: Comparison between 3d face scans with eyes opened (right) and closed (left). The colormap account for the distance between the meshes (in mm). Closing the eyes should only affect face geometry around the eyes, but we can see that all areas are touched. We observed differences in the neck, mouth, cheeks, and forehead. These differences are most certainly due to posture changes.

dimensionality reduction to the data. The objective is to transform the data into a space of lower dimension to make statistics easier and more reliable. The dimensionality reduction operation maps the data from the coordinate space of dimension $3n$ to a space of dimension l. Naturally, this low dimension space has to retain as much information as possible.

For facial expression neutralization, the dimensionality reduction algorithm must produce different latent variables for facial expressions and identity. This procedure may be referred to as disentanglement learning in the machine learning community. When this has been done, we can manipulate the facial expressions and remove them. Arguably, we can say that the challenge of facial expression neutralization lies in the disentanglement of facial expressions and facial attributes.

Several methods have been proposed to disentangle facial expressions from facial attributes. All of these methods make use of annotated databases of 3d scanned face meshes and supervised learning algorithms. To sum up, there are three main categories of methods:

HOSVD methods The first category of learning method uses tensor algebra and tensor decompositions. This methodology proposed by Vlasic et al. [88], has been used to build the FaceWarehouse model [27], the FLAME model [89], and recently the FaceScape model [26]. The main idea is to arrange the dataset into a 3d tensor whose modes[1] correspond to either facial attributes, facial expressions, or the positions of vertices. The tensor can be noted $\mathcal{T} \in \mathbb{R}^{3n \times m \times k}$, where n is the number of vertices, m is the number of faces, and k the number of expressions. This data tensor $\mathcal{T}$ is decomposed into a tensor of lower rank called the core tensor, noted $\mathcal{C}$,

[1]Each dimension of the tensor is called a mode.

	Median of volume errors (mL) before and after neutralization on 5 zones of interest				
					
Expression	Face	Cheeks	Chin	Mouth	Nasol. Folds
3 mouth stretch	144.43	30.72	23.11	14.61	5.64
16 grin	86.39	31.19	6.57	9.70	5.35
17 cheek blowing	79.60	37.71	5.15	10.16	10.55
12 lip puckerer	65.89	18.08	5.16	12.40	4.36
13 lip funneler	65.38	18.88	8.13	11.42	4.33
7 jaw forward	63.96	22.62	12.11	7.11	2.80
11 chin raiser	62.90	19.57	7.94	7.76	3.01
5 jaw left	57.01	20.13	6.95	4.61	3.73
10 dimpler	55.84	22.98	5.42	8.16	4.08
6 jaw right	53.61	19.56	6.80	4.62	3.39
14 sadness	52.57	11.56	4.50	3.34	1.95
15 lip roll	52.17	15.92	9.72	5.37	2.80
9 mouth right	52.12	20.29	5.17	5.35	3.41
20 brow lower	50.83	13.70	4.37	2.51	1.82
4 anger	48.90	13.26	4.13	2.47	1.79
8 mouth left	48.26	20.48	5.07	4.96	3.51
19 brow raiser	43.02	10.17	3.24	2.34	1.94
2 smile	41.63	13.86	3.47	5.98	2.80
18 eye closed	30.77	8.03	2.74	1.90	1.37
1 neutral	0.00	0.00	0.00	0.00	0.00

Table 4.1: This table shows the median of absolute volume errors (in mL) caused by facial expression. It is computed on 30% of the FaceScape dataset [26]. The errors are given for 5 different zones on the face. The expressions are sorted by amount of caused error on the face zone. We note that the volume errors caused by facial expressions make unusable the 3d scans for precise volume measurements.

using Higher-Order Singular Value Decomposition (HOSVD). With this method, an observed face is seen as the result of the n-mode product of the core tensor with identity and expression coefficients. For example, we can decompose a face mesh encoded in a vector $\mathbf{v} \in \mathbb{R}^n$ as:

$$\mathbf{v} = \mathcal{C} \times_2 \mathbf{w}_{subj}^{\intercal} \times_3 \mathbf{w}_{expr}^{\intercal}.$$

Where $\mathbf{w}_{subj}^{\intercal}$ are the weights corresponding to facial identity attributes and $\mathbf{w}_{expr}^{\intercal}$ are the weights of facial expressions.

Graph Convolutional Autoencoders (GCA) based methods The second category of methods makes use of Graph Convolutional Autoencoders (GCA [90]) to transform the data into a space of lower dimension on which facial attributes manipulations are made easier. Similar to the objective function optimized by PCA, the GCA minimizes a reconstruction error[1] of the

[1]The reconstruction error is the point-to-point distance between the original face mesh and a face mesh that has been encoded and then decoded.

data after being reconstructed from the latent representation.

• In Kacem et al. [1] they first used a GCA to transform the facial attribute deformations[1] into a latent representation, then a Generative Adversarial Network (GAN)[2] architecture is used to transform the latent face representation from an expressive face deformation to a neutral face. The face deformations can be reconstructed from a latent space using another GAN network.

• In Jiang et al. [91], they trained two parallel Graph Convolutional Encoders (GCA) with different objective functions. One GCA encodes facial expressions, and the other encodes the morphological changes of identity. Both networks minimize a reconstruction error to achieve dimension reduction. For the disentanglement, a constraint is added to force each network not to reconstruct the opposite network's attributes (facial expressions or identities). The first network minimizes the reconstruction error of facial identity changes but maximizes the reconstruction of facial expression changes. These two networks provide two distinct latent codes for a given expressive face that can be modified and reconstructed using a fusion module.

PCA based methods The third category of methods uses Principal Component Analysis to learn how to reduce the number of dimensions required to represent the shape deformations. A first PCA is used to learn the facial expressions, while another one is used to learn the facial attribute changes.

In the work of B. Chu [84] they used this methodology to neutralize facial expressions to make a face recognition algorithm robust to facial expressions. They found that the two latent spaces of the PCA have an intersection. In practice, it is sometimes hard to determine if a given face feature is due to a facial expression or a identity-specific attribute. This intersection is not a limitation in their work as the neutral reference face is known. In Egger et al. [85] the same intersection problem is discovered and verified on several datasets. They referred to this problem as an "ambiguity" between face identities and facial expressions. The Basel face model of Gerig et al. 2017 [25] employed the same methodology to train a generative 3d morphable model that can generate 3d realistic faces.

Limitations and solutions The proposed methods do not ensure that the deformation spaces are orthogonal. In practice, this means that we have no disentanglement guarantees between identities and expressions. Therefore, we propose also to optimize a criterion of orthogonality during training. We position our method between PCA-based methods [84] and the GCA-based method of Jiang et al. [91]. Like Jiang et al., we learned a latent representation of the data by optimizing a reconstruction error and a disentanglement term. In contrast, our dimensionality reduction is linear (i.e., only rotations and scaling are allowed), as it is the case in the work of Chu [84]. Compared to PCA-based methods and HOSVD methods, our method guarantees control of the orthogonality between the different bases (expression and identity), which leads to more accurate expression manipulations. The remainder of this chapter will study the PCA-based method from an optimization perspective. We call this method joint PCA. Then, we show how to modify the optimization problem to ensure a better disentanglement between facial expressions and identities and call this method Quasi Orthogonal Joint PCA (QOJPCA).

[1] It is a common practice in face modeling to represent faces by shape "differentials." This means that the face is not encoded by vertex positions but as the displacement of the vertices around a mean mesh.

[2] GAN are typically composed of two networks (a generator and a discriminator) in competition.

4.1.4 The available data

Throughout this chapter, we used the FaceScape dataset [26]. The dataset contains **938** distinct identities. Each person in the dataset performed **20** different facial expressions resulting in **18,760** textured 3d face scans. The authors fitted a template mesh toward all the 3d face scans. These fitted template meshes are called TU meshes (Topologically Uniform meshes). We used 70% of the TU meshes to train our model, and 30% has been kept for testing the trained models.

4.1.5 Chapter organisation

In section 4.2, we first study Principal Component Analysis (PCA) and Joint PCA (J-PCA) from an optimization perspective. This study helps to understand better the intersection problem between the expression and identity subspaces. Then we show how we can neutralize a facial expression on a given expressive face using expressions and identity bases.

In section 4.3, we formulate the problem of learning two deformation bases with a control on the orthogonality as an optimization problem. Then, we propose an efficient and straightforward iterative procedure to optimize this problem. We provide a proof of convergence for the algorithm. The subspaces resulting from our optimization procedure are made quasi orthogonal. Thus, we call our algorithm Quasi Orthogonal Joint PCA (QOJPCA).

In section 4.4, we tested our neutralization methodology on 30% of FaceScape dataset [26]. We compare OJ-PCA to J-PCA to evaluate the impact of using orthogonalization. We also evaluate the impact of neutralization on facial volumes, which are the measures of interest for QuantifiCare.

In section 4.5, we discuss the consequences of making the PCA subspaces orthogonal and the potential benefits of using neutralization for longitudinal volume measurements on face scans.

4.2 Disentangling facial expressions from identities with PCA and supervision

As we have seen in previous chapters, a simple model of shape deformations can be a basis [1] whose vectors span the allowed deformations of the shape. Formally, let $\mathbf{P} \in \mathbb{R}^{3n \times l}$ be a matrix whose columns vectors are the basis for a deformation space. Then, a shape $\mathbf{v} \in \mathbb{R}^{3n}$ writes: $\mathbf{v} = \bar{\mathbf{v}} + \mathbf{P}\boldsymbol{\alpha}$, where $\boldsymbol{\alpha} \in \mathbb{R}^l$ is a vector of parameters and l is the basis dimension.

Space of all deformations (i.e. the coordinate space $\mathbb{R}^{3n}$) The vector spaces for the facial expressions $\mathcal{V}_{exp}$ and for the identity change $\mathcal{V}_{subj}$ are subspaces of a high dimensional vector space called the coordinate space and simply noted $\mathbb{R}^{3n}$. Its basis is simply the canonical basis of dimension $3n$ (three times larger than the number of vertices on the template mesh). This

[1] Within the meaning of a set of vectors in a vector space.

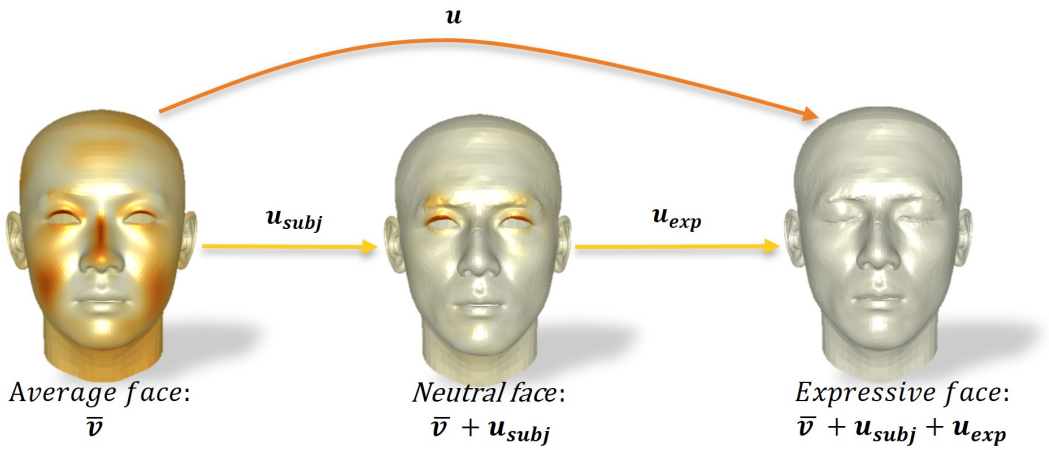


Figure 4.3: An expressive face (on the right) can be encoded as the linear combination of a displacement vector $\mathbf{u} \in \mathbb{R}^{3n}$ and a mean vector $\bar{\mathbf{u}} \in \mathbb{R}^{3n}$ containing the positions of the mean templates vertices. This displacement vector must be decomposed into two other displacement vectors corresponding to identity and facial expressions to neutralize facial expressions. The color map represents the distance to the expressive face on the right.

high-dimensional basis contains all possible deformations of the template mesh. An isotropic Gaussian random element of this space writes:

$$\mathbf{v} = \bar{\mathbf{v}} + \boldsymbol{\alpha}, \ \boldsymbol{\alpha} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{3n}). \tag{4.1}$$

We generated on the top of figure 4.4 some random samples from this space using equation (4.1). As expected, we observe random displacements of the mesh vertices.

Subspaces corresponding to facial expressions and identity changes Building a shape model consists of the creation of a subspace of the coordinate space $\mathbb{R}^{3n}$. In our case, we want to find two subspaces of $\mathbb{R}^{3n}$ one for the deformations related to changes in facial expressions and a second subspace corresponding to the changes in identity. We note $\mathbf{P} \in \mathbb{R}^{l_{subj} \times 3n}$ and $\mathbf{Q} \in \mathbb{R}^{l_{expr} \times 3n}$ respectively the basis matrices for identity changes and facial expressions changes. In the following sections (4.2.1), we will see how we can choose these two bases to be representatives of some observed data. In other words, we choose basis vectors such that the variance of the observed data on these basis vectors is maximal. The vectors in the bases are orthonormal, thus we have: $\mathbf{P}\mathbf{P}^\intercal = \mathbf{I}_{l_{subj}}$ and $\mathbf{Q}\mathbf{Q}^\intercal = \mathbf{I}_{l_{expr}}$. These vectors can be scaled to be proportional to the variance of the observed deformations in the bases. The variances are encoded into diagonal matrices: $\boldsymbol{\Sigma}_P \in \mathbb{R}^{l_{subj} \times l_{subj}}$ and $\boldsymbol{\Sigma}_Q \in \mathbb{R}^{l_{expr} \times l_{expr}}$.

Statistical meaning of expression and identity bases Once we have found the bases $\mathbf{P}, \mathbf{Q}$ and associated variances $\boldsymbol{\Sigma}_P, \boldsymbol{\Sigma}_Q$ we can generate new faces with the same reasoning as above (see figure 4.4). For example:

- A random identity in the range of the standard deviation (a.k.a. *1-sigma range*)[1] of the

[1] Approximately 68% of the observed data is in that range, sigma is the standard deviation.

observed data is: $\mathbf{v} = \bar{\mathbf{v}} + \mathbf{P}\boldsymbol{\Sigma}_P^{\frac{1}{2}}\boldsymbol{\alpha}$, with $\boldsymbol{\alpha} \sim \mathcal{N}(0, \mathbf{I}_{l_{subj}})$.

- A random expression on the mean face in the range of "1-Sigma" of the observed data is: $\mathbf{v} = \bar{\mathbf{v}} + \mathbf{Q}\boldsymbol{\Sigma}_Q^{\frac{1}{2}}\boldsymbol{\beta}$, with $\boldsymbol{\beta} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{l_{expr}})$.
- A random expression on a random face in the range of "1-Sigma" of the observed data is: $\mathbf{v} = \bar{\mathbf{v}} + \mathbf{P}\boldsymbol{\Sigma}_P^{\frac{1}{2}}\boldsymbol{\alpha} + \mathbf{Q}\boldsymbol{\Sigma}_Q^{\frac{1}{2}}\boldsymbol{\beta}$, with $\boldsymbol{\beta} \sim \mathcal{N}(0, \mathbf{I}_{l_{expr}})$ and $\boldsymbol{\alpha} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{l_{sub}})$.

Using expression and identity bases to manipulate attributes To perform facial expression manipulations, one can then project an unseen expressive face into these subspaces. Our objective is to decompose an expressive face into two distinct components corresponding to the identity-specific attributes and the facial expression. We only know the expressive face in the neutralization stage, and we are looking for the neutral face. In the learning stage, we want to learn the possible deformations for identity changes (i.e., $\mathbf{P}$) and facial expressions (i.e., $\mathbf{Q}$). This correspond to a supervised and longitudinal analysis.

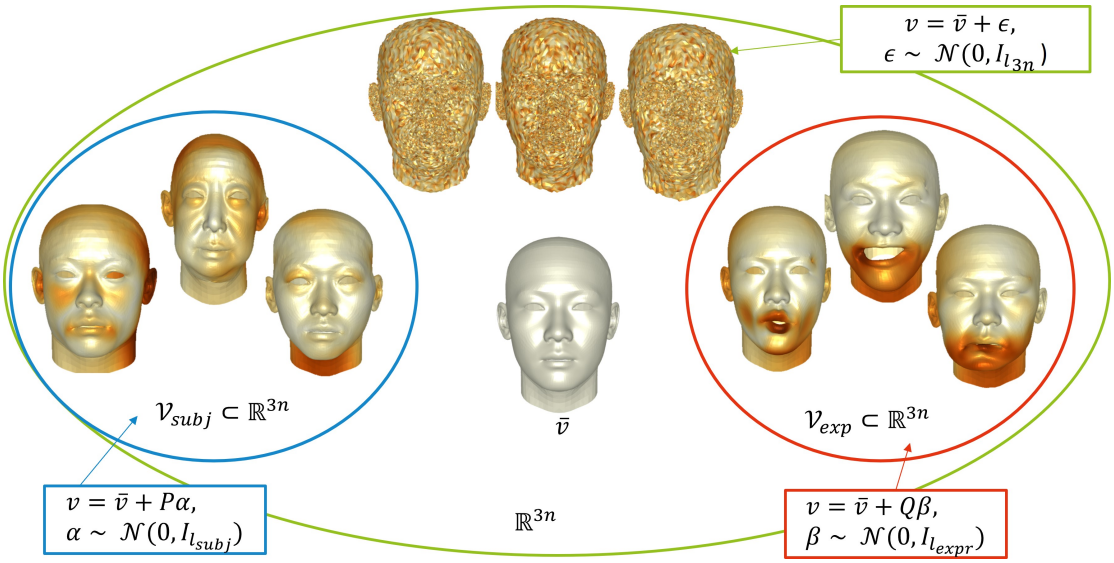


Figure 4.4: The mean face noted $\bar{v}$ is in the center. On top, we applied a deformation randomly sampled from an isotropic Gaussian of $\mathbb{R}^{3n}$, to the mean face. On the left subset, we have meshes randomly sampled from the identity space; on the right, meshes sampled from the expression space.

4.2.1 Learning space of vertex displacements (using PCA)

We have seen that the space of shape deformations is the union of two linear vector spaces $\mathcal{V}_{exp}$ and $\mathcal{V}_{subj}$. These spaces are themselves subspaces of the coordinate space $\mathbb{R}^{3n}$, which is also the space of all linear transformations (i.e., the field $\mathbb{R}^{3n}$).

We encode shapes as vectors of the real coordinate space $\mathbb{R}^{3n}$ where each coordinate of this large vector encodes one coordinate of the position of one vertex of the mesh. The whole dataset

is represented with a third order tensor:

$$\mathcal{T}_{pos} = \begin{array}{c} \xrightarrow{\text{expression}} \\ \left[\begin{array}{ccc} \mathbf{v}_1^1 & \dots & \mathbf{v}_1^k \\ \vdots & \ddots & \vdots \\ \mathbf{v}_m^1 & \dots & \mathbf{v}_m^k \end{array}\right] \\ \xleftarrow{\text{identity}} \end{array}$$

where k is the number of facial expressions and m is the number of identities. From this tensor we extract two data matrices written as: $\mathbf{X} \in \mathbf{R}^{3n \times m}$ and $\mathbf{Y} \in \mathbf{R}^{3n \times km}$:

- Each column of $\mathbf{X} \in \mathbb{R}^{3n \times m}$ contains a displacement field between the mean face with neutral expression and a person from the dataset with neutral expression. Where n is the number of mesh vertices and m the number of person in the dataset.
- Each column of $\mathbf{Y} \in \mathbb{R}^{3n \times km}$ contains a displacement field between a face with neutral expression and a face of the same person with an expression. Where n is the number of mesh vertices and m the number of people in the dataset and k is the number of facial expressions.

4.2.2 Dimension reduction (based on joint PCA)

Our data lives in a high dimension space of size $3n$. We would like to exploit the spatial correlations between the vertex positions (i.e. the variables) to reduce the number of dimensions used to represent the faces. To do so, PCA is appropriated as it has been developed to find a representation of data in a space of lower dimension with a minimum loss of information. In other words, PCA produces a set of vectors (a.k.a components, loadings, modes or factors) that are good "summaries" of the data. We can use a subset of these loadings vectors to build a basis for the shape model. More formally:

- We search a first linear transformation $f_{subj} : \mathbb{R}^{3n} \to \mathbb{R}^{l_{subj}} : \mathbf{x} \mapsto \mathbf{Px}$, where $\mathbf{P} \in \mathbb{R}^{l_{subj} \times 3n}$ is called the transformation matrix. Its columns are known as the identities' *loadings*.
- We search a second linear transformation $f_{expr} : \mathbb{R}^{3n} \to \mathbb{R}^{l_{expr}} : \mathbf{y} \mapsto \mathbf{Qy}$, where $\mathbf{Q} \in \mathbb{R}^{l_{expr} \times 3n}$ is called the transformation matrix. Its columns are known as the expressions'*loadings*.

We apply the transformation to the rows of our data matrix, thus the transformation of our data in matrix notation writes:

$$\mathbf{Z}_{subj} = \mathbf{QX} \quad \text{and} \quad \mathbf{Z}_{exp} = \mathbf{PY}, \tag{4.2}$$

where the transformed data $\mathbf{Z}_{subj}$ and $\mathbf{Z}_{exp}$ retains as much information as possible.

$\mathbf{Z}_{subj} \in \mathbb{R}^{l_{subj} \times m}$ and $\mathbf{Z}_{exp} \in \mathbb{R}^{l_{exp} \times km}$ are the coordinates of the data points in the lower dimension space (a.k.a. the identities' and expressions' *scores*).

