

# 3D Localization of Quadratic Surfaces via Optimization on a Manifold

Pei Yean Lee, John B Moore  
National ICT Australia Ltd., Australia.  
Australian National University, Australia.  
Email: peiyea@syseng.anu.edu.au, john.moore@anu.edu.au

## Abstract

Our task is to locate surfaces from noisy laser scanning data, this being important in machine vision and industrial robotics. our focus is the location of quadratic surfaces which are of special importance since they are commonly occurring shapes in man made objects, or can be used in patches to approximate other shapes. We seek to achieve insensitive to noise and occlusion, and achieve quadratic convergence rates. We address the task as minimizing a smooth cost function over the constraint manifold of the Special Euclidean group  $SE_3$ . This leads us to a novel geometric optimization framework which combines ideas from optimization theory and differential geometry. The proposed method involves successive parameterization of the constraint manifold, carrying out optimization of the local cost function in parameter space and subsequently projecting the optimal vector back to the manifold. Newton-like algorithm on manifolds based on this method are devised. A rigorous proof establishes the local quadratic convergence of the algorithm. Simulation results demonstrate the robustness of the method against measurement noise and occlusions.

## I. INTRODUCTION

Quadratic surfaces, also known as quadrics, are commonly occurring shapes in man made objects. Accurate and fast localization of quadrics or surfaces consisting of quadric patches from measurement data is important in many industrial robotics and machine vision tasks.

Two main strategies towards solving this problem in the literatures are the feature based method and the model based method. The feature based method is based on the geometric relation between a set of 3D feature correspondences extracted from the actual surface data and the surface model stored in the database. This approach has been well studied in the literature, [1], [7], [9].

The model based approach, which we adopt in this paper, minimizes the error between the data measured on an actual surface and the CAD model of that surface, see [3], [5], [8], [11], [2], [4], [6]. As opposed to the feature based method, this approach does not require 3D data preprocessing such as feature extraction and explicit correspondences. In the simplest setting, the only information required from the database are the quadratic surface parameters.

In this paper, building on the work in an earlier conference paper [10], we propose a new geometric approach based on locally convergent Gauss-Newton-type iterations on the constraint manifold. Our optimization involves iterative cost function reduction on the smooth manifold of the Special Euclidean Group,  $SE_3$ . To achieve this, some analysis of the underlying geometric constraint is required for algorithm construction and quadratic convergence proof.

The proposed algorithm has the following features: There is local convergence at a quadratic rate. There is robustness to additive Gaussian noise and occlusion. The algorithm works for arbitrary rotations and translations and the algorithm iteration is data independent. The algorithm may not give as accurate pose estimates as working with more sophisticated cost indices based on the sum of minimum distances from each data point to the quadratic surface, since such estimates are typically less biased by the noise. Even so, working with simpler indices as here is of interest for fast processing.

In Section II, the task of locating quadratic surfaces from 3D measurement data is formulated as an optimization problem cast on the Special Euclidean group. A new parameterization-based geometric algorithm

This work was partially supported by ARC Grants A00105829, DP0450539 and the National ICT Australia which is funded by the Australian Department of Communications, Information Technology and the Arts and the Australian Research Council through Backing Australia's Ability and the ICT Centre of Excellence Program. Address: Locked Bag 8001, Canberra ACT 2601, Australia. Email: peiyea@syseng.anu.edu.au, john.moore@anu.edu.au

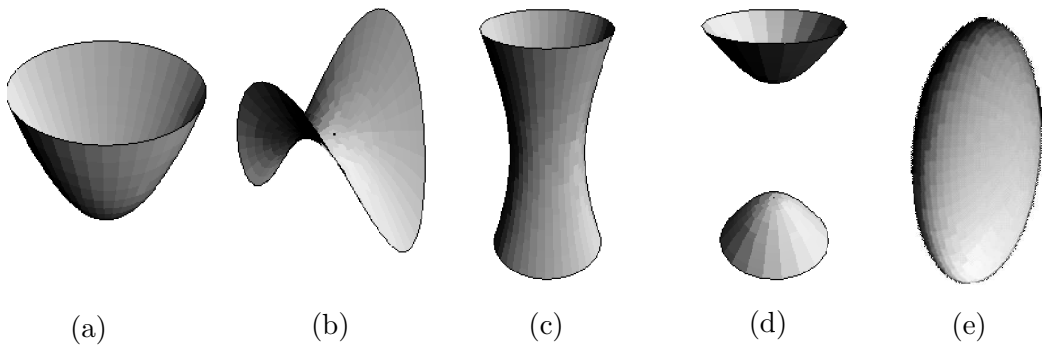


Fig. 1. Examples of quadratic surface: (a) Elliptic Paraboloid, (b) Hyperbolic Paraboloid, (c) Elliptic Hyperboloid of One Sheet, (d) Elliptic Hyperboloid of Two Sheets, (e) Ellipsoid.

to address the problem is presented in Section III with convergence results in Section IV. Initializations of the algorithm are proposed in Section V, and preliminary cyclic coordinate descent iterations are discussed in Section VI. Implementation of the algorithm is outlined in Section VII. A series of simulations are presented in Section VIII, followed by a conclusion in Section IX. Appendix A details the quadratic convergence proof and Appendix B presents the alternative cyclic coordinate descent algorithm.

## II. PROBLEM FORMULATION

### A. Quadratic Surfaces

Quadratic surfaces are defined by the zero set of degree 2 polynomials in 3 variables, as  $\{\tilde{m} \in \mathbb{R}^3 \mid \tilde{m}^\top Q_{11} \tilde{m} + 2\tilde{m}^\top Q_{12} + Q_{22} = 0\}$ . Equivalently, using homogeneous coordinates, a quadratic surface is given by

$$m^\top Q m = 0, \quad m := \begin{bmatrix} \tilde{m} \\ 1 \end{bmatrix}, \quad Q := \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{12}^\top & Q_{22} \end{bmatrix}, \quad (1)$$

where  $Q$  is the symmetric surface coefficient matrix. Without loss of generality, we take  $\text{tr}(Q_{11}^\top Q_{11}) = 1$ . Examples of quadrics are illustrated in Fig. 1.

Now, consider the quadric being rotated by a matrix  $R \in SO_3$ , that is  $R^\top R = I$  and  $\det R = 1$ , and translated by a vector  $t \in \mathbb{R}^3$ . Each point on the transformed quadric  $\tilde{p} \in \mathbb{R}^3$  is given by,

$$\left\{ p = \begin{bmatrix} \tilde{p} \\ 1 \end{bmatrix} \in \mathbb{R}^4 \mid p^\top A(R, t) p = 0 \right\}. \quad (2)$$

Here  $A(R, t) := T(R, t)^\top Q T(R, t)$  is the surface coefficient of the transformed quadric, and

$$T(R, t) := \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \in SE_3. \quad (3)$$

is the 3D rigid body transformation matrix, and  $SE_3$  is the Euclidean group,

$$SE_3 := \{(R, t) \mid R \in SO_3, t \in \mathbb{R}^3\} = SO_3 \times \mathbb{R}^3.$$

### B. Cost Function on the Manifold $SE_3$

Given surface measurement data  $p_i \in \mathbb{R}^4$  and known surface coefficient  $Q$ , the task is to find the transformation matrix  $T(R, t) \in SE_3$  satisfying (2). We work with the cost function that penalizes the algebraic distance of the measurement data to the quadric,

$$f : SE_3 \rightarrow \mathbb{R}, \quad (4)$$

$$f(T) = \frac{1}{n} \sum_{i=1}^n (p_i^\top T^\top Q T p_i)^2 = \frac{1}{n} \sum_{i=1}^n \text{tr}(p_i p_i^\top T^\top Q T)^2.$$

