

# Geometric Optimization for 3D Pose Estimation of Quadratic Surfaces

Pei Yean Lee, John B. Moore

Department of Information Engineering, RSISE,  
Australian National University, ACT 0200 Australia and  
National ICT Australia Limited,  
Locked Bag 8001, Canberra ACT 2601, Australia.  
peiyea@syseng.anu.edu.au, john.moore@syseng.anu.edu.au

**Abstract**—Our task is 3D pose estimation for on-line application in industrial robotics and machine vision. It involves the estimation of object position and orientation relative to a known model. Since most man made objects can be approximated by a small set of quadratic surfaces, in this paper we focus on pose estimation of such surfaces. Our optimization is of an error measure between the CAD model and the measured data. Most existing algorithms are sensitive to noise and occlusion or only converge linearly. Our optimization involves iterative cost function reduction on the smooth manifold of the Special Euclidean Group,  $SE_3$ . The optimization is based on locally quadratically convergent Newton-type iterations on this constraint manifold. A careful analysis of the underlying geometric constraint is required.

## I. INTRODUCTION

Quadratic surfaces, also known as quadrics, are commonly occurring shapes in man made objects. Accurate and fast localization of quadrics or surfaces consists of quadric patches from measurement data is important in many industrial robotics and machine vision tasks as well as a step towards developing a flexible manufacturing system.

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], [10], [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. The only information required from the database are the equation of the surface.

In this paper, we propose a new geometric approach based on Newton-type iterations on the constraint manifold. Our optimization involves iterative cost function reduction on the smooth manifold of the Special Euclidean Group,  $SE_3$ . The optimization is based on locally quadratically convergent Newton-type iterations on this constraint manifold. To achieve this, careful analysis of the underlying geometric constraint is required.

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.

Section II presents some background knowledge about the geometry of Special Euclidean Group and Section III formulates the problem mathematically in differential geometry framework.

We propose a geometric approach to address the problem in Section IV and the initialization of the algorithm in Section V. Implementation of the algorithm is outlined in Section VI. The convergence properties of the algorithm are given in section VII and a series of simulations are presented in Section VIII, followed by a conclusion section.

## II. GEOMETRY OF SPECIAL EUCLIDEAN GROUP

### A. Special Euclidean Group

Rigid body motions in  $\mathbb{R}^3$  can be represented by the special Euclidean group, denoted  $SE_3$ ,

$$SE_3 := \{(R, t) \mid R \in SO_3, t \in \mathbb{R}^3\} = SO_3 \times \mathbb{R}^3. \quad (1)$$

Here  $SO_3$  is the group of  $3 \times 3$  orthogonal matrices with determinant +1, which represent the rotations and  $t$  is the translation vector. 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

$$\Omega : \mathbb{R}^3 \rightarrow \mathfrak{so}_3, \quad \Omega(\omega) = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}, \quad (2)$$

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

$$\zeta : \mathbb{R}^6 \rightarrow \mathfrak{se}_3 \quad \zeta(x) = \begin{bmatrix} \Omega(\omega) & v \\ 0 & 0 \end{bmatrix}, \quad x := \begin{bmatrix} \omega \\ v \end{bmatrix} \quad (3)$$

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

### B. Tangent Space of $SE_3$

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

$$\mathcal{T}_T SE_3 = \{T\zeta \mid \zeta \in \mathfrak{se}_3\} \quad (4)$$

and the affine tangent space is

$$\mathcal{T}_T^{\text{aff}} SE_3 = \{T + T\zeta \mid \zeta \in \mathfrak{se}_3\} \quad (5)$$

### C. 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

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

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 [11].