The process to find the transformation matrices is equivalent for $\mathbf{X}$ and $\mathbf{Y}$. Therefore, in the following, we will show how this can be done for the first data matrix $\mathbf{X}$.

Recall that the objective is to minimize the loss of information after having transformed the data into a space of lower dimension. The precept: "Minimizing the loss of information after having transformed the data" translates well into an optimization problem. The latter can be approached through two equivalent point of views:

First point of view: Finding the subspace that maximizes the variance of the data after projection onto the lower dimension space. The optimization problem writes:

$$\mathbf{P} = \arg\max_{\mathbf{P}} \quad \text{Tr}(\mathbf{PX}(\mathbf{PX})^\intercal) \quad s.t. \quad \mathbf{PP}^\intercal = \mathbf{I}_{l_{subj}}. \tag{4.3}$$

Here, the term $\mathbf{PX}(\mathbf{PX})^\intercal$ corresponds to the covariance matrix of the transformed data[1]. Since we want to maximize the variance on each axis, we take the diagonal of the covariance matrix using the trace. Because we want distances to be conserved within the selected subspace, the projector $\mathbf{P}$ should be an *orthogonal* projector, that is, it must satisfy $\mathbf{PP}^\intercal = \mathbf{I}_{l_{subj}}$.

A second and equivalent way of thinking stands from the reconstruction error point of view (i.e., error after transformation to the latent space then transformed back to the input space).

$$\hat{\mathbf{P}} = \arg\min_{\mathbf{P}} \quad ||\mathbf{P}^\intercal\mathbf{PX} - \mathbf{X}||_f^2 \quad s.t. \quad \mathbf{PP}^\intercal = \mathbf{I}_{l_{subj}}. \tag{4.4}$$

In this case, the product $\mathbf{P}^\intercal\mathbf{PX}$ is the data transformed twice (to the latent space and back to the feature space). Some arithmetic operations shows that this formulation is equivalent to the variance formulation:

$$\begin{aligned}
&= \arg\min_{\mathbf{P}} \quad \text{Tr}((\mathbf{P}^\intercal\mathbf{PX} - \mathbf{X})^\intercal(\mathbf{P}^\intercal\mathbf{PX} - \mathbf{X})) \quad s.t. \quad \mathbf{PP}^\intercal = \mathbf{I}_{l_{subj}} \\
&= \arg\min_{P} \quad \text{Tr}((\mathbf{P}^\intercal\mathbf{PX})^\intercal\mathbf{P}^\intercal\mathbf{PX} - (\mathbf{P}^\intercal\mathbf{PX})^\intercal\mathbf{X} - \mathbf{X}^\intercal\mathbf{P}^\intercal\mathbf{PX} - \mathbf{X}^\intercal\mathbf{X} \quad s.t. \quad \mathbf{PP}^\intercal = \mathbf{I}_{l_{subj}} \\
&= \arg\min_{\mathbf{P}} \quad \text{Tr}((\mathbf{X}^\intercal\mathbf{P}^\intercal\cancel{\mathbf{PP}^\intercal}\mathbf{PX}) - (\mathbf{P}^\intercal\mathbf{PX})^\intercal\mathbf{X} - \mathbf{X}^\intercal\mathbf{P}^\intercal\mathbf{PX} - \cancel{\mathbf{X}^\intercal\mathbf{X}} \quad s.t. \quad \mathbf{PP}^\intercal = \mathbf{I}_{l_{subj}} \\
&= \arg\min_{\mathbf{P}} \quad \text{Tr}(-\mathbf{X}^\intercal\mathbf{P}^\intercal\mathbf{PX}) \quad s.t. \quad \mathbf{PP}^\intercal = \mathbf{I}_{l_{subj}} \\
&= \arg\min_{\mathbf{P}} \quad \text{Tr}(-\mathbf{PXX}^\intercal\mathbf{P}^\intercal) \quad s.t. \quad \mathbf{PP}^\intercal = \mathbf{I}_{l_{subj}} \\
&= \arg\max_{\mathbf{P}} \quad \text{Tr}(\mathbf{PX}(\mathbf{PX})^\intercal) \quad s.t. \quad \mathbf{PP}^\intercal = \mathbf{I}_{l_{subj}}
\end{aligned}$$

Solution to this optimization problem using SVD This constrained optimization problem can be turned into an unconstrained problem where we maximize a function called the Lagrangian, it writes:

$$\mathcal{L}(\mathbf{P}, \mathbf{\Lambda_P}) = \text{Tr}\left[\mathbf{PX}(\mathbf{PX})^\intercal\right] - \text{Tr}\left[\mathbf{\Lambda}_P(\mathbf{PP}^\intercal - \mathbf{I}_{l_{subj}})\right], \tag{4.5}$$

where the Lagrange multiplier matrix $\mathbf{\Lambda}_P$ is a symmetric matrix of size $l_{subj} \times l_{subj}$. The optimization problems turns to be:

$$\begin{aligned}
\hat{\mathbf{P}}, \hat{\mathbf{\Lambda_P}} = \arg\max_{\mathbf{P}, \mathbf{\Lambda_P}} \mathcal{L}(\mathbf{P}, \mathbf{\Lambda_P}) \iff \hat{\mathbf{P}} = &\arg\max_{\mathbf{P}} \quad \text{Tr}(\mathbf{PX}(\mathbf{PX})^\intercal) \\
&s.t. \quad \mathbf{PP}^\intercal = \mathbf{I}_{l_{subj}}.
\end{aligned} \tag{4.6}$$

[1]Here this is true only because we consider that the data is centered (the mean is zero). Also we omit the division by the number of samples to simplify the notations.

A necessary optimality condition is $\nabla\mathcal{L} = 0$, thus we compute the derivatives of $\mathcal{L}$ with respect to $\mathbf{P}$ and $\mathbf{\Lambda}_P$:

$$\frac{\partial[\mathrm{Tr}(\mathbf{PX}(\mathbf{PX})^\intercal)]}{\partial\mathbf{P}} = (\mathbf{X}(\mathbf{PX})^\intercal + \mathbf{PXX}^\intercal) = 2(\mathbf{PXX}^\intercal), \tag{4.7}$$

$$\frac{\partial[\mathrm{Tr}(\mathbf{\Lambda_P PP}^\intercal)]}{\partial\mathbf{P}} = 2(\mathbf{\Lambda_P P}), \tag{4.8}$$

$$\frac{\partial\mathcal{L}(\mathbf{P}, \mathbf{\Lambda_P})}{\partial\mathbf{\Lambda_P}} = 2(\mathbf{PP}^\intercal) - \mathbf{I}_{l_{subj}}. \tag{4.9}$$

A necessary condition to set the Lagrangian derivative to zero is that:

$$\begin{cases} -\mathbf{PXX}^\intercal + \mathbf{\Lambda}_P\mathbf{P} = 0 \\ \mathbf{PP}^\intercal = \mathbf{I}_{l_{subj}}. \end{cases} \tag{4.10}$$

The trick to solve equation (4.10) is to recognize an eigenvalue/eigenvector equation, where $\mathbf{P}$ are the eigenvectors of $\mathbf{XX}^\intercal$ and Λ_P the eigenvalues. Since the matrix $\mathbf{XX}^\intercal$ is larger than $\mathbf{X}$ it is preferable to compute the SVD decomposition on the data matrix $\mathbf{X}$.

Now if we have two spaces to which we want to apply dimension reduction we optimize:

$$\mathbf{P}, \mathbf{Q} = \underset{\mathbf{P},\mathbf{Q}}{\arg\max} \quad \mathrm{Tr}(\mathbf{PY}(\mathbf{PY})^\intercal) + \mathrm{Tr}(\mathbf{QX}(\mathbf{QX})^\intercal) \quad s.t. \quad \begin{cases} \mathbf{PP}^\intercal = \mathbf{I}_{l_{expr}} \\ \mathbf{QQ}^\intercal = \mathbf{I}_{l_{subj}} \end{cases} \tag{4.11}$$

A similar reasoning (differentiation of the Lagrangian) yields a necessary condition of the optimization problem (4.11):

$$\begin{cases} -\mathbf{PXX}^\intercal + \mathbf{\Lambda}_P\mathbf{P} = 0 \\ -\mathbf{QYY}^\intercal + \mathbf{\Lambda}_Q\mathbf{Q} = 0 \\ \mathbf{PP}^\intercal = \mathbf{I}_{l_{subj}} \\ \mathbf{QQ}^\intercal = \mathbf{I}_{l_e} \end{cases} \tag{4.12}$$

In this case (equation (4.12)) we recognise two eigenvalue/eigenvector equations, where $\mathbf{P} \in \mathbb{R}^{3n\times L_{subj}}$ are the eigenvectors of the identity covariance matrix $\mathbf{XX}^\intercal$ and $\mathbf{Q} \in \mathbb{R}^{3n\times l_{expr}}$ are the eigenvectors of the expression covariance matrix $\mathbf{YY}^\intercal$. These eigenvalue/eigenvectors equations can be again solved using Singular Value Decomposition (SVD). We call this methodology Joint Principal Component Analysis (JPCA) as it is equivalent to perform PCA on two data matrices.

4.2.3 Expression neutralization using the deformation model

In the previous section we have seen how we can compute two bases to express facial expressions and identity specific changes. We write these bases in matrix notations. That is to say, the column vectors in $\mathbf{P} \in \mathbb{R}^{3n\times l_{subj}}$ are the orthonormal basis vectors of the identity basis and the column vectors in $\mathbf{Q} \in \mathbb{R}^{3n\times l_{expr}}$ are the orthonormal basis vectors for the facial expressions. As a reminder, in point distribution models (PDM), the 3d faces are represented as a linear combination of a mean face and displacements of the vertices around that mean face. This writes for a given mesh $\mathbf{v} \in \mathbb{R}^{3n}$ as:

$$\mathbf{v} = \bar{\mathbf{v}} + \mathbf{P}\boldsymbol{\alpha} + \mathbf{Q}\boldsymbol{\beta} + \boldsymbol{\epsilon} = \bar{\mathbf{v}} + \mathbf{u}_{subj} + \mathbf{u}_{expr} + \boldsymbol{\epsilon}, \tag{4.13}$$

where:

- $\mathbf{u}_{subj} \in \mathbb{R}^{3n}$ is in the space of inter-identities face deformations $\mathcal{V}_{subj}$.
- $\mathbf{u}_{expr} \in \mathbb{R}^{3n}$ is in the space of facial expression deformations $\mathcal{V}_{expr}$.

The idea of our neutralization method is to first fit as best as possible the PDM by minimizing the residual term in equation (4.13), then to set the expression part of the deformation (i.e. $\mathbf{u}_{subj}$) to zero to remove the facial expressions.

Fitting the shape and expression model Fitting a face $\mathbf{v}$ consists in finding the $\mathbf{u}_{subj}$ and $\mathbf{u}_{expr}$ minimizing the error ϵ. Formally we optimize:

$$\hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\beta}} = \arg\min_{\alpha,\beta} \|\mathbf{v} - (\bar{\mathbf{v}} + \mathbf{P}\boldsymbol{\alpha} + \mathbf{Q}\boldsymbol{\beta})\|^2, \tag{4.14}$$

where $\bar{\mathbf{v}} \in \mathbb{R}^{3n}$ is the vector containing the positions of the mean face and $\mathbf{v} \in \mathbb{R}^{3n}$ is the vector containing the vertex positions of the mesh on which we want to fit the PDM.

This problem can be solved using a *block relaxation scheme* (see section 4.3.1 in [92]). The idea is that the set of parameters can be optimized by small "chunks" of parameters. In practice, we optimize alternatively α by fixing β and conversely. The pseudo code for this algorithm is given in Algorithm 2.

Algorithm 2: Alternated shape model fitting

Data: A basis matrix for the facial expressions deformations $\mathbf{Q} \in \mathbb{R}^{l_{expr} \times 3n}$, a basis matrix for the face attributes $\mathbf{P} \in \mathbb{R}^{l_{subj} \times 3n}$, a target mesh represented as a vector $\mathbf{v} \in \mathbb{R}^{3n}$.

Result: The parameters to fit the facial identity attributes $\hat{\boldsymbol{\alpha}} \in \mathbb{R}^{l_{subj}}$, and the parameters for the facial expressions $\hat{\boldsymbol{\beta}} \in \mathbb{R}^{l_{expr}}$.

```
begin
    β̂ := [0, ..., 0]ᵀ
    while ! converged do
        α̂ = Pᵀ(v − v̄ − Qβ̂)
        β̂ = Qᵀ(v − v̄ − Pα̂)
    end
end
```

Non-uniqueness of solution The fact that the bases for identity changes and facial expressions are not orthogonal implies that there is no unique solution to the optimization problem. Moreover, the non-orthogonality prevents from using direct methods such as Gaussian elimination or LU decomposition to solve the problem.

Removing the expression The facial attributes and facial expressions parameters can be used to reconstruct (i.e. approximate) the geometry of a face using equation (4.13). To neutralize a facial expression we simply remove the facial expression component from the equation to get the neutralized face. Formally, the vector of $\mathbf{v_{neutral}} \in \mathbb{R}^{3n}$ containing the vertices position of the neutralized mesh $\mathbf{v}$ writes:

$$\mathbf{v_{neutral}} = \bar{\mathbf{v}} + \mathbf{P}\hat{\boldsymbol{\alpha}} + \hat{\boldsymbol{\epsilon}}. \tag{4.15}$$

Here, $\mathbf{v_{neutral}} \in \mathbb{R}^{3n}$ gives us the new positions for the mesh vertices with a neutral expression.

4.2.4 Limitation: intersection of identity and expression subspace

It has been shown in Egger et al. [85], and Chu et al. [84] that the intersection of the expression space $\mathcal{V}_{exp}$ and the identity space $\mathcal{V}_{subj}$ computed using this Joint PCA methodology is not empty. From the algebraic point of view, the two spaces are not orthogonal. Therefore, there exist several different face deformations from each subspace that produce the same face. We can synthesize identity-specific changes on the basis of facial expressions and conversely. This intersection is illustrated in figure 4.5. This may not necessarily be a major issue for image synthesis. Still, in our use case of facial expression neutralization, the non-orthogonality is problematic because modifying the facial expressions will also modify the identity's identity.

In the next sections, we study the problem of orthogonality between subspaces computed using PCA.

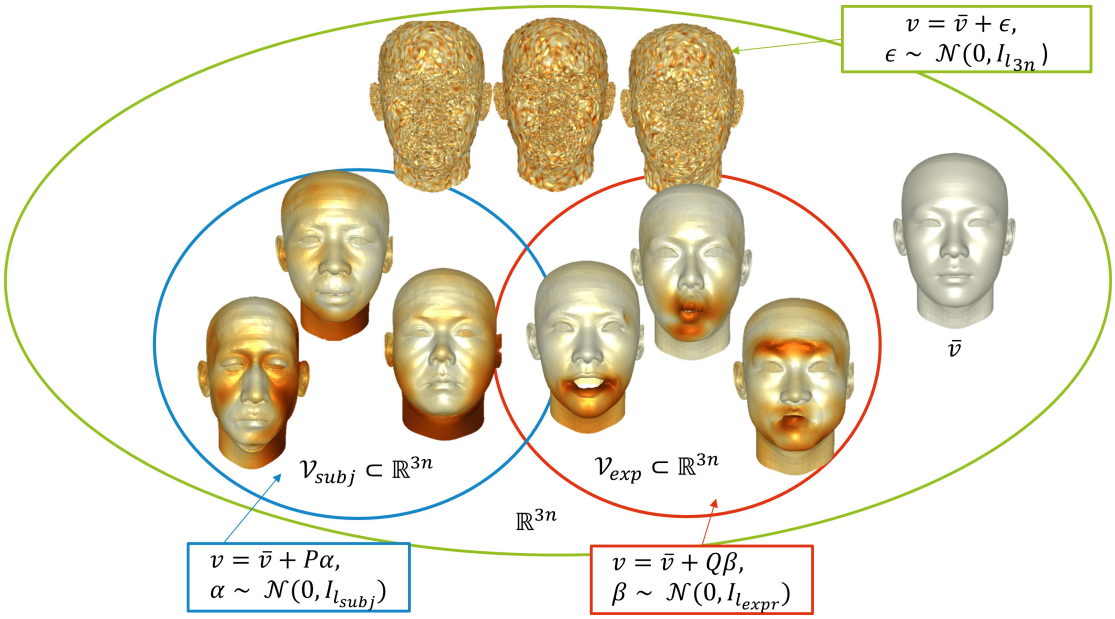


Figure 4.5: Shape spaces representation when using Joint PCA. The space of facial expressions has an intersection with the space of identity changes. This means that some faces can be represented either with facial expressions or morphological changes.

4.3 Improving the orthogonality between the two PCA spaces

To solve the intersection problem, we would like to transform a face vector $\mathbf{v} \in \mathbb{R}^{3n}$ into two low dimensional spaces that are mutually orthogonal. However, we will see that adding an or-

thogonality constraint leads to a non-linear system of equations. We have solutions in particular cases, but they are not carried easily to the general case. Therefore, we relax the constraint into a penalization term. We will see that we can make the two subspaces for expressions and identity almost orthogonal with our penalization term. Therefore, we call our method Quasi Orthogonal Joint Principal Component Analysis (QOJPCA).

4.3.1 Orthogonality as a constrained problem

First, we study the fully constrained problem. Similar to the methodology described in 4.2.2, we want that the transformation of the data to the space of a lower dimension preserves as much information as possible. We have expressed this principle as an optimization problem, where we minimize a reconstruction error[1]. Then, the straightforward idea to make the two subspaces orthogonal is to add an orthogonality constraint for each pair of basis vectors.

More precisely, given the two datasets $\mathbf{X} \in \mathbb{R}^{3n \times N_{\text{subj}}}$ and $\mathbf{Y} \in \mathbb{R}^{3n \times N_{\text{expr}}}$, we want to find the projection matrices $\mathbf{P} \in \mathbb{R}^{l_{subj} \times 3n}$ and $\mathbf{Q} \in \mathbb{R}^{l_e \times 3n}$ such that:

$$\begin{aligned} \hat{\mathbf{P}}, \hat{\mathbf{Q}} = \arg\min_{\mathbf{P},\mathbf{Q}} \quad & \|(\mathbf{I} - \mathbf{P}^\intercal\mathbf{P})\mathbf{X}\|^2 + \|(\mathbf{I} - \mathbf{Q}^\intercal\mathbf{Q})\mathbf{Y}\|^2 \\ \text{s.t.} \quad & \mathbf{P}^\intercal\mathbf{Q} = \mathbf{0}, \quad \mathbf{P}\mathbf{P}^\intercal = \mathbf{I}_{l_{subj}}, \quad \mathbf{Q}\mathbf{Q}^\intercal = \mathbf{I}_{l_e}. \end{aligned}$$

Here it is necessary to notice the addition of the constraint $\mathbf{P}^\intercal\mathbf{Q} = 0$ that ensures that the basis vectors between the two bases are perfectly orthogonal. In this case, the Lagrangian writes:

$$\begin{aligned} \mathcal{L}(\mathbf{P}, \mathbf{\Lambda_P}, \mathbf{Q}, \mathbf{\Lambda_Q}, \mathbf{K}) = & - \text{Tr}[\mathbf{PX}(\mathbf{PX})^\intercal] - \text{Tr}[\mathbf{QY}(\mathbf{QY})^\intercal] \\ & + \text{Tr}[\mathbf{\Lambda}_P(\mathbf{P}\mathbf{P}^\intercal - \mathbf{I}_{l_{subj}})] + \text{Tr}[\mathbf{\Lambda}_Q(\mathbf{Q}\mathbf{Q}^\intercal - \mathbf{I}_{l_{expr}})] + \text{Tr}[\mathbf{KPQ}^\intercal] \end{aligned} \tag{4.16}$$

Setting the Lagrangian derivative to zero gives us a necessary condition to the optimization problem (see Appendix C, section C.4 for the computation of the partial derivatives):

$$-\mathbf{PX}\mathbf{X}^\intercal + \mathbf{\Lambda}_P\mathbf{P} + \mathbf{K}^\intercal\mathbf{Q} = 0 \tag{4.17}$$

$$-\mathbf{QY}\mathbf{Y}^\intercal + \mathbf{KP} + \mathbf{\Lambda}_Q\mathbf{Q} = 0 \tag{4.18}$$

$$\mathbf{P}\mathbf{P}^\intercal = \mathbf{I}_{l_{subj}} \tag{4.19}$$

$$\mathbf{Q}\mathbf{Q}^\intercal = \mathbf{I}_{l_e} \tag{4.20}$$

$$\mathbf{P}^\intercal\mathbf{Q} = 0 \tag{4.21}$$

where the Lagrange multiplier matrix $\mathbf{\Lambda}_P$ (resp. $\mathbf{\Lambda}_Q$) is a symmetric matrix of size $l_{subj} \times l_{subj}$ (resp. $l_e \times l_e$) and $\mathbf{K}$ is a matrix of size $l_e \times l_{subj}$.

In equations (4.17) and (4.18), some variables are multiplied together, leading to a non-linear system of equations. The non-linear relationship between $\mathbf{K}, \mathbf{Q}, \mathbf{P}$ makes the system hard to solve, and we do not know if a solution exists in the general case. In section 4.3.1.1 we propose two ideas to solve the penalized problem but we assume restrictive properties on the covariance matrices $\mathbf{X}\mathbf{X}^\intercal$ and $\mathbf{Y}\mathbf{Y}^\intercal$. In section 4.3.2, we relax the constraint into a penalization term and propose an iterative procedure to solve the problem.

[1]The reconstructed face vector is the vector resulting from the application of the transformation and its inverse. The reconstruction error is the distance between the input face vector and the reconstructed face.

