JOURNAL OF INDUSTRIAL AND
MANAGEMENT OPTIMIZATION
Volume **1**, Number **4**, November **2005**

Website: http://AIMsciences.org

pp. **565–587**

GAUSS-NEWTON-ON-MANIFOLD FOR POSE ESTIMATION

Pei Yean Lee † and John B Moore †‡

National ICT Australia Ltd., Australia
Australian National University, Australia

(Communicated by Kok Lay Teo)

Abstract. We consider the task of estimating the relative pose (position and orientation) between a 3D object and its projection on a 2D image plane from a set of point correspondences. Our approach is to formulate the task as an unconstrained optimization problem on the intersection of the special orthogonal group and a cone, and exploit as much as possible the geometry of the underlying parameter space. The optimization does not require Riemannian geometry. It involves successive parameterization of the constraint manifold and is based on Newton-type iterations in local parameter space. A direct proof of local quadratical convergence to the optimum is provided. A key feature of the proposed approach, not used in earlier studies, is an analytic geodesic search, alternating between gradient, Gauss, Newton and random directions, which ensures the escape from local minima and convergence to a global minimum without the need to reinitialize the algorithm. Indeed, 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.

1. **Introduction.** In machine vision, robotics and computer graphics, a fundamental problem is to estimate the relative pose (position and orientation) between a 3D object and its 2D image from a set of feature correspondences. The pose estimation can be viewed as either determining the camera pose with respect to a known object or as estimating the object pose with respect to a known camera.

There are two approaches for solving this problem presented in the literature. The first approach adopts linear methods that yield closed-form solutions [2, 15, 1] which are simple enough to implement in online computation. They are very sensitive to noise because the orthonormality constraint of the rotation matrix is not taken into account. The second approach is to formulate the task as a constrained nonlinear optimization problem [3, 9, 10, 11] and solve using iterative techniques. These provide numerical solutions that are more accurate and robust to measurement noise. Also, when Newton-based recursions are used, fast (quadratic) local convergence rate can be achieved. However, the methods rely on a good initial pose

2000 *Mathematics Subject Classification.* 90C90, 49M37, 68T45, 68U05.

Key words and phrases. geometric optimization, manifold, differential geometry, local parameterization, Newton-like methods, machine vision, pose estimation.

† National ICT Australia which is funded by the Australian Department of Communications, Information Technology and the Arts and the Australian Research Council through Backing Australia's Ability and the ICT Centre of Excellence Program. Address: Locked Bag 8001, Canberra ACT 2601, Australia.

‡ This work was partially supported by ARC Grants A00105829, DP0450539.

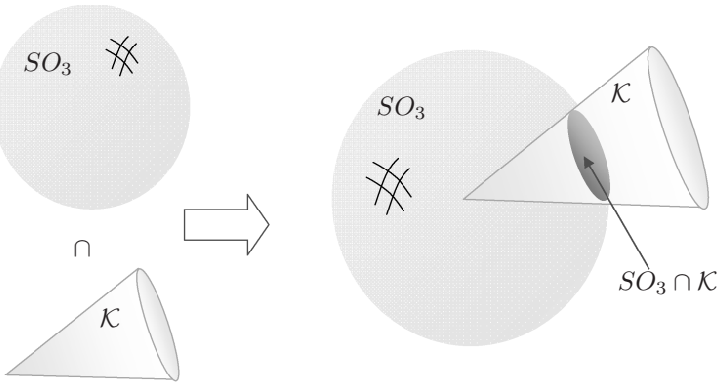


FIGURE 1. The special orthogonal group SO_3, the cone constraint $\mathcal{K}$, and the resulting constraint manifold $SO_3 \cap \mathcal{K}$, which is the intersection between SO_3 and $\mathcal{K}$.

estimate. Also, without random reinitializations and selection of the lowest cost function outcome, such methods might converge to a local minimum which is not the global minimum, or converge to an infeasible solution, for which the object is estimated to be behind the camera. Among the existing recursive algorithms, only the technique proposed by Lu, Hager and Mjolsness [11] exploits the geometry of the underlying constraint manifold.

This paper, building on our earlier conference presentations [8, 5], addresses the 2D-3D pose recovery task as minimizing a smooth function over the intersection of a rotation group and a cone, as depicted in Fig. 1. A new geometrical optimization framework based on successive parameterizations of the constraint manifold is proposed for the task, see also [12] for related optimization techniques. In contrast to the 'fast' linearly convergent geometric algorithm of [11], the Newton-like recursions devised using the proposed geometrical framework are locally quadratically convergent, as demonstrated by simulation results and rigorous mathematical proof.

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 any local minimum, not the global minimum, and avoids the infeasible domain. That is, it converges to a global minimum, at least eventually, without the need for multiple reinitializations of the algorithm to seek a domain of attraction of the global minimum.

The *Newton decrement* or its estimate is used as an empirical indicator for selecting between gradient, Gauss, or Newton directions for a geodesic search, and for algorithm termination. We also put forward a new method for algorithm initialization, which leads to an exact solution in the noise free case, and a good initial estimate in the presence of noise. Simulation results suggests the proposed algorithms achieve significantly lower parameter estimation errors than techniques presented by [11] and convergence to a global minimum occurs in typically 5–10 iterations.

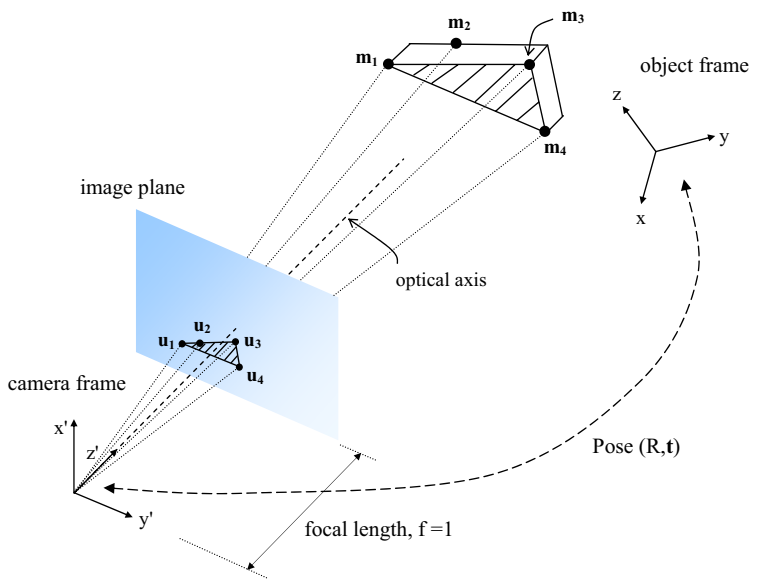


FIGURE 2. The 2D-3D pose estimation problem: given the model $\{m_1, \cdots, m_4\}$ expressed in object frame, and its corresponding image $\{u_1, \cdots, u_4\}$ expressed in camera frame, find the relative pose (R, t) between the object frame and the camera frame.

2. Problem Formulation.

2.1. Definitions.

Model: 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

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

Transformed model: 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 $R \in SO_3$, i.e. $R^\top R = I$, $\det(R) = 1$, representing the rotation, and $t \in \mathbb{R}^3$ is the translation vector.

Image: Each transformed model point m_i' is observed by a camera. The origin of the camera frame coincides with the center of projection, and the optical axis is along the positive z'-axis, as illustrated in Fig. 2. 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, as

$$u_i = F^{-1} p_i.$$

The camera calibration matrix F is an upper triangular matrix consisting of intrinsic camera parameters, details can be found in [4]. Under full perspective projection, each point in the model m_i is related to its corresponding

normalized image point by the following equation,

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

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

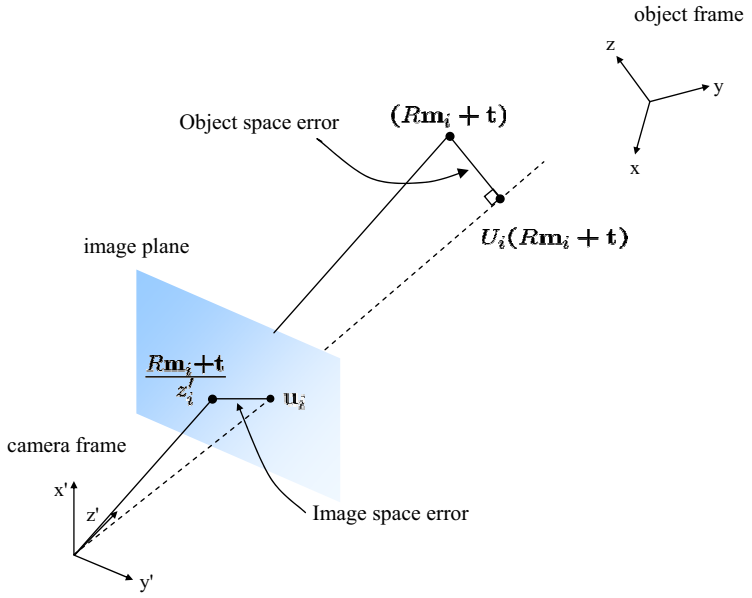


FIGURE 3. The image space error and the object space error.

