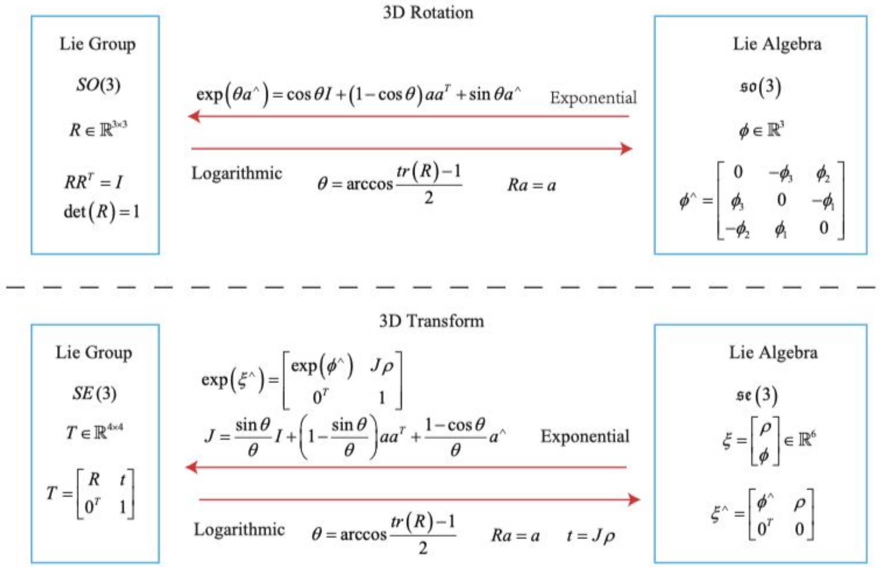
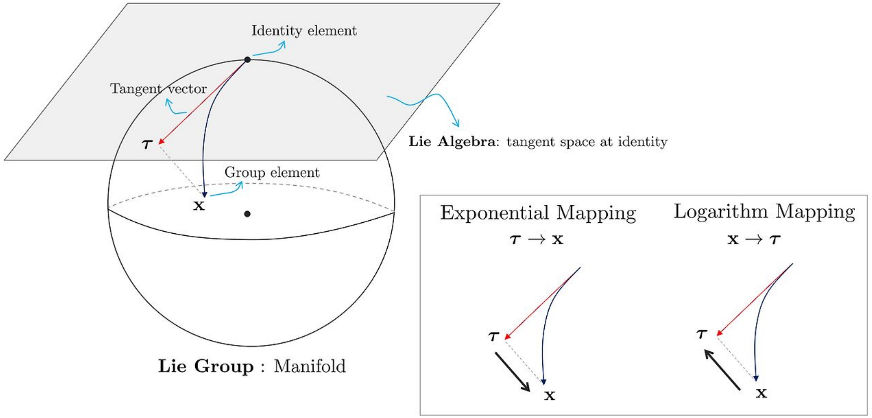
자율주행을 위한 비주얼 슬램 스터디

리 대수 미분 및 Sophus

Presenter. 김대완

Review


3D Rotation
Lie Group
$SO(3)$
$R \in \mathbb{R}^{3\times3}$
$RR^T = I$
$\det(R) = 1$
$\exp(\theta a^\wedge) = \cos\theta I + (1-\cos\theta) aa^T + \sin\theta a^\wedge$
Exponential
Logarithmic
$\theta = \arccos\frac{tr(R)-1}{2}$
$Ra = a$
Lie Algebra
$\mathfrak{so}(3)$
$\phi \in \mathbb{R}^3$
$\phi^\wedge = \begin{bmatrix} 0 & -\phi_3 & \phi_2 \\ \phi_3 & 0 & -\phi_1 \\ -\phi_2 & \phi_1 & 0 \end{bmatrix}$
3D Transform
Lie Group
$SE(3)$
$T \in \mathbb{R}^{4\times4}$
$T = \begin{bmatrix} R & t \\ 0^T & 1 \end{bmatrix}$
$\exp(\xi^\wedge) = \begin{bmatrix} \exp(\phi^\wedge) & J\rho \\ 0^T & 1 \end{bmatrix}$
$J = \frac{\sin\theta}{\theta} I + \left(1 - \frac{\sin\theta}{\theta}\right) aa^T + \frac{1-\cos\theta}{\theta} a^\wedge$
Exponential
Logarithmic
$\theta = \arccos\frac{tr(R)-1}{2}$
$Ra = a$
$t = J\rho$
Lie Algebra
$\mathfrak{se}(3)$
$\xi = \begin{bmatrix} \rho \\ \phi \end{bmatrix} \in \mathbb{R}^6$
$\xi^\wedge = \begin{bmatrix} \phi^\wedge & \rho \\ 0^T & 0 \end{bmatrix}$


Identity element
Tangent vector
τ
Group element
x
Lie Algebra: tangent space at identity
Lie Group : Manifold
Exponential Mapping
$\tau \to \mathbf{x}$
Logarithm Mapping
$\mathbf{x} \to \tau$
τ
x
τ
x

BCH 수식 및 근사치

$$a, b \in \mathbb{R},\ e^a e^b = e^{a+b}$$

BCH 수식 및 근사치

$$a, b \in \mathbb{R},\ e^a e^b = e^{a+b}$$

$$\exp(\phi^\wedge) = \sum \frac{1}{n!}(\phi^\wedge)^n \in SO(3)$$

then…

BCH 수식 및 근사치

$$a, b \in \mathbb{R},\ e^a e^b = e^{a+b}$$

$$\exp(\phi^\wedge) = \sum \frac{1}{n!}(\phi^\wedge)^n \in SO(3)$$

then…

$$\exp(\phi_1^\wedge)\exp(\phi_2^\wedge) = \exp((\phi_1 + \phi_2)^\wedge)$$

$$\ln(\exp(A)\exp(B)) = A + B$$

BCH 수식 및 근사치

$$a, b \in \mathbb{R},\ e^a e^b = e^{a+b}$$

$$\exp(\phi^\wedge) = \sum \frac{1}{n!}(\phi^\wedge)^n \in SO(3)$$

then…

$$\exp(\phi_1^\wedge)\exp(\phi_2^\wedge) = \exp((\phi_1 + \phi_2)^\wedge)$$

$$\ln(\exp(A)\exp(B)) = A + B$$

incorrect!