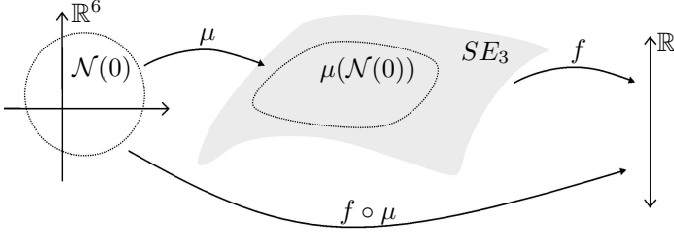


Fig. 1. Local parameterization and cost function

### III. PROBLEM FORMULATION

In what follows,  $\text{vec}$  is an operator which creates a column vector from a matrix  $A$  by stacking the column vectors of  $A$  below one another,  $\otimes$  represents the Kronecker product:  $A \otimes B$  denotes a matrix with the elements of  $A$ , denoted  $a_{ij}$  replaced by  $a_{ij}B$ . Recall that  $(A \otimes B)^\top = A^\top \otimes B^\top$ ,  $(A \otimes B)(C \otimes D) = (AC \otimes BD)$ ,  $\text{vec}(ABC) = (C^\top \otimes A) \text{vec}(B)$ , and  $(A \otimes B)^{-1} = (A^{-1} \otimes B^{-1})$  when the inverses exist.

#### A. Quadratic Surface

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

$$m^\top Q m = 0, \quad m := \begin{bmatrix} \tilde{m} \\ 1 \end{bmatrix}, \quad Q := \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{12}^\top & Q_{22} \end{bmatrix} \quad (7)$$

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

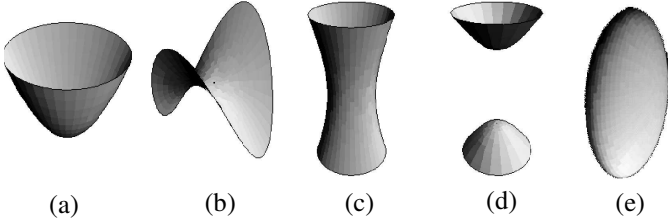


Fig. 2. (a) Elliptic Paraboloid, (b) Hyperbolic Paraboloid, (c) Elliptic Hyperboloid of One Sheet, (d) Elliptic Hyperboloid of Two Sheets, (e) Ellipsoid

Now, consider the quadric being rotated by a matrix  $R \in SO_3$  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,

$$\left\{ p = \begin{bmatrix} \tilde{p} \\ 1 \end{bmatrix} \in \mathbb{R}^4 \mid p^\top A(R, t) p = 0 \right\}. \quad (8)$$

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

$$T(R, t) := \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \in SE_3. \quad (9)$$

is the 3D rigid body transformation matrix.

#### B. The Cost Function

Given surface measurement data  $p_i \in \mathbb{R}^4$  and known surface coefficient  $Q$ , the task is to find the transformation matrix  $T(R, t) \in SE_3$  that satisfies (8). We work with the cost function that penalizes the algebraic distance of the measurement data to the quadric,

$$f : SE_3 \rightarrow \mathbb{R},$$

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

Exploiting the relationship between the trace and  $\text{vec}$  operators, we can reformulated the cost function as

$$f(T) = \|B \text{vec}(T^\top Q T)\|^2, \quad B := \frac{1}{\sqrt{n}} \begin{bmatrix} \text{vec}^\top(p_1 p_1^\top) \\ \vdots \\ \text{vec}^\top(p_n p_n^\top) \end{bmatrix}. \quad (10)$$

A modification of this cost function taking into account noise statistics is considered in a later section. A geometric cost function gives improved results, but our techniques are more accessible for algebraic cost function, and space does not permit here a solution of the perhaps more useful geometric results.

#### C. 2-jet of the cost function expressed 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,

$$(f \circ \mu)(tx)|_{t=0} = \|B \text{vec}(A)\|^2, \quad (11)$$

(ii) a linear term,

$$\begin{aligned} \frac{d}{dt}(f \circ \mu)(tx) \Big|_{t=0} &= 2 \text{vec}^\top(A \zeta(x) + \zeta^\top(x) A) B^\top B \text{vec}(A), \\ &= 2x^\top \nabla(f \circ \mu)(0). \end{aligned} \quad (12)$$