4.3.1.1 Direct Solution with hypotheses on the covariance matrices

Simultaneously diagonalizable covariance matrices First, we make the hypothesis that the covariance matrices $\mathbf{X}\mathbf{X}^\intercal$ and $\mathbf{Y}\mathbf{Y}^\intercal$ are simultaneously diagonalizable, which means that they have the same eigenvectors and that the matrices commute: $\mathbf{X}\mathbf{X}^\intercal\mathbf{Y}\mathbf{Y}^\intercal = \mathbf{Y}\mathbf{Y}^\intercal\mathbf{X}\mathbf{X}^\intercal$. We note $\mathbf{U} \in \mathbb{R}^{3n\times 3n}$ the eigenvectors of $\mathbf{X}\mathbf{X}^\intercal$ and $\mathbf{Y}\mathbf{Y}^\intercal$, thus we have:

$$\begin{aligned}\mathbf{X}\mathbf{X}^\intercal &= \mathbf{U}^\intercal\mathbf{D}_X\mathbf{U},\\ \mathbf{Y}\mathbf{Y}^\intercal &= \mathbf{U}^\intercal\mathbf{D}_Y\mathbf{U},\\ \mathbf{U}^\intercal\mathbf{U} &= \mathbf{I}_{3n},\end{aligned}$$

where the diagonal matrix $\mathbf{D}_X \in \mathbb{R}^{3n\times 3n}$ (respectively $\mathbf{D}_X \in \mathbb{R}^{3n\times 3n}$) contains the eigenvalues of $\mathbf{X}$ (respectively $\mathbf{Y}$).

Then, we can pick l_{subj} eigenvectors from $\mathbf{U}$ to create the matrix $\mathbf{P}$ and l_e eigenvectors (distinct from the previous l_{subj} ones) to create the matrix $\mathbf{Q}$:

$$\mathbf{P} = \mathbf{U}_s \in \mathbb{R}^{l_{subj}\times 3n},\quad \mathbf{Q} = \mathbf{U}_e \in \mathbb{R}^{l_e\times 3n},\quad \mathbf{U}_r \in \mathbb{R}^{(3n-l_{subj}-l_e)\times 3n}$$
$$\mathbf{U} = \left(\begin{array}{c}\mathbf{U}_s\\ \hline \mathbf{U}_e\\ \hline \mathbf{U}_r\end{array}\right),\quad \mathbf{D}_X = \mathrm{Diag}(\mathbf{D}_X^s, \mathbf{D}_X^e, \mathbf{D}_X^r),\quad \mathbf{D}_Y = \mathrm{Diag}(\mathbf{D}_Y^s, \mathbf{D}_Y^e, \mathbf{D}_Y^r)$$

We note $l_r = 3n - l_{subj} - l_e$. By construction, we have:

$$\begin{aligned}&\mathbf{U}_s\mathbf{U}_s^\intercal = \mathbf{I}_{l_{subj}},\quad \mathbf{U}_e\mathbf{U}_e^\intercal = \mathbf{I}_{l_e},\quad \mathbf{U}_r\mathbf{U}_r^\intercal = \mathbf{I}_{l_r}\\ &\mathbf{U}_s\mathbf{U}_e^\intercal = 0,\quad \mathbf{U}_s\mathbf{U}_r^\intercal = 0,\quad \mathbf{U}_e\mathbf{U}_r^\intercal = 0\end{aligned}$$

Therefore, the last 3 constraints (4.19), (4.20), and (4.21) are satisfied. The first two constraints write as:

$$\begin{aligned}-\mathbf{D}_X^s\mathbf{U}_s + \mathbf{\Lambda}_P\mathbf{U}_s + \mathbf{K}^\intercal\mathbf{U}_e = (\mathbf{\Lambda}_P - \mathbf{D}_X^s)\mathbf{U}_s + \mathbf{K}^\intercal\mathbf{U}_e = 0\\ -\mathbf{D}_Y^e\mathbf{U}_e + \mathbf{\Lambda}_Q\mathbf{U}_e + \mathbf{K}\mathbf{U}_s = (\mathbf{\Lambda}_Q - \mathbf{D}_X^e)\mathbf{U}_e + \mathbf{K}\mathbf{U}_s = 0\end{aligned}$$

It is sufficient to pick $\mathbf{\Lambda}_P = \mathbf{D}_X^s$, $\mathbf{\Lambda}_Q = \mathbf{D}_X^e$, $\mathbf{K} = 0$ to satisfy the constraint.

Direct Solution with simultaneously triangularisable matrices Similarly, we can find a solution if the covariance matrices $\mathbf{X}\mathbf{X}^\intercal$ and $\mathbf{Y}\mathbf{Y}^\intercal$ are simultaneously triangularisable. We proceed the same way except that we have triangular matrices instead of diagonal matrices:

$$\begin{aligned}\mathbf{X}\mathbf{X}^\intercal &= \mathbf{U}^\intercal\mathbf{T}_X\mathbf{U}\\ \mathbf{Y}\mathbf{Y}^\intercal &= \mathbf{U}^\intercal\mathbf{T}_Y\mathbf{U}\\ \mathbf{U}^\intercal\mathbf{U} &= \mathbf{I}_{3n}\end{aligned}$$

Discussion about simultaneously diagonalizable matrices: We have seen that if the covariances matrices are simultaneously diagonalizable, we can find orthogonal bases for the facial expressions and identity. In practice, this means that the covariances matrices should have the same eigenvectors, and there is very little chance that this property is valid for our data.

Indeed, it would mean that facial expressions and identity changes span the same subspaces. The following section will replace the strict orthogonality constraint with a penalization term. We will see that this modification leads to a system of equations that has a simple solution.

4.3.2 Quasi-orthogonality as a penalized problem

We relax the strict constraint of orthogonality between the two matrices $\mathbf{PQ}^\intercal = 0$ with a penalty on the square norm : $\lambda\|\mathbf{PQ}^\intercal\|^2$. The optimization problem then becomes:

$$\begin{aligned}\hat{\mathbf{P}}, \hat{\mathbf{Q}} = \arg\min_{\mathbf{P},\mathbf{Q}} \quad & \|(\mathbf{I} - \mathbf{P}^\intercal\mathbf{P})X\|^2 + \|(\mathbf{I} - \mathbf{Q}^\intercal\mathbf{Q})Y\|^2 + \lambda\|\mathbf{PQ}^\intercal\|^2 \\ \text{s.t.} \quad & \mathbf{PP}^\intercal = \mathbf{I}_{l_{subj}}, \quad \mathbf{QQ}^\intercal = \mathbf{I}_{l_e}.\end{aligned}$$

We rewrite the problem with the trace operator instead of the Frobenius norm:

$$\begin{aligned}\hat{\mathbf{P}}, \hat{\mathbf{Q}} = \arg\min_{\mathbf{P},\mathbf{Q}} \quad & \mathrm{Tr}[-\mathbf{PX}(\mathbf{PX})^\intercal] - \mathrm{Tr}[\mathbf{QY}(\mathbf{QY})^\intercal] + \mathrm{Tr}[\lambda(\mathbf{QP}^\intercal\mathbf{PQ}^\intercal)] \\ \text{s.t.} \quad & \mathbf{PP}^\intercal = \mathbf{I}_{l_{subj}}, \quad \mathbf{QQ}^\intercal = \mathbf{I}_{l_e},\end{aligned}$$

which is equivalent to minimizing the following Lagrangian function:

$$\begin{aligned}\hat{\mathbf{P}}, \hat{\mathbf{Q}} = \arg\min_{\mathbf{P},\mathbf{Q}} \quad & - \mathrm{Tr}[\mathbf{PX}(\mathbf{PX})^\intercal] - \mathrm{Tr}[\mathbf{QY}(\mathbf{QY})^\intercal] + \mathrm{Tr}[\boldsymbol{\lambda}(\mathbf{QP}^\intercal\mathbf{PQ}^\intercal)] \\ & + \mathrm{Tr}[\boldsymbol{\Lambda}_P(\mathbf{PP}^\intercal - \mathbf{I}_{l_{subj}})] + \mathrm{Tr}[\Lambda_Q(\mathbf{QQ}^\intercal - \mathbf{I}_{l_{expr}})].\end{aligned}$$

The partial derivative (see Appendix C, section C.5 for the derivation) of the Lagrangian are:

$$\frac{\partial[\mathrm{Tr}(\mathbf{PX}(\mathbf{PX})^\intercal)]}{\partial\mathbf{P}} = \mathbf{X}(\mathbf{PX})^\intercal + \mathbf{PXX}^\intercal = 2\mathbf{PXX}^\intercal \tag{4.22}$$

$$\frac{\partial[\mathrm{Tr}(\mathbf{QY}(\mathbf{QY})^\intercal)]}{\partial\mathbf{Q}} = 2\mathbf{QYY}^\intercal \tag{4.23}$$

$$\frac{\partial[\mathrm{Tr}(\mathbf{QP}(\mathbf{QP})^\intercal)]}{\partial\mathbf{P}} = 2\mathbf{QP}^\intercal\mathbf{Q}^\intercal \tag{4.24}$$

$$\frac{\partial[\mathrm{Tr}(\mathbf{QP}(\mathbf{QP})^\intercal)]}{\partial\mathbf{Q}} = 2\mathbf{PQ}^\intercal\mathbf{P}^\intercal \tag{4.25}$$

$$\frac{\partial[\mathrm{Tr}[\boldsymbol{\Lambda}_P(\mathbf{PP}^\intercal - \mathbf{I}_{l_{subj}})]]}{\partial\mathbf{P}} = 2\boldsymbol{\Lambda}_P\mathbf{P} - \mathbf{I}_{l_{subj}} \tag{4.26}$$

$$\frac{\partial[\mathrm{Tr}[\boldsymbol{\Lambda}_Q(\mathbf{QQ}^\intercal - \mathbf{I}_{l_{expr}})]]}{\partial\mathbf{Q}} = 2\boldsymbol{\Lambda}_Q\mathbf{Q} - \mathbf{I}_{l_{expr}} \tag{4.27}$$

Therefore, setting the Lagrangian derivative to zero leads to a system of equations whose solution is a necessary condition of our penalized optimization problem:

$$-\mathbf{PXX}^\intercal + \boldsymbol{\Lambda}_P\mathbf{P} + \lambda\mathbf{PQ}^\intercal\mathbf{Q} = 0 \tag{4.28}$$

$$-\mathbf{QYY}^\intercal + \boldsymbol{\Lambda}_Q\mathbf{Q} + \lambda\mathbf{QP}^\intercal\mathbf{P} = 0 \tag{4.29}$$

$$\mathbf{PP}^\intercal = \mathbf{I}_{l_{subj}} \tag{4.30}$$

$$\mathbf{QQ}^\intercal = \mathbf{I}_{l_e} \tag{4.31}$$

It is not clear if a direct solution exists, but one can solve iteratively as follows:

1. Start with $\mathbf{P}$ (resp. $\mathbf{Q}$) as the first l_{subj} (resp. l_e) eigenvectors of $\mathbf{X}\mathbf{X}^\intercal$ (resp. $\mathbf{Y}\mathbf{Y}^\intercal$)
2. **Step P** : for a fixed $\mathbf{Q}$ matrix, pick the first l_{subj} eigenvectors of $\mathbf{X}\mathbf{X}^\intercal - \lambda\mathbf{Q}^\intercal\mathbf{Q}$.
3. **Step Q** : for a fixed $\mathbf{P}$ matrix, pick the first l_e eigenvectors of $\mathbf{Y}\mathbf{Y}^\intercal - \lambda\mathbf{P}^\intercal\mathbf{P}$.

Then, iterate between steps 2 and 3 until the matrices $\mathbf{P}$ and $\mathbf{Q}$ do not change too much between consecutive iterations.

Normalization between reconstruction terms: The data matrix $\mathbf{Y}$ containing the facial expressions has k more times data samples than the matrix $\mathbf{X}$ containing the identity variability. Thus the first term in the objective function has less impact on the overall objective. We multiply the first term by k to make the norms of the two terms comparable. The problem then becomes:

$$\hat{\mathbf{P}}, \hat{\mathbf{Q}} = \arg\min_{\mathbf{P},\mathbf{Q}} k\|(\mathbf{I} - \mathbf{P}^\intercal\mathbf{P})\mathbf{X}\|^2 + \|(\mathbf{I} - \mathbf{Q}^\intercal\mathbf{Q})\mathbf{Y}\|^2 + \lambda\|\mathbf{P}\mathbf{Q}^\intercal\|^2 \text{ such that}$$
$$\mathbf{P}\mathbf{P}^\intercal = \mathbf{I}_{l_{subj}}, \quad \mathbf{Q}\mathbf{Q}^\intercal = \mathbf{I}_{l_e}.$$

Therefore, in the proposed algorithm the matrix $\mathbf{X}\mathbf{X}^\intercal$ is multiplied by k at each iteration.

Choosing the penalization parameter λ: We hypothesize that the value of λ should be at least in the same order of magnitude as the largest eigenvalues of $k\mathbf{X}\mathbf{X}^\intercal$. In the experiments, we have chosen $\lambda = 10^8$. With lower values of λ, the penalization term will probably not be significant enough to improve subspaces' orthogonality compared to the reconstruction term. However, as we increase λ above that $\lambda = 10^8$ value, the problem becomes very stiff and difficult to solve, ultimately leading to the constrained problem.

4.4 Neutralization experiments on faceScape dataset

4.4.1 Impact of penalization on subspace quasi-orthogonality

In our first experiment, we evaluate the impact of our penalization on the quasi-orthogonality between the two bases (expression and identity). To do so, we trained three models on **70**% of the FaceScape dataset [26]. Then, we compared the orthogonality score between the facial expression and identity bases. Our metric to account for the orthogonality is simply the norm: $||\mathbf{P}\mathbf{Q}^\intercal||$. The results are given in figure 4.6. In this plot, the orthogonality score is a function of the number of basis vectors in the basis.

• The first model (in blue) has been trained using the joint PCA procedure. That is to say, we performed two PCA separately to learn facial expressions and identity variations. We give the detailed procedure in section 4.2. We observe that the orthogonality score seems to increase linearly as the number of components [1] grow for the whole range. As soon as the number of components is higher than two, we observe non-orthogonality between expressions and identity (i.e., $||\mathbf{P}\mathbf{Q}^\intercal|| \neq 0$).

• The second model (in red) has been trained using QOJPCA with a regularization parameter $\lambda \approx 10^8$. We set λ to be ten times larger than the eigenvalue of the covariance matrix for identities

[1] The number of components is the number of basis vectors in the basis.

$\mathbf{X}\mathbf{X}^\intercal$. The regularisation term produces the expected results. Indeed, the orthogonality score is almost zero up to 30 components. Then it starts to increase up to a 20% of the orthogonality obtained with Joint PCA.

- The third model (in green) has been trained using QOJPCA with a regularization parameter $\lambda \approx 10^9$. We set λ to be 100 times larger than the eigenvalue of the covariance matrix for identities $\mathbf{X}\mathbf{X}^\intercal$. With this value of λ, we can keep the orthogonality score low even up to 50 components.

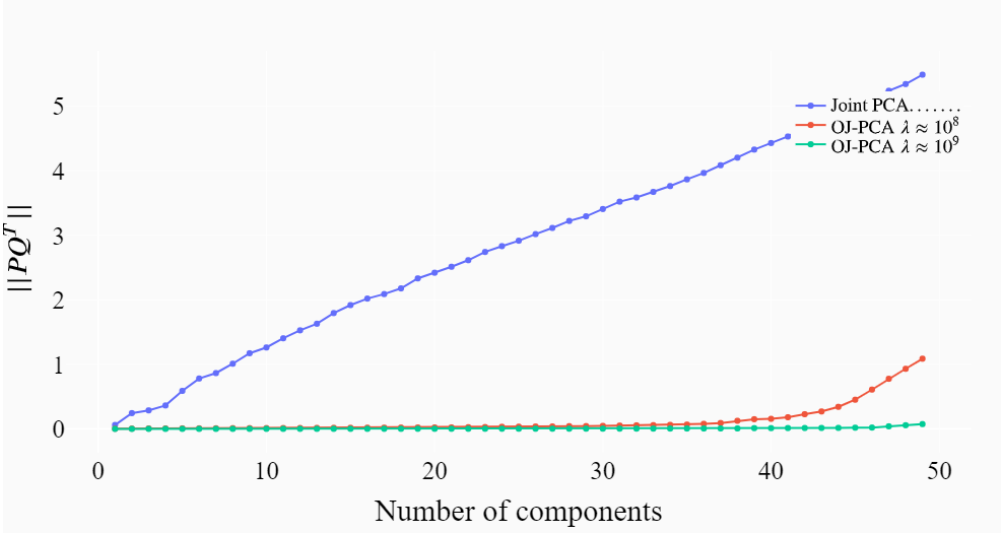


Figure 4.6: This figure compares the orthogonality score between joint PCA and quasi orthogonal joint PCA with two different regularisation parameters.

4.4.2 Effect of quasi-orthogonalization on facial expression neutralization

We now evaluate the effect of our quasi-orthogonalization on the neutralization of facial expressions (i.e., the estimation of neutral faces from expressive faces), which is described in section 4.2.3. We compared the model trained without orthogonalization (J-PCA model with fifty expression and identity modes) to the model with quasi-orthogonalization (QOJPCA model with $\lambda = 10^8$ and fifty expression and identity modes) in the neutralization procedure. The test set used for this evaluation represents 30% of the Facescape dataset [26], i.e., 5020 3d face scans. We compute the distances between the neutral faces and the estimated neutral faces using both methods. Then, we evaluate the significance of differences using Cohen's d effect size and the probability of superiority.

Average distance and Cohen's d Cohen's d effect size [93] allows for measuring the effect of using one method over another by considering the mean and variance of both methods' errors. Considering that the vector $\mathbf{v}_i \in \mathbb{R}^{3n}$ is the i-th ground truth neutral face and $\hat{\mathbf{v}}_\mathbf{i} \in \mathbb{R}^{3n}$ is the i-th estimated neutral face (either QOJPCA or J-PCA). We note $x_i \in \mathbb{R}$ the average distance

between the two meshes as being: $x_i = \frac{1}{3n}||\hat{\mathbf{v}}_\mathbf{i} - \mathbf{v}_i||_2$. Then, we can compute the mean $\mu \in \mathbb{R}$ and variance $\sigma \in \mathbb{R}$ of average distance over the $n = 5020$ samples of the test set as:

$$\mu = \frac{1}{n}\sum_{1<i\leq n} x_i \quad \text{and} \quad \sigma = \frac{1}{n}\sum_{1<i\leq n}(x_i - \mu). \tag{4.32}$$

Let μ_1 and σ_1 be the mean and variance of errors for J-PCA and μ_2 and σ_2 be the mean and variance of errors for QOJPCA. Then, the Cohen's d effect size between the two methods is:

$$d = \frac{\mu_1 - \mu_2}{\sqrt{\sigma_1^2 + \sigma_2^2}} \tag{4.33}$$

A general rule of thumb to interpret Cohen's d effect is to consider that: the effect is small under d=0.2, the effect is moderate under d=0.8, and the effect is large above d=0.8.

Probability of superiority We also calculate the probability that the QOJPCA error is smaller than the J-PCA error. This probability writes: $P(X \geq Y) = P(X - Y \geq 0)$ where X is the average distance after J-PCA's neutralization and Y is the average distance after QOJPCA's neutralization. This probability is given by the cumulative probability distribution of a normal distribution with mean $\mathbb{E}[X - Y]$ and variance $\mathbb{V}[X - Y]$. We assume that X and Y are independent, thus we have:

$$\begin{aligned} \mathbb{E}[X - Y] &= \mathbb{E}[X] - \mathbb{E}[Y] = \mu_1 - \mu_2, \\ \mathbb{V}[X - Y] &= \mathbb{V}[X] + \mathbb{V}[Y] - \mathrm{cov}[X, Y] = \sigma_1^2 + \sigma_2^2, \\ \text{and finally} \quad P(X \geq Y) &= \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{0} e^{-\frac{1}{2}\left(\frac{t-\mu_1+\mu_2}{\sigma_1+\sigma_2}\right)}\mathrm{dt} \end{aligned} \tag{4.34}$$

This probability is easily interpretable, and in some references, it is referred to as the Common Language (CL) effect size [94]. We compute the effect size and the probability of superiority for each of the twenty facial expressions (fig. 4.2).

Results from table fig. 4.2 We observe a strong positive effect of our quasi-orthogonality on the neutralization of mouth stretch and cheek-blowing expressions. For these two expressions, the average distance error before any neutralization is 2.16mm for the mouth stretch and 1.41 for the cheek blowing expression. The average error distance drops from 2.0mm (with J-PCA) to 0.7mm (with QOJPCA) for the mouth stretch expression and from 1.1mm to 0.7mm for cheek blowing expression. In both cases, the significance is high with a Cohen's d-effect above 1.0 (i.e., $d = 4.17$ for mouth stretch and $d = 1.23$ for cheek blowing). In particular, the probability that QOJPCA is superior to J-PCA is 98.5% for the mouth stretch expression.

Then, we have a series of six expressions for which the effect of quasi orthogonality is positive but moderate. For expressions, lip puckerer, lip funneler, grin, smile, dimpler and chin raiser, the d-effect ranges from $d = 0.86$ to $d = 0.21$ and the superiority probability from 66.7% to 54.2%.

However, the neutralization QOJPCA performs slightly worse than J-PCA for a set of three expressions. This is the case for eye closed, brow raiser, and when applying the neutralization to an already neutral face. For these 3 expressions, the d-effect range from $d = -0.48$ to $d = -0.70$. Among all expressions, these three expressions produce fewer geometrical changes

on the face (from 0.8mm to 0.0mm for the neutral expression). One hypothesis is that our quasi-orthogonalization is obtained at the expense of the reconstruction error (see eq. (4.22)), meaning that without quasi-orthogonality, the template will fit with slightly fewer details. This is visible in the average error after neutralizing the neutral expression, which is 0.2mm for J-PCA and 0.5mm for QOJPCA.

