

---

# Differential Geometry Applications to Vision Systems

John B. Moore and Pei Yean Lee

The Australian National University and the National ICT Australia Ltd.  
John.Moore@anu.edu.au, PeiYean.Lee@nicta.com.au

## 1 Introduction

Integrating ideas from differential geometry and optimization, this chapter develops and applies novel parameterization-based framework to address optimization problems formulated on a smooth manifold, particularly those arise in computer vision. The framework views the manifold as a collection of local coordinate charts. It involves successive parameterizations of a manifold, carrying out optimization of the local cost function in parameter space and then projecting the optimal vector back to the manifold. Newton-type algorithms based on this approach are devised and its local quadratic convergent rate is established via mathematical proof.

*Outline.* In this section, we present some background knowledge about computer vision and optimization. Relevant literatures in the fields are reviewed in Section 2. We highlight the research issues, state the research goals and describe the proposed methodology in Section 3. This is followed by four computer vision tasks involving recovering pose information from images, addressed using proposed geometric optimization framework in Section 5–8. Full details of these applications, see Lee’s PhD thesis [18].

### 1.1 Computer Vision

As humans, our vision system allows us to understand the world that we see. Can computers compete? This becomes a central challenge of computer vision. In recent years, computers with high speed processors and larger memories at reduced prices have emerged. This, together with cheap off-the-shelf digital imaging devices have made computer vision system increasingly used for various real world applications.

In robotics, computer vision has been used by industrial robots for parts assembly and by autonomous vehicles as a means to localize and navigate. In the area of human computer interaction, computer vision algorithms are

used to create a user friendly computer interfaces. In the automobile industry, vision based systems can parallel park a car “automatically”. In the arts and entertainment industries, computer vision techniques are adopted to construct 3D computer models of real objects or scenes to make animation in games or movies more natural and realistic.

All applications discussed above involve recovering pose information from images, that is the position and orientation of the object. For example, mobile robots and intelligent cars need to know where the obstacle is in order to avoid it. Industrial robots need to know the pose of the parts in order to pick them up and assemble them correctly. Computer software needs to know how the head of the user is oriented in order to decide whether to scroll the page up or down. To construct a 3D computer model, multiple partial views of the real object need to be taken and the unknown pose information between the views need to be recovered. These applications motivate us to revisit the pose estimation problem using mathematical tools from differential geometry and optimization theory and thereby develop new enhanced algorithms that carefully exploit the underlying geometry of the problem.

## 1.2 Geometric Optimization

Various computer vision problems, including pose estimation can be addressed using an optimization framework. The optimization is usually constrained, and typically the first step is to transform the constrained problem into an unconstrained problem using the method of Lagrange multipliers. This approach is relatively simple and allows one to exploit the wealth of existing Euclidean optimization techniques. However, algorithms developed using this strategy usually only preserve the constraint asymptotically. Any premature termination of the algorithm may fail to satisfy the constraint set. Also, this approach always increases the dimensionality of the optimization.

An alternative strategy, known as geometric optimization, which we adopt, is to exploit the geometry of the underlying parameter space. When the underlying parameter space is non-Euclidean, as often occurs in constrained (vision) problems, then there can be advantage in exploiting the geometric structure of this space. That is, develop a strategy which views constrained problems as equivalent to unconstrained problems posed on the constraint sets. Unlike the traditional Lagrange multiplier method, this approach deals with the original problem directly and ensures that the constraints are fulfilled at every iteration. Philosophically, this approach is geometrically more intuitive, conceptually simpler and mathematically more elegant. From a practical perspective, geometric optimization ensures that the constraints are satisfied at every iteration. Thus, any premature termination still gives a meaningful solution.

Here, we focus on optimization problems formulated on manifolds. Generically, a manifold is a space which ‘locally looks like’ Euclidean space but does not do so globally. For example, Euclidean space is itself a manifold, as are

spheres and doughnuts. Since often a manifold can be implicitly defined by a set of equality constraints, it deals with a rather large class of optimization problems, for background theory of manifolds, see [19].

## 2 Literature Review

### 2.1 Riemannian Approach

Since manifolds are ‘locally’ Euclidean, it is not surprising that some nice Euclidean mathematical objects such as a straight line, gradient and Hessian of a cost function have their counterparts in a manifold setting. The Riemannian approach proceeds by first endowing manifolds with a metric structure. This gives rise to the corresponding mathematical objects being a geodesic, Riemannian gradient and Riemannian Hessian. Most existing numerical algorithms on manifolds have been proposed based on this approach.

*Gradient method on manifolds.* The gradient method was first extended to manifolds by Luenberger [23] in 1972, see also the work by Gabay [9]. In the early 1990s, gradient flow methods on manifolds were applied in the area of systems and control theory and linear algebra, see for example Brockett [4], Helmke and Moore [11], Smith [33], and Mahony [25]. See also independent work by Udriste [37].

*Newton’s method on manifolds.* The gradient method on manifolds, as for its Euclidean counterpart has a slow (linear) local convergence rate. Thus, it is not always suited for applications requiring online computation. As a result, many researchers turn their attention to the classical Newton’s method which is well known for its fast (quadratic) local convergence rate. In 1982, Gabay [9] proposed Newton’s method formulated on a submanifold of Euclidean space and each update is carried out along a geodesic. There is independent work by Shub [32] who defined Newton’s method for finding the zeros of a vector field on a manifold. Other independent work to extend Newton’s method for general optimization problems on Riemannian manifolds has been carried out by Smith [33] and Mahony [25] restricting to compact Lie group, and by Udriste [37] restricting to convex optimization problems on Riemannian manifolds. Edelman *et al.* [7] developed Newton’s method specific for orthogonality constraints, being then Stiefel and Grassmann manifolds. This method was later adapted by Ma *et al.* [24] for solving the structure-from-motion problem in computer vision. There is also a recent paper by Adler and Dedieu [2] which studied Newton’s method to finding a zero of a vector field defined on general Riemannian manifolds with applications in medicine.

*Quasi-Newton method on manifolds.* In Euclidean space, a variant of Newton’s method, known as the quasi-Newton method was introduced by Davidon [6] in late 1950s. The technique approximates Hessian inverses using first order gradient information and has a superlinear local convergence rate. To the

best of our knowledge, the paper by Gabay [9] is the first and only published work focusing on generalizing the quasi-Newton method to a manifold.

*Conjugate gradient method on manifolds.* In Euclidean space, a conjugate gradient technique was introduced by Hestenes and Stiefel [14] in the 1950s to solve large scale systems of linear equations. In 1960s, Fletcher and Reeves [8] extended the method to solve general unconstrained optimization problems. Thirty years or so later Smith [33] first generalized the method to manifolds. Since then, he also collaborated with Edelman and Arias [7] to study the method specifically on Grassmann and Stiefel manifolds.

*Others.* In Euclidean space, there is always a trade off between gradient methods (good global convergence properties but slow local convergence rate) and Newton’s method (fast local convergence rate but poor global convergence properties). Two strategies have been presented to either globalize the Newton’s method or speed up the gradient method, namely the line search and the trust region methods. Since both methods also exhibit similar properties in a manifold setting, Absil [1] has extended the trust region strategy to a manifold, while line search strategies on manifolds can be found in the work of Udriste [37] and Yang [39].

## 2.2 Non-Riemannian Approach

The non-Riemannian approach could well be the dominant approach in future. As pointed out in [28], the Riemannian approach is justified when the cost function is somewhat related to the Riemannian geometry and works best when the cost function is convex with respect to the Riemannian geometry. However, in applying this approach it may be difficult to find Riemannian metrics as in the case for non-compact Lie groups [26]. Moreover, many algorithms proposed based on this approach are either abstract conceptually and difficult to implement on a computer, or are computationally inefficient. Consequently, it is not suitable for many computer vision tasks which require online processing. In presenting our work using non-Riemannian approach, we should mention two independent prior works which present a similar framework to ours: The unpublished report by Taylor and Kriegman [36] and the paper by Manton [27] to address problems arise in computer vision and signal processing, respectively. Since our work, Hüper and Trumpf also presented a related approach in a recent paper [15].

## 3 Research Issue, Aims and Methodology

In our work, we seek to reduce the gap between mathematical development of the geometric optimization approach and its practical application. A key issue for us is that researchers develop rigorous abstract mathematical results using the tools of differential geometry and optimization, yet these are

not readily accessible to engineers and computer scientists, nor are they convinced of the practicality of applying this material, particularly to on-line applications. Only a handful of geometric optimization algorithms find applications in robotics [12], signal processing [7, 35], computer vision [24] and the medical applications [2]. Although algorithms on manifolds have evolved from abstract notions [9, 33, 25] to concrete implementable recursions [7, 24], they are simply not suitable for many real time applications. The majority of iterative computer vision algorithms merely adopt standard optimization techniques without exploiting the geometric structure of the underlying parameter space.

Our aims are threefold:

- To devise new iterative computer vision algorithms that exploit the geometric structure of the underlying parameter space, and explicitly preserve constraints at every iteration,
- To explore and extend a relatively new geometrical framework for optimization on manifolds that is simpler and more accessible to computer scientists and engineers,
- To develop the practical importance of geometric optimization algorithms as a tool that is not only implementable, but also suitable for real time applications, particularly in the area of computer vision.

To achieve these, we develop further a relatively new parametrization-based framework, focussing on the special class of problems arising in computer vision. Since a manifold ‘locally is’ Euclidean, at each iteration, a local parametrization of the manifold is constructed in Euclidean parameter space, there being a mapping both forwards and backwards between a neighborhood of a point on the manifold and about the origin in the Euclidean space. The optimal vector that maximizes or minimizes the cost function expressed in local Euclidean parameter space is then forward mapped to the manifold.

## 4 Application 1: 2D-3D Pose Estimation

In this section, we estimate the relative position and orientation between a known 3D object and its 2D image from a set of point correspondences. The task is formulated as minimizing a smooth function over the intersection of a rotation group and a cone. To address the problem, we exploit successive local parametrization of the constraint manifold. In contrast to the ‘fast’ linearly convergent geometric algorithm of [22], the Newton-like recursions devised in this way are locally quadratically convergent. The main novelty of our approach is the use of a closed-form global geodesic search step, which requires only the solution of a quartic equation. It assists in escaping local minimum and infeasible domains, as well as converges to a global minimum without the need to reinitialize the algorithm. Introduction of the Newton decrement as an

indicator for selection of gradient, Gauss, or Newton directions and for algorithm termination. For a prescribed number of iterations, the proposed algorithm achieves significantly lower pose estimation errors than earlier methods and it converges to a global minimum in typically 5–10 iterations.

*Outline.* In Section 4.1, the 2D-3D pose recovery task is formulated as an optimization problem cast on the intersection of rotation group and cone constraint. We devise a Newton-type algorithm evolving on this constraint manifold in Section 4.2. The proposed algorithm is summarized in Section 4.3. For simulation studies, mathematical proof of local quadratic convergence and a robust version of the algorithm for reducing the influence of outliers, see [20].

