Geometry Derivatives and Other Hairy Math

Frank Dellaert

February 28, 2010

1 Review of Lie Groups

A Lie group G is a manifold that possesses a smooth group operation. Associated with it is a Lie Algebra $\mathfrak{g}$ which, loosely speaking, can be identified with the tangent space at the identity and completely defines how the groups behaves around the identity. There is a mapping from $\mathfrak{g}$ back to G, called the exponential map

$$\exp : \mathfrak{g} \rightarrow G$$

and a corresponding inverse

$$\log : G \rightarrow \mathfrak{g}$$

that maps elements in G to an element in $\mathfrak{g}$. For n-dimensional matrix Lie groups, the Lie algebra $\mathfrak{g}$ is isomorphic to $\mathbb{R}^n$, and we can define the map

$$\hat{} : \mathbb{R}^n \rightarrow \mathfrak{g}$$

$$\hat{} : x \rightarrow \hat{x}$$

which maps n-vectors $x \in \mathbb{R}^n$ to elements of $\mathfrak{g}$. In the case of matrix Lie groups, the elements $\hat{x}$ of $\mathfrak{g}$ are $n \times n$ matrices.

Below we frequently make use of the equality[1]

$$ge^{\hat{x}}g^{-1} = e^{Ad_g \hat{x}}$$

where $Ad_g : \mathfrak{g} \rightarrow \mathfrak{g}$ is a map parameterized by a group element g. The intuitive explanation is that a change $\exp(\hat{x})$ defined around the orgin, but applied at the group element g, can be written in one step by taking the adjoint $Ad_g \hat{x}$ of $\hat{x}$. In the case of a matrix group the ajoint can be written as [2]

$$Ad_T \hat{x} \stackrel{\Delta}{=} Te^{\hat{x}}T^{-1}$$

and hence we have

$$Te^{\hat{x}}T^{-1} = e^{T\hat{x}T^{-1}}$$

where both T and $\hat{x}$ are $n \times n$ matrices for an n-dimensional Lie group. Below we introduce the most important Lie groups that we deal with.

[1] http://en.wikipedia.org/wiki/Exponential_map

[2] http://en.wikipedia.org/wiki/Adjoint_representation_of_a_Lie_group

2 Derivatives of Mappings

The derivatives for *inverse, compose*, and *between* can be derived from Lie group principles. Specifically, to find the derivative of a function $f(g)$, we want to find the Lie algebra element $\hat{y} \in \mathfrak{g}$, that will result from changing g using $\hat{x}$, also in exponential coordinates:

$$f(g)\,e^{\hat{y}} = f\left(ge^{\hat{x}}\right)$$

Calculating these derivatives requires that we know the form of the function f.

Starting with **inverse**, i.e., $f(g) = g^{-1}$, we have

$$\begin{aligned} g^{-1}e^{\hat{y}} &= \left(ge^{\hat{x}}\right)^{-1} = e^{-\hat{x}}g^{-1} \\ e^{\hat{y}} &= ge^{-\hat{x}}g^{-1} = e^{Ad_g(-\hat{x})} \\ \hat{y} &= Ad_g(-\hat{x}) \end{aligned} \tag{1}$$

In other words, and this is very intuitive in hindsight, the inverse is just negation of $\hat{x}$, along with an adjoint to make sure it is applied in the right frame!

Compose can be derived similarly. Let us define two functions to find the derivatives in first and second arguments:

$$f_1(g) = gh \text{ and } f_2(h) = gh$$

The latter is easiest, as a change $\hat{x}$ in the second argument h simply gets applied to the result gh:

$$\begin{aligned} f_2(h)e^{\hat{y}} &= f_2\left(he^{\hat{x}}\right) \\ ghe^{\hat{y}} &= ghe^{\hat{x}} \\ \hat{y} &= \hat{x} \end{aligned} \tag{2}$$

The derivative for the first argument is a bit trickier:

$$\begin{aligned} f_1(g)e^{\hat{y}} &= f_1\left(ge^{\hat{x}}\right) \\ ghe^{\hat{y}} &= ge^{\hat{x}}h \\ e^{\hat{y}} &= h^{-1}e^{\hat{x}}h = e^{Ad_{h^{-1}}\hat{x}} \\ \hat{y} &= Ad_{h^{-1}}\hat{x} \end{aligned} \tag{3}$$

In other words, to apply a change $\hat{x}$ in g we first need to undo h, then apply $\hat{x}$, and then apply h again. All can be done in one step by simply applying $Ad_{h^{-1}}\hat{x}$.

