Geometry Derivatives and Other Hairy Math

Frank Dellaert

March 4, 2010

1 Derivatives of Lie Group 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:

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

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

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

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

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:

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

The derivative for the first argument is a bit trickier:

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

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.

Numerical Derivatives

Let's examine

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

and multiply with $f(g)^{-1}$ on both sides then take the log (which in our case returns y, not $\hat{y}$):

$$y(x) = \log\left[f(g)^{-1}f\left(ge^{\hat{x}}\right)\right]$$

Let us look at $x = 0$, and perturb in direction i, $e_i = [0,0,d,0,0]$. Then take derivative,

$$\frac{\partial y(d)}{\partial d} \stackrel{\Delta}{=} \lim_{d->0}\frac{y(d)-y(0)}{d} = \lim_{d->0}\frac{1}{d}\log\left[f(g)^{-1}f\left(ge^{\hat{e}_i}\right)\right]$$

which is the basis for a numerical derivative scheme.

Let us also look at a chain rule. If we know the behavior at the origin I, we can extrapolate

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

2 Derivatives of Actions

2.1 Forward Action

The (usual) action of an n-dimensional matrix group G is matrix-vector multiplication on $\mathbb{R}^n$,

$$q = Tp$$

with $p,q \in \mathbb{R}^n$ and $T \in GL(n)$. Let us first do away with the derivative in p, which is easy:

$$\frac{\partial(Tp)}{\partial p} = T$$

We would now like to know what an incremental action $\hat{x}$ would do, through the exponential map

$$q(x) = Te^{\hat{x}}p$$

with derivative

$$\frac{\partial q(x)}{\partial x} = T\frac{\partial}{\partial x}\left(e^{\hat{x}}p\right)$$

Since the matrix exponential is given by the series

$$e^{A} = I + A + \frac{A^2}{2!} + \frac{A^3}{3!} + \dots$$

we have, to first order

$$e^{\hat{x}}p = p + \hat{x}p + \dots$$

and the derivative of an incremental action x for matrix Lie groups becomes

$$\frac{\partial q(x)}{\partial x} = T\frac{\partial\left(\hat{x}p\right)}{\partial x} \stackrel{\Delta}{=} TH_p$$

where H_p is an $n \times n$ Jacobian matrix that depends on p.

2.2 Inverse Action

When we apply the inverse transformation

$$q = T^{-1}p$$

we would now like to know what an incremental action $\hat{x}$ on T would do:

$$\begin{aligned} q(x) &= \left(Te^{\hat{x}}\right)^{-1}p \\ &= e^{-\hat{x}}T^{-1}p \\ &= T^{-1}Te^{-\hat{x}}T^{-1}p \\ &= -T^{-1}\exp\left(T\hat{x}T^{-1}\right)p \end{aligned}$$

Hence

$$\frac{\partial q(x)}{\partial x} = -T^{-1}\frac{\partial\left(T\hat{x}T^{-1}\ p\right)}{\partial x} \tag{5}$$

The derivative in p is again easy for matrix Lie groups:

$$\frac{\partial\left(T^{-1}p\right)}{\partial p} = T^{-1}$$

3 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

$$\frac{\partial(a \times b)}{\partial a} = [-b]_\times \tag{6}$$

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

4 2D Rotations

4.1 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.

4.2 Derivatives of Mappings

We have the derivative of **inverse**,

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

compose,

$$\frac{\partial(R_1R_2)}{\partial\theta_1} = Ad_{R_2^T} = 1 \text{ and } \frac{\partial(R_1R_2)}{\partial\theta_2} = 1$$

*and***between:**

$$\frac{\partial\left(R_1^TR_2\right)}{\partial\theta_1} = -Ad_{R_2^TR_1} = -1 \text{ and } \frac{\partial\left(R_1^TR_2\right)}{\partial\theta_2} = 1$$

4.3 Derivatives of Actions

In the case of $SO(2)$ the vector space is $\mathbb{R}^2$, and the group action corresponds to rotating a point

$$q = Rp$$

We would now like to know what an incremental rotation parameterized by θ would do:

$$q(\theta) = Re^{[\theta]_+}p$$

hence the derivative (following the exposition in Section 2):

$$\frac{\partial q(\omega)}{\partial\omega} = R\frac{\partial}{\partial\omega}\left(e^{[\theta]_+}p\right) = R\frac{\partial}{\partial\omega}\left([\theta]_+p\right) = RH_p$$