#### 4.1 Problem Formulation

The model of a known 3D object is a set of points described in an object centered frame that lie within the field of view of a camera, as

$$\{m_i\}_{i=1,\dots,n}, \quad m_i := [x_i \ y_i \ z_i]^\top \in \mathbb{R}^3.$$

To represent each model point in the camera centered frame  $m'_i := [x'_i \ y'_i \ z'_i]^\top$ , a rigid body transformation is performed as follows,

$$m'_i = Rm_i + t,$$

where  $R \in SO_3$  represents the rotation, and  $t \in \mathbb{R}^3$  is the translation vector. Each transformed model point  $m'_i$  is observed by a camera, as illustrated in Fig. 1. The point  $m'_i$  on the image plane is described in pixel coordinates, denoted  $p_i$ . Such an image point is then normalized by the camera calibration matrix  $F$ , assumed known, to obtain the corresponding normalized image point  $u_i = F^{-1}p_i$ . Under full perspective projection, each point in the model  $m_i$  is related to its corresponding normalized image point, as

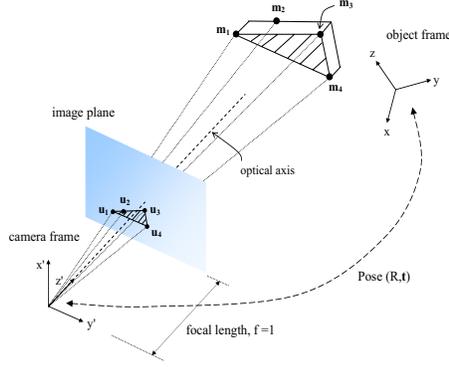
$$u_i = \frac{Rm_i + t}{z'_i}, \quad z'_i = e_3^\top (Rm_i + t), \quad e_3 := [0 \ 0 \ 1]^\top. \quad (1)$$

Here,  $\{z'_i\}_{i=1,\dots,n}$  are depth parameters which must satisfy the cone constraint  $\mathcal{K}$ , that is  $\{z'_i > 0\}_{i=1,\dots,n}$  to ensure that the estimated pose always locates the object in front of the camera.

#### Object Space Error

Instead of recovering the pose  $\{R, t\}$  using a least squares cost function penalizing the classical image space collinearity error via (1), we adopt the object space collinearity error introduced in [22]. Let  $U_i$  be the projection operator,

$$U_i = U_i^\top := \frac{u_i u_i^\top}{u_i^\top u_i}, \quad U_i^2 = U_i, \quad (2)$$



**Fig. 1.** The 2D-3D pose estimation problem: given the model  $\{m_1, \dots, m_4\}$  expressed in object frame, and its corresponding image  $\{u_1, \dots, u_4\}$  expressed in camera frame, find the relative pose  $(R, t)$  between the object frame and the camera frame.

the orthogonal projection of  $m'_i = Rm_i + t$  on to the line-of-sight of the corresponding image point  $u_i$  should be equal to  $m'_i$  itself, as

$$Rm_i + t = U_i(Rm_i + t), \quad i = 1, \dots, n. \quad (3)$$

In the presence of pixel noise, the cost function penalizing the object space collinearity error is given as,

$$\phi : SO_3 \times \mathbb{R}^3 \rightarrow \mathbb{R}, \quad \phi(R, t) = \frac{1}{2} \sum_{i=1}^n \|(I - U_i)(Rm_i + t)\|^2. \quad (4)$$

Observe that the cost function is quadratic in terms of errors, which are linear in the unknown pose parameters  $R, t$ . The optimization is made nontrivial by the constraints that  $R$  is an element of the manifold  $SO_3$  and the presence of a cone constraint  $\mathcal{K}$  resulting from the requirement that the object must be in front of the camera. Actually optimizing quadratic cost functions on  $SO_3$  is well studied, as is optimization on cones. What is interesting here from an optimization point of view is to tackle the much harder problem of optimizing a cost function on the intersection of  $SO_3$  and a cone.

### Optimal Translation

We eliminate the translation vector  $t$  from the cost function (4) via least squares optimization to reduce the number of parameters for optimization. Denoting

$$\tilde{U} := \sum_{i=1}^n (I - U_i), \quad \mathcal{U} = \tilde{U}^{-1} \sum_{i=1}^n (I - U_i)(m_i^\top \otimes I), \quad (5)$$

with  $R$  fixed, an optimal  $t$  that minimizes (4) is readily verified as

$$t = -\mathcal{U}\text{vec}(R), \quad (6)$$

Substituting (6) into (1), the depth parameters  $z'_i$  can be reformulated as,

$$z'_i = \mathcal{B}_i \text{vec}(R), \quad \mathcal{B}_i := e_3^\top ((m_i^\top \otimes I) - \mathcal{U}). \quad (7)$$

The cone constraint  $\mathcal{K}$  can now be expressed in terms of  $R \in SO_3$  as

$$\mathcal{K} := \{R \mid \{\mathcal{B}_i \text{vec}(R) > 0\}_{i=1,2,\dots,n}\}. \quad (8)$$

Substituting (6) into (4), the cost function can be reformulated as,

$$f : SO_3 \rightarrow \mathbb{R}, \quad f(R) = \frac{1}{2} \|\mathcal{D}\text{vec}(R)\|^2, \quad (9)$$

where

$$\mathcal{D} = [D_1^\top \ D_2^\top \ \dots \ D_n^\top]^\top, \quad D_i = (I - U_i) ((m_i^\top \otimes I) - \mathcal{U}). \quad (10)$$

## 4.2 Optimization on the Special Orthogonal Group

### Geometry of the Special Orthogonal Group

Rotational motion in  $\mathbb{R}^3$  can be represented by the special orthogonal group  $SO_3$ , which consists of  $3 \times 3$  orthogonal matrices with determinant  $+1$ . It is a Lie group and its associated Lie algebra  $\mathfrak{so}_3$  is the set of  $3 \times 3$  skew symmetric matrices. There is a well known isomorphism from the Lie algebra  $(\mathbb{R}^3, \times)$  to the Lie algebra  $(\mathfrak{so}_3, [.,.])$ , where  $\times$  denotes the cross product and  $[.,.]$  denotes the matrix commutator. This allows one to identify  $\mathfrak{so}_3$  with  $\mathbb{R}^3$  using the following mapping,

$$\Omega : \mathbb{R}^3 \rightarrow \mathfrak{so}_3, \quad \omega = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \mapsto \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}. \quad (11)$$

Notice that  $\Omega$  can be written as,

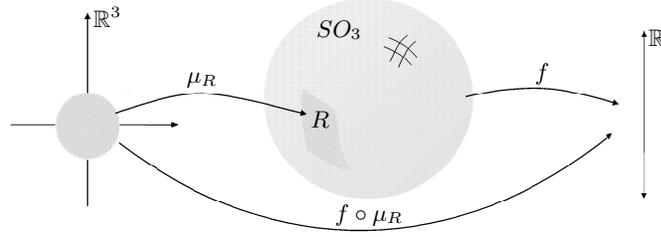
$$\Omega(\omega) = Q_x \omega_x + Q_y \omega_y + Q_z \omega_z, \quad (12)$$

where

$$Q_x := \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}, \quad Q_y := \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix}, \quad Q_z := \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \quad (13)$$

*Tangent Space of  $SO_3$ .* Consider the tangent space of  $SO_3$  at point  $R$ ,

$$\mathbb{T}_R SO_3 = \{R\Omega \mid \Omega \in \mathfrak{so}_3\}, \quad (14)$$



**Fig. 2.** The mapping  $\mu_R$  is a local parameterization of  $SO_3$  around the point  $R$  such that  $R = \mu_R(0)$ ,  $f$  is the smooth function defined on  $SO_3$  and  $f \circ \mu_R$  is the function  $f$  expressed in local parameter space  $\mathbb{R}^3$ .

and the affine tangent space of  $SO_3$  at the point  $R$  is given as,

$$T_R^{\text{aff}} SO_3 = \{R + R\Omega \mid \Omega \in \mathfrak{so}_3\}. \quad (15)$$

*Local Parameterization of  $SO_3$ .* Computations on a manifold are often conveniently carried out in local parameter spaces. Let  $R \in SO_3$ , there exist a smooth exponential map

$$\mu_R : \mathbb{R}^3 \rightarrow SO_3, \quad \omega \mapsto Re^{\Omega(\omega)}, \quad (16)$$

which is a local diffeomorphism around the origin in  $\mathbb{R}^3$ .

### Cost Function

*Cost Function on the Manifold  $SO_3$ .* Recall  $\mathcal{D}$  in (10) and the smooth function from (9). Minimizing this function penalizes the alignment error between all object points and the line-of-sight of the corresponding image points.

*Local Cost Function.* Consider the mappings as in Fig. 2. The cost function  $f$  at  $R \in SO_3$  expressed in local parameter space using the local parameterization  $\mu_R$  defined in (16) is,

$$f \circ \mu_R : \mathbb{R}^3 \rightarrow \mathbb{R}, \quad f \circ \mu_R(\omega) = \frac{1}{2} \|\mathcal{D}\text{vec}(Re^{\Omega(\omega)})\|^2. \quad (17)$$

*Quadratic Model of the Local Cost Function.* The second order Taylor approximation of  $f \circ \mu_R$  about  $0 \in \mathbb{R}^3$  in direction  $\omega$  consists of three terms:

- (i) A constant term  $(f \circ \mu_R)(t\omega)|_{t=0} = \frac{1}{2} \|\mathcal{D}\text{vec}(R)\|^2$ .
- (ii) A term linear in  $\omega$

$$\text{vec}^\top(R\Omega(\omega))\mathcal{D}^\top \mathcal{D}\text{vec}(R) = \omega^\top \nabla_{f \circ \mu_R}(0),$$

and by denoting  $Q := [\text{vec}(Q_x) \text{vec}(Q_y) \text{vec}(Q_z)]$ , the Euclidean gradient of the local cost function evaluated at zero is given as,

$$\nabla_{f \circ \mu_R}(0) = Q^\top (I \otimes R^\top) \mathcal{D}^\top \mathcal{D} \text{vec}(R), \quad (18)$$

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

$$\text{vec}^\top(R\Omega(\omega)) \mathcal{D}^\top \mathcal{D} \text{vec}(R\Omega(\omega)) = \omega^\top \widehat{H}_{f \circ \mu_R}(0) \omega, \quad (19)$$

and the second term is,

$$\text{vec}^\top(R) \mathcal{D}^\top \mathcal{D} \text{vec}(R\Omega^2(\omega)) = \omega^\top \widetilde{H}_{f \circ \mu_R}(0) \omega.$$

Thus, the Hessian matrix of  $f \circ \mu_R$  evaluated at  $0 \in \mathbb{R}^3$  is,

$$H_{f \circ \mu_R}(0) = \widehat{H}_{f \circ \mu_R}(0) + \widetilde{H}_{f \circ \mu_R}(0), \quad (20)$$

with  $\text{vec}(C) := \mathcal{D}^\top \mathcal{D} \text{vec}(R)$ , we have

$$\begin{aligned} \widehat{H}_{f \circ \mu_R}(0) &= Q^\top (I \otimes R^\top) \mathcal{D}^\top \mathcal{D} (I \otimes R) Q \geq 0, \\ \widetilde{H}_{f \circ \mu_R}(0) &= -\frac{1}{2} Q^\top ((I \otimes R^\top C) + (I \otimes C^\top R)) Q. \end{aligned} \quad (21)$$

