

KernelGPA: A Globally Optimal Solution to Deformable SLAM in Closed-form

Journal Title
XX(X):1–24
©The Author(s) 2016
Reprints and permission:
sagepub.co.uk/journalsPermissions.nav
DOI: 10.1177/ToBeAssigned
www.sagepub.com/

SAGE

Fang Bai¹, Kanzhi Wu² and Adrien Bartoli³

Abstract

We study the generalized Procrustes analysis (GPA), as a minimal formulation to the simultaneous localization and mapping (SLAM) problem. We propose KernelGPA, a novel global registration technique to solve SLAM in the deformable environment. We propose the concept of deformable transformation which encodes the entangled pose and deformation. We define deformable transformations using a kernel method, and show that both the deformable transformations and the environment map can be solved globally in closed-form, up to global scale ambiguities. We solve the scale ambiguities by an optimization formulation that maximizes rigidity. We demonstrate KernelGPA using the Gaussian kernel, and validate the superiority of KernelGPA with various datasets. Code and data are available at <https://bitbucket.org/clermontferrand/deformableprocrustes>.

Keywords

Deformable SLAM, deformable registration, nonrigid reconstruction, surgical robotics

1 Introduction

The simultaneous localization and mapping (SLAM), as an enabling technology for sensor localization and scene reconstruction, has witnessed a huge success in the past decade [Cadena, Carlone, Carrillo, Latif, Scaramuzza, Neira, Reid and Leonard \(2016\)](#). However, the successful application of SLAM critically relies on the assumption of a rigid (or static) scene [Dissanayake, Newman, Clark, Durrant-Whyte and Csorba \(2001\)](#).

Deformable SLAM. Recently, researchers have started to consider SLAM in the nonrigid and dynamic cases. While both terms seem close, they are referred to as quite different problems. The nonrigid case typically occurs in medical or surgical applications, whereas the dynamic case occurs in outdoor applications with moving pedestrians or traffic. In contrast to SLAM in the dynamic case, where the movement in the scene is almost random and thus is difficult to model, SLAM in the nonrigid environment is largely well-posed, because *the deformation is typically low dimensional, or follows certain structures or constraints*. It is therefore possible to model and estimate the deformation in the nonrigid scene, revealing the possibility of a *deformable SLAM* approach. We use the term *deformable SLAM* to refer to SLAM in the nonrigid (or deformable) case. The research of deformable SLAM is gaining popularity and has found its applications in surgical applications [Huang, Chen, Zhao, Zhang and Xu \(2021\)](#).

SfT and deformable tracking. The first generation of deformable SLAM systems are basically based on tracking technologies. In vision and graphics, matching a deformed shape to a *given template*, termed shape-from-template (SfT) in [Bartoli, Gérard, Chadebecq, Collins and Pizarro \(2015\)](#); [Malti and Herzet \(2017\)](#), is a well researched problem. These days, SfT can be solved under a large range of deformation

models, see a brief review in Section 2.1. The SfT methods are the pillar of deformable tracking systems published in robotics, for instance DynamicFusion [Newcombe, Fox and Seitz \(2015\)](#), Surfelwarp [Gao and Tedrake \(2018\)](#), KillingFusion [Slavcheva, Baust, Cremers and Ilic \(2017\)](#), SobolevFusion [Slavcheva, Baust and Ilic \(2018\)](#), and MIS-SLAM [Song, Wang, Zhao, Huang and Dissanayake \(2018\)](#), to name but a few. In SLAM, the template (*i.e.*, the environment map) is never known ahead. Thus these systems rely on an open-loop mechanism that incrementally construct the template. As a consequence, the estimation error of these tracking systems accumulates along the trajectory, due to the lack of global feedback. Hence, these solutions are inevitably suboptimal.

Loop-closure and global registration. In SLAM, the global feedback is constructed under the term of *loop-closures*, which has been well understood in the case of a rigid scene. In specific, when traveling in the scene, the sensor observes identifiable geometric points at different poses to form global feedback. In SLAM, such a global feedback is referred to as a *loop-closure*, and an identifiable geometric point in the scene as a *correspondence*. In essence, the re-observation of correspondences at different poses provides additional information, and thus reduces the uncertainty of estimation. It must be noted that the observations are defined

¹Under transition. Unknown affiliation to be assigned. ²vivo Mobile Communication Co., Ltd., Shenzhen, China. ³ENCOV, TGI, Institut Pascal, UMR6602 CNRS, Université Clermont Auvergne, Clermont-Ferrand, France; Department of Clinical Research and Innovation, CHU de Clermont-Ferrand.

Corresponding author:

Fang Bai, unknown affiliation to be assigned. The work was done during the author's visit to Tongji University, Shanghai, China.

Email: Fang.Bai@yahoo.com

in local coordinate frames relative to the sensor’s poses. Thus a *global registration* technique is required to fuse the observations of correspondences together. This technique is the generalized Procrustes analysis (GPA), see Section 3.2 for details, or structure-from-motion (SfM) if the sensors are projective cameras. We emphasize that GPA and SfM are minimal formulations of SLAM, as they decide the poses and the scene reconstruction completely. In the rigid case, both GPA and SfM are well solved — that is why SLAM in the rigid case is considered a solved problem.

Global registration with deformations. If the scene is nonrigid, we envision that a global registration technique that handles deformations is the key to solve deformable SLAM. Unfortunately, at this stage, the research of such techniques is rather sparse. Some representative works include: a) the low-rank shape basis decomposition Bregler, Hertzmann and Biermann (2000); Xiao, Chai and Kanade (2006); Dai, Li and He (2014), b) the isometric nonrigid structure-from-motion Parashar, Pizarro and Bartoli (2017), implemented in the DefSLAM system Lamarca, Parashar, Bartoli and Montiel (2020), and more recently c) DefGPA Bai and Bartoli (2022b), a GPA method with the linear basis warps (LBWs), see Section 2.2 for a brief review and comparison. All these methods are developed under certain assumptions about the deformations the scene undergoes. For example, methods a) assume structural deformations (e.g., gestures or facial expressions) to ensure the existence of a low-rank shape basis; methods b) assume isometric deformations which are suitable for foldable surfaces (e.g., papers or cloths). We feature method c) which assumes *smooth and low-dimensional deformations*, which is more suitable for visceral deformations occurring in surgical applications. In this work, we contribute further to the GPA family with a novel kernel based deformation model.

Problem statement. We study GPA with smooth and low-dimensional deformations, termed *deformable GPA*, a global registration technique for deformable SLAM. Deformable GPA can be considered as a minimal formulation of deformable SLAM. To make the context clear, deformable GPA is formulated under the following constraints:

- 1) *No temporal information.* We assume observations are made without sequential information, thus technologies based on tracking do not apply here.
- 2) *No template.* We assume a template of the scene is not available, and disallow inexact methods that incrementally construct and refine a template.
- 3) *No aids on pose estimation.* We assume additional information on the sensor’s pose is not available.

We assume that the only available information is observations of correspondences at different poses. The correspondences are used to capture two pieces of information: a) the sensor’s motion, and b) the deformation of the scene. As we shall see shortly in Section 3.1, the sensor’s pose and the deformation of the scene are entangled in deformable SLAM, making the registration extra difficult.

Contributions. This article is an extension to the KernelGPA method initially appeared in the proceedings of *Robotics: Science and Systems (RSS)* Bai and Bartoli (2022a). Concretely, this work contains the following contributions:

1. We unify the entangled poses and deformations together, and formally introduce the concept of *deformable transformation*. This way, we avoid the ambiguities in poses and deformations, because the deformable transformation is well defined and can be estimated (up to scale ambiguities).
2. We introduce a novel deformable transformation, termed *kernel based transformation (KBT)*. As the name suggests, the KBT is motivated from the kernel method. Compared with the LBWs in Bai and Bartoli (2022b), the KBT is more flexible and easier to design.
3. We propose KernelGPA, using KBT as the deformable transformation in GPA. We enforce implicit transformation constraints by constraining: a) the geometric center of the correspondence point-cloud to be at the origin of the coordinate frame, and b) the point-cloud covariance to be diagonalized as an unknown Λ .
4. We show that KernelGPA can be solved globally in closed-form up to $\sqrt{\Lambda}$ whose diagonal elements represent the global scale ambiguities. Our solution is based on a special eigenvalue problem first proposed in Bai and Bartoli (2022b). However, the exposition of relevant proofs is more concise in this paper.
5. We give a novel method to estimate the unknown $\sqrt{\Lambda}$. Compared with Bai and Bartoli (2022b), the novel method does not require the existence of globally visible correspondences, thus is more suitable for partial observations occurring in SLAM. We give an affine relaxation to obtain a closed-form $\sqrt{\Lambda}$.
6. We demonstrate the registration performance of KernelGPA using various datasets. We use three 3D datasets with correspondences. The first one comprises a set of 3D liver meshes with simulated smooth deformations. The second one comprises a set of 3D face meshes with various facial expressions. The third one comprises six deformed point-clouds extracted from computerized tomography (CT) data. We will release the relevant data to foster future research.

This article makes several improvements over the initial version appeared in RSS Bai and Bartoli (2022a). We have rewritten most of the text for better clarity, for instance, the exposition of the constraints in Section 5 and the special eigenvalue problem in Section 6. Importantly, we have refined the method to estimate $\sqrt{\Lambda}$ in Section 7, and have additionally added the discussion of degeneracies in Section 8. Lastly, we have used more advanced experiments in this version to demonstrate the usefulness of our method.

The remainder of this paper is organized as follows. We briefly review related work on deformation models and global registration techniques in Section 2. We introduce the concepts of deformable transformation and deformable GPA in Section 3. We present the KBT in Section 4, and registration constraints in Section 5. We draw the connection to a special eigenvalue problem in Section 6, and propose the method to estimate $\sqrt{\Lambda}$ in Section 7. We discuss degeneracies in Section 8, and implementation details in Section 9. We present our experimental results in Section 10, and conclude the paper in Section 11.

2 Related Work

2.1 Deformation Models

We shall use landmarks, *i.e.*, points, as the environment representation and define deformations accordingly. This representation has a long history in shape analysis Kendall (1984); Kilian, Mitra and Pottmann (2007). There has been a rich class of smooth deformation models (also termed smooth warps) developed based on landmark representations, *e.g.*, the Free-Form Deformations (FFD) Rueckert, Sonoda, Hayes, Hill, Leach and Hawkes (1999); Szeliski and Coughlan (1997), the Radial Basis Functions (RBF) Bookstein (1989); Fornefett, Rohr and Stiehl (2001) and the Thin-Plate Spline (TPS) Duchon (1976); Bookstein (1989). Beyond smooth models, there exist a class of models defined piece-wisely by implementing local transformations associated to a set of control points and modeling the deformations on other parts by interpolation. Representatives of such models include the ARAP deformation model Sorkine and Alexa (2007a), the embedded deformation graph Allen, Curless and Popović (2003); Sumner, Schmid and Pauly (2007), and Lie-bodies Freifeld and Black (2012).

Beyond landmark based models, other models based on curves Joshi, Klassen, Srivastava and Jermyn (2007); Younes, Michor, Shah and Mumford (2008) or surfaces have been proposed. Some well-known models include level sets Osher and Fedkiw (2003), medial surfaces Bouix, Pruessner, Collins and Siddiqi (2005), Q-maps Kurtek, Klassen, Ding and Srivastava (2010); Kurtek, Klassen, Gore, Ding and Srivastava (2011), and Square Root Normal Fields (SRNF) Jermyn, Kurtek, Klassen and Srivastava (2012); Laga, Xie, Jermyn and Srivastava (2017). Some models implement an articulated skeleton structure. Representative works include the medial axis representations (M-rep) Fletcher, Lu, Pizer and Joshi (2004), and SCAPE Anguelov, Srinivasan, Koller, Thrun, Rodgers and Davis (2005). We refer interested readers to the review papers Younes (2012); Laga (2018) for more details.

2.2 Global Registration Techniques

Generalized Procrustes analysis. The GPA framework was used as a fundamental technique in shape analysis to obtain an initial alignment. Both the rigid and affine transformations were recovered in the classical literature Kendall (1984); Goodall (1991); Rohlf and Slice (1990). Recently, a novel GPA technique with deformation models was proposed in Bai and Bartoli (2022b). The deformation model in Bai and Bartoli (2022b) is termed LBWs, which includes the affine transformation and a rich class of nonlinear deformation models Rueckert, Sonoda, Hayes, Hill, Leach and Hawkes (1999); Szeliski and Coughlan (1997); Bookstein (1989); Fornefett, Rohr and Stiehl (2001); Bartoli, Perriollat and Chambon (2010) using radial-basis functions, *e.g.*, the well-known TPS Bookstein (1989).

The work Bai and Bartoli (2022b) is the closest to ours. However, we use a kernel method to model deformations, which is a novel deformation model compared to the LBWs used in Bai and Bartoli (2022b). In addition, we propose a novel method to estimate the global scale ambiguities, which does not require some correspondences to be globally visible, thus is more suitable for SLAM applications.

Nonrigid structure-from-motion. SfM is a well-known global registration method that handles camera projections Hartley and Zisserman (2004). We do not consider projective cameras in this work, thus will only mention several nonrigid SfM (NRSfM) methods for references. One line of NRSfM methods use low-rank shape bases Bregler, Hertzmann and Biermann (2000); Xiao, Chai and Kanade (2006); Dai, Li and He (2014). These methods model deformations as a linear combination of the basis shapes, which are jointly factorized by the singular value decomposition (SVD). Another line of NRSfM methods use differential geometry, where the deformations are constrained to be isometric or conformal, *e.g.*, the isometric NRSfM Parashar, Pizarro and Bartoli (2017) which has been successfully implemented in DefSLAM Lamarca, Parashar, Bartoli and Montiel (2020). We refer interested readers to a recent work using Cartan’s connections Parashar, Pizarro and Bartoli (2019) and references therein.

Characterization by deformations. The work Bai and Bartoli (2022b) assumes smooth and low-dimensional deformations, as implied by the usage of the LBW. The works Bregler, Hertzmann and Biermann (2000); Xiao, Chai and Kanade (2006); Dai, Li and He (2014) require the existence of the low-rank shape basis. This is possible if the scene undergoes structural deformations *e.g.*, gestures or facial expressions. The works Parashar, Pizarro and Bartoli (2017); Lamarca, Parashar, Bartoli and Montiel (2020) require the deformation to follow isometry, preserving infinitesimal rigidity on the surface of the scene. This is usually true for foldable surfaces like papers or cloths.

In general, the visceral deformation is neither structural nor isometric, but is smooth (to avoid visceral damages) and low-dimensional (as driven by a limited number of force sources, *e.g.*, from muscles). In this work, we propose the KBT, a smooth and low-dimensional model suitable for visceral deformations, to meet the demand of surgical applications.

3 Formulation of Deformable SLAM and its Connection to GPA

3.1 Deformable SLAM

Our method is based on correspondences, and is independent of the detailed dense environment model to be used.

Environment modeling. We are concerned with a collection of m landmarks $\mathbf{M} \in \mathbb{R}^{d \times m}$ residing in the d -dimensional environment, where $d = 2$ or $d = 3$. The onboard sensor observes these landmarks in \mathbf{M} at discrete time points $t = 1, 2, \dots, n$. We denote the sensor’s pose at time t by $(\mathbf{R}_t \in \text{SO}(d), \mathbf{t}_t \in \mathbb{R}^d)$. The sensor at t observes m_t partial landmarks in \mathbf{M} , denoted by $\mathbf{M}\Gamma_t \in \mathbb{R}^{d \times m_t}$, with the help of a *visibility matrix* Γ_t to be defined below. It can be easily verified that $\Gamma_t^T \mathbf{1}_m = \mathbf{1}_{m_t}$.

Definition 1. (Visibility matrix). We denote the identity matrix in $\mathbb{R}^{m \times m}$ as a set of standard basis vectors in \mathbb{R}^m :

$$\mathbf{I}_m = [\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_m] \in \mathbb{R}^{m \times m}.$$

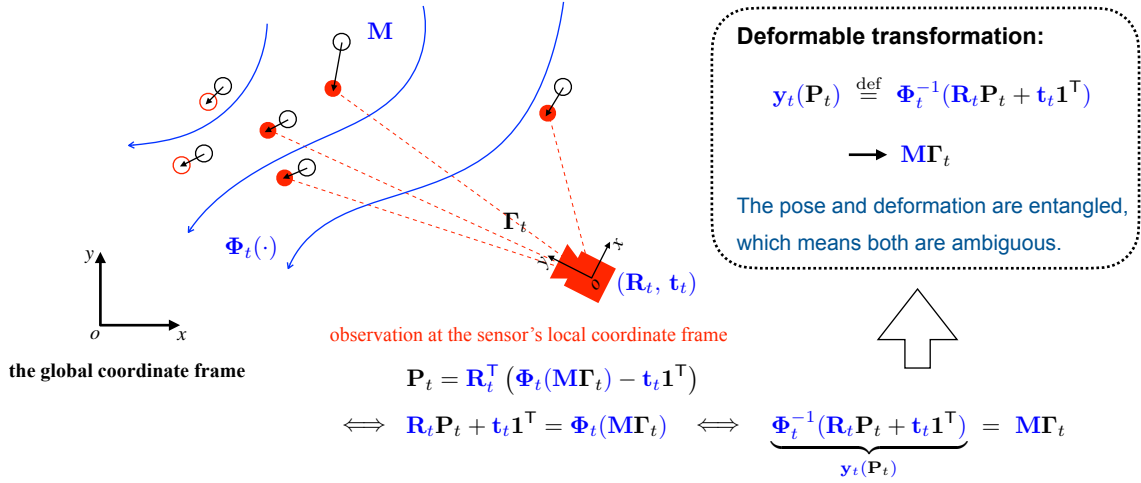


Figure 1. Deformable SLAM as the generalized Procrustes analysis (GPA) problem with deformable transformations. Our method is based on correspondences whose movements reflect deformations. The movements of the correspondences, as plotted by the arrows from the black to the red circles, are driven by a low-dimensional deformation field $\Phi_t(\cdot)$. The unknowns are colored in blue, including a) the rigid pose $(\mathbf{R}_t, \mathbf{t}_t)$, b) the low-dimensional deformation $\Phi_t(\cdot)$, and c) a canonical environment map \mathbf{M} . From the observation model, we notice that a) and b) are entangled, which means we need to know one in order to infer the other. In this work, we instead propose to solve $\mathbf{y}_t(\cdot)$, a unified deformable transformation which encodes both poses and deformations. We derive that both $\mathbf{y}_t(\cdot)$ and the environment map \mathbf{M} can be estimated globally in closed-form up to d scale ambiguities. The global coordinate frame is implicitly specified by the transformation constraints to be illustrated in Figure 2.

Obviously, $\mathbf{M} \mathbf{I}_m = \mathbf{M}$. The columns of a visibility matrix Γ_t are constructed from the standard basis vectors in \mathbb{R}^m :

$$\Gamma_t = [\mathbf{e}_{j_1}, \mathbf{e}_{j_2}, \dots, \mathbf{e}_{j_{m_t}}] \in \mathbb{R}^{m \times m_t},$$

where the subscripts $j_1, j_2, \dots, j_{m_t} \in [1 : m]$ denote the m_t points visible in \mathbf{P}_t .

Remark 1. In Bai and Bartoli (2022b), the authors use the augmented visibility matrix $\bar{\Gamma}_t$ defined as:

$$\bar{\Gamma}_t = \Gamma_t \Gamma_t^\top = \sum_{j=1}^{m_t} \mathbf{e}_{t_j} \mathbf{e}_{t_j}^\top \in \mathbb{R}^{m \times m}.$$

Such a $\bar{\Gamma}_t$ is a diagonal matrix whose (k, k) -th element is 1 if the k -th point in \mathbf{M} occurs in \mathbf{M}_t , and 0 otherwise. Γ_t is obtained by deleting the columns of zeros in $\bar{\Gamma}_t$.

Example 1. Given 5 points, if the first and the third points are visible, the visibility matrices are defined as:

$$\Gamma = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad \bar{\Gamma} = \Gamma \Gamma^\top = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Point-cloud observation of deformable environment. In deformable SLAM, the environment deforms over time. We denote the deformation as a time varying function $\Phi_t(\cdot)$. In particular the deformed environment at time t is:

$$\Phi_t(\mathbf{M}_t) = \Phi_t(\mathbf{M} \Gamma_t).$$

We denote the sensor's measurement at t by a point-cloud $\mathbf{P}_t \in \mathbb{R}^{d \times m_t}$ defined in the sensor's local coordinate frame. In the noise-free case, the measurement \mathbf{P}_t at t is the

observation of the deformed environment $\Phi_t(\mathbf{M} \Gamma_t)$:

$$\begin{aligned} \mathbf{P}_t &= \mathbf{R}_t^\top (\Phi_t(\mathbf{M} \Gamma_t) - \mathbf{t}_t \mathbf{1}^\top) \\ \Leftrightarrow \mathbf{R}_t \mathbf{P}_t + \mathbf{t}_t \mathbf{1}^\top &= \Phi_t(\mathbf{M} \Gamma_t). \end{aligned} \quad (1)$$

Composed transformation. From the above, we see that *the deformation and the pose are entangled*. In order to estimate one, we need to know the other (see Remark 2). To resolve this ambiguity, we fairly assume the deformation function $\Phi_t(\cdot)$ is invertible, and thus define $\mathbf{y}_t(\cdot)$ as a composition of both the pose $(\mathbf{R}_t, \mathbf{t}_t)$ and the deformation $\Phi_t^{-1}(\cdot)$:

$$\mathbf{y}_t(\mathbf{P}_t) \stackrel{\text{def}}{=} \Phi_t^{-1}(\mathbf{R}_t \mathbf{P}_t + \mathbf{t}_t \mathbf{1}^\top) = \mathbf{M} \Gamma_t. \quad (2)$$

In what follows, we term $\mathbf{y}_t(\cdot)$ *deformable transformation*.