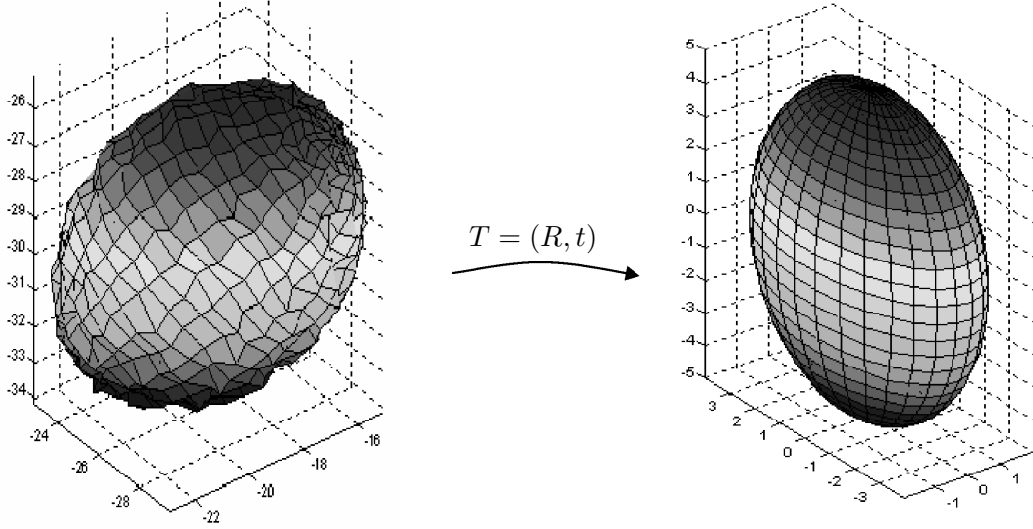


Fig. 2. Given noisy measurement data of a quadric (ellipsoid) and its CAD model, find the position and orientation of the transformed quadric.

Exploiting the relationship between the trace and vec operators, (4) can be re-expressed as,

$$f(T) = \|B \text{vec}(T^\top Q T)\|^2, \quad B := \frac{1}{\sqrt{n}} \begin{bmatrix} \text{vec}^\top(p_1 p_1^\top) \\ \vdots \\ \text{vec}^\top(p_n p_n^\top) \end{bmatrix}. \quad (5)$$

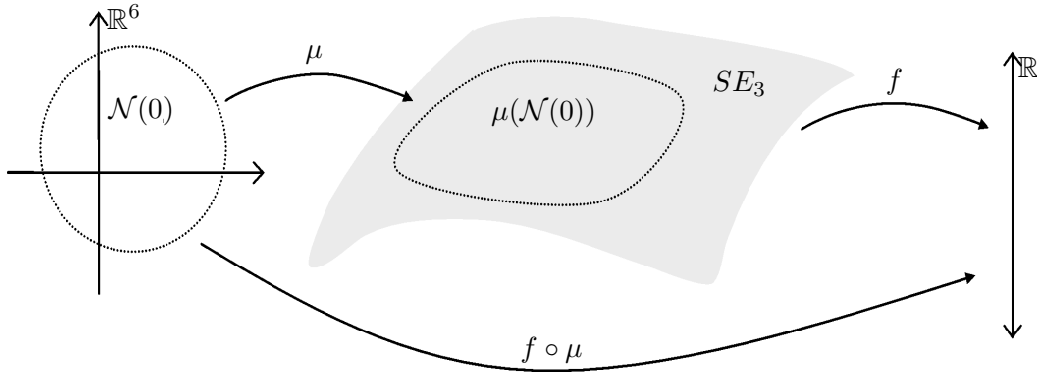


Fig. 3. The mapping  $\mu$  is the local parameterization of  $SE_3$  around point  $T$  such that  $T = \mu(0)$ ,  $f$  is the smooth function defined on  $SE_3$  and  $f \circ \mu$  is  $f$  expressed in local parameter space  $\mathbb{R}^6$ .

### III. OPTIMIZATION ON THE MANIFOLD $SE_3$

#### A. Geometry of the Special Euclidean Group $SE_3$

Both  $SO_3$  and  $SE_3$  are Lie groups. Associated with every Lie group is its Lie algebra. For  $SO_3$ , its Lie algebra  $\mathfrak{so}_3$  can be identified with the set of  $3 \times 3$  skew symmetric matrix of the form,

$$\Omega : \mathbb{R}^3 \rightarrow \mathfrak{so}_3, \quad \Omega(\omega) = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}, \quad (6)$$

and the Lie algebra of  $SE_3$ , denoted  $\mathfrak{se}_3$  is the set of  $4 \times 4$  matrix of the form,

$$\zeta : \mathbb{R}^6 \rightarrow \mathfrak{se}_3 \quad \zeta(x) = \begin{bmatrix} \Omega(\omega) & v \\ 0 & 0 \end{bmatrix}, \quad x := \begin{bmatrix} \omega \\ v \end{bmatrix}, \quad (7)$$

where  $\Omega(w) \in \mathfrak{so}_3$  and  $v \in \mathbb{R}^3$ .

The *tangent space* of  $SE_3$  at  $T$  is

$$\mathcal{T}_T SE_3 = \{T\zeta \mid \zeta \in \mathfrak{se}_3\}, \quad (8)$$

and the affine tangent space is

$$\mathcal{T}_T^{\text{aff}} SE_3 = \{T + T\zeta \mid \zeta \in \mathfrak{se}_3\}. \quad (9)$$

To achieve a *local Parameterization of  $SE_3$* , let  $\mathcal{N}(0) \subset \mathbb{R}^6$  denote a sufficiently small open neighbourhood of the origin in  $\mathbb{R}^6$ , and let  $T \in SE_3$ . Then the *exponential mapping*

$$\mu : \mathcal{N}(0) \subset \mathbb{R}^6 \rightarrow SE_3, \quad x \mapsto Te^{\zeta(x)}, \quad (10)$$

is a local diffeomorphism from  $\mathcal{N}(0)$  onto a neighbourhood of  $T$  in  $SE_3$ . More details of the exponential map can be found in [12].

### B. Cost Function in Local Parameter Space

The 2-jet (second order Taylor approximation) of  $f$  about the point  $T \in SE_3$  expressed in local parameter space using the smooth local parameterization  $\mu$  is given as

$$j_0^{(2)}(f \circ \mu) : \mathcal{N}(0) \subset \mathbb{R}^6 \rightarrow \mathbb{R}, \\ x \mapsto \left( (f \circ \mu)(tx) + \frac{d}{dt}(f \circ \mu)(tx) + \frac{1}{2} \frac{d^2}{dt^2}(f \circ \mu)(tx) \right) \Big|_{t=0}.$$

The mapping consists of

(i) a constant term,

$$(f \circ \mu)(tx) \Big|_{t=0} = \|B\text{vec}(A)\|^2,$$

(ii) a linear term,

$$\begin{aligned} \frac{d}{dt}(f \circ \mu)(tx) \Big|_{t=0} &= 2\text{vec}^\top(A\zeta(x) + \zeta^\top(x)A)B^\top B\text{vec}(A), \\ &= 2x^\top \nabla(f \circ \mu)(0). \end{aligned}$$

Denoting  $\text{vec}(\zeta(x)) := Gx$ ,  $\text{vec}(\zeta^\top(x)) := Jx$ , then  $G, J$  are  $16 \times 6$  matrices consisting of 1, -1, 0, the explicit formula for the gradient of  $f \circ \mu$  evaluated at 0 is given as,

$$\nabla(f \circ \mu)(0) = C^\top B\text{vec}(A), \quad C := B[(I \otimes A) (A \otimes I)] \begin{bmatrix} G \\ J \end{bmatrix}. \quad (11)$$