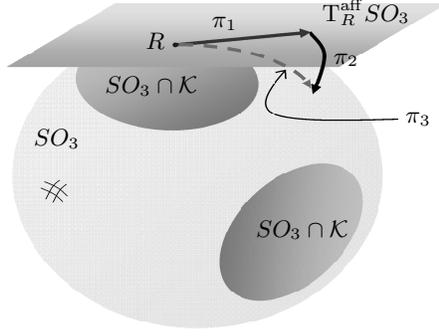
### Newton Decrement

The Newton decrement  $\delta$  is defined in terms of the gradient  $\nabla_{f \circ \mu_R}(0)$  and the Hessian  $H_{f \circ \mu_R}(0)$ , as

$$\delta := \sqrt{[\nabla_{f \circ \mu_R}(0)]^\top [H_{f \circ \mu_R}(0)]^{-1} \nabla_{f \circ \mu_R}(0)}. \quad (22)$$

This decrement  $\delta$  approaches zero as the algorithm converges to a local or global minimum. It features in the work of Nesterov [29] for optimizing convex self-concordant functions in Euclidean space. Recall that self concordant functions are those where the second derivative terms to the power  $\frac{3}{2}$  dominate third derivatives. A key result is that there is a *domain of attraction* for the Newton step using a unity step size if  $\delta < \frac{3-\sqrt{5}}{2}$ , a global constant.

Although the theory of [29] does not apply immediately for optimization on a manifold, yet since manifolds are locally Euclidean, it can be used as a guideline. In [16], the notion of convex self-concordant functions is explored in a manifold setting. Here after some manipulations, it can be shown that on  $SO_3$ , in the neighbourhood of a minimum, the cost function  $f$  is locally convex and self-concordant. Thus here we conservatively estimate a *domain of attraction* as  $\delta \ll \frac{3-\sqrt{5}}{2}$ . Here as curvature of the manifold increases, it makes sense to use more conservative values. We use Newton decrement as an indicator on selecting the appropriate direction of geodesic search. It is also used to assist in the decision for using a Newton step size instead of carrying out geodesic search and for algorithm termination.



**Fig. 3.** Each iteration of the proposed algorithm consists of three mappings, namely  $\pi_1$  maps a point  $R \in (SO_3 \cap \mathcal{K})$  to an element of the affine tangent space  $T_R^{\text{aff}} SO_3$ , followed by  $\pi_2$  which projects that vector back to the manifold and  $\pi_3$  which carries out geodesic search on  $SO_3$  in the direction of the projected vector.

### Algorithm

Each iteration of the proposed algorithm consists of three mappings as,

$$s = \pi_3 \circ \pi_2 \circ \pi_1 : (SO_3 \cap \mathcal{K}) \rightarrow (SO_3 \cap \mathcal{K}). \quad (23)$$

At each iteration, a local parameterization  $\mu_R$  of the manifold around  $R \in (SO_3 \cap \mathcal{K})$  is constructed. The point  $R$  is pulled back to the Euclidean space via  $\mu_R$ . The optimal vector that minimizes the quadratic model of the local cost function  $f \circ \mu_R$ , achieved by the operation  $\pi_1$ , is then pushed forward to the manifold via the mapping  $\pi_2$ . Finally, in operation  $\pi_3$ , a one dimensional search along the geodesic on  $SO_3$  in the direction of this projected vector is carried out to ensure cone constraint is satisfied. By appropriately identifying the local parameter space  $\mathbb{R}^3$  with the affine tangent space  $T_R^{\text{aff}} SO_3$ , the first two steps of the algorithm can also be interpreted geometrically as carrying out an optimization procedures defined on  $T_R^{\text{aff}} SO_3$ , followed by a nonlinear projection back to the manifold to give a geodesic search direction, as illustrated in Fig. 3.

*Optimization in Local Parameter Space.* Consider the optimization step,

$$\pi_1 : (SO_3 \cap \mathcal{K}) \rightarrow (SO_3 \cap \mathcal{K}, T^{\text{aff}} SO_3), \quad R \mapsto (R, R + R\Omega(\omega_{\text{opt}}(R))), \quad (24)$$

where  $\omega_{\text{opt}}$  as a function of  $R = \mu_R(0)$  is a suitable descent direction of  $f$  expressed in local parameter space. Actually, three possibilities for the  $\omega_{\text{opt}}$  calculation are of interest to our algorithm.

1. The Newton direction,

$$\omega_{\text{opt}}^{\text{Newton}}(R) = -[H_{f \circ \mu_R}(0)]^{-1} \nabla_{f \circ \mu_R}(0). \quad (25)$$

2. The Gauss direction is given as,

$$\omega_{\text{opt}}^{\text{Gauss}}(R) = -[\widehat{\mathbf{H}}_{f \circ \mu_R}(0)]^{-1} \nabla_{f \circ \mu_R}(0). \quad (26)$$

3. The negative gradient direction,

$$\omega_{\text{opt}}^{\text{gradient}}(R) = -\nabla_{f \circ \mu_R}(0). \quad (27)$$

*Remark 1.* When  $\mathbf{H}_{f \circ \mu_R}(0)$  or  $\widehat{\mathbf{H}}_{f \circ \mu_R}(0)$  is singular, pseudo inverse replaces inverse in the above equations.

*Projecting Back via Local Parameterization.* The mapping  $\pi_2$  projects the optimal affine tangent vector back to the manifold by means of the local parameterization  $\mu_R$ ,

$$\begin{aligned} \pi_2 : \left( (SO_3 \cap \mathcal{K}), \mathbf{T}^{\text{aff}} SO_3 \right) &\rightarrow SO_3, \\ (R, R + R\Omega(\omega_{\text{opt}}(R))) &\mapsto Re^{\Omega(\omega_{\text{opt}}(R))}. \end{aligned} \quad (28)$$

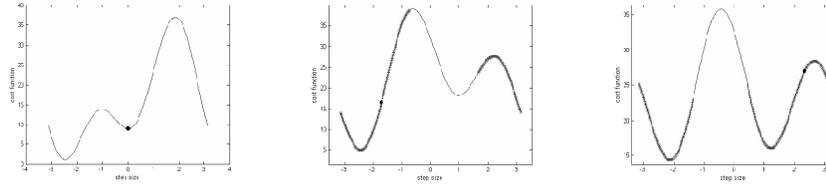
*Analytic Geodesic Search on  $SO_3$ .* The mapping  $\pi_3$  involves a one dimensional search along the geodesic curve,

$$\pi_3 : SO_3 \rightarrow SO_3 \cap \mathcal{K}, \quad Re^{\Omega(\omega_{\text{opt}}(R))} \mapsto Re^{\theta_{\text{opt}} \Omega(\omega_{\text{opt}}(R))}, \quad (29)$$

where  $\theta_{\text{opt}}$  is the optimum step size which minimizes cost function along the geodesic, as well as satisfying the cone constraint.

Certain Newton-type algorithms use a heuristic line search in a particular direction to ensure that the cost function decreases at every step. When the optimization is on a manifold, the line search translates to a geodesic search, and in the case of manifold  $SO_3$ , a finite range search. In ill-conditioned problems, as arise in high noise, the number of trial step sizes can be very large for many iterations until the optimization path steers clear of the boundary or saddle point or other sources of ill-conditioning. This is one motivation for us to use an analytic geodesic search, since this is possible for the manifold  $SO_3$ . The other motivations are to avoid violating the cone constraint  $\mathcal{K}$  and to assist in achieving the global minimum, rather than some other local minimum. The proposed analytic geodesic search is described on an *arbitrary geodesic* on  $SO_3$ . It involves the solution of a quartic equation derived using Rodrigues' rotation formula, see [20] for details.

Figure 4 shows the plots of cost function  $\varphi(\theta)$  vs. step size  $\theta$  in the range  $[-\pi, \pi]$ . It illustrates the idea of an analytic geodesic search. The black dot in each plot indicates a Newton step,  $\theta = \|\omega_{\text{opt}}\|$  in the search direction. The darkened portion of the curves represents the infeasible region where the cone constraint fails. In Fig. 4(a), there is no infeasible region. If a Newton step is taken, one will be trapped in the local minimum. However, the analytic geodesic search will select the  $\theta$  corresponding to the minimum cost in the



(a) No infeasible region, but if Newton step is taken, one will be trapped in local minimum. (b) Both minimum cost and Newton step lie in infeasible region. (c) All minima lie in infeasible region.

**Fig. 4.** Analytic geodesic search: plots of cost function  $f$  restricted to the geodesic on  $SO_3$  for various step size in the range  $[-\pi, \pi]$ , the black dot indicates Newton step, darkened portion of the curves represents the infeasible region.

search direction and hence escapes from the local minimum and heads towards the global minimum. In Fig. 4(b), the minimum cost lies in the infeasible region, so the geodesic search will select the  $\theta$  corresponding to the second minimum cost value that is in the feasible region. Also, the search directions might not yield any feasible local minimum, as depicted in Fig. 4(c). In this case, no parameter update is made. Nevertheless, by carrying out a search in a random direction periodically, the probability of achieving a feasible local minimum, or indeed a global minimum, is increased.

### Convergence Analysis of Algorithm.

*Global Convergence* The algorithm is such that the non-negatively valued cost function  $f$  decreases monotonically at each iteration, and thus converges. Consequently, point iterate  $R_k$  converges to  $\{R \in SO_3 \mid f(R) = c\}$  for non-negative scalar  $c$ . However, since  $f$  is smooth on  $SO_3$ , the algorithm step given by the gradient or Gauss or Newton direction is downhill on  $f$  and zero when the gradient is zero. Thus  $R_k$  converges to a critical point of  $f$ , when its gradient is zero. Of course, critical points of  $f$  where the gradient is zero are fixed points of the algorithm. However, critical points other than local minima are unstable fixed points. That is, small random perturbations of the algorithm from these points will result in further iterations which give cost reductions. The algorithm is designed to escape local minima that are not the global minima by virtue of the geodesic searches including periodic geodesic searches in a random direction.

*Local Quadratic Convergence at Global Minimum.*

**Theorem 1.** *Let  $R_* \in SO_3$  be the unique and nondegenerate global minimum of the smooth function  $f : SO_3 \rightarrow \mathbb{R}$  defined in (9). Let  $R_k$  be a point in an open neighbourhood of  $R_*$ . Consider the proposed iteration on  $SO_3$ ,*

$$R_{k+1} = s(R_k), \quad s = \pi_3 \circ \pi_2 \circ \pi_1, \quad (30)$$

where  $\pi_1$  is given by the Newton direction defined in (25),  $\pi_2$  involves projection back to  $SO_3$  via the smooth exponential map of (28), and  $\pi_3$  is an analytic geodesic search described in (29). Then the point sequence generated by  $s$  converges quadratically to  $R_*$ .

