Registration of algebraic varieties using Riemannian optimization

Florentin Goyens[1,2], Coralia Cartis[2], and Stéphane Chrétien[3]

[1]LAMSADE Université Paris Dauphine PSL, France
[2]University of Oxford, UK, mathematical institute
[3]National Physical Laboratory, UK*
Corresponding author: goyensflorentin@gmail.com

January 16, 2024

Abstract

We consider the point cloud registration problem, the task of finding a transformation between two point clouds that represent the same object but are expressed in different coordinate systems. Our approach is not based on a point-to-point correspondence, matching every point in the source point cloud to a point in the target point cloud. Instead, we assume and leverage a low-dimensional nonlinear geometric structure of the data. Firstly, we approximate each point cloud by an algebraic variety (a set defined by finitely many polynomial equations). This is done by solving an optimization problem on the Grassmann manifold, using a connection between algebraic varieties and polynomial bases. Secondly, we solve an optimization problem on the orthogonal group to find the transformation (rotation + translation) which makes the two algebraic varieties overlap. We use second-order Riemannian optimization methods for the solution of both steps. Numerical experiments on real and synthetic data are provided, with encouraging results. Our approach is particularly useful when the two point clouds describe different parts of an objects (which may not even be overlapping), on the condition that the surface of the object may be well approximated by a set of polynomial equations. The first procedure—the approximation—is of independent interest, as it can be used for denoising data that belongs to an algebraic variety. We provide statistical guarantees for the estimation error of the denoising using Stein's unbiased estimator.

keywords: point cloud registration, Riemannian optimization, algebraic varieties.

1 Introduction

The problem of point cloud registration consists in finding a transformation between two point clouds that represent the same shape but are expressed in different coordinate systems. It appears for instance in computer vision (Sharp et al., 2004), distributed approaches to molecular conformation (Cucuringu et al., 2012b), and sensor network localization (Cucuringu et al., 2012a). In this work, we consider instances where the shape of the point clouds can be represented by an algebraic variety (a set defined by finitely many polynomial equations). We develop a registration method based on manifold optimization which exploits the algebraic variety structure of the data.

*This work was generously supported by the National Physical Laboratory (NPL), United Kingdom.

Consider two algebraic varieties $V_1, V_2 \subset \mathbb{R}^n$ of the same degree d (Definition 2.1). We assume the existence of a rigid transformation $\mathcal{T}\colon \mathbb{R}^n \to \mathbb{R}^n$ that makes the varieties V_1 and V_2 overlap.

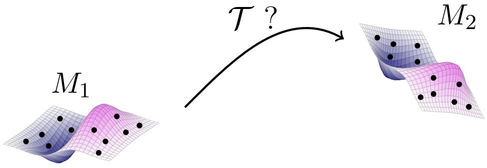
A1. *There exists $Q \in \mathrm{O}(n)$ and $a \in \mathbb{R}^n$ such that for all $x_1 \in V_1$, we have*

$$\mathcal{T}(x_1) := Qx_1 + a \in V_2, \tag{1.1}$$

where $\mathrm{O}(n) = \left\{Q \in \mathbb{R}^{n\times n} : Q^\top Q = \mathrm{I}_n\right\}$ *denotes the* orthogonal group.

Intuitively, V_1 and V_2 represent the same shape but in different coordinate systems, which are connected by a rotation and translation.

A matrix $X \in \mathbb{R}^{n\times s}$ represents a *point cloud* in $\mathbb{R}^n$ and each column of X is called a *data point* or *sample*. Let $M_1 \in \mathbb{R}^{n\times s_1}$ and $M_2 \in \mathbb{R}^{n\times s_2}$ be composed of respectively s_1 samples in V_1 and s_2 samples in V_2. Given M_1 and M_2, our task is to estimate $Q \in \mathrm{O}(n)$ and $a \in \mathbb{R}^n$ which define the transformation $\mathcal{T}$. The samples of $M_1 \in \mathbb{R}^{n\times s_1}$ and $M_2 \in \mathbb{R}^{n\times s_2}$ are independent and the two point clouds may have a different number of samples ($s_1 \neq s_2$). The goal is therefore not to establish a point-to-point matching, but to find a transformation such that the varieties V_1 and V_2 — observed through the samples in M_1 and M_2 — overlap as best as possible.



The situation described in A1 is a *noiseless* setting, in which all the columns of M_1 belong exactly to the same algebraic variety. This is not likely to occur in practice, instead we expect that the data approximately belongs to an algebraic variety. It is this *noisy* setting that we consider. Our strategy is to first tackle the problem of finding the algebraic variety that best approximates a point cloud. We do so in Section 3. Each point cloud is approximated by an algebraic variety, and then we compute a rigid transformation between these two algebraic varieties (Section 4).

We model each of these two recovery problems (approximation by an algebraic variety and registration) as the minimization of a function defined on a combination of the following sets: the orthogonal group, the Grassmann manifold, and the Euclidean spaces $\mathbb{R}^{n\times s}$ and $\mathbb{R}^n$. These optimization problems are instances of the problem class

$$\min_{x\in\mathcal{M}} f(x) \tag{1.2}$$

where $f\colon \mathcal{M} \to \mathbb{R}$ is a smooth nonconvex function defined on a Riemannian manifold $\mathcal{M}$. Hence, all the optimization problems that we formulate can be minimized using off-the-shelf Riemannian optimization methods (Absil et al., 2008; Boumal, 2023), which are designed for problems of the form (1.2). When the problems are nonconvex, there are no a priori guarantees to find the global minimizer of f on $\mathcal{M}$, and our numerical results show that local Riemannian optimization methods perform well on our test problems in Sections 3.3 and 4.2. We present a summary of the main existing approaches for registration in order to highlight the novelties in our contribution.

1.1 Related work

Several approaches have been devised for the various types of registration problems that appear in image and pattern analysis. Registration algorithms are classified into rigid and non-rigid approaches. Rigid registration methods find affine transformations—rotations and translations—that preserve distances. The non-rigid registration methods also consider nonlinear transformations.

One of the early instances of registration in the literature is the (orthogonal) Procruste problem (Schönemann, 1966). Given two matrices $A, B \in \mathbb{R}^{n\times s}$, one tries to find the orthogonal matrix $Q^* \in \mathrm{O}(n)$ that best matches A to B in Frobenius norm, that is,

$$Q^* = \operatorname*{arg\,min}_{Q\in\mathrm{O}(n)} \|QA - B\|_{\mathrm{F}} .$$

This problem is equivalent to finding the nearest orthogonal matrix to $BA^\top$,

$$Q^* = \operatorname*{arg\,min}_{Q\in\mathrm{O}(n)} \left\|Q - BA^\top\right\|_{\mathrm{F}} ,$$

which, given the singular value decomposition $BA^\top = U\Sigma V^\top$, is $Q^* = UV^\top$.

A natural extension is to consider a transformation which combines an orthogonal matrix with a translation. Consider two point clouds $M_1 = \{x_1, \ldots, x_s\}$ and $M_2 = \{y_1, \ldots, y_s\}$ in $\mathbb{R}^n$, respectively called the *source* and the *target* point clouds. One is obtained through a *rigid transformation* of the other:

$$y_k = Qx_k + a \qquad\qquad \text{for } k = 1, \ldots, s, \tag{1.3}$$

where $Q \in \mathrm{O}(n)$ is a rotation and $a \in \mathbb{R}^n$ is a translation. In this setting, *exact point matching* is assumed: the two point clouds have the same number of samples and each sample matches with one and only one sample from the other point cloud. This leads to the following least squares problem

$$(Q^*, a^*) = \operatorname*{arg\,min}_{Q\in\mathrm{O}(n), a\in\mathbb{R}^n} \sum_{k=1}^{s} \|y_k - Qx_k - a\|_2^2 . \tag{1.4}$$

The optimization over $\mathrm{O}(n) \times \mathbb{R}^n$ seems difficult since the set $\mathrm{O}(n)$ is nonconvex. Remarkably, there exists a closed-form solution for the global minimizer of (1.4). Arun et al. (1987) show that the optimal orthogonal matrix is given by $Q^* = UV^\top$ where $U\Sigma V^\top$ is the SVD of

$$\sum_{k=1}^{s} (x_k - x_c)(y_k - y_c)^\top ,$$

with $x_c = (x_1 + \cdots + x_s)/s$ and $y_c = (y_1 + \cdots + y_s)/s$, the centroids of the two point clouds. The optimal translation is $a^* = y_c - Q^* x_c$. In the case of three or more point clouds to align, no closed-form solution is known. Krishnan et al. (2005) consider the case of multiple point clouds using a rigid transformation and exact point matching. They propose to solve a single optimization problem with one variable on $\mathrm{SO}(n) = \left\{Q \in \mathbb{R}^{n\times n} : Q^\top Q = \mathrm{I}_n, \det(Q) = +1\right\}$ for each point cloud. Chaudhury et al. (2015) also consider an arbitrary number of point clouds and show how the least square formulation, which generalizes (1.4) to several point clouds, can be relaxed into a convex program, with guarantees of tightness.

These approaches, as well as the Procruste problem, require knowledge of the *point correspondence*: which point from the target corresponds to a given point in the source point cloud.

This is unknown in a number of applications. For problems with exact point matching, the crux of the problem then resides in finding the correct point correspondence. The registration is then solved using a single SVD to find the global solution of (1.4).