Note that

$$[\theta]_+\begin{bmatrix} x \\ y \end{bmatrix} = \theta R_{\pi/2}\begin{bmatrix} x \\ y \end{bmatrix} = \theta\begin{bmatrix} -y \\ x \end{bmatrix} \tag{8}$$

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

$$[\theta]_+p = \begin{bmatrix} -y \\ x \end{bmatrix}\theta = H_p\theta$$

with $H_p = R_{pi/2}p$. Hence, the final derivative of an action in its first argument is

$$\frac{\partial q(\theta)}{\partial\theta} = RH_p = RR_{pi/2}p = R_{pi/2}Rp = R_{pi/2}q$$

5 2D Rigid Transformations

5.1 Derivatives of Mappings

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

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

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

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

5.2 The derivatives of Actions

The action of $SE(2)$ on 2D points is done by embedding the points in $\mathbb{R}^3$ by using homogeneous coordinates

$$\hat{q} = \begin{bmatrix} q \\ 1 \end{bmatrix} = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} p \\ 1 \end{bmatrix} = T\hat{p}$$

Analoguous to $SE(3)$, we can compute a velocity $\hat{\xi}\hat{p}$ in the local T frame:

$$\hat{\xi}\hat{p} = \begin{bmatrix} [\omega]_+ & v \\ 0 & 0 \end{bmatrix} \begin{bmatrix} p \\ 1 \end{bmatrix} = \begin{bmatrix} [\omega]_+ p + v \\ 0 \end{bmatrix}$$

By only taking the top two rows, we can write this as a velocity in $\mathbb{R}^2$, as the product of a 2×3 matrix H_p that acts upon the exponential coordinates ξ directly:

$$[\omega]_+ p + v = v + R_{\pi/2} p \omega = \begin{bmatrix} I_2 & R_{\pi/2} p \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} = H_p \xi$$

Hence, the final derivative of the group action is

$$\frac{\partial q(\xi)}{\partial \xi} = R \begin{bmatrix} I_2 & R_{\pi/2} p \end{bmatrix} = \begin{bmatrix} R & R_{\pi/2} q \end{bmatrix}$$

The derivative of the inverse action $\hat{q} = T^{-1}\hat{p}$ is given by (5), specialized to $SE(2)$:

$$\frac{\partial \left(T^{-1}\hat{p}\right)}{\partial \xi} = -T^{-1} \frac{\partial \left(Ad_T \hat{\xi}\right) \hat{p}}{\partial \xi}$$

where the velocity now is

$$\left(Ad_T \hat{\xi}\right) \hat{p} = \begin{bmatrix} [\omega]_+ & Rv - \omega R_{\pi/2} t \\ 0 & 0 \end{bmatrix} \begin{bmatrix} p \\ 1 \end{bmatrix} = \begin{bmatrix} Rv + R_{\pi/2}(p-t)\omega \\ 0 \end{bmatrix}$$

and hence

$$\frac{\partial q(\xi)}{\partial \xi} = -R^T \begin{bmatrix} R & R_{\pi/2}(p-t) \end{bmatrix} = \begin{bmatrix} -I_2 & -R_{\pi/2} q \end{bmatrix}$$

6 3D Rotations

6.1 Derivatives of Mappings

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,

$$[\omega']_\times = Ad_{R_2^T}\left([\omega]_\times\right) = [R_2^T \omega]_\times$$

$$\frac{\partial\left(R_1 R_2\right)}{\partial \omega_1} = R_2^T \text{ and } \frac{\partial\left(R_1 R_2\right)}{\partial \omega_2} = I_3$$

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,

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

6.2 Derivatives of Actions

In the case of $SO(3)$ the vector space is $\mathbb{R}^3$, and the group action corresponds to rotating a point

$$q = Rp$$

We would now like to know what an incremental rotation parameterized by ω would do:

$$q(\omega) = Re^{[\omega]_\times} p$$

hence the derivative (following the exposition in Section 2):

$$\frac{\partial q(\omega)}{\partial \omega} = R\frac{\partial}{\partial \omega}\left(e^{[\omega]_\times} p\right) = R\frac{\partial}{\partial \omega}\left([\omega]_\times p\right) = RH_p$$