(iii) a quadratic term which consists of a sum of two terms. The first term is given as

$$\|B\text{vec}(A\zeta(x) + \zeta^\top(x)A)\|^2 = x^\top \hat{H}_{(f \circ \mu)(0)} x,$$

and the second term is

$$\text{vec}^\top(A)B^\top B\text{vec}(A\zeta^2(x) + 2\zeta^\top(x)A\zeta(x) + \zeta^{\top 2}(x)A) = x^\top \tilde{H}_{(f \circ \mu)(0)} x.$$

Thus, the Hessian of  $f \circ \mu$  evaluated at zero is

$$H_{(f \circ \mu)(0)} = \hat{H}_{(f \circ \mu)(0)} + \tilde{H}_{(f \circ \mu)(0)}, \quad (12)$$

and denoting  $\text{vec}(D) := B^\top B\text{vec}(A)$ , we have

$$\begin{aligned} \hat{H}_{(f \circ \mu)(0)} &= C^\top C \geq 0, \\ \tilde{H}_{(f \circ \mu)(0)} &= [G^\top \quad J^\top] \begin{bmatrix} (D^\top \otimes A) & (D^\top A \otimes I) \\ (AD \otimes I) & (A \otimes D^\top) \end{bmatrix} \begin{bmatrix} G \\ J \end{bmatrix}. \end{aligned} \quad (13)$$

The proposed algorithm consists of the iterations,

$$s = \pi_2 \circ \pi_1 : SE_3 \rightarrow SE_3, \quad (14)$$

where  $\pi_1$  maps a point  $T$  on the manifold  $SE_3$  to an element in the affine tangent space (by identifying  $\mathbb{R}^6$  with  $\mathcal{T}_T SE_3$  appropriately) and  $\pi_2$  projects that element back to  $SE_3$  by means of the parametrization  $\mu$ , as illustrated in Fig. 4.

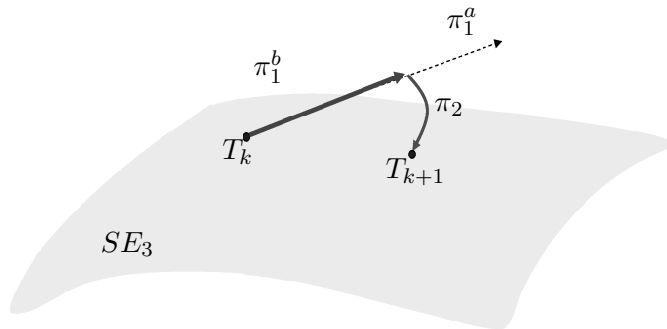


Fig. 4. The proposed algorithm first maps a point  $T \in SE_3$  to an element of the affine tangent space via  $\pi_1 = \pi_1^b \circ \pi_1^a$ , followed by step  $\pi_2$  to project that vector back to manifold.

### C.1 Optimization in Local Parameter Space, $\pi_1$

The optimization in parameter space consists of two steps, first calculate a suitable descent direction and then search for a step length that ensures reduction in cost function, as described by the mapping

$$\pi_1 = \pi_1^b \circ \pi_1^a : SE_3 \rightarrow \mathbb{R}^{4 \times 4}. \quad (15)$$

Here  $\pi_1^a$  is used to obtain a descent direction,

$$\pi_1^a : SE_3 \rightarrow \mathbb{R}^{4 \times 4}, \quad T \mapsto T + \zeta(x_{\text{opt}}(T)),$$

where

$$x_{\text{opt}} = \arg \min_{y \in \mathcal{N}(0)} j_0^{(2)}(f \circ \mu)(x). \quad (16)$$

The Newton direction is given by,

$$x_{\text{opt}}(x) = -\mathbf{H}_{(f \circ \mu)(x)}^{-1} \nabla(f \circ \mu)(x), \quad (17)$$

or a Gauss direction,

$$x_{\text{opt}}(x) = -\widehat{\mathbf{H}}_{(f \circ \mu)(x)}^{-1} \nabla(f \circ \mu)(x). \quad (18)$$

Once an optimal direction is computed, a one dimensional line search is carried out in this direction. An exact one dimensional line search will find the minimizer of the cost function along this line, but this is not feasible in this case. We proceed with an inexact search that ensure the cost function is reduced at every step. Here we use backtracking line search. Since we are using a descent direction, for sufficiently small step size, the cost function will go downhill. Backtracking line search first tries a step size of 1, if this is unacceptable, it reduces the step size until an acceptable step length is found. Details of the approach can be found in [13]. Thus,

$$\begin{aligned} \pi_1^b : \mathbb{R}^{4 \times 4} &\rightarrow \mathbb{R}^{4 \times 4}, \\ T + T\zeta(x_{\text{opt}}) &\mapsto T + T\zeta(\lambda_{\text{opt}}x_{\text{opt}}), \end{aligned} \quad (19)$$

where  $\lambda_{\text{opt}}$  is the step length that reduces the cost function in direction  $x_{\text{opt}}$ , and is found using the simple backtracking line search.

## C.2 Projecting Back via Parametrization $\mu$

Once the descent direction and downhill step size has been obtained, we project it back to the manifold via the parametrization  $\mu$ ,

$$\pi_2 : \mathbb{R}^{4 \times 4} \rightarrow SE_3, \quad T + T\zeta(\lambda_{\text{opt}}x_{\text{opt}}) \mapsto Te^{\zeta(\lambda_{\text{opt}}x_{\text{opt}})}. \quad (20)$$

## IV. CONVERGENCE ANALYSIS OF ALGORITHM

### A. Local Quadratic Convergence

The following result establishes local quadratic convergent of proposed Newton-like algorithms.

*Theorem IV.1:* Consider the proposed algorithmic mapping

$$T_{k+1} = s(T_k),$$

and denote  $T_* = \mu(0)$  as belonging to the set of local minima of  $j_0^{(2)}(f \circ \mu)(x)$ . Further assume that  $T_*$  is an isolated minimum in that  $H_{(f \circ \mu)(0)}^{-1}$  exists. Then  $s$  converges locally quadratically to  $T_*$ . Mathematical proof can be found in appendix.

### B. Global Convergence

The proposed algorithm achieves local quadratic convergence rates. However, the algorithm does not address the issue of escaping local minima. From implementation of the algorithm, convergence to local minima is particularly frequent for elliptic hyperboloid, hyperbolic paraboloid and hyperboloid of two sheets. Simulations suggest that the simplest approach is to initialize the algorithm randomly at different points on the manifold, and select the one with lowest cost.

## V. ALGORITHM INITIALIZATION

For initialization of the algorithm, we use a two steps least squares approach that gives closed form solution. In this approach, the pose estimation problem is split up into two subproblems, namely the surface fitting problem to recover the surface coefficient followed by pose estimation. Details of this strategies can be found in [2]. We summarize the steps used to estimate the pose in closed form and add in few new results that help in pose computation.