The iterative closest point (ICP) algorithm, introduced by Besl and McKay (1992), addresses the issue of the unknown point correspondence. Assuming exact point matching, the ICP algorithm performs rigid registration in an iterative fashion. The method repeats the following two steps until the least squares error becomes smaller than some threshold:

1. find a point correspondence using the nearest neighbour rule;

2. estimate a rotation and translation by solving (1.4) using SVD.

In the original version of ICP, the matching in step 1 is determined by the nearest neighbour criteria. For a point in M_1, its corresponding point is the closest point from M_2 in Euclidean norm. Note that this may not produce a one-to-one matching. Because the point correspondence changes throughout the iterations, it is difficult to establish theoretical guarantees of convergence for the classical ICP algorithm. The use of the least squares residual (1.4) yields a method referred to as point-to-point ICP and is known to be sensitive to noise and outliers (Bellekens et al., 2014).

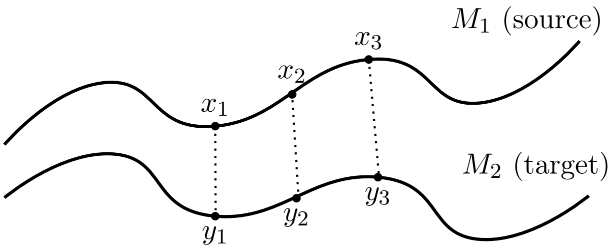


Figure 1: Point-to-point ICP.

Numerous adaptations of the original ICP algorithm have been proposed, impacting all stages of the algorithm.

The point-to-plane ICP (also called point-to-surface) is one such adaptation (Chen and Medioni, 1992). To reduce the algorithm's sensitivity to noise, step 2 minimizes the distance between a source point and the linear approximation of the point cloud at its target point (Figure 2). This gives

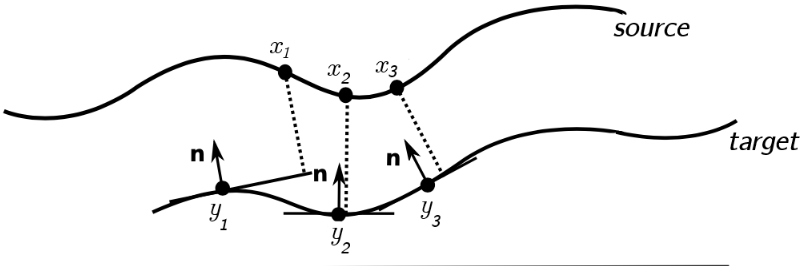


Figure 2: Point-to-plane ICP (Bellekens et al., 2014).

$$(Q^*, a^*) = \underset{Q \in \mathrm{O}(n), a \in \mathbb{R}^n}{\arg\min} \sum_{k=1}^{s} |\, \langle y_k - Qx_k - a, \mathbf{n}_k \rangle \,|^2, \tag{1.5}$$

where y_k is the corresponding point to x_k and $\mathbf{n}_k$ is the normal vector to the linear approximation of the point cloud at y_k. This point-to-plane error metric does not have a closed-form solution and is minimized using nonlinear least squares methods. Although each iteration of the point-to-plane ICP is typically slower than an iteration of the point-to-point ICP, the former displays a faster convergence rate in practice (Rusinkiewicz and Levoy, 2001), as is predicted by theoretical analysis (Pottmann et al., 2004). Plane-to-plane error metrics have also been proposed, which minimize the distance between the two tangent approximations around corresponding points. Generalized ICP allows to leverage the covariance matrix of the point clouds (Segal et al., 2009). Variants of ICP are summarized in (Rusinkiewicz and Levoy, 2001).

In the non-rigid registration paradigm, the transformation is no longer restricted to the combination of a rotation and a translation. There exists a wide range of approaches for the non-rigid registration which we do not cover in depth. Some of the popular approaches for non-rigid registration include robust point matching (Gold et al., 1998); thin plate spline robust point matching (Chui and Rangarajan, 2003); kernel correlation approach (Tsin and Kanade, 2004) and coherent point drift (Myronenko and Song, 2010).

The idea of learning the equations that define an algebraic variety also appears in (Breiding et al., 2018), where the authors investigate topological properties of the algebraic variety based on samples. Fan and Chow (2018) consider the denoising of images using the polynomial kernel.

1.2 Contributions and outline of the paper

We attempt to find a rigid transformation between two point clouds and assume that the target and source point clouds belong to the same algebraic variety (up to a change in coordinates). Our method does not assume that a point correspondence is given, nor does it attempt to compute one. This avoids the convergence issues that stem from the iterative estimation of a point correspondence. The transformation that we compute aims that the image of the source by the rigid transformation belongs to the algebraic variety of the target point cloud. This conceptually aligns the algebraic varieties of the source and target point clouds.

In order to reduce the sensitivity to noise and outliers, variants of the ICP algorithm minimize a point-to-plane or plane-to-plane metric using local linear approximations of the point clouds. The approach we consider can be labelled as a *point-to-algebraic variety* matching, which does not minimize a distance between pairs of points or planes. Our residual is computed in a space of polynomial features, and measures the distance between a source point and the algebraic variety that defines the target point cloud. This also allows us to seamlessly consider two point clouds of different sizes and handle cases where the source point cloud only overlaps partially with the target point cloud, which is an issue for point-to-point methods.

In Section 2, we describe the connection between algebraic varieties and polynomial bases, which allow us to computationally represent algebraic varieties. For noisy data that does not belong to an algebraic variety, we have developed a technique to identify an approximation of each point cloud by an algebraic variety (Section 3). This is then used to compute the rigid transformation (Section 4). We show numerical results on synthetic examples and 3D medical scans that illustrate the efficiency and accuracy of our approach.

2 Preliminaries: algebraic varieties and polynomial features

We describe how points that belong to an algebraic variety interact with a polynomial basis. An algebraic variety is a subset of $\mathbb{R}^n$ defined by finitely many polynomial equations.

Definition 2.1 (Algebraic variety model (Cox et al., 1994))**.** *Let $\mathbb{R}_d[x]$ be the set of real-valued polynomials of degree at most d over $\mathbb{R}^n$. A real (affine)* algebraic variety *of degree d is defined* *as the roots of a system of polynomials $P \subset \mathbb{R}_d[x]$:*

$$V(P) = \{x \in \mathbb{R}^n : p(x) = 0 \textit{ for all } p \in P\}.$$

We say that the matrix $X = \begin{bmatrix} x_1 & x_2 & \cdots & x_s \end{bmatrix} \in \mathbb{R}^{n \times s}$ follows an algebraic variety model if every column of X belongs to the same algebraic variety, i.e., $x_i \in V(P)$ for all $i = 1, \dots, s$.

Let

$$N = \binom{n+d}{d} = \frac{(n+d)!}{d!\, n!}, \tag{2.1}$$

be the dimension of $\mathbb{R}_d[x]$ and consider polynomials $\varphi_d^{(1)}, \dots, \varphi_d^{(N)}$ that form a basis of $\mathbb{R}_d[x]$. We define the *polynomial features* as

$$\varphi_d \colon \mathbb{R}^n \to \mathbb{R}^N \colon \varphi_d(x) = \begin{pmatrix} \varphi_d^{(1)}(x) \\ \varphi_d^{(2)}(x) \\ \vdots \\ \varphi_d^{(N)}(x) \end{pmatrix}. \tag{2.2}$$

For $X \in \mathbb{R}^{n \times s}$, we obtain the matrix of polynomial features $\Phi_d(X)$ by applying φ_d to each column of X,

$$\Phi_d(X) = \begin{bmatrix} \varphi_d(x_1) & \cdots & \varphi_d(x_s) \end{bmatrix} \in \mathbb{R}^{N \times s}.$$

Let the algebraic variety $V(P) \subset \mathbb{R}^n$ be defined by a set of q linearly independent polynomials $P = \{p_1, \dots, p_q\}$, where each polynomial p_i is at most of degree d. Consider the features $\Phi_d(X)$ of a matrix $X \in \mathbb{R}^{n \times s}$ whose columns $x_1, \dots, x_s$ belong to $V(P)$ and assume the following:

A2. *The dimensions of the matrix $X \in \mathbb{R}^{n \times s}$ satisfy $s \geq N - q$.*

Under A2, we have

$$x_i \in V(P) \text{ for all } i = 1, \dots, s \qquad \text{if and only if} \qquad \Phi_d(X)^\top U = 0, \tag{2.3}$$

where the q columns of $U \in \mathbb{R}^{N \times q}$ contain the coefficients of the polynomials $p_1, \dots, p_q$ in the basis φ_d. In this way, we view any matrix $U \in \mathbb{R}^{N \times q}$ as representing q polynomials in the basis φ_d. The zero set of these polynomials is an algebraic variety, and hence we associate the matrix $U \in \mathbb{R}^{N \times q}$ with that algebraic variety.

This algebraic variety is only a function of U through $\operatorname{range}(U)$. Take $U_2 \in \mathbb{R}^{N \times q}$ such that $\operatorname{range}(U_2) = \operatorname{range}(U)$ (there exists $W \in \mathbb{R}^{q \times q}$ such that $U_2 = UW$). From (2.3), the algebraic variety defined by U_2 is also $V(P)$. Thus, the algebraic variety $V(P)$ depends on the q-dimensional subspace $\mathcal{U} := \operatorname{range}(U) \subset \mathbb{R}^N$ but does not depend on the specific matrix U chosen as a basis for that subspace. Hence, it makes sense to use the notation $V_{\mathcal{U}}$ for the algebraic variety $V(P)$.