2.2. **Cost function.** 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 [11], as depicted in Fig. 3. Let U_i be the projection operator,

$$U_i = U_i^\top := \frac{u_i u_i^\top}{u_i^\top u_i}, \quad U_i^2 = U_i. \tag{2}$$

For object space collinearity, the orthogonal projection of $m_i' = Rm_i + t$ onto the line-of-sight of the corresponding image point u_i should be equal to m_i' itself, as described by the following equation,

$$Rm_i + t = U_i(Rm_i + t), \quad i = 1, \cdots, n. \tag{3}$$

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

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

Observe that the cost function is quadratic in terms of errors, which are linear in the unknown pose parameters R, t. It is zero if and only if (3) is satisfied. 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.

We have also used more conventional image space cost functions, but using the object space cost function (4) gives better estimate of R, denoted $\hat{R}$, why? Perhaps the object space index penalizes more directly the term $\|R - \hat{R}\|^2$ or equivalently maximizes the term $\mathrm{tr}(R\hat{R}^\top)$.

2.3. **Optimal translation.** We first eliminate the translation vector t from the cost function (4) via least squares optimization to reduce the number of parameters for optimization. Note that there is no noise-free assumption for this. It might appear that we formulate the problem as two-stage optimization, first optimize R then optimize t. However, by eliminating t via least squares, this two stage optimization actually gives optimal solutions equivalent to optimizing over R, t in one stage. Exploiting the fact that $(I - U_i)^\top (I - U_i) = (I - U_i)$, cost function (4) can be re-expressed as,

$$\phi = \frac{1}{2}\sum_{i=1}^{n} m_i^\top R^\top (I-U_i) R m_i + t^\top \sum_{i=1}^{n} (I-U_i)(m_i^\top \otimes I)\mathrm{vec}(R) + \frac{1}{2} t^\top \sum_{i=1}^{n} (I-U_i) t. \quad (5)$$

Denoting

$$\widetilde{U} := \sum_{i=1}^{n} (I - U_i), \;\; \mathcal{U} = \widetilde{U}^{-1} \sum_{i=1}^{n} (I - U_i)(m_i^\top \otimes I), \quad (6)$$

with R fixed, an optimal t that minimizes (5) is given by

$$t = -\mathcal{U}\mathrm{vec}(R), \quad (7)$$

Substituting (7) into (1), the depth parameters z_i' can be reformulated as,

$$z_i' = \mathcal{B}_i \,\mathrm{vec}(R), \;\; \mathcal{B}_i := e_3^\top ((m_i^\top \otimes I) - \mathcal{U}). \quad (8)$$

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

$$\mathcal{K} := \{R \mid \{\mathcal{B}_i \mathrm{vec}(R) > 0\}_{i=1,2,\cdots,n}\}. \quad (9)$$

2.4. **Cost function independent of translation.** Substituting (7) into (4), the cost function can be reformulated as,

$$f : SO_3 \to \mathbb{R}, \quad f(R) = \frac{1}{2}\|\mathcal{D}\mathrm{vec}(R)\|^2, \quad (10)$$

where

$$\mathcal{D} = \begin{bmatrix} D_1^\top & D_2^\top & \cdots & D_n^\top \end{bmatrix}^\top, \quad D_i = (I - U_i)\left((m_i^\top \otimes I) - \mathcal{U}\right). \quad (11)$$

Remark 2.1. *The function*

$$g : \mathbb{R}^{3\times 3} \to \mathbb{R}, \quad g(X) = \frac{1}{2}\|\mathcal{D}\mathrm{vec}(X)\|^2, \quad (12)$$

is quadratic in the Euclidean space $\mathbb{R}^{3\times 3}$. However, the cost function $f = g|_{SO_3}$, which is the restriction of g to manifold SO_3 is no longer a quadratic function. Also,

we do not linearize this cost by using earlier estimates of R and t in the optimization process as in [11].

2.5. **Shifted centroid.** It turns out to be useful to refer all measurements to the centroids. Thus, denoting $\bar{m} := \frac{1}{n}\sum_{i=1}^{n} m_i$, (3) can be rewritten as follows,

$$(I - U_i)(R(m_i - \bar{m}) + (t + R\bar{m})) = 0,\ i = 1, \cdots, n.$$

The shifted model $\{m_i - \bar{m}\}$ has created a new object frame at its centroid and a new translation vector $(t + R\bar{m})$. Once the translation vector $(t + R\bar{m})$ is estimated from the algorithm, it needs to be transformed back into the original object frame. Without loss of generality, we assume subsequently that $\bar{m} = 0$, and the desired translation parameter is t.

3. **Geometry of the Special Orthogonal Group SO_3.** Rotational motion in $\mathbb{R}^3$ can be represented by the special orthogonal group SO_3, which consists of 3×3 orthogonal matrices with determinant $+1$. It is a Lie group and its associated Lie algebra $\mathfrak{so}_3$ is the set of 3×3 skew symmetric matrices of the form,

$$\Omega = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}. \tag{13}$$

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 mapping in (13), which maps a vector $\omega = \begin{bmatrix}\omega_x & \omega_y & \omega_z\end{bmatrix}^\top \in \mathbb{R}^3$ to a matrix $\Omega \in \mathfrak{so}_3$. Notice that Ω can be written as,

$$\Omega = Q_x\omega_x + Q_y\omega_y + Q_z\omega_z, \tag{14}$$

where

$$Q_x := \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix},\ Q_y := \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix},\ Q_z := \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \tag{15}$$

3.1. **Tangent space of SO_3.** Consider the tangent space of SO_3 at R,

$$\mathrm{T}_R\, SO_3 = \{R\Omega \mid \Omega \in \mathfrak{so}_3\}. \tag{16}$$

Shifting the origin of this space by R results in the affine tangent space of SO_3 at the point R given as,

$$\mathrm{T}_R^{\mathrm{aff}}\, SO_3 := \{R + R\Omega \mid \Omega \in \mathfrak{so}_3\}. \tag{17}$$

This is illustrated in Fig. 5.

3.2. **Local parameterization of SO_3.** Recall that a manifold is as a collection of local coordinate charts. Computations on a manifold are often conveniently carried out in these local parameter spaces. Let R be a point in SO_3, then there exist a smooth exponential map

$$\mu_R : \mathbb{R}^3 \to SO_3,\ \ \omega \mapsto Re^{\Omega(\omega)}, \tag{18}$$

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

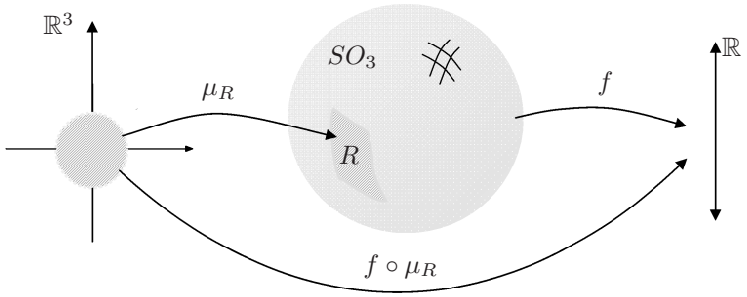


FIGURE 4. The mapping μ_R is a local parameterization of SO_3 around the point R such that $R = \mu_R(0)$, f is the smooth function defined on SO_3 and $f \circ \mu_R$ is the function f expressed in local parameter space $\mathbb{R}^3$.

4. Optimization on the Manifold SO_3.

4.1. Cost function on the manifold SO_3.

Recall the smooth function from (10),

$$f : SO_3 \to \mathbb{R}, \quad f(R) = \frac{1}{2}\|\mathcal{D}\mathrm{vec}(R)\|^2.$$

In the noise free case, the value of this function is zero if and only if there is a rotation matrix which aligns all object points with the line-of-sight of the corresponding image points exactly. In the presence of noise, the value of the cost function is no longer zero. Thus, we seek the minima of this cost function.