*Proof.* See [20].

### 4.3 Summary of Algorithm

Start with  $R = R_0$  using initialization procedures presented [20].

Step 1: Carry out the optimization step,

- Compute the gradient  $\nabla_{f \circ \mu_R}(0)$  and the Hessian matrix,  $H_{f \circ \mu_R}(0)$  via (18) and (20) respectively.
- Calculate the Newton decrement  $\delta$  from (22). If  $\delta < \epsilon_4$ , go to Step 5,
- Compute the optimal direction in the local parameter space,

$$\omega_{\text{opt}}(R) = \begin{cases} \omega_{\text{opt}}^{\text{gradient}} = -\nabla_{f \circ \mu_R}(0), & \text{if } \delta \geq \epsilon_1 \\ \omega_{\text{opt}}^{\text{Gauss}} = -[\widehat{H}_{f \circ \mu_R}(0)]^\dagger \nabla_{f \circ \mu_R}(0), & \text{if } \epsilon_2 < \delta < \epsilon_1 \\ \omega_{\text{opt}}^{\text{Newton}} = -[H_{f \circ \mu_R}(0)]^\dagger \nabla_{f \circ \mu_R}(0), & \text{if } \delta < \epsilon_2 \\ \omega_{\text{opt}}^{\text{rand}}, & \text{periodically, or if any of the above direction} \\ & \text{ends up in an infeasible region} \end{cases}$$

Here,  $\omega_{\text{opt}}^{\text{rand}} \in \mathbb{R}^3$  is a random vector with elements, in the range  $[0, 1]$ ,

- Form the normalized direction  $\bar{\omega}_{\text{opt}}(R) = \frac{\omega_{\text{opt}}(R)}{\|\omega_{\text{opt}}(R)\|}$ .

Step 2: Projecting back to the manifold  $SO_3$  via  $\pi_2$ ,

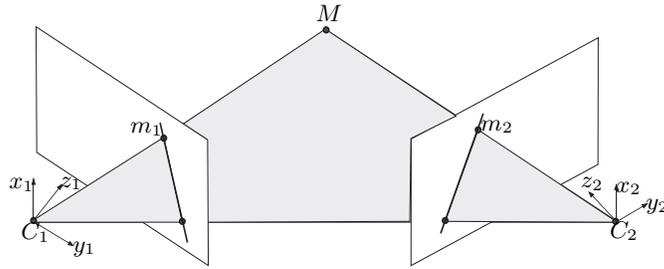
Step 3: Carry out a one dimensional search along the geodesic  $Re^{\theta\Omega(\bar{\omega}_{\text{opt}}(R))}$ , see [20] for details.

Step 4: Set  $R = \widehat{R}$ , go back to Step 1,

Step 5: The pose estimates are  $R$  and  $t = -\mathcal{U}\text{vec}(R)$  respectively.

## 5 Application 2: Estimation of the Essential Matrix

The pose information between a pair of calibrated 2D images is algebraically captured by an essential matrix. Geometric optimization techniques using Riemannian geodesics on the essential manifold have been presented by Ma *et al.* [24], which are adapted from [7]. With the view to improving on these algorithms, we adopt the non-Riemannian approach of the previous section. The algorithm requires relatively little computational effort per iteration, and



**Fig. 5.** The epipolar geometry

is independent of the number of point correspondences used, Our algorithm achieves the same rate of local convergence as Ma *et al.* [24] but is computationally simpler with effort per iteration being significantly lower, see [18].

*Outline.* In Section 5.1, the essential matrix estimation task is formulated as minimizing a smooth function on the essential manifold. Details of the optimization procedures are presented in Section 5.2 and the proposed algorithm is summarized in Section 5.3. The algorithm converges locally quadratically, as demonstrated via simulation results and mathematical proof, which can be found in [13, 18].

### 5.1 Problem Formulation

Two images of the same scene are related by epipolar geometry as illustrated in Fig. 5. The images can be taken by two cameras or the images can be taken by a mobile camera at two different positions. Given an object point  $M$  and its two dimensional projections  $m_1$  and  $m_2$  on both image planes with camera centers  $C_1$  and  $C_2$  and with known internal camera parameters, the epipolar constraint can be represented algebraically in terms of a  $3 \times 3$  matrix, called the essential matrix  $E$ , as

$$m_2^T E m_1 = 0. \quad (31)$$

The essential matrix encapsulates both the rotation and translation parameters, thus it plays a crucial role in motion analysis. It is known that with 8 or more generic point correspondences in the noise-free case, the essential matrix is uniquely determined.

### 5.2 Optimization on the Essential Manifold

#### Geometry of Essential Manifold

Recall, that an essential matrix is a real  $(3 \times 3)$ -matrix in factored form

$$E = \Omega R, \quad (32)$$

where  $R \in SO_3$  is a rotation matrix and  $\Omega \in \mathfrak{so}_3$ , as defined in (11) represents the translation vector between two views. It is well-known, that the essential matrices are characterized by the property that they have exactly one positive singular value of multiplicity two, consequently  $E$  must be rank 2 [10]. In particular, normalized essential matrices are those of Frobenius norm equal to  $\sqrt{2}$  and which are therefore characterized by having the set of singular values  $\{1, 1, 0\}$ , as

$$\mathcal{E} := \{UE_0V^\top \mid U, V \in SO_3\}, \quad E_0 := \begin{bmatrix} I_2 & 0 \\ 0 & 0 \end{bmatrix}. \quad (33)$$

*Tangent Space of Essential Manifold.* The tangent space at the normalized essential matrix  $E = UE_0V^\top$  is

$$T_E\mathcal{E} = \{U(\Omega E_0 - E_0\Psi)V^\top \mid \Omega, \Psi \in \mathfrak{so}_3\}, \quad (34)$$

The affine tangent space at  $E = UE_0V^\top \in \mathcal{E}$  can be identified with tangent space  $T_E\mathcal{E}$  via translating  $T_E\mathcal{E}$  by  $E$ , as

$$T_E\mathcal{E} = \{U(E_0 + \Omega E_0 - E_0\Psi)V^\top \mid \Omega, \Psi \in \mathfrak{so}_3\}, \quad (35)$$

*Parameterization of Essential Manifold.* Let  $U, V \in SO_3$  be arbitrary, let  $x^\top = [x_1, \dots, x_5] \in \mathbb{R}^5$ , and let  $E_0$  be defined as in (33). Consider the mappings

$$\Omega_1 : \mathbb{R}^5 \rightarrow \mathfrak{so}_3, \quad [x_1, \dots, x_5]^\top \mapsto \frac{1}{\sqrt{2}} \begin{bmatrix} 0 & -\frac{x_3}{\sqrt{2}} & x_2 \\ \frac{x_3}{\sqrt{2}} & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix}, \quad (36)$$

and

$$\Omega_2 : \mathbb{R}^5 \rightarrow \mathfrak{so}_3, \quad [x_1, \dots, x_5]^\top \mapsto \frac{1}{\sqrt{2}} \begin{bmatrix} 0 & \frac{x_3}{\sqrt{2}} & x_5 \\ -\frac{x_3}{\sqrt{2}} & 0 & -x_4 \\ -x_5 & x_4 & 0 \end{bmatrix}. \quad (37)$$

Consider also

$$\mu_E : \mathbb{R}^5 \rightarrow \mathcal{E}, \quad x \mapsto U e^{\Omega_1(x)} E_0 e^{-\Omega_2(x)} V^\top. \quad (38)$$

Then the mapping  $\mu_E$  is a local diffeomorphism around  $0 \in \mathbb{R}^5$ .

### Cost Function

*Cost Function on the Essential Manifold.* Let  $M^{(i)} := m_1^{(i)} m_2^{(i)\top}$ , where  $m_1^{(i)}, m_2^{(i)} \in \mathbb{R}^3$  correspond to the normalized  $i^{\text{th}}$  point image pair in the left and in the right camera, respectively, for which the correspondence is assumed to be known. Consider the smooth function

$$f : \mathcal{E} \rightarrow \mathbb{R}, \quad f(E) = \frac{1}{2} \sum_{i=1}^n \left( m_2^{(i)\top} E m_1^{(i)} \right)^2 = \frac{1}{2} \sum_{i=1}^n \text{tr}^2(M^{(i)} E). \quad (39)$$

The value of this cost function attains zero if and only if there is an essential matrix which fulfills the epipolar constraint for each image point pair. In the noisy case the zero value will in general not be attained. Thus, we search for the minima of this cost function.

*Local Cost Function.* The cost function  $f$  at  $E \in \mathcal{E}$  expressed in local parameter space using the local parameterization  $\mu_E$  is,

$$f \circ \mu_E : \mathbb{R}^5 \rightarrow \mathbb{R}, \quad f \circ \mu_E(x) = \frac{1}{2} \sum_{i=1}^n \text{tr}^2(M^{(i)} U e^{\Omega_1(x)} E_0 e^{\Omega_2(x)\top} V^\top). \quad (40)$$

*Quadratic Model of the Local Cost Function.* The second order Taylor expansion of  $f \circ \mu_E$  around the point  $E = U E_0 V^\top \in \mathcal{E}$  consists of three terms:

- (i) A constant term,  $(f \circ \mu_E)(tx)|_{t=0} = \frac{1}{2} \sum_{i=1}^n \text{tr}^2 M^{(i)} E$ ,
- (ii) A linear term,

$$\sum_{i=1}^n \left( \text{tr} M^{(i)} E \right) \left( \text{tr} M^{(i)} U \left( \Omega_1(x) E_0 - E_0 \Omega_2(x) \right) V^\top \right) = \nabla_{f \circ \mu_E}(0)^\top x,$$

which can be interpreted as the transposed Euclidean gradient of  $f \circ \mu_E : \mathbb{R}^5 \rightarrow \mathbb{R}$  evaluated at zero acting on  $x \in \mathbb{R}^5$ . Recall  $Q_x, Q_y, Q_z$  from (13) and let

$$\begin{aligned} \text{vec}(\Omega_1(x)) &:= Q_1 x, \quad Q_1 := \left[ \text{vec}\left(\frac{1}{\sqrt{2}} Q_x\right) \text{vec}\left(\frac{1}{\sqrt{2}} Q_y\right) \text{vec}\left(\frac{1}{2} Q_z\right) 0 \ 0 \right] \in \mathbb{R}^{9 \times 5}, \\ \text{vec}(\Omega_2(x)) &:= Q_2 x, \quad Q_2 := \left[ 0 \ 0 \ \text{vec}\left(\frac{1}{2} Q_z\right) \text{vec}\left(\frac{1}{\sqrt{2}} Q_x\right) \text{vec}\left(\frac{1}{2} Q_y\right) \right] \in \mathbb{R}^{9 \times 5}, \\ J &:= \mathcal{M}(V \otimes U) \left[ (E_0 \otimes I) - (I \otimes E_0) \right] \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix}, \quad \mathcal{M} := \begin{bmatrix} \text{vec}^\top(M^{(1)\top}) \\ \vdots \\ \text{vec}^\top(M^{(n)\top}) \end{bmatrix}, \end{aligned}$$

the explicit formula for the Euclidean gradient of the local cost function evaluated at zero is,

$$\nabla_{f \circ \mu_E}(0) = J^\top \mathcal{M} \text{vec}(E), \quad (41)$$

- (iii) A quadratic term in  $x$ , which consists of a sum of two terms. The first one,

$$\sum_{i=1}^n \text{tr}^2 M^{(i)} U \left( \Omega_1(x) E_0 - E_0 \Omega_2(x) \right) V^\top = x^\top \hat{\text{H}}_{f \circ \mu_E}(0) x,$$

is a quadratic form on  $\mathbb{R}^5$  with the corresponding matrix  $\hat{\text{H}}_{f \circ \mu_E}(0)$  being positive (semi)definite for all  $U, V \in \mathcal{SO}_3$ . The second term is given as,

$$\begin{aligned} & \sum_{i=1}^n \left( \text{tr } M^{(i)} E \right) \left( \text{tr } M^{(i)} U \left( \Omega_1^2(x) E_0 + E_0 \Omega_2^2(x) - 2\Omega_1(x) E_0 \Omega_2(x) \right) V^\top \right), \\ & = x^\top \tilde{\mathbf{H}}_{f \circ \mu_E}(0) x. \end{aligned}$$