The space of all q-dimensional subspaces in $\mathbb{R}^N$ is called the Grassmann manifold—written $\operatorname{Grass}(N, q)$— and is a smooth Riemannian manifold. Any $\mathcal{U} \in \operatorname{Grass}(N, q)$ therefore defines a unique algebraic variety. A point $\mathcal{U} \in \operatorname{Grass}(N, q)$ is represented on a computer by a matrix $U \in \mathbb{R}^{N \times q}$ such that $\operatorname{range}(U) = \mathcal{U}$.

3 Approximation by an algebraic variety (denoising)

Consider a point cloud $M \in \mathbb{R}^{n\times s}$ and its corrupted version

$$\hat{M} = M + \omega, \tag{3.1}$$

where the columns of M belong to an algebraic variety $V(P)$ (Definition 2.1) and the noise $\omega \in \mathbb{R}^{n\times s}$ has small Frobenius norm compared to M: $\|\omega\|_{\mathrm{F}} \ll \|M\|_{\mathrm{F}}$. The goal of denoising is to recover M from $\hat{M}$ as accurately as possible, without knowing the polynomials that define $V(P)$ (Figure 3). The distribution of the noisy perturbation ω may be unknown. In computing M, we want to find the algebraic variety that best approximates the point cloud $\hat{M}$. The noise ω may be interpreted as the model mismatch between the raw data (e.g medical scans) and an algebraic variety.

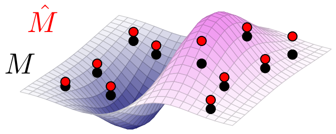


Figure 3: Denoising of an algebraic variety

3.1 Formulation as an optimization problem

We start with an estimate of q, the number of linearly independent polynomials that define the algebraic variety $V(P)$. In order to estimate $V(P)$, we use that every subspace $\mathcal{U} \in \mathrm{Grass}(N,q)$ defines an algebraic variety $V_{\mathcal{U}}$. From (2.3), the subspace which defines the algebraic variety $V(P)$ belongs to the null space of $\Phi(M)^\top$. It follows from simple linear algebra that null $\left(\Phi(M)^\top\right) = (\mathrm{range}\,\Phi(M))^\perp$. So we aim to find a subspace orthogonal to range $\Phi(M)$, that is, a subspace $\mathcal{U}$ such that $\mathrm{P}_{\mathcal{U}}\Phi_d(M) = 0$, where $\mathrm{P}_{\mathcal{U}} = UU^\top \in \mathbb{R}^{N\times N}$ denotes the orthogonal projection on the subspace $\mathcal{U}$ with $\mathrm{range}(U) = \mathcal{U}$ and $U^\top U = \mathrm{I}_N$. Let $\{p_1, \dots, p_q\}$ be the polynomials that define the algebraic variety $V_{\mathcal{U}}$, whose coefficients are given by $U = \begin{bmatrix} u_1 & u_2 & \cdots & u_q \end{bmatrix} \in \mathbb{R}^{N\times q}$ in the basis φ_d. For $X \in \mathbb{R}^{n\times s}$, it follows

$$\|\mathrm{P}_{\mathcal{U}}\Phi_d(X)\|_{\mathrm{F}}^2 = \left\|U^\top\Phi_d(X)\right\|_{\mathrm{F}}^2 = \sum_{i=1}^{s}\sum_{j=1}^{q}((u_j)^\top\varphi_d(x_i))^2 = \sum_{i=1}^{s}\sum_{j=1}^{q} p_j(x_i)^2. \tag{3.2}$$

We minimize (3.2) under the constraint that the variable X is not "far away" from the noisy input $\hat{M}$. Given some $\eta > 0$, an estimate of the noise level, we

$$\underset{\substack{\mathcal{U}\in\mathrm{Grass}(N,q)\\ X\in\mathbb{R}^{n\times s}}}{\text{minimize}} \; \|\mathrm{P}_{\mathcal{U}}\Phi_d(X)\|_{\mathrm{F}}^2 \qquad \text{subject to} \qquad \left\|X - \hat{M}\right\|_{\mathrm{F}}^2 \leq \eta. \tag{3.3}$$

In (3.3), the cost function is nonconvex but smooth. It expresses the least square error of the polynomial system that defines the algebraic variety $V_{\mathcal{U}}$, calculated for every column of X. The above formulation can be made unconstrained in the variable X; for some $\lambda \geq 0$,

$$\underset{\substack{\mathcal{U}\in\mathrm{Grass}(N,q)\\ X\in\mathbb{R}^{n\times s}}}{\text{minimize}} \; \|\mathrm{P}_{\mathcal{U}}\Phi_d(X)\|_{\mathrm{F}}^2 + \lambda\left\|X - \hat{M}\right\|_{\mathrm{F}}^2. \tag{3.4}$$

It can be shown that there exists a $\lambda \geq 0$ such that (3.3) and (3.4) are equivalent. Through the lens of Riemannian optimization, Problem (3.4) can be viewed as the unconstrained minimization of a function defined on the manifold $\mathcal{M} = \mathbb{R}^{N\times s} \times \mathrm{Grass}(N, q)$. This general framework allows to use off-the-shelf Riemannian optimization methods (Boumal, 2023).

Problem (3.4) is nonconvex but there are good initial guesses for the variables X and $\mathcal{U}$, which are respectively $\hat{M}$ and the last q (left) singular vectors of $\Phi_d(\hat{M})$. In Section 3.3, we observe numerically that solving (3.4) with second-order Riemannian optimization methods is robust and appears to find the global minimum despite the nonconvexity. In order to solve (3.4), we do not assume knowledge of the noise level η. To find a suitable value for the regularization parameter λ, the problem is first solved with $\lambda = 10^{-6}$ (a small value) and then solved again several times with increasing values of λ. Each problem is initialized with the solution of the previous solve (warm starting). This process is stopped when a good trade-off between the values of $\|\mathrm{P}_{\mathcal{U}}\Phi_d(X)\|_{\mathrm{F}}$ and $\left\|X - \hat{M}\right\|_{\mathrm{F}}$ is reached.

When the residual vanishes, the points belong exactly to an algebraic variety. Equation (2.3) implies that

$$\|\mathrm{P}_{\mathcal{U}}\Phi_d(X)\|_{\mathrm{F}} = 0 \qquad \text{if and only if} \qquad x_i \in V_{\mathcal{U}} \text{ for all } i = 1, \ldots, s \tag{3.5}$$

where $x_1, \ldots, x_s$ denote the columns of $X \in \mathbb{R}^{n\times s}$. In that case, the last q singular values of $\Phi_d(X)$ are zero and $V_{\mathcal{U}}$ is defined by the subspace corresponding to the last q singular vectors $\Phi_d(X)$, which may be computed by a singular value decomposition. Recovering the algebraic variety of the data—despite the presence of noise—is the first part of the registration process in Section 4, where we estimate a transformation such that two algebraic varieties overlap.

An important practical concern is to estimate q, the number of linearly independent polynomials which define the algebraic variety. For hypersurfaces of dimension $n - 1$, one polynomial equation defines the surface and $q = 1$. In the section on numerical results 3.3, we discuss the choice of a suitable degree d for the model. The next section provides an estimate of the accuracy of the recovery in the case of Gaussian noise, which gives an estimate of the error at a solution which is obtained for a given degree.

3.2 Statistical error estimation

In this section, we use Stein's unbiased risk estimate (Stein, 1981) to give guarantees on the error of our denoising procedure. Assume that the noise perturbation ω in (3.1) follows a normal distribution. That is, for some $\sigma > 0$, we have $\omega_{ij} \sim \mathcal{N}(0, \sigma^2)$ for every entry (i, j) of ω. For a given $\hat{M}$, let $\left(X^*(\hat{M}), \mathcal{U}^*(\hat{M})\right)$ be an isolated local minimizer of (3.4). We view $X^*(\hat{M})$ as an estimator of M which depends on the random variable ω. The expected error is defined as $R = \mathbb{E}_\omega\left[\left\|M - X^*(\hat{M})\right\|_{\mathrm{F}}^2\right]$. Based on (Stein, 1981), we have

$$\hat{R} = \left\|\hat{M} - X^*(\hat{M})\right\|_{\mathrm{F}}^2 - ns\sigma^2 + 2\sigma^2 \sum_{i=1}^{n}\sum_{j=1}^{s} \frac{\partial X^*_{ij}(\hat{M})}{\partial \hat{M}_{ij}}, \tag{3.6}$$

as an unbiased estimate for R, i.e. $\mathbb{E}(\hat{R}) = R$. The estimate $\hat{R}$ is called *Stein's unbiased risk estimate* (SURE). The quantity $\sum_{i=1}^{n}\sum_{j=1}^{s} \partial X^*_{ij}(\hat{M})/\partial \hat{M}_{ij}$ is known as the *divergence* of the estimator $X^*(\hat{M})$ and describes the sensitivity of the solution to the input data. It measures the complexity of the model, that is, its tendency to overfit the data (Ghojogh and Crowley, 2019, Eq. 33).

We propose to use SURE as an alternative to cross validation to evaluate the estimation error and detect the possibility of overfitting. For polynomials of degree 2, which we use in the numerical examples, the model is simple and the likelihood of overfitting is small. Estimating the generalization error with formula (3.6) is most useful for several values of model parameters like the degree d.

To compute the divergence in the context of Problem (3.4), we use a version of the implicit function theorem for functions defined on manifolds. For a smooth map $F\colon \mathcal{M}_1 \times \mathcal{M}_2 \to \mathcal{M}_3$, we write $\mathrm{D}_1 F$ and $\mathrm{D}_2 F$ for the differential of F (Boumal, 2023, Definition 3.34), with respect to its first and second argument, respectively.