### A. Recovering Surface Coefficient

Consider the transformed surface coefficient  $A := T^\top QT$ . This belongs to the class of symmetric matrices  $\mathcal{S}^{4 \times 4}$ . We now consider an associated cost function,

$$\phi : \mathcal{S}^{4 \times 4} \rightarrow \mathbb{R}, \quad S \mapsto \|B\text{vec}(S)\|^2. \quad (21)$$

This cost function specializes as  $f$  on  $\{S \mid S = T^\top QT, T \in SE_3\}$ . Consider the mapping, with  $a = [a_1 \ a_2 \ \cdots \ a_{10}]^\top$ ,

$$\nu : \mathbb{R}^{10} \rightarrow \mathcal{S}^{4 \times 4}, \quad \nu(a) = \begin{bmatrix} a_1 & a_2 & a_3 & a_7 \\ a_2 & a_4 & a_5 & a_8 \\ a_3 & a_5 & a_6 & a_9 \\ a_7 & a_8 & a_9 & a_{10} \end{bmatrix}. \quad (22)$$

It is clear that this mapping is bijective. Now,

$$\phi \circ \nu : \mathbb{R}^{10} \rightarrow \mathbb{R}, \quad a \mapsto \|B\text{vec}(\nu(a))\|^2 = \|BK a\|^2, \quad (23)$$

where  $K$  is a matrix consisting of elements 1 and 0 such that  $Ka = \nu(a)$ . The vector  $a^*$  that minimizes the cost (23) subject to constraint  $\|a\| = 1$  is the right singular vector corresponding to the zero singular value of the matrix  $BK$ . Of course, when the data has noise, then  $a^*$  is the right singular vector associated with

the minimum singular value of matrix  $BK$ . However, to achieve Euclidean invariance, it is usual to constrain solution such that  $\text{tr}(A_{11}^\top A_{11}) = \text{tr}(Q_{11}^\top Q_{11}) = 1$ , i.e

$$\bar{a}^\top C \bar{a} = 1, \quad \bar{a} = [a_1 \cdots a_6], \quad C := \text{diag}(1, 2, 2, 1, 2, 1). \quad (24)$$

Once  $a^*$  is obtained, then since the map  $\nu$  is bijective, the optimum  $A \in \mathcal{S}$  is  $A^* = \nu(a^*)$ .

### B. Recovering Motion Parameters

Once the surface coefficient of the transformed quadric is determined, we can obtain  $R \in SO_3$  and  $t \in \mathbb{R}^3$  as follow. We know that

$$\kappa T(R, t)^\top QT(R, t) = A, \quad \text{for some scalar } \kappa, \quad (25)$$

and since  $Q_{11}, A_{11}$  are symmetric, singular value decomposition of both matrices will give,

$$Q_{11} = V_Q \Sigma_Q V_Q^\top, \quad A_{11} = V_A \Sigma_A V_A^\top, \quad (26)$$

where  $V_Q, V_A \in \mathcal{O}_3$  and  $\Sigma_Q, \Sigma_A$  are diagonal matrices with diagonal elements decreasing in magnitude, then we have

$$\{R^i = V_Q \Gamma^i V_A^\top \in SO_3\}, \quad (27)$$

where  $\Gamma^i$  is a diagonal matrix with diagonal elements  $\pm 1$ . There are  $2^3 = 8$  possible  $\Gamma^i$  matrices. We claim here that the optimal rotation is the one associated with minimum cost and minimum distance from the original position, i.e.,

$$i^* = \text{argmin}_i \|R^i - I\|^2 = \text{argmax}_i \text{tr}(R^i). \quad (28)$$

Once optimal  $R^* := R^{i^*}$  is found, we can compute an optimal  $t^*$  from,

$$t^* = Q_{11}^{-1}(\kappa^{-1} R A_{12} - Q_{12}), \quad \kappa = \frac{1}{3} \text{tr}(\Sigma_A \Sigma_Q^{-1}). \quad (29)$$

Note that when  $Q_{11}$  is singular, such as in the case of paraboloid surfaces, the solution is not unique, so a pseudo inverse is used to recover one optimal solution  $t^*$ .

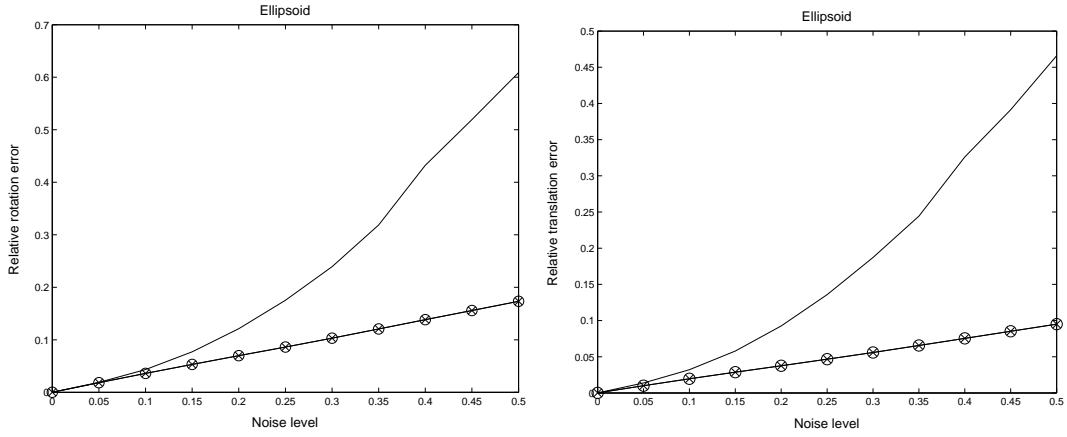
## VI. CYCLIC COORDINATE DESCENT METHOD

Cyclic coordinate descent (CCD) is a variant of our proposed approach. It exploits the property that  $SE_3$  is the product manifold  $SO_3 \times \mathbb{R}^3$ . At each iteration, we first freeze the rotation and optimize only the translation, then next freeze the translation and optimize only the rotation. The motivation for this is that we can carry out analytic geodesic searches in  $SO_3$ , requiring the solution of an 8<sup>th</sup> order polynomial. Likewise, in a line search for  $t \in \mathbb{R}^3$  a 3<sup>rd</sup> order polynomial is solved. The advantage of this approach is that it can eventually escape from a local minimum, not the global minimum, and usually in a few iterations, and so could be useful for the high noise situation when an initial estimate is not in the domain of attraction of the global minimum. Being a linearly convergent algorithm, and typically converging more rapidly than a Newton algorithm in the first few iterations as illustrated in simulations below, it is primarily useful for the first few iterations to get close to the global minimum, and may then be subsequently replaced by the Gauss-Newton method to achieve quadratic convergence rates. Details for algorithm implementation are given in Appendix B where its disadvantage becomes clear in that it involves significant programming effort. Of course, another approach than the CCD approach to avoiding local minima not the global minimum is to re-initialize the Newton algorithm a number of times and select the result with the lowest cost.

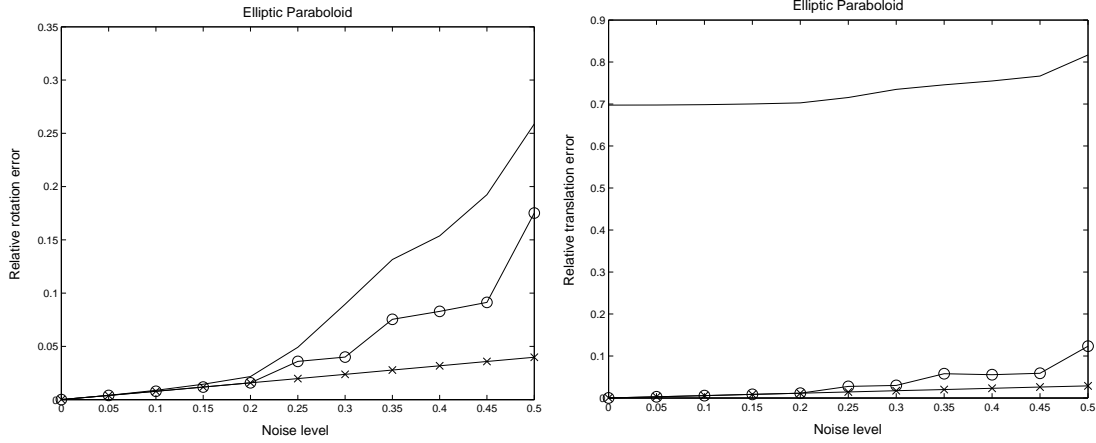
## VII. IMPLEMENTATION OF ALGORITHM

*Initialize* Start with an initial estimate of the rigid body transformation matrix  $T$  obtained from initialization algorithm.