Deformable SLAM. We define deformable SLAM as the problem that estimates 1) the deformable transformations $\mathbf{y}_t(\cdot)$ and 2) the environment map \mathbf{M} , using a collection of sensor measurements (\mathbf{P}_t, Γ_t) at time points $t = 1, 2, \dots, n$. Formally, we formulate deformable SLAM as:

$$\min \sum_{t=1}^n \varphi_t \quad \text{with } \varphi_t = \|\mathbf{y}_t(\mathbf{P}_t) - \mathbf{M} \Gamma_t\|_{\mathcal{F}}^2. \quad (3)$$

Remark 2. Given the pose $(\mathbf{R}_t, \mathbf{t}_t)$ and \mathbf{M} , the deformation field $\Phi_t(\cdot)$ is characterized by the vector flow:

$$\mathbf{M} \Gamma_t \rightarrow \mathbf{R}_t \mathbf{P}_t + \mathbf{t}_t \mathbf{1}^\top.$$

Conversely, give the deformation field $\Phi_t(\cdot)$, the pose $(\mathbf{R}_t, \mathbf{t}_t)$ is characterized by the rigid Procrustes analysis. Thus given \mathbf{M} , the disentanglement is possible once either the deformation $\Phi_t(\cdot)$ or the pose $(\mathbf{R}_t, \mathbf{t}_t)$ is known. In this work, we focus on how to solve \mathbf{M} and $\mathbf{y}_t(\cdot)$.

3.2 Generalized Procrustes Analysis

The deformable SLAM formulation (3) is essentially a GPA problem with deformable transformations, see Figure 1. In the classical literature, GPAs with both the rigid transformation and the affine transformation are well studied.

GPA with the rigid transformation. In this case, from formulation (3), we define $\mathbf{y}_t(\cdot)$ as:

$$\mathbf{y}_t(\mathbf{P}_t) \stackrel{\text{def}}{=} \mathbf{R}_t \mathbf{P}_t + \mathbf{t}_t \mathbf{1}^\top, \quad (\mathbf{R}_t \in \text{SO}(d), \mathbf{t}_t \in \mathbb{R}^d).$$

There exists a closed-form solution for the case of $n = 2$ point-clouds. In general, for $n \geq 3$, the solution is computed iteratively by nonlinear least squares (NLS) optimization techniques, e.g., Gauss-Newton or Levenberg-Marquardt.

GPA with the affine transformation. In this case, from formulation (3), we define $\mathbf{y}_t(\cdot)$ as:

$$\mathbf{y}_t(\mathbf{P}_t) \stackrel{\text{def}}{=} \mathbf{A}_t \mathbf{P}_t + \mathbf{a}_t \mathbf{1}^\top, \quad (\mathbf{A}_t \in \mathbb{R}^{d \times d}, \mathbf{a}_t \in \mathbb{R}^d).$$

The resulting GPA problem is degenerate. The optimal solution is $\mathbf{A}_t = \mathbf{O}$, $\mathbf{a}_t = \mathbf{0}$, $\mathbf{M} = \mathbf{O}$, which however is useless. In order to construct a meaningful solution, we need to build a set of constraints, for example in the rigid case the transformation preserves the distance.

We shall term GPA with the rigid transformation as Rigid-GPA, and GPA with the affine transformation as Affine-GPA.

4 Deformable Transformation

4.1 Linear Basis Warp

The linear basis warp (LBW) in Bai and Bartoli (2022b), is a generalization of a class of deformable transformations, e.g., the free-form deformations (FFD) Rueckert, Sonoda, Hayes, Hill, Leach and Hawkes (1999); Szeliski and Coughlan (1997), and the thin-plate spline (TPS) Duchon (1976); Bookstein (1989).

Definition 2. (LBW in Bai and Bartoli (2022b)). Given a query point $\mathbf{p} \in \mathbb{R}^d$, the LBW is defined as:

$$\mathbf{y}_t(\mathbf{p}) \stackrel{\text{def}}{=} \mathbf{W}_t^\top \beta_t(\mathbf{p}), \quad (\mathbf{W}_t \in \mathbb{R}^{l \times d}), \quad (4)$$

where $\beta_t(\cdot) : \mathbb{R}^d \rightarrow \mathbb{R}^l$ is an embedding to the l -dimensional feature space. $\beta_t(\cdot)$ is typically designed from radial basis functions (RBFs) Fornetfett, Rohr and Stiehl (2001).

Regularization. Typically, the LBW is used together with a regularization term:

$$\mathcal{R}_t = \mu_t \text{tr}(\mathbf{W}_t^\top \Xi_t \mathbf{W}_t), \quad (\mu_t > 0), \quad (5)$$

where Ξ_t is a known matrix. Intuitively, the regularization \mathcal{R}_t acts as a penalty to control the allowed deformation.

Example 2. The affine transformation is a special case of the LBW where we use:

$$\mathbf{W}_t = [\mathbf{A}_t \quad \mathbf{a}_t]^\top, \quad \beta_t(\mathbf{p}) = \begin{bmatrix} \mathbf{p} \\ 1 \end{bmatrix}.$$

There is no regularization in this case, $\mathcal{R}_t = 0$.

Example 3. In case of the TPS warp, $\beta_t(\cdot)$ is designed as:

$$\beta_t(\mathbf{p}) = \mathcal{E}^\top \begin{bmatrix} \rho(\|\mathbf{c}_1 - \mathbf{p}\|) \\ \rho(\|\mathbf{c}_2 - \mathbf{p}\|) \\ \vdots \\ \rho(\|\mathbf{c}_l - \mathbf{p}\|) \\ \mathbf{p} \\ 1 \end{bmatrix},$$

where $\mathbf{c}_1, \dots, \mathbf{c}_l \in \mathbb{R}^d$ are l control points, and $\rho(\cdot)$ is a scalar function called the TPS kernel function. $\mathcal{E} \in \mathbb{R}^{(l+d+1) \times l}$ is a matrix constant decided from the control points and the TPS kernel function. The TPS warp thus defined implicitly includes a free affine transformation Bai and Bartoli (2022b).

Matrix Ξ_t used for regularization is chosen as the bending energy matrix Bookstein (1989). With this choice, the regularization is imposed on the nonlinear deformation only, thus leaving the implicit affine transformation free.

4.2 Kernel Based Transformation

Definition 3. (Kernel function). A kernel function $k(\cdot, \cdot) : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ evaluates the inner product in some feature space \mathcal{H} defined by $\phi(\cdot) : \mathcal{X} \rightarrow \mathcal{H}$ as:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle_{\mathcal{H}}, \quad \mathbf{x}_i, \mathbf{x}_j \in \mathcal{X}.$$

The spirit of a kernel method is to transform all the computation related to $\phi(\cdot)$ to the inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$, thus an explicit $\phi(\cdot)$ will never be required. This way, one can design a kernel method based on $k(\cdot, \cdot)$ directly.

Definition 4. (Kernel matrix). Given any $\mathbf{x}_1, \dots, \mathbf{x}_m \in \mathcal{X}$, and a kernel function $k(\cdot, \cdot)$, the kernel matrix $\mathbf{K} \in \mathbb{R}^{m \times m}$ (also called the Gram matrix) is constructed as:

$$\mathbf{K} = \begin{bmatrix} k(\mathbf{x}_1, \mathbf{x}_1) & \cdots & k(\mathbf{x}_1, \mathbf{x}_m) \\ \vdots & \ddots & \vdots \\ k(\mathbf{x}_m, \mathbf{x}_1) & \cdots & k(\mathbf{x}_m, \mathbf{x}_m) \end{bmatrix}.$$

Lemma 1. (Shawe-Taylor and Cristianini (2004)). If $k(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle_{\mathcal{H}}$ for some feature mapping $\phi(\cdot) : \mathcal{X} \rightarrow \mathcal{H}$, then for any $\mathbf{x}_1, \dots, \mathbf{x}_m \in \mathcal{X}$ the kernel matrix \mathbf{K} is symmetric positive (semi-)definite. The converse is also true. If the kernel matrix \mathbf{K} constructed from a kernel function $k(\cdot, \cdot)$ is symmetric positive (semi-)definite for any $\mathbf{x}_1, \dots, \mathbf{x}_m \in \mathcal{X}$, then there exists a feature mapping $\phi(\cdot) : \mathcal{X} \rightarrow \mathcal{H}$ such that $k(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle_{\mathcal{H}}$.

Definition 5. (Positive (semi-)definite kernel). A kernel function $k(\cdot, \cdot)$ that ensures the symmetric positive (semi-)definiteness of \mathbf{K} for any $\mathbf{x}_1, \dots, \mathbf{x}_m \in \mathcal{X}$ is called a positive (semi-)definite kernel.

By Lemma 1, a positive (semi-)definite kernel function $k(\cdot, \cdot)$ implicitly determines a feature mapping $\phi(\cdot)$ by the inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ in some Hilbert space \mathcal{H} . Such an \mathcal{H} induced from the kernel function $k(\cdot, \cdot)$ is termed the reproducing kernel Hilbert space (RKHS). For more details,

we refer interested readers to Chapter 3 of the book [Shawe-Taylor and Cristianini \(2004\)](#).

Function representer. Given m data points $\mathbf{x}_1, \dots, \mathbf{x}_m \in \mathcal{X}$, and a positive (semi-)definite kernel $k(\cdot, \cdot)$, we parameterize a function $f(\cdot) : \mathcal{X} \rightarrow \mathbb{R}$ as an expansion of kernel functions $k(\mathbf{x}_j, \cdot)$ over all data points:

$$f(\mathbf{x}) = \sum_{j=1}^m \alpha_j k(\mathbf{x}_j, \mathbf{x}), \quad \forall \mathbf{x} \in \text{domain } f. \quad (6)$$

Such an expansion is motivated from the *reproducing property* of the RKHS, and its expressiveness is backed by the representer theorem in [Schölkopf, Herbrich and Smola \(2001\)](#).

Assumption 1. We assume positive definite kernel $k(\cdot, \cdot)$. Thus the constructed kernel matrix is positive definite.

Definition 6. (Kernel based transformation). Given the point-cloud $\mathbf{P}_t = [\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_{m_t}] \in \mathbb{R}^{d \times m_t}$, and a query point $\mathbf{p} \in \mathbb{R}^d$, we propose a KBT, as:

$$\mathbf{y}_t(\mathbf{p}) \stackrel{\text{def}}{=} \mathbf{A}_t \mathbf{p} + \mathbf{a}_t + \mathbf{\Omega}_t^\top \mathbf{k}_t(\mathbf{p})$$

$$(\mathbf{A}_t \in \mathbb{R}^{d \times d}, \mathbf{a}_t \in \mathbb{R}^d, \mathbf{\Omega}_t \in \mathbb{R}^{m_t \times d}), \quad (7)$$

where:

$$\mathbf{k}_t(\mathbf{p}) = \begin{bmatrix} k(\mathbf{p}_1, \mathbf{p}) \\ \vdots \\ k(\mathbf{p}_{m_t}, \mathbf{p}) \end{bmatrix},$$

with $k(\cdot, \cdot)$ a user specified positive (semi-)definite kernel.

The deformable transformation $\mathbf{y}_t(\cdot)$ thus constructed contains two components: the affine part $(\mathbf{A}_t, \mathbf{a}_t)$ and the deformation part $\mathbf{\Omega}_t^\top \mathbf{k}_t(\cdot)$. The deformation part is an obvious extension from the expansion (6) to each of x -, y -, and z -coordinates. The motivation behind the affine part is that we require $\mathbf{y}_t(\cdot)$ to model global orientations and translations. Such information can indeed be lost in the kernel construction, for example if we choose $k(\mathbf{x}_i, \mathbf{x}_j) = \kappa(\|\mathbf{x}_i - \mathbf{x}_j\|_2)$. In addition, orientations and translations are global, meaning consistently applied to each point, which is not emphasized in the deformation part.

Regularization. For the KBT (7), we propose to use the following regularization:

$$\mathcal{R}_t = \mu_t \text{tr} \left(\mathbf{\Omega}_t^\top \mathbf{K}_t \mathbf{\Omega}_t \right), \quad (\mu_t > 0), \quad (8)$$

where we define the *kernel matrix* $\mathbf{K}_t \in \mathbb{R}^{m_t \times m_t}$:

$$\mathbf{K}_t = \begin{bmatrix} k(\mathbf{p}_1, \mathbf{p}_1) & \cdots & k(\mathbf{p}_1, \mathbf{p}_{m_t}) \\ \vdots & \ddots & \vdots \\ k(\mathbf{p}_{m_t}, \mathbf{p}_1) & \cdots & k(\mathbf{p}_{m_t}, \mathbf{p}_{m_t}) \end{bmatrix}. \quad (9)$$

The motivation of this regularization will be given shortly, near equation (16).

4.3 Operating on the Point-cloud

Given the point-cloud $\mathbf{P}_t = [\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_{m_t}] \in \mathbb{R}^{d \times m_t}$, we apply the deformable transformation $\mathbf{y}_t(\cdot)$ to each point of \mathbf{P}_t in sequence:

$$\mathbf{y}_t(\mathbf{P}_t) \stackrel{\text{def}}{=} [\mathbf{y}_t(\mathbf{p}_1), \mathbf{y}_t(\mathbf{p}_2), \dots, \mathbf{y}_t(\mathbf{p}_{m_t})].$$

For the LBW, the result is:

$$\mathbf{y}_t(\mathbf{P}_t) \stackrel{\text{def}}{=} \mathbf{W}_t^\top \underbrace{[\beta_t(\mathbf{p}_1) \quad \beta_t(\mathbf{p}_2) \quad \dots \quad \beta_t(\mathbf{p}_{m_t})]}_{\mathbf{B}_t(\mathbf{P}_t)}$$

$$\stackrel{\text{def}}{=} \mathbf{W}_t^\top \mathbf{B}_t(\mathbf{P}_t), \quad (\mathbf{W}_t \in \mathbb{R}^{l \times d}). \quad (10)$$

For the KBT, the result is:

$$\mathbf{y}_t(\mathbf{P}_t) \stackrel{\text{def}}{=} \mathbf{A}_t \mathbf{P}_t + \mathbf{a}_t \mathbf{1}^\top + \mathbf{\Omega}_t^\top \mathbf{K}_t,$$

$$(\mathbf{A}_t \in \mathbb{R}^{d \times d}, \mathbf{a}_t \in \mathbb{R}^d, \mathbf{\Omega}_t \in \mathbb{R}^{m_t \times d}), \quad (11)$$

where \mathbf{K}_t is defined in equation (9).

4.4 Derivation of the KBT from the LBW

Derivation of the deformation part. We consider the task of transforming the point-cloud \mathbf{P}_t to a given target point-cloud \mathbf{Z}_t , using the LBW and an identity regularization term. This task can be formulated as minimizing a regression cost:

$$\eta_t(\mathbf{W}_t) = \|\mathbf{W}_t^\top \mathbf{B}_t(\mathbf{P}_t) - \mathbf{Z}_t\|_{\mathcal{F}}^2 + \mu_t \|\mathbf{W}_t\|_{\mathcal{F}}^2. \quad (12)$$

Cost (12) is convex. Its global minimum is attained when the gradient vanishes:

$$\frac{\partial \eta_t}{\partial \mathbf{W}_t} = \mathbf{O}.$$

After computing the matrix differential, and with some trivial matrix calculations, we rewrite the above equation as:

$$\mathbf{W}_t = \mathbf{B}_t(\mathbf{P}_t) \underbrace{\left(-\frac{1}{\mu_t} \mathbf{W}_t^\top \mathbf{B}_t(\mathbf{P}_t) + \frac{1}{\mu_t} \mathbf{Z}_t \right)^\top}_{\mathbf{\Omega}_t}$$

$$\stackrel{\text{def}}{=} \mathbf{B}_t(\mathbf{P}_t) \mathbf{\Omega}_t. \quad (13)$$

In this form, $\mathbf{\Omega}_t$ is called the *dual variable*, as it converts the LBW to the KBT as:

$$\mathbf{W}_t^\top \mathbf{B}_t(\mathbf{P}_t) = \mathbf{\Omega}_t^\top \underbrace{\mathbf{B}_t(\mathbf{P}_t)^\top \mathbf{B}_t(\mathbf{P}_t)}_{\mathbf{K}_t} \stackrel{\text{def}}{=} \mathbf{\Omega}_t^\top \mathbf{K}_t, \quad (14)$$

$$\text{where } \mathbf{K}_t = \mathbf{B}_t(\mathbf{P}_t)^\top \mathbf{B}_t(\mathbf{P}_t). \quad (15)$$

Note that the dimension l of the feature space of $\beta_t(\cdot)$ may go to infinity; however we can still express $\mathbf{\Omega}_t^\top \mathbf{B}_t(\mathbf{P}_t)$ as $\mathbf{\Omega}_t^\top \mathbf{K}_t$ within m_t points in the kernel based model.

Derivation of the regularization. From equation (13), the regularization $\mu_t \|\mathbf{W}_t\|_{\mathcal{F}}^2$ can be reformulated with respect to the dual variable $\mathbf{\Omega}_t$ and the kernel matrix \mathbf{K}_t as:

$$\mu_t \|\mathbf{W}_t\|_{\mathcal{F}}^2 = \mu_t \|\mathbf{B}_t(\mathbf{P}_t) \mathbf{\Omega}_t\|_{\mathcal{F}}^2 = \mu_t \text{tr} \left(\mathbf{\Omega}_t^\top \mathbf{K}_t \mathbf{\Omega}_t \right), \quad (16)$$

which is how we obtain the regularization in equation (8).

Remark 3. (The independent affine transformation). In the KBT (7), we include an independent affine transformation, which is different from the LBWs. This is because for the LBWs, the affine transformation is typically implemented by the design of the basis function $\beta_t(\cdot)$. However, for the KBT, the kernel function $k(\cdot, \cdot)$ uniformly decides the elements in \mathbf{K}_t , excluding the possibility to use a handcrafted affine transformation. The usage of the independent affine transformation can be equivalently thought of as singling out the affine part in the LBW (4) as:

$$\mathbf{y}_t(\mathbf{p}) \stackrel{\text{def}}{=} \mathbf{A}_t \mathbf{p} + \mathbf{a}_t + \mathbf{W}_t^\top \beta_t(\mathbf{p}),$$

$$(\mathbf{A}_t \in \mathbb{R}^{d \times d}, \mathbf{a}_t \in \mathbb{R}^d, \mathbf{W}_t \in \mathbb{R}^{l \times d}).$$

In this form, $\beta_t(\cdot)$ only models deformations. By regularization (16), we see the affine part is free, which is in the same spirit of common LBWs.

5 Transformation Constraint

Definition 7. (Zero-centered point-cloud). A point-cloud \mathbf{M} is zero-centered if and only if $\mathbf{M}\mathbf{1} = \mathbf{0}$. In particular, $\bar{\mathbf{M}}$ is a zero-centered point-cloud of \mathbf{M} where:

$$\bar{\mathbf{M}} = \mathbf{M} - \frac{1}{m} \mathbf{M}\mathbf{1}\mathbf{1}^\top.$$

Definition 8. (Point-cloud covariance). We define the point-cloud covariance $\mathbb{C}\text{ov}(\mathbf{M}) = \bar{\mathbf{M}}\bar{\mathbf{M}}^\top$ with $\bar{\mathbf{M}} = \mathbf{M} - \frac{1}{m} \mathbf{M}\mathbf{1}\mathbf{1}^\top$ being the zero-centered point-cloud of \mathbf{M} .

We can simplify $\mathbb{C}\text{ov}(\mathbf{M})$ in Definition 8 to $\mathbf{M}\mathbf{M}^\top$ by requiring \mathbf{M} to be zero-centered as in Definition 7.

Lemma 2. (Lemma 2 in Bai and Bartoli (2022b)). For any \mathbf{M} , any rotation \mathbf{R} and any translation \mathbf{t} , we have:

$$\mathbb{C}\text{ov}(\mathbf{R}\mathbf{M} + \mathbf{t}\mathbf{1}^\top) = \mathbf{R} \mathbb{C}\text{ov}(\mathbf{M}) \mathbf{R}^\top. \quad (17)$$

Lemma 2 shows that: a) $\mathbb{C}\text{ov}(\mathbf{M})$ is only related to rotations, and b) the eigenvalues of $\mathbb{C}\text{ov}(\mathbf{M})$ are preserved when applying rotations to \mathbf{M} .

Definition 9. (Eigenvalues of point-cloud covariance). We denote $\mathbf{\Lambda} = \text{diag}(\lambda_1, \dots, \lambda_d)$, where $\lambda_1 \geq \dots \geq \lambda_d \geq 0$ are the d eigenvalues of the point-cloud covariance $\mathbb{C}\text{ov}(\mathbf{M})$.

In addition, Lemma 2 provides a means to diagonalize $\mathbb{C}\text{ov}(\mathbf{M})$ by rotating the point-cloud \mathbf{M} . We consider the eigenvalue decomposition:

$$\mathbb{C}\text{ov}(\mathbf{M}) = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^\top = \sum_{k=1}^d \lambda_k \mathbf{q}_k \mathbf{q}_k^\top,$$

$$\text{where } \mathbf{Q} \stackrel{\text{def}}{=} [\mathbf{q}_1 \ \dots \ \mathbf{q}_d].$$

It is always possible to have $\mathbf{Q} \in \text{SO}(d)$ by flipping the signs of \mathbf{q}_k . In Lemma 2, if we use $\mathbf{R} = \mathbf{Q}^\top$, then:

$$\mathbb{C}\text{ov}(\mathbf{Q}^\top \mathbf{M}) = \mathbf{Q}^\top \mathbb{C}\text{ov}(\mathbf{M}) \mathbf{Q} = \mathbf{\Lambda},$$

where $\mathbb{C}\text{ov}(\mathbf{Q}^\top \mathbf{M})$ is of diagonal form.

We present the general result below, and give an illustration in Figure 2.