Theorem 3.1 ((Abraham et al., 2012), Prop. 3.3.13)**.** *Let $\mathcal{M}_1, \mathcal{M}_2, \mathcal{M}_3$ be manifolds. Let $F\colon \mathcal{M}_1 \times \mathcal{M}_2 \to \mathcal{M}_3$ be smooth and let $(x_0, y_0) \in \mathcal{M}_1 \times \mathcal{M}_2$ with $F(x_0, y_0) = 0$. If $\mathrm{D}_2 F(x_0, y_0)$ is invertible, there exists open neighbourhoods V_1 of x_0 in $\mathcal{M}_1$ and V_2 of y_0 in $\mathcal{M}_2$, and a smooth function $g : V_1 \to V_2$ such that for all $x \in V_1$, $F(x, g(x)) = 0$. In addition,*

$$\mathrm{D}g(x_0) = -\left(\mathrm{D}_2 F(x_0, y_0)\right)^{-1} \mathrm{D}_1 F(x_0, y_0).$$

Recall that the root mean square error (RMSE) is defined by

$$\mathrm{RMSE} = \|M - X^*\|_{\mathrm{F}} / \sqrt{ns}. \tag{3.7}$$

Proposition 3.2. *Let the noise $\omega \sim \mathcal{N}(0, \sigma^2)$ for some given $\sigma \geq 0$ and let $f(X,\mathcal{U}; \hat{M})$ denote the cost function of (3.4). Consider $\left(X^*(\hat{M}),\mathcal{U}^*(\hat{M})\right)$, an isolated local minimizer of f for a given $\hat{M}$. Stein's Unbiased risk estimate of the RMSE is given by*

$$\mathrm{SURE} := \sqrt{\hat{R}/ns} \tag{3.8}$$

where $\hat{R}$ is computed as in (3.6) with

$$\frac{\partial X^*(\hat{M})}{\partial \hat{M}} = -\left[\nabla^2_{XX} f(X,\mathcal{U}; \hat{M})\right]^{-1} \frac{\partial}{\partial \hat{M}} \left(\nabla_X f(\hat{M}, g(\hat{M}); \hat{M})\right). \tag{3.9}$$

where

$$\frac{\partial}{\partial \hat{M}_{ij}} \left(\nabla_X f(\hat{M}, g(\hat{M}); \hat{M})\right) = -2\lambda \Delta_{ij}$$

and $\Delta_{ij} \in \mathbb{R}^{n \times s}$ is zero except for entry (i, j) equal to 1.

Proof. For clarity, we write $(X^*,\mathcal{U}^*)$ for $\left(X^*(\hat{M}),\mathcal{U}^*(\hat{M})\right)$. Recalling that $\mathcal{M} = \mathbb{R}^{n\times s} \times \mathrm{Grass}(N, q)$, we define $F\colon \mathbb{R}^{n\times s} \times \mathcal{M} \to \mathrm{T}\mathcal{M}$ by

$$F(\hat{M}, X,\mathcal{U}) = \mathrm{grad}_{\mathcal{M}} f(X,\mathcal{U}; \hat{M}) = \begin{bmatrix} \nabla_X f(X,\mathcal{U}; \hat{M}) \\ \mathrm{grad}_{\mathcal{U}} f(X,\mathcal{U}; \hat{M}) \end{bmatrix}. \tag{3.10}$$

where $\nabla_X f(X,\mathcal{U}; \hat{M}) \in \mathbb{R}^{n\times s}$ is the Euclidean gradient of f with respect to $X \in \mathbb{R}^{n\times s}$ and $\mathrm{grad}_{\mathcal{U}} f(X,\mathcal{U}; \hat{M}) \in \mathrm{T}_{\mathcal{U}}\mathrm{Grass}(N, q)$ is the Riemannian gradient which belongs to the tangent space of $\mathrm{Grass}(N, q)$ at $\mathcal{U}$. First-order necessary optimality conditions for Problem (3.4) can be stated as

$$F(\hat{M}, X,\mathcal{U}) = 0.$$

We apply Theorem 3.1 to the function F, with the identification of the variables $\hat{M} \in \mathcal{M}_1 = \mathbb{R}^{n\times s}$ and $z = (X,\mathcal{U}) \in \mathcal{M}_2 = \mathcal{M} = \mathbb{R}^{n\times s} \times \mathrm{Grass}(N,q)$. The theorem applies to derivatives as abstract objects defined by tangent vectors. In particular, affine connections (Boumal, 2023, Definition 5.1)—including the Riemannian connection—satisfy the axioms of derivatives. When applying this theorem to the function F, we view the derivative of a vector field as the covariant derivative associated to the Riemannian connection. Therefore, the derivative of the map F with respect to its second variable $z \in \mathcal{M}$, is given by the Riemannian Hessian of f, that is, $\mathrm{D}_2 F(\hat{M}, X, \mathcal{U}) = \mathrm{Hess}_{\mathcal{M}} f(X,\mathcal{U};\hat{M})$. Therefore, if $\mathrm{Hess}_{\mathcal{M}} f(X^*,\mathcal{U}^*;\hat{M})$ is invertible, there exists an open set $V_1 \in \mathbb{R}^{n\times s}$ and a map $g\colon V_1 \to \mathcal{M}$ such that $g(\hat{M}) = (X^*,\mathcal{U}^*)$ and $F(\hat{M}, g(\hat{M})) = 0$. Also,

$$\frac{\partial g(\hat{M})}{\partial \hat{M}} = \frac{\partial}{\partial \hat{M}} \left((X^*(\hat{M}),\mathcal{U}^*(\hat{M}) \right) = - \left[\mathrm{Hess}_{\mathcal{M}} f(X^*,\mathcal{U}^*;\hat{M}) \right]^{-1} \left[\frac{\partial F(\hat{M}, g(\hat{M}))}{\partial \hat{M}} \right]. \tag{3.11}$$

If $(X^*,\mathcal{U}^*)$ is an isolated minimizer of (3.4), the matrix $\mathrm{Hess} f(X^*,\mathcal{U}^*)$ is positive definite and therefore invertible. In order to compute $\partial F(\hat{M}, g(\hat{M}))/\partial \hat{M}$, we see from the definition of $f(X,U;\hat{M})$ in (3.4) that $\mathrm{grad}_{\mathcal{U}} f(X,U;\hat{M})$ does not depend on $\hat{M}$ and that $\nabla_X f(\hat{M}, g(\hat{M}); \hat{M})$ depends on $\hat{M}$ through the term $2\lambda(X - \hat{M})$. Thus, we have

$$\frac{\partial}{\partial \hat{M}} \left(\mathrm{grad}_{\mathcal{U}} f(\hat{M}, g(\hat{M}); \hat{M}) \right) = 0 \tag{3.12}$$

and

$$\frac{\partial}{\partial \hat{M}_{ij}} \left(\nabla_X f(\hat{M}, g(\hat{M}); \hat{M}) \right) = -2\lambda \Delta_{ij}$$

with where $\Delta_{ij} \in \mathbb{R}^{n\times s}$ has entry ij equal to one and the other entries are zero. Using (3.12), the upper block of (3.11) reads

$$\frac{\partial X^*(\hat{M})}{\partial \hat{M}} = - \left[\nabla^2_{XX} f(X,\mathcal{U};\hat{M}) \right]^{-1} \frac{\partial}{\partial \hat{M}} \left(\nabla_X f(\hat{M}, g(\hat{M}); \hat{M}) \right).$$

This gives (3.9). Equation (3.8) follows from (3.6) and the definition of the RMSE. □

3.3 Numerical results for denoising of algebraic varieties

In this section, we numerically evaluate the performance of our denoising method. Our code for the denoising of algebraic varieties can be found online at https://github.com/flgoyens/variety-denoising, currently with implementations in Matlab and Python. We use the Manopt (Boumal et al., 2014) and Pymanopt (Townsend et al., 2016) toolboxes for the Matlab and Python versions of the code, respectively.

The Riemannian trust-region solver (Boumal, 2023, Algorithm 6.3) in the toolbox is our algorithm of choice; it is applied to Problem (3.4) with default parameters. The trust-region algorithm underlying this solver has been shown to have fast local rate of convergence (at least superlinear, near a non-degenerate minimizer (Absil et al., 2007)). Furthermore, under suitable global smoothness assumptions, this algorithm is guaranteed to converge to a first- or second-order critical point of the problem from an arbitrary initial guess (Boumal et al., 2019), with a rate that can be quantified as follows: for a given accuracy parameter ε, it takes at most $\mathcal{O}(\varepsilon^{-2})$ to generate an ε-approximate first-order critical point, while an ε-approximate second-order critical point is found within most $\mathcal{O}(\varepsilon^{-3})$ iterations.

The numerical examples in this section were generated using the Matlab version of our code, with the termination condition that the norm of the gradient in (3.4) becomes smaller than 10^{-6}. For an output of the algorithm $(X^*, \mathcal{U}^*)$, we define

$$\text{RESIDUAL} = \|\mathrm{P}_{\mathcal{U}^*}\Phi_d(X^*)\|_{\mathrm{F}}^2 ,$$

which is interpreted using (3.2) as the least squares error of the system of polynomial equations that define the algebraic variety $V_{\mathcal{U}^*}$. The RESIDUAL tells us how close the columns of X^* are to some algebraic variety. The RMSE tells us how close X^* is to the original matrix M (3.7). We begin with randomly generated examples. We use a monomial basis of degree $d = 2$ for the feature map φ_d.