The effect of quasi-orthogonalization is small for the other nine expressions (lip roll, brow lower, mouth left, mouth right, sadness, jaw left, anger, jaw right, jaw forward). In any case, the neutralization (either QOJPCA or J-PCA) reduced the average distance error.

	Average distance error (mm)			Significance	
	Baseline	J-PCA	QOJPCA	Cohen's d	> probability (%)
3_mouth_stretch	2.16(±0.36)	2.00(±0.38)	0.71(±0.21)	4.17	98.5
17_cheek_blowing	1.41(±0.33)	1.08(±0.32)	0.75(±0.21)	1.23	73.5
12_lip_puckerer	1.02(±0.20)	0.85(±0.20)	0.69(±0.17)	0.86	66.7
13_lip_funneler	1.26(±0.28)	0.89(±0.28)	0.70(±0.25)	0.71	63.9
16_grin	2.24(±0.49)	1.52(±0.56)	1.13(±0.57)	0.70	63.7
2_smile	1.15(±0.23)	0.74(±0.25)	0.59(±0.22)	0.61	62.1
10_dimpler	2.08(±0.54)	1.69(±0.57)	1.40(±0.54)	0.52	60.3
11_chin_raiser	1.49(±0.60)	1.28(±0.61)	1.16(±0.57)	0.21	54.2
15_lip_roll	1.47(±0.55)	1.21(±0.60)	1.10(±0.57)	0.18	53.6
20_brow_lower	1.67(±0.48)	1.60(±0.49)	1.54(±0.46)	0.12	52.4
8_mouth_left	1.27(±0.18)	0.67(±0.15)	0.67(±0.18)	0.03	50.6
9_mouth_right	1.26(±0.15)	0.66(±0.15)	0.67(±0.15)	-0.01	49.9
14_sadness	1.40(±0.58)	1.09(±0.65)	1.11(±0.58)	-0.03	49.5
5_jaw_left	1.11(±0.37)	0.76(±0.39)	0.78(±0.35)	-0.04	49.3
4_anger	1.21(±0.60)	1.05(±0.69)	1.11(±0.77)	-0.08	48.4
6_jaw_right	0.93(±0.23)	0.65(±0.19)	0.68(±0.17)	-0.18	46.4
7_jaw_forward	0.78(±0.22)	0.70(±0.20)	0.74(±0.18)	-0.18	46.4
18_eye_closed	0.75(±0.28)	0.57(±0.27)	0.70(±0.23)	-0.48	40.4
19_brow_raiser	0.66(±0.19)	0.55(±0.17)	0.65(±0.17)	-0.57	38.8
1_neutral	0.00(±0.00)	0.19(±0.25)	0.47(±0.50)	-0.70	35.5

Table 4.2: This table shows the effect of quasi-orthogonalization on the neutralization of facial expression. The significance tests are computed between J-PCA and QOJPCA. The average distance is computed on the face zone (i.e., the neck and the back of head does not account).

4.4.2.1 Qualitative assessment of neutralization

We can also observe the results of our neutralization methodology on some examples of 3d face scans and compare them visually to the ground truth neutral scan. We generated a figure for a face from the publishable list (see figure 4.7.). It is interesting to note that visually the neutralization with orthogonalization can realistically correct the facial expression even when the average distance metric does not show an important improvement. Therefore, it is possible that

the remaining error after neutralization (fig. 4.2) is due to other factors such as postural changes or reconstruction errors. For example, we can see that our facial expression neutralization can open the eyes of the person (second row on figure 4.7) while the average distance is almost equivalent before and after the neutralization(see table 4.2).

4.4.3 Effect of neutralization on volume measurement errors

In our last experiment, we evaluated the effect of the neutralization of facial expressions on the accuracy of volume measurements. Indeed, we have seen that facial expressions strongly impact the volume measures. In the current setting, the presence of facial expressions on 3d face scans makes the data unusable. This experiment aims to study if the neutral faces resulting from our neutralization methodology could be used for longitudinal volume measurements.

To do so, we compute the absolute volume between the 3d face scans with expression and the 3d face scan with a neutral expression on our test set. This volume accounts for the volume error caused by the facial expression on a given 3d face scan. The high-level objective of our approach is to reduce this volume error with neutralization. Therefore, we compare the following volumes:

- Volume 1: between the 3d face scan with a neutral expression and the neutralized face.
- Volume 2: between the 3d face scan with a neutral expression and the expressive face scan.

All the volumes are computed using QuantifiCare's algorithms to reproduce a realistic scenario. To reduce the impact of registration problems and posture changes, we have computed the volumes only on interest zones: the face zone, the nasolabial folds, the mouth, the chin, and the cheeks. The zones are illustrated in fig. 4.1. The separation of the experiment by zone allows for better analysis of the impact of the neutralization on a specific expression and zone. The 3d face scans from FaceScape are registered on the back of the head. In Facescape's acquisition protocol, all the subjects have their head leaning on a support behind them. However, the position of the neck is not controlled, which may produce errors in this zone. We provide also Cohen's d-effect of QOJPCA neutralization and the probability that our neutralization reduces the volume measurement error. We show the results for the face zone in table 4.3. The results for the other zones are in the appendices A.2. As expected, in appendix A.2, we can see that the reduction of volume measurements's errors due to our neutralization is proportionnaly better on small zones, such as the mouth and nasolabial folds, because the posture changes has less impact on these areas.

Results from the table 4.3: We found a strong positive effect of facial expression neutralization on mouth stretch and grin expressions. The average volume errors drop respectively from 149.2mL to 59.5mL for mouth stretch and 91ml to 55.1mL for the grin expression. In both cases, the d-effect is above 1.0. For the mouth stretch expression, the probability that our neutralization algorithm improves the measurements is 87.0%. For the grin expression, this probability is 69.2%.

For the five following expressions (cheek blowing, lip funneler, lip puckerer, and chin raiser), the effect of our neutralization is positive but moderate. In particular, the probability of improvement is between 55% and 60%.

However, when applying the neutralization to a neutral expression or to the eye closed expression, the effect is moderately negative, which is not surprising as the neutral row is the row

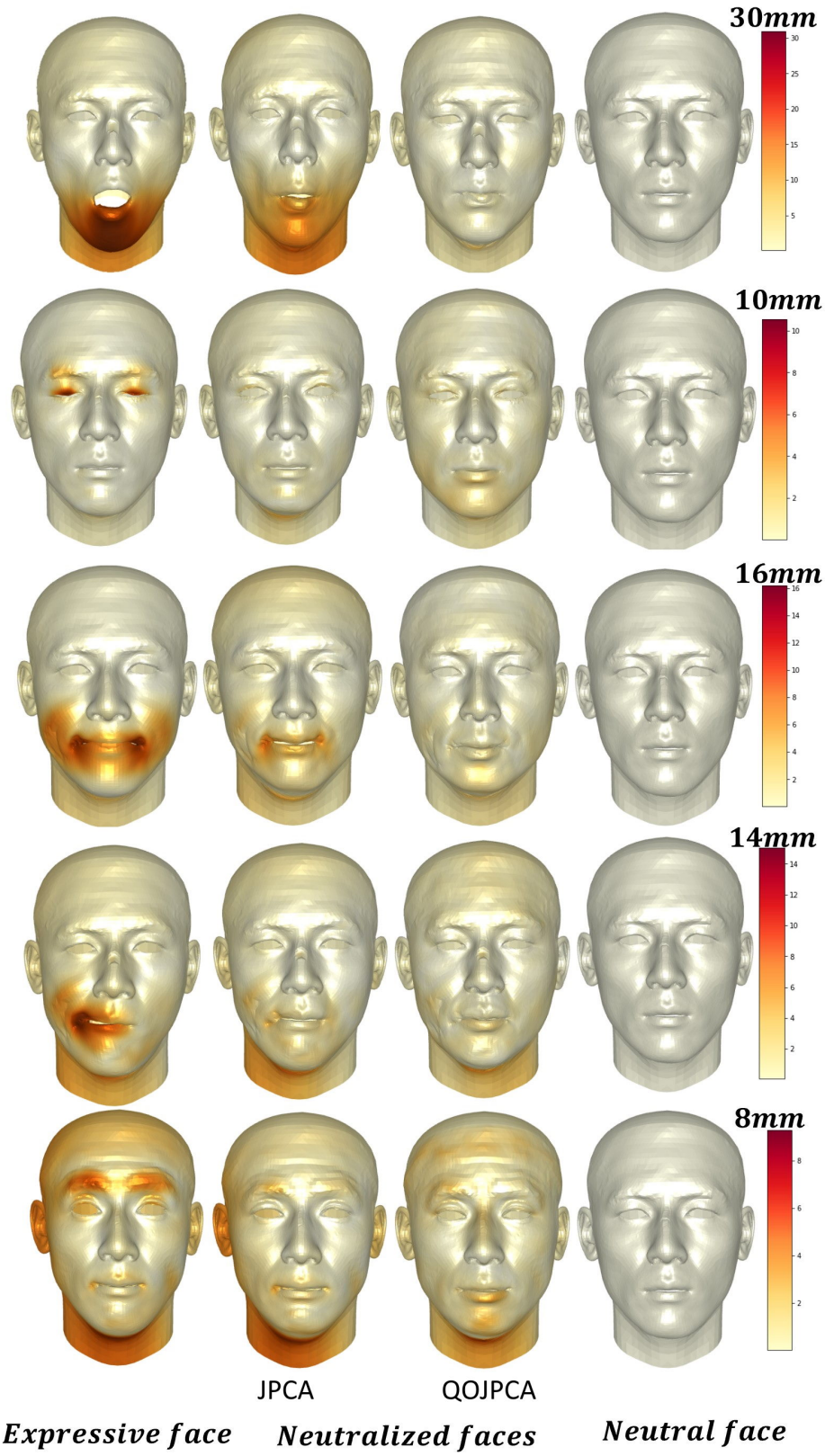


Figure 4.7: Examples of facial expression neutralization with and without the orthogonalisation. We note that without orthogonalisation the method failed on the first and third expressions.

where there is no expression, and closing the eyes is a tangential movement that does not change the volume.

For all the other expressions, the effect of neutralization is low or insignificant. The d-effect is below 0.2. One hypothesis is that these expressions, such as mouth right, involve tangential movements of the skin that do not change the volume too much, minimizing the effect of neutralization.

In any case, we have an average volume error that is around 50mL with QOJPCA neutralization. This shows a reduction in volume errors, but volume errors caused by facial expressions make the scans unusable for volume measurements. Recall that the order of magnitude of the measurements is about 5 mL.

A note about posture changes Interestingly, we found modifications of posture between the 3d face scans, which inevitably increases the observed volume errors. In other words, we observe that the head position of the person is not consistent between several 3d scans with different facial expressions. Several clues support this information. First, we remark that closing the eyes impacts 30mL on the face volume. This is surprising because closing the eyes produces a movement tangent to the eyes' surface, and thus it is not supposed to change the volume of the face significantly. Besides, facial expressions localized to a small area impact other parts of the visage. For example, a facial expression such as blinking (i.e., 18_eye_closed) should slightly affect the volume around the eyes, but we observe the modification of volumes in the mouth zone. It is clear that, the postural changes contributes to the volume errors that we have on the data. Therefore, the changes in posture can explain a part of the remaining error after our netralization.

4.4.4 Comparing QOJPCA to Kacem et al. [1]

A comparison with the GCA method of Kacem et al. [1] would have been interesting, but currently, we cannot reproduce their experiment. In practice, we would need to know which partitions of the dataset have been used as a test set in their experiments. However, we computed the same metric as described in their publication (i.e., the mean error between the mesh with expression and the neutralized mesh). The results are shown in table 4.4. With GCAs [1], the average error after neutralization is 2.02mm. With our QOJPCA method, we obtain an average error of 1.29mm. Therefore, our method improves the neutralization results significantly.

Two methodological differences between our QOJPCA and [1] might explain this improvement. First, the disentanglement method is different. In their case, the disentanglement is achieved by penalizing the reconstruction of facial expressions with the identity basis and, conversely, penalizing the reconstruction of identity changes with the expression basis. On the other hand, we achieve a disentanglement by optimizing the orthogonality between expressions and identity subspaces.

Secondly, our neutralization method preserves the high-resolution details on the expressive face, while the GCA produces a smoothed face mesh. This preservation of details is possible because we estimate the deformation from the expressive to neutral face in the mesh's space and not in the latent space. Therefore, the details that our model cannot reconstruct are kept intact by our neutralization algorithm. Formally, the error term $\hat{\boldsymbol{\epsilon}}$ in equation (4.15) represents the geometric details not fitted is conserved in the neutralized face. Conversely, they manipulate the expression in the latent space of the GCA. Therefore details are lost during the encoding and

	Average volume error (mL)			Significance	
	Baseline	J-PCA	QOJPCA	Cohen's d	> probability (%)
3_mouth_stretch	149.2(±43.7)	145.3(±40.3)	59.5(±36.1)	2.24	87.0
16_grin	91.2(±36.2)	78.4(±34.6)	55.1(±35.7)	1.00	69.2
17_cheek_blowing	85.1(±36.7)	71.3(±36.9)	65.3(±36.1)	0.54	60.7
13_lip_funneler	70.8(±34.6)	65.2(±33.2)	53.9(±34.2)	0.49	59.7
12_lip_puckerer	69.6(±36.3)	67.7(±34.8)	54.5(±34.3)	0.43	58.5
11_chin_raiser	67.2(±36.0)	64.1(±35.6)	57.6(±34.7)	0.27	55.4
10_dimpler	60.2(±37.0)	62.8(±36.4)	50.7(±35.4)	0.26	55.2
14_sadness	57.1(±36.0)	52.5(±35.2)	48.0(±34.8)	0.25	55.1
15_lip_roll	56.4(±33.5)	51.4(±33.1)	51.2(±34.1)	0.15	53.0
9_mouth_right	57.7(±34.8)	49.8(±34.2)	54.0(±34.7)	0.10	52.1
7_jaw_forward	67.8(±35.0)	63.2(±33.8)	64.2(±34.1)	0.10	52.1
5_jaw_left	65.1(±77.2)	52.9(±71.5)	58.9(±63.9)	0.09	51.8
2_smile	46.8(±37.3)	52.8(±36.4)	44.3(±34.8)	0.07	51.4
20_brow_lower	54.8(±34.4)	48.1(±33.2)	52.8(±34.1)	0.06	51.2
6_jaw_right	55.9(±33.3)	47.2(±32.9)	55.4(±33.7)	0.01	50.3
8_mouth_left	53.2(±36.7)	48.7(±35.5)	53.4(±35.4)	-0.00	49.9
4_anger	51.3(±16.5)	46.6(±31.7)	52.7(±77.7)	-0.03	49.4
19_brow_raiser	48.6(±38.2)	44.1(±38.2)	53.0(±36.8)	-0.12	47.7
18_eye_closed	36.0(±36.6)	36.4(±34.9)	44.3(±34.9)	-0.23	45.4
1_neutral	0.0(±0.0)	13.2(±19.6)	33.0(±62.9)	-0.74	30.0

Table 4.3: This table compares the effect of our QOJPCA expression neutralization on the volume measurements (on the face zone). Baseline correspond to the (absolute) average volume generated by a specific facial expression on a neutral face. For example, the mouth stretch modifies the volume by 149mL. The significance tests are computed between the QOJPCA column and the baseline column. We give the neutralization results with J-PCA, but they do not impact the significance columns.

Method	Mean error (mm)	Maximum error(mm)
Initial	2.75	8.83
Kacem [1]	2.02	n/a
J-PCA [84; 85]	2.53	6.46
OJ-PCA (ours)	2.08	5.34

Table 4.4: The mean and maximum error between the mesh with expression and the neutralized mesh. The mean error is computed as mean of the vertex-to-vertex absolute distance between the meshes. The Initial row shows the error that is caused by the facial expressions. The next rows shows the different expression neutralization methodologies.

the decoding stages.

4.5 Conclusion

Conclusions on the orthogonality of expression and identity bases The experiments we have done show that ensuring quasi-orthogonality between the basis of expressions and the basis for identity improves the accuracy of neutral face predictions. In a more general context, the orthogonalization between expressions and identity can be helpful in manipulating facial expressions with a minimum on identity. Although we do not have a solution for the strict orthogonality constraint, we can force subspaces' quasi-orthogonality with a penalization term in the optimization problem. Our formulation of the problem led to a computationally tractable algorithm involving the resolution of eigenvalue/eigenvector equations.

Conclusions on neutralization accuracy We verified that orthogonality between the subspaces is necessary to disentangle deformations properly. However, the problem of neutralizing facial expressions to improve volume measurement accuracy seems to be more complicated than expected. Even with our quasi-orthogonalization, the neutralization algorithm produces neutral faces that are not usable for precise 3d face measurements. The 50 mL volume error after neutralization on the face zone is too high. It is not evident to determine the causes of this error. A first hypothesis is that our penalization impacts the PCA's reconstruction term to enforce more orthogonality. Therefore, we introduce some reconstruction error when tryng to neutralize a facial expression. This reconstruction error problem is more visible for expressions that do not impact the volume, such as blinking or when we apply the neutralization to a face that already has a neutral expression. Other hypotheses, for the remaining neutralization error, are variability between neutral expressions across identities, posture changes, registration problems, and overfitting or underfitting of our expression model to the training set.

Limitations and future work We have identified that the posture of the identities can change across the 3d scans. Future work could use a model of human motion such as openSim [95; 96; 97] to "standardize" the posture of identities and certainly improve the volume measures achieved by QuantifiCare.

Besides, we could certainly improve the accuracy of facial expression neutralization by reducing the number of facial expressions modeled simultaneously. On the one hand, we have seen that some facial expressions (e.g., closing eyes) do not impact volume measurements because the movement is tangential to the surface. On the other hand, some expressions are not reproduced evenly among the identities and are hard to model (anger, sadness). Therefore, it would be interesting to keep only expressions that are well reproduced in the dataset and are problematic for the volume measurements.

From a theoretical point of view, OJ-PCA can be seen as a form of tensor decomposition that ensures orthogonal properties between the two subspaces. Therefore, the question arises whether our method is a particular case of an orthogonal tensor decomposition such as higher orthogonal iteration of tensors (HOOI). At first glance, the joint principal component analysis (J-PCA) and Higher-Order Singular Value decomposition (HOSVD) have similar objectives.

We have several hyperparameters in our method and have not optimized them. For example, a first parameter allows for adjusting the orthogonality of the expression and identity subspaces.

Another parameter is the number of components that we used on each basis. There is probably room for improvement in the neutralization of facial expressions if we optimize these parameters.

4.5.1 Summary of contributions

We studied the problem of independence (a.k.a. disentanglement) between facial expressions and identities. In particular, we were interested in separating facial expressions from identity and better understanding the problem. We identified that a part of the problem comes from the absence of orthogonality between the identity and expression subspaces. We evaluated the feasibility of neutralizing facial expressions for longitudinal volume measurements despite this ambiguity.

We proposed a novel algorithm to build a model of facial expressions orthogonal to a model of face identities. Our algorithm is similar to PCA, but we added a penalization term in the optimization procedure to obtain quasi-orthogonal bases. We showed that the quasi-orthogonality property of our expression model leads to better results in expression neutralization experiments than in state-of-the-art methods.

Chapter 5
Conclusions

5.1 Summary of research problems and proposed solutions

This research aimed to build statistical shape and deformation models of human body parts to analyze shapes automatically and improve digital measuring tools in dermatology and aesthetic medicine. This study was conducted in a context where the amount of available data is limited and where there is some noise in the data labels, which motivated the development of interpretable and comprehensive models.

This thesis covers many problems raised by the modeling of *shapes* and *deformations*. We cover all the stages of the modeling process, from the *registration* of 3d scans to the *clinical applications*. Our research allowed us to understand the scope of application of these models and identify and propose solutions to the methodological challenges. We developed models of face *morphology, expressions, and aging* that can be used to estimate or simulate related deformations on the patient's 3d scans. We propose *interpretable methods* and trained models that work when little data is available. We adapted state-of-the-art registration methods to QuantifiCare's 3d scans and finally proposed a solution to the problem of *entanglement* between identities and expressions occurring in face morphable models.

Correspondence problem One of the main challenges encountered when building statistical shape models was the *correspondence problem*[1]. Indeed, coherent features between the shapes, such as corresponding points between the meshes, are often needed to make statistics. The accuracy of registration has an immediate and dramatic effect on the statistical model to be developed. Therefore, it was necessary to solve this problem accurately. The state-of-the-art methods for registering 3d face scans are based on Non-rigid Iterative Closest Point (NICP)[25] and Gaussian Process Morphable Models (GPMMs) [42], which work well on faces with a neutral expression. However, some improvements are needed to register *facial expressions* properly. For instance, mouth openings are changing the mesh's geometry too much and led to wrong registration results in many cases. We proposed a new kernel based on geodesic distances for the GPMM framework that is more flexible and allows more accurate registration of complex deformations, such as registering facial expressions, which was impossible before. We also adapted the NICP algorithm to the registration of QuantifiCare's 3d scans, where there can be irrelevant parts such as hair and clothes in the meshes. With the advent of fast methods to compute geodesic distances (such as [54]), the overall computation drawback of our kernel is not significant. Moreover, it relies on much more meaningful relationships between the points on the surface.

Finding predictive variables for biological markers The second main problem specific to dermatology, aesthetic medicine, and, more generally, medical imaging is to find *predictive variables* for the biomarkers of interest. In our case, we had to find the right geometric features to discriminate the *skin sagging* along the jawline. The problem is exacerbated by the lack of reliable data to train the algorithms. Indeed, the most complicated therapeutic treatments or

[1]The correspondence problem is the problem of having different mesh structures (number of vertices, connectivity) between the 3d scans.

severe diseases are often rare and confidential, limiting the available data to train and validate the models. Moreover, the annotations provided by the physicians can be inconsistent because of the vague definition of symptoms or differences in perceptions and interpretation among experts, making it difficult to validate the models using only the data.