BCH(Baker-Campbell-Hausdorff) 수식 및 근사치

$$\ln(\exp(A)\exp(B)) \approx A + B + \frac{1}{2}[A, B] + \frac{1}{12}[A, [A, B]] - \frac{1}{12}[B, [B, A]] + \cdots$$

$$[A, B] = AB - BA$$

correct!

https://en.wikipedia.org/wiki/Baker%E2%80%93Campbell%E2%80%93Hausdorff_formula

BCH(Baker-Campbell-Hausdorff) 수식 및 근사치

$$\ln(\exp(A)\exp(B)) \approx A + B + \frac{1}{2}[A, B] + \frac{1}{12}[A, [A, B]] - \frac{1}{12}[B, [B, A]] + \cdots$$

$$[A, B] = AB - BA$$

$\text{if } \phi_1, \phi_2 \text{ small}$

$$\ln(\exp(A)\exp(B)) \approx A + B \boxed{+ \frac{1}{2}[A, B] + \frac{1}{12}[A, [A, B]] - \frac{1}{12}[B, [B, A]] + \cdots}$$

Ignore!

BCH(Baker-Campbell-Hausdorff) 수식 및 근사치

- Approximate BCH formulas

$$\ln\left(\exp(\boldsymbol{\phi}_1^\wedge)\exp(\boldsymbol{\phi}_2^\wedge)\right)^\vee \approx \begin{cases} \mathbf{J}_\ell(\boldsymbol{\phi}_2)^{-1}\,\boldsymbol{\phi}_1 + \boldsymbol{\phi}_2 & \text{if } \boldsymbol{\phi}_1 \text{ small} \\ \boldsymbol{\phi}_1 + \mathbf{J}_r(\boldsymbol{\phi}_1)^{-1}\,\boldsymbol{\phi}_2 & \text{if } \boldsymbol{\phi}_2 \text{ small} \end{cases}$$

$$\mathbf{J}_r(\boldsymbol{\phi})^{-1} = \sum_{n=0}^{\infty} \frac{B_n}{n!}\left(-\boldsymbol{\phi}^\wedge\right)^n = \frac{\phi}{2}\cot\frac{\phi}{2}\mathbf{1} + \left(1 - \frac{\phi}{2}\cot\frac{\phi}{2}\right)\mathbf{a}\mathbf{a}^T + \frac{\phi}{2}\mathbf{a}^\wedge \qquad \phi = \mathbf{a}\theta$$

$$\mathbf{J}_\ell(\boldsymbol{\phi})^{-1} = \sum_{n=0}^{\infty} \frac{B_n}{n!}\left(\boldsymbol{\phi}^\wedge\right)^n = \frac{\phi}{2}\cot\frac{\phi}{2}\mathbf{1} + \left(1 - \frac{\phi}{2}\cot\frac{\phi}{2}\right)\mathbf{a}\mathbf{a}^T - \frac{\phi}{2}\mathbf{a}^\wedge$$

BCH(Baker-Campbell-Hausdorff) 수식 및 근사치

$$\exp(\Delta\phi^\wedge)\exp(\phi^\wedge) = \exp((\phi + J_l^{-1}(\phi)\Delta\phi)^\wedge)$$

Lie algebra에서 덧셈 연산을 한다면?

$$\exp((\phi + \Delta\phi)^\wedge) = \exp((J_l\Delta\phi)^\wedge)\exp(\phi^\wedge) = \exp(\phi^\wedge)\exp((J_r\Delta\phi)^\wedge)$$

$$\exp((J_l\Delta\phi)^\wedge)\exp(\phi^\wedge) = \exp((\phi + J_l J_l^{-1}\Delta\phi)^\wedge)$$

proof

$$= \exp((\phi + \Delta\phi)^\wedge)$$

SO(3)에서 리 대수 유도

$$\mathbf{z} = \mathbf{T}\mathbf{p} + \omega$$

$$\mathbf{e} = \mathbf{z} - \mathbf{T}\mathbf{p}$$

n개의 measurement 에서 error를 최소로 하는 T를 찾으려면??

$$\min_{\mathbf{T}} J(\mathbf{T}) = \sum_{i=1}^{N} \|\mathbf{z}_i - \mathbf{T}\mathbf{p}\|_2^2$$

SO(3)에서 리 대수 유도

$$\mathbf{z} = \mathbf{T}\mathbf{p} + \omega$$

Measurement

$$\mathbf{e} = \mathbf{z} - \mathbf{T}\mathbf{p}$$

Error function

n개의 measurement 에서 error를 최소로 하는 T를 찾으려면??

$$\min_{\mathbf{T}} J(\mathbf{T}) = \sum_{i=1}^{N} \|\mathbf{z}_i - \mathbf{T}\mathbf{p}\|_2^2$$

Cost function

SO(3)에서 리 대수 유도

- Cost function을 Optimization 하려면? -> derivative!
  - Gradient descent, Gauss Newton, LM etc…

$$\min_{\mathbf{T}} J(\mathbf{T}) = \sum_{i=1}^{N} \|\mathbf{z}_i - \mathbf{T}\mathbf{p}\|_2^2$$

$$\lim_{\Delta x \to 0} \frac{f(x + \Delta x) - f(x)}{\Delta x}$$

SO(3)에서 리 대수 유도

- Cost function을 Optimization 하려면? -> derivative!
  - Gradient descent, Gauss Newton, LM etc…

$$\min_{\mathbf{T}} J(\mathbf{T}) = \sum_{i=1}^{N} \|\mathbf{z}_i - \mathbf{T}\mathbf{p}\|_2^2$$

$$\lim_{\Delta x \to 0} \frac{f(x + \Delta x) - f(x)}{\Delta x}$$

$$\text{if } A, B \in SO(3)$$

$$A + B \notin SO(3)$$

Lie group에서 별도의 constraint 필요!

SO(3)에서 리 대수 유도

- Cost function을 Optimization 하려면? -> derivative!
  - Gradient descent, Gauss Newton, LM etc…

$$\min_{\mathbf{T}} J(\mathbf{T}) = \sum_{i=1}^{N} \|\mathbf{z}_i - \mathbf{T}\mathbf{p}\|_2^2$$

$$\lim_{\Delta x \to 0} \frac{f(x+\Delta x) - f(x)}{\Delta x}$$

$$\text{if } A, B \in SO(3)$$

$$A + B \notin SO(3)$$

Lie group에서 별도의 constraint 필요

Lie algebra 관점에서 해결해보자!

리 대수 미분 모델

$$\mathbf{p} \in \mathbb{R}^3,\ R \in SO(3)$$

p라는 점을 R로 회전시킨 후 회전에 의한 점 좌표의 미분

$$\frac{\partial(\mathbf{Rp})}{\partial \mathbf{R}}$$

덧셈 연산이 없으므로 불가능한 식!