4.2. Local Cost Function.

Consider the mappings as in Fig. 4. The cost function f at $R \in SO_3$ expressed in local parameter space using the smooth local parameterization μ_R is given by,

$$f \circ \mu_R : \mathbb{R}^3 \to \mathbb{R}, \quad f \circ \mu_R(\omega) = \frac{1}{2}\|\mathcal{D}\mathrm{vec}(Re^{\Omega(\omega)})\|^2. \tag{19}$$

The second order Taylor approximation of $f \circ \mu_R$ about $0 \in \mathbb{R}^3$ in direction ω is

$$j_0^{(2)}(f \circ \mu_R) : \mathbb{R}^3 \to \mathbb{R},$$
$$\omega \mapsto \left((f \circ \mu_R)(t\omega) + \frac{\mathrm{d}}{\mathrm{d}\,t}(f \circ \mu_R)(t\omega) + \frac{1}{2}\frac{\mathrm{d}^2}{\mathrm{d}\,t^2}(f \circ \mu_R)(t\omega) \right)\Big|_{t=0}.$$

This expansion contains three terms:

(i) A constant

$$(f \circ \mu_R)(t\omega)|_{t=0} = \frac{1}{2}\|\mathcal{D}\mathrm{vec}(R)\|^2,$$

(ii) A term linear in ω

$$\frac{\mathrm{d}}{\mathrm{d}\,t}(f \circ \mu_R)(t\omega)\Big|_{t=0} = \mathrm{vec}^\top(R\Omega(\omega))\mathcal{D}^\top\mathcal{D}\mathrm{vec}(R) = \omega^\top \nabla_{f\circ\mu_R}(0),$$

recall Ω from (13), and let $\text{vec}(\Omega) := Q\omega$, then we have the Euclidean gradient of the $f \circ \mu_R$ evaluated at zero,

$$\nabla_{f\circ\mu_R}(0) = Q^\top (I \otimes R^\top)\mathcal{D}^\top \mathcal{D}\text{vec}(R), \tag{20}$$

where $Q := [\text{vec}(Q_x)\ \text{vec}(Q_y)\ \text{vec}(Q_z)]$.

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

$$\text{vec}^\top(R\Omega(\omega))\mathcal{D}^\top\mathcal{D}\text{vec}(R\Omega(\omega)) = \omega^\top \widehat{\mathsf{H}}_{f\circ\mu_R}(0)\omega. \tag{21}$$

Let $\text{vec}(C) := \mathcal{D}^\top\mathcal{D}\text{vec}(R)$, the second term is,

$$\begin{aligned}\text{vec}^\top(R)\mathcal{D}^\top\mathcal{D}\text{vec}(R\Omega^2(\omega)) &= \text{vec}^\top(R)\mathcal{D}^\top\mathcal{D}(\Omega^\top(\omega)\otimes R)\text{vec}(\Omega(\omega)),\\ &= \text{vec}^\top(R^\top C\Omega^\top(\omega))\text{vec}(\Omega(\omega)),\\ &= \omega^\top \widetilde{\mathsf{H}}_{f\circ\mu_R}(0)\omega.\end{aligned}$$

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

$$\mathsf{H}_{f\circ\mu_R}(0) = \widehat{\mathsf{H}}_{f\circ\mu_R}(0) + \widetilde{\mathsf{H}}_{f\circ\mu_R}(0), \tag{22}$$

where

$$\begin{aligned}\widehat{\mathsf{H}}_{f\circ\mu_R}(0) &= Q^\top(I\otimes R^\top)\mathcal{D}^\top\mathcal{D}(I\otimes R)Q \ \ \geqslant 0,\\ \widetilde{\mathsf{H}}_{f\circ\mu_R}(0) &= -Q^\top(I\otimes R^\top C),\end{aligned} \tag{23}$$

and since $\widetilde{\mathsf{H}}_{f\circ\mu_R}(0)$ is always symmetric, we have

$$\widetilde{\mathsf{H}}_{f\circ\mu_R}(0) = -\frac{1}{2}Q^\top\left((I\otimes R^\top C) + (I\otimes C^\top R)\right)Q. \tag{24}$$

4.3. **Critical points.** Recall f from (10), the element $R = \mu(0)$ is a critical point of f if and only if the following holds,

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

A positive definite Hessian matrix $\mathsf{H}_{f\circ\mu_R}(0) > 0$ indicates that R is a local minimum. In the noise free case, $\widetilde{\mathsf{H}}_{f\circ\mu_R}(0) = 0$ and hence $\mathsf{H}_{f\circ\mu_R}(0) = \widehat{\mathsf{H}}_{f\circ\mu_R}(0) \geqslant 0$. For generic objects we expect that the number of critical points is finite, and that each is isolated.

4.4. **Newton decrement.** The Newton decrement δ is defined in terms of the gradient $\nabla_{f\circ\mu_R}(0)$ and the Hessian $\mathsf{H}_{f\circ\mu_R}(0)$, as

$$\delta := \sqrt{[\nabla_{f\circ\mu_R}(0)]^\top[\mathsf{H}_{f\circ\mu_R}(0)]^{-1}\nabla_{f\circ\mu_R}(0)}. \tag{25}$$

This decrement δ approaches zero as the algorithm converges to a local or global minimum. It features in the work of Nesterov [14] 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 [14] does not apply immediately for optimization on a manifold, yet since manifolds are locally Euclidean, it can be used as a guideline. In [7], 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, we conservatively estimate a *domain of attraction* as $\delta \ll \frac{3-\sqrt{5}}{2}$. Also, as curvature of the manifold increases, it makes sense to use more conservative values. We use

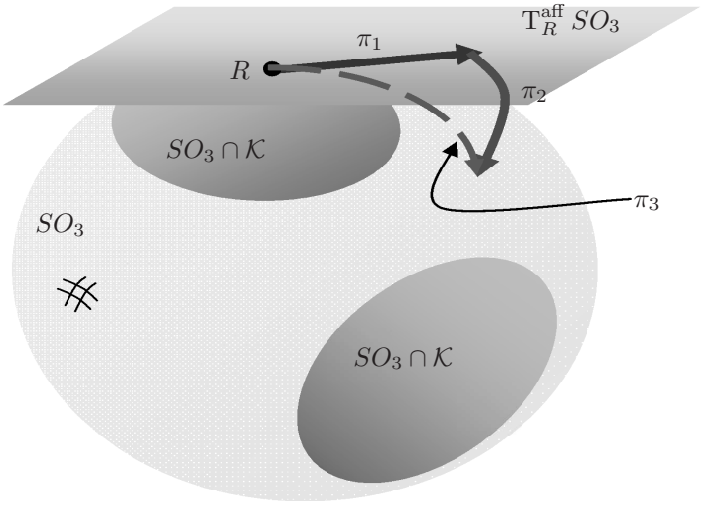


FIGURE 5. Each iteration of the proposed algorithm consists of three mappings, namely π_1 maps a point $R \in (SO_3 \cap \mathcal{K})$ to an element of the affine tangent space $\mathrm{T}_R^{\mathrm{aff}}\, SO_3$, followed by π_2 which projects that vector back to the manifold and π_3 which carries out geodesic search on SO_3 in the direction of the projected vector.

Newton decrement as an indicator on selecting the appropriate direction of geodesic search as,

$$\begin{aligned} \text{Gradient direction} &: \delta \geqslant \epsilon_1 \ \ (\text{eg. } 10^{-1}), \\ \text{Gauss direction} &: \epsilon_1 > \delta > \epsilon_2 \ \ (\text{eg. } 10^{-2}), \\ \text{Newton direction} &: \delta \leqslant \epsilon_2. \end{aligned} \tag{26}$$

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, as

$$\begin{aligned} \text{Newton step size}{=}\|\omega_{\text{Newton}}\| &: \delta < \epsilon_3 \ \ (\text{eg. } 10^{-3}), \\ \text{Terminate} &: \delta < \epsilon_4 \ \ (\text{eg. } 10^{-6}). \end{aligned} \tag{27}$$

4.5. **Algorithm Description.** The proposed algorithm is iterative in nature. Each iteration consists of three mappings as,

$$s = \pi_3 \circ \pi_2 \circ \pi_1 : (SO_3 \cap \mathcal{K}) \rightarrow (SO_3 \cap \mathcal{K}). \tag{28}$$

At each iteration, a local parameterization μ_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 μ_R. The optimal vector that minimizes the quadratic model of the local cost function $f \circ \mu_R$, achieved by the operation π_1, is then pushed forward to the manifold via the mapping π_2. Finally, in operation π_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 $\mathrm{T}_R^{\mathrm{aff}}\, SO_3$, the first two steps of the algorithm can also be

interpreted geometrically as carrying out an optimization procedures defined on $\mathrm{T}_R^{\mathrm{aff}} SO_3$, followed by a nonlinear projection back to the manifold to give a geodesic search direction, as illustrated in Fig. 5.

4.5.1. *Optimization in local parameter space.* Consider the optimization step,

$$\begin{aligned} \pi_1 : (SO_3 \cap \mathcal{K}) &\rightarrow \left(SO_3 \cap \mathcal{K},\ \mathrm{T}^{\mathrm{aff}}\, SO_3\right) \\ R &\mapsto \left(R,\ R + R\Omega(\omega_{\mathrm{opt}}(R))\right), \end{aligned} \tag{29}$$

where ω_{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 ω_{opt} calculation are of interest to our algorithm.