Synthetic examples

Example 1: Denoising a circle We generate a point cloud $\hat{M} \in \mathbb{R}^{2\times 150}$ as a noisy corruption of a set of points on the unit circle using $\sigma = \{10^{-3}, 10^{-2}, 10^{-1}, 2\cdot 10^{-1}\}$ for the standard deviation of the Gaussian noise ($\hat{M}$ in red in Figure 4). We notice that the solution X^* (blue) reached by the optimization algorithm is close to the original circle even for a visually large noise. For $\sigma = 10^{-2}$ and $\sigma = 2\cdot 10^{-1}$, the RESIDUAL reaches values of 10^{-9} and 10^{-5}, respectively. The solver's output X^* is therefore very well approximated by an algebraic variety. This example also makes it clear that the approach is more general than a classical polynomial approximation, as the approximation does not need to be the image of a polynomial function. Table 1 reports the RMSE (3.7) and SURE estimate of the RMSE (3.8). SURE accurately predicts the RMSE and that the RMSE increases proportionally with the standard deviation of the noise.

Noise σ	10^{-3}	10^{-2}	10^{-1}	$2\cdot 10^{-1}$
SURE	$5.49\cdot 10^{-4}$	$6.76\cdot 10^{-3}$	$5.69\cdot 10^{-2}$	$1.22\cdot 10^{-1}$
RMSE	$6.77\cdot 10^{-4}$	$8.53\cdot 10^{-3}$	$7.52\cdot 10^{-2}$	$1.44\cdot 10^{-1}$

Table 1: SURE (3.8) and RMSE (3.7) for denoising of a circle

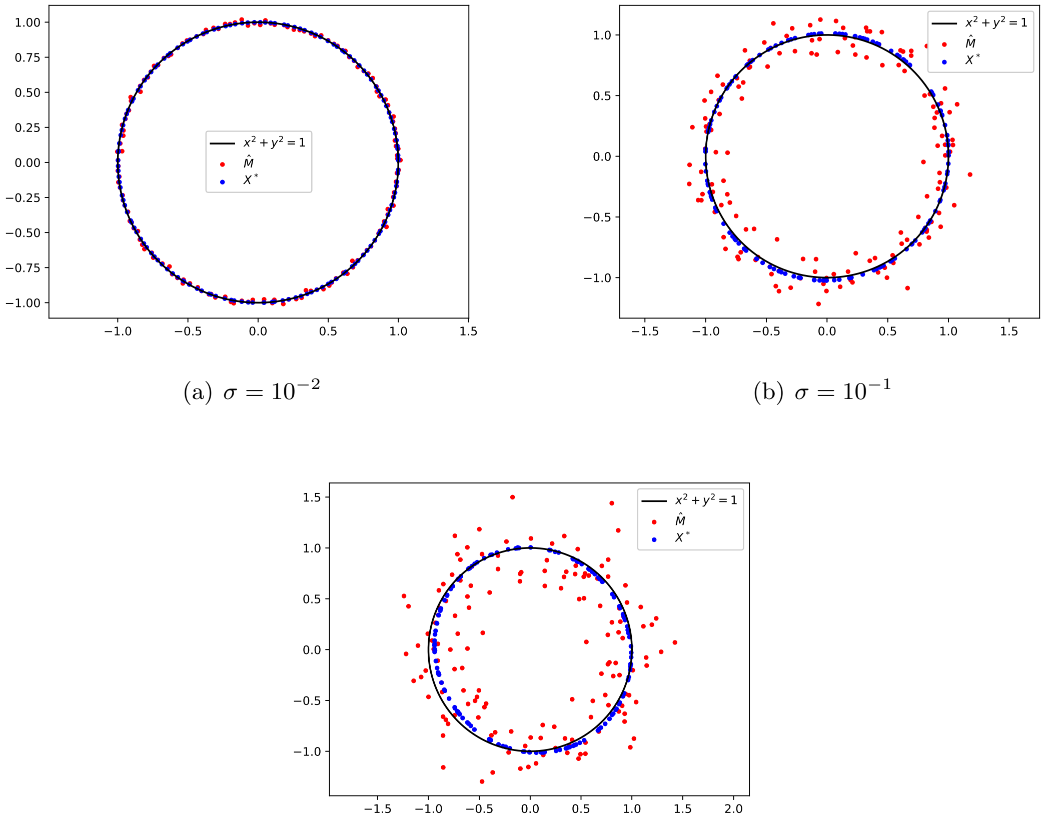
Example 2: Denoising a union of two subspaces In Figure 5, we denoise a point cloud which is near the union of two subspaces with $\sigma = \{10^{-3}, 10^{-2}, 10^{-1}\}$ for the standard deviation of the noise. This shows that the algebraic variety doesn't need to be a smooth set. It is an algebraic variety described by polynomial equations of degree 2, hence using polynomials of degree 2 is sufficient. (The union of k subspaces is an algebraic variety described by polynomial equations of degree k.) For $\sigma = 10^{-2}$, the output satisfies RESIDUAL $= 3\cdot 10^{-10}$. Table 2 reports the RMSE and the SURE estimate of the RMSE.

Noise σ	10^{-3}	10^{-2}	10^{-1}
SURE	$7.05\cdot 10^{-4}$	$6.32\cdot 10^{-3}$	$7.04\cdot 10^{-2}$
RMSE	$7.10\cdot 10^{-4}$	$6.98\cdot 10^{-3}$	$7.37\cdot 10^{-2}$

Table 2: SURE for denoising of a union of subspaces

Dental Tomography Scans

Our research was originally motivated by an industrial partnership with the National Physical Laboratory (NPL), which concerned 3D dental scans (X-ray computed tomography scans). Each

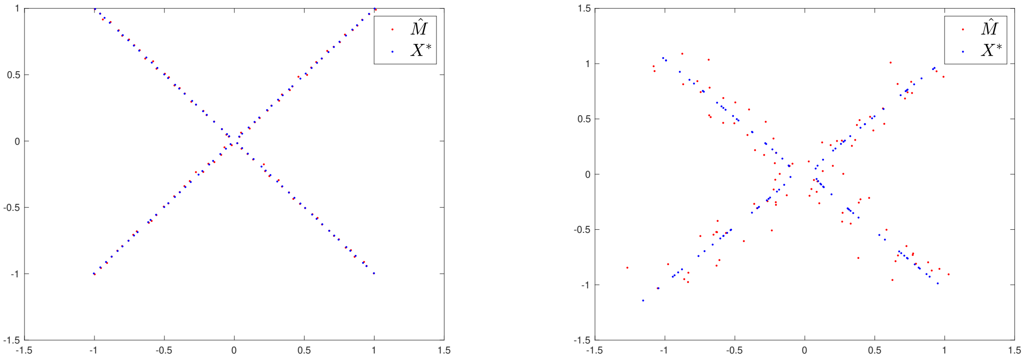


(a) $\sigma = 10^{-2}$

(b) $\sigma = 10^{-1}$

(c) $\sigma = 2 \cdot 10^{-1}$

Figure 4: Denoising a circle.



(a) $\sigma = 10^{-2}$

(b) $\sigma = 10^{-1}$

Figure 5: Denoising a nonsmooth algebraic variety.

scan contains several millions of data points. It is desirable to remove the noise and outliers before they can be processed. Figure 6 shows an example of the dental scans available. After random subsampling—which helps visualization—it is composed of 2048 data points in $\mathbb{R}^3$.

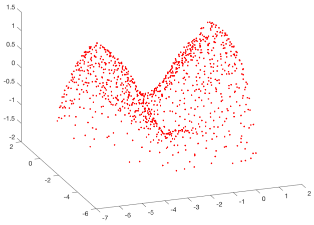


Figure 6: Tomographic dental scans (subsampled)

For data sets such as dental scans, it may not be obvious which degree of the feature map will yield the best result. In theory, increasing the degree should give a model with more degrees of freedom and a better fit. Unfortunately, the dimension of the optimization problem $N \approx n^d/d!$ blows up as d increases, even for moderate values of n. The number of data points necessary also increases with the degree, since $s > N - q$ is required A2. Additionally, ill-conditioning may appear when the degree d becomes large, this is especially true for monomial basis because $\Phi_d(X)$ is a multivariate Vandermonde matrix. Previous works using the monomial kernel for high-rank matrix completion problems restricted themselves to degrees two or three (Ongie et al., 2017; Fan and Chow, 2018; Goyens et al., 2022). We try values of d going from 1 to 5 and choose the one that yields the best result, which is usually $d = 2$ in practice.

Example 3: approximation of a dental scan by an algebraic variety Figure 7 shows an XCT scan in red and the output of our denoising algorithm in blue. The value RESIDUAL has been reduced from the order of 1 to the order of 10^{-3} by the algorithm, and X^* is therefore closer to an algebraic variety than $\hat{M}$. Since no noise was artificially added, the value of σ for the noise that represents the mismatch between the real dental scan and the algebraic variety model is unknown. For $\sigma = 10^{-2}$, Stein's estimation of the RMSE gives SURE $= 0.21$.