리 대수 미분 모델

$\mathbf{p} \in \mathbb{R}^3,\ R \in SO(3)$

p라는 점을 R로 회전시킨 후 회전에 의한 점 좌표의 미분

$$\frac{\partial(\mathbf{Rp})}{\partial\,\mathbf{R}}$$

Lie algebra로 표현 →

$$\frac{\partial(\exp(\phi^\wedge)\mathbf{p})}{\partial\phi}$$

리 대수 미분 모델

$$\begin{aligned}
\frac{\partial\left(\exp\left(\boldsymbol{\phi}^{\wedge}\right)\mathbf{p}\right)}{\partial\boldsymbol{\phi}} &= \lim_{\delta\boldsymbol{\phi}\to\mathbf{0}}\frac{\exp\left((\boldsymbol{\phi}+\delta\boldsymbol{\phi})^{\wedge}\right)\mathbf{p}-\exp\left(\boldsymbol{\phi}^{\wedge}\right)\mathbf{p}}{\delta\boldsymbol{\phi}} \\
&= \lim_{\delta\boldsymbol{\phi}\to\mathbf{0}}\frac{\exp\left((\mathbf{J}_l\delta\boldsymbol{\phi})^{\wedge}\right)\exp\left(\boldsymbol{\phi}^{\wedge}\right)\mathbf{p}-\exp\left(\boldsymbol{\phi}^{\wedge}\right)\mathbf{p}}{\delta\boldsymbol{\phi}} \\
&= \lim_{\delta\boldsymbol{\phi}\to\mathbf{0}}\frac{\left(\mathbf{I}+(\mathbf{J}_l\delta\boldsymbol{\phi})^{\wedge}\right)\exp\left(\boldsymbol{\phi}^{\wedge}\right)\mathbf{p}-\exp\left(\boldsymbol{\phi}^{\wedge}\right)\mathbf{p}}{\delta\boldsymbol{\phi}} \\
&= \lim_{\delta\boldsymbol{\phi}\to\mathbf{0}}\frac{(\mathbf{J}_l\delta\boldsymbol{\phi})^{\wedge}\exp\left(\boldsymbol{\phi}^{\wedge}\right)\mathbf{p}}{\delta\boldsymbol{\phi}} \\
&= \lim_{\delta\boldsymbol{\phi}\to\mathbf{0}}\frac{-\left(\exp\left(\boldsymbol{\phi}^{\wedge}\right)\mathbf{p}\right)^{\wedge}\mathbf{J}_l\delta\boldsymbol{\phi}}{\delta\boldsymbol{\phi}} = -(\mathbf{R}\mathbf{p})^{\wedge}\mathbf{J}_l.
\end{aligned}$$

BCH 근사치

$$\exp((\phi+\Delta\phi)^{\wedge}) = \exp((J_l\Delta\phi)^{\wedge})\exp(\phi^{\wedge}) = \exp(\phi^{\wedge})\exp((J_r\Delta\phi)^{\wedge})$$

리 대수 미분 모델

$$
\begin{aligned}
\frac{\partial\left(\exp\left(\boldsymbol{\phi}^{\wedge}\right)\mathbf{p}\right)}{\partial\boldsymbol{\phi}} &= \lim_{\delta\boldsymbol{\phi}\to\mathbf{0}} \frac{\exp\left((\boldsymbol{\phi}+\delta\boldsymbol{\phi})^{\wedge}\right)\mathbf{p}-\exp\left(\boldsymbol{\phi}^{\wedge}\right)\mathbf{p}}{\delta\boldsymbol{\phi}} \\
&= \lim_{\delta\boldsymbol{\phi}\to\mathbf{0}} \frac{\exp\left((\mathbf{J}_l\delta\boldsymbol{\phi})^{\wedge}\right)\exp\left(\boldsymbol{\phi}^{\wedge}\right)\mathbf{p}-\exp\left(\boldsymbol{\phi}^{\wedge}\right)\mathbf{p}}{\delta\boldsymbol{\phi}} \\
&= \lim_{\delta\boldsymbol{\phi}\to\mathbf{0}} \frac{\left(\mathbf{I}+(\mathbf{J}_l\delta\boldsymbol{\phi})^{\wedge}\right)\exp\left(\boldsymbol{\phi}^{\wedge}\right)\mathbf{p}-\exp\left(\boldsymbol{\phi}^{\wedge}\right)\mathbf{p}}{\delta\boldsymbol{\phi}} \\
&= \lim_{\delta\boldsymbol{\phi}\to\mathbf{0}} \frac{(\mathbf{J}_l\delta\boldsymbol{\phi})^{\wedge}\exp\left(\boldsymbol{\phi}^{\wedge}\right)\mathbf{p}}{\delta\boldsymbol{\phi}} \\
&= \lim_{\delta\boldsymbol{\phi}\to\mathbf{0}} \frac{-\left(\exp\left(\boldsymbol{\phi}^{\wedge}\right)\mathbf{p}\right)^{\wedge}\mathbf{J}_l\delta\boldsymbol{\phi}}{\delta\boldsymbol{\phi}} = -(\mathbf{R}\mathbf{p})^{\wedge}\mathbf{J}_l.
\end{aligned}
$$

$$\exp(\phi^{\wedge}) = \sum \frac{1}{n!}(\phi^{\wedge})^{n} \in SO(3)$$

값이 작으므로 Taylor series의 고차항을 제거함

리 대수 미분 모델

$$\begin{aligned}
\frac{\partial\left(\exp\left(\boldsymbol{\phi}^{\wedge}\right)\mathbf{p}\right)}{\partial\boldsymbol{\phi}} &= \lim_{\delta\boldsymbol{\phi}\to\mathbf{0}} \frac{\exp\left(\left(\boldsymbol{\phi}+\delta\boldsymbol{\phi}\right)^{\wedge}\right)\mathbf{p}-\exp\left(\boldsymbol{\phi}^{\wedge}\right)\mathbf{p}}{\delta\boldsymbol{\phi}} \\
&= \lim_{\delta\boldsymbol{\phi}\to\mathbf{0}} \frac{\exp\left(\left(\mathbf{J}_l\delta\boldsymbol{\phi}\right)^{\wedge}\right)\exp\left(\boldsymbol{\phi}^{\wedge}\right)\mathbf{p}-\exp\left(\boldsymbol{\phi}^{\wedge}\right)\mathbf{p}}{\delta\boldsymbol{\phi}} \\
&= \lim_{\delta\boldsymbol{\phi}\to\mathbf{0}} \frac{\left(\mathbf{I}+\left(\mathbf{J}_l\delta\boldsymbol{\phi}\right)^{\wedge}\right)\exp\left(\boldsymbol{\phi}^{\wedge}\right)\mathbf{p}-\exp\left(\boldsymbol{\phi}^{\wedge}\right)\mathbf{p}}{\delta\boldsymbol{\phi}} \\
&= \lim_{\delta\boldsymbol{\phi}\to\mathbf{0}} \frac{\left(\mathbf{J}_l\delta\boldsymbol{\phi}\right)^{\wedge}\exp\left(\boldsymbol{\phi}^{\wedge}\right)\mathbf{p}}{\delta\boldsymbol{\phi}} \\
&= \lim_{\delta\boldsymbol{\phi}\to\mathbf{0}} \frac{-\left(\exp\left(\boldsymbol{\phi}^{\wedge}\right)\mathbf{p}\right)^{\wedge}\mathbf{J}_l\delta\boldsymbol{\phi}}{\delta\boldsymbol{\phi}} = -(\mathbf{R}\mathbf{p})^{\wedge}\mathbf{J}_l.
\end{aligned}$$

$$(\mathbf{I}+(\mathbf{J}_l\delta\phi)^{\wedge}-\mathbf{I})\exp(\phi^{\wedge})\mathbf{p}$$