Finally, let us find the derivative of **between**, defined as $between(g,h) = compose(inverse(g),h)$. The derivative in the second argument h is similarly trivial: $\hat{y} = \hat{x}$. The first argument goes as follows:

$$\begin{aligned} f_1(g)e^{\hat{y}} &= f_1\left(ge^{\hat{x}}\right) \\ g^{-1}he^{\hat{y}} &= \left(ge^{\hat{x}}\right)^{-1}h = e^{(-\hat{x})}g^{-1}h \\ e^{\hat{y}} &= \left(h^{-1}g\right)e^{(-\hat{x})}\left(h^{-1}g\right)^{-1} = e^{Ad_{\left(h^{-1}g\right)}(-\hat{x})} \\ \hat{y} &= Ad_{\left(h^{-1}g\right)}(-\hat{x}) = Ad_{between(h,g)}(-\hat{x}) \end{aligned} \tag{4}$$

Hence, now we undo h and then apply the inverse $(-\hat{x})$ in the g frame.

3 Important Lie Groups

3.1 3D Rotations

The Lie group $SO(3)$ is a subgroup of the general linear group $GL(3)$ of 3×3 invertible matrices. Its Lie algebra $\mathfrak{so}(3)$ is the vector space of 3×3 skew-symmetric matrices. The exponential map can be computed in closed form using Rodrigues' formula.

Since $SO(3)$ is a three-dimensional manifold, $\mathfrak{so}(3)$ is isomorphic to $\mathbb{R}^3$ and we define the map

$$\hat{}: \mathbb{R}^3 \to \mathfrak{so}(3)$$

$$\hat{}: \omega \to \hat{\omega} = [\omega]_\times$$

which maps 3-vectors ω to skew-symmetric matrices $[\omega]_\times$:

$$[\omega]_\times = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}$$

For every $3-$vector ω there is a corresponding rotation matrix

$$R = e^{[\omega]_\times}$$

and this is defines the canonical parameterization of $SO(3)$, with ω known as the canonical or exponential coordinates. It is equivalent to the axis-angle representation for rotations, where the unit vector $\omega / \|\omega\|$ defines the rotation axis, and its magnitude the amount of rotation θ.

We can prove the following identity for rotation matrices R,

$$\begin{aligned} R[\omega]_\times R^T &= R[\omega]_\times \begin{bmatrix} a_1 & a_2 & a_3 \end{bmatrix} \\ &= R \begin{bmatrix} \omega \times a_1 & \omega \times a_2 & \omega \times a_3 \end{bmatrix} \\ &= \begin{bmatrix} a_1(\omega \times a_1) & a_1(\omega \times a_2) & a_1(\omega \times a_3) \\ a_2(\omega \times a_1) & a_2(\omega \times a_2) & a_2(\omega \times a_3) \\ a_3(\omega \times a_1) & a_3(\omega \times a_2) & a_3(\omega \times a_3) \end{bmatrix} \\ &= \begin{bmatrix} \omega(a_1 \times a_1) & \omega(a_2 \times a_1) & \omega(a_3 \times a_1) \\ \omega(a_1 \times a_2) & \omega(a_2 \times a_2) & \omega(a_3 \times a_2) \\ \omega(a_1 \times a_3) & \omega(a_2 \times a_3) & \omega(a_3 \times a_3) \end{bmatrix} \\ &= \begin{bmatrix} 0 & -\omega a_3 & \omega a_2 \\ \omega a_3 & 0 & -\omega a_1 \\ -\omega a_2 & \omega a_1 & 0 \end{bmatrix} \\ &= [R\omega]_\times \end{aligned} \tag{5}$$

where a_1, a_2, and a_3 are the *rows* of R. Above we made use of the orthogonality of rotation matrices and the triple product rule:

$$a(b \times c) = b(c \times a) = c(a \times b)$$

Hence, given property (5), the adjoint map for $\mathfrak{so}(3)$ simplifies to

$$Ad_R[\omega]_\times = R[\omega]_\times R^T = [R\omega]_\times$$

and this can be expressed in exponential coordinates simply by rotating the axis ω to $R\omega$.