1. *Newton direction.* First, we have the Newton direction, which minimizes the quadratic model of the local cost function,

$$\begin{aligned} \omega_{\mathrm{opt}}^{\mathrm{Newton}}(R) &= \arg\min_{y \in \mathbb{R}^3}\ j_0^{(2)}(f \circ \mu_R)(y) \\ &= -[\mathsf{H}_{f\circ\mu_R}(0)]^{-1} \nabla_{f\circ\mu_R}(0). \end{aligned} \tag{30}$$

2. *Gauss direction.* Consider the restriction of the cost function g (12) to the affine tangent space $T^{\mathrm{aff}} SO_3$,

$$h : \mathrm{T}^{\mathrm{aff}}\, SO_3 \to \mathbb{R}, \quad h(\xi) = g|_{\mathrm{T}^{\mathrm{aff}}\, SO_3} = \frac{1}{2}\|\mathcal{D}\mathrm{vec}(\xi)\|^2,$$

and the function h expressed in the local parameter space,

$$\begin{aligned} \nu_R : \mathbb{R}^3 \to \mathrm{T}^{\mathrm{aff}}\, SO_3, \quad y \mapsto R + R\Omega(y), \\ h \circ \nu_R : \mathbb{R}^3 \to \mathbb{R}, \quad h \circ \nu_R(y) = \frac{1}{2}\|\mathcal{D}\mathrm{vec}(R + R\Omega(y)\|^2. \end{aligned}$$

The Gauss direction is the minimizer of the function $h \circ \nu_R$,

$$\begin{aligned} \omega_{\mathrm{opt}}^{\mathrm{Gauss}}(R) &= \arg\min_{y \in \mathbb{R}^3} (h \circ \nu_R)(y), \\ &= -[\mathsf{H}_{h\circ\nu_R}(0)]^{-1} \nabla_{h\circ\nu_R}(0), \\ &= -[\widehat{\mathsf{H}}_{f\circ\mu_R}(0)]^{-1} \nabla_{f\circ\mu_R}(0). \end{aligned} \tag{31}$$

3. *Gradient direction.* The third descent direction of interest to us is in the negative gradient direction,

$$\omega_{\mathrm{opt}}^{\mathrm{gradient}}(R) = -\nabla_{f\circ\mu_R}(0). \tag{32}$$

Remark 4.1. *When* $\mathsf{H}_{f\circ\mu_R}(0)$ *or* $\widehat{\mathsf{H}}_{f\circ\mu_R}(0)$ *is singular, pseudo inverse replaces inverse in the above equations.*

4.5.2. *Projecting Back via Parameterization.* The mapping π_2 projects the optimal affine tangent vector back to the manifold by means of the parameterization μ_R,

$$\begin{aligned} \pi_2 : \left((SO_3 \cap \mathcal{K}),\ \mathrm{T}^{\mathrm{aff}}\, SO_3\right) &\rightarrow SO_3 \\ \left(R,\ R + R\Omega(\omega_{\mathrm{opt}}(R))\right) &\mapsto R e^{\Omega(\omega_{\mathrm{opt}}(R))}. \end{aligned} \tag{33}$$

4.5.3. *Analytic Geodesic Search on SO_3.* The mapping π_3 involves a one dimensional search along the geodesic curve, $Re^{\theta\Omega(\omega_{\text{opt}}(R))}$,

$$\begin{array}{rcl} \pi_3 : SO_3 & \rightarrow & SO_3 \cap \mathcal{K}, \\ Re^{\Omega(\omega_{\text{opt}}(R))} & \mapsto & Re^{\theta_{\text{opt}}\Omega(\omega_{\text{opt}}(R))}, \end{array} \tag{34}$$

where θ_{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. An important result for us, which does not generalize to matrices higher order than 3×3, is as follows.

Lemma 4.1. *Given a vector $\omega \in \mathbb{R}^3$, a 3×3 skew symmetric matrix $\Omega \in \mathfrak{so}_3$, and a step size $\theta \in [0, 2\pi)$, with $\bar{\omega} := \frac{\omega}{\|\omega\|}$, $\|\omega\| := \sqrt{(\omega^\top \omega)}$, then*

$$\text{vec}(e^{\theta\Omega(\bar{\omega})}) = G \begin{bmatrix} \cos(\theta) \\ \sin(\theta) \\ 1 \end{bmatrix}, \; G := \begin{bmatrix} -\text{vec}(\Omega(\bar{\omega})^2) & \text{vec}(\Omega(\bar{\omega}) & \text{vec}(I_3 + \Omega(\bar{\omega})^2) \end{bmatrix} . \tag{35}$$

Moreover, the function (35) is 2π periodic in θ, so that for any integer k,

$$e^{2\pi k\Omega(\bar{\omega})} = I.$$

Proof. The proof follows from Rodrigues rotation formula [13]. □

To apply the results of Lemma 4.1, consider the cost function f restricted to the geodesic $Re^{\theta\Omega(\bar{\omega}_{\text{opt}}(R))}$ given as,

$$\begin{array}{rcl} \varphi(\theta) & = & f|_{Re^{\theta\Omega(\bar{\omega}_{\text{opt}}(R))}} = \frac{1}{2}\|\mathcal{D}\,\text{vec}(Re^{\theta\Omega(\bar{\omega}_{\text{opt}}(R))})\|^2, \\ & = & \frac{1}{2}\|\mathcal{D}(I \otimes R)G \begin{bmatrix} \cos(\theta) & \sin(\theta) & 1 \end{bmatrix}^\top \|^2. \end{array} \tag{36}$$

Now, the task is to 'walk' along the geodesic on the manifold SO_3 and search for a step size θ that minimizes the cost function $\varphi(\theta)$ and satisfies the cone constraint. To achieve this, we find all critical step sizes $\{\theta_*\}$ by setting the first derivative of the cost function (36) to zero. Denoting $G^\top(I \otimes R^\top)\mathcal{D}^\top\mathcal{D}(I \otimes R)G$ as (a_{ij}),

$$\frac{\mathrm{d}}{\mathrm{d}\,\theta}\varphi(\theta) = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 1 \end{bmatrix} (a_{ij}) \begin{bmatrix} \sin(\theta) \\ \cos(\theta) \\ 0 \end{bmatrix} = 0,$$

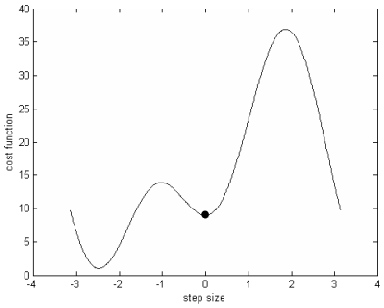
by eliminating $\sin(\theta)$ from the above equation using trigonometric formula $\cos^2(\theta) = 1 - \sin^2(\theta)$, we obtain a quartic equation,

$$\sum_{j=0}^{4} b_j(\cos(\theta))^j = 0, \tag{37}$$

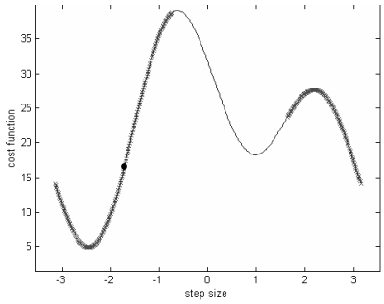
$$\begin{aligned} b_0 &:= {a_{12}}^2 - {a_{13}}^2, \quad b_1 := -2\left(a_{12}a_{23} + a_{13}(a_{11} - a_{22})\right), \\ b_2 &:= {a_{23}}^2 + {a_{13}}^2 - (a_{11} - a_{22})^2 - 4{a_{12}}^2, \\ b_3 &:= 2\left(2a_{12}a_{23} + a_{13}(a_{11} - a_{22})\right), \quad b_4 := 4{a_{12}}^2 + (a_{11} - a_{22})^2. \end{aligned}$$

Next, the critical step size that gives minimum cost and fulfills the cone constraint $\mathcal{K}$, denoted the optimum step size θ_{opt} is given as,

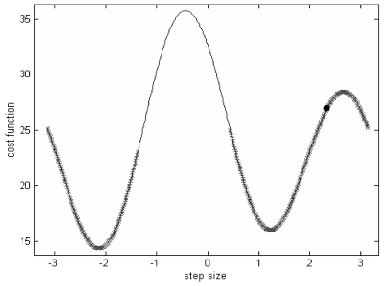
$$\theta_{\text{opt}} = \arg\min_{\theta} \ \varphi(\theta), \quad \text{where } \theta \in \{\theta_* \mid Re^{\theta_* \Omega(\bar{\omega}_{\text{opt}}(R))} \in (SO_3 \cap \mathcal{K})\}. \tag{38}$$



(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.

FIGURE 6. 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, i.e., object behind camera.

Figure 6 shows the plots of cost function $\varphi(\theta)$ vs. step size θ 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. 6(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 θ corresponding to the minimum cost in the search direction and hence escapes from the local minimum and heads towards the global minimum. In Fig. 6(b), the minimum cost lies in the infeasible region, so the geodesic search will select the θ 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. 6(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.