By 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, and

$$C := B[(I \otimes A)(A \otimes I)] \begin{bmatrix} G \\ J \end{bmatrix}.$$

The explicit formula for the gradient of  $f \circ \mu$  evaluated at 0 is

$$\nabla(f \circ \mu)(0) = C^\top B \text{vec}(A), \quad (13)$$

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

$$\|B \text{vec}(A \zeta(x) + \zeta^\top(x) A)\|^2 = x^\top \hat{H}_{(f \circ \mu)(0)} x, \quad (14)$$

and the second term is

$$\begin{aligned} & \text{vec}^\top(A)B^\top B\text{vec}\left(A\zeta^2(x) + 2\zeta^\top(x)A\zeta(x) + \zeta^{\top^2}(x)A\right), \\ & = x^\top \tilde{H}_{(f \circ \mu)(0)} x. \end{aligned} \quad (15)$$

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

$$H_{(f \circ \mu)(0)} = \hat{H}_{(f \circ \mu)(0)} + \tilde{H}_{(f \circ \mu)(0)}, \quad (16)$$

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

$$\begin{aligned} \hat{H}_{(f \circ \mu)(0)} &= C^\top C \geq 0, \\ \tilde{H}_{(f \circ \mu)(0)} &= \begin{bmatrix} G^\top & J^\top \end{bmatrix} \begin{bmatrix} (D^\top \otimes A) & (D^\top A \otimes I) \\ (AD \otimes I) & (A \otimes D^\top) \end{bmatrix} \begin{bmatrix} G \\ J \end{bmatrix}. \end{aligned} \quad (17)$$

#### IV. THE ALGORITHM

The proposed algorithm consists of the iteration,

$$s = \pi_2 \circ \pi_1 : SE_3 \rightarrow SE_3, \quad (18)$$

where  $\pi_1$  maps a point  $T$  on the manifold  $SE_3$  to an element in the affine tangent space that minimizes  $j_0^{(2)}(f \circ \mu)(0)$  and  $\pi_2$  projects that element back to  $SE_3$  by means of the parametrization  $\mu$ .

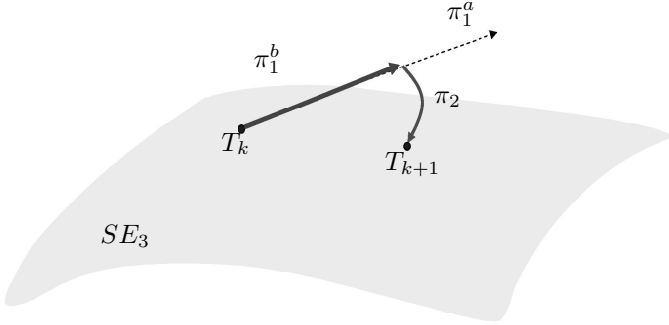


Fig. 3. The Algorithmic Mapping

##### A. 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

$$\pi_1 = \pi_1^b \circ \pi_1^a : SE_3 \rightarrow \mathbb{R}^{4 \times 4}. \quad (19)$$

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

$$\begin{aligned} x_{\text{opt}} : SE_3 \supset \mu(\mathcal{N}(0)) &\rightarrow \mathcal{N}(0) \subset \mathbb{R}^6, \\ x_{\text{opt}}(\mu(x)) &= \arg \min_{y \in \mathcal{N}(0)} j_0^{(2)}(f \circ \mu)(x). \end{aligned} \quad (20)$$

The Newton direction is given by,

$$x_{\text{opt}}(\mu(x)) = -H_{(f \circ \mu)(x)}^{-1} \nabla(f \circ \mu)(x), \quad (21)$$

or a Gauss direction,

$$x_{\text{opt}}(\mu(x)) = -\hat{H}_{(f \circ \mu)(x)}^{-1} \nabla(f \circ \mu)(x). \quad (22)$$

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 [12]. Thus,

$$\pi_1^b : \mathbb{R}^{4 \times 4} \rightarrow \mathbb{R}^{4 \times 4}, \quad (23)$$

$$T + T\zeta(x_{\text{opt}}) \mapsto T + T\zeta(\lambda_{\text{opt}} x_{\text{opt}}), \quad (24)$$

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.