*CCD Iterations* Implement say up to four steps of the Cyclic Co-ordinate Descent algorithm of the previous section ( useful for the high noise case, or alternatively use re-initializations).



(a)Relative pose error vs. noise level for ellipsoid.



(b)Relative pose error vs. noise level for elliptic paraboloid.

Fig. 5. Robustness against additive Gaussian noise for closed form least squares approach (solid line), geometric approach without restart ( $\circ$ ), geometric approach with 5 random restart ( $\times$ ).

*Gauss-Newton Iterations* Implement the Gauss-Newton algorithm with each iteration consisting of two mappings. 1) The optimization step,

- Compute the gradient  $\nabla(f \circ \mu)(0)$  and the Hessian  $\mathbf{H}_{(f \circ \mu)(0)}$  via (11), (12) respectively,
- If  $\mathbf{H}_{(f \circ \mu)(0)} > 0$ ,

compute the Newton step,  $x_{\text{opt}} = \mathbf{H}_{(f \circ \mu)(0)}^{-1} \nabla(f \circ \mu)(0)$ ,

otherwise compute the Gauss step  $x_{\text{opt}} = \hat{\mathbf{H}}_{(f \circ \mu)(0)}^{-1} \nabla(f \circ \mu)(0)$ ,

- Compute the optimum step size  $\lambda_{\text{opt}}$  in direction  $x_{\text{opt}}$  using backtracking line search.

2) Carry out the projection step,  $\hat{T} = T e^{\zeta(\lambda_{\text{opt}} x_{\text{opt}})}$ ,

*Termination* Terminate if  $\|\nabla(f \circ \mu)(0)\| > \epsilon$ , a prescribed accuracy.

## VIII. SIMULATIONS

A series of simulations were performed on artificially generated uniformly distributed points on quadric surface. Our proposed geometric approach (GA) has been compared with the 2 steps closed form least squares solutions presented under algorithm initialization (LS) and the cyclic coordinate descent method (CCD). For comparison purposes the methods have been implemented separately rather than in co-ordination.

Performances of the different techniques are evaluated by comparing the relative Euclidean distance between the pose parameters. All simulations are implemented using Matlab.

### A. Sensitivity Analysis

We investigate the robustness of the algorithms as the measured data are corrupted with increasing amount of additive Gaussian noise and when only partial views of the surface are available. Simulations show both CCD and GA have the same performance accuracy when the noise level and size of the surface are varied. Thus, only results for GA and LS are plotted. Figure 5 indicates that GA is far less sensitive to additive Gaussian noise and than LS. Interestingly, for an ellipsoid, the initialization achieves the global minimum in the presence of high noise level (Fig. 5(a)) since random restart converges to the same minimum. Similar results are also observed for an elliptic hyperboloid of one sheet. However, for elliptic paraboloid (Fig. 5(b)), hyperboloid of two sheets and hyperbolic paraboloid, we observe the presence of many local minima, thus random reinitializations of the algorithm or initial CCD iterations are required to achieve ‘global’ minimum.

Figure 6 shows the performance of GA and LS when part of the surface are occluded. It clearly indicates that GA is far more robust to occlusion than LS. Likewise, for relative translation error.

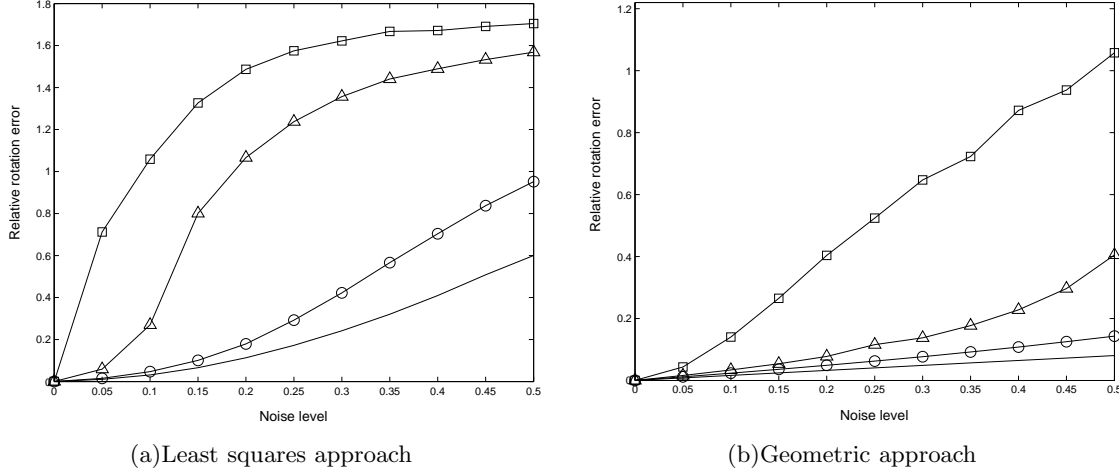


Fig. 6. Robustness against occlusion: whole surface (solid line), half surface ( $\circ$ ), quarter surface ( $\Delta$ ), small patch ( $\square$ ).

### B. Speed of Convergence

Figure 7 illustrates that GA converges at a local quadratic rate but the CCD can converge more rapidly during its first few iterations, but then converges linearly. Such typical results suggest that we use CCD for the first few iterations then switch to the Newton method, at least in the high noise case where there are extra benefits of avoiding local minima not the global minimum. Also, the local quadratic convergence rate GA is also better than the approach presented in [11] which claims to converge at an exponential rate.

*Remark VIII.1:* For symmetric quadrics such as sphere and cylinder, the estimated pose is no longer belonging to  $SE_3$  of 6 dimension, rather it belongs to the reduced space (sphere – 3 dimension, cylinder – 4 dimension). Similar algorithmic approach apply with minor modification. For cylinder, we have

$$\Omega : \mathbb{R}^2 \rightarrow \mathfrak{so}_3, \quad \Omega(\omega) = \begin{bmatrix} 0 & 0 & \omega_2 \\ 0 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}, \quad (30)$$

and the Lie algebra of  $SE_3$ , denoted  $\mathfrak{se}_3$  is the set of  $4 \times 4$  matrix of the form

$$\zeta : \mathbb{R}^4 \rightarrow \mathfrak{se}_3 \quad \zeta(x) = \begin{bmatrix} \Omega(\omega) & v \\ 0 & 0 \end{bmatrix}, \quad x := \begin{bmatrix} \omega \\ v \end{bmatrix} \quad (31)$$

where  $\Omega(w) \in \mathfrak{so}_3$  and  $v \in \mathbb{R}^3$ , results in a  $4 \times 4$  Hessian.

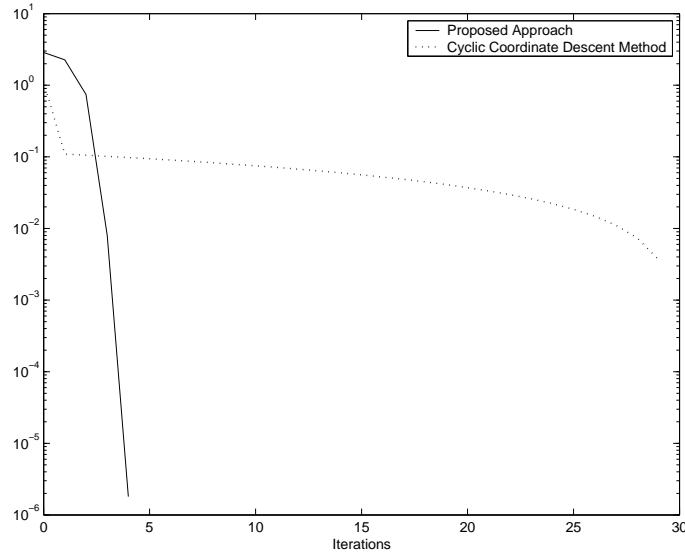


Fig. 7. Rate of local convergence.