5. **Algorithm Initialization.**

5.1. **Noise free solution.** In the noise-free case, the optimal value of the cost function (10) is zero. That is there exists an $R \neq 0$ such that

$$\mathcal{D}^\top \mathcal{D}\mathrm{vec}(R) = 0. \tag{39}$$

This clearly indicates that $\mathcal{D}^\top\mathcal{D} \geqslant 0$ cannot be full rank. Subsequently, the rank of $\mathcal{D}$ is of interest. For the generic case,

$$\mathrm{rank}(\mathcal{D}) \leqslant \min\{(2n-3), 9\}. \tag{40}$$

Derivation of this inequality can be found in Appendix A.

Let Y be the set of right singular vectors that span the null space of $\mathcal{D}$, with $Y^\top Y = I_q$ with $q \geqslant 1$, we express $\mathrm{vec}(R)$ linearly in Y as follows,

$$\mathrm{vec}(R) = Y\alpha, \quad R = \begin{bmatrix} Y_1\alpha & Y_2\alpha & Y_3\alpha \end{bmatrix}, \tag{41}$$

where Y is a $9 \times q$ matrix,

$$Y := \begin{bmatrix} Y_1^\top & Y_2^\top & Y_3^\top \end{bmatrix}^\top, \quad \alpha := \begin{bmatrix} \alpha_1 & \alpha_2 & \dots & \alpha_q \end{bmatrix}^\top.$$

Here Y_i is a $3 \times q$ matrix. Based on R from (41) and the fact that $R^\top R = I$, then

$$\alpha^\top Y_k^\top Y_l\, \alpha = \begin{cases} 1, & \text{if } k = l, \\ 0, & \text{otherwise.} \end{cases} \tag{42}$$

Hence, we can solve for α from (42). A lower bound for q, the dimension of α, is calculated using (40) as

$$q := (9 - \mathrm{rank}(\mathcal{D}) \geqslant \max(1, 9 - (2n-3)), \tag{43}$$

with equality holding in the generic model case. This is consistent with the known result that a unique solution (39) in the generic noise-free case for $n = 6$, since then $q = 1$.

In the generic case for $n \geqslant 6$ then $q = 1$. There are six possible α solutions from (42) as follows,

$$\alpha = \pm\frac{1}{\sqrt{\mathcal{Y}_{11}}} = \pm\frac{1}{\sqrt{\mathcal{Y}_{22}}} = \pm\frac{1}{\sqrt{\mathcal{Y}_{33}}}. \tag{44}$$

Each solution is tested to achieve a feasible parameter estimate which yields a zero cost. For $n = 5$ in generic case, $q = 2$. In this case, we can obtain the two elements of α by solving a pair of quadratic equations. Based on (42), there are 9 pairs of quadratic equations to be solved. To illustrate, consider the following pair,

$$\alpha^\top \mathcal{Y}_{33}\alpha = 1, \tag{45}$$

$$\alpha^\top \mathcal{Y}_{12}\alpha = 0. \tag{46}$$

We proceed as follows,

Step 1:: Let $\alpha = S_\alpha[\alpha_1 \;\; 1]^\top$, S_α is a scaling factor, solve for α_1 from (46).

Step 2:: Solve for S_α by substituting α_1 into $\alpha^\top \mathcal{Y}_{33}\alpha = 1$ and finally solve for α.

Here we can obtain up to 4 possible solutions for α. Among all possible solutions, half of them are numerically equivalent but associated with opposite signs.

For $n = 4$ and $n = 3$, the details of solutions for R and t (not unique for $n = 3$) can be found in [15] and [2] respectively. The cases $n = 1$ and $n = 2$ yield an infinite set of solutions. Once α is known, we can solve for $R(\alpha),\ z_i',\ t$ from (41), (8) and (7) respectively. There may be multiple solutions, including possibly complex solutions for α. Only the real α that results in $z_i' > 0$ and $R(\alpha)$ associated with minimum cost will be accepted.

5.2. **Low noise initialization.** We solve for α as in the noise-free case. However, now Y is chosen from the set of right singular vectors associated with the q smallest singular values, which may not be zero. The singular vectors might not result in a special orthogonal matrix R based on (41). Hence, a reasonable approach is to look for an optimum $R_{\text{opt}}(\alpha)$ as

$$\begin{aligned} R_{\text{opt}}(\alpha) &= \arg\min_{R\in SO_3} \|\text{vec}(R) - Y\alpha\|, \\ &= \arg\min_{R\in SO_3} \|R - [Y_1\alpha \;\; Y_2\alpha \;\; Y_3\alpha]\|, \\ &= \arg\max_{R\in SO_3} \text{tr}(R^\top G(\alpha)), \end{aligned} \tag{47}$$

where $G(\alpha) := [Y_1\alpha \;\; Y_2\alpha \;\; Y_3\alpha]$. By applying an SVD on $G(\alpha)$, we obtain

$$\begin{aligned} G(\alpha) &= U_R S_R V_R^\top, \\ R_{\text{opt}}(\alpha) &:= U_R \begin{bmatrix} I_2 & 0 \\ 0 & \det(U_R V_R^\top) \end{bmatrix} V_R^\top. \end{aligned} \tag{48}$$

Observe that $\det(R_{\text{opt}}(\alpha)) = 1$. Subsequently, t can be obtained from (7). Only $R_{\text{opt}}(\alpha)$ associated with minimum cost and fulfilling the cone constraint $\mathcal{K}$ will be accepted. If there is no estimates $R_{\text{opt}}(\alpha)$ such that the cone constraint is satisfied or there is no real solution for α at all, then we refer to this situation as a high noise case, the following high noise initialization is adopted. Such an optimization as in (47) is the basis of the Orthogonal Iteration (OI) proposed in [11], and indeed the SVD based solutions of (48) which ensures that $\det(R_{\text{opt}}) = +1$, could be used to strengthen the robustness of the OI algorithm.

5.3. **High noise initialization.** Select any random $R \in SO_3$ which also satisfies the cone constraint $\mathcal{K}$ of (9). Alternatively, one can start with an arbitrary initialization on SO_3, perhaps not feasible in that the cone constraint is not satisfied, and achieve feasibility using the geodesic search.

6. **Implementation of the Algorithm.** Start with $R = R_0$ using initialization procedures described in Section 5.

Step 1: Carry out the optimization step,

- Compute the gradient $\nabla_{f\circ\mu_R}(0)$ and the Hessian, $\mathsf{H}_{f\circ\mu_R}(0)$ via (20) and (22) respectively.
- Calculate the Newton decrement δ from (25). If $\delta < \epsilon_4$, go to Step 5,

- Compute the optimal direction in the local parameter space,

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

  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 local parameterization μ_R, as described in (33),

Step 3: Carry out a one dimensional search along the geodesic $Re^{\theta\Omega(\bar{\omega}_{\text{opt}}(R))}$,
- Solve the quartic equation in (37) to obtain all critical step sizes $\{\theta_*\}$,
- For each $\theta \in \{\theta_*\}$, calculate the corresponding rotation $Re^{\theta\Omega(\bar{\omega}_{\text{opt}}(R))}$, its cost via (36) and the depth parameter estimates via (8).
- Obtain the critical rotation angle θ_{opt} that gives minimum cost and fulfills the cone constraint $\mathcal{K}$ via (38).
- Compute $\widehat{R} = Re^{\theta_{\text{opt}}\Omega(\bar{\omega}_{\text{opt}}(R))}$,

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.

7. Convergence Analysis of Algorithm.

7.1. 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 $\{R \in SO_3 \,|\, f(R) = c\}$ 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. Even so for practical purposes, since the algorithm is only implemented for say 5–10 iterations, there may well be a low probability that there is 'convergence' to a local minima which is not a global minimum. Given this background, we provide only a rigorous quadratic convergence analysis in the vicinity of a global minimum, and for simplicity assume that the global minimum is unique and isolated, as we expect in the case of generic objects located in front of the camera.

