Minimization on the Lie Group SO(3) and Related Manifolds

Camillo J. Taylor and David J. Kriegman
Yale University

Technical Report No. 9405
April, 1994

Minimization on the Lie Group SO(3) and Related Manifolds [1]

Camillo J. Taylor and David J. Kriegman

Center for Systems Science
Department of Electrical Engineering
Yale University
New Haven, CT 06520-8267
taylor@trurl.eng.yale.edu, kriegman@cs.yale.edu

Abstract

This paper presents a novel approach to carrying out numerical minimization procedures on the Lie Group $SO(3)$ and related manifolds. The approach constructs a sequence of local parameterizations of the manifold $SO(3)$ rather than relying on a single global parameterization such as Euler angles. Thus, the problems caused by the singularities in these global parameterizations are avoided.

[1]This research was supported by the NSF under grant number NYI IRI-9257990.

1 Introduction

A number of interesting problems in computer vision and robotics involve finding the "optimal" rotation matrix in the set of rigid rotations $SO(3) = \{R \in \mathbb{R}^{3\times 3} : R^t R = I, det(R) = 1\}$ [GO89, Hor90]. For example, the problem of determining the position of a camera with respect to a known constellation of feature points from image data can be cast in terms of a real valued objective function

$$\mathcal{O} : SO(3) \to \mathbb{R} \tag{1}$$

which measures how well the proposed rotation matrix R explains the observed image data. Most of these optimization problems do not have closed form solutions, which means that numerical optimization procedures are usually applied in order to find the solutions. [2]

In most numerical minimization paradigms, the unknown parameters are assumed to lie in some vector space isomorphic to $\mathbb{R}^n$. Therefore, in order to apply these optimization techniques, the user constructs some parameterization of the set of rotation matrices, like Euler angles, that maps the vectors in $\mathbb{R}^3$ onto the three-dimensional manifold $SO(3)$. Unfortunately, the Lie group $SO(3)$ is not isomorphic to the vector space $\mathbb{R}^3$ [Stu64]. This means that all of these global parameterizations will exhibit singularities or other anomalies at various points in the parameter space. These anomalies can cause serious problems for gradient based minimization procedures.

There are several approaches to optimization problems on a manifold such as $SO(3)$ that involve calculating incremental steps in the tangent space to the manifold. That is, given an objective function $\mathcal{O} : M \to \mathbb{R}$ and a particular element p on the manifold M we could compute the steepest descent direction, a, in the tangent space to M at p. We would then have to decide on a step size, λ, to use in the update step $p' = p + \lambda a$. In general, the new vector p' will *not* lie on the manifold, so some procedure will be required to rescale p' back onto the manifold.

The technique presented in this paper avoids this rescaling problem entirely by constructing an actual local parameterization of the manifold at each stage rather than a simple linear approximation. This means that the parameterization will map any incremental step in its domain to a distinct point on the manifold. The method also provides a systematic technique for computing the size and direction of the incremental step at each iteration.

Our solution is to forgo the global parameterization approach in favor of an atlas of local parameterizations. That is, at every point R_0 on the manifold $SO(3)$ we construct a continuous,

[2] A notable exception to this generalization is the work of Nadas[Nad78].

invertible differentiable mapping between a neighborhood of R_0 on the manifold and an open set in $\mathbb{R}^3$ as shown in equation (2).

$$R(\omega) = R_0 \exp\{J(\omega)\} \quad \omega \in \mathbb{R}^3, \sqrt{\omega^t\omega} < \pi \tag{2}$$

Where $\omega = (\omega_x, \omega_y, \omega_z)^t$, exp is the matrix exponential operator [Cur79], and $J(\omega)$ is the skew symmetric operator $J : \mathbb{R}^3 \to so(3)$ given by:

$$J(\omega) = \begin{pmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{pmatrix}.$$

Equation (2) provides a one to one mapping between vectors in the open ball $\sqrt{\omega^t\omega} < \pi$ and a local region of the manifold $SO(3)$ centered around the point R_0.

The objective function $\mathcal{O}$ can be expressed in terms of this local parameterization and a quadratic approximation for the objective function around the point R_0 can be constructed as follows:

$$\mathcal{O}(R(\omega)) \approx \mathcal{O}(R_0) + \mathbf{g}^t\omega + \omega^t H\omega \tag{3}$$

$\mathbf{g}$ and H represent the gradient and Hessian of the function respectively evaluated at the point $\omega = 0$ which corresponds to the rotation matrix R_0.