Theorem 1. For any \mathbf{M} , it is always possible to find a rigid transformation (\mathbf{R}, \mathbf{t}) such that the rigidly transformed \mathbf{M}_r :

$$\mathbf{M}_r = \mathbf{R}\mathbf{M} + \mathbf{t}\mathbf{1}^\top, \quad \mathbf{R} \in \text{SO}(d), \mathbf{t} \in \mathbb{R}^d,$$

is a zero-centered point-cloud and has a diagonal form point-cloud covariance:

$$\mathbf{M}_r \mathbf{1} = \mathbf{0}, \quad \mathbb{C}\text{ov}(\mathbf{M}_r) = \mathbf{M}_r \mathbf{M}_r^\top = \mathbf{\Lambda},$$

where $\mathbf{\Lambda}$, as defined in Definition 9, contains the eigenvalues of the point-cloud covariances $\mathbb{C}\text{ov}(\mathbf{M})$ and $\mathbb{C}\text{ov}(\mathbf{M}_r)$.

Proof. It suffices to set $\mathbf{R} = \mathbf{Q}^\top$ and $\mathbf{t} = -\frac{1}{m} \mathbf{Q}^\top \mathbf{M}\mathbf{1}$.

We are interested in the geometry *i.e.*, the shape, of point-cloud \mathbf{M} , discarding its position and orientation in the global coordinate system. Thus we propose to solve for an \mathbf{M} that is zero-centered with diagonal covariance:

$$\text{constraints} \begin{cases} \mathbf{M}\mathbf{1} = \mathbf{0} & (18) \\ \mathbf{M}\mathbf{M}^\top = \mathbf{\Lambda} = \text{diag}(\lambda_1, \dots, \lambda_d), & (19) \end{cases}$$

where $\lambda_1 \geq \dots \geq \lambda_d \geq 0$ are *unknown parameters* representing the eigenvalues of the point-cloud covariance.

Remark 4. The constraints $\mathbf{M}\mathbf{1} = \mathbf{0}$, $\mathbf{M}\mathbf{M}^\top = \mathbf{\Lambda}$ allow \mathbf{M} to take all possible geometries, as implied by Theorem 1.

6 Globally Optimal Solution

6.1 Formulation of Deformable SLAM

Using deformable transformation (11) and regularization (8), we write the cost function at time t as:

$$\varphi_t(\mathbf{A}_t, \mathbf{a}_t, \mathbf{\Omega}_t, \mathbf{M}) = \left\| \mathbf{A}_t \mathbf{P}_t + \mathbf{a}_t \mathbf{1}^\top + \mathbf{\Omega}_t^\top \mathbf{K}_t - \mathbf{M}\mathbf{\Gamma}_t \right\|_{\mathcal{F}}^2$$

$$+ \mu_t \text{tr} \left(\mathbf{\Omega}_t^\top \mathbf{K}_t \mathbf{\Omega}_t \right). \quad (20)$$

Then we use constraints $\mathbf{M}\mathbf{1} = \mathbf{0}$ and $\mathbf{M}\mathbf{M}^\top = \mathbf{\Lambda}$ to implicitly specify the free coordinate frame where to express the solution. We complete formulation (3) as the following optimization problem:

$$\min_{\{\mathbf{A}_t, \mathbf{a}_t, \mathbf{\Omega}_t\}, \mathbf{M}} \sum_{t=1}^n \varphi_t(\mathbf{A}_t, \mathbf{a}_t, \mathbf{\Omega}_t, \mathbf{M}) \quad (21)$$

$$\text{s.t. } \mathbf{M}\mathbf{1} = \mathbf{0}, \mathbf{M}\mathbf{M}^\top = \mathbf{\Lambda}.$$

In the remainder of this section, we derive the globally optimal solution to problem (21) in function of the unknown $\mathbf{\Lambda}$. We will recast problem (21) as a special eigenvalue problem, and derive the solution in closed-form, see Bai and Bartoli (2022b) for affine models and TPS warps (a brief recapitulation is provided in Appendix M).

6.2 Reduced Problem in \mathbf{M}

We notice that in problem (21), the transformation parameters \mathbf{A}_t , \mathbf{a}_t and $\mathbf{\Omega}_t$ are linearly dependent on \mathbf{M} . This presents a separable structure and allows us to reduce the optimization to \mathbf{M} only using the variable projection method Golub and Pereyra (2003).

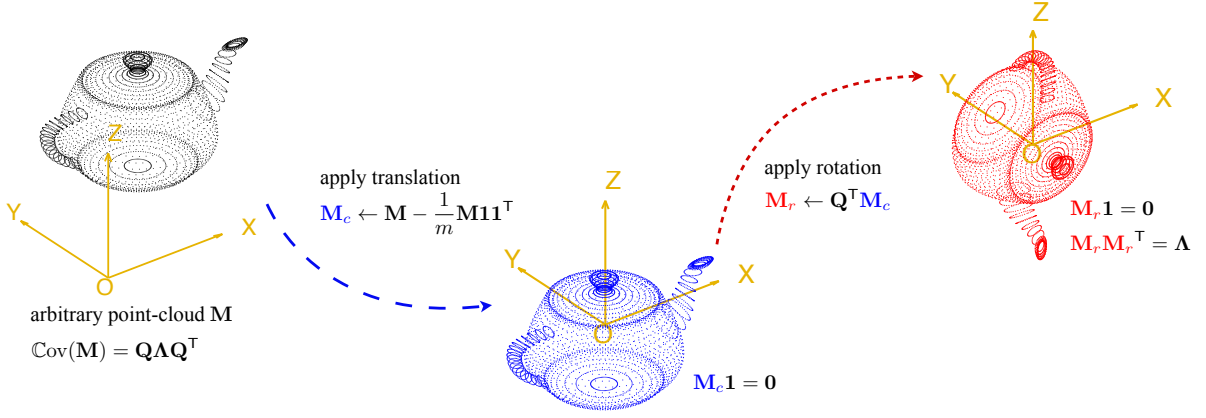


Figure 2. The proposed constraints $M\mathbf{1} = \mathbf{0}$, $MM^T = \Lambda$ with an unknown diagonal matrix Λ , allow all possible geometries (*i.e.*, shapes) for M . This is explained as follows. We assume M to be an arbitrary point-cloud, and denote $\text{Cov}(M) = Q\Lambda Q^T$ the eigenvalue decomposition of its point-cloud covariance. Then we can rigidly transform M to M_r , which has exactly the same geometry as M while $M_r\mathbf{1} = \mathbf{0}$, $M_r M_r^T = \Lambda$. Note that as M is unknown, we do not know the eigenvalues Λ . Fortunately Λ is never required explicitly to derive the globally optimal solution to the GPA formulation, and thus can be estimated afterwards. The constraints $M\mathbf{1} = \mathbf{0}$, $MM^T = \Lambda$ implicitly specify the global coordinate frame in Figure 1, by requiring M positioned this way.

The linear dependence of A_t , a_t and Ω_t on M . We first notice that in problem (21), given M , the summands in the cost function become independent. This allows us to derive the dependence of A_t , a_t and Ω_t on M by solving a linear least squares (LLS) optimization from cost (20):

$$\min_{\{A_t, a_t, \Omega_t\}} \varphi_t(A_t, a_t, \Omega_t, M), \quad \text{given } M. \quad (22)$$

With some trivial calculations, see appendix C.1, we write:

$$[A_t, a_t, \Omega_t^T] = M\Gamma_t [\tilde{P}_t^T \quad K_t] \Delta_t^\dagger + F_t (\mathbf{I} - \Delta_t \Delta_t^\dagger), \quad (23)$$

where $\tilde{P}_t = [P_t^T, \mathbf{1}]^T$, and $F_t \in \mathbb{R}^{d \times (m_t + d + 1)}$ is a free matrix. Δ_t^\dagger is the Moore–Penrose pseudo-inverse of a positive definite (or positive semi-definite) matrix Δ_t defined as:

$$\Delta_t \stackrel{\text{def}}{=} \begin{bmatrix} \tilde{P}_t \tilde{P}_t^T & \tilde{P}_t K_t \\ K_t \tilde{P}_t^T & K_t K_t + \mu_t K_t \end{bmatrix}.$$

Remark 5. The free matrix F_t is used to describe general solutions of the LLS problem (22), in case that Δ_t is rank deficient (and thus not invertible). If Δ_t is positive definite, then F_t is not required since $F_t(\mathbf{I} - \Delta_t \Delta_t^\dagger) = \mathbf{0}$.

Lemma 3. If K_t is positive definite and $\mu_t > 0$, then Δ_t is positive definite if and only if $\tilde{P}_t \tilde{P}_t^T$ is positive definite.

Proof. See Appendix D.

Otherwise stated, Δ_t is invertible if and only if \tilde{P}_t has full row rank which is the case if the point-cloud P_t is not degenerate, *e.g.*, not flat if $d = 3$ (namely residing in a plane in the 3D space) or not a line if $d = 2$.

The reduced problem in M . Substituting equation (23) into the cost (20), we obtain a cost with respect to M only, denoted as $\varphi_t(M)$. With some trivial calculations, see appendix C.2, we show:

$$\varphi_t(M) = \text{tr}(M\Gamma_t Q_t \Gamma_t^T M^T),$$

where Q_t is independent of the free matrix F occurring in equation (23), defined as:

$$Q_t \stackrel{\text{def}}{=} \mathbf{I} - [\tilde{P}_t^T \quad K_t] \Delta_t^\dagger \begin{bmatrix} \tilde{P}_t \\ K_t \end{bmatrix}.$$

Lastly problem (21) is reduced to:

$$\min_M \text{tr}(MQM^T) \quad (24)$$

$$\text{s.t. } M\mathbf{1} = \mathbf{0}, \quad MM^T = \Lambda,$$

with:

$$Q = \sum_{t=1}^n \Gamma_t Q_t \Gamma_t^T.$$

Problem (24) is an optimization problem with respect to M only. In particular, problem (24) can be solved globally in closed-form if the all-one vector $\mathbf{1}$ is an eigenvector of Q .

Properties of Q_t and Q . We can work out closed-form expressions for Δ_t^\dagger using the Schur complement Gallier (2010), see Appendix E. With some trivial calculations, see Appendix F, we show that Q_t can be rewritten as follows:

$$Q_t = (\mathbf{I} - \mathcal{P}_t) - (\mathbf{I} - \mathcal{P}_t) K_t S_t^{-1} K_t (\mathbf{I} - \mathcal{P}_t), \quad (25)$$

with $\mathcal{P}_t \stackrel{\text{def}}{=} \tilde{P}_t^T (\tilde{P}_t \tilde{P}_t^T)^\dagger \tilde{P}_t$, and:

$$S_t \stackrel{\text{def}}{=} K_t (\mathbf{I} - \mathcal{P}_t) K_t + \mu_t K_t, \quad (26)$$

being symmetric positive definite (and thus invertible), since we assume K_t is chosen positive definite and $\mu_t > 0$.

Proposition 1. If K_t is chosen positive definite and $\mu_t > 0$, then Q_t is symmetric positive semidefinite where:

- $\mathbf{I} \succeq \mathbf{I} - \mathcal{P}_t \succeq Q_t \succeq \mathbf{0}$
- $Q_t \mathbf{1}_{m_t} = \mathbf{0}$

where $A \succeq B$ means $A - B$ is positive semidefinite.

Proof. See Appendix G.

Theorem 2. In problem (24), $\mathcal{Q}\mathbf{1} = \mathbf{0}$ which means $\mathbf{1}$ is an eigenvector of \mathcal{Q} corresponding to eigenvalue 0.

Proof. This is obvious as $\Gamma_t^\top \mathbf{1}_m = \mathbf{1}_{m_t}$ and $\mathbf{Q}_t \mathbf{1}_{m_t} = \mathbf{0}$.

Remark 6. The expression of \mathbf{Q}_t in equation (25) is much more elegant than the one in Bai and Bartoli (2022a). See Appendix I for details.

6.3 Globally Optimal Estimate of \mathbf{M}

We recapitulate necessary results to describe the globally optimal solution to problem (24).

Definition 10. (The d top eigenvectors and the d bottom eigenvectors). We consider a symmetric matrix $\mathbf{\Pi} \in \mathbb{R}^{m \times m}$ and its eigenvalue decomposition:

$$\mathbf{\Pi} = \mathbf{U}\mathbf{\Sigma}\mathbf{U}^\top = \sum_{k=1}^m \sigma_k \mathbf{u}_k \mathbf{u}_k^\top,$$

with $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_m]$ being orthonormal, and $\mathbf{\Sigma} = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_m)$ whose diagonal elements are arranged in the non-ascending order as $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_m$. We term:

$$\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_d,$$

in sequence the d top eigenvectors of $\mathbf{\Pi}$, and:

$$\mathbf{u}_m, \mathbf{u}_{m-1}, \dots, \mathbf{u}_{m-d+1},$$

in sequence the d bottom eigenvectors of $\mathbf{\Pi}$.

Lemma 4. We consider a symmetric matrix $\mathbf{\Pi} \in \mathbb{R}^{m \times m}$, and $\mathbf{X} \in \mathbb{R}^{m \times d}$. Let $\mathbf{\Lambda} = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_d)$ be a diagonal matrix with $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_d \geq 0$. If \mathbf{u} is an eigenvector of the symmetric matrix $\mathbf{\Pi}$, then we have:

1. The globally optimal solution of:

$$\begin{aligned} \max_{\mathbf{X}} \quad & \text{tr}(\mathbf{X}^\top \mathbf{\Pi} \mathbf{X} \mathbf{\Lambda}) \\ \text{s.t.} \quad & \mathbf{X}^\top \mathbf{X} = \mathbf{I}, \mathbf{X}^\top \mathbf{u} = \mathbf{0}, \end{aligned} \quad (27)$$

is $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_d]$, where $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_d$ are the d top eigenvectors of $\mathbf{\Pi}$ excluding \mathbf{u} .

2. The globally optimal solution of:

$$\begin{aligned} \min_{\mathbf{X}} \quad & \text{tr}(\mathbf{X}^\top \mathbf{\Pi} \mathbf{X} \mathbf{\Lambda}) \\ \text{s.t.} \quad & \mathbf{X}^\top \mathbf{X} = \mathbf{I}, \mathbf{X}^\top \mathbf{u} = \mathbf{0}, \end{aligned} \quad (28)$$

is $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_d]$, where $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_d$ are the d bottom eigenvectors of $\mathbf{\Pi}$ excluding \mathbf{u} .

Proof. An initial version of the proof was given in Bai and Bartoli (2022b). Here we provide a conciser version without any further assumption on $\mathbf{\Pi}$, see Appendix B. Some preliminaries are provided in Appendix A.

Theorem 3. The globally optimal solution to problem (24) is in closed-form:

$$\mathbf{M} = \sqrt{\mathbf{\Lambda}} \mathbf{X}^\top, \quad \text{where } \mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_d] \in \mathbb{R}^{m \times d},$$

where $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_d$ in sequence are the d bottom eigenvectors of \mathcal{Q} excluding the vector $\mathbf{1}$.

Proof. In problem (24), by letting $\mathbf{M} = \sqrt{\mathbf{\Lambda}} \mathbf{X}^\top$, we have:

$$\begin{aligned} \min_{\mathbf{X}} \quad & \text{tr}(\mathbf{X}^\top \mathcal{Q} \mathbf{X} \mathbf{\Lambda}) \\ \text{s.t.} \quad & \mathbf{X}^\top \mathbf{X} = \mathbf{I}, \mathbf{X}^\top \mathbf{1} = \mathbf{0}. \end{aligned} \quad (29)$$

From Proposition 2, we see $\mathbf{1}$ is an eigenvector of \mathcal{Q} (with eigenvalue 0). The result is immediate by applying Lemma 4.

Remark 7. (Shifting eigenvectors). Since $\mathcal{Q}\mathbf{1} = \mathbf{0}$, we can shift the eigenvector $\mathbf{1}$ of \mathcal{Q} to the top by letting:

$$\mathcal{Q}' = \mathcal{Q} + n\mathbf{1}\mathbf{1}^\top,$$

and solve for the d bottom eigenvectors of \mathcal{Q}' to form \mathbf{X} .

6.4 Globally Optimal Estimate of the Deformable Transformation

Upon obtaining the estimate of \mathbf{M} , we can decide the optimal transformation parameters. From equation (23), we set $\mathbf{F}_t = \mathbf{O}$, and take the specific solution:

$$[\mathbf{A}_t, \mathbf{a}_t], \mathbf{\Omega}_t^\top = \mathbf{M} \Gamma_t [\tilde{\mathbf{P}}_t^\top \quad \mathbf{K}_t] \mathbf{\Delta}_t^\dagger. \quad (30)$$

We expand $\mathbf{\Delta}_t^\dagger$ in the term $[\tilde{\mathbf{P}}_t^\top \quad \mathbf{K}_t] \mathbf{\Delta}_t^\dagger$, see equation (50) in Appendix F, and write the final result as:

$$[\mathbf{A}_t, \mathbf{a}_t] = \mathbf{M} \Gamma_t (\mathbf{I} - \mathbf{H}_t \mathbf{K}_t) \tilde{\mathbf{P}}_t^\dagger \quad (31)$$

$$\mathbf{\Omega}_t^\top = \mathbf{M} \Gamma_t \mathbf{H}_t, \quad (32)$$

with:

$$\mathbf{H}_t \stackrel{\text{def}}{=} (\mathbf{I} - \mathcal{P}_t) \mathbf{K}_t \mathbf{S}_t^{-1}.$$

Proposition 2. If \mathbf{K}_t is chosen positive definite and $\mu_t > 0$, then \mathbf{H}_t is symmetric positive definite where:

- $\mathbf{H}_t = \mathbf{H}_t^\top$
- $\mathbf{Q}_t = \mu_t \mathbf{H}_t$

Proof. See Appendix H.

The optimal KBT $\mathbf{y}_t(\mathbf{p})$ in equation (7), for an arbitrary query point \mathbf{p} , can be written as:

$$\mathbf{y}_t(\mathbf{p}) = \mathbf{M} \Gamma_t \left((\mathbf{I} - \mathbf{H}_t \mathbf{K}_t) \tilde{\mathbf{P}}_t^\dagger \begin{bmatrix} \mathbf{p} \\ 1 \end{bmatrix} + \mathbf{H}_t \mathbf{k}_t(\mathbf{p}) \right). \quad (33)$$

Since $\mathbf{M} = \sqrt{\mathbf{\Lambda}} \mathbf{X}^\top$, we establish the estimate of both \mathbf{M} and $\mathbf{y}_t(\cdot)$ up to an unknown $\mathbf{\Lambda}$. It should be noted that any $\mathbf{\Lambda}$ admits a globally optimal solution to problem (21). Thus this is what we can maximally achieve by solving problem (21).

6.5 Coordinate Transformation of Data

In the data acquisition process, the point-cloud data \mathbf{P}_t can be expressed in any user defined coordinate frames.

Definition 11. (Coordinate transformation). We refer to the coordinate transformation of data \mathbf{P}_t as $\mathbf{P}_t = \mathbf{R}_t \mathbf{P}_t + \mathbf{t}_t \mathbf{1}^\top$, with $(\mathbf{R}_t, \mathbf{t}_t)$ being any arbitrary rigid transformation.

Ideally, we want the estimate of \mathbf{M} to be invariant under coordinate transformations of data. By equation (25), \mathcal{Q}_t can be expressed with $\mathbf{I} - \mathcal{P}_t$ and \mathbf{K}_t . If both \mathcal{P}_t and \mathbf{K}_t are invariant to the coordinate transformation of \mathbf{P}_t , then \mathcal{Q}_t is invariant to the coordinate transformation, thus so will be \mathcal{Q} .

Lemma 5. (Lemma 5 in Bai and Bartoli (2022b)). The orthogonal projection matrix $\mathcal{P}_t = \tilde{\mathbf{P}}_t^\top (\tilde{\mathbf{P}}_t \tilde{\mathbf{P}}_t^\top)^\dagger \tilde{\mathbf{P}}_t$ remains unchanged under any coordinate transformation of \mathbf{P}_t .

Proof. See Appendix J.

Proposition 3. If the kernel function $k(\cdot, \cdot)$ is chosen as the RBFs, *i.e.*, $k(\mathbf{x}_i, \mathbf{x}_j) = \phi(\|\mathbf{x}_i - \mathbf{x}_j\|)$, where $k(\mathbf{x}_i, \mathbf{x}_j)$ is only related to the Euclidean distance of \mathbf{x}_i and \mathbf{x}_j , then \mathbf{K}_t is invariant to the coordinate transformation.

Proposition 4. If the kernel function $k(\cdot, \cdot)$ is chosen as the RBFs, then matrix \mathcal{Q} in problem (24) remains unchanged. In this case, the optimal estimate of \mathbf{M} remains unchanged under the coordinate transformation.

7 Global Scale Ambiguity $\sqrt{\Lambda}$

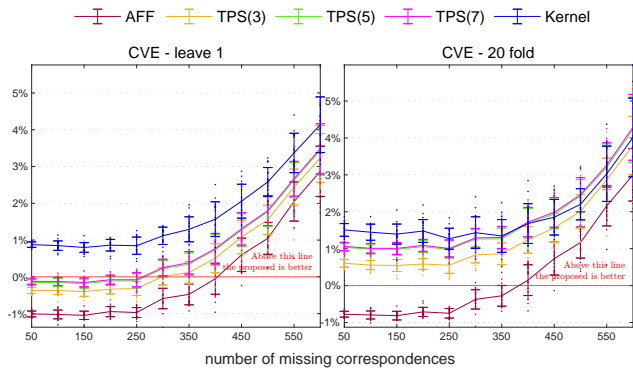


Figure 3. Performance improvements of the proposed $\sqrt{\Lambda}$ estimation method over the one in Bai and Bartoli (2022b). We gradually remove the correspondences in the HandBag dataset Gallardo, Collins and Bartoli (2017) (with 155×8 correspondences in total), and report statistics of 20-trial Monte-Carlo runs based on leave-1 and 20-fold cross-validations Bai and Bartoli (2022a), for five transformation models: affine, TPS with 3×3 , 5×5 and 7×7 control points, and the proposed KBT.