7.2. Local quadratic convergence at global minimum.

Theorem 7.1. *Let $R_* \in SO_3$ be the unique and nondegenerate global minimum of the smooth function $f : SO_3 \to \mathbb{R}$ defined in* (10). *Let R_k be a point in an open*

neighbourhood of R_*. Consider the proposed iteration on SO_3,

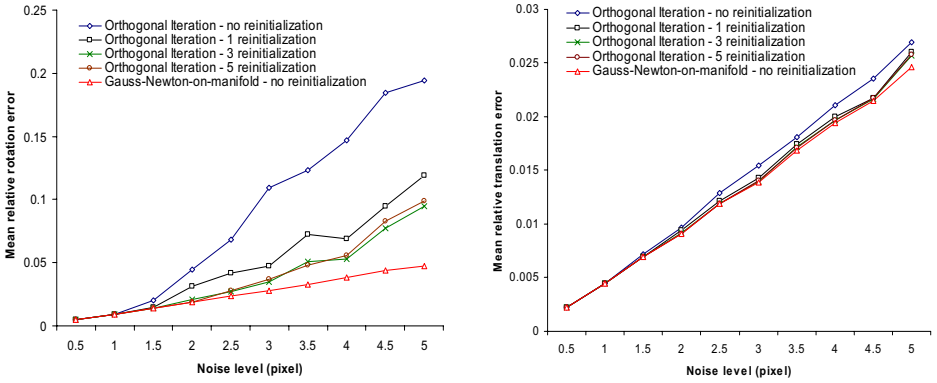
$$R_{k+1} = s(R_k), \quad s = \pi_3 \circ \pi_2 \circ \pi_1, \tag{49}$$

where π_1 is given by the Newton direction defined in (30)*, π_2 involves projection back to SO_3 via the smooth exponential map of* (33)*, and π_3 is an analytic geodesic search described in* (34)*. Then the point sequence generated by s converges quadratically to R_*.*

Proof. See Appendix B. □

8. **Simulations.** A set of 3D points that are generated uniformly within a cube of size [-5,5] are chosen as the model. The model is rotated and translated randomly within the field of view of the camera and then projected onto an image plane. Gaussian noise of mean zero and varying standard deviations σ is added to both image plane coordinates independently. The different standard deviations in the Gaussian noise corresponds to various noise levels. All simulations are repeated for 1000 times. A CCD camera with focal length of 600, aspect ratio of 1.0 and principal point of (256,256) is used. All simulations are carried out in MATLAB. Throughout the simulation, we define the relative pose error as the normalized distance between the actual pose and its estimate. That is, $E_A := 2\frac{\|A-\hat{A}\|}{\|A\|+\|\hat{A}\|}$, $A \in \{R, t\}$. The performance of the Gauss-Newton-on-manifold algorithm is compared against the Orthogonal Iteration (OI) algorithm of [11].

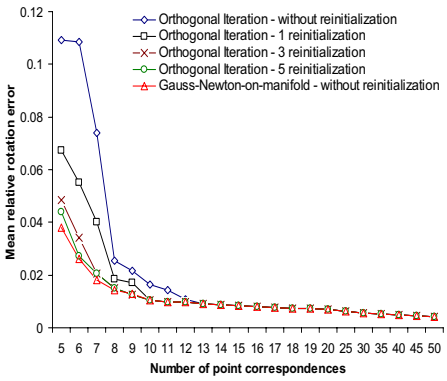
8.1. **Relative pose error vs. noise level.** Points and poses are generated as described earlier. With the number of matched points fixed at 12, we vary the noise level between 0.5 to 5.0 pixels. It is clear from Fig. 7 that the Gauss-Newton-on-



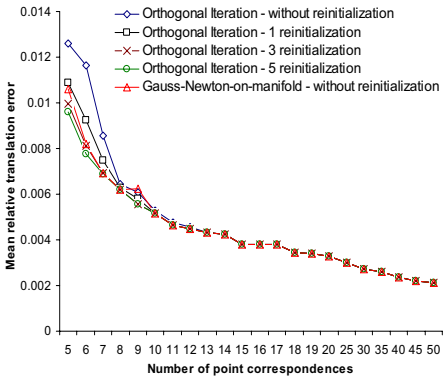
(a) Relative rotation error vs. noise level. (b) Relative translation error vs. noise level.

FIGURE 7. Relative pose error vs. noise level.

manifold approach (without reinitialization) performs significantly better than the OI algorithm in terms of relative pose error, even after the OI algorithm has been reinitialized 5 times. This is particularly the case for noise level greater than 1.5 pixel.



(a) Relative rotation error vs. number of point correspondences.

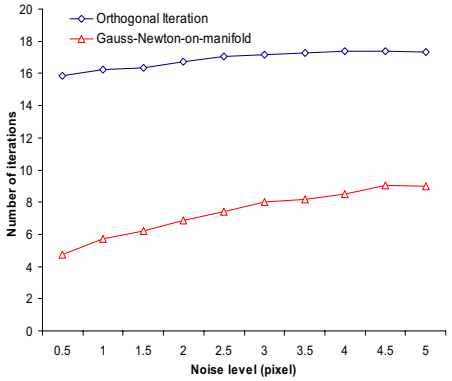


(b) Relative translation error vs. number of point correspondences.

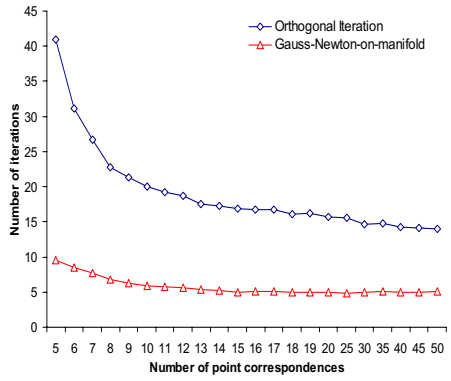
FIGURE 8. Relative pose error vs. number of point correspondences.

8.2. **Relative pose error vs. number of points.** Here, we vary the number of point correspondences from 5 to 50 and add 1.0 x 1.0 pixel Gaussian noise to the images. Figure 8 indicates that the performance of both algorithms improve as the number of point matches increase. However, the proposed recursions (without reinitialization) gives better pose estimates than the OI algorithm, even after 5 reinitialization, for point correspondences less than 15.

8.3. **Number of iterations.** Figure 9 shows the number of iterations (without reinitialization) required by each method to achieve the performance shown in Fig. 7 and Fig. 8. It is clear that the Gauss-Newton-on-manifold approach always converges in 5–10 iterations as opposed to the OI algorithm that requires at least 10 iterations.



(a)



(b)

FIGURE 9. Number of iterations vs. noise level and number of corresponding points.

8.4. **Rate of local convergence.** Figure 10 shows that the proposed Gauss-Newton-on-manifold algorithm converges to a fixed point R_* at a quadratic rate as opposed to the OI algorithm which converges at a fast linear rate.

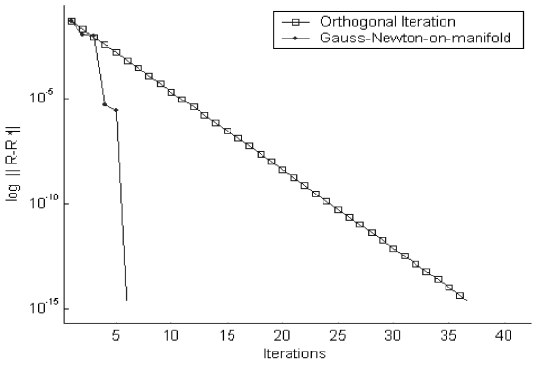


FIGURE 10. Rate of local convergence.

Remark 8.1. *The extra cost per iteration is small since implementing the Newton step with* 3×3 *Hessian matrices in not onerous, nor is the solution of a quartic equation for the optimal step size selection. We observe the computational cost per iteration of OI algorithm is much less than ours when implemented using Matlab. But we expect that when implemented in C/C++, the relative computational cost of our algorithm will improve significantly.*

9. **Algorithm Robust to Outliers.** Recall that the proposed algorithm minimizes the sum of squared residuals,

$$\frac{1}{2}\sum_{i=1}^{n} r_i^2, \quad r_i = \|D_i \mathrm{vec}(R)\|,$$

and gives accurate fitting of data when the underlying noise in the data is Gaussian but is very vulnerable to outliers.

To avoid this, a robust version of the algorithm which adapt the M-estimator technique proposed by Huber [6] is presented. The M-estimator technique is based on the idea of reducing the effect of outliers by replacing the squared residuals, r_i^2, with another function of the residuals, yielding the cost function

$$\frac{1}{2}\sum_{i=1}^{n} \rho(r_i), \tag{50}$$

where ρ is a symmetric, positive function with a unique minimum at zero and is chosen to be growing slower than the squared function. The solution to this problem can be found by setting its derivative with respect to the unknown parameter vector θ to zero,

$$\sum_{i=1}^{n} \frac{\mathrm{d}\rho}{\mathrm{d}r_i}\frac{\partial r_i}{\partial \theta_j} = 0 \tag{51}$$

Defining a weight function

$$w_i = r_i \frac{\mathrm{d}\rho}{\mathrm{d}r_i},$$

then (51) becomes

$$\sum_{i=1}^{n} w_i r_i \frac{\partial r_i}{\partial \theta_j} = 0.$$

Observe that this is the solution to the following problem

$$\min \sum_{i=1}^{n} w_i r_i^2.$$

In practice, instead of solving (50), it is more efficient to solve a sequence of reweighted least squares problem. Thus, to increase the robustness of the proposed algorithm against outlier, we proceed to minimize the following weighted cost function

$$\begin{aligned} f_w : SO_3 &\rightarrow \mathbb{R}, \\ f_w(R) &= \frac{1}{2}\sum_{i=1}^{n} w_i \|D_i \mathrm{vec}(R)\|^2 = \frac{1}{2}\|\mathcal{D}\mathrm{vec}(R)\|_W^2, \end{aligned}$$

where $W = \mathrm{diag}(w_1, \cdots, w_n) \otimes I_3$.