리 대수 미분 모델

$$\begin{aligned}
\frac{\partial(\exp(\phi^\wedge)\mathbf{p})}{\partial\phi} &= \lim_{\delta\phi\to\mathbf{0}} \frac{\exp\left((\phi+\delta\phi)^\wedge\right)\mathbf{p}-\exp(\phi^\wedge)\mathbf{p}}{\delta\phi} \\
&= \lim_{\delta\phi\to\mathbf{0}} \frac{\exp\left((\mathbf{J}_l\delta\phi)^\wedge\right)\exp(\phi^\wedge)\mathbf{p}-\exp(\phi^\wedge)\mathbf{p}}{\delta\phi} \\
&= \lim_{\delta\phi\to\mathbf{0}} \frac{\left(\mathbf{I}+(\mathbf{J}_l\delta\phi)^\wedge\right)\exp(\phi^\wedge)\mathbf{p}-\exp(\phi^\wedge)\mathbf{p}}{\delta\phi} \\
&= \lim_{\delta\phi\to\mathbf{0}} \frac{(\mathbf{J}_l\delta\phi)^\wedge\exp(\phi^\wedge)\mathbf{p}}{\delta\phi} \\
&= \lim_{\delta\phi\to\mathbf{0}} \frac{-(\exp(\phi^\wedge)\mathbf{p})^\wedge\mathbf{J}_l\delta\phi}{\delta\phi} = -(\mathbf{R}\mathbf{p})^\wedge\mathbf{J}_l.
\end{aligned}$$

복잡한 형태의 J 존재,
섭동 모델을 이용하여 계산한다면?

섭동 모델

- 해석적으로 풀 수 없는 문제의 해를 매우 작다고 여길 수 있는 매개변수들의 테일러 급수로 나타냄
- R에서 섭동 ΔR을 수행

$$\mathbf{R} = \phi^\wedge, \Delta\mathbf{R} = \varphi^\wedge$$

$$\begin{aligned}
\frac{\partial(\mathbf{Rp})}{\partial\varphi} &= \lim_{\varphi\to 0} \frac{\boxed{\exp(\varphi^\wedge)\exp(\phi^\wedge)}\mathbf{p} - \exp(\phi^\wedge)\mathbf{p}}{\varphi} \\
&= \lim_{\varphi\to 0} \frac{(\mathbf{I} + \varphi^\wedge)\exp(\phi^\wedge)\mathbf{p} - \exp(\phi^\wedge)\mathbf{p}}{\varphi} \\
&= \lim_{\varphi\to 0} \frac{\varphi^\wedge \mathbf{Rp}}{\varphi} = \lim_{\varphi\to 0} \frac{-(\mathbf{Rp})^\wedge \varphi}{\varphi} = -(\mathbf{Rp})^\wedge.
\end{aligned}$$

섭동 모델

- 해석적으로 풀 수 없는 문제의 해를 매우 작다고 여길 수 있는 매개변수들의 테일러 급수로 나타냄
- R에서 섭동 ΔR을 수행

$$\mathbf{R} = \phi^\wedge, \Delta\mathbf{R} = \varphi^\wedge$$

$$\begin{aligned}
\frac{\partial(\mathbf{Rp})}{\partial\varphi} &= \lim_{\varphi \to 0} \frac{\exp(\varphi^\wedge)\exp(\phi^\wedge)\mathbf{p} - \exp(\phi^\wedge)\mathbf{p}}{\varphi} \\
&= \lim_{\varphi \to 0} \frac{\boxed{(\mathbf{I} + \varphi^\wedge)}\exp(\phi^\wedge)\mathbf{p} - \exp(\phi^\wedge)\mathbf{p}}{\varphi} \\
&= \lim_{\varphi \to 0} \frac{\varphi^\wedge \mathbf{Rp}}{\varphi} = \lim_{\varphi \to 0} \frac{-(\mathbf{Rp})^\wedge \varphi}{\varphi} = -(\mathbf{Rp})^\wedge.
\end{aligned}$$

값이 작으므로 Taylor series의 고차항을 제거함

섭동 모델

- 해석적으로 풀 수 없는 문제의 해를 매우 작다고 여길 수 있는 매개변수들의 테일러 급수로 나타냄
- R에서 섭동 ΔR을 수행

$$\mathbf{R} = \phi^{\wedge}, \Delta\mathbf{R} = \varphi^{\wedge}$$

$$\begin{aligned}\frac{\partial(\mathbf{R}\mathbf{p})}{\partial\varphi} &= \lim_{\varphi\to 0}\frac{\exp(\varphi^{\wedge})\exp(\phi^{\wedge})\mathbf{p} - \exp(\phi^{\wedge})\mathbf{p}}{\varphi} \\ &= \lim_{\varphi\to 0}\frac{(\mathbf{I}+\varphi^{\wedge})\exp(\phi^{\wedge})\mathbf{p} - \exp(\phi^{\wedge})\mathbf{p}}{\varphi} \\ &= \lim_{\varphi\to 0}\frac{\varphi^{\wedge}\mathbf{R}\mathbf{p}}{\varphi} = \lim_{\varphi\to 0}\frac{-(\mathbf{R}\mathbf{p})^{\wedge}\varphi}{\varphi} = \boxed{-(\mathbf{R}\mathbf{p})^{\wedge}}.\end{aligned}$$