We found that the *surface curvature* is a relevant variable to discriminate jawline's skin sagging, and performs better than using point displacements. However, we combined point displacements with the curvature in our model to visualize this aging biomarker as a *deformation* on the face template. Ultimately, our model can automate skin sagging ratings before and after toxin injections to evaluate the patient's response to these injections. Our model can also simulate realistic aging deformations on the face. Our experiments demonstrate that human ratings contain many *rating inconsistencies* that can jeopardize clinical trials. We have demonstrated that we can limit the variability in the data labels using cross-validation and regularization. We used Partial Least Square Regression (PLSR) combined with cross-validation to filter the noise present in the labels and to increase the ability of our model to generalize to unseen data. PLSR is a simple *linear model* whose interpretability allows a better understanding of why the model makes a prediction. An interesting finding is that with PLSR, the model's performances are not much impacted by adding many features containing redundant information.

Entanglement of deformations in training data The third problem is *disentangling shape deformations from shape morphologies*. The typical disentangling question in the face modelling community is how to decompose a 3d face scan into facial expressions deformations and morphological deformations. One application to disentanglement would be to learn to *neutralize the expressions* on 3d scans and, this way, make the measurements done on 3d face scans invariant to facial expressions. State-of-the-art methods propose using two PCA to create subspaces for expressions and identity. However, it has been shown that the resulting subspaces are not independent, which dramatically limits such models' usefulness.

We bring a partial solution to this problem. In particular, we show that the two subspaces are not orthogonal. Then, we hypothesized that enforcing orthogonality between the subspaces should support the disentanglement. We formalized the baseline problem (joint-PCA) and added a penalization on the orthogonality. Our penalization leads to *quasi-orthogonal* expression and identity subspaces (Quasi-Orthogonal Joint PCA). We propose an iterative procedure for the penalized problem, solving two eigenvalue/eigenvector equations similar to a Jacobi relaxation problem. Solving an eigenvalue/eigenvector equation is a well-known problem, allowing for using fast linear solvers. We can obtain quasi-orthogonal linear bases learned through PCA, which was not possible prior to our work. Our experiments show that this quasi-orthogonalization improves the expression neutralization models, particularly for extreme expressions, such as grinning and opening the mouth.

5.2 Publications and industrial transfers

Our work has been valorized by multiple industrial transfers and scientific publications when the confidentiality of the results allowed it. I directly contributed to the industrial transfer by implementing all the concepts discussed in this thesis into QuantifiCare's codebase.

Scientific publications

- Jousse, F., Pennec, X., Delingette, H., & Gonzalez, M. (2021). Geodesic squared exponential kernel for non-rigid shape registration. Proceedings - 2021 16th IEEE International Conference on Automatic Face and Gesture Recognition, FG 2021.

- Jousse, F., Pennec, X., Delingette, H., & Bletterer, A. Quasi-orthogonality between expressions and identities in face morphable models as a penalized problem. In the process of submission.

Industrial transfers

• I co-developed the C++ statistical shape modeling library used at QuantifiCare, writing more than 7K C++ lines in the QuantifiCare code base. Experienced software developers have reviewed the code, attesting to its compliance with the industry code standards. The library implements all the concepts described in chapter 2: linear mesh deformation, Gaussian processes, kernels, low-rank approximations, Non-rigid ICP, weighted least squares, and visualization techniques.

• I co-developed a Python/C++ application used to estimate jawline sagging grades at QuantifiCare. A clinical study uses this application to evaluate the effect of toxin injections on more than 200 patients followed over several months. I developed all the methods described in chapter 3 and supervised the learning stage. In particular, I have implemented feature extraction (principal curvature, displacements), partial least square regression, template fitting, and visualization methods. The methodology has not been published for confidentiality reasons.

• I implemented a prototype to standardize facial expressions on QuantifiCare's 3d face scans. This Python code implements all the concepts and algorithms described in chapter 4. In particular, I implemented the dimensionality reduction methods (i.e., PCA, J-PCA, QOJ-PCA) and the Jacobi optimization to fit the model.

5.3 Final thoughts and future work

From non-rigid ICP to non-linear deformations The GPMMs and NICP provide a flexible and versatile framework that can be adapted to many registration scenarios and used as a starting point in many cases. However, there remain several limitations. The first is the inability to register shapes when non-linear deformations are involved. Although we have not encountered such a problem, this can be a fundamental limitation. For example, the deformation between a closed hand and an open hand involves non-linear motions and thus cannot be registered with these methods. A second possible limitation of GPMMs is that they can generate self-intersecting surfaces, creating non-manifold mesh structures, which we know cannot exist. This may occur when the template has areas of high curvature or when applying large deformations. The non-linearity can be implemented in the registration by replacing the concept of displacement fields used to deform the template mesh with velocity fields, leading to diffeomorphic deformations between the meshes even when the deformations are large and involve non-linearities. The concepts of velocity fields are, for example, implemented in the LDDMM framework [98], or in the Stationary Velocity Field (SVF) framework [99].

Improving 3d scans standardization Our experiments show that the 3d face scans may not be used for precise measurement even after the facial expression neutralization. This is because the facial expressions may have permanently altered the 3d scans, or the postural motions have changed the mesh too much. Conceptually, standardizing the 3d scans could be possible, but this may come with a loss of details, as any cleaning method would. There is a balance between expression neutralization and signal preservation to be found and defined. A significant improvement would be to extend our methodology to a 3d face scan "cleaning" framework allowing for cleaning reconstruction noise, standardizing postures, and neutralizing facial expressions.

Gathering more data for more powerful models The small amount of data available guided a part of the methods we chose. In particular, we developed models with few parameters and linear relationships to improve their interpretability. However, the small amount of data limits the potential for modeling delicate phenomena, such as wrinkles or more complex facial operations. Popular methods in machine learning, such as deep neural networks, are potent but prone to overfitting when there is insufficient data and lack some interpretability to support the results. An interesting background work would be collect more datasets of shapes and images and then to expertiment with more complex models.

Bridging the gap between meshes and textures This thesis does not study the modeling of images while we know that images have an essential role in dermatology. For example, the condition of certain wounds or lesions can be assessed directly on images. Some models, such as [25; 27], consider the texture information as a single colorimetric information per mesh vertex, losing much information in the images. In contrast, others do not model the texture information [26]. One direction of research could be to combine the information from the textures and meshes in an unique model.

Chapter A
Additional experiments

A.1 Inter-rater and intra-rater variability within 9 jawline sagging raters

One can wonder the amount of variability among the raters when rating jawline sagging. In this section, we provide additional results on the study about the rating variability within the group of 9 jawline sagging raters. In particular, we give confusions matrices when compared with the scores given by a consensus of raters. This allows to evaluate a potential variability among the raters. The conclusion is that we did not found any rater that is particularly better or worse than the others. The best raters still have a high intra and inter variability. This is true for the inter-raters errors and the intra-raters errors.

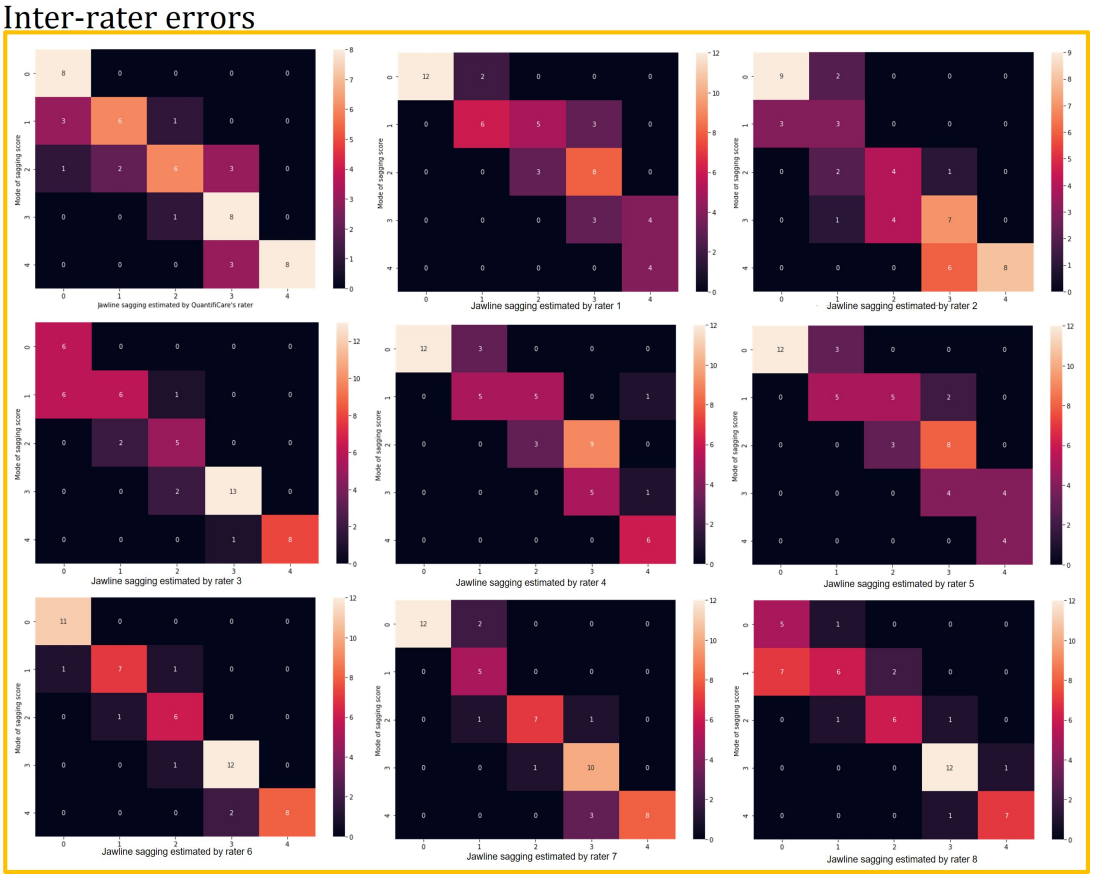


Figure A.1: Confusion matrices accounting for the inter-rater variability among the 9 raters. Each matrix shows the disagreements between one rater and the mode of sagging scores. In this case the mode is the most voted grade.

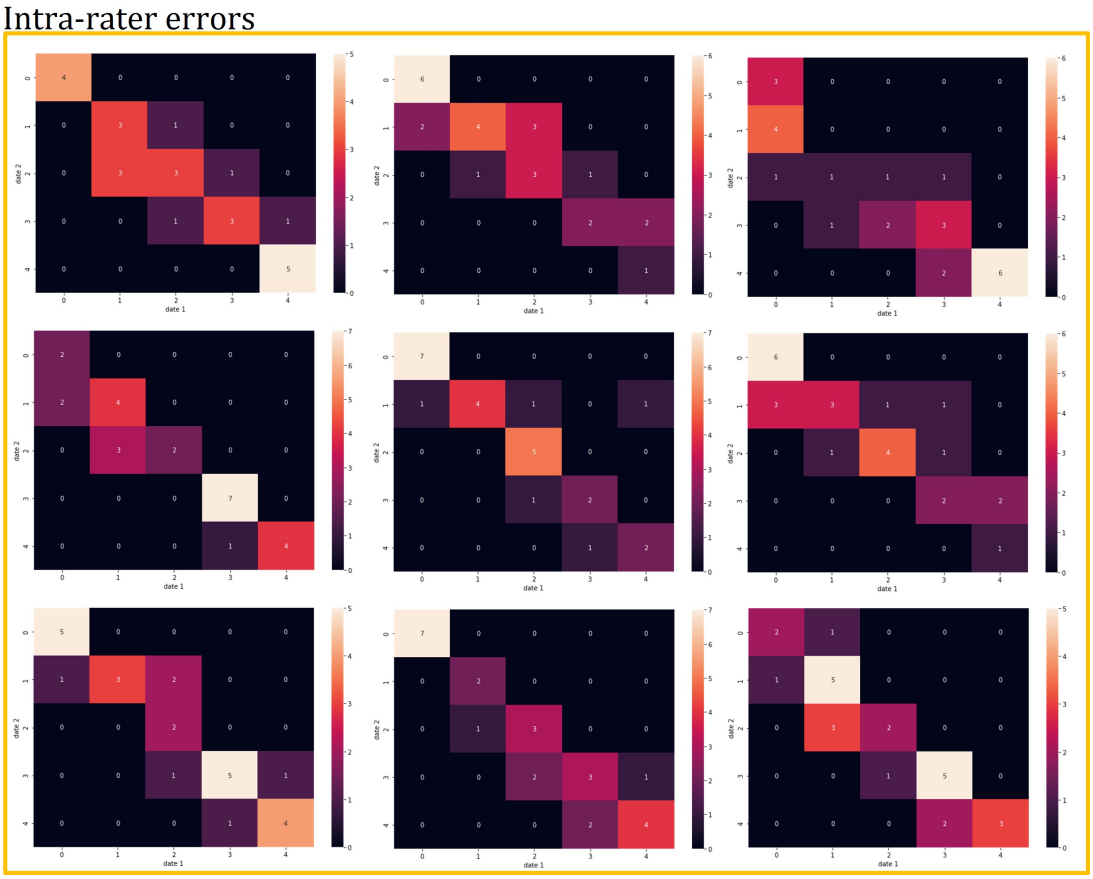


Figure A.2: Confusion matrices accounting for the intra-rater variability among the 9 raters. Each matrix shows the disagreement of a rater with itself after a period of 2 weeks.

A.2 Effect of facial expression neutralisation on facial volume measurements

In this section, we compare the effect of the facial expression neutralisation on different zones of the face. It was hypothetized that the expression neutralization might be more interesting in some areas of interest used to make measures in clinical trials. Unsurprisingly the effect of our neutralization method is more visible on small areas located around a facial expression. For example, the nasolabial folds are more impacted by a grin than the chin, thus the neutralisation is more beneficial in this area. But in general the tendency observed on the whole face as shown in chapter 4 is respected.

Mouth zone	Average volume error (mL)			Significance		
	Baseline	J-PCA	QOJ-PCA	Cohen's d	> probability	
3_mouth_stretch	15.1(±4.0)	8.5(±2.9)	3.8(±1.9)	3.57	97.1	Positive effect
13_lip_funneler	11.6(±3.5)	10.4(±3.3)	4.0(±1.8)	2.68	92.0	
12_lip_puckerer	12.9(±4.2)	9.6(±3.2)	4.3(±2.0)	2.59	91.5	
16_grin	9.9(±3.4)	8.9(±3.1)	3.6(±1.7)	2.35	89.2	
10_dimpler	8.4(±3.0)	9.4(±3.2)	3.2(±1.7)	2.12	86.4	
17_cheek_blowing	10.6(±4.2)	10.7(±4.2)	4.0(±1.9)	2.03	86.2	
11_chin_raiser	8.1(±2.7)	5.9(±2.2)	4.2(±2.0)	1.65	79.8	
7_jaw_forward	7.7(±3.0)	5.3(±2.5)	4.2(±1.8)	1.43	77.0	
2_smile	6.3(±2.8)	7.6(±2.9)	3.0(±1.7)	1.41	76.6	
9_mouth_right	5.6(±1.9)	4.8(±1.9)	3.3(±1.7)	1.28	73.9	
15_lip_roll	5.6(±1.9)	5.2(±2.2)	3.5(±1.7)	1.17	72.2	
8_mouth_left	5.3(±2.0)	4.1(±2.0)	3.3(±1.6)	1.12	71.3	
6_jaw_right	4.8(±1.9)	3.6(±2.0)	3.5(±1.7)	0.74	64.4	
5_jaw_left	5.0(±4.4)	3.6(±4.6)	3.5(±3.0)	0.39	57.8	
14_sadness	4.0(±2.3)	3.4(±1.9)	3.3(±1.7)	0.35	57.1	
20_brow_lower	3.0(±2.0)	3.3(±1.8)	2.9(±1.7)	0.07	51.4	Not significant
4_anger	3.0(±1.7)	3.0(±2.2)	3.7(±13.5)	-0.08	48.0	
19_brow_raiser	3.0(±2.2)	3.0(±2.1)	3.4(±1.8)	-0.21	45.8	Negative effect
18_eye_closed	2.3(±1.9)	2.5(±1.7)	2.7(±1.6)	-0.25	45.1	
1_neutral	0.0(±0.0)	1.2(±1.8)	2.9(±10.5)	-0.39	39.1	

Table A.1: This table compares the effect of our QOJ-PCA expression neutralization on the volume measurements (on the mouth zone). Baseline corresponds to the (absolute) average volume generated by a specific facial expression on a neutral face. The significance tests are computed between the QOJ-PCA column and the baseline column. We give the neutralization results with J-PCA, but they do not impact the significance columns.

Nasolabial	Average volume error (mL)			Significance		
Folds	Baseline	J-PCA	QOJ-PCA	d-effect	> probability	
17_cheek_blowing	11.0(±4.0)	5.9(±3.0)	4.1(±1.8)	2.21	88.1	Positive effect
3_mouth_stretch	5.9(±2.3)	6.1(±1.9)	2.8(±1.7)	1.55	78.3	
16_grin	5.8(±2.4)	5.3(±1.6)	3.0(±1.5)	1.38	75.9	
13_lip_funneler	4.8(±2.0)	4.9(±2.2)	2.7(±1.5)	1.17	72.3	
12_lip_puckerer	4.7(±2.0)	4.7(±1.9)	2.8(±1.3)	1.16	72.2	
10_dimpler	4.4(±2.2)	5.0(±2.1)	2.6(±1.5)	0.93	68.2	
6_jaw_right	3.8(±1.9)	2.6(±1.5)	2.8(±1.4)	0.57	61.3	
9_mouth_right	3.7(±1.6)	2.8(±1.4)	2.8(±1.5)	0.56	61.0	
8_mouth_left	3.6(±2.0)	2.6(±1.5)	2.7(±1.5)	0.54	60.7	
11_chin_raiser	3.5(±1.9)	3.1(±1.6)	2.7(±1.4)	0.47	59.4	
2_smile	3.1(±2.0)	4.3(±2.0)	2.4(±1.5)	0.39	57.8	
5_jaw_left	4.2(±4.0)	2.8(±3.1)	2.9(±2.6)	0.38	57.8	
15_lip_roll	3.0(±1.4)	3.1(±1.3)	2.6(±1.4)	0.28	55.5	
7_jaw_forward	3.1(±1.8)	3.0(±1.5)	2.8(±1.4)	0.21	54.1	
19_brow_raiser	2.4(±1.8)	2.1(±1.5)	2.3(±1.4)	0.07	51.3	Not significant
20_brow_lower	2.2(±1.5)	2.2(±1.4)	2.3(±1.4)	-0.12	47.5	
14_sadness	2.2(±1.7)	2.2(±1.4)	2.4(±1.4)	-0.15	47.0	
4_anger	2.0(±1.0)	2.2(±1.7)	2.6(±4.4)	-0.18	45.7	
18_eye_closed	1.7(±1.6)	1.6(±1.3)	2.2(±1.4)	-0.35	43.1	Negative effect
1_neutral	0.0(±0.0)	0.8(±1.1)	1.9(±3.5)	-0.78	29.1	

Table A.2: This table compares the effect of our QOJ-PCA expression neutralization on the volume measurements (on the nasolabial folds zone). Baseline corresponds to the (absolute) average volume generated by a specific facial expression on a neutral face. The significance tests are computed between the QOJ-PCA column and the baseline column. We give the neutralization results with J-PCA, but they do not impact the significance columns.

Chin	Average volume error (mL)			Significance		
area	Baseline	J-PCA	QOJ-PCA	d-effect	> probability	
3_mouth_stretch	24.1(±8.0)	21.9(±7.0)	6.9(±4.3)	2.68	91.9	Positive effect
15_lip_roll	10.3(±4.4)	6.3(±3.1)	6.1(±3.2)	1.09	71.0	
13_lip_funneler	9.4(±5.3)	6.7(±3.7)	5.7(±3.7)	0.80	65.8	
6_jaw_right	7.4(±3.2)	5.6(±2.9)	5.7(±3.1)	0.54	60.6	
16_grin	7.5(±4.1)	6.5(±3.8)	5.7(±4.3)	0.41	58.2	
11_chin_raiser	9.1(±5.2)	7.9(±4.0)	7.4(±4.1)	0.35	56.9	
5_jaw_left	7.5(±4.8)	5.4(±5.5)	6.1(±5.4)	0.29	55.7	
7_jaw_forward	12.2(±5.3)	10.1(±4.6)	11.1(±5.2)	0.21	54.2	
9_mouth_right	5.7(±3.6)	5.5(±3.9)	5.3(±3.9)	0.12	52.4	Not significant
8_mouth_left	5.8(±3.5)	5.4(±3.4)	5.4(±3.6)	0.11	52.3	
17_cheek_blowing	5.8(±3.5)	6.8(±3.8)	5.5(±3.6)	0.07	51.4	
12_lip_puckerer	6.2(±4.1)	6.1(±3.6)	5.9(±3.8)	0.07	51.3	
14_sadness	5.1(±3.7)	5.1(±3.6)	5.0(±3.6)	0.03	50.6	
2_smile	3.7(±2.8)	5.3(±3.1)	3.8(±2.9)	-0.01	49.8	
4_anger	4.8(±2.7)	4.6(±3.5)	5.5(±13.9)	-0.07	48.2	
18_eye_closed	3.2(±2.9)	3.5(±3.1)	3.8(±3.2)	-0.17	46.5	
19_brow_raiser	3.8(±2.9)	4.0(±3.1)	4.3(±3.1)	-0.17	46.5	
10_dimpler	5.9(±3.6)	7.5(±3.8)	6.7(±3.9)	-0.19	46.1	
20_brow_lower	4.9(±2.7)	4.4(±2.7)	5.5(±3.0)	-0.22	45.7	Negative effect
1_neutral	0.0(±0.0)	1.8(±2.6)	3.0(±11.7)	-0.36	39.9	

Table A.3: This table compares the effect of our QOJ-PCA expression neutralization on the volume measurements (on the chin zone). Baseline corresponds to the (absolute) average volume generated by a specific facial expression on a neutral face. The significance tests are computed between the QOJ-PCA column and the baseline column. We give the neutralization results with J-PCA, but they do not impact the significance columns.