As an example, to apply an axis-angle rotation ω to a point p in the frame R, we could:

1. First transform p back to the world frame, apply ω, and then rotate back:

$$q = Re^{[\omega]_\times} R^T$$

2. Immediately apply the transformed axis-angle transformation $Ad_R[\omega]_\times = [R\omega]_\times$:

$$q = e^{[R\omega]_\times} p$$

Hence, we are now in a position to simply posit the derivative of **inverse**,

$$\begin{aligned} [\omega']_\times &= Ad_R\left([-\omega]_\times\right) = [R(-\omega)]_\times \\ \frac{\partial R^T}{\partial \omega} &= -R \end{aligned}$$

compose in its first argument,

$$\begin{aligned} [\omega']_\times &= Ad_{R_2^T}\left([\omega]_\times\right) = [R_2^T \omega]_\times \\ \frac{\partial \left(R_1 R_2\right)}{\partial \omega_1} &= R_2^T \end{aligned}$$

compose in its second argument,

$$\frac{\partial \left(R_1 R_2\right)}{\partial \omega_2} = I_3$$

between in its first argument,

$$\begin{aligned} [\omega']_\times &= Ad_{R_2^T R_1}\left([-\omega]_\times\right) = [R_2^T R_1(-\omega)]_\times \\ \frac{\partial \left(R_1^T R_2\right)}{\partial \omega_1} &= -R_2^T R_1 = -between(R_2, R_1) \end{aligned}$$

and between in its second argument,

$$\frac{\partial \left(R_1^T R_2\right)}{\partial \omega_2} = I_3$$

3.2 3D Rigid Transformations

The Lie group $SE(3)$ is a subgroup of the general linear group $GL(4)$ of 4×4 invertible matrices of the form

$$T \stackrel{\Delta}{=} \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix}$$

where $R \in SO(3)$ is a rotation matrix and $t \in \mathbb{R}^3$ is a translation vector. Its Lie algebra $\mathfrak{se}(3)$ is the vector space of 4×4 twists $\hat{\xi}$ parameterized by the *twist coordinates* $\xi \in \mathbb{R}^6$, with the mapping [1]

$$\xi \stackrel{\Delta}{=} \begin{bmatrix} \omega \\ v \end{bmatrix} \rightarrow \hat{\xi} \stackrel{\Delta}{=} \begin{bmatrix} [\omega]_\times & v \\ 0 & 0 \end{bmatrix}$$

Note we follow Frank Park's convention and reserve the first three components for rotation, and the last three for translation. Applying the exponential map to a twist ξ yields a screw motion yielding an element in $SE(3)$:

$$T = \exp \hat{\xi}$$

A closed form solution for the exponential map is given in [1, page 42].

The adjoint is

$$\begin{aligned}
Ad_T \hat{\xi} &= T \hat{\xi} T^{-1} \\
&= \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} [\omega]_\times & v \\ 0 & 0 \end{bmatrix} \begin{bmatrix} R^T & -R^T t \\ 0 & 1 \end{bmatrix} \\
&= \begin{bmatrix} [R\omega]_\times & -[R\omega]_\times t + Rv \\ 0 & 0 \end{bmatrix} \\
&= \begin{bmatrix} [R\omega]_\times & t \times R\omega + Rv \\ 0 & 0 \end{bmatrix}
\end{aligned}$$

From this we can express the Adjoint map in terms of twist coordinates (see also [1] and FP):

$$\begin{bmatrix} \omega' \\ v' \end{bmatrix} = \begin{bmatrix} R & 0 \\ [t]_\times R & R \end{bmatrix} \begin{bmatrix} \omega \\ v \end{bmatrix}$$

Hence, as with $SO(3)$, we are now in a position to simply posit the derivative of **inverse**,

$$\frac{\partial T^{-1}}{\partial \xi} = - \begin{bmatrix} R & 0 \\ [t]_\times R & R \end{bmatrix}$$

(but unit test on the above fails !!!), **compose** in its first argument,

$$\frac{\partial (T_1 T_2)}{\partial \xi_1} = \begin{bmatrix} R_2^T & 0 \\ [-R_2^T t]_\times R_2^T & R_2^T \end{bmatrix}$$

compose in its second argument,

$$\frac{\partial (T_1 T_2)}{\partial \xi_2} = I_6$$

between in its first argument,

$$\frac{\partial \left(T_1^{-1} T_2 \right)}{\partial \xi_1} = - \begin{bmatrix} R & 0 \\ [t]_\times R & R \end{bmatrix}$$

with

$$\begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} = T_1^{-1} T_2 = between(T_2, T_1)$$

and between in its second argument,

$$\frac{\partial \left(T_1^{-1} T_2 \right)}{\partial \xi_1} = I_6$$

3.3 2D Rotations

The Lie group $SO(2)$ is a subgroup of the general linear group $GL(2)$ of 2×2 invertible matrices. Its Lie algebra $\mathfrak{so}(2)$ is the vector space of 2×2 skew-symmetric matrices. Though simpler than $SO(3)$ it is *commutative* and hence things simplify in ways that do not generalize well, so we treat it only now. Since $SO(2)$ is a one-dimensional manifold, $\mathfrak{so}(2)$ is isomorphic to $\mathbb{R}$ and we define