##### B. 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  $\mu$ ,

$$\begin{aligned} \pi_2 : \mathbb{R}^{4 \times 4} &\rightarrow SE_3, \\ T + T\zeta(\lambda_{\text{opt}} x_{\text{opt}}) &\mapsto T e^{\zeta(\lambda_{\text{opt}} x_{\text{opt}})}. \end{aligned} \quad (25)$$

#### 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,

$$\phi : \mathcal{S}^{4 \times 4} \rightarrow \mathbb{R}, \quad S \mapsto \|B\text{vec}(S)\|^2. \quad (26)$$

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

$$\nu : \mathbb{R}^{10} \rightarrow \mathcal{S}^{4 \times 4}, \quad \nu(a) = \begin{bmatrix} a_1 & a_2 & a_3 & a_7 \\ a_2 & a_4 & a_5 & a_8 \\ a_3 & a_5 & a_6 & a_9 \\ a_7 & a_8 & a_9 & a_{10} \end{bmatrix}. \quad (27)$$

It is clear that this mapping is bijective. Now,

$$\phi \circ \nu : \mathbb{R}^{10} \rightarrow \mathbb{R}, \quad a \mapsto \|B\text{vec}(\nu(a))\|^2 = \|BKa\|^2, \quad (28)$$

where  $K$  is a matrix consisting of elements 1 and 0 so that  $Ka = \nu(a)$ . The vector  $a^*$  that minimizes the cost (28) 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

$$\bar{a}^\top C \bar{a} = 1, \quad \bar{a} = [a_1 \cdots a_6], \quad C := \text{diag}(1, 2, 2, 1, 2, 1). \quad (29)$$

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

$$\kappa T(R, t)^\top Q T(R, t) = A, \quad \text{for some scalar } \kappa, \quad (30)$$

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

$$Q_{11} = V_Q \Sigma_Q V_Q^\top, \quad A_{11} = V_A \Sigma_A V_A^\top, \quad (31)$$

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

$$\{R^i = V_Q \Gamma^i V_A^\top \in SO_3\}, \quad (32)$$

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.,

$$i^* = \text{argmin}_i \|R^i - I\|^2 = \text{argmax}_i \text{tr}(R^i). \quad (33)$$

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

$$t^* = Q_{11}^{-1}(\kappa^{-1} R A_{12} - Q_{12}), \quad \kappa = \frac{1}{3} \text{tr}(\Sigma_A \Sigma_Q^{-1}). \quad (34)$$

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. ALGORITHM IMPLEMENTATION

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

*Step 1:* Carry out the optimization step,

- Compute the gradient  $\nabla(f \circ \mu)(0)$  and the Hessian  $H_{(f \circ \mu)(0)}$  via (12), (16) respectively.

- If  $H_{(f \circ \mu)(0)} > 0$ ,

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

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

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

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

*Step 3:* Set  $T = \hat{T}$ , go back to Step 1 if  $\|\nabla(f \circ \mu)(0)\| > \epsilon$ , a prescribed accuracy.

## VII. CONVERGENCE ANALYSIS OF THE ALGORITHM

### A. Local Convergence

#### A.1 Local Quadratic Convergence

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

$$T_{k+1} = s(T_k), \quad (35)$$

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  $T_*$ .

Due to space limit, details of the proof will be omitted here.

### 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.

## 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).

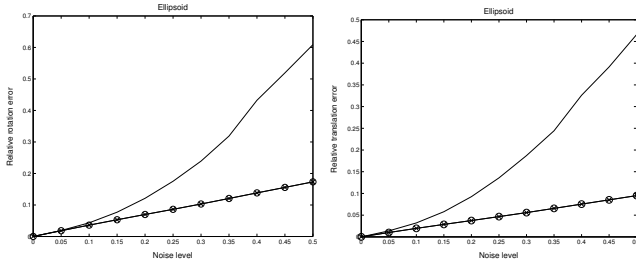
Cyclic coordinate descent is a variant of our proposed approach. It exploits the property that  $SE_3$  is the product manifold  $SO_3 \times \mathbb{R}^3$ . At each iteration, we first freeze the rotation and optimize only the translation, then next freeze the translation and optimize only the rotation. 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 potentially escape from a local minimum, not the global minimum in a few iterations, without any random reinitializations. The algorithm is not useful after the first few iterations since it converges only linearly and the analyzed line search has not then add value.