실용적이게 포즈 추정이 가능

SE(3)에서 섭동 모델

$$
\begin{aligned}
\frac{\partial(\mathbf{T}\mathbf{p})}{\partial \delta \boldsymbol{\xi}} &= \lim_{\delta \boldsymbol{\xi} \rightarrow \mathbf{0}} \frac{\exp\left(\delta \boldsymbol{\xi}^{\wedge}\right) \exp\left(\boldsymbol{\xi}^{\wedge}\right) \mathbf{p} - \exp\left(\boldsymbol{\xi}^{\wedge}\right) \mathbf{p}}{\delta \boldsymbol{\xi}} \\
&= \lim_{\delta \boldsymbol{\xi} \rightarrow \mathbf{0}} \frac{\left(\mathbf{I} + \delta \boldsymbol{\xi}^{\wedge}\right) \exp\left(\boldsymbol{\xi}^{\wedge}\right) \mathbf{p} - \exp\left(\boldsymbol{\xi}^{\wedge}\right) \mathbf{p}}{\delta \boldsymbol{\xi}} \\
&= \lim_{\delta \boldsymbol{\xi} \rightarrow \mathbf{0}} \frac{\delta \boldsymbol{\xi}^{\wedge} \exp\left(\boldsymbol{\xi}^{\wedge}\right) \mathbf{p}}{\delta \boldsymbol{\xi}} \\
&= \lim_{\delta \boldsymbol{\xi} \rightarrow \mathbf{0}} \frac{\begin{bmatrix} \delta \boldsymbol{\phi}^{\wedge} & \delta \boldsymbol{\rho} \\ \mathbf{0}^{\mathrm{T}} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{R}\mathbf{p} + \mathbf{t} \\ 1 \end{bmatrix}}{\delta \boldsymbol{\xi}} \\
&= \lim_{\delta \boldsymbol{\xi} \rightarrow \mathbf{0}} \frac{\begin{bmatrix} \delta \boldsymbol{\phi}^{\wedge}(\mathbf{R}\mathbf{p} + \mathbf{t}) + \delta \boldsymbol{\rho} \\ \mathbf{0}^{\mathrm{T}} \end{bmatrix}}{[\delta \boldsymbol{\rho}, \delta \boldsymbol{\phi}]^{\mathrm{T}}} = \begin{bmatrix} \mathbf{I} & -(\mathbf{R}\mathbf{p} + \mathbf{t})^{\wedge} \\ \mathbf{0}^{\mathrm{T}} & \mathbf{0}^{\mathrm{T}} \end{bmatrix} \triangleq (\mathbf{T}\mathbf{p})^{\odot}.
\end{aligned}
$$

SE(3)에서 섭동 모델

$$
\begin{aligned}
\frac{\partial(\mathbf{Tp})}{\partial \delta \boldsymbol{\xi}} &= \lim_{\delta \boldsymbol{\xi} \to \mathbf{0}} \frac{\exp(\delta \boldsymbol{\xi}^\wedge)\exp(\boldsymbol{\xi}^\wedge)\mathbf{p} - \exp(\boldsymbol{\xi}^\wedge)\mathbf{p}}{\delta \boldsymbol{\xi}} \\
&= \lim_{\delta \boldsymbol{\xi} \to \mathbf{0}} \frac{(\mathbf{I} + \delta \boldsymbol{\xi}^\wedge)\exp(\boldsymbol{\xi}^\wedge)\mathbf{p} - \exp(\boldsymbol{\xi}^\wedge)\mathbf{p}}{\delta \boldsymbol{\xi}} \\
&= \lim_{\delta \boldsymbol{\xi} \to \mathbf{0}} \frac{\delta \boldsymbol{\xi}^\wedge \exp(\boldsymbol{\xi}^\wedge)\mathbf{p}}{\delta \boldsymbol{\xi}} \\
&= \lim_{\delta \boldsymbol{\xi} \to \mathbf{0}} \frac{\begin{bmatrix} \delta \boldsymbol{\phi}^\wedge & \delta \boldsymbol{\rho} \\ \mathbf{0}^\mathrm{T} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{Rp} + \mathbf{t} \\ 1 \end{bmatrix}}{\delta \boldsymbol{\xi}} \\
&= \lim_{\delta \boldsymbol{\xi} \to \mathbf{0}} \frac{\begin{bmatrix} \delta \boldsymbol{\phi}^\wedge (\mathbf{Rp} + \mathbf{t}) + \delta \boldsymbol{\rho} \\ \mathbf{0}^\mathrm{T} \end{bmatrix}}{[\delta \boldsymbol{\rho}, \delta \boldsymbol{\phi}]^\mathrm{T}} = \begin{bmatrix} \mathbf{I} & -(\mathbf{Rp} + \mathbf{t})^\wedge \\ \mathbf{0}^\mathrm{T} & \mathbf{0}^\mathrm{T} \end{bmatrix} \triangleq (\mathbf{Tp})^\odot.
\end{aligned}
$$

왜 리 대수 섭동모델 처럼 안될까?

$$
\lim_{\boldsymbol{\varphi} \to \mathbf{0}} \frac{\boldsymbol{\varphi}^\wedge \mathbf{Rp}}{\boldsymbol{\varphi}} = \lim_{\boldsymbol{\varphi} \to \mathbf{0}} \frac{-(\mathbf{Rp})^\wedge \boldsymbol{\varphi}}{\boldsymbol{\varphi}} = -(\mathbf{Rp})^\wedge.
$$

SE(3)에서 섭동 모델

$$\begin{aligned}
\frac{\partial(\mathbf{Tp})}{\partial\delta\boldsymbol{\xi}} &= \lim_{\delta\boldsymbol{\xi}\to\mathbf{0}} \frac{\exp(\delta\boldsymbol{\xi}^\wedge)\exp(\boldsymbol{\xi}^\wedge)\mathbf{p} - \exp(\boldsymbol{\xi}^\wedge)\mathbf{p}}{\delta\boldsymbol{\xi}} \\
&= \lim_{\delta\boldsymbol{\xi}\to\mathbf{0}} \frac{(\mathbf{I}+\delta\boldsymbol{\xi}^\wedge)\exp(\boldsymbol{\xi}^\wedge)\mathbf{p} - \exp(\boldsymbol{\xi}^\wedge)\mathbf{p}}{\delta\boldsymbol{\xi}} \\
&= \boxed{\lim_{\delta\boldsymbol{\xi}\to\mathbf{0}} \frac{\delta\boldsymbol{\xi}^\wedge\exp(\boldsymbol{\xi}^\wedge)\mathbf{p}}{\delta\boldsymbol{\xi}}} \\
&= \lim_{\delta\boldsymbol{\xi}\to\mathbf{0}} \frac{\begin{bmatrix} \delta\boldsymbol{\phi}^\wedge & \delta\boldsymbol{\rho} \\ \mathbf{0}^T & 0 \end{bmatrix}\begin{bmatrix} \mathbf{Rp}+\mathbf{t} \\ 1 \end{bmatrix}}{\delta\boldsymbol{\xi}} \\
&= \lim_{\delta\boldsymbol{\xi}\to\mathbf{0}} \frac{\begin{bmatrix} \delta\boldsymbol{\phi}^\wedge(\mathbf{Rp}+\mathbf{t})+\delta\boldsymbol{\rho} \\ \mathbf{0}^T \end{bmatrix}}{[\delta\boldsymbol{\rho},\delta\boldsymbol{\phi}]^T} = \begin{bmatrix} \mathbf{I} & -(\mathbf{Rp}+\mathbf{t})^\wedge \\ \mathbf{0}^T & \mathbf{0}^T \end{bmatrix} \triangleq (\mathbf{Tp})^\odot.
\end{aligned}$$