Cheeks	Average volume error (mL)			Significance		
area	Baseline	J-PCA	QOJ-PCA	d-effect	> probability	
16_grin	32.7(±13.2)	25.2(±10.3)	18.3(±9.5)	1.25	73.7	Positive effect
3_mouth_stretch	32.4(±13.4)	40.7(±12.2)	18.6(±9.8)	1.17	72.3	
17_cheek_blowing	39.3(±12.4)	22.3(±9.9)	27.6(±11.0)	1.00	69.2	
10_dimpler	23.8(±11.7)	21.9(±10.5)	16.7(±9.4)	0.67	63.1	
8_mouth_left	22.0(±10.0)	17.2(±9.1)	17.1(±9.6)	0.51	60.1	
9_mouth_right	21.5(±10.2)	16.8(±9.6)	17.7(±9.3)	0.39	57.7	
13_lip_funneler	20.4(±11.2)	18.6(±9.0)	16.4(±9.3)	0.38	57.6	
7_jaw_forward	23.9(±11.3)	24.0(±10.7)	20.2(±9.4)	0.36	57.2	
11_chin_raiser	21.6(±11.9)	21.0(±11.6)	17.7(±9.5)	0.36	57.2	
6_jaw_right	20.2(±9.0)	16.4(±8.0)	18.2(±8.8)	0.22	54.4	
12_lip_puckerer	19.4(±10.0)	17.8(±9.0)	17.6(±9.4)	0.18	53.7	Not significant
5_jaw_left	22.3(±20.0)	17.6(±15.6)	19.1(±15.4)	0.18	53.7	
2_smile	15.7(±11.0)	17.1(±10.4)	13.9(±10.2)	0.17	53.4	
15_lip_roll	17.5(±9.3)	16.8(±8.4)	16.5(±8.8)	0.11	52.2	
19_brow_raiser	12.3(±10.7)	12.1(±10.3)	13.7(±9.5)	-0.14	47.2	
14_sadness	13.4(±10.5)	13.9(±9.4)	15.0(±9.3)	-0.16	46.8	
20_brow_lower	15.7(±10.6)	13.3(±9.0)	18.2(±9.8)	-0.25	45.1	Negative effect
4_anger	14.6(±7.4)	13.1(±9.8)	17.5(±13.6)	-0.27	44.4	
18_eye_closed	9.5(±9.9)	10.1(±8.9)	13.0(±9.0)	-0.37	42.6	
1_neutral	0.0(±0.0)	4.6(±6.9)	10.2(±13.8)	-1.05	22.9	

Table A.4: This table compares the effect of our QOJ-PCA expression neutralization on the volume measurements (on the cheeks zone). Baseline corresponds to the (absolute) average volume generated by a specific facial expression on a neutral face. The significance tests are computed between the QOJ-PCA column and the baseline column. We give the neutralization results with J-PCA, but they do not impact the significance columns.

Chapter B
Notations

B.1 General notations

- Matrices are always bold upper-case letters (eg. $\mathbf{X}$).
- Column vectors are bold lower-case letters (eg. $\mathbf{x_i}$).
- Row vectors are bold lower-case letters and a transpose sign (eg. $\mathbf{x_i}^\intercal$).
- Scalars are simply lower-case letters (eg. y).

B.2 Notations for Chapter 3

Warnings:

- A column vector can be a row of a matrix. For example, $\mathbf{w_i}$ is the **row** index i of $\mathbf{W}$ but $\mathbf{w_i}$ is still a column vector (i.e $(17n \times 1)$.).

Matrices dimensions:

Matrix name	Notations	Exponents
Feature matrix	$\mathbf{X} \in \mathbb{R}^{m \times 17n}$	m is the number of training faces
Sagging scores vector	$\mathbf{y} \in \mathbb{R}^{m \times 1}$	n is the number of mesh vertices
PCA loadings[1]	$\mathbf{P} \in \mathbb{R}^{l \times 17n}$	l is the number of latent variables
PCA scores [2]	$\mathbf{Z} = \mathbf{P}\mathbf{X}^\intercal \in \mathbb{R}^{l \times m}$	
PLS X-weights	$\mathbf{W} \in \mathbb{R}^{l \times 17n}$	
PLS X-loadings	$\mathbf{P} \in \mathbb{R}^{l \times 17n}$	
PLS X-scores	$\mathbf{Z} = \mathbf{P}\mathbf{X}^\intercal \in \mathbb{R}^{l \times m}$	
PLS prediction coefs.	$\mathbf{q} \in \mathbb{R}^{17n \times 1}$	
Covariance of features	$\mathbf{X}^\intercal\mathbf{X} \in \mathbb{R}^{17n \times 17n}$	
Covariance of latent vars.	$\mathbf{Z}\mathbf{Z}^\intercal \in \mathbb{R}^{l \times l}$	

B.3 Notations for Chapter 4

Warnings:

- The data matrices are not arranged in the same way as it was in the previous chapter. (i.e here the faces are rows of the data matrices)

Matrices dimensions:

[1] aka. Transformation matrix
[2] aka. Transformed data

Matrix name	Notations	Exponents
Subject matrix	$\mathbf{X} \in \mathbb{R}^{3n \times m}$	m is the number of subjects
Expressions matrix	$\mathbf{Y} \in \mathbb{R}^{3n \times km}$	n is the number of mesh vertices
Subjects loadings[1]	$\mathbf{P} \in \mathbb{R}^{l_{exp} \times 3n}$	k is the number of expressions
Expressions loadings[2]	$\mathbf{Q} \in \mathbb{R}^{l_{subj} \times 3n}$	k is the number of expressions
Subjects scores	$\mathbf{Z}_{subj} = \mathbf{PX} \in \mathbb{R}^{l_{subj} \times m}$	l_{subj} is the dimension of the subject latent space
Expressions scores	$\mathbf{Z}_{exp} = \mathbf{QY} \in \mathbb{R}^{l_{exp} \times km}$	l_{expr} is the dimension of the expression latent space

Chapter C
Derivatives

To compute derivatives of matrix and vector valued functions it is safer to start from the definitions of gradient and directional derivatives rather than using matrix derivative formulas. We notice in fact that it is enough to write the Taylor expansion to the order one to get the directional derivatives and the gradient of a particular function. The calculation of the derivative is only a matter of developing a term of the form: $f(x+\epsilon\dot{x})$. In the following we give the detailed calculations for the derivation of the cost functions encountered in this thesis.

C.1 Gradients for ridge regression

Problem statement We want to compute the gradient of a function $\mathbf{E}$ that writes:

$$\begin{aligned} \mathbf{E}(\boldsymbol{\alpha}) : \mathbb{R}^l &\to \mathbb{R} & \mathbf{A} &\in \mathbb{R}^{3n\times l} \\ \boldsymbol{\alpha} &\mapsto ||\mathbf{y}-\mathbf{A}\boldsymbol{\alpha}||^2 + \rho||\boldsymbol{\alpha}||^2 & \boldsymbol{\alpha} &\in \mathbb{R}^l \\ & & \mathbf{y} &\in \mathbb{R}^{3n} \end{aligned}$$

Gradient definition In this case, the gradient is a vector valued function noted: $\nabla\mathbf{E}(\boldsymbol{\alpha}) : \mathbb{R}^l \to \mathbb{R}^l$. We can define the gradient similarly to derivatives using limits:

$$\nabla\mathbf{E}(\boldsymbol{\alpha}) = \begin{bmatrix} \lim\limits_{\epsilon\to 0} \frac{\mathbf{E}(\boldsymbol{\alpha}+\mathbf{e}_1\epsilon)-\mathbf{E}(\boldsymbol{\alpha})}{\epsilon} \\ \vdots \\ \lim\limits_{\epsilon\to 0} \frac{\mathbf{E}(\boldsymbol{\alpha}+\mathbf{e}_l h)-\mathbf{E}(\boldsymbol{\alpha})}{\epsilon} \end{bmatrix}, \quad \text{where the } \mathbf{e}_i \text{ are standard basis vectors of } \mathbb{R}^l. \tag{C.1}$$

Gradient computation Computing the gradient consists in computing the limits in (C.1). There is an undertermination of the type $\lim\limits_{\epsilon\to 0}\frac{0}{0}$, that can be solved by developing the terms inside the limit. For the first term, we have:

$$\begin{aligned} \mathbf{E}(\boldsymbol{\alpha}) &= (\mathbf{y}-\mathbf{A}\boldsymbol{\alpha})^\intercal(\mathbf{y}-\mathbf{A}\boldsymbol{\alpha}) + \rho\boldsymbol{\alpha}^\intercal\boldsymbol{\alpha} \\ &= \mathbf{y}^\intercal\mathbf{y} - \mathbf{y}^\intercal\mathbf{A}\boldsymbol{\alpha} + (\mathbf{A}\boldsymbol{\alpha})^\intercal\mathbf{A}\boldsymbol{\alpha} - (\mathbf{A}\boldsymbol{\alpha})^\intercal\mathbf{y} + \rho\boldsymbol{\alpha}^\intercal\boldsymbol{\alpha}. \end{aligned} \tag{C.2}$$

In the previous line, each term is a scalar (e.g. $\mathbf{y}^\intercal\mathbf{y}$ is a scalar) and for a scalar $a \in \mathbb{R}$ with have $a^\intercal = a$. Thus we can write:

$$\begin{aligned} \mathbf{E}(\boldsymbol{\alpha}) &= \mathbf{y}^\intercal\mathbf{y} - \mathbf{y}^\intercal\mathbf{A}\boldsymbol{\alpha} + (\mathbf{A}\boldsymbol{\alpha})^\intercal\mathbf{A}\boldsymbol{\alpha} - \mathbf{y}^\intercal\mathbf{A}\boldsymbol{\alpha} + \rho\boldsymbol{\alpha}^\intercal\boldsymbol{\alpha} \\ &= \mathbf{y}^\intercal\mathbf{y} - 2\mathbf{y}^\intercal\mathbf{A}\boldsymbol{\alpha} + (\mathbf{A}\boldsymbol{\alpha})^\intercal\mathbf{A}\boldsymbol{\alpha} + \rho\boldsymbol{\alpha}^\intercal\boldsymbol{\alpha} \end{aligned} \tag{C.3}$$

For the other term we have:

$$\begin{aligned}\mathbf{E}(\boldsymbol{\alpha}+\mathbf{e}_1\epsilon) &= \mathbf{y}^\intercal\mathbf{y} - 2\mathbf{y}^\intercal\mathbf{A}(\boldsymbol{\alpha}+\mathbf{e}_1\epsilon) + ((\mathbf{A}(\boldsymbol{\alpha}+\mathbf{e}_1\epsilon))^\intercal\,\mathbf{A}(\boldsymbol{\alpha}+\mathbf{e}_1\epsilon)\\ &\quad + \rho(\boldsymbol{\alpha}+\mathbf{e}_1\epsilon)^\intercal(\boldsymbol{\alpha}+\mathbf{e}_1\epsilon)\\ &= \mathbf{y}^\intercal\mathbf{y} - 2\mathbf{y}^\intercal\mathbf{A}\boldsymbol{\alpha} - 2\mathbf{y}^\intercal\mathbf{A}\mathbf{e}_1\epsilon + (\boldsymbol{\alpha}+\mathbf{e}_1\epsilon)^\intercal\mathbf{A}^\intercal\mathbf{A}(\boldsymbol{\alpha}+\mathbf{e}_1\epsilon)\\ &\quad + \rho(\boldsymbol{\alpha}+\mathbf{e}_1\epsilon)^\intercal(\boldsymbol{\alpha}+\mathbf{e}_1\epsilon)\\ &= \mathbf{y}^\intercal\mathbf{y} - 2\mathbf{y}^\intercal\mathbf{A}\boldsymbol{\alpha} - 2\mathbf{y}^\intercal\mathbf{A}\mathbf{e}_1\epsilon + \boldsymbol{\alpha}^\intercal\mathbf{A}^\intercal\mathbf{A}\boldsymbol{\alpha} + \boldsymbol{\alpha}^\intercal\mathbf{A}^\intercal\mathbf{A}\mathbf{e}_1\epsilon\\ &\quad + (\mathbf{e}_1\epsilon)^\intercal\mathbf{A}^\intercal\mathbf{A}\boldsymbol{\alpha} + (\mathbf{e}_1\epsilon)^\intercal\mathbf{A}^\intercal\mathbf{A}\mathbf{e}_1\epsilon + \rho\boldsymbol{\alpha}^\intercal\boldsymbol{\alpha} + \rho\boldsymbol{\alpha}^\intercal\mathbf{e}_1\epsilon\\ &\quad + \rho(\mathbf{e}_1\epsilon)^\intercal\boldsymbol{\alpha} + \rho(\mathbf{e}_1\epsilon)^\intercal\mathbf{e}_1\epsilon.\end{aligned} \tag{C.4}$$

Then, we subtract $\mathbf{E}(\boldsymbol{\alpha}+\mathbf{e}_1\epsilon)$ and $\mathbf{E}(\boldsymbol{\alpha})$ to get:

$$\begin{aligned}\mathbf{E}(\boldsymbol{\alpha}+\mathbf{e}_1\epsilon) - \mathbf{E}(\boldsymbol{\alpha}) &= -2\mathbf{y}^\intercal\mathbf{A}\mathbf{e}_1\epsilon + 2\boldsymbol{\alpha}^\intercal\mathbf{A}^\intercal\mathbf{A}\mathbf{e}_1\epsilon + (\mathbf{e}_1\epsilon)^\intercal\mathbf{A}^\intercal\mathbf{A}\mathbf{e}_1\epsilon\\ &\quad + 2\rho\boldsymbol{\alpha}^\intercal\mathbf{e}_1\epsilon + \rho(\mathbf{e}_1\epsilon)^\intercal\mathbf{e}_1\epsilon\end{aligned} \tag{C.5}$$

Here we notice that the terms that are not functions of h have disappeared as expected. Then, the first limit writes:

$$\lim_{\epsilon\to 0}\frac{\mathbf{E}(\boldsymbol{\alpha}+\mathbf{e}_1\epsilon)-\mathbf{E}(\boldsymbol{\alpha})}{\epsilon} = -2\mathbf{y}^\intercal\mathbf{A}\mathbf{e}_1 + 2\boldsymbol{\alpha}^\intercal\mathbf{A}^\intercal\mathbf{A}\mathbf{e}_1 + 2\rho\boldsymbol{\alpha}^\intercal\mathbf{e}_1. \tag{C.6}$$

We do the same for the other limits to get the gradient:

$$\nabla\mathbf{E}(\boldsymbol{\alpha}) = \begin{bmatrix} -2\mathbf{y}^\intercal\mathbf{A}\mathbf{e}_1 + 2\boldsymbol{\alpha}^\intercal\mathbf{A}^\intercal\mathbf{A}\mathbf{e}_1 + 2\rho\boldsymbol{\alpha}^\intercal\mathbf{e}_1 \\ \vdots \\ -2\mathbf{y}^\intercal\mathbf{A}\mathbf{e}_l + 2\boldsymbol{\alpha}^\intercal\mathbf{A}^\intercal\mathbf{A}\mathbf{e}_l + 2\rho\boldsymbol{\alpha}^\intercal\mathbf{e}_l \end{bmatrix} \tag{C.7}$$

Since $(\mathbf{e}_1, ..., \mathbf{e}_l)$ are standard basis vectors we can rewrite the gradient into matrix notations:

$$\boxed{\begin{aligned}\nabla\mathbf{E}(\boldsymbol{\alpha}) &= [-2\mathbf{y}^\intercal\mathbf{A} + 2\boldsymbol{\alpha}^\intercal\mathbf{A}^\intercal\mathbf{A} + 2\rho\boldsymbol{\alpha}^\intercal]^\intercal\,\mathbf{I}_l\\ &= -2\mathbf{A}^\intercal\mathbf{y} + 2\mathbf{A}^\intercal\mathbf{A}\boldsymbol{\alpha} + 2\rho\boldsymbol{\alpha},\end{aligned}} \tag{C.8}$$

where $\mathbf{A}\in\mathbb{R}^{3n\times l}$ is a matrix and the other terms $\mathbf{A}^\intercal\mathbf{y}\in\mathbb{R}^l$, $\mathbf{A}^\intercal\mathbf{A}\boldsymbol{\alpha}\in\mathbb{R}^l$, $\boldsymbol{\alpha}\in\mathbb{R}^l$ are vectors of dimension l, which is consistent with our definition of gradient.

C.2 Gradients for PCA

We want to compute the gradient of a Lagrangian function that writes:

$$\mathcal{L}(\mathbf{P},\boldsymbol{\Lambda}) = \mathrm{Tr}\left(\mathbf{P}\mathbf{X}^\intercal\mathbf{X}\mathbf{P}^\intercal\right) - \mathrm{Tr}\left(\boldsymbol{\Lambda}\mathbf{P}\mathbf{P}^\intercal\right), \quad \begin{aligned}&\text{where} \quad \mathbf{P}\in\mathbb{R}^{l\times 17n}, \quad \boldsymbol{\Lambda}\in\mathbb{R}^{l\times l}\\ &\text{and} \quad \mathbf{X}\in\mathbb{R}^{17n\times m}.\end{aligned} \tag{C.9}$$

We write its gradient as two functions:

$$\begin{aligned}&\nabla_{\mathbf{P}}\mathcal{L}(\mathbf{P},\boldsymbol{\Lambda}) \quad \text{that is the gradient with respect to } \mathbf{P}\\ \text{and} \quad &\nabla_{\boldsymbol{\Lambda}}\mathcal{L}(\mathbf{P},\boldsymbol{\Lambda}) \quad \text{that is the gradient with respect to } \boldsymbol{\Lambda}.\end{aligned} \tag{C.10}$$

Gradient definition (with respect to a matrix) We introduce $\dot{\mathbf{P}} \in \mathbb{R}^{l\times 17n}$ (notice the dot): a matrix with the same dimension as $\mathbf{P} \in \mathbb{R}^{l\times 17n}$ with only one element that is one and the others are zeros. This allows to define a directional derivative (in the direction of the element that is not zero) as the limit:

$$\nabla_{\dot{\mathbf{P}}}\mathcal{L} = \lim_{\epsilon\to 0} \frac{\mathcal{L}(\mathbf{P}+\epsilon\dot{\mathbf{P}},\mathbf{\Lambda}) - \mathcal{L}(\mathbf{P},\mathbf{\Lambda})}{\epsilon} \quad \text{and} \quad \nabla_{\dot{\mathbf{\Lambda}}}\mathcal{L} = \lim_{\epsilon\to 0} \frac{\mathcal{L}(\mathbf{P},\mathbf{\Lambda}+\epsilon\dot{\mathbf{\Lambda}}) - \mathcal{L}(\mathbf{P},\mathbf{\Lambda})}{\epsilon}.$$

From this, we can define the gradient with respect to $\mathbf{P}$ as the matrix whose elements are the directional derivative $\nabla_{\dot{\mathbf{P}}}\mathcal{L}$, the same reasoning applies to $\nabla_{\dot{\mathbf{\Lambda}}}$.