$$\hat{}: \mathbb{R} \to \mathfrak{so}(2)$$

$$\hat{}: \theta \to \hat{\theta} = [\theta]_+$$

which maps the angle θ to the 2×2 skew-symmetric matrix $[\theta]_+$:

$$[\theta]_+ = \begin{bmatrix} 0 & -\theta \\ \theta & 0 \end{bmatrix}$$

Note that

$$[\theta]_+ \begin{bmatrix} x \\ y \end{bmatrix} = \theta R_{\pi/2} \begin{bmatrix} x \\ y \end{bmatrix} = \theta \begin{bmatrix} -y \\ x \end{bmatrix} \tag{6}$$

which acts like a restricted "cross product" in the plane.

The exponential map can be computed in closed form as

$$R = e^{[\theta]_+} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

The adjoint map for $\mathfrak{so}(2)$ is trivially equal to the identity, as is the case for *all* commutative groups, and we have the derivative of **inverse**,

$$\frac{\partial R^T}{\partial \theta} = -Ad_R = -1$$

compose in its first argument,

$$\frac{\partial (R_1 R_2)}{\partial \theta_1} = Ad_{R_2^T} = 1$$

compose in its second argument,

$$\frac{\partial (R_1 R_2)}{\partial \theta_2} = 1$$

between in its first argument,

$$\frac{\partial \left(R_1^T R_2\right)}{\partial \theta_1} = -Ad_{R_2^T R_1} = -1$$

and between in its second argument,

$$\frac{\partial \left(R_1^T R_2\right)}{\partial \theta_2} = 1$$

3.4 2D Rigid Transformations

The Lie group $SE(2)$ is a subgroup of the general linear group $GL(3)$ of 3×3 invertible matrices of the form

$$T \stackrel{\Delta}{=} \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix}$$

where $R \in SO(2)$ is a rotation matrix and $t \in \mathbb{R}^2$ is a translation vector. Its Lie algebra $\mathfrak{se}(2)$ is the vector space of 3×3 twists $\hat{\xi}$ parameterized by the *twist coordinates* $\xi \in \mathbb{R}^3$, with the mapping

$$\xi \stackrel{\Delta}{=} \begin{bmatrix} v \\ \omega \end{bmatrix} \rightarrow \hat{\xi} \stackrel{\Delta}{=} \begin{bmatrix} [\omega]_+ & v \\ 0 & 0 \end{bmatrix}$$

Note we think of robots as having a pose (x, y, θ) and hence I switched the order above, reserving the first two components for translation and the last for rotation. Applying the exponential map to a twist ξ yields a screw motion yielding an element in $SE(2)$:

$$T = \exp \hat{\xi}$$

A closed form solution for the exponential map is in the works...

The adjoint is

$$\begin{aligned} Ad_T \hat{\xi} &= T \hat{\xi} T^{-1} \\ &= \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} [\omega]_+ & v \\ 0 & 0 \end{bmatrix} \begin{bmatrix} R^T & -R^T t \\ 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} [\omega]_+ & -[\omega]_+ t + Rv \\ 0 & 0 \end{bmatrix} \\ &= \begin{bmatrix} [\omega]_+ & Rv - \omega R_{\pi/2} t \\ 0 & 0 \end{bmatrix} \end{aligned}$$

From this we can express the Adjoint map in terms of plane twist coordinates:

$$\begin{bmatrix} v' \\ \omega' \end{bmatrix} = \begin{bmatrix} R & -R_{\pi/2} t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}$$

We can just define all derivatives in terms of the above adjoint map:

$$\frac{\partial T^{-1}}{\partial \xi} = -Ad_T$$

$$\frac{\partial (T_1 T_2)}{\partial \xi_1} = Ad_{T_2^{-1}} = 1 \text{ and } \frac{\partial (T_1 T_2)}{\partial \xi_2} = I_3$$

$$\frac{\partial \left(T_1^{-1} T_2\right)}{\partial \xi_1} = -Ad_{T_2^{-1} T_1} = -Ad_{between(T_2, T_1)} \text{ and } \frac{\partial \left(T_1^{-1} T_2\right)}{\partial \xi_2} = I_3$$