Hence, the Hessian matrix of the local cost function evaluated at zero is,

$$\mathbf{H}_{f \circ \mu_E}(0) = \hat{\mathbf{H}}_{f \circ \mu_E}(0) + \tilde{\mathbf{H}}_{f \circ \mu_E}(0), \quad (42)$$

where

$$\hat{\mathbf{H}}_{f \circ \mu_E}(0) = J^\top J \geq 0, \quad (43)$$

and by denoting  $\text{vec}(D) := (V^\top \otimes U^\top) \mathcal{M}^\top \mathcal{M} \text{vec}(E)$ , we have

$$\tilde{\mathbf{H}}_{f \circ \mu_E}(0) = \begin{bmatrix} Q_1^\top & Q_2^\top \end{bmatrix} \begin{bmatrix} -(DE_0 \otimes I) & (D \otimes E_0) \\ (D^\top \otimes E_0) & -(E_0 D \otimes I) \end{bmatrix} \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix}. \quad (44)$$

### Algorithm

The proposed algorithm is a self map

$$s = \pi_2 \circ \pi_1 : \mathcal{E} \rightarrow \mathcal{E}, \quad (45)$$

which consists of two partial steps. The first one being an optimization procedure defined on an appropriate affine tangent space to  $\mathcal{E}$  (by identifying  $\mathbb{R}^5$  with  $T_E^{\text{aff}} \mathcal{E}$  appropriately), and the second one is a nonlinear projection back to the manifold.

*Optimization in Local Parameter Space.* The mapping  $\pi_1$  constitutes the optimization step, as

$$\begin{aligned} \pi_1 : \mathcal{E} & \rightarrow (\mathcal{E}, T^{\text{aff}} \mathcal{E}), \\ E = U E_0 V^\top & \mapsto (E, U(E_0 + \Omega_1(x_{\text{opt}}(E)) E_0 - E_0 \Omega_2(x_{\text{opt}}(E))) V^\top), \end{aligned} \quad (46)$$

where  $x_{\text{opt}} \in \mathbb{R}^5$  as a function of  $E = \mu_E(0)$  can be given by the Newton direction when  $\mathbf{H}_{f \circ \mu_E}(0) > 0$ ,

$$x_{\text{opt}}^{\text{Newton}}(E) = -[\mathbf{H}_{f \circ \mu_E}(0)]^{-1} \nabla_{f \circ \mu_E}(0), \quad (47)$$

or a Gauss direction otherwise,

$$x_{\text{opt}}^{\text{Gauss}}(E) = -[\hat{\mathbf{H}}_{f \circ \mu_E}(0)]^{-1} \nabla_{f \circ \mu_E}(0). \quad (48)$$

*Projecting onto the Essential Manifold.* The mapping  $\pi_2$  involves projecting back via local parameterization. For fixed  $x_{\text{opt}} \in \mathbb{R}^5$  consider the smooth mapping,

$$\begin{aligned} \pi_2 : (\mathcal{E}, T^{\text{aff}} \mathcal{E}) & \rightarrow \mathcal{E}, \\ (E, U(E_0 + \Omega_1(x_{\text{opt}}(E)) E_0 - E_0 \Omega_2(x_{\text{opt}}(E))) V^\top) & \\ & \mapsto U e^{\Omega_1(x_{\text{opt}}(E))} E_0 e^{-\Omega_2(x_{\text{opt}}(E))} V^\top. \end{aligned} \quad (49)$$

### 5.3 Summary of Algorithm

Start with  $E = UE_0V^\top$  obtained from the standard 8-point algorithm [10].

Step 1. Carry out the optimization step  $\pi_1$ ,

- Compute the gradient  $\nabla_{f_{o\mu E}}(0)$  and the Hessian matrix  $H_{f_{o\mu E}}(0)$  via (41), (42) respectively.
- If  $H_{f_{o\mu E}}(0) > 0$ , compute the Newton step  $x_{\text{opt}} = -[H_{f_{o\mu E}}(0)]^{-1}\nabla_{f_{o\mu E}}(0)$ , otherwise compute the Gauss step  $x_{\text{opt}} = -[\widehat{H}_{f_{o\mu E}}(0)]^{-1}\nabla_{f_{o\mu E}}(0)$ .

Step 2. Carry out the projection step  $\pi_2 : \widehat{U} = U e^{\Omega_1(x_{\text{opt}})}$ ,  $\widehat{V} = V e^{\Omega_2(x_{\text{opt}})}$ ,  $\widehat{E} = \widehat{U}E_0\widehat{V}^\top$ .

Step 3. Set  $E = \widehat{E}$ ,  $U = \widehat{U}$ ,  $V = \widehat{V}$ , go back to Step 1 if  $\|\nabla_{f_{o\mu E}}(0)\| > \varepsilon$ .

## 6 Application 3: 3D Localization of Quadratic Surfaces

Quadratic surfaces are commonly occurring shapes in man made objects. Our approach can address the problem of locating quadratic surfaces from 3D range image data as minimizing a smooth function over the special Euclidean group. The optimization is based on locally quadratically convergent Newton-type iterations on this constraint manifold. To achieve this, analysis of the underlying geometric constraint is required. Analysis shows that the proposed algorithm is relatively robust against additive Gaussian noise and occlusion.

*Outline.* In Section 6.1, the task of locating quadratic surfaces is addressed as an optimization problem evolving on the special Euclidean group. Details of the optimization techniques can be found in Section 6.2 and the proposed algorithm implementation is summarized in Section 6.3. Full details of this section can be found in [21].

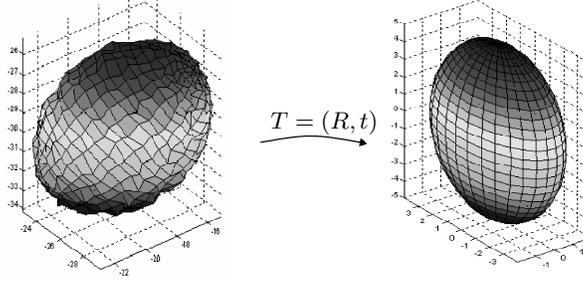
### 6.1 Problem Formulation

A quadric, also known as quadratic surface, is 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 quadric is given by

$$m^\top Qm = 0, \quad m := \begin{bmatrix} \tilde{m} \\ 1 \end{bmatrix}, \quad Q := \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{12}^\top & Q_{22} \end{bmatrix}, \quad (50)$$

where  $Q$  is the symmetric surface coefficient matrix. Without loss of generality, we take  $\text{tr}(Q_{11}^\top Q_{11}) = 1$ . Now, consider the quadric being rotated by  $R \in SO_3$  and translated by  $t \in \mathbb{R}^3$ . Each point on the transformed quadric  $\tilde{p} \in \mathbb{R}^3$  is given by,

$$\left\{ p = \begin{bmatrix} \tilde{p} \\ 1 \end{bmatrix} \in \mathbb{R}^4 \mid p^\top A(R, t)p = 0 \right\}. \quad (51)$$



**Fig. 6.** Given noisy measurement data of a quadric (ellipsoid) and its CAD model, find the position and orientation of the transformed quadric.

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

$$T(R, t) := \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \in SE_3. \quad (52)$$

which represents 3D rigid body transformation is an element of special Euclidean group,  $SE_3$ . Thus, 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 (51).

## 6.2 Optimization on the Special Euclidean Group

### Geometry of the Special Euclidean Group

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

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

where  $SO_3$  is the group of  $3 \times 3$  rotation matrices, as introduced in Section ??, and  $t$  is the translation vector. The special Euclidean group is a Lie group and its associated Lie algebra is denoted  $\mathfrak{se}_3$ . Due to isomorphism, one can identify  $\mathfrak{se}_3$  with  $\mathbb{R}^6$  using the following mapping,

$$\zeta : \mathbb{R}^6 \rightarrow \mathfrak{se}_3, \quad \zeta(x) = \begin{bmatrix} \Omega(\omega) & v \\ 0 & 0 \end{bmatrix}, \quad x := \begin{bmatrix} \omega \\ v \end{bmatrix}, \quad (53)$$

where  $\Omega(\omega)$  as defined in (11) and  $v \in \mathbb{R}^3$ .

*Tangent Space of  $SE_3$ .* The tangent space of  $SE_3$  at  $T$  is

$$\mathbb{T}_T SE_3 = \{T\zeta \mid \zeta \in \mathfrak{se}_3\}, \quad (54)$$

and the affine tangent space of  $SE_3$  at  $T$  is

$$\mathbb{T}_T^{\text{aff}} SE_3 = \{T + T\zeta \mid \zeta \in \mathfrak{se}_3\}. \quad (55)$$

*Local Parameterization of  $SE_3$ .* For every point  $T \in SE_3$ , there exists a smooth exponential map

$$\mu_T : \mathbb{R}^6 \rightarrow SE_3, \quad x \mapsto Te^{\zeta(x)}, \quad (56)$$

which is a local diffeomorphism around origin in  $\mathbb{R}^6$ . Such a map will be known as a local parameterization of  $SE_3$  around  $T$ .

### Cost Function

*Cost Function on the Manifold  $SE_3$ .* We work with the cost function that penalizes the algebraic distance of the measurement data to the quadric,

$$f : SE_3 \rightarrow \mathbb{R}, \quad f(T) = \frac{1}{n} \sum_{i=1}^n (p_i^\top T^\top Q T p_i)^2 = \|B\text{vec}(T^\top Q T)\|^2. \quad (57)$$

where  $B := \frac{1}{\sqrt{n}} [\text{vec}(p_1 p_1^\top) \cdots \text{vec}(p_n p_n^\top)]$ .