Among the weighting function proposed in the statistics literature, Huber's weight function is given as

$$w_i = \begin{cases} 1 & \text{if } |r_i| \leqslant cs \\ \frac{cs}{|r_i|} & \text{otherwise} \end{cases}, \quad (c = 1.345),$$

and Tukey's biweight function is

$$w_i = \begin{cases} \left(1 - \left(\frac{r_i}{cs}\right)^2\right)^2 & \text{if } |r_i| \leqslant cs \\ 0 & \text{otherwise} \end{cases}, \quad (c = 4.6851).$$

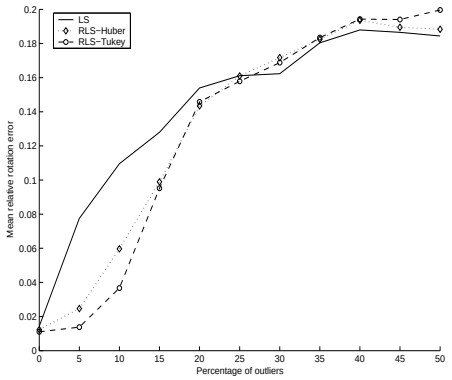
Here the scale

$$s = \frac{\mathrm{median}_i |r_i|}{0.6745}$$

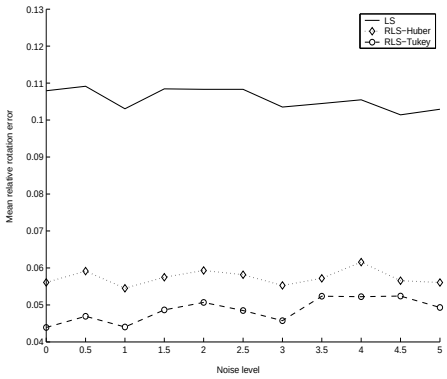
is introduced to give a scale invariant version of an M-estimator and the value 0.6745 is one half of the interquartile range of the Gaussian normal distribution $N(0, 1)$.

To evaluate the robustness against outliers, the algorithm which minimizes a least squares cost function (LS) is compared against the algorithm which minimizes the reweighted least squares cost function (RLS) using Huber and Tukey weights. Data are generated as in earlier simulation, but now the number of point correspondences is fixed at 20. The outliers are generated by shifting certain image points at least 50 pixels from their actual location. In addition, Gaussian noise of mean zero and σ pixel standard deviation is added to the rest of the point correspondences. 1000 trials were carried out.

Figure 11(a) shows the plot of mean relative rotation error vs. percentage of outliers with $\sigma = 1$ pixel. In the absence of outliers, both LS and RLS give similar performance. In the presence of less than 20% of outliers, RLS gives significantly better performance than LS but when the percentage of outliers greater than 25%, we observe no advantage of using RLS against LS. Figure 11(b) shows the plot of mean relative rotation error vs. noise level with 5% outliers. This plot indicates that even in the presence of a small relative number of outliers, RLS outperforms LS significantly regardless of the noise level. Also among RLS methods, Tukey's biweight functions seems to give better performance than Huber's weight function.



(a) Mean relative rotation error vs. percentage of outliers with $\sigma = 1$ pixel.



(b) Mean relative rotation error vs. noise level with 5% outliers.

10. **Conclusion.** In this paper, the task of estimating position and orientation of a 3D object from single 2D image has been formulated as an unconstrained optimization problem cast on the intersection of a rotation group and a cone constraint. Newton-type algorithms based on the proposed geometrical framework have been developed. The techniques take into account the underlying geometry of the constraint manifold to achieve 'best possible' speed of convergence, indeed local quadratic convergence. Unlike most existing numerical procedures, the proposed algorithms evolve on the constraint manifold and thus preserve constraints at every iteration. The key differentiating features, each adding measurable value to the algorithm, concern

- A new approach of obtaining noise-free solutions for $n \geqslant 5$, which can be used for low noise initializations,
- The use of 3×3 Hessian inversion, as opposed to the conventional Newton approach that works with a 6×6 Hessian matrix [9],
- The analytic geodesic search for step size selection, requiring a solution of a quartic, and facilitating escape from a local minimum but not from the global minimum,
- Achieving local quadratic convergence by means of simulations and mathematical proof,
- No need to reinitialize the algorithm to achieve global minimum within the feasible region,
- 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.
- To increase robustness against outliers, the proposed algorithm adapting the M-estimator technique is also presented. It is not easy to derive analytical results on this robustness, but simulations appear convincing.

These features suggest that as digital signal processing hardware becomes more powerful, the algorithm can be conveniently applied to on-line pose recovery tasks, and can be a basic tool for more sophisticated tasks involving adaptive object recognition and pose tracking.

11. **Appendix A: Rank of $\mathcal{D}$.** Consider $\mathcal{D}$ as defined in (11). In order to evaluate $\text{rank}(\mathcal{D})$, observe that $\mathcal{D}$ can be reformulated as

$$\mathcal{D} = \hat{U}(M^\top \otimes I),$$

where $M := \begin{bmatrix} m_1 & m_2 & \cdots & m_n \end{bmatrix}$, and denoting $\mathbf{1}_n$ as $n \times 1$ vector consists of 1,

$$\begin{aligned} \hat{U} &:= \text{diag}(\tilde{U}_1, \cdots, \tilde{U}_n)(I - (\mathbf{1}_n \otimes \bar{A})), \;\; \tilde{U}_i := I - U_i, \\ \bar{A} &:= (\mathcal{A}^\top \mathcal{A})^{-1} \mathcal{A}^\top \begin{bmatrix} \tilde{U}_1 & \cdots & \tilde{U}_n \end{bmatrix}. \end{aligned}$$

Then,

$$\text{rank}(\mathcal{D}) \leqslant \min\{\text{rank}(\hat{U}), \text{rank}(M^\top \otimes I)\}$$

To analyze the minimum rank of $\hat{U}$, observe that

$$\hat{U}X = 0, \tag{52}$$

where

$$X := \begin{bmatrix} I & \tilde{U}_1 & I & \cdots & I \\ \vdots & I & \tilde{U}_2 & & \vdots \\ & & & \ddots & \\ I & \cdots & & & \tilde{U}_n \end{bmatrix}.$$

Elementary row and column operations on X gives that $\text{rank}(X) = (n+3)$ in the generic case when $\{\mathbf{u}_i\}_{\{i=1,\ldots,n\}}$ are linearly independent. Since $\hat{U}$ has full rank of $3n$ and it can loose at most $(n+3)$ rank, then $\text{rank}(\hat{U}) = 3n-(n+3)$. The necessary conditions for such existence is that $3n-(n+3) \geqslant 0$ or $n \geqslant 2$. For generic models when $\text{rank}(M) = 3$, then

$$\begin{aligned} \text{rank}(\mathcal{D}) &\leqslant \min\{(3n-(n+3)), (3 \times \text{rank}(M))\} \\ &\leqslant \min\{(2n-3), 9\}. \end{aligned}$$

12. **Appendix B: Proof of Theorem 7.1.**

12.1. **Fixed Point of the Algorithm.** Let $R_* \in SO_3$ be a fixed point of $s = \pi_3 \circ \pi_2 \circ \pi_1$, we have

$$s(R_*) = R_* \;\Leftrightarrow\; \pi_3 \circ \pi_2 \circ \pi_1(R_*) = R_* \;\Leftrightarrow\; R_* e^{\Omega(\theta_{\text{opt}}\omega_{\text{opt}})} = R_* \;\Leftrightarrow\; e^{\Omega(\theta_{\text{opt}}\omega_{\text{opt}})} = I.$$

If the exponential map in the projection step π_2 is within its injectivity radius, then only $\omega_{\text{opt}} = 0$ satisfies the above equation (since θ_{opt} is a positive scalar). Thus, the only fixed points of the algorithm are critical points. However, as noted in the text only local minima are stable fixed points. Indeed with our geodesic search feature, only global minimum is a stable fixed point. Notice that $\omega_{\text{opt}} = 0$ is the (unique) minimum of $j_0^{(2)}(f \circ \mu_{R_*})(\omega)$ if and only if $\mu_{R_*}(0) = R_*$ is a non-degenerate minimum of $f : SO_3 \to \mathbb{R}$.

12.2. **Smoothness Properties of the Algorithm.** Under the assumption that the cost function f is smooth and Hessian of $f \circ \mu_R$ is invertible everywhere, the optimization step π_1 is smooth. The projection step π_2 which involves only the exponential mappings is also smooth. Although the operation π_3 is designed to be discontinuous to escape local minima not a global minimum, yet in a sufficiently small neighbourhood of R_*, the operation π_3 is continuous since all critical points other than R_* have a higher cost than the current cost. It is then more than twice

differentiable since it concerns the isolated minimal cost solution of a polynomial equation.