Part I
Old Stuff

4 Rot2 (in gtsam)

A rotation is stored as $(\cos\theta, \sin\theta)$. An incremental rotation is applied using the trigonometric sum rule:

$$\cos\theta' = \cos\theta\cos\delta - \sin\theta\sin\delta$$
$$\sin\theta' = \sin\theta\cos\delta + \cos\theta\sin\delta$$

where δ is an incremental rotation angle. The derivatives of ***rotate*** are then found easily, using

$$\begin{aligned}
\frac{\partial x'}{\partial\delta} &= \frac{\partial(x\cos\theta' - y\sin\theta')}{\partial\delta} \\
&= \frac{\partial(x(\cos\theta\cos\delta - \sin\theta\sin\delta) - y(\sin\theta\cos\delta + \cos\theta\sin\delta))}{\partial\delta} \\
&= x(-\cos\theta\sin\delta - \sin\theta\cos\delta) - y(-\sin\theta\sin\delta + \cos\theta\cos\delta) \\
&= -x\sin\theta - y\cos\theta = -y'
\end{aligned}$$

$$\begin{aligned}
\frac{\partial y'}{\partial\delta} &= \frac{\partial(x\sin\theta' + y\cos\theta')}{\partial\delta} \\
&= \frac{\partial(x(\sin\theta\cos\delta + \cos\theta\sin\delta) + y(\cos\theta\cos\delta - \sin\theta\sin\delta))}{\partial\delta} \\
&= x(-\sin\theta\sin\delta + \cos\theta\cos\delta) + y(-\cos\theta\sin\delta - \sin\theta\cos\delta) \\
&= x\cos\theta - y\sin\theta = x'
\end{aligned}$$

$$\frac{\partial p'}{\partial p} = \frac{\partial(Rp)}{\partial p} = R$$

Similarly, unrotate

$$\begin{aligned}
\frac{\partial x'}{\partial\delta} &= \frac{\partial(x\cos\theta' + y\sin\theta')}{\partial\delta} \\
&= \frac{\partial(x(\cos\theta\cos\delta - \sin\theta\sin\delta) + y(\sin\theta\cos\delta + \cos\theta\sin\delta))}{\partial\delta} \\
&= x(-\cos\theta\sin\delta - \sin\theta\cos\delta) + y(-\sin\theta\sin\delta + \cos\theta\cos\delta) \\
&= -x\sin\theta + y\cos\theta = y'
\end{aligned}$$

$$
\begin{aligned}
\frac{\partial y'}{\partial \delta} &= \frac{\partial(-x\sin\theta' + y\cos\theta')}{\partial \delta} \\
&= \frac{\partial(-x(\sin\theta\cos\delta + \cos\theta\sin\delta) + y(\cos\theta\cos\delta - \sin\theta\sin\delta))}{\partial \delta} \\
&= -x(-\sin\theta\sin\delta + \cos\theta\cos\delta) + y(-\cos\theta\sin\delta - \sin\theta\cos\delta) \\
&= -x\cos\theta - y\sin\theta = -x'
\end{aligned}
$$

$$
\frac{\partial p'}{\partial p} = \frac{\partial(Rp)}{\partial p} = R
$$

5 Point3

A cross product $a \times b$ can be written as a matrix multiplication

$$
a \times b = [a]_\times b
$$

where $[a]_\times$ is a skew-symmetric matrix defined as

$$
[x, y, z]_\times = \begin{bmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{bmatrix}
$$

We also have

$$
a^T [b]_\times = -([b]_\times a)^T = -(a \times b)^T
$$

The derivative of a cross product

$$
\frac{\partial(a \times b)}{\partial a} = [-b]_\times \tag{7}
$$

$$
\frac{\partial(a \times b)}{\partial b} = [a]_\times \tag{8}
$$

6 Rot3

An incremental rotation is applied as (switched to right-multiply Jan 25 2010)

$$
R' = R(I + \Omega)
$$

where $\Omega = [\omega]_\times$ is the skew symmetric matrix corresponding to the incremental rotation angles $\omega = (\omega_x, \omega_y, \omega_z)$. The derivatives of ***rotate*** are then found easily, using (7):

$$
\frac{\partial(R(I+\Omega)x)}{\partial \omega} = \frac{\partial(R\Omega x)}{\partial \omega} = \frac{\partial(R(\omega \times x))}{\partial \omega} = R\frac{\partial(\omega \times x)}{\partial \omega} = R[-x]_\times
$$

$$
\frac{\partial(Rx)}{\partial x} = R
$$

For composition and transposing of rotation matrices the situation is a bit more complex. We want to figure out what incremental rotation Ω' on the composed matrix, will yield the same change as Ω applied to either the first (A) or second argument (B). Hence, the derivative with respect to the second argument is now easy:

$$\begin{aligned} (AB)(I+\Omega') &= A\left[B(I+\Omega)\right] \\ AB+AB\Omega' &= AB+AB\Omega \\ \Omega' &= \Omega \\ \omega' &= \omega \end{aligned}$$

i.e. the derivative is the identity matrix.