This quadratic approximation can be used to construct an incremental step which leads to a lower point on the error surface.

$$\omega_s = -H^{-1}\mathbf{g} \tag{4}$$

This incremental step can be applied to determine the new rotation matrix as follows: $R = R_0 \exp\{J(\omega_s)\}$. Note that the user must ensure that each minimization step lies within the range of the local parameterization, ie. $\sqrt{\omega_s^t\omega_s} < \pi$.

The updating step can be accomplished in a particularly elegant manner if the rotations are represented by unit quaternions. The rotation R_0 can be represented by the unit quaternion $\mathbf{q}_0$ and the incremental step $\exp\{J(\omega_s)\}$ by the quaternion $\mathbf{q}_s = (\cos(\theta/2), (\sin(\theta/2)/\theta)\,\omega)$ where $\theta = \sqrt{\omega^t\omega}$. The product of these two rotations can be obtained by carrying out a standard quaternion multiplication $\mathbf{q}_0\mathbf{q}_s$ (see the Appendix for a definition of quaternion multiplication).

The entire minimization procedure is outlined below in psuedo-code

```
Choose starting estimate R_0
Loop
        Compute gradient g and Hessian H wrt local parameterization
        If local gradient is small enough
                end
        else
                Compute minimization step :  ω_s ← −H^{-1}g
                Update R_0 :   R_0 ← R_0 exp{J(ω_s)}.
```

Note that the process described above is almost identical to the procedure employed in standard Newtonian minimization algorithms [JS83], the only real difference is that at each iteration of this procedure the quadratic approximation and the minimization step are expressed in terms of a *local* parameter system rather than a *global* one.

The general approach described above is identical to the one proposed recently by Steven Smith [Smi93] in his work on minimization on Riemanninan manifolds. However, this paper deals specifically with the special case of $SO(3)$ and closely related manifolds rather than the more general problem discussed in Smith's thesis. Thus, we are able to take advantage of the special relationship between $SO(3)$ and the group of unit quaternions $Sp(1)$ to construct computationally efficient algorithms for this class of problems.

It can be shown that in a Euclidean parameter space, standard Newtonian minimization algorithms exhibit quadratic convergence when provided with a starting point sufficiently close to a minimum [JS83]. Steven Smith was able to show that the algorithm described above will also exhibit quadratic convergence on Riemannian manifolds like $SO(3)$ [Smi93].

2 An Example

Consider the objective function given in equation (5). The maximum of this objective function corresponds to the rotation matrix that bring the vectors $\mathbf{u}_i$ and $\mathbf{v}_i$ into their best alignment. Such a problem might arise in computer vision during pose estimation from range data where the $\mathbf{u}_i$ are a set of model vectors and the $\mathbf{v}_i$ are measurements.

$$\mathcal{O}(R) = \sum_{i=1}^{n} (\mathbf{u}_i^t R \mathbf{v}_i)^2 \quad \mathbf{u}_i, \mathbf{v}_i \in \mathbb{R}^3; R \in SO(3) \tag{5}$$

The method described in the previous section can be applied to find the critical points of this function with respect to the rotation matrix R. At each stage of this minimization process, the Jacobian and Hessian of the objective function can be calculated as follows.

$$R(\omega) = R_0 \exp\{J(\omega)\} \tag{6}$$

$$\mathcal{O}(\omega) = \sum_{i=1}^{n} (\mathbf{u}_i^t R(\omega) \mathbf{v}_i)^2 \tag{7}$$

$$\frac{\partial}{\partial \omega_x} \mathcal{O}(\omega) = 2 \sum_{i=1}^{n} (\mathbf{u}_i^t R \mathbf{v}_i)(\mathbf{u}_i^t \frac{\partial}{\partial x} R(\omega) \mathbf{v}_i) \tag{8}$$

$$\frac{\partial}{\partial \omega_x \partial \omega_y} \mathcal{O}(\omega) = 2 \sum_{i=1}^{n} \{(\mathbf{u}_i^t \frac{\partial}{\partial \omega_x} R(\omega)\mathbf{v}_i)(\mathbf{u}_i^t \frac{\partial}{\partial \omega_y} R(\omega)\mathbf{v}_i) + (\mathbf{u}_i^t R(\omega)\mathbf{v}_i)(\mathbf{u}_i^t \frac{\partial}{\partial \omega_x \partial \omega_y} R(\omega)\mathbf{v}_i)\} \tag{9}$$