In Bai and Bartoli (2022b), the authors proposed a method to estimate the global scale ambiguities (*i.e.*, the diagonal elements of $\sqrt{\Lambda}$) using pairwise rigid Procrustes analysis. The method in Bai and Bartoli (2022b) requires the existence of some points to be globally visible across all point-clouds. In this section, we propose a novel method to estimate Λ which does not require such visibility assumptions.

In addition, the method in this work solves $\sqrt{\Lambda}$ by a global optimization formulation, whereas the method in Bai and Bartoli (2022b) relies on local pairwise registrations. Thus the $\sqrt{\Lambda}$ estimation method proposed in this work can be superior even if the globally visible correspondences are available. We provide a justification to this claim in Figure 3.

7.1 As Rigid as Possible

We want the deformable transformation to be as-rigid-as-possible, that means we want $\Phi_t(\cdot)$ in equation (2) to be close to an identity mapping. In particular, without the deformation $\Phi_t(\cdot)$, we have:

$$\mathbf{y}_t(\mathbf{P}_t) = \mathbf{R}_t \mathbf{P}_t + \mathbf{t}_t \mathbf{1}^\top = \mathbf{M} \Gamma_t = \sqrt{\Lambda} \mathbf{X}^\top \Gamma_t.$$

This motivates us to characterize $\sqrt{\Lambda}$ by an optimization formulation as follows:

$$\min_{\{\mathbf{R}_t, \mathbf{t}_t\}, \Lambda, \mathbf{R}_g} \sum_{t=1}^n \left\| \mathbf{R}_t \mathbf{P}_t + \mathbf{t}_t \mathbf{1}^\top - \sqrt{\Lambda} \mathbf{R}_g \mathbf{G}_t \right\|_{\mathcal{F}}^2, \quad (34)$$

with:

$$\mathbf{G}_t = \mathbf{X}^\top \Gamma_t.$$

where $(\mathbf{R}_t, \mathbf{t}_t)$ denotes the rigid transformation. Here we have introduced an orthonormal matrix $\mathbf{R}_g \in O(d)$ for a reason we will explain later in Section 8.1. At the moment, it suffices to think of \mathbf{R}_g as an identity matrix.

7.2 Reduced Formulation

In formulation (34), we notice \mathbf{t}_t is linearly dependent on the other parameters $\mathbf{R}_t, \Lambda, \mathbf{R}_g$. Thus formulation (34) admits a separable structure which allows us to eliminate \mathbf{t}_t from the formulation Golub and Pereyra (2003). In specific, given \mathbf{R}_t, Λ and \mathbf{R}_g , the estimates of \mathbf{t}_t are expressed as:

$$\mathbf{t}_t = -\frac{1}{m_t} \left(\mathbf{R}_t \mathbf{P}_t - \sqrt{\Lambda} \mathbf{R}_g \mathbf{G}_t \right) \mathbf{1}, \quad (t \in [1 : n]). \quad (35)$$

After substituting equation (35) into formulation (34), we obtain a reduced problem:

$$\min_{\{\mathbf{R}_t\}, \Lambda, \mathbf{R}_g} \sum_{t=1}^n \left\| \mathbf{R}_t \tilde{\mathbf{P}}_t - \sqrt{\Lambda} \mathbf{R}_g \tilde{\mathbf{G}}_t \right\|_{\mathcal{F}}^2, \quad (36)$$

with $\tilde{\mathbf{P}}_t = \mathbf{P}_t - \frac{1}{m_t} \mathbf{P}_t \mathbf{1} \mathbf{1}^\top$ and $\tilde{\mathbf{G}}_t = \mathbf{G}_t - \frac{1}{m_t} \mathbf{G}_t \mathbf{1} \mathbf{1}^\top$.

7.3 Closed-form Evaluation of $\sqrt{\Lambda}$ and \mathbf{R}_g

From formulation (36), we consider an affine relaxation of $\sqrt{\Lambda} \mathbf{R}_g$, and establish its linear dependence on \mathbf{R}_t as:

$$\sqrt{\Lambda} \mathbf{R}_g \leftarrow \mathbf{R}_t \tilde{\mathbf{P}}_t \tilde{\mathbf{G}}_t^\dagger, \quad (t \in [1 : n]). \quad (37)$$

From relaxation (37), we then compute $(\sqrt{\Lambda} \mathbf{R}_g)^\top \sqrt{\Lambda} \mathbf{R}_g$ and apply the orthonormal constraint $\mathbf{R}_t^\top \mathbf{R}_t = \mathbf{I}$, as:

$$\mathbf{R}_g^\top \Lambda \mathbf{R}_g = \left(\mathbf{R}_t \tilde{\mathbf{P}}_t \tilde{\mathbf{G}}_t^\dagger \right)^\top \mathbf{R}_t \tilde{\mathbf{P}}_t \tilde{\mathbf{G}}_t^\dagger = (\tilde{\mathbf{P}}_t \tilde{\mathbf{G}}_t^\dagger)^\top \tilde{\mathbf{P}}_t \tilde{\mathbf{G}}_t^\dagger.$$

Given n point-clouds, we take the average with respect to t which corresponds to the maximum likelihood estimate:

$$\mathbf{R}_g^\top \Lambda \mathbf{R}_g = \frac{1}{n} \sum_{t=1}^n (\tilde{\mathbf{P}}_t \tilde{\mathbf{G}}_t^\dagger)^\top \tilde{\mathbf{P}}_t \tilde{\mathbf{G}}_t^\dagger \stackrel{\text{def}}{=} \mathcal{L}. \quad (38)$$

We see the lefthand of equation (38) forms the eigenvalue decomposition of \mathcal{L} . We thus compute the diagonals of Λ as the eigenvalues of \mathcal{L} , and the rows of \mathbf{R}_g as the corresponding eigenvectors. We arrange the eigenvalues of \mathcal{L} in the non-descending order. We notice \mathcal{L} is positive definite (or semi-definite), thus the eigenvalues of \mathcal{L} are non-negative. Therefore $\sqrt{\Lambda}$ is well-defined in the real domain.

Remark 8. The idea to factorize $\sqrt{\Lambda}$ from equation (38) is maturer than the initial version in Bai and Bartoli (2022a). In particular, the eigenvalue decomposition in equation (38) was not realized in Bai and Bartoli (2022a). Critically, the method in Bai and Bartoli (2022a) may lead to negative diagonals in Λ , causing undefined $\sqrt{\Lambda}$.

Initialization of \mathbf{R}_t and \mathbf{t}_t . Given $\sqrt{\Lambda}$ and \mathbf{R}_g , the rotation \mathbf{R}_t can be solved from formulation (36) in closed-form by the special orthogonal Procrustes analysis Arun, Huang and Blostein (1987); Horn, Hilden and Negahdaripour (1988) between $\bar{\mathbf{P}}_t$ and $\sqrt{\Lambda}\mathbf{R}_g\bar{\mathbf{G}}_t$. Alternatively, we can simply round $\sqrt{\Lambda}\mathbf{R}_g\mathbf{L}_t$ to a proper rotation matrix \mathbf{R}_t . Afterwards, we compute \mathbf{t}_t from equation (35).

7.4 Iterative Refinement

We can solve formulation (36) exactly using iterative NLS optimization techniques, *e.g.*, by Gauss-Newton or Levenberg-Marquardt. The rotation \mathbf{R}_t can be readily handled with Lie group techniques. The diagonal elements of $\sqrt{\Lambda}$ are constrained to be non-negative, thus requiring special consideration.

Reflection. We notice that the columns of \mathbf{X} (as the eigenvectors of \mathbf{Q}), and thus the rows of \mathbf{G}_t (and $\bar{\mathbf{G}}_t$), are defined up to signs. This means that if we flip the sign of one column in \mathbf{X} , the solution is still optimal. Using a specific \mathbf{X} , the optimal \mathbf{R}_t in formulation (36) may have negative determinants $\det(\mathbf{R}_t) = -1$, which is called a *reflection*.

We thus extend formulation (36) using $\boldsymbol{\eta} \in \mathbb{R}^d$ to handle the possible reflections caused by the specification of \mathbf{X} :

$$\min_{\{\mathbf{R}_t \in \text{SO}(d)\}, \boldsymbol{\eta} \in \mathbb{R}^d} \sum_{t=1}^n \|\mathbf{R}_t \bar{\mathbf{P}}_t - \text{diag}(\boldsymbol{\eta}) \bar{\mathbf{G}}_t\|_{\mathcal{F}}^2, \quad (39)$$

where $\text{diag}(\boldsymbol{\eta})$ is a diagonal matrix taking the components in $\boldsymbol{\eta}$. We further denote $\text{sign}(\boldsymbol{\eta})$ a vector containing the signs of the components in $\boldsymbol{\eta}$.

If there exist reflections, the optimal $\boldsymbol{\eta}$ in formulation (39) can have negative components. In this case, we flip the sign of the columns of \mathbf{X} accordingly. In general, we set:

$$\begin{aligned} \mathbf{X} &\leftarrow \mathbf{X} \text{diag}(\text{sign}(\boldsymbol{\eta})) \\ \sqrt{\Lambda} &\leftarrow \text{diag}(\boldsymbol{\eta}) \text{diag}(\text{sign}(\boldsymbol{\eta})). \end{aligned}$$

These operations preserve the optimality of both \mathbf{X} and $\sqrt{\Lambda}$.

Remark 9. It should be noted that the rigid transformations $(\mathbf{R}_t, \mathbf{t}_t)$ solved from formulation (34) are different from the ambiguous poses defined in equation (1). In essence, formulation (34) approximately solves Rigid-GPA, by constraining \mathbf{M} as $\mathbf{M} = \sqrt{\Lambda}\mathbf{X}^\top$. Thus the optimal $(\mathbf{R}_t, \mathbf{t}_t)$ obtained from formulation (34) are similar to the poses obtained from Rigid-GPA, as shown in Figure 4.

8 Degeneracies

8.1 Zero-deformation and Noise-free

We consider the case where matrix \mathbf{Q} in problem (24) has $d+1$ zero eigenvalues, where one of them corresponds to the eigenvector $\mathbf{1}$ (Theorem 2). Following Remark 7, we can drop the eigenvector $\mathbf{1}$ easily by solving the d bottom eigenvectors of $\mathbf{Q}' = \mathbf{Q} + n\mathbf{1}\mathbf{1}^\top$ instead to form the columns of \mathbf{X} . Note that in this case:

$$\mathbf{Q}\mathbf{X} = \mathbf{O} = \mathbf{X}\mathbf{U}_g \text{diag}(\mathbf{0}),$$

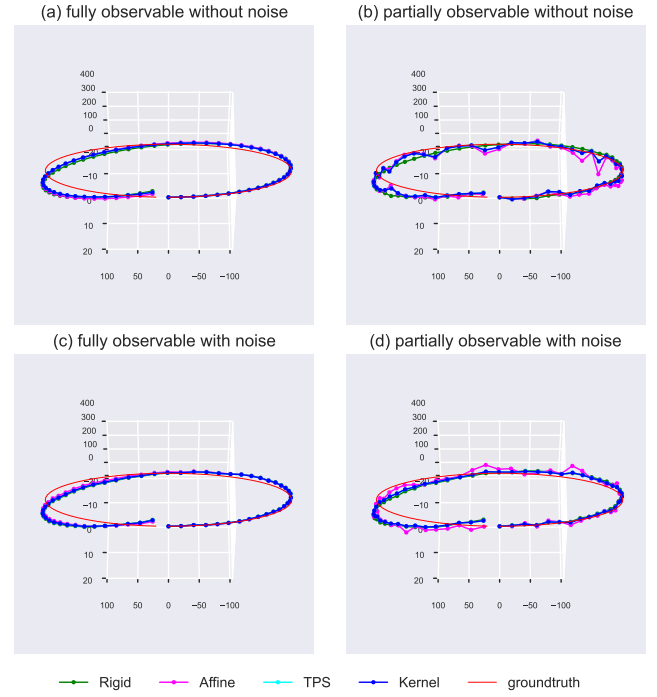


Figure 4. Estimated trajectories of Rigid-GPA, Affine-GPA, TPS-GPA, and Kernel-GPA on the liver dataset.

where \mathbf{U}_g is an arbitrary orthonormal matrix, *i.e.*, $\mathbf{U}_g\mathbf{U}_g^\top = \mathbf{U}_g^\top\mathbf{U}_g = \mathbf{I}$. We see that any $\mathbf{X}_g \stackrel{\text{def}}{=} \mathbf{X}\mathbf{U}_g$ is a valid solution as $\mathbf{X}_g\mathbf{X}_g^\top = \mathbf{I}$. Thus, the optimal solution of problem (24) will be defined up to an arbitrary \mathbf{U}_g as:

$$\mathbf{M} = \sqrt{\Lambda}\mathbf{U}_g^\top\mathbf{X}^\top.$$

In this case, \mathbf{U}_g is fundamentally ambiguous, which means there is no way to decide \mathbf{U}_g from problem (24) directly.

This is the reason why we introduce \mathbf{R}_g in formulation (34, 36), where we essentially denote $\mathbf{R}_g = \mathbf{U}_g^\top$. If such a degeneracy occurs, we can factorize \mathbf{R}_g (and thus \mathbf{U}_g) from equation (38) by the eigenvalue decomposition.

8.2 Flat Point-cloud in 3D Space

The cost function. We consider the case of $d=3$ and denote $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3]$. With some matrix manipulations, see Appendix K, we show that the cost of problem (29) can be rewritten as:

$$\begin{aligned} \text{tr}(\mathbf{X}^\top \mathbf{Q} \mathbf{X} \Lambda) = & \sum_{k=1}^3 \lambda_k \left\| \begin{bmatrix} \sqrt{\mathbf{Q}_1} (\mathbf{I} - \mathcal{P}_1) \Gamma_1^\top \\ \vdots \\ \sqrt{\mathbf{Q}_n} (\mathbf{I} - \mathcal{P}_n) \Gamma_n^\top \end{bmatrix} \mathbf{x}_k \right\|_{\mathcal{F}}^2. \quad (40) \end{aligned}$$

Matrix \mathcal{P}_t is the orthogonal projector to the range space of $\tilde{\mathbf{P}}_t^\top$ Meyer (2000). In particular, if \mathbf{y} lies in the range space of $\tilde{\mathbf{P}}_t^\top$, then $(\mathbf{I} - \mathcal{P}_t)\mathbf{y} = \mathbf{0}$. Hence, this cost is zero (and thus minimized) if each of $\Gamma_t^\top \mathbf{x}_k$ can be chosen from the respective range space of $\tilde{\mathbf{P}}_t^\top$, which is usually impossible due to the existence of noise and deformations.

The canonical planar point-cloud. If the point-cloud \mathbf{P}_t is flat, then \mathbf{P}_t can be rigidly transformed to the xy -plane. In

addition, from Theorem 1, we conclude that for a flat \mathbf{P}_t , there exists a rigid transformation $(\mathbf{R}_c, \mathbf{t}_c)$ and a canonical 2D point-cloud \mathbf{P}_t in the xy -plane such that:

$$\tilde{\mathbf{P}}_t = \begin{bmatrix} \mathbf{P}_t \\ \mathbf{1}^\top \end{bmatrix} = \begin{bmatrix} \mathbf{R}_c & \mathbf{t}_c \\ \mathbf{0}^\top & 1 \end{bmatrix} \begin{bmatrix} \mathbf{P}_t \\ \mathbf{0}^\top \\ \mathbf{1}^\top \end{bmatrix}, \quad \text{with } \mathbf{P}_t = \begin{bmatrix} \mathbf{u}_{tx}^\top \\ \mathbf{u}_{ty}^\top \end{bmatrix},$$

where $\mathbf{u}_{tx}^\top \mathbf{u}_{ty} = 0$, $\mathbf{u}_{tx}^\top \mathbf{1} = 0$, $\mathbf{u}_{ty}^\top \mathbf{1} = 0$. Note that \mathbf{u}_{tx} , \mathbf{u}_{ty} , and $\mathbf{1}$ form an orthogonal basis of the range space of $\tilde{\mathbf{P}}_t^\top$.

The solution of \mathbf{X} . Vector $\mathbf{1}$ lies in the range space of each $\tilde{\mathbf{P}}_t$. However, due to the constraint $\mathbf{X}^\top \mathbf{1} = \mathbf{0}$, we require the columns of \mathbf{X} to be orthogonal to $\mathbf{1}$. As a result, $\mathbf{1}$ must be excluded from \mathbf{X} . Hence, the columns of \mathbf{X} are essentially constructed based on the ‘‘closeness’’ to the range space of each \mathbf{P}_t , or equivalently to the range space of each $\tilde{\mathbf{P}}_t^\top$, by evaluating the cost (40). Note that since each $\tilde{\mathbf{P}}_t^\top$ has a two dimensional range space, the last column of \mathbf{X} , *i.e.*, \mathbf{x}_3 will be pushed toward the null space of \mathbf{P}_t by the orthogonality constraint $\mathbf{x}_1^\top \mathbf{x}_3 = 0$ and $\mathbf{x}_2^\top \mathbf{x}_3 = 0$.

The solution of $\sqrt{\Lambda}$. After solving \mathbf{X} , we leverage formulation (39) to estimate $\sqrt{\Lambda}$. In particular, we consider the following problem by using the canonical 2D point-clouds \mathbf{P}_t in the xy -plane, as:

$$\min_{\{\mathbf{R}_t \in \text{SO}(d)\}, \boldsymbol{\eta} \in \mathbb{R}^d} \sum_{t=1}^n \left\| \mathbf{R}_t \begin{bmatrix} \mathbf{P}_t \\ \mathbf{0}^\top \end{bmatrix} - \text{diag}(\boldsymbol{\eta}) \bar{\mathbf{G}}_t \right\|_{\mathcal{F}}^2. \quad (41)$$

If the optimal \mathbf{R}_t of problem (41) implements a rotation in the xy -plane, then the last component in $\sqrt{\Lambda}$ is zero, *i.e.*, $\lambda_3 = 0$, see Appendix L for more details. In this case, the optimal \mathbf{M} will be flat, residing in the xy -plane. This happens if GPA solved from the canonical 2D point-clouds \mathbf{P}_t with $t \in [1 : n]$ is optimal in the embedded 3D space.

Remark 10. In general, if the 2D data are generated by flattening 3D observations to 2D, *e.g.*, a) by a projective function or b) by simply ignoring the z -coordinates, the optimal reconstruction in the embedded 3D space is usually not flat! Such an example is the SfM problem.

Remark 11. Similar discussions hold for the case of $d = 2$, if the point-clouds degenerate to lines in the plane.

9 Implementation

9.1 Regularization Strength μ_t

We rewrite $\mathbf{K}_t \mathbf{S}_t^{-1}$ as:

$$\mathbf{K}_t \mathbf{S}_t^{-1} = \frac{1}{\mu_t} \left(\frac{1}{\mu_t} \mathbf{K}_t (\mathbf{I} - \mathcal{P}_t) + \mathbf{I} \right)^{-1}.$$

If $\mu_t \rightarrow +\infty$, then $\frac{1}{\mu_t} \mathbf{K}_t \rightarrow \mathbf{O}$. As a result, $\mathbf{K}_t \mathbf{S}_t^{-1} \rightarrow \mathbf{O}$ and thus $\mathbf{H}_t = (\mathbf{I} - \mathcal{P}_t) \mathbf{K}_t \mathbf{S}_t^{-1} \rightarrow \mathbf{O}$. From equations (25, 31, 32), we conclude when $\mu_t \rightarrow +\infty$, KernelGPA becomes the Affine-GPA:

$$\text{Affine GPA} \begin{cases} \mathbf{Q}_t = \mathbf{I} - \mathcal{P}_t \\ [\mathbf{A}_t, \mathbf{a}_t] = \mathbf{M} \boldsymbol{\Gamma}_t \tilde{\mathbf{P}}_t^\dagger \\ \boldsymbol{\Omega}_t^\top = \mathbf{O}. \end{cases}$$

For general cases, from equations (25, 31), we notice that both \mathbf{Q}_t and $[\mathbf{A}_t, \mathbf{a}_t]$ make use of the kernel matrix \mathbf{K}_t in the form of $\mathbf{K}_t \mathbf{S}_t^{-1} \mathbf{K}_t$:

$$\mathbf{K}_t \mathbf{S}_t^{-1} \mathbf{K}_t = \left((\mathbf{I} - \mathcal{P}_t) + \left(\frac{1}{\mu_t} \mathbf{K}_t \right)^{-1} \right)^{-1},$$

where μ_t controls the influence of \mathbf{K}_t as $\frac{1}{\mu_t} \mathbf{K}_t$, and thus the allowed deformation. The larger μ_t , the smaller the influence of \mathbf{K}_t , and thus the lower the allowed deformation.

We use the same regularization strength for all point-clouds, by setting $\mu_t = \mu$ for $t \in [1 : n]$.

9.2 Gaussian Kernel

The proposed KernelGPA can be implemented with a range of kernel functions, up to the choice of the user. Following Proposition 3 and Proposition 4, we suggest designing the kernel function $k(\cdot, \cdot)$ as the RBFs. Other than that, we do not pose any extra constraint on the possibilities of $k(\cdot, \cdot)$.

We specifically implement $k(\cdot, \cdot)$ using the Gaussian kernel, which is an RBF taking the form:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp \left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2} \right). \quad (42)$$

We decide the *kernel bandwidth* σ as $\sigma = p\bar{d}$, where \bar{d} denotes the mean of the pairwise Euclidean distances between all the discrete training points:

$$\bar{d} = \text{mean}(\|\mathbf{x}_i - \mathbf{x}_j\|), \quad \text{for all } (i \neq j), \quad (43)$$

and $p > 0$ is a tunable scale factor.