*Local Cost Function.* The cost function  $f$  at  $T \in SE_3$  expressed in local parameter space using the smooth local parameterization  $\mu_T$  defined in (56) is given by,

$$f \circ \mu_T : \mathbb{R}^6 \rightarrow \mathbb{R}, \quad f \circ \mu_T(x) = \|B\text{vec}(e^{\zeta(x)\top} T^\top Q T e^{\zeta(x)})\|^2. \quad (58)$$

*Quadratic Model of the Local Cost Function.* The second order Taylor approximation of  $f \circ \mu_T$  about  $0 \in \mathbb{R}^6$  in direction  $x$  consists of three terms,

- (i) a constant term,  $(f \circ \mu_T)(tx)|_{t=0} = \|B\text{vec}(A)\|^2$ ,
- (ii) a linear term,

$$2\text{vec}^\top(A\zeta(x) + \zeta^\top(x)A)B^\top B\text{vec}(A) = 2x^\top \nabla_{f \circ \mu_T}(0).$$

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  $f \circ \mu_T$  evaluated at  $0 \in \mathbb{R}^6$  is given as,

$$\nabla_{f \circ \mu_T}(0) = C^\top B\text{vec}(A), \quad C := B[(I \otimes A)(A \otimes I)] \begin{bmatrix} G \\ J \end{bmatrix}. \quad (59)$$

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

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

and the second term is

$$\text{vec}^\top(A)B^\top B\text{vec}\left(A\zeta^2(x) + 2\zeta^\top(x)A\zeta(x) + \zeta^{\top^2}(x)A\right) = x^\top \widetilde{H}_{f \circ \mu_T}(0)x.$$

Thus, the Hessian of  $f \circ \mu_T$  evaluated at  $0 \in \mathbb{R}^6$  is

$$\mathbf{H}_{f \circ \mu_T}(0) = \widehat{\mathbf{H}}_{f \circ \mu_T}(0) + \widetilde{\mathbf{H}}_{f \circ \mu_T}(0), \quad (60)$$

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

$$\begin{aligned} \widehat{\mathbf{H}}_{f \circ \mu_T}(0) &= C^\top C \geq 0, \\ \widetilde{\mathbf{H}}_{f \circ \mu_T}(0) &= [G^\top \ J^\top] \begin{bmatrix} (D^\top \otimes A) & (D^\top A \otimes I) \\ (AD \otimes I) & (A \otimes D^\top) \end{bmatrix} \begin{bmatrix} G \\ J \end{bmatrix}. \end{aligned} \quad (61)$$

### 6.3 Summary of Algorithm

Start with  $T = T_0 \in SE_3$  using the initialization procedure described in [21].

Step 1: Carry out the optimization step,

- Compute the gradient vector  $\nabla_{f \circ \mu_T}(0)$  and the Hessian matrix  $\mathbf{H}_{f \circ \mu_T}(0)$  via (59), (60) respectively,
- If  $\mathbf{H}_{f \circ \mu_T}(0) > 0$ , compute the Newton step,  $x_{\text{opt}}(T) = -[\mathbf{H}_{f \circ \mu_T}(0)]^{-1} \nabla_{f \circ \mu_T}(0)$ , otherwise compute the Gauss step  $x_{\text{opt}}(T) = -[\widehat{\mathbf{H}}_{f \circ \mu_T}(0)]^{-1} \nabla_{f \circ \mu_T}(0)$ ,
- Compute the optimum step size  $\lambda_{\text{opt}}$  in direction  $x_{\text{opt}}(T)$  using backtracking line search, as described in [30].

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

Step 3: Set  $T = \widehat{T}$ , go back to Step 1 if  $\|\nabla_{f \circ \mu_T}(0)\| > \epsilon$ .

## 7 Application 4: Global Registration of Multiple Laser Scans

Constructing a 3D computer model of a real object from 3D surface measurement data has various applications in computer graphics, virtual reality, computer vision and reverse engineering. To construct such a model, a single view of the object is often not sufficient due to self occlusion, presence of shadow and the limited field of view of the 3D scanner. Thus multiple partial views of the object from different viewpoints are needed to describe the entire object. Typically the views are obtained from multiple scanners or a single scanner being stationed at different locations and orientation, or a fixed scanner taking time-sampled images of an object on the moving turntable. The images are often simplified as a set of features, such as points and the relative position and orientation (pose) between views are unknown. Thus, these partially overlapping views need to be registered within a common reference frame to determine the unknown relative pose.

We restrict attention to the registration of multiview 3D point sets with known correspondences between overlapping images. Some work in this area can be found in [3, 34, 31, 38]. These techniques have been compared in [5].

Here, we address the global registration task as an unconstrained optimization problem on a constraint manifold. Our novel algorithm involves iterative cost function reduction on the smooth manifold formed by the  $N$ -fold product of special orthogonal groups. The optimization is based on locally quadratically convergent Newton-type iterations on this constraint manifold. In addition, we present a new closed form solution based on a singular value decomposition to achieve simultaneous registration of multiple point sets. In the noise free case, it gives correct registrations in a single step. In the presence of noise additional  $3 \times 3$  singular value decompositions for projection to the constraint manifold are required. This analytical solution is a useful initial estimate for any iterative algorithm.

*Outline.* In Section 7.1, the global multiview registration task is formulated as minimizing smooth function on the  $N$ -fold product of special orthogonal group. Details of the geometric optimization method are discussed in Section 7.2. Next, we put forward a new analytic noise-free solution and the noisy initialization steps in Section 7.3. Finally, a summary of the proposed algorithm is outlined in Section 7.4. Full details of the material in this section can be found in [17].

## 7.1 Problem Formulation

### 3D Object Points and Multiple Views

Consider a 3D object as a set of 3D points  $W := \{w^k \in \mathbb{R}^3 \mid k = 1, 2, \dots, n\}$  in a ‘world’ reference frame. Also consider multiple views of the object. Each view being from a different vantage point and viewing direction and each viewing being of possibly only a subset of the  $n$  3D points. For  $N$  views, let us denote the relative rotations and translations as  $(R_1, t_1), \dots, (R_N, t_N)$ , that is, relative to the ‘world’ reference frame, where  $R_i$  is a  $3 \times 3$  rotation matrix and  $t_i \in \mathbb{R}^3$  is a translation vector.

The  $i^{\text{th}}$  view is limited to  $n_i$  points  $W_i = \{w_i^k \in \mathbb{R}^3 \mid k = 1, 2, \dots, n_i\} \subset W$  and is denoted  $V_i = \{v_i^k \in \mathbb{R}^3 \mid k = 1, 2, \dots, n_i\}$  and consists of the images of the  $n_i$  points in  $W_i$  with relative rotation matrices and translation vectors given by  $(R_i, t_i)$ . Thus in the noise free case,

$$w_i^k = R_i v_i^k + t_i, \quad k = 1, 2, \dots, n_i.$$

Let  $W_{ij} = W_i \cap W_j$  be the set of  $n_{ij}$  points in  $W_i$  for which there are corresponding points in  $W_j$ , for  $i, j = 1, \dots, N$ . That is,  $W_{ij} = W_{ji}$  consists of  $n_{ij} = n_{ji}$  points  $w_{ij}^k = w_{ji}^k \in \mathbb{R}^3$ ,  $k = 1, \dots, n_{ij}$ . In view  $V_i$  the set of images of these points is denoted  $V_{ij} := \{v_{ij}^k \in \mathbb{R}^3 \mid k = 1, 2, \dots, n_{ij}\} \subset V_i$  and of course for view  $V_j$  it is denoted  $V_{ji} := \{v_{ji}^k \in \mathbb{R}^3 \mid k = 1, 2, \dots, n_{ij}\} \subset V_j$ . In the noise free case, it is immediate that

$$w_{ij}^k = R_i v_{ij}^k + t_i = R_j v_{ji}^k + t_j, \quad \forall i, j = 1, 2, \dots, N, \quad k = 1, 2, \dots, n_{ij}. \quad (62)$$

### Global Registration Error

When there is measurement noise, it makes sense to work with a cost function that penalizes the error  $(R_i v_{ij}^k + t_i) - (R_j v_{ji}^k + t_j)$  for all  $i, j = 1, 2, \dots, N$  and  $k = 1, 2, \dots, n_{ij}$ . Trivially the error is zero for  $i = j$ . The cost index for all the registrations which first comes to mind is given by the sum of the squared Euclidean distances between the corresponding points in all overlaps,

$$g = \sum_{i=1}^N \sum_{j=i+1}^N \sum_{k=1}^{n_{ij}} \|(R_i v_{ij}^k + t_i) - (R_j v_{ji}^k + t_j)\|^2, \quad (63)$$

Before optimizing this index, we first reformulate the index in a more convenient notation. Let

$$\mathcal{R} := [R_1 \ R_2 \ \dots \ R_N] \in \mathbb{R}^{3 \times 3N}, \quad \mathcal{T} := [t_1 \ t_2 \ \dots \ t_N] \in \mathbb{R}^{3 \times N}, \quad (64)$$

and  $e_i := i^{\text{th}}$  column of  $N \times N$  identity matrix,  $I_N$ , then we have

$$R_i = \mathcal{R}(e_i^\top \otimes I_3), \quad t_i = \mathcal{T}e_i, \quad t_i - t_j = \mathcal{T}e_{ij}, \quad e_{ij} := e_i - e_j.$$

Also define,

$$\begin{bmatrix} A & B \\ B^\top & C \end{bmatrix} = \sum_{i=1}^N \sum_{j=i+1}^N \sum_{k=1}^{n_{ij}} \begin{bmatrix} a_{ij}^k \\ e_{ij} \end{bmatrix} \begin{bmatrix} a_{ij}^{k\top} & e_{ij}^\top \end{bmatrix} \geq 0, \quad a_{ij}^k := (e_i \otimes I_3)v_{ij}^k - (e_j \otimes I_3)v_{ji}^k.$$

Now, simple manipulations show that the registration error cost function (63) can be rewritten in convenient matrix notation as

$$g(\mathcal{R}, \mathcal{T}) = \text{tr}(\mathcal{R}A\mathcal{R}^\top + 2\mathcal{R}B\mathcal{T}^\top + \mathcal{T}C\mathcal{T}^\top), \quad (65)$$

or equivalently, as

$$g(\mathcal{R}, \mathcal{T}) = \text{tr}(\mathcal{R}A\mathcal{R}^\top) + 2\text{vec}^\top(\mathcal{T})\text{vec}(\mathcal{R}B) + \text{vec}^\top(\mathcal{T})(C \otimes I_3)\text{vec}(\mathcal{T}). \quad (66)$$

### A More Compact Reformulation

*Optimal Translation.* Observe that (66) is readily minimized over all  $\text{vec}(\mathcal{T}) \in \mathbb{R}^{3N}$ , or  $\mathcal{T} \in \mathbb{R}^{3 \times N}$ . Thus,

$$\mathcal{T}^*(\mathcal{R}) := \arg \min_{\mathcal{T}} g(\mathcal{R}, \mathcal{T}), \quad (67)$$

satisfies

$$\text{vec}(\mathcal{T}^*(\mathcal{R})) = -(C^\dagger \otimes I_3)\text{vec}(\mathcal{R}B) = -\text{vec}(\mathcal{R}BC^\dagger) \Leftrightarrow \mathcal{T}^*(\mathcal{R}) = -\mathcal{R}BC^\dagger. \quad (68)$$

Note that the data matrix  $C$  is singular, since  $C1_N = 0$ , where  $1_N$  is the  $N \times 1$  vector consisting of unity elements. Thus the pseudo inverse  $C^\dagger$  is used.