왜 리 대수 섭동모델 처럼 안될까?

$$\lim_{\boldsymbol{\varphi}\to\mathbf{0}} \frac{\boldsymbol{\varphi}^\wedge\mathbf{Rp}}{\boldsymbol{\varphi}} = \lim_{\boldsymbol{\varphi}\to\mathbf{0}} \frac{-(\mathbf{Rp})^\wedge\boldsymbol{\varphi}}{\boldsymbol{\varphi}} = -(\mathbf{Rp})^\wedge.$$

SE(3)은 skew-symmetric matrix가
아니기 때문!

$$\boldsymbol{\xi}^\wedge = \begin{bmatrix} \boldsymbol{\phi}^\wedge & \boldsymbol{\rho} \\ \mathbf{0}^T & 0 \end{bmatrix} \in \mathbb{R}^{4\times 4}.$$

SE(3)에서 섭동 모델

$$\begin{aligned}
\frac{\partial (\mathbf{T}\mathbf{p})}{\partial \delta \boldsymbol{\xi}} &= \lim_{\delta\boldsymbol{\xi}\to\mathbf{0}} \frac{\exp(\delta\boldsymbol{\xi}^\wedge)\exp(\boldsymbol{\xi}^\wedge)\mathbf{p} - \exp(\boldsymbol{\xi}^\wedge)\mathbf{p}}{\delta\boldsymbol{\xi}} \\
&= \lim_{\delta\boldsymbol{\xi}\to\mathbf{0}} \frac{(\mathbf{I} + \delta\boldsymbol{\xi}^\wedge)\exp(\boldsymbol{\xi}^\wedge)\mathbf{p} - \exp(\boldsymbol{\xi}^\wedge)\mathbf{p}}{\delta\boldsymbol{\xi}} \\
&= \lim_{\delta\boldsymbol{\xi}\to\mathbf{0}} \frac{\delta\boldsymbol{\xi}^\wedge \exp(\boldsymbol{\xi}^\wedge)\mathbf{p}}{\delta\boldsymbol{\xi}} \\
&= \lim_{\delta\boldsymbol{\xi}\to\mathbf{0}} \frac{\begin{bmatrix} \delta\boldsymbol{\phi}^\wedge & \delta\boldsymbol{\rho} \\ \mathbf{0}^{\mathrm{T}} & 0 \end{bmatrix}\begin{bmatrix} \mathbf{R}\mathbf{p} + \mathbf{t} \\ 1 \end{bmatrix}}{\delta\boldsymbol{\xi}} \\
&= \lim_{\delta\boldsymbol{\xi}\to\mathbf{0}} \frac{\begin{bmatrix} \delta\boldsymbol{\phi}^\wedge(\mathbf{R}\mathbf{p} + \mathbf{t}) + \delta\boldsymbol{\rho} \\ \mathbf{0}^{\mathrm{T}} \end{bmatrix}}{[\delta\boldsymbol{\rho}, \delta\boldsymbol{\phi}]^{\mathrm{T}}} = \begin{bmatrix} \mathbf{I} & -(\mathbf{R}\mathbf{p} + \mathbf{t})^\wedge \\ \mathbf{0}^{\mathrm{T}} & \mathbf{0}^{\mathrm{T}} \end{bmatrix} \triangleq (\mathbf{T}\mathbf{p})^\odot.
\end{aligned}$$

a X b = -b X a = a^b

$$\begin{bmatrix} \frac{\partial(-(\mathbf{R}\mathbf{p}+\mathbf{t}+\delta\rho)^\wedge+\delta\phi^\wedge)}{\partial\delta\rho} & \frac{\partial(-(\mathbf{R}\mathbf{p}+\mathbf{t}+\delta\rho)^\wedge+\delta\phi^\wedge)}{\partial\delta\phi} \\ \frac{\partial\mathbf{0}^T}{\partial\delta\rho} & \frac{\partial\mathbf{0}^T}{\partial\delta\phi} \end{bmatrix}$$

Sophus 실습

- c++ implementation of Lie groups commonly used for 2d and 3d geometric problems

https://github.com/strasdat/Sophus

Sophus 실습 - useSophus

```
*******************************
SE3 from R,t=
2.22045e-16          -1           0           1
          1 2.22045e-16           0           0
          0           0           1           0
          0           0           0           1
SE3 from q,t=
2.22045e-16          -1           0           1
          1 2.22045e-16           0           0
          0           0           1           0
          0           0           0           1
se3 =  0.785398 -0.785398           0           0           0      1.5708
se3 hat =
        0   -1.5708         0  0.785398
   1.5708         0        -0 -0.785398
       -0         0         0         0
        0         0         0         0
se3 hat vee =  0.785398 -0.785398           0           0           0      1.5708
SE3 updated =
2.22045e-16          -1           0      1.0001
          1 2.22045e-16           0           0
          0           0           1           0
          0           0           0           1
```

https://github.com/gaoxiang12/slambook2

Sophus 실습 - GaussNewton

```
initialGauss: 1 0 0 0
0 1 0 0
0 0 1 0
0 0 0 1
#Iteration 0.
cost: 8.00112
T:
  0.63235  0.170351  0.755721 -0.410987
  0.11205  0.945152 -0.306809 -0.191596
-0.766536  0.278689   0.57858   1.05265
        0         0         0         1
#Iteration 1.
cost: 4.43452
T:
  0.931598   0.142259   0.334495   -0.47993
-0.0196055   0.938559  -0.344561   0.141249
  -0.36296   0.314435   0.877149   0.364711
         0          0          0          1
#Iteration 2.
cost: 3.6357
T:
 0.719135  0.183931  0.670085 -0.665767
 0.182926  0.880205 -0.437924  0.145222
 -0.67036  0.437502   0.59934  0.763202
        0         0         0         1
#Iteration 3.
cost: 3.56049
T:
 0.905197  0.173098  0.388144  -0.54283
0.0194336  0.895482 -0.444674  0.303279
-0.424548  0.410061  0.807223    0.3302
        0         0         0         1
#Iteration 4.
cost: 3.54844
T:
 0.745608  0.185691   0.63999  -0.67915
 0.191033  0.860522 -0.472237  0.229778
-0.638416  0.474363  0.606139  0.635543
```