Computation of $\nabla_{\mathbf{P}}\mathcal{L}(\mathbf{P},\mathbf{\Lambda})$ To evaluate the limit that has an indetermination of the form $\frac{0}{0}$, we develop the following terms:

$$\begin{aligned}
\mathcal{L}\left(\mathbf{P}+\epsilon\dot{\mathbf{P}},\mathbf{\Lambda}\right) &= \mathrm{Tr}\Big((\mathbf{P}+\epsilon\dot{\mathbf{P}})(\mathbf{X}^\intercal\mathbf{X})(\mathbf{P}+\epsilon\dot{\mathbf{P}})^\intercal\Big) - \mathrm{Tr}\Big(\mathbf{\Lambda}(\mathbf{P}+\epsilon\dot{\mathbf{P}})(\mathbf{P}+\epsilon\dot{\mathbf{P}})^\intercal\Big) \\
&= \mathrm{Tr}\Big(\mathbf{P}\mathbf{X}^\intercal\mathbf{X}\mathbf{P}^\intercal + \mathbf{P}\mathbf{X}^\intercal\mathbf{X}\epsilon\dot{\mathbf{P}}^\intercal + \epsilon\dot{\mathbf{P}}\mathbf{X}^\intercal\mathbf{X}\mathbf{P}^\intercal + \epsilon\dot{\mathbf{P}}\mathbf{X}^\intercal\mathbf{X}\epsilon\dot{\mathbf{P}}^\intercal - \mathbf{\Lambda}\mathbf{P}\mathbf{P}^\intercal \\
&\qquad - \mathbf{\Lambda}\mathbf{P}\epsilon\dot{\mathbf{P}}^\intercal - \mathbf{\Lambda}\epsilon\dot{\mathbf{P}}\mathbf{P}^\intercal - \mathbf{\Lambda}\epsilon\dot{\mathbf{P}}\epsilon\dot{\mathbf{P}}^\intercal\Big),
\end{aligned}$$

$$\begin{aligned}
\mathcal{L}\left(\mathbf{P}+\epsilon\dot{\mathbf{P}},\mathbf{\Lambda}\right) - \mathcal{L}\left(\mathbf{P},\mathbf{\Lambda}\right) = \mathrm{Tr}\Big(&- \mathbf{P}\mathbf{X}^\intercal\mathbf{X}\epsilon\dot{\mathbf{P}}^\intercal + \epsilon\dot{\mathbf{P}}\mathbf{X}^\intercal\mathbf{X}\mathbf{P}^\intercal + \epsilon\dot{\mathbf{P}}\mathbf{X}^\intercal\mathbf{X}\epsilon\dot{\mathbf{P}}^\intercal - \mathbf{\Lambda}\mathbf{P}\epsilon\dot{\mathbf{P}}^\intercal \\
&- \mathbf{\Lambda}\epsilon\dot{\mathbf{P}}\mathbf{P}^\intercal - \mathbf{\Lambda}\epsilon\dot{\mathbf{P}}\epsilon\dot{\mathbf{P}}^\intercal\Big),
\end{aligned}$$

which gives the first directionnal derivative:

$$\begin{aligned}
\nabla_{\dot{\mathbf{P}}}\mathcal{L} &= \lim_{\epsilon\to 0}\frac{1}{\epsilon}\mathrm{Tr}\Big(\mathbf{P}\mathbf{X}^\intercal\mathbf{X}\epsilon\dot{\mathbf{P}}^\intercal + \epsilon\dot{\mathbf{P}}\mathbf{X}^\intercal\mathbf{X}\mathbf{P}^\intercal + \epsilon\dot{\mathbf{P}}\mathbf{X}^\intercal\mathbf{X}\epsilon\dot{\mathbf{P}}^\intercal \\
&\quad - \mathbf{\Lambda}\mathbf{P}\epsilon\dot{\mathbf{P}}^\intercal - \mathbf{\Lambda}\epsilon\dot{\mathbf{P}}\mathbf{P}^\intercal - \mathbf{\Lambda}\epsilon\dot{\mathbf{P}}\epsilon\dot{\mathbf{P}}^\intercal\Big) \\
&= \mathrm{Tr}\left(\mathbf{P}\mathbf{X}^\intercal\mathbf{X}\dot{\mathbf{P}}^\intercal + \dot{\mathbf{P}}\mathbf{X}^\intercal\mathbf{X}\mathbf{P}^\intercal - \mathbf{\Lambda}\mathbf{P}\dot{\mathbf{P}}^\intercal - \mathbf{\Lambda}\dot{\mathbf{P}}\mathbf{P}^\intercal\right) \\
&= \mathrm{Tr}\left(2\mathbf{P}\mathbf{X}^\intercal\mathbf{X}\dot{\mathbf{P}}^\intercal - 2\mathbf{\Lambda}\mathbf{P}\dot{\mathbf{P}}^\intercal\right),
\end{aligned} \tag{C.11}$$

and finaly the gradient with respect to $\mathbf{P}$: $\boxed{\nabla_{\mathbf{P}}\mathcal{L}(\mathbf{P},\mathbf{\Lambda}) = 2\mathbf{P}\mathbf{X}^\intercal\mathbf{X} - 2\mathbf{\Lambda}\mathbf{P}}$ (C.12)

Computation of $\nabla_{\mathbf{\Lambda}}\mathcal{L}(\mathbf{P},\mathbf{\Lambda})$ Similarly, we develop the terms:

$$\mathcal{L}\left(\mathbf{P},\mathbf{\Lambda}+\epsilon\dot{\mathbf{\Lambda}}\right) - \mathcal{L}(\mathbf{P},\mathbf{\Lambda}) = -\mathrm{Tr}(\epsilon\dot{\mathbf{\Lambda}}\mathbf{P}\mathbf{P}^\intercal), \tag{C.13}$$

which gives the directional derivative:

$$\begin{aligned}
\nabla_{\dot{\mathbf{\Lambda}}}\mathcal{L} &= \lim_{\epsilon\to 0} -\frac{\mathrm{Tr}(\epsilon\dot{\mathbf{\Lambda}}\mathbf{P}\mathbf{P}^\intercal)}{\epsilon} \\
&= -\mathrm{Tr}(\dot{\mathbf{\Lambda}}\mathbf{P}\mathbf{P}^\intercal)
\end{aligned} \quad \text{and the gradient:} \quad \boxed{\nabla_{\mathbf{\Lambda}}\mathcal{L}(\mathbf{P},\mathbf{\Lambda}) = -\mathbf{P}\mathbf{P}^\intercal} \tag{C.14}$$

C.3 Gradients for PLS

We want to optimise a Lagrangian function that writes:

$$\mathcal{L}(\mathbf{w}_1, \lambda_1) = \mathrm{Tr}[\mathbf{y}\mathbf{w}_1^\intercal \mathbf{X}^\intercal \mathbf{X}\mathbf{w}_1\mathbf{y}^\intercal] - \lambda_1 \mathbf{w}_1^\intercal \mathbf{w}_1. \tag{C.15}$$

Computation of $\nabla_{\mathbf{w}_1}\mathcal{L}$: We develop $\mathcal{L}(\mathbf{w}_1 + \epsilon\dot{\mathbf{w}}_1, \lambda_1)$ and we remove the terms that are not linearly proportional to $\epsilon\dot{w}$ (because they will not be involved in the gradient):

$$\begin{aligned}\mathcal{L}(\mathbf{w}_1 + \epsilon\dot{\mathbf{w}}_1, \lambda_1) &= \mathrm{Tr}\left[\mathbf{y}(\mathbf{w}_1 + \epsilon\dot{\mathbf{w}}_1)^\intercal \mathbf{X}^\intercal \mathbf{X}(\mathbf{w}_1 + \epsilon\dot{\mathbf{w}}_1)\mathbf{y}^\intercal\right] - \lambda(\mathbf{w}_1 + \epsilon\dot{\mathbf{w}}_1)^\intercal(\mathbf{w}_1 + \epsilon\dot{\mathbf{w}}_1) \\ &= \mathrm{Tr}\left[\cancel{\mathbf{y}\mathbf{w}_1^\intercal \mathbf{X}^\intercal \mathbf{X}\mathbf{w}_1\mathbf{y}^\intercal} + \mathbf{y}\mathbf{w}_1^\intercal \mathbf{X}^\intercal \mathbf{X}\epsilon\dot{\mathbf{w}}_1\mathbf{y}^\intercal + \mathbf{y}\epsilon\dot{\mathbf{w}}_1{}^\intercal \mathbf{X}^\intercal \mathbf{X}\mathbf{w}_1\mathbf{y}^\intercal + \cancel{\mathbf{y}\epsilon\dot{\mathbf{w}}_1{}^\intercal \mathbf{X}^\intercal \mathbf{X}\epsilon\dot{\mathbf{w}}_1\mathbf{y}^\intercal}\right] \\ &\quad - 2\lambda\epsilon\dot{\mathbf{w}}_1{}^\intercal\mathbf{w}_1 - \cancel{\lambda\mathbf{w}_1^\intercal\mathbf{w}_1} - \cancel{\lambda\epsilon\dot{\mathbf{w}}_1{}^\intercal\epsilon\dot{\mathbf{w}}_1{}^\intercal},\end{aligned}$$

which gives the directional derivative:$\nabla_{\dot{\mathbf{w}}_1}\mathcal{L} = 2\mathbf{y}^\intercal\mathbf{y}\mathbf{w}_1^\intercal\mathbf{X}^\intercal\mathbf{X}\dot{\mathbf{w}}_1 - 2\lambda\mathbf{w}_1^\intercal\dot{\mathbf{w}}_1$ and the gradient:

$$\boxed{\begin{aligned}\nabla_{\mathbf{w}_1}\mathcal{L} &= [2\mathbf{y}^\intercal\mathbf{y}\mathbf{w}_1^\intercal\mathbf{X}^\intercal\mathbf{X} - 2\lambda\mathbf{w}_1^\intercal]^\intercal \\ &= 2\mathbf{y}^\intercal\mathbf{y}\mathbf{X}^\intercal\mathbf{X}\mathbf{w}_1 - 2\lambda\mathbf{w}_1 \\ &= 2\mathbf{y}\mathbf{X}^\intercal\mathbf{X}\mathbf{y}^\intercal\mathbf{w}_1 - 2\lambda\mathbf{w}_1\end{aligned}} \tag{C.16}$$

Computation of $\nabla_{\lambda_1}\mathcal{L}$: We develop $\mathcal{L}(\mathbf{w}_1, \lambda_1 + \epsilon\dot{\lambda}_1)$ and we remove the terms that are not linearly proportional to $\epsilon\dot{\lambda_1}$ (because they will not be involved in the gradient):

$$\begin{aligned}\mathcal{L}(\mathbf{w}_1, \lambda_1 + \epsilon\dot{\lambda}_1) &= \cancel{\mathrm{Tr}[\mathbf{y}\mathbf{w}_1\mathbf{X}^\intercal\mathbf{X}\mathbf{w}_1\mathbf{y}^\intercal]} - (\lambda + \epsilon\dot{\lambda})\mathbf{w}_1^\intercal\mathbf{w}_1 \\ &= \cancel{-\lambda\mathbf{w}_1^\intercal\mathbf{w}_1} - \epsilon\dot{\lambda}\mathbf{w}_1^\intercal\mathbf{w}_1,\end{aligned} \tag{C.17}$$

which gives the directional derivative:

$$\nabla_{\dot{\lambda_1}}\mathcal{L} = \dot{\lambda}\mathbf{w}_1^\intercal\mathbf{w}_1. \tag{C.18}$$

Finally, the gradient with respect to lambda is:

$$\boxed{\nabla_{\lambda_1}\mathcal{L} = \mathbf{w}_1^\intercal\mathbf{w}_1.} \tag{C.19}$$

C.4 Gradients for orthogonal joint PCA

Problem statement We want to compute the gradient of a Langrangian that writes:

$$\mathcal{L}(\mathbf{P}, \Lambda_\mathbf{P}, \mathbf{Q}, \Lambda_\mathbf{Q}) = \mathrm{Tr}\left[\mathbf{P}\mathbf{X}(\mathbf{P}\mathbf{X})^\intercal\right] + \left[\mathbf{Q}\mathbf{Y}(\mathbf{Q}\mathbf{Y})^\intercal\right] - \left[\Lambda_\mathbf{P}\mathbf{P}\mathbf{P}^\intercal\right] - \left[\Lambda_\mathbf{Q}\mathbf{Q}\mathbf{Q}^\intercal\right] - \left[\mathbf{K}\mathbf{P}\mathbf{Q}^\intercal\right]. \tag{C.20}$$

We use the definition of derivatives and we proceed by parts. The first directionnal derivative is:

$$\begin{aligned}\nabla_{\dot{\mathbf{P}}}\mathrm{Tr}\left(\mathbf{P}\mathbf{X}(\mathbf{P}\mathbf{X})^\intercal\right) = \lim_{\epsilon\to 0}\frac{\mathrm{Tr}\left((\mathbf{P} + \epsilon\dot{\mathbf{P}})\mathbf{X}((\mathbf{P} + \epsilon\dot{\mathbf{P}})\mathbf{X})^\intercal - \mathbf{P}\mathbf{X}(\mathbf{P}\mathbf{X})^\intercal\right)}{\epsilon} &= \lim_{\epsilon\to 0}\frac{\mathrm{Tr}(2\mathbf{P}\mathbf{X}\mathbf{X}^\intercal\epsilon\dot{\mathbf{P}})}{\epsilon} \\ &= \mathrm{Tr}(2\mathbf{P}\mathbf{X}\mathbf{X}^\intercal\dot{\mathbf{P}}^\intercal).\end{aligned}$$

which gives the gradient:

$$\boxed{\implies \nabla_{\mathbf{P}} \mathrm{Tr}\left(\mathbf{PX}(\mathbf{PX})^\intercal\right) = 2\mathbf{PXX}^\intercal} \tag{C.21}$$

The directional derivative for the second term (involving $\mathbf{P}$) is:

$$\begin{aligned}\nabla_{\dot{\mathbf{P}}} \mathrm{Tr}\left(-\Lambda_{\mathbf{P}}\mathbf{PP}^\intercal\right) = \lim_{\epsilon\to 0} \frac{\mathrm{Tr}\left(-\lambda(\mathbf{P}+\epsilon\dot{\mathbf{P}})(\mathbf{P}+\epsilon\dot{\mathbf{P}})^\intercal + \Lambda_{\mathbf{P}}\mathbf{PP}^\intercal\right)}{\epsilon} &= \lim_{\epsilon\to 0} -\frac{\mathrm{Tr}(2\Lambda_{\mathbf{P}}\mathbf{P}\epsilon\dot{\mathbf{P}}^\intercal)}{\epsilon} \\ &= -\mathrm{Tr}(2\Lambda_{\mathbf{P}}\mathbf{P}\dot{\mathbf{P}}^\intercal).\end{aligned}$$

And its gradient is:

$$\boxed{\implies \nabla_{\mathbf{P}} \mathrm{Tr}\left(-\Lambda_{\mathbf{P}}\mathbf{PP}^\intercal\right) = -2\Lambda_{\mathbf{P}}\mathbf{P}} \tag{C.22}$$

The directional derivative for the last term involving $\mathbf{P}$ is:

$$\begin{aligned}\nabla_{\dot{\mathbf{P}}} \mathrm{Tr}\left(-\mathbf{KPQ}^\intercal\right) = \lim_{\epsilon\to 0} \frac{\mathrm{Tr}\left(-\mathbf{K}(\mathbf{P}+\epsilon\dot{\mathbf{P}})\mathbf{Q}^\intercal + \mathbf{QPQ}^\intercal\right)}{\epsilon} &= \lim_{\epsilon\to 0} -\frac{\mathrm{Tr}(\mathbf{K}\epsilon\dot{\mathbf{P}}\mathbf{Q}^\intercal)}{\epsilon} \\ &= -\mathrm{Tr}(\mathbf{Q}^\intercal\mathbf{KP}\dot{\mathbf{P}}).\end{aligned}$$

And its gradient is:

$$\boxed{\implies \nabla_{\mathbf{P}} \mathrm{Tr}\left(-\mathbf{KPQ}^\intercal\right) = -\mathbf{K}^\intercal\mathbf{Q}} \tag{C.23}$$

Similarly, we get the gradients with respect to $\mathbf{Q}$:

$$\boxed{\begin{array}{c}\nabla_{\mathbf{Q}} \mathrm{Tr}\left(\mathbf{QY}(\mathbf{QY})^\intercal\right) = 2\mathbf{QYY}^\intercal \\ \nabla_{\mathbf{Q}} \mathrm{Tr}\left(-\Lambda_{\mathbf{Q}}\mathbf{QQ}^\intercal\right) = -2\Lambda_{\mathbf{Q}}\mathbf{Q} \\ \nabla_{\mathbf{Q}} \mathrm{Tr}\left(-\mathbf{KPQ}^\intercal\right) = \mathbf{KP}\end{array}} \tag{C.24}$$

C.5 Gradients for quasi-orthogonal joint PCA

Problem statement We want to compute the gradient of a Langrangian that writes:

$$\mathcal{L}(\mathbf{P}, \Lambda_{\mathbf{P}}, \mathbf{Q}, \Lambda_{\mathbf{Q}}) = \mathrm{Tr}\left[\mathbf{PX}(\mathbf{PX})^\intercal\right] + \left[\mathbf{QY}(\mathbf{QY})^\intercal\right] - \left[\Lambda_p\mathbf{PP}^\intercal\right] - \left[\Lambda_{\mathbf{Q}}\mathbf{QQ}^\intercal\right] - \left[\mathbf{KPQ}^\intercal\right]. \tag{C.25}$$

From the previous section C.4, we already have:

$$\boxed{\begin{array}{c}\nabla_{\mathbf{P}} \mathrm{Tr}\left(-\mathbf{PX}(\mathbf{PX})^\intercal\right) = -2\mathbf{PXX}^\intercal \quad , \quad \nabla_{\mathbf{Q}} \mathrm{Tr}\left(-\mathbf{QY}(\mathbf{QY})^\intercal\right) = -2\mathbf{QYY}^\intercal \\ \nabla_{\mathbf{P}} \mathrm{Tr}\left(-\Lambda_{\mathbf{P}}\mathbf{PP}^\intercal\right) = -2\Lambda_{\mathbf{P}}\mathbf{P} \quad \text{and} \quad \nabla_{\mathbf{Q}} \mathrm{Tr}\left(-\Lambda_{\mathbf{Q}}\mathbf{QQ}^\intercal\right) = -2\Lambda_{\mathbf{Q}}\mathbf{Q}\end{array}} \tag{C.26}$$

For the term remaining term, we derive using the definition of directional derivatives:

$$\begin{aligned}\nabla_{\dot{\mathbf{P}}} \mathrm{Tr}\left(\lambda(\mathbf{QP}^\intercal\mathbf{PQ}^\intercal)\right) &= \lim_{\epsilon\to 0} \frac{\mathrm{Tr}\left(\lambda(\mathbf{Q}(\mathbf{P}+\epsilon\dot{\mathbf{P}})^\intercal(\mathbf{P}+\epsilon\dot{\mathbf{P}})\mathbf{Q}^\intercal) - \lambda(\mathbf{QP}^\intercal\mathbf{PQ}^\intercal)\right)}{\epsilon} \\ &= \lim_{\epsilon\to 0} -\frac{\mathrm{Tr}(2\lambda\mathbf{QP}^\intercal\epsilon\dot{\mathbf{P}}\mathbf{Q}^\intercal)}{\epsilon} = -\mathrm{Tr}(2\lambda\mathbf{Q}^\intercal\mathbf{QP}^\intercal\dot{\mathbf{P}}).\end{aligned} \tag{C.27}$$

From which we get the gradients:

$$\boxed{\begin{aligned}\nabla_{\mathbf{P}}\mathrm{Tr}\left(\lambda(\mathbf{Q}\mathbf{P}^\intercal\mathbf{P}\mathbf{Q}^\intercal)\right) &= 2\lambda\mathbf{P}\mathbf{Q}^\intercal\mathbf{Q}\\ \nabla_{\mathbf{Q}}\mathrm{Tr}\left(\lambda(\mathbf{Q}\mathbf{P}^\intercal\mathbf{P}\mathbf{Q}^\intercal)\right) &= 2\lambda\mathbf{Q}\mathbf{P}^\intercal\mathbf{P}.\end{aligned}} \tag{C.28}$$

References

[1] A. Kacem, K. Cherenkova, and D. Aouada, "Disentangled Face Identity Representations for joint 3D Face Recognition and Expression Neutralisation," apr 2021. v, 79, 97, 98

[2] P. I. Spuls, L. L. Lecluse, M. L. N. Poulsen, J. D. Bos, R. S. Stern, and T. Nijsten, "How good are clinical severity and outcome measures for psoriasis: Quantitative evaluation in a systematic review," *Journal of Investigative Dermatology*, vol. 130, no. 4, pp. 933–943, 2010. 1

[3] H. Cartier, P. Trevidic, B. Rzany, G. Sattler, P. Kestemont, N. Kerrouche, and J.-C. Dhuin, "Perioral rejuvenation with a range of customized hyaluronic acid fillers: efficacy and safety over six months with a specific focus on the lips.," *Journal of drugs in dermatology : JDD*, vol. 11, pp. s17–26, jan 2012. 1, 74, 75, 76

[4] B. Mailey, J. L. Baker, A. Hosseini, J. Collins, A. Suliman, A. M. Wallace, and S. R. Cohen, "Evaluation of Facial Volume Changes after Rejuvenation Surgery Using a 3-Dimensional Camera," *Aesthetic Surgery Journal*, vol. 36, no. 4, pp. 379–387, 2016. 1, 75

[5] J. H. Silva, B. C. de Sá, A. L. R. de Avila, G. Landman, and J. P. Duprat Neto, "Atypical mole syndrome and dysplastic nevi: identification of populations at risk for developing melanoma - review article.," *Clinics (Sao Paulo, Brazil)*, vol. 66, no. 3, pp. 493–499, 2011. 1

[6] J. E. Schreiber, J. Terner, C. S. Stern, J. Beut, E. B. Jelks, G. W. Jelks, and O. M. Tepper, "The Boomerang Lift: A Three-Step Compartment-Based Approach to the Youthful Cheek," *Plastic and Reconstructive Surgery*, vol. 141, no. 4, 2018. 1

[7] F. Urdiales-Gálvez, S. Martín-Sánchez, M. Maíz-Jiménez, and E. Viruel-Ortega, "Body Contouring Using a Combination of Pulsed Ultrasound and Unipolar Radio Frequency: A Prospective Pilot Study," *Aesthetic Plastic Surgery*, 2022. 1

[8] O. E. Linden, J. Kit He, C. S. Morrison, S. R. Sullivan, and H. O. Taylor, "The relationship between age and facial asymmetry," *Plastic and Reconstructive Surgery*, vol. 142, no. 5, pp. 1145–1152, 2018. 1

[9] V. Noisser, A. Eigenberger, M. Weiherer, S. Seitz, L. Prantl, and V. Brébant, "Surgery of congenital breast asymmetry—which objective parameter influences the subjective satisfaction with long-term results," *Archives of Gynecology and Obstetrics*, vol. 305, no. 1, pp. 95–102, 2022. 1

[10] A. Zhavoronkov and P. Mamoshina, "Deep Aging Clocks: The Emergence of AI-Based Biomarkers of Aging and Longevity," *Trends in Pharmacological Sciences*, vol. 40, no. 8, pp. 546–549, 2019. 2

[11] E. Bobrov, A. Georgievskaya, K. Kiselev, A. Sevastopolsky, A. Zhavoronkov, S. Gurov, K. Rudakov, M. Del Pilar Bonilla Tobar, S. Jaspers, and S. Clemann, "PhotoAgeClock: deep learning algorithms for development of non-invasive visual biomarkers of aging.," *Aging*, vol. 10, pp. 3249–3259, nov 2018. 2