For sphere, due to symmetry, we only need to find the relative translation, thus reduce the dimension from 6 to 3.

$$\zeta(v) = \begin{bmatrix} 0 & v \\ 0 & 1 \end{bmatrix} \quad (32)$$

Thus, each Newton step requires only inversion of a  $3 \times 3$  Hessian matrix. Similarly for other symmetric quadric.

## IX. CONCLUSIONS

This paper presents a novel geometric algorithm capable of locating quadratic surfaces quickly and accurately, even in the presence of measurement noise and occlusions. The main contributions of this chapter can be summarized as follows,

- A novel parameterization-based framework for minimizing smooth function on the special Euclidean group,
- An efficient algorithm that is data independent at each iteration and converges at a local quadratic rate,
- An algorithm that explicitly preserve constraints at every iteration,
- Comparison of the proposed algorithm with existing two steps least squares methods,
- Cyclic co-ordinate descent iterations, or parallel random initializations assist in escaping local minima, not the global minimum.
- A proof of local quadratic convergence of the devised Gauss-Newton-like algorithms.

## X. APPENDIX

### A. Proof of Theorem IV.1.

Let  $T_*$  denotes a fixed point of  $s = \pi_2 \circ \pi_1$ , i.e.,  $T_*$  is a minimum of the function  $f$ . Applying chain rule and using the fact that  $\pi_1(T_*) = \pi_2(T_*) = T_*$ , for all elements  $\xi \in \mathcal{T}_{T_*}SE_3$ , the first derivative of  $s$  at fixed point  $T_*$  is,

$$Ds(T_*) \cdot \xi = D\pi_2(T_*) \cdot D\pi_1(T_*) \cdot \xi. \quad (33)$$

Considering  $s$  in the local parameter space, we have the self map

$$\mu^{-1} \circ s \circ \mu : \mathbb{R}^6 \rightarrow \mathbb{R}^6. \quad (34)$$

Thus, rewriting (33) in terms of local parameterization defined by

$$\mu : \mathbb{R}^6 \supset \mathcal{N}(0) \rightarrow SE_3, \quad x \mapsto T_* e^{\zeta(x)}, \quad (35)$$

with  $\mu(0) = T_*$ , and  $\zeta$  as in (7), we have

$$\begin{aligned} D(\mu^{-1} \circ s \circ \mu)(0) \cdot h &= D\mu^{-1}(T_*) \cdot Ds(T_*) \cdot D\mu(0) \cdot h, \\ &= D\mu^{-1}(T_*) \cdot D\pi_2(T_*) \cdot D\pi_1^b(T_*) \cdot D(\pi_1^a \circ \mu)(0) \cdot h. \end{aligned} \quad (36)$$

Next, consider the composite function

$$\pi_1 \circ \mu : \mathbb{R}^6 \supset \mathcal{N}(0) \rightarrow \mathbb{R}^{6 \times 6}, \quad x \mapsto T_* e^{\zeta(x)} + T_* \zeta(x_{\text{opt}}(x)), \quad (37)$$

where

$$x_{\text{opt}} = \arg \min_{x \in \mathcal{N}(0)} j_0^{(2)}(f \circ \mu)(x). \quad (38)$$

Exploiting linearity of the mapping  $\zeta$ , using the well known formula for differentiating the matrix exponential and the fact that  $x_{\text{opt}}(0) = 0$ , we compute the first derivative of  $\pi_1$  in local coordinates as

$$D(\pi_1 \circ \mu)(0) \cdot h = T_* \zeta(h) + T_* \zeta(Dx_{\text{opt}}(0) \cdot h). \quad (39)$$

Thus, we need an expression for  $Dx_{\text{opt}}(0) \cdot h$ . Knowing that Newton step  $x_{\text{opt}} = -\mathbf{H}_{(f \circ \mu)(0)}^{-1} \nabla(f \circ \mu)(0)$  is used around the fixed point  $\mu(0) = T_*$ , we have

$$\begin{aligned} Dx_{\text{opt}}(0) \cdot h &= -\mathbf{H}_{(f \circ \mu)(0)}^{-1} D(\nabla(f \circ \mu)(0)) \cdot h - D(\mathbf{H}_{(f \circ \mu)(0)}^{-1}) \cdot h \nabla(f \circ \mu)(0), \\ &= -\mathbf{H}_{(f \circ \mu)(0)}^{-1} D(\nabla(f \circ \mu)(0)) \cdot h \quad \text{since } \nabla(f \circ \mu)(0) = 0, \\ &= -\mathbf{H}_{(f \circ \mu)(0)}^{-1} \mathbf{H}_{(f \circ \mu)(0)} h, \\ &= -h. \end{aligned} \quad (40)$$

Substituting (40) into (39), we obtain

$$D(\pi_1 \circ \mu)(0) \cdot h = 0. \quad (41)$$

Plugging (41) into (36) shows that for all  $h \in \mathbb{R}^6$

$$D(\mu^{-1} \circ s \circ \mu)(0) \cdot h = 0, \quad (42)$$

irrespective of the line search and projection step. Let  $(T^{(k)})$  denote the sequence of rigid body transformation matrices generated by the algorithm. Let  $x^{(k)} = \mu^{-1}(T^{(k)})$  denote the corresponding elements in  $\mathbb{R}^6$ . For sufficiently large  $j$  we may assume that for all  $k \geq j$  the iterates  $x^{(k)}$  stay in a sufficiently small neighborhood of the origin in  $\mathbb{R}^6$ . Vanishing of the first derivative then implies local quadratic convergence by the Taylor-type argument

$$\left\| (\mu^{-1} \circ s \circ \mu)(x^{(k)}) \right\| \leq \sup_{y \in \mathcal{N}(0)} \|D^2(\mu^{-1} \circ s \circ \mu)(y)\| \cdot \|x^{(k)}\|^2. \quad (43)$$

### B. Cyclic Coordinate Descent

Recall the cost function defined on manifold  $SE_3$  is,

$$f : SE_3 \rightarrow \mathbb{R}, \quad f(T) = \frac{1}{n} \sum_{i=1}^n (p_i^\top T^\top Q T p_i)^2, \quad p_i = \begin{bmatrix} \tilde{p}_i \\ 1 \end{bmatrix}. \quad (44)$$

Since  $SE_3$  is the product manifold  $SO_3 \times \mathbb{R}^3$ , the cost function can also be expressed as,

$$\begin{aligned} g : SO_3 \times \mathbb{R}^3 &\rightarrow \mathbb{R}, \\ g(R, t) &= \frac{1}{n} \sum_{i=1}^n \left( \tilde{p}_i^\top R^\top Q_{11} R \tilde{p}_i + 2\tilde{p}_i^\top R^\top (Q_{11}t + Q_{12}) + t^\top Q_{11}t + 2t^\top Q_{12} + Q_{22} \right)^2. \end{aligned} \quad (45)$$