Equations (8) and (9) refer to individual terms in the Jacobian and Hessian respectively. The derivatives of the rotation matrix can be computed quite easily as the following examples indicate.

$$\begin{aligned}
\frac{\partial}{\partial \omega_x} \left(R_0 \exp\{J(\omega)\}\right) \Big|_{\omega=0} &= R_0 \frac{\partial}{\partial \omega_x} \left(\sum_{n=0}^{\infty} \{ \frac{1}{n!} (J(\omega))^n \} \right) \Big|_{\omega=0} \\
&= R_0 J(\hat{x})
\end{aligned}$$

$$\begin{aligned}
\frac{\partial}{\partial \omega_x \partial \omega_y} \left(R_0 \exp\{J(\omega)\}\right) \Big|_{\omega=0} &= \frac{\partial}{\partial \omega_x \partial \omega_y} R_0 \left(\sum_{n=0}^{\infty} \{ \frac{1}{n!} (J(\omega))^n \} \right) \Big|_{\omega=0} \\
&= R_0 \frac{1}{2} (J(\hat{x})J(\hat{y}) + J(\hat{y})J(\hat{x}))
\end{aligned}$$

Notice that these expressions are quite simple to compute, and they do not involve evaluating any transcendental functions as an Euler parameterization would.

3 Further Applications

The following subsections describe how the the local parameterization of $SO(3)$ described in the previous section can be used to construct local parameterizations of several other manifolds commonly used in vision and robotics. As in section 2, the Jacobian and Hessian of these parameterizations can be readily computed, and the optimization procedure described in section 1 can be used to minimize objective functions over these manifolds.

3.1 The Group of Rigid Transformations $SE(3)$

Elements of the group of rigid transformations $SE(3)$ are generally denoted by a tuple $S = \langle R, T \rangle$ where $R \in SO(3)$ and $T \in \mathbb{R}^3$. Given a particular element of this group, we can construct a local parameterization $S : \mathbb{R}^6 \rightarrow SE(3)$ that maps elements from an open ball in $\mathbb{R}^6$ onto a neighborhood around $\langle R_0, T_0 \rangle$ on the manifold as follows:

$$S(\omega, t) = \langle R_0 \exp\{J(\omega)\}, T_0 + t \rangle \quad \omega, t \in \mathbb{R}^3 \tag{10}$$

Where the tuple $\langle \omega, t \rangle$ can be viewed as an element of an open region in $\mathbb{R}^6$.

3.2 The Surface of a Sphere

The parameterization of $SO(3)$ described in the previous section can also be used to parameterize various smooth manifolds in $\mathbb{R}^3$. For example, the surface of the unit sphere in $\mathbb{R}^3$ can be denoted as follows : $S^2 = \{\hat{\mathbf{v}} : \hat{\mathbf{v}} \in \mathbb{R}^3, \hat{\mathbf{v}}^t \hat{\mathbf{v}} = 1\}$. Any element of this set can be expressed in terms of a rotation matrix $R \in SO(3) : \hat{\mathbf{v}} = R\hat{\mathbf{z}},\ \hat{\mathbf{z}} = (0\,0\,1)^t$. [3]

Given a particular element on the manifold, $\hat{\mathbf{v}}_0 = R_0 \hat{\mathbf{z}}$, we can construct a local parameterization $\hat{\mathbf{v}} : \mathbb{R}^2 \rightarrow S^2$ that maps elements from an open region of $\mathbb{R}^2$ onto a neighborhood of $\hat{\mathbf{v}}_0$ on the manifold as follows.

$$\hat{\mathbf{v}}(\omega_x, \omega_y) = R_0 \exp\{J \begin{pmatrix} \omega_x \\ \omega_y \\ 0 \end{pmatrix}\} \hat{\mathbf{z}} \qquad \omega_x^2 + \omega_y^2 < \pi \tag{11}$$

It is trivial to show that the Jacobian of this local parameterization has rank 2 for all points on the manifold. This parameterization can be used to carry out minimization operations over the manifold. The update step at each iteration can be performed using quaternion multiplication as described earlier.

3.3 The Set of Infinite Straight Lines

Every straight line in $\mathbb{R}^3$ can be represented in terms of a unit vector $\hat{\mathbf{v}}$ which indicates the direction of the line, and a vector $\mathbf{d}$ which designates the point on the line that is closest to the origin. In other words, the set of straight lines in $\mathbb{R}^3$ can be represented by the set of tuples $\mathcal{L} = \{\langle \hat{\mathbf{v}}, \mathbf{d} \rangle \in R^3 \times \mathbb{R}^3 : \hat{\mathbf{v}}^t \hat{\mathbf{v}} = 1, \hat{\mathbf{v}}^t \mathbf{d} = 0\}$

This set of tuples defines an algebraic set which can be viewed as a 4 dimensional manifold embedded in $\mathbb{R}^6$. [4]

[3] Note that for a given $\hat{\mathbf{v}}$ there is a one-dimensional subset of $SO(3)$ that will satisfy this equation.