In our case, for each point-cloud \mathbf{P}_t and thus each \mathbf{K}_t , we implement a Gaussian kernel with kernel bandwidth σ_t . We set $\sigma_t = p\bar{d}_t$ where \bar{d}_t denotes the mean pairwise Euclidean distances between all the corresponding points in \mathbf{P}_t .

10 Experimental Results

We evaluate the performance of different GPA methods using three datasets: a) the semi-synthetic liver dataset for smooth organ deformations, b) the facial expression dataset for structural deformations, and c) the TOPACS point-clouds extracted from computerized tomography (CT) scans for real medical scenarios.

10.1 Preliminary

Correspondences. The proposed GPA registration is based on correspondences, which can be extracted from RGB-D cameras, segmented meshes or raw point-clouds. The computational complexity is determined by the dimension of the \mathcal{Q} matrix, and is thus decided by the number of used correspondences. Since we have assumed low-dimensional deformations, the GPA registration does not require a large number of correspondences. In contrast, in most cases, the redundancy of correspondences does not improve much the accuracy of the GPA registration, but cause strains on the computation. Hence, we always suggest using a reasonable amount of correspondences, as long as they are sufficient to capture the underlying motions and deformations.

Test points. After solving GPA, we obtain an estimate of the deformable transformations $\mathbf{y}_t(\cdot)$ and a reference map of used correspondences. While formulated in the cost function, it is not a good idea to evaluate the residual $\mathbf{y}_t(\mathbf{P}_t) - \mathbf{M}\mathbf{\Gamma}_t$, because $\mathbf{y}_t(\cdot)$ may overfit the correspondences. Therefore, we use correspondences to solve GPA, and afterwards benchmark the performance of GPA registration using the idea of *test points*. Importantly, the test points are never used to solve GPA (as the correspondences of these points are typically not available), but usually serve as a dense representation of the geometry of the scene.

Consistency by extrapolation. We denote the test points as ${}^\tau\mathbf{P}_t$ ($t \in [1 : n]$). After solving deformable transformations $\mathbf{y}_t(\cdot)$, we evaluate the coherence of the transformed points $\mathbf{y}_t({}^\tau\mathbf{P}_t)$ for all $t \in [1 : n]$. To benchmark the closeness of these transformed points, we need to define a distance metric, based on *e.g.*, surface-to-surface or nearest neighboring point distances etc. To simplify the evaluation, we assume the correspondence information for the test points are also known. We use ${}^\tau\mathbf{\Gamma}_t$ to denote the corresponding visibility information of ${}^\tau\mathbf{P}_t$. Such assumption allows us to evaluate the deviation of the transformed points $\mathbf{y}_t({}^\tau\mathbf{P}_t)$ directly.

Evaluation metrics. We define the *mean map* of the test points using the mean of $\mathbf{y}_t({}^\tau\mathbf{P}_t)$, as:

$${}^\tau\hat{\mathbf{M}} \stackrel{\text{def}}{=} \left(\sum_{t=1}^n \mathbf{y}_t({}^\tau\mathbf{P}_t) {}^\tau\mathbf{\Gamma}_t^\top \right) \left(\sum_{t=1}^n {}^\tau\mathbf{\Gamma}_t {}^\tau\mathbf{\Gamma}_t^\top \right)^\dagger, \quad (44)$$

where $\sum_{t=1}^n {}^\tau\mathbf{\Gamma}_t {}^\tau\mathbf{\Gamma}_t^\top$ count the total visibilities of each correspondence. We shall use the mean map ${}^\tau\hat{\mathbf{M}}$ as the reconstruction of the test points. Then we benchmark the accuracy of the mean map ${}^\tau\hat{\mathbf{M}}$ using the consistencies of the transformed test points. In specific, for each point in ${}^\tau\hat{\mathbf{M}}$, we define the *point-wise consistencies* of the test points as:

$${}^\tau\delta = \sqrt{\mathbf{1}^\top \left(\sum_{t=1}^n (\mathbf{\Sigma}_t * \mathbf{\Sigma}_t) {}^\tau\mathbf{\Gamma}_t^\top \right) \left(\sum_{t=1}^n {}^\tau\mathbf{\Gamma}_t {}^\tau\mathbf{\Gamma}_t^\top \right)^\dagger},$$

where $\mathbf{\Sigma}_t = \mathbf{y}_t({}^\tau\mathbf{P}_t) - {}^\tau\hat{\mathbf{M}}{}^\tau\mathbf{\Gamma}_t$, ($t \in [1 : n]$). Here $\mathbf{\Sigma}_t * \mathbf{\Sigma}_t$ denotes element-wise matrix multiplication, and the outermost square-root is also computed element-wise.

Benchmark methods. We term GPA with the TPS warp as TPS-GPA, and GPA with the KBT as Kernel-GPA. We compare Kernel-GPA with the Rigid-GPA, Affine-GPA and TPS-GPA methods. We use in total $125 = 5 \times 5 \times 5$ control points for the TPS warp, which are evenly distributed along the principle axes of the point-cloud. The regularization strength of the TPS warp is set to 0.01 as suggested in Bai and Bartoli (2022b) for 3D data.

10.2 Liver

Data generation. We use a segmented *liver mesh* model, as shown in Figure 5, which has 2002 vertices and 201 of them are selected as correspondences. We simulate deformations using the As-Rigid-As-Possible method Sorkine and Alexa (2007b) implemented in the CGAL* library. We simulate a circular trajectory comprising 60 poses, as shown in Figure 5. For the reason of clarity, only parts of the poses are plotted

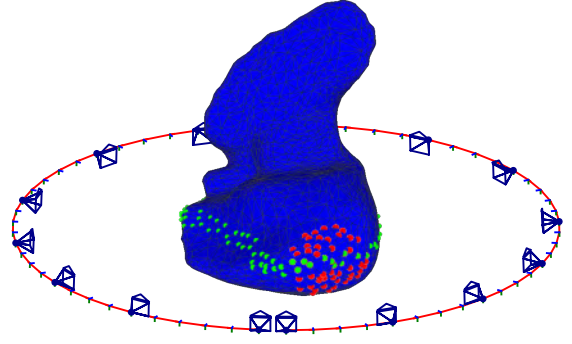


Figure 5. The simulated liver dataset. This dataset contains 60 meshes, deformed from a template liver mesh by the As-Rigid-As-Possible method Sorkine and Alexa (2007b). Each mesh has 2002 vertices with known correspondences. We observe these 60 meshes from different perspectives, by assigning 60 poses along a simulated circular trajectory, where only parts of the sensor poses are shown as blue pyramids. The green dots denote the control points used to generate deformations, and the red dots denote the disabled correspondences in Figure 8.

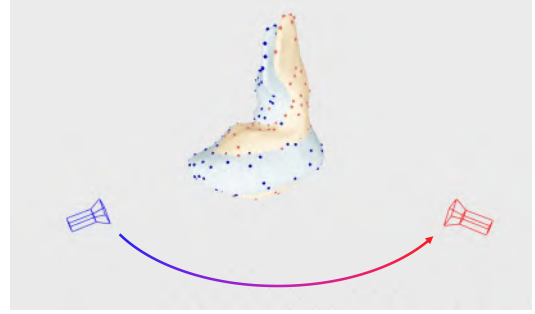


Figure 6. The simulated partial visibility of correspondences. The blue and red dots represent correspondences visible in the blue and red views, respectively. For a region without any correspondences, the deformation therein is never captured and thus is subject to information loss. Thus we drop correspondences randomly to simulate partial visibilities.

as the pyramid shapes. At each pose, the sensor observes a deformed mesh in its local coordinate frame, subject to partial visibilities and measurement noise:

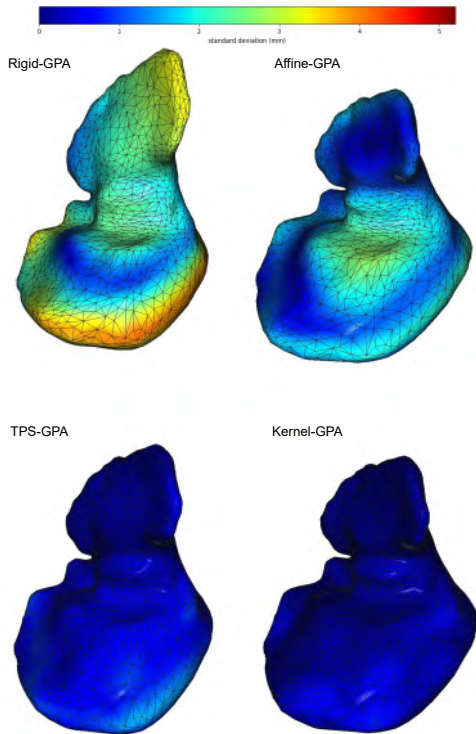
- *Partial visibility.* We randomly drop 30% of the 201 correspondences to simulate partial visibilities caused by correspondence detection failures, see Figure 6.
- *Measurement noise.* We add zero-mean Gaussian noise with its standard-deviation set to 1 mm, to simulate imperfect sensor measurements.

Evaluation. We set tuning parameters $p = 0.25$ and $\mu = 0.1$. We compute the GPA registration using the downsampled 201 correspondences, and then evaluate the performance of different GPA methods using all the 2002 correspondences. For each tested case, we report the minimum, maximum and mean of the point-wise registration error ${}^\tau\delta$ in Table 1. We visualize the mean map ${}^\tau\hat{\mathbf{M}}$, and the point-wise registration error ${}^\tau\delta$ in Figure 7, by using the case where the meshes

*<https://www.cgal.org>

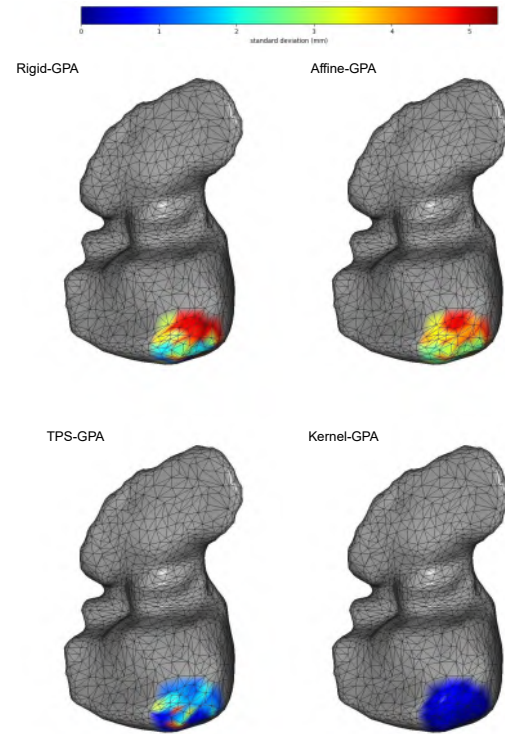
Table 1. The statistics of different GPA methods on the liver dataset.

			Rigid-GPA	Affine-GPA	TPS-GPA	Kernel-GPA
full visibility	no noise	min (mm)	0.389	0.140	0.022	0.006
		max (mm)	5.208	3.008	1.567	1.310
		mean (mm)	2.470	1.423	0.459	0.174
partial visibility	no noise	min (mm)	0.384	0.238	0.062	0.042
		max (mm)	5.207	3.031	1.518	1.868
		mean (mm)	2.480	1.435	0.503	0.453
full visibility	with noise	min (mm)	1.517	1.501	1.215	0.557
		max (mm)	5.691	3.506	2.404	2.448
		mean (mm)	3.062	2.229	1.713	1.749
partial visibility	with noise	min (mm)	1.531	1.499	1.328	1.127
		max (mm)	5.717	3.524	2.423	2.542
		mean (mm)	3.070	2.251	1.790	1.928

**Figure 7.** The accuracy of different GPA methods on the liver dataset. We visualize the mean map $\tau \hat{M}$ and encode point-wise consistencies of the test points $\tau \delta$ with color. Both TPS-GPA and Kernel-GPA give significantly better performance.

are fully-observable without noise. It can be seen that GPA with deformable transformations (*i.e.*, TPS-GPA and Kernel-GPA) can significantly outperform classical Rigid-GPA and Affine-GPA methods. The proposed Kernel-GPA method gives better results for regions with larger deformations.

We further set a small region of the liver to be invisible in all the 60 measurements, as seen in Figure 8, and use the mesh vertices therein as test points. In this test, we extrapolate the situation in the invisible region using $y_t(\cdot)$ computed from correspondences outside the invisible region. The predicted mean map $\tau \hat{M}$ and the point-wise consistencies of the transformed test points $\tau \delta$ are shown in Figure 8 for each GPA method. This result further backs our

**Figure 8.** Extrapolation on the liver dataset. We disable the correspondences in the colored region across all the 60 meshes. We solve GPA without the disabled correspondences, and then transform the test points in the region to construct a predicted mean surface. The reconstruction error of the predicted mean surface is given as the point-wise consistencies of the transformed test points, color coded.

claim on the superior performance of TPS-GPA and Kernel-GPA, where both methods can extrapolate the deformation in the invisible region with very similar performances.

Overall, for smooth deformations, we find both TPS-GPA and Kernel-GPA can give satisfactory results.

10.3 Facial Expression

Data generation. We create a *facial expression* dataset which contains the meshes of 5 facial expressions: 1) smiling, 2) curling the lip to the left, 3) curling the lip to the right, 4) cheek blowing and 5) opening the mouth, as shown



Figure 9. The facial expression dataset. We reconstruct the 3D model using DECA. The expressions from left to the right are respectively: 1) smile face, 2) curling the lip to the left, 3) curling the lip to the right, 4) cheek blowing, and 5) opening the mouth.

Table 2. The statistics of different GPA methods on the facial expression dataset.

		Rigid-GPA	Affine-GPA	TPS-GPA	Kernel-GPA
smiling	min (mm)	0.129	0.138	0.152	0.003
	max (mm)	9.901	10.536	10.518	2.797
	mean (mm)	1.552	1.564	1.553	0.374
curling left	min (mm)	0.099	0.138	0.152	0.002
	max (mm)	8.415	7.583	7.559	4.290
	mean (mm)	1.797	1.579	1.567	0.309
curling right	min (mm)	0.158	0.103	0.090	0.004
	max (mm)	8.265	8.283	8.251	3.613
	mean (mm)	1.730	1.652	1.640	0.400
cheek blowing	min (mm)	0.186	0.065	0.073	0.004
	max (mm)	16.941	13.580	13.507	7.209
	mean (mm)	3.445	2.984	2.955	0.527
opening mouth	min (mm)	0.240	0.156	0.145	0.003
	max (mm)	11.998	10.280	10.207	4.747
	mean (mm)	2.173	2.343	2.317	0.368

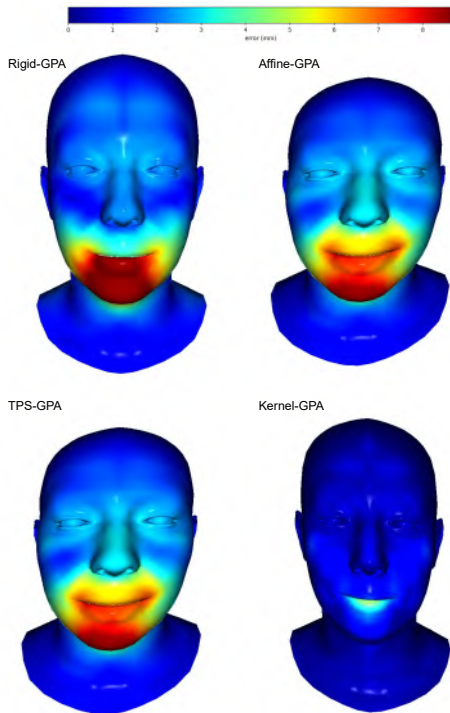


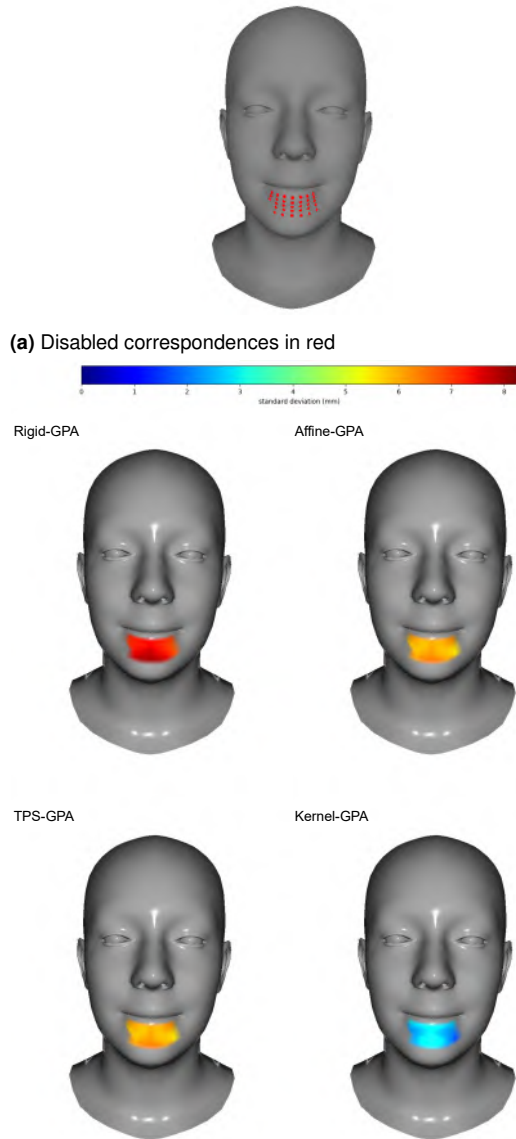
Figure 10. The face model $\tau \hat{\mathbf{M}}$ reconstructed from each GPA method, textured with the point-wise consistencies $\tau \delta$.

in Figure 9. The meshes of the head model are reconstructed with detailed facial geometry from a single input image using off-the-shelf toolbox DECA[†] from Feng, Feng, Black and Bolkart (2021). In the reconstructed meshes, the indices of the vertices are consistent thus the correspondences are available. There are 5118 vertices in total for each mesh, and we select 326 as correspondences for GPA registration and the rest for test.

Evaluation. We set tuning parameters $p = 0.25$ and $\mu = 0.2$. We use the selected 326 correspondences to solve GPA, and test the registration performance using all the 5118 points. We first show the reconstructed mean maps $\tau \hat{\mathbf{M}}$ for each GPA method in Figure 10, and encode the point-wise consistencies $\tau \delta$ with textures. We specifically examine the discrepancy between the transformed test points $\mathbf{y}_t(\tau \mathbf{P}_t)$ and the mean map $\tau \hat{\mathbf{M}}$, for each t individually. The statistics are reported in Table 2, and the visualization is given in Figure 14. For structural deformations, Kernel-GPA significantly outperforms the other methods, owing to its capability to handle *e.g.*, the 4-th *cheek blowing* point-cloud. Such data are challenging for TPS-GPA, as facial expressions are less smooth, with particularly large deformations on the cheek, around the nose and the mouth.

We examine the extrapolation ability of $\mathbf{y}_t(\cdot)$ around the chin area, as shown in Figure 11a, by disabling the

[†]<https://deca.is.tue.mpg.de>



(b) Extrapolation error of each GPA method

Figure 11. Extrapolation on the facial expression dataset. We solve GPA without the correspondences around the chin, and then transform the test points in the chin area to construct a predicted mean surface to fill the hole. The point-wise consistencies of the transformed test points are color coded.

correspondences in the selected region. We solve GPA without the disabled correspondences, and use the computed $y_t(\cdot)$ to extrapolate the deformation. Within the region, the predicted mean map ${}^T\hat{M}$ and the point-wise consistencies of the transformed test points ${}^T\delta$ are visualized in Figure 11b. It can be seen that the Kernel-GPA gives significantly better prediction compared with the other three GPA methods, confirming the superior modeling power of the KBT.

Overall, for structural deformations, we find the proposed Kernel-GPA method outperforms the TPS-GPA, the Affine-GPA and the Rigid-GPA methods.

10.4 CT Point-cloud

Data generation. We provide a dataset, termed TOPACS, for computerized tomography (CT) registration. The CT data

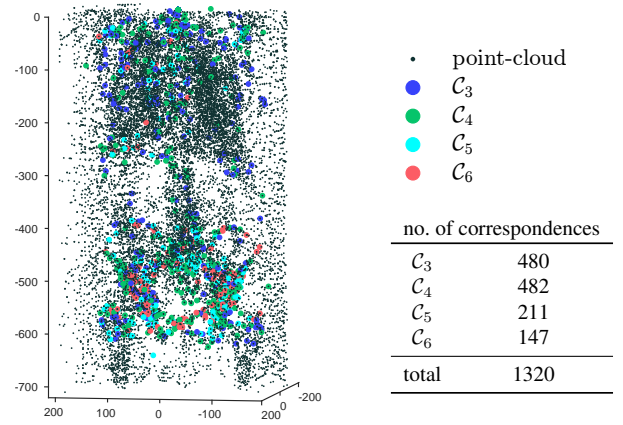


Figure 12. The TOPACS point-cloud dataset. The point-clouds are processed from real CT scans using SURF3D features. This dataset contains 6 point-clouds, with 20000 points for each point-cloud. There are in total 1320 correspondences classified into four categories according to their occurrences.