12.3. **Local Quadratic Convergence of the Algorithm.** Let R_* denote a stable fixed point of $s = \pi_3 \circ \pi_2 \circ \pi_1$, being also the unique and non-degenerate global minimum of the function f, as already established under our assumptions. We will compute the first derivative of s at this fixed point. Applying the chain rule to the algorithmic mapping $s = \pi_3 \circ \pi_2 \circ \pi_1$, and using the fact that $\pi_1(R_*) = \pi_2(R_*) = \pi_3(R_*) = R_*$, for all elements $\xi \in \mathrm{T}_{R_*} SO_3$, the first derivative of s at fixed point R_* is,

$$\mathrm{D}\, s(R_*) \cdot \xi = \mathrm{D}\, \pi_3(R_*) \cdot \mathrm{D}\, \pi_2(R_*) \cdot \mathrm{D}\, \pi_1(R_*) \cdot \xi. \tag{53}$$

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

$$\mu_{R_*}^{-1} \circ s \circ \mu_{R_*} : \mathbb{R}^3 \to \mathbb{R}^3. \tag{54}$$

Thus, rewriting (53) in terms of local parameterization defined by

$$\mu_{R_*} : \mathbb{R}^3 \to SO_3, \quad \omega \mapsto R_* e^{\Omega(\omega)}, \tag{55}$$

with Ω as in (13), we have

$$\begin{aligned} & \mathrm{D}\, \mu_{R_*}^{-1} \circ s \circ \mu_{R_*}(0) \cdot h \\ = \;& \mathrm{D}\, \mu_{R_*}^{-1}(R_*) \cdot \mathrm{D}\, s(R_*) \cdot \mathrm{D}\, \mu_{R_*}(0) \cdot h, \\ = \;& \mathrm{D}\, \mu_{R_*}^{-1}(R_*) \cdot \mathrm{D}\, \pi_3(R_*) \cdot \mathrm{D}\, \pi_2 \circ \pi_1 \circ \mu_{R_*}(0) \cdot h. \end{aligned} \tag{56}$$

Consider the composite function

$$\pi_2 \circ \pi_1 \circ \mu_{R_*} : \mathbb{R}^3 \to SO_3, \quad \omega \mapsto \mu_{R_*}(\omega) e^{\Omega\left(\omega_{\mathrm{opt}}^{\mathrm{Newton}}(\mu_{R_*}(\omega))\right)}. \tag{57}$$

where

$$\omega_{\mathrm{opt}}^{\mathrm{Newton}} \circ \mu_{R_*}(\omega) = \omega_{\mathrm{opt}}^{\mathrm{Newton}}(\mu_{R_*}(\omega)) = -[\mathsf{H}_{f \circ \mu_{R_*}}(\omega)]^{-1} \nabla_{f \circ \mu_{R_*}}(\omega). \tag{58}$$

Exploiting linearity of the mapping Ω, using the well known formula for differentiating the matrix exponential and the fact that

$$\mu_{R_*}(0) = R_*, \quad \omega_{\mathrm{opt}}^{\mathrm{Newton}}(R_*) = 0,$$

we have

$$\mathrm{D}\, \mu_{R_*}(0) \cdot h = R_* \Omega(h), \tag{59}$$

and

$$\begin{aligned} & \mathrm{D}\, \omega_{\mathrm{opt}}^{\mathrm{Newton}} \circ \mu_{R_*}(0) \cdot h \\ = \;& -[\mathsf{H}_{f \circ \mu_{R_*}}(0)]^{-1} \mathrm{D}\, \nabla_{f \circ \mu_{R_*}}(0) \cdot h - \mathrm{D}[(\mathsf{H}_{f \circ \mu_{R_*}}(0)]^{-1} \cdot h \nabla_{f \circ \mu_{R_*}}(0), \\ = \;& -[\mathsf{H}_{f \circ \mu_{R_*}}(0)]^{-1} \mathsf{H}_{f \circ \mu_{R_*}}(0) h, \quad \text{since } \nabla_{f \circ \mu_{R_*}}(0) = 0 \\ = \;& -h. \end{aligned} \tag{60}$$

Now, we compute the first derivative of the composite function (57) in the limit as ω approaches zero as,

$$\begin{aligned} \mathrm{D}\, \pi_2 \circ \pi_1 \circ \mu_{R_*}(0) \cdot h &= R_* \Omega(h) + R_* \Omega(\mathrm{D}\, \omega_{\mathrm{Newton}} \circ \mu_{R_*}(0) \cdot h) \\ &= 0. \end{aligned} \tag{61}$$

Substituting (61) into (56) shows that for all $h \in \mathbb{R}^3$

$$\mathrm{D}\, \mu_{R_*}^{-1} \circ s \circ \mu_{R_*}(0) \cdot h = 0. \tag{62}$$

Since the iterate R_k is in an open neighbourhood of R_*, then by inverse mapping, $\omega_k = \mu_{R_*}^{-1}(R_k)$ stays in a sufficiently small open neighborhood of the origin in $\mathbb{R}^3$.

Vanishing of the first derivative then implies local quadratic convergence by the Taylor-type argument, for some positive κ

$$\left\|\mu_{R_*}^{-1} \circ s \circ \mu_{R_*}(\omega_k)\right\| \leqslant \sup_{y \in \overline{\mathcal{N}(0)}} \kappa \left\|\mathrm{D}^2\, \mu_{R_*}^{-1} \circ s \circ \mu_{R_*}(y)\right\| \cdot \|\omega_k\|^2 \tag{63}$$

with $\overline{\mathcal{N}(0)}$ the topological closure of a sufficiently small open neighbourhood of origin in $\mathbb{R}^3$.

Remark 12.1. *The result holds for π_3 being an identity operation as for a Newton step, or is given by a smooth geodesic search. Notice that if the geodesic search is switched off in the neighbourhood of R_*, so that π_3 is an identity operator, then the assumption of uniqueness of R_* can be relaxed for this proof to still hold.*

One might ask: what happens in the nongeneric case when the Hessian matrix is singular at a local minima? In this case, the local minima are not isolated, and one expects quadratic convergence to this connected set by similar but more technical arguments.

REFERENCES

[1] A. Ansar and K. Daniilidis, *Linear pose estimation from points or lines*, International Conference on Computer Animation, Switzerland, June, 2002.

[2] M.A. Fischler and R.C. Bolles, *Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography*, Comm. Assoc. Comp. Mach., (6) 24 (1981), 381–395.

[3] R.M. Haralick, C.N. Lee, K. Otterberg and M. Nölle, *Analysis and Solutions of The Three Point Perspective Pose Estimation Problem*, IEEE conference on Computer Vision and Pattern Recognition, pp. 592–598, Hawaii, June, 1991.

[4] R. Hartley and A. Zisserman, "Multiple View Geometry", Cambridge University Press, 2000.

[5] U. Helmke, K. Hüper, P.Y. Lee and J. Moore, *Essential Matrix Estimation via Newton-type Methods*. Proceedings of the 16th International Conference of Mathematical Theory of Networks and Systems, 2004.

[6] P.J. Huber, "Robust Statistics", John Wiley & Sons, Inc., 1981.

[7] 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, Dec 2004.

[8] P.Y. Lee and J.B. Moore, *Pose Estimation via Gauss-Newton-on-manifold*, Proceedings of the 16th International Symposium on Mathematical Theory of Networks and Systems, Leuven, Belgium, Session TP9, July 2004.

[9] D.G. Lowe, *Three-dimensional object recognition from single two-dimensional images*, Artificial Intell., (3) 31 (1987), 355–395.

[10] D.G. Lowe, *Fitting parametrized three-dimensional models to images*, IEEE Trans. Pattern Anal. Machine Intell., (5) 13 (1991), 441–450.

[11] C.P. Lu, G.D. Hager and E. Mjolsness, *Fast and Globally Convergent Pose Estimation from Video Images*, IEEE Trans. Pattern Anal. Machine Intell., (6) 22 (2000), 610–622.

[12] J.H. Manton, *Optimization algorithms exploiting unitary constraints*, *IEEE Transactions on Signal Processing*, (3) 50 (2000), 635–650.

[13] R.M. Murray, Z. Li and S.S. Sastry, "A Mathematical Introduction to Robotic Manipulation", CRC Press, 1994.

[14] Y. Nesterov, "Introductory Lectures on Convex Optimization - A Basic Course", Kluwer Academic Publishers, 2004

[15] L. Quan and Z. Lan, *Linear N-Point Camera Pose Determination*, IEEE Trans. Pattern Anal. Machine Intell., (8) 21 (1999), 774–780.

Received May 2005; revised August 2005.

E-mail address: PeiYean.Lee@nicta.com.au
E-mail address: John.Moore@anu.edu.au