https://github.com/dawan0111/Simple-SE3-Optimization

Sophus 실습 - GaussNewton

```
initialGauss: 1 0 0 0
0 1 0 0
0 0 1 0
0 0 0 1
#Iteration 0.
cost: 8.00112
T:
  0.63235  0.170351  0.755721 -0.410987
  0.11205  0.945152 -0.306809 -0.191596
-0.766536  0.278689   0.57858   1.05265
        0         0         0         1
#Iteration 1.
cost: 4.43452
T:
  0.931598   0.142259   0.334495   -0.47993
-0.0196055   0.938559  -0.344561   0.141249
  -0.36296   0.314435   0.877149   0.364711
         0          0          0          1
#Iteration 2.
cost: 3.6357
T:
 0.719135  0.183931  0.670085 -0.665767
 0.182926  0.880205 -0.437924  0.145222
 -0.67036  0.437502   0.59934  0.763202
        0         0         0         1
#Iteration 3.
cost: 3.56049
T:
 0.905197  0.173098  0.388144  -0.54283
0.0194336  0.895482 -0.444674  0.303279
-0.424548  0.410061  0.807223    0.3302
        0         0         0         1
#Iteration 4.
cost: 3.54844
T:
 0.745608  0.185691   0.63999  -0.67915
 0.191033  0.860522 -0.472237  0.229778
-0.638416  0.474363  0.606139  0.635543
```

$$\mathbf{z} = \mathbf{T}\mathbf{p} + \omega$$

Measurement

$$\mathbf{e} = \mathbf{z} - \mathbf{T}\mathbf{p}$$

Error function

$$\min_{\mathbf{T}} J(\mathbf{T}) = \sum_{i=1}^{N} \|\mathbf{z}_i - \mathbf{T}\mathbf{p}\|_2^2$$

Cost function

https://github.com/dawan0111/Simple-SE3-Optimization

Sophus 실습 - GaussNewton

$$
\begin{aligned}
\frac{\partial(\mathbf{T}\mathbf{p})}{\partial\delta\boldsymbol{\xi}} &= \lim_{\delta\boldsymbol{\xi}\to\mathbf{0}} \frac{\exp\left(\delta\boldsymbol{\xi}^{\wedge}\right)\exp\left(\boldsymbol{\xi}^{\wedge}\right)\mathbf{p}-\exp\left(\boldsymbol{\xi}^{\wedge}\right)\mathbf{p}}{\delta\boldsymbol{\xi}} \\
&= \lim_{\delta\boldsymbol{\xi}\to\mathbf{0}} \frac{\left(\mathbf{I}+\delta\boldsymbol{\xi}^{\wedge}\right)\exp\left(\boldsymbol{\xi}^{\wedge}\right)\mathbf{p}-\exp\left(\boldsymbol{\xi}^{\wedge}\right)\mathbf{p}}{\delta\boldsymbol{\xi}} \\
&= \lim_{\delta\boldsymbol{\xi}\to\mathbf{0}} \frac{\delta\boldsymbol{\xi}^{\wedge}\exp\left(\boldsymbol{\xi}^{\wedge}\right)\mathbf{p}}{\delta\boldsymbol{\xi}} \\
&= \lim_{\delta\boldsymbol{\xi}\to\mathbf{0}} \frac{\begin{bmatrix}\delta\boldsymbol{\phi}^{\wedge} & \delta\boldsymbol{\rho} \\ \mathbf{0}^{\mathrm{T}} & 0\end{bmatrix}\begin{bmatrix}\mathbf{R}\mathbf{p}+\mathbf{t} \\ 1\end{bmatrix}}{\delta\boldsymbol{\xi}} \\
&= \lim_{\delta\boldsymbol{\xi}\to\mathbf{0}} \frac{\begin{bmatrix}\delta\boldsymbol{\phi}^{\wedge}\left(\mathbf{R}\mathbf{p}+\mathbf{t}\right)+\delta\boldsymbol{\rho} \\ \mathbf{0}^{\mathrm{T}}\end{bmatrix}}{[\delta\boldsymbol{\rho},\delta\boldsymbol{\phi}]^{\mathrm{T}}} = \begin{bmatrix}\mathbf{I} & -\left(\mathbf{R}\mathbf{p}+\mathbf{t}\right)^{\wedge} \\ \mathbf{0}^{\mathrm{T}} & \mathbf{0}^{\mathrm{T}}\end{bmatrix} \triangleq (\mathbf{T}\mathbf{p})^{\odot}.
\end{aligned}
$$

```cpp
ector6d gaussNewton(const std::vector<Eigen::Vector3d> &measures,
                    const std::vector<Eigen::Vector3d> &landmarks,
                    const Sophus::SE3<double> &T) {
Eigen::Matrix<double, 6, 6> H = Eigen::Matrix<double, 6, 6>::Zero();
Vector6d b = Vector6d::Zero();
Eigen::Matrix3d informationMatrix = Eigen::Matrix3d::Identity();

for (int i = 0; i < measures.size(); ++i) {
  auto measure = measures[i];
  auto landmark = landmarks[i];      You, 4 minutes ago • second commit!

  Eigen::Matrix<double, 3, 6> jacobian = Eigen::Matrix<double, 3, 6>::Zero();
  Eigen::Vector3d so3 = Eigen::Vector3d::Zero();
  so3 = T.rotationMatrix() * landmark + T.translation();
  auto error = so3 - measure;

  jacobian.block(0, 0, 3, 3) = Eigen::Matrix3d::Identity();
  jacobian.block(0, 3, 3, 3) = Sophus::SO3d::hat(so3) * -1.0;

  b += error.transpose() * informationMatrix * jacobian;
  H += jacobian.transpose() * informationMatrix * jacobian;
}

Vector6d dx = H.colPivHouseholderQr().solve(-b);
return dx;
```

https://github.com/dawan0111/Simple-SE3-Optimization

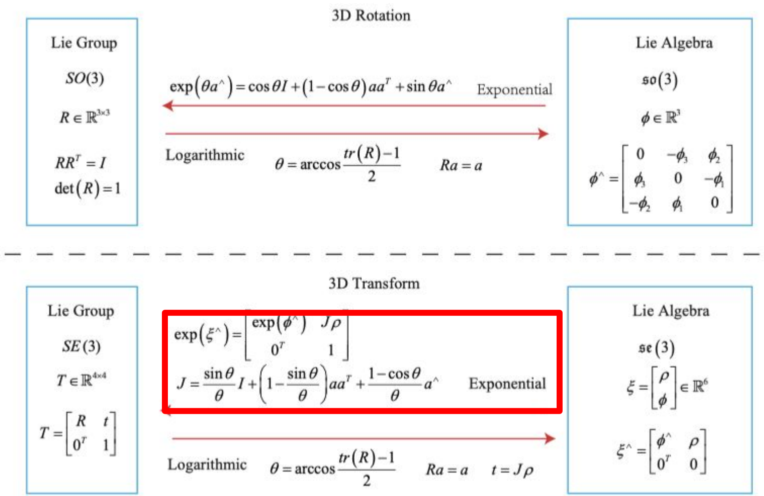
Sophus 실습 - GaussNewton

```
for (int i = 0; i < 10; ++i) {
  std::cout << "#Iteration " << i << "." << std::endl;
  auto cost = costFunction(measures, landmarks, T);
  std::cout << "cost: " << cost << std::endl;

  auto dT = gaussNewton(measures, landmarks, T);
  T = Sophus::SE3d::exp(dT) * T;
  std::cout << "T: " << std::endl;
  std::cout << T.matrix() << std::endl;
}
```