Clearly, the absolute transformations can not be recovered. Since rotating each view by  $R_0$  and then translating it by  $t_0$  does not change the value of the cost function, as now indicated,

$$\| (R_0(R_i v_{ij}^k + t_i) + t_0) - (R_0(R_j v_{ji}^k + t_j) + t_0) \| = \| (R_i v_{ij}^k + t_i) - (R_j v_{ji}^k + t_j) \|,$$

it makes sense to estimate the rigid body transformations relative to (say) the first reference frame, by fixing  $(R_1, t_1)$  as  $(I_3, 0_3)$  where  $0_3$  denotes the  $3 \times 1$  zero vector.

*Global Registration Error Independent of Translation.* From the previous subsection, we see that optimizing the translation can be decoupled from optimizing the rotation. Substituting  $\mathcal{T}^*(\mathcal{R})$  from (68) into (65) and denoting  $\mathcal{M} := A - BC^\dagger B^\top$  leads to a registration error cost function depending only on rotations,

$$f(\mathcal{R}) := g(\mathcal{R}, \mathcal{T}(\mathcal{R})) = \text{tr}(\mathcal{R}\mathcal{M}\mathcal{R}^\top) = \text{vec}^\top(\mathcal{R}^\top)(I_3 \otimes \mathcal{M})\text{vec}(\mathcal{R}^\top). \quad (69)$$

## 7.2 Optimization on $SO_3^N$

### Geometry of $SO_3^N$

Recall the geometry of the special orthogonal group in Section 4.2. Our interest here is the  $N$ -fold product manifold of  $SO_3$  which is a smooth manifold of dimension  $3N$ , given by

$$SO_3^N = \overbrace{SO_3 \times \cdots \times SO_3}^{N \text{ times}}.$$

*Tangent Space of  $SO_3^N$ .* Recall that the tangent space of  $SO_3$  at  $R_i$  for  $i = 1, 2, \dots, N$  is given as  $\text{T}_{R_i}SO_3 = \{R_i\Omega_i \mid \Omega_i \in \mathfrak{so}_3\}$  and the corresponding affine tangent space is  $\text{T}_{R_i}^{\text{aff}}SO_3 = \{R_i + R_i\Omega_i \mid \Omega_i \in \mathfrak{so}_3\}$ . Note that direct sum  $\oplus$  of matrices is equal to a block diagonal matrix with the individual matrices as the diagonal blocks and define

$$\tilde{\Omega} := \Omega_1 \oplus \Omega_2 \oplus \cdots \oplus \Omega_N, \quad \Omega_i \in \mathfrak{so}_3. \quad (70)$$

Due to isomorphism, the tangent space of  $SO_3^N$  at  $\mathcal{R} = [R_1 \ R_2 \ \cdots \ R_N] \in SO_3^N$  can be identified as,

$$\text{T}_{\mathcal{R}}SO_3^N = \mathcal{R}\tilde{\Omega}, \quad (71)$$

and the affine tangent space is

$$\text{T}_{\mathcal{R}}^{\text{aff}}SO_3^N = \mathcal{R} + \mathcal{R}\tilde{\Omega}. \quad (72)$$

*Local Parameterization of  $SO_3^N$ .* For every point  $R_i \in SO_3$ , there exists a smooth exponential map

$$\mu_{R_i} : \mathbb{R}^{3N} \rightarrow SO_3, \quad \omega_i \mapsto R_i e^{\Omega(\omega_i)},$$

which is a diffeomorphism about the origin in  $\mathbb{R}^3$ . Due to isomorphism, every point  $\mathcal{R} \in SO_3^N$  can be locally parameterized by the smooth map

$$\begin{aligned} \varphi_{\mathcal{R}} : \mathbb{R}^{3N} &\rightarrow SO_3^N, \\ \omega = [\omega_1^\top \cdots \omega_N^\top]^\top &\mapsto \mathcal{R} \left( e^{\Omega(\omega_1)} \oplus e^{\Omega(\omega_2)} \oplus \cdots \oplus e^{\Omega(\omega_N)} \right) = \mathcal{R} e^{\tilde{\Omega}(\omega)}. \end{aligned} \quad (73)$$

### Cost Function

*Cost Function on  $SO_3^N$ .* Recall  $\mathcal{M}$  from (69), consider the smooth function,

$$f : SO_3^N \rightarrow \mathbb{R}, \quad f(\mathcal{R}) = \text{tr}(\mathcal{R}\mathcal{M}\mathcal{R}^\top) = \text{vec}^\top(\mathcal{R}^\top)(I_3 \otimes \mathcal{M})\text{vec}(\mathcal{R}^\top). \quad (74)$$

Minimization of this function penalizes the alignment error among all range images simultaneously.

*Local Cost Function.* The cost function  $f$  at  $\mathcal{R} \in SO_3^N$  expressed in local parameter space using the local parameterization  $\varphi_{\mathcal{R}}$  defined in (73) is,

$$f \circ \varphi_{\mathcal{R}} : \mathbb{R}^{3N} \rightarrow \mathbb{R}, \quad f \circ \varphi_{\mathcal{R}}(\omega) = \text{tr}(\mathcal{R} e^{\tilde{\Omega}(\omega)} \mathcal{M} e^{\tilde{\Omega}(\omega)^\top} \mathcal{R}^\top). \quad (75)$$

*Quadratic Model of Local Cost Function.* The second order Taylor approximation of  $f \circ \varphi_{\mathcal{R}}$  about  $0 \in \mathbb{R}^{3N}$  in direction  $\omega$  consists of three terms,

- (i) a constant term,  $(f \circ \varphi_{\mathcal{R}})(t\omega)|_{t=0} = \text{tr}(\mathcal{R}\mathcal{M}\mathcal{R}^\top)$ .
- (ii) a linear term,

$$2 \text{tr}(\mathcal{R}\tilde{\Omega}\mathcal{M}\mathcal{R}^\top) = 2\omega^\top \nabla_{f \circ \varphi_{\mathcal{R}}}(0),$$

Recall (70), let  $\text{vec}(\tilde{\Omega}^\top) := \tilde{Q}\omega$ ,

$$J := (\mathcal{R} \otimes I_{3N})\tilde{Q}, \quad \tilde{Q} := Q_{e_1} \oplus Q_{e_2} \oplus \cdots \oplus Q_{e_N}, \quad Q_{e_i} := \begin{bmatrix} e_i \otimes Q_x \\ e_i \otimes Q_y \\ e_i \otimes Q_z \end{bmatrix}, \quad (76)$$

the explicit formula for the gradient of  $f \circ \varphi_{\mathcal{R}}$  evaluated at  $0 \in \mathbb{R}^{3N}$  is

$$\nabla_{f \circ \varphi_{\mathcal{R}}}(0) = J^\top \text{vec}(\mathcal{M}\mathcal{R}^\top), \quad (77)$$

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

$$\text{tr}(\mathcal{R}\tilde{\Omega}\mathcal{M}\tilde{\Omega}^\top\mathcal{R}^\top) = \omega^\top \hat{\text{H}}_{f \circ \varphi_{\mathcal{R}}}(0)\omega,$$

and the second quadratic term is

$$\text{tr}(\mathcal{R}\tilde{\Omega}^2\mathcal{M}\mathcal{R}^\top) = \text{vec}^\top(\tilde{\Omega}^\top)\text{vec}(\mathcal{M}\mathcal{R}^\top\mathcal{R}\tilde{\Omega}) = \omega^\top \tilde{\text{H}}_{f \circ \varphi_{\mathcal{R}}}(0)\omega.$$

Thus the Hessian of  $f \circ \varphi_{\mathcal{R}}$  evaluated at zero is

$$\text{H}_{f \circ \varphi_{\mathcal{R}}}(0) = \hat{\text{H}}_{f \circ \varphi_{\mathcal{R}}}(0) + \tilde{\text{H}}_{f \circ \varphi_{\mathcal{R}}}(0), \quad (78)$$

where

$$\hat{\text{H}}_{f \circ \varphi_{\mathcal{R}}}(0) = J^\top (I_3 \otimes \mathcal{M})J \geq 0, \quad \tilde{\text{H}}_{f \circ \varphi_{\mathcal{R}}}(0) = -\tilde{Q}^\top (I_{3N} \otimes \mathcal{M}\mathcal{R}^\top\mathcal{R})\tilde{Q}.$$

### 7.3 Algorithm Initialization

In the noise free case, for  $\mathcal{R} \in SO_3^N$ , the optimal value of the cost function (69) is zero, as

$$\text{vec}^\top(\mathcal{R}^\top)\text{vec}(\mathcal{M}\mathcal{R}^\top) = 0 \Rightarrow \text{vec}(\mathcal{M}\mathcal{R}^\top) = 0 \Rightarrow \mathcal{M}\mathcal{R}^\top = 0. \quad (79)$$

Since  $\mathcal{M}$  is symmetric, a singular value decomposition gives

$$\mathcal{M} = U\Sigma U^\top = [U_a \ U_b] \begin{bmatrix} \Sigma_a & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} U_a^\top \\ U_b^\top \end{bmatrix} \Rightarrow \mathcal{M}U_b = 0. \quad (80)$$

To obtain  $\mathcal{R}$  such that  $R_1 = I_3$ , let  $\hat{U} := [I_3 \ 0] U_b$ , then we have

$$\mathcal{R} = \hat{U}^{-\top} U_b^\top. \quad (81)$$

In the presence of noise, the optimal cost function is no longer equal to zero. In this case,  $U_b$  is chosen to be the set of right singular vectors associated with 3 least singular values of  $\mathcal{M}$ , which may not be zero. These singular vectors might not be on  $SO_3^N$ . Thus, an additional projection step is required. Denoting  $G_i := \hat{U}^{-\top} U_b(e_i \otimes I_3)$ , we have

$$R_i^{\text{opt}} = \arg \min_{R_i \in SO_3} \|R_i - G_i\| = \arg \max_{R_i \in SO_3} \text{tr}(R_i^\top G_i). \quad (82)$$

By applying a singular value decomposition on  $G_i$ , we obtain

$$G_i = WAZ^\top, \quad R_i^{\text{opt}} = W \begin{bmatrix} I_2 & 0 \\ 0 & \det(WZ^\top) \end{bmatrix} Z^\top. \quad (83)$$

### 7.4 Summary of Algorithm

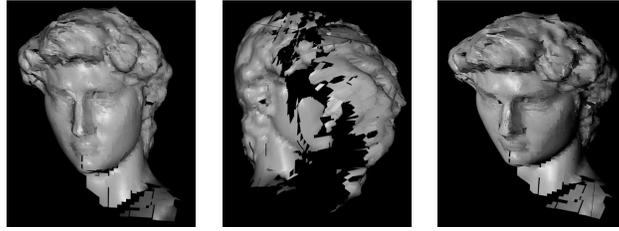
Start with  $\mathcal{R} = \mathcal{R}_0$  using initialization algorithm in Section 7.3.