Assuming the translation is known, and denoting  $b := Q_{11}t + Q_{12}$ ,  $c := t^\top Q_{11}t + 2t^\top Q_{12} + Q_{22}$ , the cost function in terms of rotation is given as,

$$\begin{aligned} g_R : SO_3 &\rightarrow \mathbb{R}, \\ g_R(R) &= \frac{1}{n} \sum_{i=1}^n \left( \tilde{p}_i^\top R^\top Q_{11} R \tilde{p}_i + 2\tilde{p}_i^\top R^\top b + c \right)^2, \\ &= \begin{bmatrix} \text{vec}^\top(R^\top Q_{11} R) & \text{vec}^\top(R) & 1 \end{bmatrix} L \begin{bmatrix} \text{vec}(R^\top Q_{11} R) \\ \text{vec}(R) \\ 1 \end{bmatrix}, \end{aligned} \quad (46)$$

where

$$\begin{aligned} L &:= \begin{bmatrix} L_{11} & L_{12} & 0 \\ L_{12}^\top & L_{22} & L_{23} \\ 0 & L_{23}^\top & c^2 \end{bmatrix}, \quad L_{11} := \frac{1}{n} \sum_{i=1}^n \text{vec}(\tilde{p}_i \tilde{p}_i^\top) \text{vec}^\top(\tilde{p}_i \tilde{p}_i^\top), \quad L_{12} := \frac{2}{n} \left( \sum_{i=1}^n \tilde{p}_i \tilde{p}_i^\top \otimes \tilde{p}_i \right) (I \otimes b^\top), \\ L_{22} &:= \frac{2}{n} \left( \sum_{i=1}^n \tilde{p}_i \tilde{p}_i^\top \otimes (2bb^\top + cQ_{11}) \right), \quad L_{23} := \frac{2c}{n} \left( \sum_{i=1}^n \tilde{p}_i \otimes I \right) b. \end{aligned} \quad (47)$$

Consider the geodesic on  $SO_3$  emanating from point  $R$  along the direction  $\Omega \in \mathfrak{so}_3$ ,

$$\gamma : \mathbb{R} \rightarrow SO_3, \quad \gamma(\theta) = R e^{\theta\Omega}, \quad (48)$$

then the cost function defined on  $SO_3$  restricted to geodesic is given by the composition function below,

$$\begin{aligned} g_R \circ \gamma : \mathbb{R} &\rightarrow \mathbb{R}, \\ \theta &\mapsto \begin{bmatrix} \text{vec}^\top(e^{\theta\Omega^\top} R^\top Q_{11} R e^{\theta\Omega}) & \text{vec}^\top(R e^{\theta\Omega}) & 1 \end{bmatrix} L \begin{bmatrix} \text{vec}(e^{\theta\Omega^\top} R^\top Q_{11} R e^{\theta\Omega}) \\ \text{vec}(R e^{\theta\Omega}) \\ 1 \end{bmatrix}. \end{aligned} \quad (49)$$

Using Rodrigues' formula,

$$e^{\theta\Omega(\omega)} = I + \Omega(\omega) \sin(\theta) + \Omega(\omega)^2 (1 - \cos(\theta)), \quad \|\omega\| = 1, \quad (50)$$

then we have

$$\text{vec}(R e^{\theta\Omega}) = G \begin{bmatrix} \cos(\theta) \\ \sin(\theta) \\ 1 \end{bmatrix}, \quad G := (I \otimes R) \begin{bmatrix} \text{vec}(-\Omega^2) & \text{vec}(\Omega) & \text{vec}(I + \Omega^2) \end{bmatrix}, \quad (51)$$

and

$$\text{vec}(e^{\theta\Omega^\top} R^\top Q_{11} R e^{\theta\Omega}) = J \begin{bmatrix} \cos^2(\theta) \\ \sin^2(\theta) \\ \sin(\theta) \cos(\theta) \\ \cos(\theta) \\ \sin(\theta) \\ 1 \end{bmatrix}, \quad J := [\text{vec}(J_1) \quad \text{vec}(J_2) \quad \cdots \quad \text{vec}(J_6)], \quad (52)$$

denoting  $\bar{Q} := R^\top Q_{11} R$ , we have

$$\begin{aligned} J_1 &:= \Omega^2 \bar{Q} \Omega^2, \quad J_2 := \Omega^\top \bar{Q} \Omega, \quad J_3 := -\Omega^\top \bar{Q} \Omega^2 - \Omega^2 \bar{Q} \Omega, \quad J_4 := -(I + \Omega^2) \bar{Q} \Omega^2 - \Omega^2 \bar{Q} (I + \Omega^2), \\ J_5 &:= (I + \Omega^2) \bar{Q} \Omega + \Omega^\top \bar{Q} (I + \Omega^2), \quad J_6 := (I + \Omega^2) \bar{Q} (I + \Omega^2) \end{aligned} \quad (53)$$

Substituting (51) – (52) into (49), we obtain

$$g_R \circ \gamma(\theta) = \begin{bmatrix} \cos^2(\theta) & \sin^2(\theta) & \sin(\theta) \cos(\theta) & \cos(\theta) & \sin(\theta) & 1 \end{bmatrix} K \begin{bmatrix} \cos^2(\theta) \\ \sin^2(\theta) \\ \sin(\theta) \cos(\theta) \\ \cos(\theta) \\ \sin(\theta) \\ 1 \end{bmatrix} \quad (54)$$

where

$$K = (k_{ij}) := \begin{bmatrix} J \\ \begin{bmatrix} 0_{9 \times 3} & G \\ 0_{1 \times 5} & 1 \end{bmatrix} \end{bmatrix}^\top L \begin{bmatrix} J \\ \begin{bmatrix} 0_{9 \times 3} & G \\ 0_{1 \times 5} & 1 \end{bmatrix} \end{bmatrix} \quad (55)$$

To find all the critical points, we take the first derivative of (54) and setting it to zero,

$$\frac{dg_R \circ \gamma(\theta)}{d\theta} = 2 \begin{bmatrix} \cos^2(\theta) & \sin^2(\theta) & \sin(\theta) \cos(\theta) & \cos(\theta) & \sin(\theta) & 1 \end{bmatrix} K \begin{bmatrix} -2 \sin(\theta) \cos(\theta) \\ 2 \sin(\theta) \cos(\theta) \\ \cos^2(\theta) - \sin^2(\theta) \\ -\sin(\theta) \\ \cos(\theta) \\ 0 \end{bmatrix} = 0 \quad (56)$$

This gives us a quartic equation in cosine and sine terms. Apply trigonometric formula such that  $\sin^2(\theta) = 1 - \cos^2(\theta)$ , results in an 8<sup>th</sup> order polynomial in terms of cosine, as given below,

$$\sum_{i=0}^8 d_i (\cos^i(\theta)) = 0 \quad (57)$$

where

$$\begin{aligned} d_0 &:= \alpha_0^2 - \beta_0^2, & d_1 &:= 2\alpha_1\alpha_0 - 2\beta_1\beta_0, & d_2 &:= \alpha_1^2 + 2\alpha_1\beta_0 - \beta_1^2 - 2\beta_2\beta_0 + \beta_0^2 \\ d_3 &:= 2\alpha_3\alpha_0 + 2\alpha_2\alpha_1 - 2\beta_3\beta_0 - 2\beta_2\beta_1 + 2\beta_1\beta_0, & d_4 &:= \alpha_2^2 + 2\alpha_4\alpha_0 + 2\alpha_3\alpha_1 - \beta_2^2 - 2\beta_3\beta_1 + \beta_1^2 + 2\beta_2\beta_0, \\ d_5 &:= 2\alpha_4\alpha_1 + 2\alpha_3\alpha_2 - 2\beta_3\beta_2 + 2\beta_3\beta_0 + 2\beta_2\beta_1, & d_6 &:= \alpha_3^2 + 2\alpha_4\alpha_2 - \beta_3^2 + \beta_2^2 + 2\beta_3\beta_1, \\ d_7 &:= 2\alpha_4\alpha_3 + 2\beta_3\beta_2, & d_8 &:= \alpha_4^2 + \beta_3^2, \end{aligned} \quad (58)$$