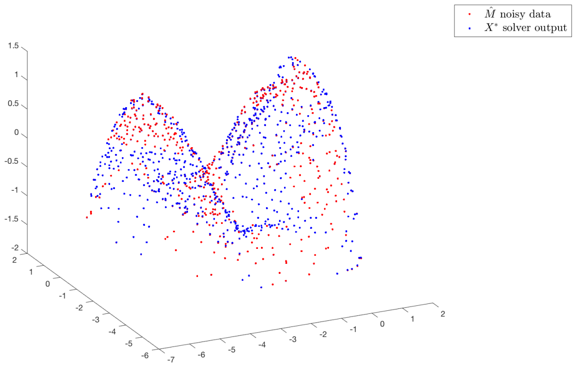


Figure 7: Original point cloud and denoised version

4 Registration of algebraic varieties

We return to the registration problem presented in the introduction. Consider two matrices $M_1 \in \mathbb{R}^{n\times s_1}$ and $M_2 \in \mathbb{R}^{n\times s_2}$ that contain data points belonging to two algebraic varieties V_1 and V_2 that overlap through a rigid transformation (A1). Assume further that the algebraic varieties V_1, V_2 have degree at most d and are described by q linearly independent polynomials, for some $0 < q \leq N$; and that M_1 and M_2 both satisfy A2, so that $\min(s_1, s_2) \geq N - q$. We are given two point clouds $\hat{M}_1$ and $\hat{M}_2$, which are approximate versions of M_1 and M_2:

$$\hat{M}_1 = M_1 + \omega_1$$
$$\hat{M}_2 = M_2 + \omega_2,$$

where ω_1, ω_2 are noisy perturbations, with $\|\omega_1\|_{\mathrm{F}} \ll \|M_1\|_{\mathrm{F}}$ and $\|\omega_2\|_{\mathrm{F}} \ll \|M_2\|_{\mathrm{F}}$. Our goal in this section is to estimate the rigid transformation $\mathcal{T}\colon \mathbb{R}^n \to \mathbb{R}^n$ which makes V_1 and V_2 overlap, or, more precisely, such that $\mathcal{T}(M_1)$ belongs to V_2.

4.1 Formulation as an optimization problem

To achieve this, we first attempt to recover M_1 and M_2 from $\hat{M}_1$ and $\hat{M}_2$, by solving twice the denoising problem (3.4), as described in Section 3. This yields estimates $X_1 \in \mathbb{R}^{n\times s}$ and $X_2 \in \mathbb{R}^{n\times s}$, along with subspaces $\mathcal{U}_1 \in \mathrm{Grass}(N,q)$ and $\mathcal{U}_2 \in \mathrm{Grass}(N,q)$ which define the algebraic varieties that X_1 and X_2 (approximately) belong to. Then, we estimate a transformation $\mathcal{T}$ such that $\mathcal{T}(X_1)$ belongs to the algebraic variety defined by $\mathcal{U}_2$.

Let $x_1^{(1)}, x_2^{(1)}, \cdots x_{s_1}^{(1)}$ denote the columns of X_1 and let $p_1, \dots, p_q$ denote the polynomials that define $V_{\mathcal{U}_2}$. To find a rotation $Q \in \mathrm{SO}(n)$ and a vector $a \in \mathbb{R}^n$ such that the points $\mathcal{T}\left(x_i^{(1)}\right) = Qx_i^{(1)} + a$ belong to $V_{\mathcal{U}_2}$ for all $1 \leq i \leq s_1$, we

$$\underset{Q,a}{\mathrm{minimize}} \sum_{i=1}^{s_1} \sum_{j=1}^{q} p_j(Qx_i^{(1)} + a)^2. \tag{4.1}$$

The distinction with the least squares minimization (1.4) appears clearly. We do not minimize a point-to-point distance; instead, we are fitting points to an algebraic variety and want each column of X_1—after application of the rigid transformation—to satisfy the equations that define $V_{\mathcal{U}_2}$. Using (3.2), we write (4.1) using the variable $\mathcal{U}_2 \in \mathrm{Grass}(N,q)$ and the matrix of polynomial features:

$$\sum_{i=1}^{s_1} \sum_{j=1}^{q} p_j(Qx_i^{(1)} + a)^2 = \sum_{i=1}^{s_1} \left\| \mathrm{P}_{\mathcal{U}_2} \varphi_d(Qx_i^{(1)} + a) \right\|_2^2 = \|\mathrm{P}_{\mathcal{U}_2} \Phi_d(QX_1 + a\mathbf{1}_{1\times s_1})\|_{\mathrm{F}}^2,$$

where $\mathbf{1}_{1\times s_1}$ is a row vector of size s_1 that is full of ones. This leads to the following two-step strategy to identify the transformation between V_1 and V_2.

Step 1: Algebraic variety identification and denoising

$$(X_1, \mathcal{U}_1) := \begin{cases} \underset{X,\mathcal{U}}{\arg\min} & \|\mathrm{P}_{\mathcal{U}} \Phi_d(X)\|_{\mathrm{F}}^2 + \lambda \left\| X - \hat{M}_1 \right\|_{\mathrm{F}}^2 \\ & \mathcal{U} \in \mathrm{Grass}(N,q). \end{cases} \tag{4.2}$$

$$(X_2, \mathcal{U}_2) := \begin{cases} \underset{X,\mathcal{U}}{\arg\min} & \|\mathrm{P}_{\mathcal{U}} \Phi_d(X)\|_{\mathrm{F}}^2 + \lambda \left\| X - \hat{M}_2 \right\|_{\mathrm{F}}^2 \\ & \mathcal{U} \in \mathrm{Grass}(N,q). \end{cases} \tag{4.3}$$

Step 2: Registration

$$(Q^*, a^*) := \begin{cases} \min\limits_{Q,a} & \|\mathrm{P}_{\mathcal{U}_2} \Phi_d \left(QX_1 + a\mathbf{1}_{1\times s_1}\right)\|_{\mathrm{F}}^2 \\ \text{subject to} & Q \in \mathrm{SO}(n) \\ & a \in \mathbb{R}^n. \end{cases} \tag{4.4}$$

For details on solving Step 1, see Section 3. The registration step (4.4) is nonconvex, and it is also a smooth optimization problem on a manifold that we minimize using Riemannian optimization (Boumal, 2023, Algorithm 6.3). We initialize the transformation $\mathcal{T}$ with a random rotation and random vector a of unit norm. If the transformation $\mathcal{T}$ includes a reflection as well as a rotation, the optimization should not be done over $\mathrm{SO}(n)$ but the other connected component of $\mathrm{O}(n)$, with matrices of determinant -1.

Remark 4.1. *Another possibility would be to compute the registration in a single step, an optimization problem over the variables $(\mathcal{U}, Q, a)$ that concatenates both point clouds. We found that this is harder to solve in practice. Performing steps 1 and 2 separately is suitable because we are able to estimate the algebraic variety from the samples of only one of the point clouds.*

Remark 4.2. *Interestingly, this two-step approach is asymmetric. If one point cloud is included in the other, the larger point cloud must be used as the target (M_2) and the smaller one as the source (M_1).*

4.2 Numerical results for registration

Our code for the registration problem is available in Python at `https://github.com/flgoyens/variety-registration`. In the numerical examples that follow, the default implementation of the Riemannian trust-region algorithm from PyManopt was used where the full Riemannian Hessian is computed through automatic differentiation (Townsend et al., 2016). This section illustrates the efficiency of our approach to identify a rigid transformation between two quadratic curves or surfaces (synthetic test problems), or two dental scans. We use the monomial basis of degree $d = 2$ for the feature map φ_d. We set the dimension $q = 1$, since the points clouds are hyper surfaces of dimension $n - 1$.

For the synthetic examples, we take the samples in M_2 as randomly generated points that belong to the quadratic $x \mapsto x^2$. The samples in M_1 come from a rotated and translated version of that curve, for some random rotation and translation that we attempt to recover. We then add random matrices of Gaussian noise $\omega_1, \omega_2 \sim \mathcal{N}(0, \sigma^2)$, which yields $\hat{M}_1$ and $\hat{M}_2$. The quality of the solution returned by the solver is assessed by the residual

$$\mathrm{RESIDUAL} = \|\mathrm{P}_{\mathcal{U}_2} \Phi_d \left(Q^* X_1 + a^* \mathbf{1}_{1\times s_1}\right)\|_{\mathrm{F}}^2 ,$$

which is interpreted using (4.1) as the residual of the polynomial equations that define an algebraic variety. We emphasize that the measure $\|Q^* X_1 + a^* \mathbf{1}_{1\times s_1} - X_2\|_{\mathrm{F}}^2$ is meaningless. We are not trying to perform a point-to-point matching, which is even undefined for point clouds of different sizes. In the figures below, the upper image shows the two point clouds of input, and the lower image shows the output of the algorithm after registration.

These numerical results illustrate that our approach successfully finds an accurate rigid transformation in these simple cases. Due to the nonconvexity of $\mathrm{SO}(n)$ and the random initialisation of the algorithm, the solver sometimes fails to find the correct transformation and appears to find a local minimum. We then restart Problem (4.4) with a new initial rotation and translation, and report that this usually finds the global minimum under five attempts. This is likely helped by the fact that registration problems are most often considered in $\mathbb{R}^2$ or $\mathbb{R}^3$, and the dimension of the search space is small. A better initialization of the transformation would lower the chances to find a suboptimal local minimizer.

Example 5 In Figure 8, the data is on a quadratic surface in $\mathbb{R}^3$ (zero noise added). The input data has dimensions $M_1 \in \mathbb{R}^{3\times 200}$, $M_2 \in \mathbb{R}^{3\times 200}$. The result satisfies RESIDUAL $\approx 10^{-7}$.