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

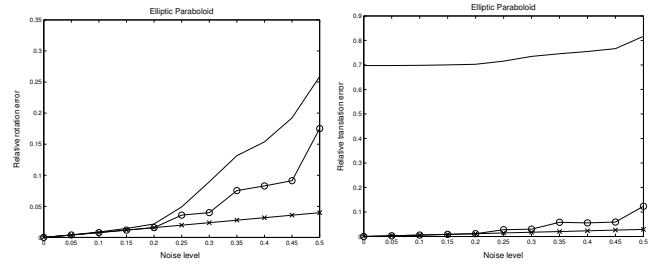
### A. Robustness Analysis

Here 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 1 indicates that GA is far less sensitive to additive Gaussian



(a)Relative pose error for ellipsoid.

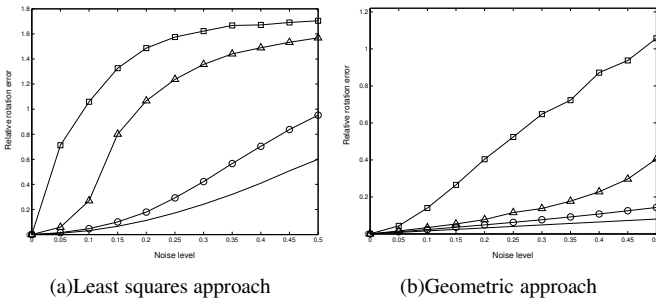


(b)Relative pose error for elliptic paraboloid.

Fig. 4. Robustness of the algorithms against additive Gaussian noise: closed form least squares approach (solid line), geometric approach without restart ( $\circ$ ), geometric approach with 5 random restart ( $\times$ )

noise and than LS. Interestingly, for an ellipsoid, the initialization achieves the global minimum in the presence of high noise level (Fig.1.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. 1.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 2 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.



(a)Least squares approach

(b)Geometric approach

Fig. 5. Robustness against occlusion: whole surface (solid line), half surface ( $\circ$ ), quarter surface ( $\Delta$ ), small patch ( $\square$ )

### B. Speed of local convergence

Figure 3 illustrates that GA converges at a local quadratic rate and CCD always converges very quickly at its first few iterations, but then converges linearly. It may make ‘sense’ to use CCD for the first few iterations then switch to the Newton method. Also, the local quadratic convergence rate GA is also better than the approach presented in [10] which claims to converge at an exponential rate.

## IX. CONCLUSION

We have presented a new geometric approach based on Newton-type iterations on special Euclidean group  $SE_3$  that is able to locate the 3D position and orientation of a quadratic surface quickly and accurately. The proposed algorithm is fast because each iteration is data independent and it converges locally quadratically fast to a minimum. It is robust to additive Gaus-

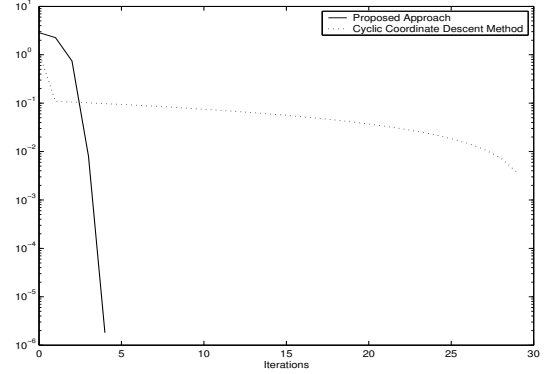


Fig. 6. Rate of local convergence

sian noise and occlusion. The notion of random initializations have been adopted to assist in escaping local minima. All these features have been demonstrated in simulations.

## 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] 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
- [11] R. M. Murray, Z. Li, and S. S. Sastry, *A Mathematical Introduction to Robotic Manipulation*, CRC Press, 1994.
- [12] J. Nocedal, S. J. Wright, *Numerical Optimization*, Springer Series in Operation Research, Springer Verlag, New York, 1999.