For compose, $R=AB$, we make use of equations (4) and (2). The second argument is easy: we always have

$$\hat{y}=\hat{x}$$

and hence the 3×3 Jacobian is the identity.

Equations (4) then becomes

$$\hat{y}=B^T\hat{x}B=\widehat{B^Tx}$$

The derivative of ***inverse*** can be found similarly, using formula (1):

$$\hat{y}=-R\hat{x}R^T=-\widehat{Rx}$$

And between becomes:

$$\hat{y}=-B^TA\hat{x}A^TB=-\widehat{B^TA}$$

$$\frac{\partial between(A,B)}{\partial a}=-B^TA=-between(B,A)$$

and of course

$$\frac{\partial between(A,B)}{\partial b}=I_3$$

Similarly, the derivative of $unrotate(R,x)=rotate(inverse(R),x)$, so

$$\frac{\partial r(i(R),x)}{\partial\omega}=dr1(i(R),x)di(R)=R^T[x]_\times R=[R^Tx]_\times$$

$$\frac{\partial r(i(R),x)}{\partial x}=i(R)=R^T$$

7 Pose3 (gtsam, old-style exmap)

In the old-style, we have

$R'=R(I+\Omega)$

$t'=t+dt$

In this case, the derivative of ***transform_from***, $Rx+t$:

$$\frac{\partial(R(I+\Omega)x+t)}{\partial\omega}=\frac{\partial(R\Omega x)}{\partial\omega}=\frac{\partial(R(\omega\times x))}{\partial\omega}=R[-x]_\times$$

and with respect to dt is easy:

$$\frac{\partial(Rx+t+dt)}{\partial dt}=I$$

The derivative of ***transform_to***, $inv(R)(x-t)$ we can obtain using the chain rule:

$$\frac{\partial(inv(R)(x-t))}{\partial\omega}=\frac{\partial unrot(R,(x-t))}{\partial\omega}=skew(R^T\,(x-t))$$

and with respect to dt is easy:

$$\frac{\partial(R^T(x-t-dt))}{\partial dt}=-R^T$$

The derivative of $inverse=R^T,-R^Tt=R^T(I,-t)$, first derivative of rotation in rotation argument:

The partials

$$\frac{\partial\omega'}{\partial\omega}=\frac{\partial inv(R)}{\partial\omega}=-R$$

$$\frac{\partial t'}{\partial\omega}=\frac{-\partial unrot(R,t)}{\partial\omega}=-skew(R^Tt)$$

$$\frac{\partial\omega'}{\partial t}=\mathbf{0}$$

$$\frac{\partial t'}{\partial t}=\frac{-\partial unrot(R,t)}{\partial t}=-R^T$$

old stuff:

$$\begin{aligned}(I+\Omega')R^T &= ((I+\Omega)R)^T\\ R^T+\Omega'R^T &= R^T(I-\Omega)\\ \Omega'R^T &= -R^T\Omega\\ \Omega' &= -R^T\Omega R=-[R^T\omega]_\times\\ \omega' &= -R^T\omega\end{aligned}$$

Now ***compose***, first w.r.t. a change in rotation in the first argument:

$$\begin{aligned}
AB &= (T_A R_A T_B)(R_A R_B) \\
\left(T_A R_A T_B \left(I + T'\right)\right)\left(R_A R_B \left(I + \Omega'\right)\right) &= \left(T_A R_A \left(I + \Omega\right) T_B\right)\left(R_A \left(I + \Omega\right) R_B\right) \\
\text{translation only:} & \\
T_A R_A T_B \left(I + T'\right) &= T_A R_A \left(I + \Omega\right) T_B \\
T_B \left(I + T'\right) &= \left(I + \Omega\right) T_B \\
T_B + T_B T' &= T_B + \Omega T_B \\
T' &= T_B^{-1} skew(\omega) T_B \\
T' &= skew(T_B \omega) \, ??? \\
\text{rotation only:} & \\
R_A R_B \left(I + \Omega'\right) &= R_A \left(I + \Omega\right) R_B \\
R_B \Omega' &= \Omega R_B \\
\Omega' &= R_B^T \Omega R_B \\
&= skew(R_B^T \omega) \\
\omega' &= R_B^T \omega
\end{aligned}$$