[4] A careful reader will notice that there is actually a two to one correspondence between points on this manifold and the set of infinite straight lines since $\langle \hat{\mathbf{v}}, \mathbf{d} \rangle$ and $\langle -\hat{\mathbf{v}}, \mathbf{d} \rangle$ denote the same line.

Any point on this manifold can be represented in terms of a rotation matrix $R \in SO(3)$ and two scalars $a, b \in \mathbb{R}$ as follows : $\hat{\mathbf{v}} = R\hat{\mathbf{z}}$, $\mathbf{d} = R(a\hat{\mathbf{x}} + b\hat{\mathbf{y}})$. We can construct a local parameterization around any point $\langle \hat{\mathbf{v}}, \mathbf{d} \rangle \in \mathcal{L}$ that maps an open region on $\mathbb{R}^4$ onto a neighborhood of $\langle \hat{\mathbf{v}}, \mathbf{d} \rangle$ on the manifold (i.e. $L : \mathbb{R}^4 \to \mathcal{L}$) as follows.

$$L(\omega_x, \omega_y, \alpha, \beta) = \langle R_0 \exp\{J \begin{pmatrix} \omega_x \\ \omega_y \\ 0 \end{pmatrix}\}\hat{\mathbf{z}}, R_0 \exp\{J \begin{pmatrix} \omega_x \\ \omega_y \\ 0 \end{pmatrix}\}((a+\alpha)\hat{\mathbf{x}} + (b+\beta)\hat{\mathbf{y}})\rangle : \omega_x^2 + \omega_y^2 < \pi \quad (12)$$

Given a particular line represented by $\langle R_0, a, b \rangle$ and a particular incremental step $\langle \omega_x, \omega_y, \alpha, \beta \rangle$ the new line will be represented by $\langle R_0 \exp\{J \begin{pmatrix} \omega_x \\ \omega_y \\ 0 \end{pmatrix}\}, a + \alpha, b + \beta \rangle$

4 Conclusion

This paper describes a novel approach to carrying out numerical optimization procedures over the Lie group $SO(3)$ and related manifolds. This technique exploits the fact that the group of rotations has a natural parameterization based on the exponential operator associated with the Lie group. This approach avoids the inevitable singularities associated with global parameterizations like Euler angles. It should also prove more effective than other techniques that approximate gradient descent on the tangent space to the manifold. It may be possible to apply similar techniques to other Lie groups by taking advantage of their exponential operators.

All of the parameterizations described in this paper have been successfully implemented as part of an actual structure from motion algorithm described in [TK94]. In this application, minimization based on the "standard" global parameterizations failed to converge in some situations, whereas the presented method never encounters the singularity issues that plagued these global parameterizations.

Acknowledgments:

The authors wish to thank Prof. Daniel Koditschek and Prof. Roger Brockett for several useful discussions which helped to crystallize the ideas presented in this monograph.

Appendix : Representations of $SO(3)$

A number of different representations for the orthogonal group $SO(3)$ are used in this paper, this appendix describes how these representations are related to each other. The orthogonal group

$SO(3)$ is defined as a matrix subgroup of the general linear group $GL(3)$.

$$SO(3) = \{R \in \mathbb{R}^{3\times 3} : R^t R = I, det(R) = 1\} \tag{13}$$

Following [Cur79], the exponential map associated with this Lie group provides a surjective map from $\mathbb{R}^3$ to $SO(3)$.

$$R(\omega) = \exp\{J(\omega)\} = \sum_{n=0}^{\infty}\{\frac{1}{n!}(J(\omega))^n\} \tag{14}$$

$J(\omega)$ is the skew symmetric operator $J : \mathbb{R}^3 \to \mathbb{R}^{3\times 3}$ given by:

$$J(\omega) = \begin{pmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{pmatrix}.$$

If the vector ω is rewritten in terms of a unit vector $\hat{\omega}$ and a magnitude θ we can explicitly compute the sum of this infinite series as shown below.

$$R(\omega) = R(\theta\hat{\omega}) = I + \sin\theta J(\hat{\omega}) + (1 - \cos\theta)(J(\hat{\omega}))^2 \tag{15}$$

This equation is usually referred to as the Rodrigues formula; the unit vector $\hat{\omega}$ can be thought of as the axis of rotation while the scalar variable θ denotes the magnitude of the rotation in radians.