[12] S. Windhager, P. Mitteroecker, I. Rupić, T. Lauc, O. Polašek, and K. Schaefer, "Facial aging trajectories: A common shape pattern in male and female faces is disrupted after menopause.," *American journal of physical anthropology*, vol. 169, pp. 678–688, aug 2019. 2

[13] C. López-Otín, M. A. Blasco, L. Partridge, M. Serrano, and G. Kroemer, "The hallmarks of aging.," *Cell*, vol. 153, pp. 1194–1217, jun 2013. 2, 38

[14] A. S. Wang and O. Dreesen, "Biomarkers of Cellular Senescence and Skin Aging.," *Frontiers in genetics*, vol. 9, p. 247, 2018. 2

[15] T. Kanaki, E. Makrantonaki, and C. C. Zouboulis, "Biomarkers of skin aging," *Reviews in Endocrine and Metabolic Disorders*, vol. 17, pp. 433–442, sep 2016. 2

[16] M. Lorenzi, X. Pennec, N. Ayache, G. Frisoni, M. Lorenzi, X. Pennec, N. Ayache, and G. Frisoni, "Disentangling the normal aging from the pathological Alzheimer ' s disease progression on cross-sectional structural MR images To cite this version : HAL Id : hal-00813836 Disentangling the Normal Aging from the Pathological Alzheimer ' s Disease Progression on," 2013. 2

[17] C. Abi Nader, N. Ayache, V. Manera, P. Robert, and M. Lorenzi, "Disentangling spatio-temporal patterns of brain changes in large-scale brain imaging databases through independent Gaussian process analysis," *Revue d'Épidémiologie et de Santé Publique*, vol. 66, p. S159, 2018. 2

[18] R. Sivera, H. Delingette, M. Lorenzi, X. Pennec, and N. Ayache, "A model of brain morphological changes related to aging and Alzheimer's disease from cross-sectional assessments," *NeuroImage*, vol. 198, pp. 255–270, 2019. 2

[19] National Institutes of Health (NIH) Image Gallery, "Telomeres," 2016. [Attribution-NonCommercial 2.0 Generic (CC BY-NC 2.0)]. 3

[20] J. Riviere, P. Gotardo, D. Bradley, A. Ghosh, and T. Beeler, "Single-shot high-quality facial geometry and skin appearance capture," *ACM Transactions on Graphics*, vol. 39, no. 4, pp. 1–12, 2020. 3

[21] A. Analytics, A. Deep, and K. Group, "Longevity Biomarkers Landscape Overview," no. October, 2021. 3

[22] D. V. Cicchetti, "Assessing inter rater reliability for rating scales: resolving some basic issues," *British Journal of Psychiatry*, 1976. 3, 38, 39

[23] K. A. Hallgren, "Computing Inter-Rater Reliability for Observational Data: An Overview and Tutorial," *Tutorials in Quantitative Methods for Psychology*, 2012. 3, 38, 39

[24] K. L. Gwet, "Computing inter-rater reliability and its variance in the presence of high agreement," *British Journal of Mathematical and Statistical Psychology*, 2008. 3, 38, 39

[25] T. Gerig, A. Morel-Forster, C. Blumer, B. Egger, M. Luthi, S. Schoenborn, and T. Vetter, "Morphable face models - An open framework," in *Proceedings - 13th IEEE International Conference on Automatic Face and Gesture Recognition, FG 2018*, 2018. 4, 5, 8, 9, 15, 16, 18, 22, 24, 28, 74, 79, 101, 104

[26] H. Yang, H. Zhu, Y. Wang, M. Huang, Q. Shen, R. Yang, and X. Cao, "FaceScape: A large-scale high quality 3D face dataset and detailed riggable 3D face prediction," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2020. 4, 5, 63, 74, 76, 77, 78, 80, 91, 92, 104

[27] C. Cao, Y. Weng, S. Zhou, Y. Tong, and K. Zhou, "FaceWarehouse: A 3D facial expression database for visual computing," *IEEE Transactions on Visualization and Computer Graphics*, 2014. 4, 5, 8, 10, 27, 77, 104

[28] F. Jousse, X. Pennec, H. Delingette, and M. Gonzalez, "Geodesic squared exponential kernel for non-rigid shape registration," *Proceedings - 2021 16th IEEE International Conference on Automatic Face and Gesture Recognition, FG 2021*, 2021. 5, 7

[29] E. Stella and A. Di Petrillo, *Standard Evaluation of the Patient: The Merz Scale*, pp. 33–50. Milano: Springer Milan, 2014. 5, 39, 41, 61, 69, 74, 75, 76

[30] H. Yang, H. Zhu, Y. Wang, M. Huang, Q. Shen, R. Yang, and X. Cao, "Facescape: A large-scale high quality 3d face dataset and detailed riggable 3d face prediction," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 8, 9

[31] T. Li, T. Bolkart, M. J. Black, H. Li, and J. Romero, "Learning a model of facial shape and expression from 4D scans," *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, vol. 36, no. 6, pp. 194:1–194:17, 2017. 8

[32] P. J. Besl and N. D. McKay, "A Method for Registration of 3-D Shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1992. 9, 76

[33] Z. Zhang, "Iterative point matching for registration of free-form curves and surfaces," *International Journal of Computer Vision*, 1994. 9

[34] C. Yang and G. Medioni, "Object modelling by registration of multiple range images," *Image and Vision Computing*, 1992. 9

[35] C. H. Menq, H. T. Yau, and G. Y. Lai, "Automated precision measurement of surface profile in CAD-directed inspection," *IEEE Transactions on Robotics and Automation*, 1992. 9

[36] S. Granger and X. Pennec, "Multi-scale EM-ICP: A fast and robust approach for surface registration," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2002. 9

[37] G. Champleboux, S. Lavallee, R. Szeliski, and L. Brunie, "From accurate range imaging sensor calibration to accurate model-based 3D object localization," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1992. 9

[38] J. Feldmar and N. Ayache, "Rigid, affine and locally affine registration of free-form surfaces," *International Journal of Computer Vision*, 1996. 9

[39] G. Subsol, J. P. Thirion, and N. Ayache, "A general scheme for automatically building 3D morphometric anatomical atlases: application to a skull and brain atlas," no. December 2014, pp. 115–122, 1996. 9

[40] G. Subsol, J. Thirion, and N. Ayache, "Non rigid registration for building 3D anatomical atlases," pp. 576–578, 2002. 9

[41] B. Amberg, S. Romdhani, and T. Vetter, "Optimal step nonrigid ICP algorithms for surface registration," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2007. 9

[42] M. Luthi, T. Gerig, C. Jud, and T. Vetter, "Gaussian Process Morphable Models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018. 9, 10, 12, 13, 48, 101

[43] 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," 2013. 10

[44] B. Egger, W. A. Smith, A. Tewari, S. Wuhrer, M. Zollhoefer, T. Beeler, F. Bernard, T. Bolkart, A. Kortylewski, S. Romdhani, C. Theobalt, V. Blanz, and T. Vetter, "3D Morphable face models - Past, present and future," 2019. 10

[45] V. Blanz and T. Vetter, "A morphable model for the synthesis of 3D faces," in *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1999*, 1999. 10, 12

[46] B. K. Horn and B. G. Schunck, "Determining optical flow," *Artificial Intelligence*, 1981. 10

[47] L. G. Brown, "A survey of image registration techniques," *ACM Computing Surveys (CSUR)*, 1992. 10

[48] V. Jain, H. Zhang, and O. Van Kaick, "Non-rigid spectral correspondence of triangle meshes," *International Journal of Shape Modeling*, 2007. 10

[49] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. 2018. 13, 19

[50] C. Williams and M. Seeger, "Using the nyström method to speed up kernel machines," in *Advances in Neural Information Processing Systems* (T. Leen, T. Dietterich, and V. Tresp, eds.), vol. 13, MIT Press, 2001. 13, 14

[51] M. Li, J. T. Kwok, and B.-L. Lu, "Making large-scale nyström approximation possible," in *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, (Madison, WI, USA), p. 631–638, Omnipress, 2010. 14

[52] R. Opfer, "Multiscale kernels," *Advances in Computational Mathematics*, 2006. 15

[53] E. Del Castillo, B. M. Colosimo, and S. D. Tajbakhsh, "Geodesic gaussian processes for the parametric reconstruction of a free-form surface," *Technometrics*, 2015. 18

[54] K. Crane, C. Weischedel, and M. Wardetzky, "Geodesics in heat," *ACM Transactions on Graphics*, 2013. 18, 19, 101

[55] A. Jacobson, D. Panozzo, *et al.*, "libigl: A simple C++ geometry processing library," 2018. https://libigl.github.io/. 19, 45

[56] A. Feragen, F. Lauze, and S. Hauberg, "Geodesic exponential kernels: When curvature and linearity conflict," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2015. 19, 36

[57] G. Guennebaud, B. Jacob, *et al.*, "Eigen v3." http://eigen.tuxfamily.org, 2010. 22, 52, 57

[58] S. Kolkur, D. Kalbande, P. Shimpi, C. Bapat, and J. Jatakia, "Human Skin Detection Using RGB, HSV and YCbCr Color Models," in *Proceedings of the International Conference on Communication and Signal Processing 2016 (ICCASP 2016)*, vol. 137, (Paris, France), pp. 324–332, Atlantis Press, 2017. 22, 25, 32

[59] J. Nocedal, "Updating Quasi-Newton Matrices with Limited Storage," *Mathematics of Computation*, vol. 35, p. 773, jul 1980. 22

[60] J. Behley, V. Steinhage, and A. B. Cremers, "Efficient Radius Neighbor Seach in Three-dimensional Point Clouds," in *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, 2015. 23

[61] J. Elseberg, S. Magnenat, R. Siegwart, and A. Nüchter, "Comparison of nearest-neighbor-search strategies and implementations for efficient shape registration," *Journal of Software Engineering for Robotics*, vol. 3, no. 1, pp. 2–12, 2012. 23

[62] R. R. Paulsen, K. A. Juhl, T. M. Haspang, T. Hansen, M. Ganz, and G. Einarsson, "Multi-view Consensus CNN for 3D Facial Landmark Placement," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11361 LNCS, pp. 706–719, 2019. 24

[63] International Society of aesthetic Plastic Surgery, "ISAPS INTERNATIONAL SURVEY ON AESTHETIC/COSMETIC PROCEDURES performed in 2020," tech. rep., ISAPS, 2020. 38

[64] S. Cotofana, A. A. Fratila, T. L. Schenck, W. Redka-Swoboda, I. Zilinsky, and T. Pavicic, "The Anatomy of the Aging Face: A Review," *Facial Plastic Surgery*, 2016. 38, 69

[65] T. J. Marten, "High smas facelift: Combined single flap lifting of the jawline, cheek, and midface," *Clinics in Plastic Surgery*, vol. 35, no. 4, pp. 569–603, 2008. Facelifts, Part II. 38

[66] J.-M. Dallara, M. Baspeyras, P. Bui, H. Cartier, M.-H. Charavel, and L. Dumas, "Calcium hydroxylapatite for jawline rejuvenation: consensus recommendations," *Journal of Cosmetic Dermatology*, vol. 13, no. 1, pp. 3–14, 2014. 38

[67] B. L. Edison, N. K. Tierney, H. A. Smith, A. Rush, D. Chua, M. Dufort, B. A. Green, C. Milikowski, and P. Farris, "27886 Optimized delivery of a novel acetyl dipeptide provides clinical and consumer perceivable benefits to antiaging attributes of the lower face, including firming and jawline contour," *Journal of the American Academy of Dermatology*, vol. 85, p. AB165, sep 2021. 38

[68] M. Baspeyras, J.-M. Dallara, H. Cartier, M.-H. Charavel, and L. Dumas, "Restoring jawline contour with calcium hydroxylapatite: A prospective, observational study," *Journal of Cosmetic Dermatology*, vol. 16, no. 3, pp. 342–347, 2017. 38

[69] J. V. Sundgaard, M. Värendh, F. Nordström, Y. Kamide, C. Tanaka, J. Harte, R. R. Paulsen, A. N. Christensen, P. Bray, and S. Laugesen, "Inter-rater reliability of the diagnosis of otitis media based on otoscopic images and wideband tympanometry measurements," *International Journal of Pediatric Otorhinolaryngology*, vol. 153, no. January, 2022. 39

[70] J. K. Tan, K. Fung, and L. Bulger, "Reliability of dermatologists in acne lesion counts and global assessments," *Journal of Cutaneous Medicine and Surgery*, 2006. 39

[71] C. Biörserud, M. Fagevik Olsén, A. Elander, and M. Wiklund, "Objective measurements of excess skin in post bariatric patients - Inter-rater reliability," *Journal of Plastic Surgery and Hand Surgery*, 2016. 39

[72] K. R. van Straalen, T. Verhagen, B. Horváth, C. Ardon, A. R. Vossen, R. Driessen, J. Boer, A. L. Rondags, E. P. Prens, and H. H. van der Zee, "Poor interrater reliability of hidradenitis suppurativa phenotypes," *Journal of the American Academy of Dermatology*, 2018. 39

[73] F. Cazals and M. Pouget, "Estimating differential quantities using polynomial fitting of osculating jets," *Computer Aided Geometric Design*, 2005. 45

[74] D. Panozzo, E. Puppo, and L. Rocca, "Efficient multi-scale curvature and crease estimation," in *2nd International Workshop on Computer Graphics, Computer Vision and Mathematics, GraVisMa 2010 - Workshop Proceedings*, 2010. 45

[75] T. Hastie, R. Tibshirani, and J. Friedman, *Elements of Statistical Learning 2nd ed.* 2009. 53

[76] J. A. Wegelin, "A survey of partial least squares (pls) methods, with emphasis on the two-block case," tech. rep., 2000. 54, 56

[77] S. Wold, M. Sjöström, and L. Eriksson, "Pls-regression: a basic tool of chemometrics," *Chemometrics and Intelligent Laboratory Systems*, vol. 58, no. 2, pp. 109–130, 2001. PLS Methods. 55

[78] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant, "Array programming with NumPy," *Nature*, vol. 585, pp. 357–362, Sept. 2020. 57

[79] M. Ojala and G. C. Garriga, "Permutation tests for studying classifier performance," *Journal of Machine Learning Research*, vol. 11, pp. 1833–1863, 2010. 60

[80] M. Lorenzi and X. Pennec, "Efficient parallel transport of deformations in time series of images: From schild's to pole ladder," *Journal of Mathematical Imaging and Vision*, vol. 50, 10 2013. 62

[81] W. contributors, "Prognathism." `https://en.wikipedia.org/w/index.php?title=Prognathism&oldid=1093555357`, 2022. Accessed: 2010-09-30. 72

[82] P. Kestemont, H. Cartier, P. Trevidic, B. Rzany, G. Sattler, N. Kerrouche, and J.-C. Dhuin, "Sustained efficacy and high patient satisfaction after cheek enhancement with a new hyaluronic acid dermal filler.," *Journal of drugs in dermatology : JDD*, vol. 11, pp. s9–16, jan 2012. 74, 75, 76

[83] K. Y. Park, J. M. Kim, J. Seok, S. J. Seo, M. N. Kim, and C. S. Youn, "Comparative split-face study of durational changes in hyaluronic acid fillers for mid-face volume augmentation," *Dermatologic Therapy*, vol. 32, p. e12950, jun 2019. 74, 75, 76

[84] B. Chu, *Neutralisation des expressions faciales pour améliorer la reconnaissance du visage*. Theses, Ecole Centrale de Lyon, Mar. 2015. 74, 79, 87, 98

[85] B. Egger, S. Sutherland, S. C. Medin, and J. Tenenbaum, "Identity-Expression Ambiguity in 3D Morphable Face Models," 2021. 74, 79, 87, 98

[86] K. Miehle, M. Stumvoll, M. Fasshauer, and T. Hierl, "Facial soft tissue volume decreases during metreleptin treatment in patients with partial and generalized lipodystrophy," *Endocrine*, vol. 58, pp. 262–266, nov 2017. 75

[87] M. Kerscher, K. Agsten, M. Kravtsov, and W. Prager, "Effectiveness evaluation of two volumizing hyaluronic acid dermal fillers in a controlled, randomized, double-blind, split-face clinical study," *Clinical, Cosmetic and Investigational Dermatology*, vol. Volume 10, pp. 239–247, jun 2017. 75

[88] D. Vlasic, M. Brand, H. Pfister, and J. Popović, "Face transfer with multilinear models," *SIGGRAPH 2006 - ACM SIGGRAPH 2006 Courses*, 2006. 77

[89] T. Li, T. Bolkart, M. J. Black, H. Li, and J. Romero, "Learning a model of facial shape and expression from 4D scans," in *ACM Transactions on Graphics*, 2017. 77

[90] A. Ranjan, T. Bolkart, S. Sanyal, and M. J. Black, "Generating 3D Faces Using Convolutional Mesh Autoencoders," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2018. 78

[91] Z.-H. Jiang, Q. Wu, K. Chen, and J. Zhang, "Disentangled Representation Learning for 3D Face Shape," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2019-June, pp. 11949–11958, IEEE, jun 2019. 79

[92] Y. Saad, "Iterative Methods for Sparse Linear Systems," *Iterative Methods for Sparse Linear Systems*, 2003. 86

[93] D. Lakens, "Calculating and reporting effect sizes to facilitate cumulative science: a practical primer for t-tests and ANOVAs," *Frontiers in Psychology*, vol. 4, 2013. 92

[94] K. O. McGraw and S. P. Wong, "A common language effect size statistic.," 1992. 93

[95] A. Seth, J. L. Hicks, T. K. Uchida, A. Habib, C. L. Dembia, J. J. Dunne, C. F. Ong, M. S. DeMers, A. Rajagopal, M. Millard, S. R. Hamner, E. M. Arnold, J. R. Yong, S. K. Lakshmikanth, M. A. Sherman, J. P. Ku, and S. L. Delp, "OpenSim: Simulating musculoskeletal dynamics and neuromuscular control to study human and animal movement," *PLOS Computational Biology*, vol. 14, p. e1006223, jul 2018. 99

[96] S. L. Delp, F. C. Anderson, A. S. Arnold, P. Loan, A. Habib, C. T. John, E. Guendelman, and D. G. Thelen, "OpenSim: open-source software to create and analyze dynamic simulations of movement.," *IEEE transactions on bio-medical engineering*, vol. 54, pp. 1940–1950, nov 2007. 99

[97] D. Pagnon, M. Domalain, and L. Reveret, "Pose2sim: An end-to-end workflow for 3D markerless sports kinematics—Part 1: Robustness," *Sensors*, vol. 21, no. 19, 2021. 99

[98] J. Glaunès, A. Qiu, M. I. Miller, and L. Younes, "Large deformation diffeomorphic metric curve mapping," *International Journal of Computer Vision*, vol. 80, no. 3, pp. 317–336, 2008. 103

[99] X. Pennec, "Statistical Analysis of Organs' Shapes and Deformations: The Riemannian and the Affine Settings in Computational Anatomy," pp. 159–183, 2021. 103

[100] M. P. do Carmo, *Differential geometry of curves and surfaces; 2nd ed.* Dover books on mathematics, Mineola, NY: Dover Publications, 2016. This is a revised, corrected and updated second edition of the work originally published in 1976 by Prentice-Hall, Inc., Englewood Cliffs, New Jersey. The autor has also provided a new preface for this edition.

[101] P. Geladi and B. R. Kowalski, "Partial least-squares regression: a tutorial," *Analytica Chimica Acta*, 1986.

[102] J. Wegelin, "A survey of Partial Least Squares (PLS) methods, with emphasis on the two-block case," *Technical Report 371, Department of Statistics, University of Washington, Seattle, 2000.*, 2000.

[103] K. Lekadir, X. Albà, M. Pereañez, and A. F. Frangi, "Statistical shape modeling using partial least squares: Application to the assessment of myocardial infarction," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2016.

[104] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham, "Active shape models - their training and application," *Computer Vision and Image Understanding*, 1995.

[105] P. Paysan, "Statistical modeling of facial aging based on 3d scans," 2010.

[106] A. Savran, N. Alyüz, H. Dibeklioğlu, O. Çeliktutan, B. Gökberk, B. Sankur, and L. Akarun, "Bosphorus database for 3D face analysis," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 5372 LNCS, pp. 47–56, 2008.

[107] S. Canavan, Y. Sun, X. Zhang, and L. Yin, "A dynamic curvature based approach for facial activity analysis in 3D space," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2012.

[108] S. Rusinkiewicz and M. Levoy, "Efficient variants of the ICP algorithm," *Proceedings of International Conference on 3-D Digital Imaging and Modeling, 3DIM*, 2001.

[109] IEEE, *A 3D Face Model for Pose and Illumination Invariant Face Recognition*, (Genova, Italy), 2009.

[110] M. Kazhdan, M. Bolitho, and H. Hoppe, "Poisson Surface Reconstruction," *Eurographics Symposium on Geometry Processing*, 2006.

[111] J. F. Cardenas-Garcia, H. G. Yao, and S. Zheng, "3D reconstruction of objects using stereo imaging," *Optics and Lasers in Engineering*, vol. 22, pp. 193–213, jan 1995.

[112] B. Egger, W. A. Smith, A. Tewari, S. Wuhrer, M. Zollhoefer, T. Beeler, F. Bernard, T. Bolkart, A. Kortylewski, S. Romdhani, C. Theobalt, V. Blanz, and T. Vetter, "3D Morphable Face Modelsa-Past, Present, and Future," *ACM Transactions on Graphics*, 2020.

[113] J. C. Bezdek and R. J. Hathaway, "Convergence of alternating optimization," *Neural, Parallel Sci. Comput.*, vol. 11, p. 351–368, dec 2003.

[114] QuantifiCare, "Lifeviz mini." `https://www.quantificare.com/3d-photography-systems_old/lifeviz-mini/`.