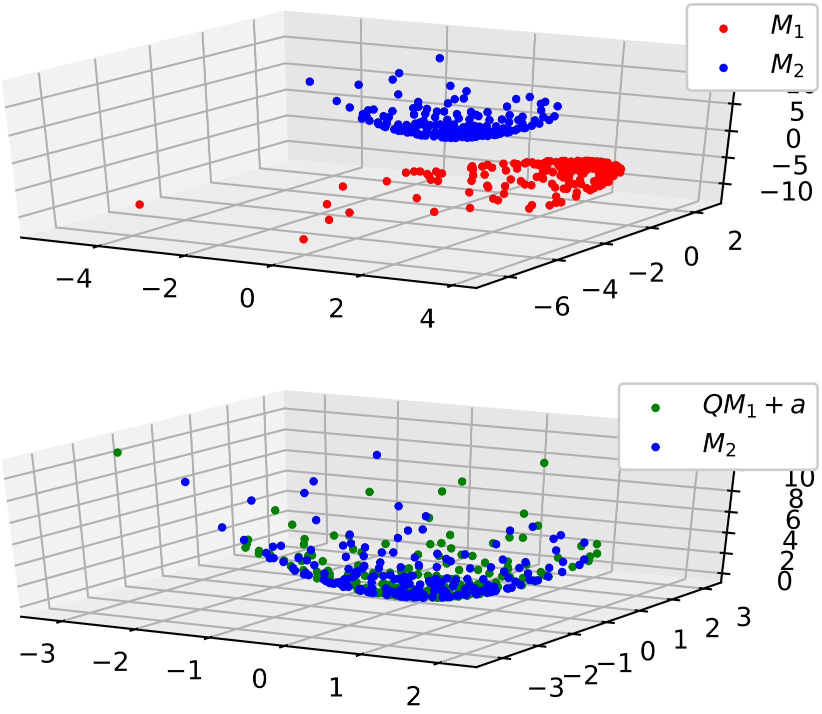


Figure 8: Registration for a quadratic surface in $\mathbb{R}^3$.

Example 6 In Figure 9, the standard deviation of the noise is $\sigma = 5 \cdot 10^{-2}$ and the algorithm's output satisfies RESIDUAL $\approx 10^{-4}$. We see that despite the presence of noise in the original data, the transformation is well estimated to create an overlap.

Example 7 In Figure 10, the magnitude of the noise is increased to $\sigma = 10^{-1}$ and the output satisfies RESIDUAL $\approx 10^{-2}$. The transformation is again estimated correctly, with the residual increasing proportionally to the magnitude of the noise.

Next we show examples where there is a partial overlap between the source and target point clouds.

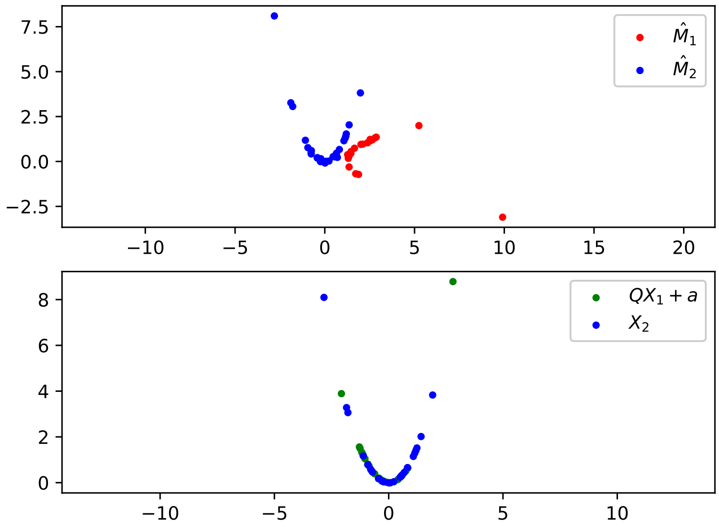


Figure 9: Noisy registration with $\sigma = 5 \cdot 10^{-2}$

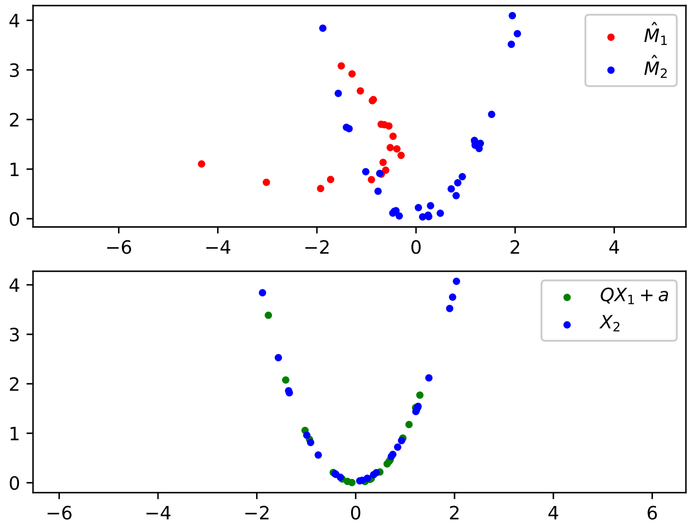


Figure 10: Noisy registration with $\sigma = 10^{-1}$

Example 8: partial overlap In Figures 11 and 12, the point cloud $\hat{M}_1$ only overlaps with a small part of $\hat{M}_2$. The noise levels are of $\sigma = 10^{-2}$ and $\sigma = 10^{-1}$ and give RESIDUAL values of 10^{-4} and 10^{-2}, respectively.

Example 9: no overlap In Figure 13, we push things even further and show that the registration may be possible even in cases where there is no overlap between the point clouds. Our approach makes this possible because the points belong to a common algebraic variety. The noise level is set to $\sigma = 10^{-2}$ and RESIDUAL $\approx 10^{-4}$.

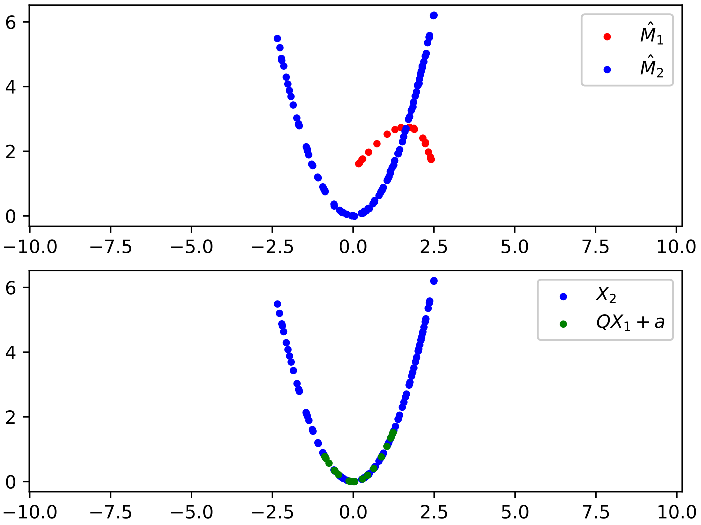


Figure 11: Noisy registration with $\sigma = 10^{-2}$ and partial overlap

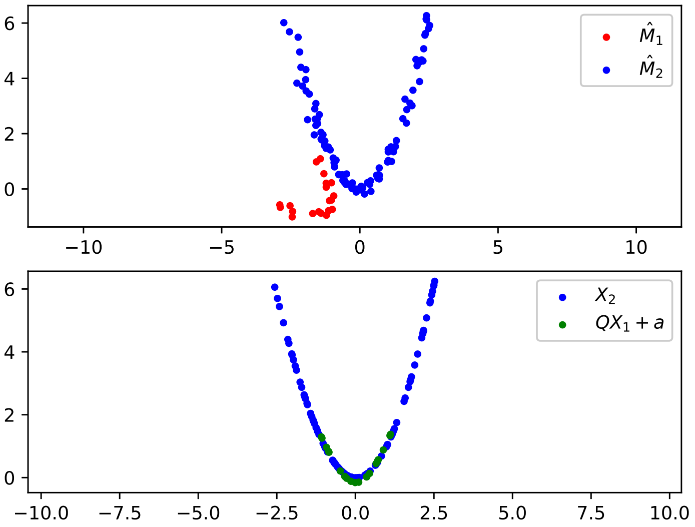


Figure 12: Noisy registration with $\sigma = 10^{-1}$ and partial overlap

Example 10: registration of dental scan We consider a computed tomography (CT) dental scan of dimension 3×2048. The data is naturally noisy and we use the algebraic variety model as an approximation. Nevertheless, we are able to estimate the transformation between two identical versions of the scan with some level of accuracy. The degree which yields the best result for the monomial features is $d = 2$ and the output of the solver gives RESIDUAL $\approx 10^{-1}$.