and

$$\begin{aligned} \alpha_0 &:= m_2 + m_{11}, & \alpha_1 &:= m_9 + m_{13}, & \alpha_2 &:= -2m_2 + m_5 + m_{10} - m_{11}, & \alpha_3 &:= m_6 - m_9, \\ \alpha_4 &:= m_1 + m_2 - m_5, & \beta_0 &:= m_7 + m_{14}, & \beta_1 &:= m_4 + m_{12}, & \beta_2 &:= -m_7 + m_8, & \beta_3 &:= m_3 - m_4 \end{aligned} \quad (59)$$

such that

$$\begin{aligned} m_1 &:= k_{13}, & m_2 &:= -k_{23}, & m_3 &:= -2k_{11} + 2k_{12} + k_{33}, & m_4 &:= -2k_{21} + 2k_{22} - k_{33}, \\ m_5 &:= -k_{13} + k_{23} - 2k_{31} + 2k_{32}, & m_6 &:= k_{15} + k_{43}, & m_7 &:= -k_{24} - k_{53}, \\ m_8 &:= -k_{14} + k_{35} - 2k_{41} + 2k_{42} + k_{53}, & m_9 &:= k_{25} - k_{34} - k_{43} - 2k_{51} + 2k_{52}, \\ m_{10} &:= k_{45} + k_{63}, & m_{11} &:= -k_{54} - k_{63}, & m_{12} &:= -k_{44} + k_{55} - 2k_{61} + 2k_{62}, \\ m_{13} &:= k_{65}, & m_{14} &:= -k_{64}, & m_{15} &:= 0. \end{aligned} \quad (60)$$

## B.1 Optimizing translation

When the rotation  $R$  is known then the translation  $t$  is optimized by minimizing a scalar quartic polynomial. This is achieved by setting its derivative to zero and solving the resulting cubic equation, as detailed below.

Denoting  $b_i := Q_{11}R\tilde{p}_i + Q_{12}$ ,  $c_i := \tilde{p}_i^\top R^\top Q_{11}R\tilde{p}_i + 2\tilde{p}_i^\top R^\top Q_{12} + Q_{22}$ , the cost function in terms of translation  $t$  is given as,

$$g_t : \mathbb{R}^3 \rightarrow \mathbb{R},$$

$$g_t(t) = \frac{1}{n} \sum_{i=1}^n (t^\top Q_{11}t + 2t^\top b_i + c_i)^2 = \begin{bmatrix} t^\top Q_{11}t \\ t \\ 1 \end{bmatrix}^\top W \begin{bmatrix} t^\top Q_{11}t \\ t \\ 1 \end{bmatrix}, \quad (61)$$

where

$$W := \begin{bmatrix} 1 & W_{12} & 0 \\ W_{12}^\top & W_{22} & W_{23} \\ 0 & W_{23}^\top & W_{33} \end{bmatrix} \in \mathbb{R}^{5 \times 5}, \quad (62)$$

and

$$W_{12} := \frac{2}{n} \sum_{i=1}^n b_i^\top \in \mathbb{R}^3, \quad W_{22} := \frac{2}{n} \left( \sum_{i=1}^n 2b_i b_i^\top + c_i Q_{11} \right) \in \mathbb{R}^{3 \times 3},$$

$$W_{23} := \frac{2}{n} \sum_{i=1}^n b_i c_i \in \mathbb{R}^3, \quad W_{33} := \frac{1}{n} \sum_{i=1}^n c_i^2 \in \mathbb{R}. \quad (63)$$

Consider the smooth mapping,

$$\varphi : \mathbb{R} \rightarrow \mathbb{R}^3, \quad \theta \mapsto t + \theta v, \quad (64)$$

and the composite function,

$$g_t \circ \varphi : \mathbb{R} \rightarrow \mathbb{R},$$

$$g_t \circ \varphi(\theta) = \begin{bmatrix} (t + \theta v)^\top Q_{11}(t + \theta v) \\ t + \theta v \\ 1 \end{bmatrix}^\top W \begin{bmatrix} (t + \theta v)^\top Q_{11}(t + \theta v) \\ t + \theta v \\ 1 \end{bmatrix}. \quad (65)$$

Equation (65) can be re-expressed in terms of  $\theta$  as,

$$g_t \circ \varphi(\theta) = \begin{bmatrix} \theta^2 & \theta & 1 \end{bmatrix} Y \begin{bmatrix} \theta^2 \\ \theta \\ 1 \end{bmatrix}, \quad Y = (y_{ij}) := Z^\top W Z, \quad (66)$$

where

$$Z := \begin{bmatrix} v^\top Q_{11}v & 2t^\top v & t^\top Q_{11}t \\ 0 & v & t \\ 0 & 0 & 1 \end{bmatrix}, \quad (67)$$

Setting first derivative of  $g_t \circ \varphi(\theta) = 0$  results in the following cubic equation,

$$2y_{11}\theta^3 + 3y_{12}\theta^2 + (y_{22} + 2y_{13})\theta + y_{23} = 0. \quad (68)$$

#### REFERENCES

- [1] K. Arun, T. S. Hwang, S. Bolstein, *Least Squares Fitting of Two 3D Point Sets*, IEEE Trans. on Pattern Anal. Machine Intelligent, pp. 698–700, 1987.
- [2] M. Baeg, H. Hashimoto, F. Harashima, J. B. Moore, *Pose Estimation of Quadratic Surface of Quadratic Surface Using Surface Fitting Technique*, IROS'95, pp. 204–209, 1995.
- [3] P. Besl, N. McKay, *A method for registration of 3D shapes*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 14, No. 2, pp. 239–256, 1992.
- [4] G. L. Bilbro, W. E. Synder, *Linear Estimation of Object pose from local fits to segments*, IEEE International Conference of Robotics and Automation, pp. 1747–1752, 1987.
- [5] O. D. Faugeras, M. Herbert, *The representation, recognition, and location of 3D objects*, International Journal of Robotics Research, vol. 5, no. 3, pp. 27–52, 1986.

- [6] K. Gunnarsson, F. Prinz, *CAD model based recognition and localization of parts in manufacturing*, Computer, vol. 20, no. 8, pp. 67–74, 1987.
- [7] R. M. Haralick, H. Joo, C. Lee, X. Zhuang, V. G. Vaidya, M. B. Kim, *Pose estimation from corresponding point data*, IEEE Transactions on Systems, Man and Cybernetics, vol. 19, no. 6, 1426–1446, 1989.
- [8] Y. Han, W. E. Synder, G. L. Bilbro, *Pose determination using tree annealing*, IEEE International Conference of Robotics and Automation, pp. 427–432, 1990.
- [9] B. K. P. Horn, *Closed form solution of absolute orientation using unit quaternions* Journal of the Optical Society of America A, 1987.
- [10] P. Y. Lee, J. B. Moore *Geometric Optimization for 3D Pose Estimation of Quadratic Surfaces*, Proceedings of the Thirty-Eighth Asilomar Conference on Signals, Systems and Computers, Pacific Grove, California, USA, pp. 131-135, November 2004.
- [11] J. B. Moore, M. Baeg, U. Helmke *Gradient Flow Techniques for Pose Estimation of Quadratic Surfaces*, Proc. of the World Congress in Computational Methods and Applied Mathematics, Atlanta, Georgia, July 1994, pp. 360–363
- [12] R. M. Murray, Z. Li, and S. S. Sastry, *A Mathematical Introduction to Robotic Manipulation*, CRC Press, 1994.
- [13] J. Nocedal, S. J. Wright, *Numerical Optimization*, Springer Series in Operation Research, Springer Verlag, New York, 1999.