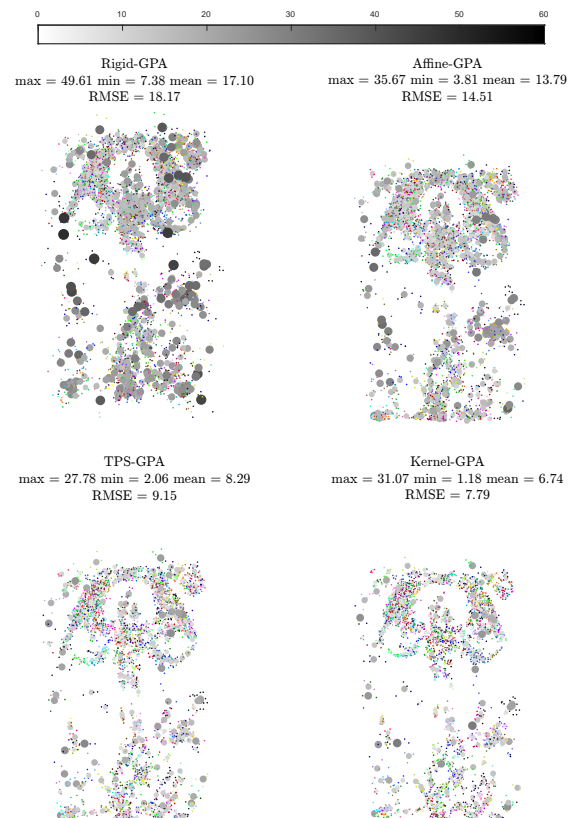


Figure 13. The registration of different GPA methods on the TOPACS dataset, with exactly the same tuning parameters used in Table 3. In this example, we use C_3 to solve GPA, and then show the final registration result by visualizing the point-wise consistencies of C_4, C_5, C_6 , as both size-coded and color-coded with the gray filled circles. The smaller the marker size and the brighter the color, the better. The transformed correspondences are also plotted as colored dots, where the corresponding points are plotted with the same color.

we use, shown in Figure 12, contain 6 scans of lungs, which are processed by the SURF3D features [Raju and Newman \(1993\)](#) resulting in 6 point-clouds (with 20000 points for each point-cloud). Initial correspondences are found by matching feature descriptors and then refined by an ICP

algorithm. The global correspondences are found by a graph matching algorithm, and the ambiguous ones are removed based on distances. We categorize the correspondences into four sets $\mathcal{C}_3, \mathcal{C}_4, \mathcal{C}_5, \mathcal{C}_6$ by their visibilities. For example, \mathcal{C}_3 collects the correspondences visible in exactly three point-clouds, and others are defined analogously.

Evaluation. We set tuning parameters $p = 0.20$ and $\mu = 0.05$. We use one category of the correspondences \mathcal{C}_k to solve GPA, which gives an estimate of deformable transformations $\mathbf{y}_t(\cdot)$. Then we use the other categories \mathcal{C}_j ($j \neq k$) as the test points. We compute the point-wise consistencies of the transformed test points ${}^\tau\delta$ (thanks to the known correspondences), and report the statistics in Table 3. For this dataset, we see a remarkable reduction of the registration error from the Rigid-GPA to the Affine-GPA. This is probably because of the fact that each point-cloud is for a different subject, and the subject’s morphology varies a lot in width, length and thickness. Another explanation is that a large portion of deformations are caused by the stretching of shoulders in the preparation process of the CT scanning. Such shearing is well-modeled by the affine transformation. We further see that both the TPS-GPA and the Kernel-GPA methods can further improve the performance of the affine-GPA, owing to their capabilities to model nonlinear deformations caused by breathing.

We provide a visualization in Figure 13, by using the correspondences \mathcal{C}_3 to solve GPA and $\mathcal{C}_4, \mathcal{C}_5, \mathcal{C}_6$ for validation, as this is the worst case in Table 3. We see that both the TPS-GPA and the Kernel-GPA methods give similar performances, while there are marginal differences in the statistics. This can be understood as the underlying deformations are smooth, similar to the liver data studied in Section 10.2. This further backs the claim that both TPS-GPA and Kernel-GPA are suitable for surgical applications, while Kernel-GPA is preferred in case of more complex deformations, *e.g.*, the facial expression data studied in Section 10.3.

In contrast to the simulated liver in Section 10.2 with perfect correspondences, the correspondences from CT scans (*i.e.*, slices of gray images) are never perfect due to the lack of textures and are thus subject to mismatches (*i.e.*, outliers). However, we show that the correspondence based method works well, even for such challenging CT data. In practice, we expect better performance of GPA methods, if the correspondences are extracted from RGB images.

11 Discussion and Conclusion

We have proposed KernelGPA, a novel GPA method using the KBT as the deformable transformation. We have given detailed mathematical derivations to show the point that KernelGPA can be solved globally in closed-form up to some global scale ambiguities. We have proposed to estimate the global scale ambiguities by an optimization formulation that favors rigidity, which has also allowed us to give insights on the degenerate cases. While we have implemented KernelGPA using the Gaussian kernel, the proposed KernelGPA can be implemented using any positive definite kernel, *e.g.*, the Laplacian kernel. We have validated the performance of KernelGPA with both simulated and real

datasets. Our Matlab code and data are publicly available for future comparison.

Computational complexity. Computationally, the complexity of KernelGPA is characterized by the number of correspondences used, and largely independent of the number of poses. In specific, the most expensive part of KernelGPA comprises: 1) the construction of matrix \mathbf{Q}_t in equation (25) which requires the inversion of matrix \mathbf{S}_t , and 2) the Eigen decomposition of matrix \mathbf{Q} to solve formulation (24). The dimension of both \mathbf{S}_t and \mathbf{Q} are decided by the number of correspondences used. In practice, for example in medical applications, the number of correspondences are typically limited to a few hundred, which can be handled effectively. For certain cases, if a large number of correspondences are available, we suggest selecting a reasonable number of robust correspondences that cover the deformable part of the scene.

Connection to the Tomasi-Kanade factorization. For the affine case, the affine transformation and the canonical map can be jointly factorized by the singular value decomposition (SVD), see Section 3.2 in Bai and Bartoli (2022b) and the *AFF_d method. This SVD approach is in the same spirit of the Tomasi-Kanade factorization Tomasi and Kanade (1992) in computer vision based on the orthographic camera model. The SVD approach has been extended to handle structural deformations, see Bregler, Hertzmann and Biermann (2000). In this work, we have proposed an alternative factorization method based on the Eigen decomposition. As for the cost function, the residual of the SVD approach is defined in the coordinate frame of the sensor, while the residual of our Eigen approach is defined in the coordinate frame of the canonical map. Critically, we show in Section 5 that the geometry of the canonical map \mathbf{M} can be defined up to d global scale ambiguities. This point is not realized in the classical SVD approach, which thus does not use the constraints in Section 5 to further reduce the ambiguities. As a result, the SVD approach gives a solution up to a global affine transformation, while our Eigen approach gives a solution up to d global scale ambiguities. Recall that the affine transformation has d^2 parameters (not considering the translation), hence more than the d of our method.

Future work. The future work includes handling pose ambiguities (for example, by adding additional sensor information or deformation assumptions), incorporating probabilistic models to handle non-isotropic noise, extending the proposed GPA method to SfM problems, exploring different kernel functions, and exploiting the sparsity of the kernel matrix for even faster computation.

A Brockett Cost Function on the Stiefel Manifold

Definition 12. (Brockett cost function on the Stiefel manifold). The matrix Stiefel manifold is the set of matrices satisfying:

$$\text{St}(d, m) = \{ \mathbf{X} \in \mathbb{R}^{m \times d} \mid \mathbf{X}^\top \mathbf{X} = \mathbf{I} \}.$$

The following function defined on the Stiefel manifold is termed the Brockett cost function Brockett (1989); Absil,

Table 3. The statistics of different GPA methods on the TOPACS dataset.

registration	test		Rigid-GPA	Affine-GPA	TPS-GPA	Kernel-GPA
C_3	$C_4C_5C_6$	min (mm)	7.38	3.81	2.06	1.18
		max (mm)	49.61	35.67	27.78	31.07
		mean (mm)	17.10	13.79	8.29	6.74
C_4	$C_3C_5C_6$	min (mm)	3.26	2.86	1.19	0.55
		max (mm)	59.41	42.36	36.79	34.68
		mean (mm)	17.30	13.29	8.15	6.62
C_5	$C_3C_4C_6$	min (mm)	2.76	2.57	1.56	0.59
		max (mm)	61.20	45.78	37.32	35.69
		mean (mm)	17.13	13.22	8.57	7.64
C_6	$C_3C_4C_5$	min (mm)	2.21	1.63	1.05	0.88
		max (mm)	63.71	56.54	47.97	46.82
		mean (mm)	17.66	14.12	10.04	9.94

Mahony and Sepulchre (2009):

$$f_{\mathbf{X} \in \text{St}(d,m)}(\mathbf{X}) = \text{tr}(\mathbf{X}^\top \mathbf{\Pi} \mathbf{X} \mathbf{\Lambda}), \quad (45)$$

where $\mathbf{\Pi} \in \mathbb{R}^{m \times m}$ is symmetric, and $\mathbf{\Lambda} = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_d)$ with $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_d \geq 0$.

Lemma 6. (Brockett (1989); Birtea, Caşu and Comănescu (2019); Absil, Mahony and Sepulchre (2009)). The critical points of the Brockett cost function $f_{\mathbf{X} \in \text{St}(d,m)}(\mathbf{X})$ on the Stiefel manifold are the eigenvectors of $\mathbf{\Pi}$.

If we choose $\mathbf{X}' = [\xi_1, \xi_2, \dots, \xi_d]$ with $\mathbf{\Pi} \xi_k = \alpha_k \xi_k$ (i.e., ξ_k is an eigenvector of $\mathbf{\Pi}$ corresponding to eigenvalue α_k), then we have cost $f_{\mathbf{X}' \in \text{St}(d,m)}(\mathbf{X}') = \lambda_1 \xi_1 + \lambda_2 \xi_2 + \dots + \lambda_d \xi_d$.

Lemma 7. (Hardy-Littlewood-Polya Hardy, Collection, Littlewood, Pólya, Pólya and Littlewood (1952)). For two sequences of real numbers $x_1 \geq x_2 \geq \dots \geq x_n$ and y_1, y_2, \dots, y_n in any order, we consider:

$$\eta = \sum_i^n x_i y_{\pi(i)},$$

where π denotes a permutation of indices in $[1 : n]$. The maximum of η is attained when $y_{\pi(1)} \geq y_{\pi(2)} \geq \dots \geq y_{\pi(n)}$. The minimum of η is attained when $y_{\pi(1)} \leq y_{\pi(2)} \leq \dots \leq y_{\pi(n)}$.

Lemma 8. (Brockett (1989)). For a symmetric $\mathbf{\Pi} \in \mathbb{R}^{m \times m}$, we denote its eigenvalue decomposition as:

$$\mathbf{\Pi} = \mathbf{U} \mathbf{\Sigma} \mathbf{U}^\top = \sum_{k=1}^m \sigma_k \mathbf{u}_k \mathbf{u}_k^\top,$$

where $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_m]$ is an orthonormal matrix, and $\mathbf{\Sigma} = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_m)$ with $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_m$. Let $\mathbf{\Lambda} = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_d)$ with $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_d \geq 0$. We have:

1. $\max_{\mathbf{X} \in \text{St}(d,m)} \text{tr}(\mathbf{X}^\top \mathbf{\Pi} \mathbf{X} \mathbf{\Lambda})$ is attained at:

$$\mathbf{X} = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_d],$$

which comprises the d top eigenvectors of $\mathbf{\Pi}$, with cost $\lambda_1 \sigma_1 + \lambda_2 \sigma_2 + \dots + \lambda_d \sigma_d$.

2. $\min_{\mathbf{X} \in \text{St}(d,m)} \text{tr}(\mathbf{X}^\top \mathbf{\Pi} \mathbf{X} \mathbf{\Lambda})$ is attained at:

$$\mathbf{X} = [\mathbf{u}_m, \mathbf{u}_{m-1}, \dots, \mathbf{u}_{m-d+1}],$$

which comprises the d bottom eigenvectors of $\mathbf{\Pi}$, with cost $\lambda_1 \sigma_m + \lambda_2 \sigma_{m-1} + \dots + \lambda_d \sigma_{m-d+1}$.

B Proof of Lemma 4

We first notice that when $\mathbf{X}^\top \mathbf{u} = \mathbf{0}$, the cost is equivalent to:

$$\text{tr}(\mathbf{X}^\top \mathbf{\Pi} \mathbf{X} \mathbf{\Lambda}) = \text{tr}(\mathbf{X}^\top (\mathbf{\Pi} - c \mathbf{u} \mathbf{u}^\top) \mathbf{X} \mathbf{\Lambda}),$$

where c is an arbitrary scalar. Importantly, by using different c , we can shift \mathbf{u} to the top or bottom eigenvector of $\mathbf{\Pi} - c \mathbf{u} \mathbf{u}^\top$. We denote the eigenvalue decomposition of $\mathbf{\Pi}$ as:

$$\mathbf{\Pi} = \sum_{k=1}^m \sigma_k \mathbf{u}_k \mathbf{u}_k^\top,$$

with $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_m$.

Case 1. We consider $c > \sigma_1 - \sigma_m$, and the following relaxation of problem (27) without constraint $\mathbf{X}^\top \mathbf{u} = \mathbf{0}$:

$$\begin{aligned} \max_{\mathbf{X}} \quad & \text{tr}(\mathbf{X}^\top (\mathbf{\Pi} - c \mathbf{u} \mathbf{u}^\top) \mathbf{X} \mathbf{\Lambda}) \\ \text{s.t.} \quad & \mathbf{X}^\top \mathbf{X} = \mathbf{I}. \end{aligned} \quad (46)$$

If \mathbf{X}_* is optimal to problem (46) and satisfies $\mathbf{X}_*^\top \mathbf{u} = \mathbf{0}$, then \mathbf{X}_* is optimal to problem (27).

Problem (46) admits a Brockett cost on the Stiefel manifold (see Appendix A), where we denote its solution by \mathbf{X}_* . The columns of \mathbf{X}_* comprise the d top eigenvectors of $\mathbf{\Pi} - c \mathbf{u} \mathbf{u}^\top$. If $c > \sigma_1 - \sigma_m$, \mathbf{u} becomes the bottom eigenvector of $\mathbf{\Pi} - c \mathbf{u} \mathbf{u}^\top$. Thus $\mathbf{X}_*^\top \mathbf{u} = \mathbf{0}$ because eigenvectors with respect to different eigenvalues are orthogonal. To conclude, if $c > \sigma_1 - \sigma_m$, problem (46) is a tight relaxation to problem (27).

Lastly, when $c > \sigma_1 - \sigma_m$, the d top eigenvectors of $\mathbf{\Pi} - c \mathbf{u} \mathbf{u}^\top$ are the d top eigenvectors of $\mathbf{\Pi}$ excluding \mathbf{u} .

Case 2. We consider $c > \sigma_1 - \sigma_m$, and the following relaxation of problem (28) without constraint $\mathbf{X}^\top \mathbf{u} = \mathbf{0}$:

$$\begin{aligned} \min_{\mathbf{X}} \quad & \text{tr}(\mathbf{X}^\top (\mathbf{\Pi} + c \mathbf{u} \mathbf{u}^\top) \mathbf{X} \mathbf{\Lambda}) \\ \text{s.t.} \quad & \mathbf{X}^\top \mathbf{X} = \mathbf{I}, \end{aligned} \quad (47)$$

which is a tight relaxation to problem (28) if $c > \sigma_1 - \sigma_m$.

C Derivation of the Reduced Problem

C.1 Linear Dependence of $\mathbf{A}_t, \mathbf{a}_t, \mathbf{\Omega}_t$ on \mathbf{M}

By defining $\tilde{\mathbf{P}}_t = [\mathbf{P}_t^\top, \mathbf{1}^\top]^\top$, we notice that the affine part can be rewritten as:

$$\mathbf{A}_t \mathbf{P}_t + \mathbf{a}_t \mathbf{1}^\top = [\mathbf{A}_t, \mathbf{a}_t] \begin{bmatrix} \mathbf{P}_t \\ \mathbf{1}^\top \end{bmatrix} = [\mathbf{A}_t, \mathbf{a}_t] \tilde{\mathbf{P}}_t,$$

Then we write cost $\varphi_t(\mathbf{A}_t, \mathbf{a}_t, \mathbf{\Omega}_t, \mathbf{M})$ in matrix form:

$$\begin{aligned} \varphi_t(\mathbf{A}_t, \mathbf{a}_t, \mathbf{\Omega}_t, \mathbf{M}) &= \left\| \mathbf{A}_t \mathbf{P}_t + \mathbf{a}_t \mathbf{1}^\top + \mathbf{\Omega}_t^\top \mathbf{K}_t - \mathbf{M} \mathbf{\Gamma}_t \right\|_{\mathcal{F}}^2 \\ &+ \mu_t \text{tr} \left(\mathbf{\Omega}_t^\top \mathbf{K}_t \mathbf{\Omega}_t \right) \\ &= \left\| \begin{bmatrix} \mathbf{A}_t, \mathbf{a}_t, \mathbf{\Omega}_t^\top \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{P}}_t & \mathbf{O} \\ \mathbf{K}_t & \sqrt{\mu_t \mathbf{K}_t} \end{bmatrix} - \begin{bmatrix} \mathbf{M} \mathbf{\Gamma}_t & \mathbf{O} \end{bmatrix} \right\|_{\mathcal{F}}^2 \\ &= \left\| \begin{bmatrix} \mathbf{A}_t, \mathbf{a}_t, \mathbf{\Omega}_t^\top \end{bmatrix} \mathbf{C}_t - \begin{bmatrix} \mathbf{M} \mathbf{\Gamma}_t & \mathbf{O} \end{bmatrix} \right\|_{\mathcal{F}}^2, \end{aligned}$$

where we have defined the matrix constant \mathbf{C}_t as:

$$\mathbf{C}_t = \begin{bmatrix} \tilde{\mathbf{P}}_t & \mathbf{O} \\ \mathbf{K}_t & \sqrt{\mu_t \mathbf{K}_t} \end{bmatrix}.$$

Given \mathbf{M} , the problem regarding $\mathbf{A}_t, \mathbf{a}_t, \mathbf{\Omega}_t$:

$$\min_{\mathbf{A}_t, \mathbf{a}_t, \mathbf{\Omega}_t} \varphi_t(\mathbf{A}_t, \mathbf{a}_t, \mathbf{\Omega}_t, \mathbf{M}), \quad \text{given } \mathbf{M},$$

is a LLS optimization problem. The optimal solution is in closed-form:

$$\begin{aligned} \begin{bmatrix} \mathbf{A}_t, \mathbf{a}_t, \mathbf{\Omega}_t^\top \end{bmatrix} &= \begin{bmatrix} \mathbf{M} \mathbf{\Gamma}_t & \mathbf{O} \end{bmatrix} \mathbf{C}_t^\dagger \\ &+ \mathbf{F}_t \left(\mathbf{I} - \mathbf{C}_t \mathbf{C}_t^\dagger \right), \quad (48) \end{aligned}$$

where \mathbf{C}_t^\dagger is the Moore–Penrose pseudo-inverse of \mathbf{C}_t , and $\mathbf{F}_t \in \mathbb{R}^{d \times (m_t + d + 1)}$ is a free matrix (*i.e.*, an arbitrary matrix with the compatible dimension). We denote:

$$\mathbf{\Delta}_t \stackrel{\text{def}}{=} \mathbf{C}_t \mathbf{C}_t^\top = \begin{bmatrix} \tilde{\mathbf{P}}_t \tilde{\mathbf{P}}_t^\top & \tilde{\mathbf{P}}_t \mathbf{K}_t \\ \mathbf{K}_t \tilde{\mathbf{P}}_t^\top & \mathbf{K}_t \mathbf{K}_t + \mu_t \mathbf{K}_t \end{bmatrix},$$

and expand the Moore–Penrose pseudo-inverse \mathbf{C}_t^\dagger as:

$$\mathbf{C}_t^\dagger = \mathbf{C}_t^\top \left(\mathbf{C}_t \mathbf{C}_t^\top \right)^\dagger = \mathbf{C}_t^\top \mathbf{\Delta}_t^\dagger.$$

Lastly, we express equation (48) using $\mathbf{\Delta}_t$ as:

$$\begin{aligned} \begin{bmatrix} \mathbf{A}_t, \mathbf{a}_t, \mathbf{\Omega}_t^\top \end{bmatrix} &= \begin{bmatrix} \mathbf{M} \mathbf{\Gamma}_t & \mathbf{O} \end{bmatrix} \mathbf{C}_t^\top \mathbf{\Delta}_t^\dagger + \mathbf{F}_t \left(\mathbf{I} - \mathbf{C}_t \mathbf{C}_t^\top \mathbf{\Delta}_t^\dagger \right) \\ &= \mathbf{M} \mathbf{\Gamma}_t \begin{bmatrix} \tilde{\mathbf{P}}_t^\top & \mathbf{K}_t \end{bmatrix} \mathbf{\Delta}_t^\dagger + \mathbf{F}_t \left(\mathbf{I} - \mathbf{\Delta}_t \mathbf{\Delta}_t^\dagger \right), \end{aligned}$$

which is the form in equation (23).