To calculate H_p we make use of

$$[\omega]_\times p = \omega \times p = -p \times \omega = [-p]_\times \omega$$

Hence, the final derivative of an action in its first argument is

$$\frac{\partial q(\omega)}{\partial \omega} = RH_p = R[-p]_\times$$

The derivative of the inverse action is given by 5, specialized to $SO(3)$:

$$\frac{\partial q(\omega)}{\partial \omega} = -R^T \frac{\partial\left([R\omega\ ]_\times p\right)}{\partial \omega} = R^T [p]_\times R = [R^T p]_\times$$

7 3D Rigid Transformations

7.1 Derivatives of Mappings

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

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

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

$$\frac{\partial (T_1 T_2)}{\partial \xi_1} = Ad_{T_2^{-1}} = \begin{bmatrix} R_2^T & 0 \\ [-R_2^T t_2]_\times R_2^T & R_2^T \end{bmatrix} = \begin{bmatrix} R_2^T & 0 \\ R_2^T [-t_2]_\times & R_2^T \end{bmatrix}$$

compose in its second argument,

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

between in its first argument,

$$\frac{\partial \left(T_1^{-1} T_2\right)}{\partial \xi_1} = Ad_{T_{21}} = -\begin{bmatrix} R & 0 \\ [t]_\times R & R \end{bmatrix}$$

with

$$T_{12} = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} = T_1^{-1} T_2 = between(T_2, T_1)$$

and between in its second argument,

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

7.2 The derivatives of Actions

The action of $SE(3)$ on 3D points is done by embedding the points in $\mathbb{R}^4$ by using homogeneous coordinates

$$\hat{q} = \begin{bmatrix} q \\ 1 \end{bmatrix} = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} p \\ 1 \end{bmatrix} = T\hat{p}$$

We would now like to know what an incremental rotation parameterized by ξ would do:

$$\hat{q}(\xi) = Te^{\hat{\xi}} \hat{p}$$

hence the derivative (following the exposition in Section 2):

$$\frac{\partial \hat{q}(\xi)}{\partial \xi} = T \frac{\partial}{\partial \xi} \left(\hat{\xi} \hat{p}\right) = TH_p$$

where $\hat{\xi}\hat{p}$ corresponds to a velocity in $\mathbb{R}^4$ (in the local T frame):

$$\hat{\xi}\hat{p} = \begin{bmatrix} [\omega]_\times & v \\ 0 & 0 \end{bmatrix} \begin{bmatrix} p \\ 1 \end{bmatrix} = \begin{bmatrix} \omega \times p + v \\ 0 \end{bmatrix}$$

Notice how velocities are anologous to points at infinity in projective geometry: they correspond to free vectors indicating a direction and magnitude of change.

By only taking the top three rows, we can write this as a velocity in $\mathbb{R}^3$, as the product of a 3×6 matrix H_p that acts upon the exponential coordinates ξ directly:

$$\omega \times p + v = -p \times \omega + v = \begin{bmatrix} -[p]_\times & I_3 \end{bmatrix} \begin{bmatrix} \omega \\ v \end{bmatrix} = H_p \xi$$

Hence, the final derivative of the group action is

$$\frac{\partial \hat{q}(\xi)}{\partial \xi} = T\hat{H}_p = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} [-p]_\times & I_3 \\ 0 & 0 \end{bmatrix}$$

in homogenous coordinates. In $\mathbb{R}^3$ this becomes:

$$\frac{\partial q(\xi)}{\partial \xi} = R \begin{bmatrix} -[p]_\times & I_3 \end{bmatrix}$$

7.3 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:

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

8 2D Line Segments (Ocaml)

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

9 Line3vd (Ocaml)

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_w^c, t^w) is done by

$$v^c = R_w^c v^w$$

$$d^c = R_w^c \left(d^w + (t^w v^w) v^w - t^w\right)$$

10 Line3 (Ocaml)

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_w^c, t^w) is done by

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

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:

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

The scalar derivatives can be found by realizing that

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

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

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

11 Aligning 3D Scans

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

Our model is

$$p^c = R\left(p^w - t\right)$$

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

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

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

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

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

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

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