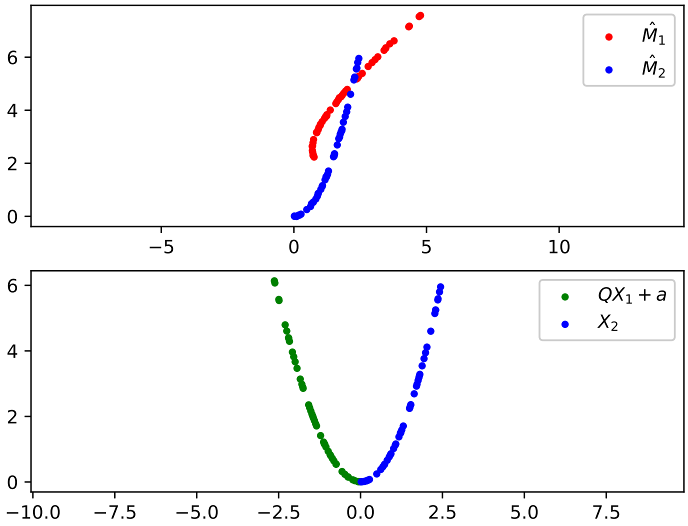


Figure 13: Noisy registration with $\sigma = 10^{-2}$ and no overlap

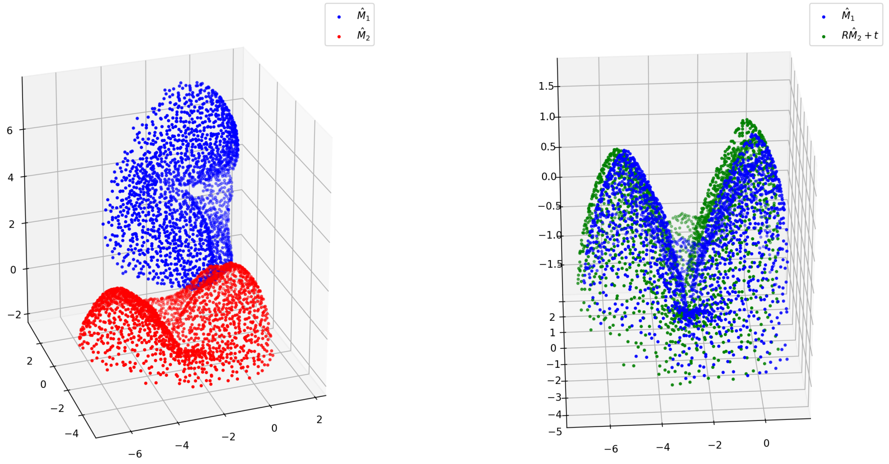


Figure 14: Registration for dental scan.

5 Conclusions

We present a framework which uses polynomial feature maps to approximate point clouds by algebraic varieties and perform point cloud registration based on these approximations. The approach is a conceptually appealing way to perform denoising on algebraic varieties and numerical results show high accuracy in the approximation for various noise levels, with a theoretical estimate of the accuracy using Stein's unbiased risk estimate. We identify the polynomial equations that define the algebraic variety from a set of noisy samples. The use of second-order Riemannian optimization methods allows to achieve high accuracy.

For the registration problem, we show numerical evidence that we are able to overlap point clouds which are approximated by algebraic varieties. We observe some robustness to noise and the method performs even on challenging data such as dental scans. An advantage of our approach over common algorithms for rigid registration is that we do not assume exact point matching and our method achieves good results even in cases of partial overlap between the two point clouds.

Our framework is so far best suited to low-dimensional data sets, because the dimension of the feature space $N \approx n^d/d!$ increases rapidly with n and d; and the number of data points must be large enough to satisfy A2. Several directions should be investigated to improve the scaling of these methods. The dimension of the feature space could be reduced using low-dimensional representations of the features, in the spirit of (Rahimi et al., 2007). To increase the degree—for challenging surfaces—it would be important to use orthogonal polynomial basis to avoid the ill-conditioning of the monomials. Instead of increasing the degree of the approximation, it may be possible to divide the point cloud in patches and apply an approximation by an algebraic variety of degree 2 on each patch.

The scalability of the algorithm with respect to the number of data points s may be improved by viewing the cost functions in (3.4) and (4.4) as a finite sum over the data points and using stochastic optimization methods (Kohler and Lucchi, 2017; Xu et al., 2020).

Acknowledgement The authors would like to thank Wenjuan Sun from the National Physical Laboratory (UK) who suggested the point cloud alignment problem that partially motivated this research and for providing the dental scan data.

References

Abraham, R., Marsden, J. E., and Ratiu, T. (2012). *Manifolds, Tensor Analysis, and Applications*, volume 75. Springer Science & Business Media.

Absil, P.-A., Baker, C., and Gallivan, K. (2007). Trust-Region Methods on Riemannian Manifolds. *Foundations of Computational Mathematics*, 7(3):303–330.

Absil, P.-A., Mahony, R., and Sepulchre, R. (2008). *Optimization Algorithms on Matrix Manifolds*. Princeton University Press.

Arun, K. S., Huang, T. S., and Blostein, S. D. (1987). Least-squares fitting of two 3-D point sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-9(5):698–700.

Bellekens, B., Spruyt, V., Berkvens, R., and Weyn, M. (2014). A survey of rigid 3d pointcloud registration algorithms. In *AMBIENT 2014: The Fourth International Conference on Ambient Computing, Applications, Services and Technologies, August 24-28, 2014, Rome, Italy*, pages 8–13.

Besl, P. J. and McKay, N. D. (1992). Method for registration of 3-D shapes. In *Sensor Fusion IV: Control Paradigms and Data Structures*, volume 1611, pages 586–607. International Society for Optics and Photonics.

Boumal, N. (2023). *An Introduction to Optimization on Smooth Manifolds*. Cambridge University Press.

Boumal, N., Absil, P.-A., and Cartis, C. (2019). Global rates of convergence for nonconvex optimization on manifolds. *IMA Journal of Numerical Analysis*, 39(1):1–33.

Boumal, N., Mishra, B., Absil, P.-A., and Sepulchre, R. (2014). Manopt, a Matlab Toolbox for Optimization on Manifolds. *Journal of Machine Learning Research*, 15:1455–1459.

Breiding, P., Kališnik, S., Sturmfels, B., and Weinstein, M. (2018). Learning algebraic varieties from samples. *Revista Matemática Complutense*, 31(3):545–593.

Chaudhury, K. N., Khoo, Y., and Singer, A. (2015). Global Registration of Multiple Point Clouds Using Semidefinite Programming. *SIAM Journal on Optimization*, 25(1):468–501.

Chen, Y. and Medioni, G. (1992). Object modelling by registration of multiple range images. *Image and vision computing*, 10(3):145–155.

Chui, H. and Rangarajan, A. (2003). A new point matching algorithm for non-rigid registration. *Computer Vision and Image Understanding*, 89(2-3):114–141.

Cox, D., Little, J., O'Shea, D., and Sweedler, M. (1994). Ideals, varieties, and algorithms. *American Mathematical Monthly*, 101(6):582–586.

Cucuringu, M., Lipman, Y., and Singer, A. (2012a). Sensor network localization by eigenvector synchronization over the Euclidean group. *ACM Transactions on Sensor Networks (TOSN)*, 8(3):1–42.

Cucuringu, M., Singer, A., and Cowburn, D. (2012b). Eigenvector synchronization, graph rigidity and the molecule problem. *Information and Inference: A Journal of the IMA*, 1(1):21–67.

Fan, J. and Chow, T. W. (2018). Non-linear matrix completion. *Pattern Recognition*, 77:378–394.

Ghojogh, B. and Crowley, M. (2019). The Theory Behind Overfitting, Cross Validation, Regularization, Bagging, and Boosting: Tutorial.

Gold, S., Rangarajan, A., Lu, C.-P., Pappu, S., and Mjolsness, E. (1998). New algorithms for 2D and 3D point matching: Pose estimation and correspondence. *Pattern recognition*, 31(8):1019–1031.

Goyens, F., Cartis, C., and Eftekhari, A. (2022). Nonlinear matrix recovery using optimization on the Grassmann manifold. *Applied and Computational Harmonic Analysis*.

Kohler, J. M. and Lucchi, A. (2017). Sub-sampled cubic regularization for non-convex optimization. In *International Conference on Machine Learning*, pages 1895–1904. PMLR.

Krishnan, S., Lee, P. Y., Moore, J. B., Venkatasubramanian, S., et al. (2005). Global Registration of Multiple 3D Point Sets via Optimization-on-a-Manifold. In *Symposium on Geometry Processing*, pages 187–196.

Myronenko, A. and Song, X. (2010). Point set registration: Coherent point drift. *IEEE transactions on pattern analysis and machine intelligence*, 32(12):2262–2275.

Ongie, G., Willett, R., Nowak, R. D., and Balzano, L. (2017). Algebraic Variety Models for High-Rank Matrix Completion. In Precup, D. and Teh, Y. W., editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 2691–2700, International Convention Centre, Sydney, Australia. PMLR.

Pottmann, H., Leopoldseder, S., and Hofer, M. (2004). Registration without ICP. *Computer Vision and Image Understanding*, 95(1):54–71.

Rahimi, A., Recht, B., et al. (2007). Random Features for Large-Scale Kernel Machines. In *NIPS*, volume 3, page 5. Citeseer.

Rusinkiewicz, S. and Levoy, M. (2001). Efficient variants of the ICP algorithm. In *Proceedings Third International Conference on 3-D Digital Imaging and Modeling*, pages 145–152. IEEE.

Schönemann, P. H. (1966). A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31(1):1–10.

Segal, A., Haehnel, D., and Thrun, S. (2009). Generalized-icp. In *Robotics: Science and Systems*, volume 2, page 435. Seattle, WA.

Sharp, G. C., Lee, S. W., and Wehe, D. K. (2004). Multiview registration of 3D scenes by minimizing error between coordinate frames. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(8):1037–1050.

Stein, C. M. (1981). Estimation of the mean of a multivariate normal distribution. *The annals of Statistics*, pages 1135–1151.

Townsend, J., Koep, N., and Weichwald, S. (2016). Pymanopt: A python toolbox for optimization on manifolds using automatic differentiation. *arXiv preprint arXiv:1603.03236*.

Tsin, Y. and Kanade, T. (2004). A correlation-based approach to robust point set registration. In *European Conference on Computer Vision*, pages 558–569. Springer.

Xu, P., Roosta, F., and Mahoney, M. W. (2020). Newton-type methods for non-convex optimization under inexact hessian information. *Mathematical Programming*, 184(1):35–70.