The group of unit quaternions $Sp(1)$ is defined as a set of tuples $Sp(1) = \{(u_0, \mathbf{u}) : u_0 \in \mathbb{R}, \mathbf{u} \in \mathbb{R}^3, u_0^2 + \mathbf{u}^t\mathbf{u} = 1\}$ with the following group operation (quaternion multiplication):

$$(u_0, \mathbf{u})(v_0, \mathbf{v}) = (u_0 v_0 - \mathbf{u}^t\mathbf{v}, J(\mathbf{u})\mathbf{v} + u_0\mathbf{v} + v_0\mathbf{u}) \tag{16}$$

Given a quaternion $q = (u_0, \mathbf{u})$ its conjugate $\bar{q}$ is given by $\bar{q} = (u_0, -\mathbf{u})$. It is a straightforward matter to show that $q\bar{q} = \bar{q}q = (1, \mathbf{0})$, where $(1, \mathbf{0})$ is the identity element of the group.

$Sp(1)$ can also be thought of as the unit 3 sphere in $\mathbb{R}^4$, in fact we can define a surjective mapping from $\mathbb{R}^3$ to $Sp(1)$ as follows.

$$q(\omega) = q(\theta\hat{\omega}) = (\cos(\theta/2), \sin(\theta/2)\hat{\omega}) \tag{17}$$

Given the mappings defined in equations (15) and (17), it is possible to show that the following equation must hold for every $\mathbf{v} \in \mathbb{R}^3$.

$$q(\omega)(0, \mathbf{v})\bar{q}(\omega) = (0, R(\omega)\mathbf{v}) \tag{18}$$

This equation suggests a possible connection between SO(3) and Sp(1). In fact, we can use this expression to define a surjective homomorphism $\rho : Sp(1) \rightarrow SO(3)$ that links the two groups.

$$\rho(q) = \rho((u_0, (u_1 u_2 u_3)^t) = \begin{pmatrix} 2(u_0^2 + u_1^2) - 1 & 2(u_1 u_2 - u_0 u_3) & 2(u_1 u_3 + u_0 u_2) \\ 2(u_1 u_2 + u_0 u_3) & 2(u_0^2 + u_2^2) - 1 & 2(u_2 u_3 - u_0 u_1) \\ 2(u_1 u_3 - u_0 u_2) & 2(u_2 u_3 + u_0 u_1) & 2(u_0^2 + u_3^2) - 1 \end{pmatrix} \tag{19}$$

Note that this is *not* an isomorphism since the quaternions $(u_0, \mathbf{u})$ and $(-u_0, -\mathbf{u})$ are mapped onto the same rotation matrix.

Since ρ is a homomorphism we know that it must preserve the group operation, that is $\rho(qv) = \rho(q)\rho(v)\ \forall q, v \in Sp(1)$. This means that we can represent elements of $SO(3)$ by unit quaternions in all of our calculations and replace the matrix multiplication operations with quaternion multiplications as defined in equation (16).

References

[Cur79] Morton Curtis. *Matrix Groups.* Springer-Verlag, 1979.

[GO89] Chaya Gurwitz and Michael L. Overton. A globally convergent algorithm for minimizing over the rotation group of quadratic forms. *IEEE Trans. Pattern Anal. Machine Intell.*, 11(11):1228–1232, November 1989.

[Hor90] Berthold K.P. Horn. Relative orientation. *International Journal of Computer Vision*, 4:59–78, 1990.

[JS83] J.E. Dennis Jr. and Robert B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations.* Prentice-Hall, 1983.

[Nad78] Arthur Nadas. Least squares and maximum likelihood estimation of rigid motion. IBM Research Report RC6945, IBM, Yorktown Heights, 1978.

[Smi93] Steven Smith. *Geometric Optimization Methods for Adaptive Filtering.* PhD thesis, Harvard University, Division of Applied Sciences, Cambridge MA, September 1993.

[Stu64] John Stuelpnagel. On the parameterization of the three-dimensional rotation group. *SIAM Review*, 6(4):422–430, October 1964.

[TK94] Camillo J. Taylor and David J. Kriegman. Structure and motion from line segments in multiple images. Center for Systems Science 9402, Yale University, January 1994.