And w.r.t. a rotation in the second argument:

$$\begin{aligned}
\left(T_A R_A T_B \left(I + T'\right)\right)\left(R_A R_B \left(I + \Omega'\right)\right) &= \left(T_A R_A T_B\right)\left(R_A R_B \left(I + \Omega\right)\right) \\
\left(R_A R_B \left(I + \Omega'\right)\right) &= \left(R_A R_B \left(I + \Omega\right)\right) \\
\omega' &= \omega \\
t' &= 0
\end{aligned}$$

w.r.t. a translation in the second argument:

$$\begin{aligned}
\left(T_A R_A T_B \left(I + T'\right)\right)\left(R_A R_B \left(I + \Omega'\right)\right) &= \left(T_A R_A T_B \left(I + T\right)\right)\left(R_A R_B\right) \\
\omega' &= 0 \\
t' &= t
\end{aligned}$$

Finally, ***between*** in the first argument:

$$\begin{aligned}
\frac{\partial A^{-1}B}{\partial A} &= \frac{\partial c\left(A^{-1}, B\right)}{\partial A^{-1}} \frac{\partial inv(A)}{A} \\
\frac{\partial A^{-1}B}{B} &= \frac{\partial c\left(A^{-1}, B\right)}{\partial B}
\end{aligned}$$

8 Pose3 (gtsam, new-style exmap)

In the new-style exponential map, Pose3 is composed with a delta pose as follows
$R' = (I + \Omega) R$

$t' = (I+\Omega)t + dt$

The derivative of transform_from, $Rx+t$:

$$\frac{\partial((I+\Omega)Rx+(I+\Omega)t)}{\partial\omega} = \frac{\partial(\Omega(Rx+t))}{\partial\omega} = \frac{\partial(\omega\times(Rx+t))}{\partial\omega} = -[Rx+t]_\times$$

and with respect to dt is easy:

$$\frac{\partial(Rx+t+dt)}{\partial dt} = I$$

The derivative of transform_to, $R^T(x-t)$, eludes me. The calculation below is just an attempt:

Noting that $R'^T = R^T(I-\Omega)$, and $(I-\Omega)(x-(I+\Omega)t) = (I-\Omega)(x-t-\Omega t) = x-t-dt-\Omega x+\Omega^2 t$

$$\frac{\partial(R'^T(x-t'))}{\partial\omega} = \frac{\partial(R^T(I-\Omega)(x-(I+\Omega)t))}{\partial\omega} = -\frac{\partial(R^T(\Omega(x-\Omega t)))}{\partial\omega}$$

$$-\frac{\partial([R^T\omega]_\times R^T x)}{\partial\omega} = [R^T x]_\times \frac{\partial(R^T\omega)}{\partial\omega} = [R^T x]_\times R^T$$

$$= \frac{\partial(R^T\Omega^2 t)}{\partial\omega} + [R^T x]_\times R^T$$

and with respect to dt is easy:

$$\frac{\partial(R^T(x-t-dt))}{\partial dt} = -R^T$$

9 Line3vd

One representation of a line is through 2 vectors (v,d), where v is the direction and the vector d points from the orgin to the closest point on the line.

In this representation, transforming a 3D line from a world coordinate frame to a camera at (R^c_w, t^w) is done by

$$v^c = R^c_w v^w$$
$$d^c = R^c_w(d^w + (t^w v^w)v^w - t^w)$$

10 Line3

For 3D lines, we use a parameterization due to C.J. Taylor, using a rotation matrix R and 2 scalars a and b. The line direction v is simply the Z-axis of the rotated frame, i.e., $v = R_3$, while the vector d is given by $d = aR_1 + bR_2$.