Step 1: Carry out the optimization step,

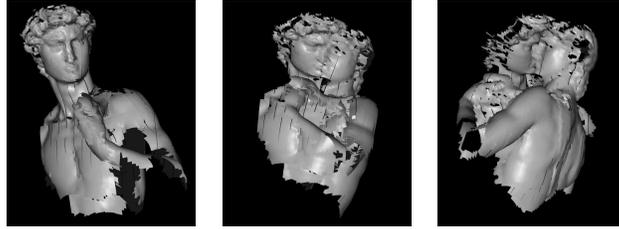
- Compute the gradient  $\nabla_{f \circ \varphi_{\mathcal{R}}}(0)$  and the Hessian  $\mathbf{H}_{f \circ \varphi_{\mathcal{R}}}(0)$  via (77), (78) respectively.
- If  $\mathbf{H}_{f \circ \varphi_{\mathcal{R}}}(0) > 0$ ,  
 Compute the Newton step,  $\omega_{\text{opt}} = -[\mathbf{H}_{f \circ \varphi_{\mathcal{R}}}(0)]^{-1} \nabla_{f \circ \varphi_{\mathcal{R}}}(0)$ ,  
 otherwise compute the Gauss step  $\omega_{\text{opt}} = -[\hat{\mathbf{H}}_{f \circ \varphi_{\mathcal{R}}}(0)]^{-1} \nabla_{f \circ \varphi_{\mathcal{R}}}(0)$ .
- Compute the optimum step size  $\lambda_{\text{opt}}$  in direction  $\omega_{\text{opt}}$  using a back-tracking line search, as described in [30].

Step 2: Carry out the projection step,  $\hat{\mathcal{R}} = \mathcal{R} \left( e^{\Omega(\lambda_{\text{opt}} \omega_1^{\text{opt}})} \oplus \dots \oplus e^{\Omega(\lambda_{\text{opt}} \omega_N^{\text{opt}})} \right)$ .

Step 3: Set  $\mathcal{R} = \hat{\mathcal{R}}$ , go back to Step 1 if  $\|\nabla_{f \circ \varphi_{\mathcal{R}}}(0)\| > \epsilon$ .



(a) The head of David (detailed: 38 scans)



(b) The head and bust of David (10 scans)

**Fig. 7.** This figure shows the results of three algorithms for simultaneous registration of multiple 3D point sets - our approach, Williams and Bennamoun’s SVD-based method [38], and Benjemaa and Schmitt’s Quaternion-based method [3] (from left to right) on different instances of the David model. We acknowledge Prof. Marc Levoy and the Digital Michelangelo Project at Stanford University for providing access to the raw scan data.

## 8 Conclusions

This paper has been motivated by the fundamental role of pose estimation in many computer vision tasks and the benefit of exploiting the geometric structure of optimization problems. We have focused on the development of proposed parameterization-based framework and its implementation in recovering pose information from images.

As digital signal processing hardware becomes more powerful, geometric optimization algorithms are potentially attractive for fundamental on-line pose recovery tasks, and potentially can be a basic tool for more sophisticated tasks involving adaptive object recognition and pose tracking. The particular approach of seeking Newton-type algorithms on local parametrization of the constraint manifold and projections to the manifold appears attractive for such applications.

## References

1. P. A. Absil, “Trust Region Methods on Riemannian Manifolds with Applications in Numerical Linear Algebra”, *Proceedings of the 16th International Symposium*

- on *Mathematical Theory of Networks and Systems*, Leuven, Belgium, 2004.
2. R. L. Adler, J. -P. Dedieu, “Newton’s Method on Riemannian Manifolds and a Geometric Model for the Human Spine”, *IMA Journal of Numerical Analysis*, vol. 22, pp. 359–390, 2002.
  3. R. Benjema, and F. Schmitt, “A Solution for the Registration of Multiple 3D Point Sets Using Unit Quaternion”, *Proceedings of the European Conference on Computer Vision*, Freiburg, Germany, pp. 34–50, 1998.
  4. R. W. Brockett, “Differential Geometry and the Design of Gradient Algorithms”, *Proceedings of the Symposium on Pure Mathematics*, Ed. R. Green and S. T. Yau, pp. 69–92, Providence, RI, 1993.
  5. S. Cunningham, and A. J. Stoddart, “N-View Point Set Registration: A Comparison”, *Proceedings of the British Machine Vision Conference*, Nottingham, U. K., 1999.
  6. W. Davidon, “Variable Metric Method for Minimization”, *Technical Report ANL-5990*, Argonne National Laboratory, 1959. See also *SIAM Journal of Optimization*, vol. 1, pp. 1–17, 1991.
  7. A. Edelman, T. A. Arias, and S. T. Smith, “The Geometry of Algorithms with Orthogonality Constraints”, *SIAM Journal on Matrix Analysis and Applications*, vol. 20, no. 2, pp. 303–353, 1998.
  8. R. Fletcher, and C. M. Reeves, “Functions Minimization by Conjugate Gradients”, *Computer Journal*, vol. 7, pp. 149–154, 1964.
  9. D. Gabay, “Minimizing a Differentiable Function over a Differentiable Manifold”, *Journal of Optimization Theory and Applications*, vol. 37, no. 2, pp. 177–219, 1982.
  10. R. Hartley, and A. Zisserman, *Multiple View Geometry*, Cambridge University Press, 2000.
  11. U. Helmke, and J. B. Moore, *Optimization and Dynamical Systems*, CCES, Springer, London, 1994.
  12. U. Helmke, K. Hüper, and J. B. Moore, “Quadratically Convergent Algorithms for Optimal Dextrous Hand Grasping”, *IEEE Transactions on Robotics and Automation*, vol. 18, no. 2, 2002.
  13. U. Helmke, K. Hüper, P. Y. Lee, and J. B. Moore, “Essential Matrix Estimation via Newton-type Methods”, *Proceedings of the 16th International Symposium on Mathematical Theory of Networks and Systems*, Belgium, Leuven, 2004.
  14. M. R. Hestenes, and E. Stiefel, “Methods of Conjugate Gradients for Solving Linear Systems”, *Journal of Research of the National Bureau of Standards*, vol. 49, pp. 409–436, 1952.
  15. K. Hüper, and J. Trumpf, “Newton-like Methods for Numerical Optimization on Manifolds”, *Proceedings of the 38th Asilomar Conference on Signals, Systems and Computers*, California, United States, 2004.
  16. D. C. Jiang, J. B. Moore, and H. B. Ji, “Self-Concordant Functions for Optimization on Smooth Manifolds”, *Proceedings of the 43rd IEEE Conference on Decision and Control*, Bahamas, pp. 3631–3636, 2004.
  17. S. Krishnan, P. Y. Lee, J. B. Moore, and S. Venkatasubramania, “Global Registration of Multiple 3D Point Sets via Optimization-on-a-Manifold”, *Proceedings of Eurographics Symposium on Geometry Processing*, pp. 187–196, 2005.
  18. P. Y. Lee, *Geometric Optimization for Computer Vision*, PhD Thesis, Australian National University, 2005.
  19. J. M. Lee, *Introduction to Smooth Manifolds*, Graduate Texts in Mathematics, Springer, New York, 2003.

20. P. Y. Lee, and J. B. Moore, “Gauss-Newton-on-Manifold for Pose Estimation”, *Journal of Industrial and Management Optimization*, vol. 1, no. 4, 2005.
21. P. Y. Lee, and J. B. Moore, “3D Localization of Quadratic Surfaces”, *Proceedings of the 38th Asilomar Conference on Signals, Systems and Computers*, California, United States, 2004.
22. C. -P. Lu, G. D. Hager, and E. Mjolsness, “Fast and Globally Convergent Pose Estimation from Video Images”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 6, pp. 610–622, 2000.
23. D. G. Luenberger, “The Gradient Projection Method along Geodesics”, *Management Science*, vol. 18, pp. 620–631, 1972.
24. Y. Ma, J. Kořecká, and S. Sastry, “Optimization Criteria and Geometric Algorithms for Motion and Structure Estimation”, *International Journal of Computer Vision*, vol. 44, no. 3, pp. 219–249, 2001.
25. R. Mahony, *Optimization Algorithms on Homogeneous Spaces*, PhD thesis, Australian National University, Canberra, March 1994.
26. R. Mahony, and J. H. Manton, “The Geometry of the Newton Method on Non-Compact Lie Groups”, *Journal of Global Optimization*, 309–327, vol. 23, 2002.
27. J. H. Manton, “Optimization Algorithms Exploiting Unitary Constraints”, *IEEE Transactions on Signal Processing*, vol. 50, no. 3, pp. 635–650, March 2002.
28. J. H. Manton, “On the Various Generalisations of Optimisation Algorithms to Manifolds”, *Proceedings of the 16th International Symposium on Mathematical Theory of Networks and Systems*, Leuven, Belgium, 2004.
29. Y. Nesterov, *Introductory Lectures on Convex Optimization - A Basic Course*, Kluwer Academic Publishers, 2004.
30. J. Nocedal, and S. J. Wright, *Numerical Optimization*, Springer Series in Operation Research, Springer Verlag, New York, 1999.
31. X. Pennec, “Multiple Registration and Mean Rigid Shapes: Application to the 3D Case”, *Proceedings of 16th Leeds Annual Statistical Workshop*, Leeds, U. K., pp 178–185, 1996.
32. M. Shub, “Some Remarks on Dynamical Systems and Numerical Analysis”, *Dynamical Systems and Partial Differential Equations, Proceedings of VII ELAM*, Caracas: Equinoccio, Universidad Simon Bolivar, pp. 69–92, 1986.
33. S. T. Smith, *Geometric Optimization Methods for Adaptive Filtering*, PhD Thesis, Harvard University, Cambridge Massachusetts, 1993.
34. A. Stoddart, and A. Hilton, “Registration of Multiple Point Sets”, *Proceedings of the International Conference on Pattern Recognition*, Vienna, pp. 40–44, 1996.
35. Special Session on Applications of Differential Geometry to Signal Processing, *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Philadelphia, United States, 2005
36. C. J. Taylor, and D. J. Kriegman, “Minimization on the Lie Group  $SO(3)$  and Related Manifolds”, *Yale University Technical Report no. 9405*, April 1994.
37. C. Udriste, *Convex Functions and Optimization Methods on Riemannian Manifolds*, Kluwer Academic Publishers, 1994.
38. J. Williams, and M. Bennamoun, “Simultaneous Registration of Multiple Corresponding Point Sets”, *Computer Vision and Image Understanding*, vol. 81, no. 1, pp. 117–142, 2001.
39. Y. Yang, “Optimization on Riemannian Manifold”, *Proceedings of the 38th Conference on Decision and Control*, pp. 888–893, 1999.