C.2 Cost $\varphi_t(\mathbf{M})$

We first notice that $\mathbf{Y} \left(\mathbf{I} - \mathbf{C}_t \mathbf{C}_t^\dagger \right) \mathbf{C}_t = \mathbf{O}$ because the Moore–Penrose pseudo-inverse satisfies $\mathbf{C}_t = \mathbf{C}_t \mathbf{C}_t^\dagger \mathbf{C}_t$. Substituting equation (48) into the cost $\varphi_t(\mathbf{A}_t, \mathbf{a}_t, \mathbf{\Omega}_t, \mathbf{M})$, we obtain the reduced cost $\varphi_t(\mathbf{M})$:

$$\begin{aligned} \varphi_t(\mathbf{M}) &= \left\| \begin{bmatrix} \mathbf{M} \mathbf{\Gamma}_t & \mathbf{O} \end{bmatrix} \left(\mathbf{C}_t^\dagger \mathbf{C}_t - \mathbf{I} \right) \right\|_{\mathcal{F}}^2 \\ &= \text{tr} \left(\begin{bmatrix} \mathbf{M} \mathbf{\Gamma}_t & \mathbf{O} \end{bmatrix} \left(\mathbf{I} - \mathbf{C}_t^\dagger \mathbf{C}_t \right) \begin{bmatrix} \mathbf{M} \mathbf{\Gamma}_t & \mathbf{O} \end{bmatrix}^\top \right), \end{aligned}$$

where we have used the fact that $\mathbf{I} - \mathbf{C}_t^\dagger \mathbf{C}_t$ is symmetric and idempotent, since it is an orthogonal projection matrix

(*i.e.*, the orthogonal projector to the null space of \mathbf{C}_t). In particular, we can write $\mathbf{C}_t^\dagger \mathbf{C}_t$ as:

$$\mathbf{C}_t^\dagger \mathbf{C}_t = \mathbf{C}_t^\top \left(\mathbf{C}_t \mathbf{C}_t^\top \right)^\dagger \mathbf{C}_t = \mathbf{C}_t^\top \mathbf{\Delta}_t^\dagger \mathbf{C}_t.$$

The matrix multiplication shows:

$$\begin{bmatrix} \mathbf{M} \mathbf{\Gamma}_t & \mathbf{O} \end{bmatrix} \mathbf{C}_t^\top = \mathbf{M} \mathbf{\Gamma}_t \begin{bmatrix} \tilde{\mathbf{P}}_t^\top & \mathbf{K}_t \end{bmatrix}.$$

Lastly, we write $\varphi_t(\mathbf{M})$ as:

$$\varphi_t(\mathbf{M}) = \text{tr} \left(\mathbf{M} \mathbf{\Gamma}_t \left(\mathbf{I} - \begin{bmatrix} \tilde{\mathbf{P}}_t^\top & \mathbf{K}_t \end{bmatrix} \mathbf{\Delta}_t^\dagger \begin{bmatrix} \tilde{\mathbf{P}}_t \\ \mathbf{K}_t \end{bmatrix} \right) \mathbf{\Gamma}_t^\top \mathbf{M}^\top \right).$$

D Positive Definiteness of $\mathbf{\Delta}_t$ and $\tilde{\mathbf{P}}_t \tilde{\mathbf{P}}_t^\top$

D.1 Preliminary

Lemma 9. (Gallier (2010)). For any symmetric matrix \mathbf{S} of the form:

$$\mathbf{S} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^\top & \mathbf{C} \end{bmatrix},$$

if \mathbf{C} is positive definite, then \mathbf{S} is positive definite if and only if $\mathbf{A} - \mathbf{B} \mathbf{C}^{-1} \mathbf{B}^\top$ is positive definite.

Lemma 10. (Woodbury matrix identity). For any invertible \mathbf{A} and \mathbf{D} , the following identity holds:

$$\begin{aligned} (\mathbf{A} + \mathbf{B} \mathbf{D} \mathbf{C})^{-1} &= \mathbf{A}^{-1} \\ &- \mathbf{A}^{-1} \mathbf{B} \left(\mathbf{D}^{-1} + \mathbf{C} \mathbf{A}^{-1} \mathbf{B} \right)^{-1} \mathbf{C} \mathbf{A}^{-1}. \end{aligned}$$

D.2 Proof of Lemma 3

We have denoted $\mathbf{\Delta}_t$ as:

$$\mathbf{\Delta}_t = \begin{bmatrix} \tilde{\mathbf{P}}_t \tilde{\mathbf{P}}_t^\top & \tilde{\mathbf{P}}_t \mathbf{K}_t \\ \mathbf{K}_t \tilde{\mathbf{P}}_t^\top & \mathbf{K}_t \mathbf{K}_t + \mu_t \mathbf{K}_t \end{bmatrix}.$$

If \mathbf{K}_t is chosen positive definite and $\mu_t > 0$, it suffices to examine the positive definiteness of:

$$\begin{aligned} \mathbf{E}_t &= \tilde{\mathbf{P}}_t \tilde{\mathbf{P}}_t^\top - \tilde{\mathbf{P}}_t \mathbf{K}_t \left(\mathbf{K}_t \mathbf{K}_t + \mu_t \mathbf{K}_t \right)^{-1} \mathbf{K}_t \tilde{\mathbf{P}}_t^\top \\ &= \tilde{\mathbf{P}}_t \left(\mathbf{I} - \left(\mathbf{I} + \mu_t \mathbf{K}_t^{-1} \right)^{-1} \right) \tilde{\mathbf{P}}_t^\top \\ &= \tilde{\mathbf{P}}_t \left(\mathbf{I} + \frac{1}{\mu_t} \mathbf{K}_t \right)^{-1} \tilde{\mathbf{P}}_t^\top, \end{aligned}$$

where the last equality holds because of the Woodbury matrix identity.

We notice that $\mathbf{I} + \frac{1}{\mu_t} \mathbf{K}_t$ is positive definite. As a result, \mathbf{E}_t is positive definite if and only if $\tilde{\mathbf{P}}_t \tilde{\mathbf{P}}_t^\top$ is positive definite.

E Expansion of $\mathbf{\Delta}_t^\dagger$

We expand $\mathbf{\Delta}_t^\dagger$ with respect to $\tilde{\mathbf{P}}_t \tilde{\mathbf{P}}_t^\top$, by the Schur complement for positive semi-definite matrices.

To this end, we denote the Moore–Penrose pseudo-inverse of $\tilde{\mathbf{P}}_t \tilde{\mathbf{P}}_t^\top$ as $(\tilde{\mathbf{P}}_t \tilde{\mathbf{P}}_t^\top)^\dagger$, and define:

$$\mathbf{S}_t \stackrel{\text{def}}{=} \mathbf{K}_t \mathbf{K}_t + \mu_t \mathbf{K}_t - \mathbf{K}_t \tilde{\mathbf{P}}_t^\top \left(\tilde{\mathbf{P}}_t \tilde{\mathbf{P}}_t^\top \right)^\dagger \tilde{\mathbf{P}}_t \mathbf{K}_t. \quad (49)$$

In equation (49), using the notation $\mathcal{P}_t = \tilde{\mathbf{P}}_t^\top (\tilde{\mathbf{P}}_t \tilde{\mathbf{P}}_t^\top)^\dagger \tilde{\mathbf{P}}_t$, we can rewrite \mathbf{S}_t in form of equation (26), as:

$$\mathbf{S}_t = \mathbf{K}_t (\mathbf{I} - \mathcal{P}_t) \mathbf{K}_t + \mu_t \mathbf{K}_t.$$

As an orthogonal projector, matrix $\mathbf{I} - \mathcal{P}_t$ is always positive semi-definite. Therefore if \mathbf{K}_t is positive definite and $\mu_t > 0$, then \mathbf{S}_t is positive definite.

By the Schur complement Gallier (2010), the expansion of Δ_t^\dagger with respect to $\tilde{\mathbf{P}}_t \tilde{\mathbf{P}}_t^\top$ is:

$$\Delta_t^\dagger = \begin{bmatrix} \tilde{\mathbf{P}}_t \tilde{\mathbf{P}}_t^\top & \tilde{\mathbf{P}}_t \mathbf{K}_t \\ \mathbf{K}_t \tilde{\mathbf{P}}_t^\top & \mathbf{K}_t \mathbf{K}_t + \mu_t \mathbf{K}_t \end{bmatrix}^\dagger = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix},$$

where the relevant blocks are defined as:

$$\begin{cases} \Sigma_{11} \stackrel{\text{def}}{=} (\tilde{\mathbf{P}}_t \tilde{\mathbf{P}}_t^\top)^\dagger + (\tilde{\mathbf{P}}_t \tilde{\mathbf{P}}_t^\top)^\dagger \tilde{\mathbf{P}}_t \mathbf{K}_t \mathbf{S}_t^{-1} \mathbf{K}_t \tilde{\mathbf{P}}_t^\top (\tilde{\mathbf{P}}_t \tilde{\mathbf{P}}_t^\top)^\dagger \\ \Sigma_{12} \stackrel{\text{def}}{=} -(\tilde{\mathbf{P}}_t \tilde{\mathbf{P}}_t^\top)^\dagger \tilde{\mathbf{P}}_t \mathbf{K}_t \mathbf{S}_t^{-1} \\ \Sigma_{21} \stackrel{\text{def}}{=} -\mathbf{S}_t^{-1} \mathbf{K}_t \tilde{\mathbf{P}}_t^\top (\tilde{\mathbf{P}}_t \tilde{\mathbf{P}}_t^\top)^\dagger \\ \Sigma_{22} \stackrel{\text{def}}{=} \mathbf{S}_t^{-1}. \end{cases}$$

F Derivation of \mathbf{Q}_t

Following Appendix E, by denoting $\tilde{\mathbf{P}}_t^\dagger = \tilde{\mathbf{P}}_t^\top (\tilde{\mathbf{P}}_t \tilde{\mathbf{P}}_t^\top)^\dagger$ which is the Moore–Penrose pseudo-inverse of $\tilde{\mathbf{P}}_t$, we further compute:

$$\begin{bmatrix} \tilde{\mathbf{P}}_t^\top & \mathbf{K}_t \end{bmatrix} \Delta_t^\dagger = \begin{bmatrix} \tilde{\mathbf{P}}_t^\dagger - \mathbf{H}_t \mathbf{K}_t \tilde{\mathbf{P}}_t^\dagger & \mathbf{H}_t \end{bmatrix}, \quad (50)$$

where:

$$\mathbf{H}_t \stackrel{\text{def}}{=} (\mathbf{I} - \mathcal{P}_t) \mathbf{K}_t \mathbf{S}_t^{-1}. \quad (51)$$

Matrix \mathbf{H}_t thus defined is symmetric, *i.e.*, $\mathbf{H}_t = \mathbf{H}_t^\top$, as to be shown in Proposition 2 and Appendix H. From equation (50), we obtain:

$$\begin{aligned} \mathbf{Q}_t &= \mathbf{I} - \begin{bmatrix} \tilde{\mathbf{P}}_t^\top & \mathbf{K}_t \end{bmatrix} \Delta_t^\dagger \begin{bmatrix} \tilde{\mathbf{P}}_t \\ \mathbf{K}_t \end{bmatrix} \\ &= \mathbf{I} - \left(\tilde{\mathbf{P}}_t^\dagger \tilde{\mathbf{P}}_t - \mathbf{H}_t \mathbf{K}_t \tilde{\mathbf{P}}_t^\dagger \tilde{\mathbf{P}}_t + \mathbf{H}_t \mathbf{K}_t \right) \\ &= (\mathbf{I} - \mathcal{P}_t) - \mathbf{H}_t \mathbf{K}_t (\mathbf{I} - \mathcal{P}_t) \\ &= (\mathbf{I} - \mathcal{P}_t) - (\mathbf{I} - \mathcal{P}_t) \mathbf{K}_t \mathbf{S}_t^{-1} \mathbf{K}_t (\mathbf{I} - \mathcal{P}_t). \end{aligned}$$

We refer to the book Meyer (2000) for some properties of the concepts of: Moore–Penrose pseudo-inverse (Exercise 5.12.16) and orthogonal projectors (Chapter 5.13).

G Proof of Proposition 1: Properties of \mathbf{Q}_t

G.1 $\mathbf{I} \succeq \mathbf{I} - \mathcal{P}_t \succeq \mathbf{Q}_t \succeq \mathbf{0}$

Lemma 11. Both \mathcal{P}_t and $\mathbf{I} - \mathcal{P}_t$ are symmetric positive semidefinite.

Lemma 12. $\mathbf{S}_t = \mathbf{K}_t (\mathbf{I} - \mathcal{P}_t) \mathbf{K}_t + \mu_t \mathbf{K}_t$ is symmetric positive definite, which is always invertible.

Proof. $\mu_t \mathbf{K}_t$ is symmetric positive definite, since we assume symmetric positive definite \mathbf{K}_t and $\mu_t > 0$. The orthogonal projector $\mathbf{I} - \mathcal{P}_t$ is symmetric positive semidefinite. Thus $\mathbf{K}_t (\mathbf{I} - \mathcal{P}_t) \mathbf{K}_t$ is symmetric positive semidefinite. As a result, \mathbf{S}_t is symmetric positive definite.

Lemma 13. $\mathbf{I} \succeq \mathbf{I} - \mathcal{P}_t \succeq \mathbf{Q}_t$.

Proof. Given the fact that \mathbf{S}_t is symmetric positive definite, we know \mathbf{S}_t^{-1} is symmetric positive definite. As a result, $(\mathbf{I} - \mathcal{P}_t) \mathbf{K}_t \mathbf{S}_t^{-1} \mathbf{K}_t (\mathbf{I} - \mathcal{P}_t)$ is symmetric positive semidefinite. Therefore:

$$(\mathbf{I} - \mathcal{P}_t) - \mathbf{Q}_t \succeq \mathbf{0} \Leftrightarrow \mathbf{I} - \mathcal{P}_t \succeq \mathbf{Q}_t.$$

$\mathbf{I} \succeq \mathbf{I} - \mathcal{P}_t$ is true since \mathcal{P}_t is positive semidefinite.

Lemma 14. \mathbf{Q}_t can be rewritten as:

$$\begin{aligned} \mathbf{Q}_t &= (\mathbf{H}_t \mathbf{K}_t - \mathbf{I})(\mathbf{I} - \mathcal{P}_t)(\mathbf{K}_t^\top \mathbf{H}_t^\top - \mathbf{I}) \\ &\quad + \mu_t \mathbf{H}_t \mathbf{K}_t \mathbf{H}_t^\top, \quad (52) \end{aligned}$$

where \mathbf{H}_t has been defined in equation (51).

Proof. The proof is given in Appendix I.

Lemma 15. \mathbf{Q}_t is symmetric positive semidefinite.

Proof. In equation (52), we notice that $\mathbf{I} - \mathcal{P}_t$ is symmetric positive semidefinite, and \mathbf{K} is symmetric positive definite.

G.2 $\mathbf{Q}_t \mathbf{1}_{m_t} = \mathbf{0}$

We notice that matrix \mathcal{P}_t is the orthogonal projector to the range space of:

$$\tilde{\mathbf{P}}_t^\top = [\mathbf{P}_t^\top, \mathbf{1}_{m_t}].$$

We further observe that $\mathbf{1}_{m_t}$ is in fact a column of $\tilde{\mathbf{P}}_t^\top$ thus lying in the range of $\tilde{\mathbf{P}}_t^\top$. As a result, we have:

$$\mathcal{P}_t \mathbf{1}_{m_t} = \mathbf{1}_{m_t} \Leftrightarrow (\mathbf{I} - \mathcal{P}_t) \mathbf{1}_{m_t} = \mathbf{0}.$$

It is thus obvious to see $\mathbf{Q}_t \mathbf{1}_{m_t} = \mathbf{0}$.

H Proof of Proposition 2: Properties of \mathbf{H}_t

We compute:

$$\begin{aligned} \mathbf{K}_t \mathbf{S}_t^{-1} \mathbf{K}_t &= \mathbf{K}_t (\mathbf{K}_t (\mathbf{I} - \mathcal{P}_t) \mathbf{K}_t + \mu_t \mathbf{K}_t)^{-1} \mathbf{K}_t \\ &= ((\mathbf{I} - \mathcal{P}_t) + \mu_t \mathbf{K}_t^{-1})^{-1}. \end{aligned}$$

Matrix $\mathbf{K}_t \mathbf{S}_t^{-1} \mathbf{K}_t$ is invertible. Thus we have:

$$\begin{cases} ((\mathbf{I} - \mathcal{P}_t) + \mu_t \mathbf{K}_t^{-1}) \mathbf{K}_t \mathbf{S}_t^{-1} \mathbf{K}_t &= \mathbf{I} \\ \mathbf{K}_t \mathbf{S}_t^{-1} \mathbf{K}_t ((\mathbf{I} - \mathcal{P}_t) + \mu_t \mathbf{K}_t^{-1}) &= \mathbf{I}. \end{cases}$$

Thus the following equalities hold true:

$$\mu_t \mathbf{S}_t^{-1} \mathbf{K}_t = \mathbf{I} - (\mathbf{I} - \mathcal{P}_t) \mathbf{K}_t \mathbf{S}_t^{-1} \mathbf{K}_t \quad (53)$$

$$\mu_t \mathbf{K}_t \mathbf{S}_t^{-1} = \mathbf{I} - \mathbf{K}_t \mathbf{S}_t^{-1} \mathbf{K}_t (\mathbf{I} - \mathcal{P}_t). \quad (54)$$

By right-multiplying equation (53) by $\mathbf{I} - \mathcal{P}_t$ and left-multiplying equation (54) by $\mathbf{I} - \mathcal{P}_t$, we have:

$$\mathbf{Q}_t = \mu_t \underbrace{\mathbf{S}_t^{-1} \mathbf{K}_t (\mathbf{I} - \mathcal{P}_t)}_{\mathbf{H}_t^\top} = \mu_t \underbrace{(\mathbf{I} - \mathcal{P}_t) \mathbf{K}_t \mathbf{S}_t^{-1}}_{\mathbf{H}_t}, \quad (55)$$

where we have used the expression of \mathbf{Q}_t in equation (25).

I Connection to the Result in Bai and Bartoli (2022a)

In Bai and Bartoli (2022a), \mathbf{Q}_t was defined as equation(52) which can be simplified to the form in equation (25).

In equation (55), we have proved that:

$$\mathbf{H} = (\mathbf{I} - \mathcal{P}_t)\mathbf{K}_t\mathbf{S}_t^{-1} = \mathbf{S}_t^{-1}\mathbf{K}_t(\mathbf{I} - \mathcal{P}_t) = \mathbf{H}^\top.$$

Hence, we can rewrite equation (54) in Appendix H as:

$$\mu_t\mathbf{K}_t\mathbf{S}_t^{-1} = \mathbf{I} - \mathbf{K}_t(\mathbf{I} - \mathcal{P}_t)\mathbf{K}_t\mathbf{S}_t^{-1}. \quad (56)$$

We examine the term $\mu_t\mathbf{H}_t\mathbf{K}_t\mathbf{H}_t^\top$ with the identity (56), as:

$$\begin{aligned} \mu_t\mathbf{H}_t\mathbf{K}_t\mathbf{H}_t^\top &= \mu_t\mathbf{H}_t\mathbf{K}_t\mathbf{S}_t^{-1}\mathbf{K}_t(\mathbf{I} - \mathcal{P}_t) \\ &= \mathbf{H}_t(\mathbf{I} - \mathbf{K}_t(\mathbf{I} - \mathcal{P}_t)\mathbf{K}_t\mathbf{S}_t^{-1})\mathbf{K}_t(\mathbf{I} - \mathcal{P}_t) \\ &= \mathbf{H}_t\mathbf{K}_t(\mathbf{I} - \mathcal{P}_t) - \mathbf{H}_t\mathbf{K}_t(\mathbf{I} - \mathcal{P}_t)\mathbf{K}_t\mathbf{H}_t^\top \\ &= -\mathbf{H}_t\mathbf{K}_t(\mathbf{I} - \mathcal{P}_t)(\mathbf{K}_t\mathbf{H}_t^\top - \mathbf{I}). \end{aligned} \quad (57)$$

Substituting equation (57) into equation (52) to cancel the term $\mu_t\mathbf{H}_t\mathbf{K}_t\mathbf{H}_t^\top$, we reach \mathbf{Q}_t in equation (25) by some trivial matrix manipulations:

$$\begin{aligned} &(\mathbf{H}_t\mathbf{K}_t - \mathbf{I})(\mathbf{I} - \mathcal{P}_t)(\mathbf{K}_t^\top\mathbf{H}_t^\top - \mathbf{I}) + \mu_t\mathbf{H}_t\mathbf{K}_t\mathbf{H}_t^\top \\ &= -(\mathbf{I} - \mathcal{P}_t)(\mathbf{K}_t\mathbf{H}_t^\top - \mathbf{I}) \\ &= -(\mathbf{I} - \mathcal{P}_t)(\mathbf{K}_t\mathbf{S}_t^{-1}\mathbf{K}_t(\mathbf{I} - \mathcal{P}_t) - \mathbf{I}) \\ &= (\mathbf{I} - \mathcal{P}_t) - (\mathbf{I} - \mathcal{P}_t)\mathbf{K}_t\mathbf{S}_t^{-1}\mathbf{K}_t(\mathbf{I} - \mathcal{P}_t) = \mathbf{Q}_t. \end{aligned}$$

J Proof of Lemma 5

It can be shown that:

$$\begin{aligned} \tilde{\mathbf{P}}_t \stackrel{\text{def}}{=} \begin{bmatrix} \mathbf{P}_t \\ \mathbf{1}^\top \end{bmatrix} &= \begin{bmatrix} \mathbf{R}_t\mathbf{P}_t + \mathbf{t}_t\mathbf{1}^\top \\ \mathbf{1}^\top \end{bmatrix} = \begin{bmatrix} \mathbf{R}_t & \mathbf{t}_t \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{P}_t \\ \mathbf{1}^\top \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{R}_t & \mathbf{t}_t \\ \mathbf{0} & 1 \end{bmatrix} \tilde{\mathbf{P}}_t. \end{aligned}$$

We notice that $\tilde{\mathbf{P}}_t^\top$ and $\tilde{\mathbf{P}}_t^\top$ have the same range space, thus the orthogonal projection matrices are the same by the uniqueness Meyer (2000). We can also verify the result by direct matrix calculations. We notice:

$$\left(\tilde{\mathbf{P}}_t\tilde{\mathbf{P}}_t^\top\right)^\dagger = \begin{bmatrix} \mathbf{R}_t & \mathbf{t}_t \\ \mathbf{0} & 1 \end{bmatrix}^{-\top} \left(\tilde{\mathbf{P}}_t\tilde{\mathbf{P}}_t^\top\right)^\dagger \begin{bmatrix} \mathbf{R}_t & \mathbf{t}_t \\ \mathbf{0} & 1 \end{bmatrix}^{-1},$$

thus we have $\tilde{\mathbf{P}}_t^\top(\tilde{\mathbf{P}}_t\tilde{\mathbf{P}}_t^\top)^\dagger\tilde{\mathbf{P}}_t = \tilde{\mathbf{P}}_t^\top(\tilde{\mathbf{P}}_t\tilde{\mathbf{P}}_t^\top)^\dagger\tilde{\mathbf{P}}_t$.