Now, we will *not* use the incremental rotation scheme we used for rotations: because the matrix R translates from the line coordinate frame to the world frame, we need to apply the incremental rotation on the right-side:

$$R' = R(I+\Omega)$$

Projecting a line to 2D can be done easily, as both v and d are also the 2D homogenous coordinates of two points on the projected line, and hence we have

$$\begin{aligned} l &= v \times d \\ &= R_3 \times (aR_1 + bR_2) \\ &= a(R_3 \times R_1) + b(R_3 \times R_2) \\ &= aR_2 - bR_1 \end{aligned}$$

This can be written as a rotation of a point,

$$l = R \begin{pmatrix} -b \\ a \\ 0 \end{pmatrix}$$

but because the incremental rotation is now done on the right, we need to figure out the derivatives again:

$$\frac{\partial(R(I+\Omega)x)}{\partial\omega} = \frac{\partial(R\Omega x)}{\partial\omega} = R\frac{\partial(\Omega x)}{\partial\omega} = R[-x]_\times \tag{9}$$

and hence the derivative of the projection l with respect to the rotation matrix $R$of the 3D line is

$$\frac{\partial(l)}{\partial\omega} = R[\begin{pmatrix} b \\ -a \\ 0 \end{pmatrix}]_\times = \begin{bmatrix} aR_3 & bR_3 & -(aR_1 + bR_2) \end{bmatrix} \tag{10}$$

or the a, b scalars:

$$\frac{\partial(l)}{\partial a} = R_2$$
$$\frac{\partial(l)}{\partial b} = -R_1$$

Transforming a 3D line $(R, (a, b))$ from a world coordinate frame to a camera frame (R^c_w, t^w) is done by

$$R' = R^c_w R$$
$$a' = a - R_1^T t^w$$
$$b' = b - R_2^T t^w$$

Again, we need to redo the derivatives, as R is incremented from the right. The first argument is incremented from the left, but the result is incremented on the right:

$$\begin{aligned} R'(I+\Omega') = (AB)(I+\Omega') &= (I + [S\omega]_\times)AB \\ I + \Omega' &= (AB)^T(I + [S\omega]_\times)(AB) \\ \Omega' &= R'^T[S\omega]_\times R' \\ \Omega' &= [R'^T S\omega]_\times \\ \omega' &= R'^T S\omega \end{aligned}$$

For the second argument R we now simply have:

$$\begin{array}{rcl} AB(I+\Omega') & = & AB(I+\Omega) \\ \Omega' & = & \Omega \\ \omega' & = & \omega \end{array}$$

The scalar derivatives can be found by realizing that

$$\begin{pmatrix} a' \\ b' \\ \ldots \end{pmatrix} = \begin{pmatrix} a \\ b \\ 0 \end{pmatrix} - R^T t^w$$

where we don't care about the third row. Hence

$$\frac{\partial((R(I+\Omega_2))^T t^w)}{\partial\omega} = -\frac{\partial(\Omega_2 R^T t^w)}{\partial\omega} = -[R^T t^w]_\times = \begin{bmatrix} 0 & R_3^T t^w & -R_2^T t^w \\ -R_3^T t^w & 0 & R_1^T t^w \\ \ldots & \ldots & 0 \end{bmatrix}$$

11 2D Line Segments

The error between an infinite line (a,b,c) and a 2D line segment $((x1,y1),(x2,y2))$ is defined in Line3.ml.

12 Recovering Pose

Below is the explanaition underlying Pose3.align, i.e. aligning two point clouds using SVD. Inspired but modified from CVOnline...

Our model is

$$p^c = R(p^w - t)$$

i.e., R is from camera to world, and t is the camera location in world coordinates. The objective function is

$$\frac{1}{2}\sum(p^c - R(p^w - t))^2 = \frac{1}{2}\sum(p^c - Rp^w + Rt)^2 = \frac{1}{2}\sum\left(p^c - Rp^w - t'\right)^2 \tag{11}$$

where $t' = -Rt$ is the location of the origin in the camera frame. Taking the derivative with respect to t' and setting to zero we have

$$\sum\left(p^c - Rp^w - t'\right) = 0$$

or

$$t' = \frac{1}{n}\sum(p^c - Rp^w) = \bar{p}^c - R\bar{p}^w \tag{12}$$

here $\bar{p}^c$ and $\bar{p}^w$ are the point cloud centroids. Substituting back into (11), we get

$$\frac{1}{2}\sum(p^c - R(p^w - t))^2 = \frac{1}{2}\sum((p^c - \bar{p}^c) - R(p^w - \bar{p}^w))^2 = \frac{1}{2}\sum(\hat{p}^c - R\hat{p}^w)^2$$

Now, to minimize the above it suffices to maximize (see CVOnline)

$$trace\left(R^T C\right)$$

where $C = \sum \hat{p}^c \left(\hat{p}^w\right)^T$ is the correlation matrix. Intuitively, the cloud of points is rotated to align with the principal axes. This can be achieved by SVD decomposition on C

$$C = USV^T$$

and setting

$$R = UV^T$$

Clearly, from (12) we then also recover the optimal t as

$$t = \bar{p}^w - R^T \bar{p}^c$$

References

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