K Expansion of $\text{tr}(\mathbf{X}^\top\mathcal{Q}\mathbf{X}\Lambda)$

We denote $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3]$. We denote the matrix square root of \mathbf{Q}_t as $\sqrt{\mathbf{Q}_t}$, where: $\mathbf{Q}_t = \sqrt{\mathbf{Q}_t}^\top\sqrt{\mathbf{Q}_t}$.

$$\begin{aligned} \text{tr}(\mathbf{X}^\top\mathcal{Q}\mathbf{X}\Lambda) &= \sum_{t=1}^n \text{tr}(\mathbf{X}^\top\Gamma_t\mathbf{Q}_t\Gamma_t^\top\mathbf{X}\Lambda) \\ &= \sum_{t=1}^n \sum_{k=1}^3 \lambda_k \text{tr}(\mathbf{x}_k^\top\Gamma_t\mathbf{Q}_t\Gamma_t^\top\mathbf{x}_k) \\ &= \sum_{t=1}^n \sum_{k=1}^3 \lambda_k \left\| \sqrt{\mathbf{Q}_t}(\mathbf{I} - \mathcal{P}_t)\Gamma_t^\top\mathbf{x}_k \right\|_{\mathcal{F}}^2 \\ &= \sum_{k=1}^3 \sum_{t=1}^n \lambda_k \left\| \sqrt{\mathbf{Q}_t}(\mathbf{I} - \mathcal{P}_t)\Gamma_t^\top\mathbf{x}_k \right\|_{\mathcal{F}}^2 \\ &= \sum_{k=1}^3 \lambda_k \left\| \begin{bmatrix} \sqrt{\mathbf{Q}_1}(\mathbf{I} - \mathcal{P}_1)\Gamma_1^\top \\ \vdots \\ \sqrt{\mathbf{Q}_n}(\mathbf{I} - \mathcal{P}_n)\Gamma_n^\top \end{bmatrix} \mathbf{x}_k \right\|_{\mathcal{F}}^2. \end{aligned}$$

L Planar Case

The optimal $\sqrt{\Lambda}$ is characterized by formulation (34), and thus formulation (39). If \mathbf{P}_t is flat, then \mathbf{P}_t can be rigidly transformed to $[\mathbf{P}_t^\top \mathbf{0}]^\top$. Thus it suffices to discuss the estimate of $\sqrt{\Lambda}$ from the canonical 2D point-clouds \mathbf{P}_t . We denote $\mathbf{G}_t = \mathbf{X}^\top\Gamma_t$ where \mathbf{X} is given at this stage. We further denote $\bar{\mathbf{G}}_t = \mathbf{G}_t - \frac{1}{m_t}\mathbf{G}_t\mathbf{1}\mathbf{1}^\top$. By the fact that \mathbf{P}_t is zero-centered, $\sqrt{\Lambda}$ is characterized by problem (41).

In problem (41), if the optimal \mathbf{R}_t implements a rotation in the xy -plane, then \mathbf{R}_t can be formed as:

$$\mathbf{R}_t = \begin{bmatrix} \mathbf{R}_{txy} & \\ & 1 \end{bmatrix}.$$

As a result, problem (41) can be decomposed as:

$$\begin{aligned} &\min_{\{\mathbf{R}_{txy} \in \text{SO}(2)\}, \eta_{xy} \in \mathbb{R}^2, \eta_z \in \mathbb{R}} \sum_{t=1}^n \eta_z \|\mathbf{g}_3^\top\|^2 \\ &+ \sum_{t=1}^n \left\| \mathbf{R}_{txy}\mathbf{P}_t - \text{diag}(\eta_{xy}) \begin{bmatrix} \mathbf{g}_1^\top \\ \mathbf{g}_2^\top \end{bmatrix} \right\|_{\mathcal{F}}^2, \end{aligned}$$

where $\bar{\mathbf{G}}_t = [\mathbf{g}_1 \ \mathbf{g}_2 \ \mathbf{g}_3]^\top$. The cost of this problem is minimized if and only if $\eta_z = 0$. Thus $\lambda_3 = |\eta_z| = 0$.

M GPA Using the LBW in Bai and Bartoli (2022b)

We recapitulate the result of Bai and Bartoli (2022b). If using the LBW, we will be solving a GPA formulation as:

$$\begin{aligned} &\min_{\{\mathbf{W}_t\}, \mathbf{M}} \sum_{t=1}^n \varphi_t(\mathbf{W}_t, \mathbf{M}) \\ &\text{s.t. } \mathbf{M}\mathbf{1} = \mathbf{0}, \mathbf{M}\mathbf{M}^\top = \Lambda, \end{aligned} \quad (58)$$

where:

$$\varphi_t(\mathbf{W}_t, \mathbf{M}) = \|\mathbf{W}_t^\top \mathbf{B}_t(\mathbf{P}_t) - \mathbf{M}\Gamma_t\|_{\mathcal{F}}^2 + \mu_t \text{tr}(\mathbf{W}_t^\top \Xi_t \mathbf{W}_t).$$

We define the matrix \mathcal{Q} as:

$$\mathcal{Q} \stackrel{\text{def}}{=} \Gamma_t \left(\mathbf{I} - \mathbf{B}_t^\top \left(\mathbf{B}_t \mathbf{B}_t^\top + \mu_t \Xi_t \right)^{-1} \mathbf{B}_t \right) \Gamma_t^\top,$$

where we have used the shorthand $\mathbf{B}_t \stackrel{\text{def}}{=} \mathbf{B}_t(\mathbf{P}_t)$.

If $\mathcal{Q}\mathbf{1} = \mathbf{0}$, then the optimal \mathbf{M} of problem (58) is:

$$\mathbf{M} = \sqrt{\Lambda} \mathbf{X}^\top, \quad \text{where } \mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_d] \in \mathbb{R}^{m \times d},$$

where $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_d$ in sequence are the d bottom eigenvectors of \mathcal{Q} excluding the vector $\mathbf{1}$, or equivalently are the d bottom eigenvectors of $\mathcal{Q}' = \mathcal{Q} + n\mathbf{1}\mathbf{1}^\top$. The optimal transformation parameters \mathbf{W}_t are:

$$\mathbf{W}_t^\top = \mathbf{M}\Gamma_t \mathbf{B}_t^\top \left(\mathbf{B}_t \mathbf{B}_t^\top + \mu_t \Xi_t \right)^{-1}, \quad (t \in [1 : n]).$$

It has been shown that $\mathcal{Q}\mathbf{1} = \mathbf{0}$ happens if the LBW has a free translation. In particular, if the LBW is chosen as the affine transformation, or the TPS warp, then $\mathcal{Q}\mathbf{1} = \mathbf{0}$.

Acknowledgements

The author would like to thank professor Yi Dong in Tongji University, Shanghai, China, for providing the financial support to finish the initial manuscript. We want to express our gratitude to the TOPACS team (project No. ANR-19-CE45-0015) who produced the original CT point-cloud for our experiments.

References

- Abzil PA, Mahony R and Sepulchre R (2009) *Optimization algorithms on matrix manifolds*. Princeton University Press.
- Allen B, Curless B and Popović Z (2003) The space of human body shapes: reconstruction and parameterization from range scans. *ACM transactions on graphics (TOG)* 22(3): 587–594.
- Anguelov D, Srinivasan P, Koller D, Thrun S, Rodgers J and Davis J (2005) SCAPE: shape completion and animation of people. In: *ACM SIGGRAPH 2005 Papers*. pp. 408–416.
- Arun KS, Huang TS and Blostein SD (1987) Least-squares fitting of two 3-d point sets. *IEEE Transactions on pattern analysis and machine intelligence* (5): 698–700.
- Bai F and Bartoli A (2022a) KernelGPA: A Deformable SLAM Back-end. In: *Proceedings of Robotics: Science and Systems*. New York City, NY, USA. DOI:10.15607/RSS.2022.XVIII.002.
- Bai F and Bartoli A (2022b) Procrustes analysis with deformations: A closed-form solution by eigenvalue decomposition. *International Journal of Computer Vision* 130(2): 567–593.
- Bartoli A, Gérard Y, Chadebecq F, Collins T and Pizarro D (2015) Shape-from-template. *IEEE transactions on pattern analysis and machine intelligence* 37(10): 2099–2118.
- Bartoli A, Perriollat M and Chambon S (2010) Generalized thin-plate spline warps. *International Journal of Computer Vision* 88(1): 85–110.
- Birtea P, Cașu I and Comănescu D (2019) First order optimality conditions and steepest descent algorithm on orthogonal stiefel manifolds. *Optimization Letters* 13(8): 1773–1791.
- Bookstein FL (1989) Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Transactions on pattern analysis and machine intelligence* 11(6): 567–585.
- Bouix S, Pruessner JC, Collins DL and Siddiqi K (2005) Hippocampal shape analysis using medial surfaces. *Neuroimage* 25(4): 1077–1089.
- Bregler C, Hertzmann A and Biermann H (2000) Recovering non-rigid 3d shape from image streams. In: *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No. PR00662)*, volume 2. IEEE, pp. 690–696.
- Brockett RW (1989) Least squares matching problems. *Linear Algebra and its applications* 122: 761–777.
- Cadena C, Carlone L, Carrillo H, Latif Y, Scaramuzza D, Neira J, Reid I and Leonard JJ (2016) Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on robotics* 32(6): 1309–1332.
- Dai Y, Li H and He M (2014) A simple prior-free method for non-rigid structure-from-motion factorization. *International Journal of Computer Vision* 107(2): 101–122.
- Dissanayake MG, Newman P, Clark S, Durrant-Whyte HF and Csorba M (2001) A solution to the simultaneous localization and map building (slam) problem. *IEEE Transactions on robotics and automation* 17(3): 229–241.
- Duchon J (1976) Interpolation des fonctions de deux variables suivant le principe de la flexion des plaques minces. *Revue française d'automatique, informatique, recherche opérationnelle. Analyse numérique* 10(R3): 5–12.
- Feng Y, Feng H, Black MJ and Bolkart T (2021) Learning an animatable detailed 3D face model from in-the-wild images. URL <https://doi.org/10.1145/3450626.3459936>.
- Fletcher PT, Lu C, Pizer SM and Joshi S (2004) Principal geodesic analysis for the study of nonlinear statistics of shape. *IEEE transactions on medical imaging* 23(8): 995–1005.
- Fornet M, Rohr K and Stiehl HS (2001) Radial basis functions with compact support for elastic registration of medical images. *Image and vision computing* 19(1-2): 87–96.
- Freifeld O and Black MJ (2012) Lie bodies: A manifold representation of 3d human shape. In: *European Conference on Computer Vision*. Springer, pp. 1–14.
- Gallardo M, Collins T and Bartoli A (2017) Dense non-rigid structure-from-motion and shading with unknown albedos. In: *Proceedings of the IEEE international conference on computer vision*. pp. 3884–3892.
- Gallier (2010) The schur complement and symmetric positive semidefinite (and definite) matrices .
- Gao W and Tedrake R (2018) Surfelwarp: Efficient non-volumetric single view dynamic reconstruction. In: *Robotics: Science and System (RSS)*.
- Golub G and Pereyra V (2003) Separable nonlinear least squares: the variable projection method and its applications. *Inverse problems* 19(2): R1.
- Goodall C (1991) Procrustes methods in the statistical analysis of shape. *Journal of the Royal Statistical Society: Series B (Methodological)* 53(2): 285–321.
- Hardy G, Collection KMR, Littlewood J, Pólya G, Pólya G and Littlewood D (1952) *Inequalities*. Cambridge Mathematical

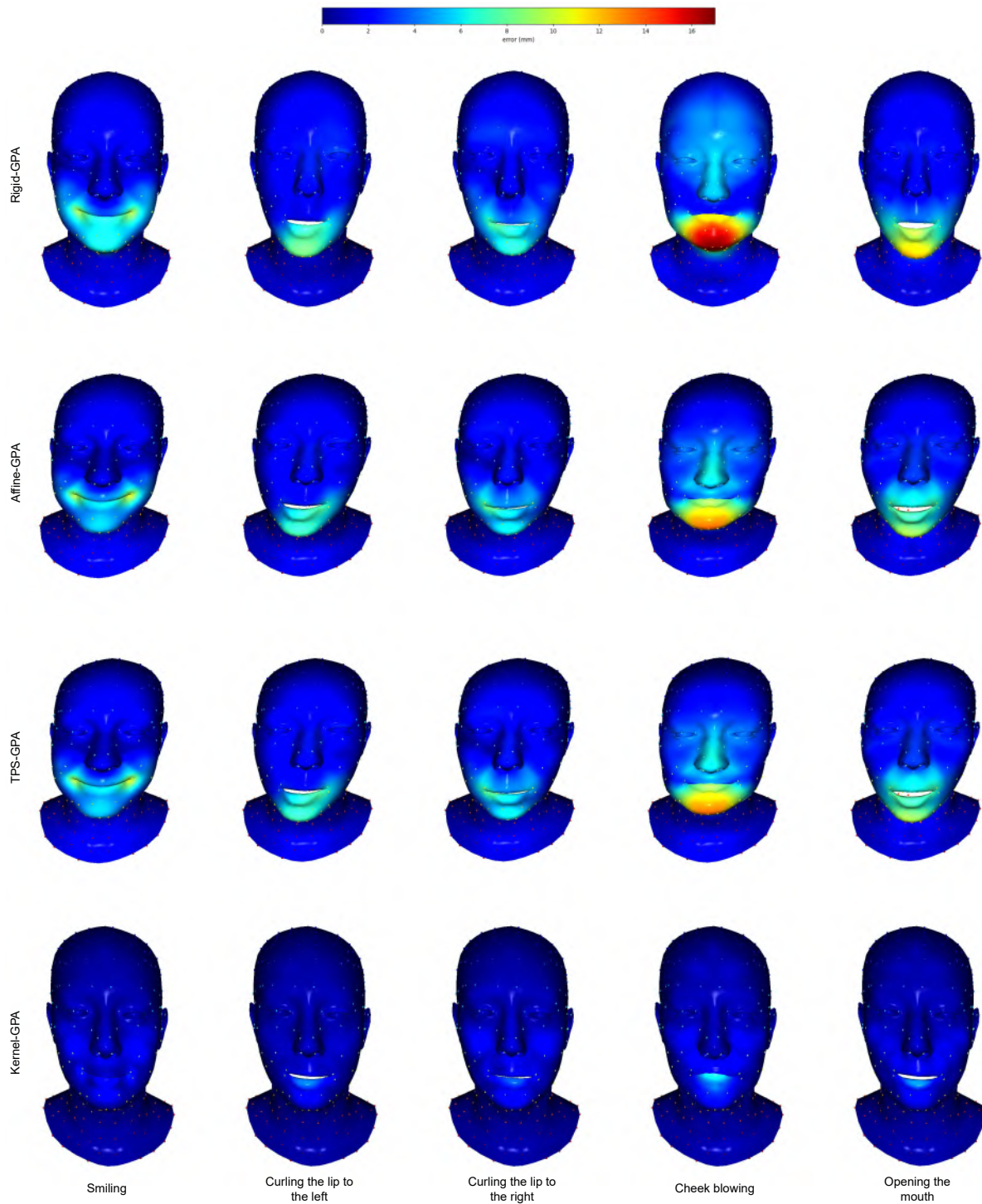


Figure 14. The deformable transformations $y_t(\cdot)$ of different GPA methods on the facial expression dataset. We visualize the shape of the transformed test points $y_t({}^T\mathbf{P}_t)$ and encode the point-wise discrepancies between $y_t({}^T\mathbf{P}_t)$ and the mean map ${}^T\hat{\mathbf{M}}^T\mathbf{\Gamma}_t$ with color. The markers on the face represent the transformed correspondences, *i.e.*, $y_t(\mathbf{P}_t)$.

Library. Cambridge University Press. ISBN 9780521358804.

Hartley R and Zisserman A (2004) *Multiple View Geometry in Computer Vision*. Second edition. Cambridge University Press, ISBN: 0521540518.

Horn BK, Hilden HM and Negahdaripour S (1988) Closed-form solution of absolute orientation using orthonormal matrices. *JOSA A* 5(7): 1127–1135.

Huang S, Chen Y, Zhao L, Zhang Y and Xu M (2021) Some research questions for slam in deformable environments. In:

2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 7653–7660.

Jermyn IH, Kurtek S, Klassen E and Srivastava A (2012) Elastic shape matching of parameterized surfaces using square root normal fields. In: *European conference on computer vision*. Springer, pp. 804–817.

Joshi SH, Klassen E, Srivastava A and Jermyn I (2007) A novel representation for riemannian analysis of elastic curves in rn. In: *2007 IEEE Conference on Computer Vision and Pattern*

- Recognition*. IEEE, pp. 1–7.
- Kendall DG (1984) Shape manifolds, procrustean metrics, and complex projective spaces. *Bulletin of the London mathematical society* 16(2): 81–121.
- Kilian M, Mitra NJ and Pottmann H (2007) Geometric modeling in shape space. In: *ACM SIGGRAPH 2007 papers*. pp. 64–es.
- Kurtek S, Klassen E, Ding Z and Srivastava A (2010) A novel riemannian framework for shape analysis of 3d objects. In: *2010 IEEE computer society conference on computer vision and pattern recognition*. IEEE, pp. 1625–1632.
- Kurtek S, Klassen E, Gore JC, Ding Z and Srivastava A (2011) Elastic geodesic paths in shape space of parameterized surfaces. *IEEE transactions on pattern analysis and machine intelligence* 34(9): 1717–1730.
- Laga H (2018) A survey on nonrigid 3d shape analysis. *Academic Press Library in Signal Processing, Volume 6* : 261–304.
- Laga H, Xie Q, Jermyn IH and Srivastava A (2017) Numerical inversion of srnf maps for elastic shape analysis of genus-zero surfaces. *IEEE transactions on pattern analysis and machine intelligence* 39(12): 2451–2464.
- Lamarca J, Parashar S, Bartoli A and Montiel J (2020) Defslam: Tracking and mapping of deforming scenes from monocular sequences. *IEEE Transactions on robotics* 37(1): 291–303.
- Malti A and Herzet C (2017) Elastic shape-from-template with spatially sparse deforming forces. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp. 3337–3345.
- Meyer CD (2000) *Matrix analysis and applied linear algebra*, volume 71. Siam.
- Newcombe RA, Fox D and Seitz SM (2015) Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp. 343–352.
- Osher S and Fedkiw RP (2003) *Level set methods and dynamic implicit surfaces*, volume 153. Springer.
- Parashar S, Pizarro D and Bartoli A (2017) Isometric non-rigid shape-from-motion with riemannian geometry solved in linear time. *IEEE transactions on pattern analysis and machine intelligence* 40(10): 2442–2454.
- Parashar S, Pizarro D and Bartoli A (2019) Local deformable 3d reconstruction with cartan’s connections. *IEEE transactions on pattern analysis and machine intelligence* 42(12): 3011–3026.
- Raju IS and Newman JC (1993) surf3d: A 3-d finite-element program for the analysis of surface and corner cracks in solids subjected to mode-1 loadings.
- Rohlf FJ and Slice D (1990) Extensions of the Procrustes method for the optimal superimposition of landmarks. *Systematic biology* 39(1): 40–59.
- Rueckert D, Sonoda LI, Hayes C, Hill DL, Leach MO and Hawkes DJ (1999) Nonrigid registration using free-form deformations: application to breast mr images. *IEEE transactions on medical imaging* 18(8): 712–721.
- Schölkopf B, Herbrich R and Smola AJ (2001) A generalized representer theorem. In: *International conference on computational learning theory*. Springer, pp. 416–426.
- Shawe-Taylor J and Cristianini N (2004) Kernel methods for pattern analysis .
- Slavcheva M, Baust M, Cremers D and Ilic S (2017) Killingfusion: Non-rigid 3d reconstruction without correspondences. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pp. 1386–1395.
- Slavcheva M, Baust M and Ilic S (2018) Sobolevfusion: 3d reconstruction of scenes undergoing free non-rigid motion. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pp. 2646–2655.
- Song J, Wang J, Zhao L, Huang S and Dissanayake G (2018) Mis-slam: Real-time large-scale dense deformable slam system in minimal invasive surgery based on heterogeneous computing. *IEEE Robotics and Automation Letters* 3(4): 4068–4075.
- Sorkine O and Alexa M (2007a) As-rigid-as-possible surface modeling. In: *Symposium on Geometry processing*, volume 4. pp. 109–116.
- Sorkine O and Alexa M (2007b) As-Rigid-As-Possible Surface Modeling. In: Belyaev A and Garland M (eds.) *Geometry Processing*. The Eurographics Association. ISBN 978-3-905673-46-3. DOI:10.2312/SGP/SGP07/109-116.
- Sumner RW, Schmid J and Pauly M (2007) Embedded deformation for shape manipulation. In: *ACM SIGGRAPH 2007 papers*. pp. 80–es.
- Szeliski R and Coughlan J (1997) Spline-based image registration. *International Journal of Computer Vision* 22(3): 199–218.
- Tomasi C and Kanade T (1992) Shape and motion from image streams under orthography: a factorization method. *International journal of computer vision* 9(2): 137–154.
- Xiao J, Chai J and Kanade T (2006) A closed-form solution to non-rigid shape and motion recovery. *International Journal of Computer Vision* 67(2): 233–246.
- Younes L (2012) Spaces and manifolds of shapes in computer vision: An overview. *Image and Vision Computing* 30(6-7): 389–397.
- Younes L, Michor PW, Shah JM and Mumford DB (2008) A metric on shape space with explicit geodesics. *Rendiconti Lincei-Matematica e Applicazioni* 19(1): 25–57.