https://github.com/dawan0111/Simple-SE3-Optimization

Sophus 실습 - GaussNewton

```cpp
for (int i = 0; i < 10; ++i) {
  std::cout << "#Iteration " << i << "." << std::endl;
  auto cost = costFunction(measures, landmarks, T);
  std::cout << "cost: " << cost << std::endl;

  auto dT = gaussNewton(measures, landmarks, T);
  T = Sophus::SE3d::exp(dT) * T;
  std::cout << "T: " << std::endl;
  std::cout << T.matrix() << std::endl;
}
```



https://github.com/dawan0111/Simple-SE3-Optimization

Sophus 실습 - TrajectoryError

```
You, 23 hours ago | 3 authors (keineahnung2345 and others)
option(USE_UBUNTU_20 "Set to ON if you are using Ubuntu 20.04" ON)
find_package(Pangolin REQUIRED)
if(USE_UBUNTU_20)
    message("You are using Ubuntu 20.04, fmt::fmt will be linked")
    find_package(fmt REQUIRED)
    set(FMT_LIBRARIES fmt::fmt)
endif()
include_directories(${Pangolin_INCLUDE_DIRS})
add_executable(trajectoryError trajectoryError.cpp)
target_link_libraries(trajectoryError ${Pangolin_LIBRARIES} ${FMT_LIBRARIES} fmt
```

Modify

https://github.com/gaoxiang12/slambook2

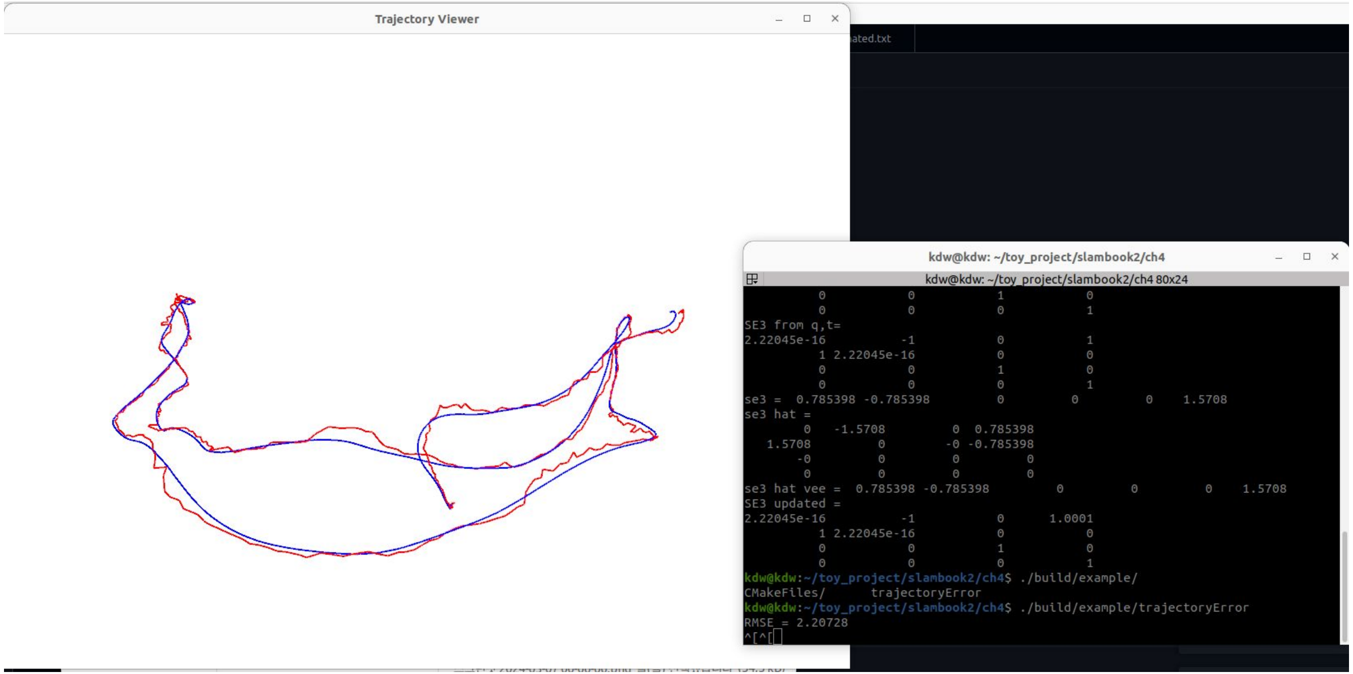
Sophus 실습 - TrajectoryError

$$\mathrm{ATE}_{\mathrm{all}} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} \| \log(\mathbf{T}_{\mathrm{gt},i}^{-1}\mathbf{T}_{\mathrm{esti},i})^{\vee} \|_2^2}, \tag{4.44}$$

```
// compute rmse
double rmse = 0;
for (size_t i = 0; i < estimated.size(); i++) {
  Sophus::SE3d p1 = estimated[i], p2 = groundtruth[i];
  double error = (p2.inverse() * p1).log().norm();
  rmse += error * error;
}       gaoxiang, 6 years ago • ch4 done
rmse = rmse / double(estimated.size());
rmse = sqrt(rmse);
cout << "RMSE = " << rmse << endl;
```

https://github.com/strasdat/Sophus

Sophus 실습 - TrajectoryError



https://github.com/strasdat/Sophus

유사 변환 군과 그의 리 대수

- mono의 경우 일반적으로 스케일 팩터를 명시적으로 처리한다. (카메라 좌표계에서 유사 변환이 전달)
- 리 대수 sim(3)는 7차원 벡터로 마지막에 스케일을 의미하는 변수가 하나 더 있다. x,y,z, roll, pitch, yaw, scale

$$\mathbf{p}' = \begin{bmatrix} s\mathbf{R} & \mathbf{t} \\ \mathbf{0}^{\mathrm{T}} & 1 \end{bmatrix} \mathbf{p} = s\mathbf{R}\mathbf{p} + \mathbf{t}. \tag{4.48}$$

$$\mathrm{Sim}(3) = \left\{ \mathbf{S} = \begin{bmatrix} s\mathbf{R} & \mathbf{t} \\ \mathbf{0}^{\mathrm{T}} & 1 \end{bmatrix} \in \mathbb{R}^{4\times4} \right\}. \tag{4.49}$$

$$\mathfrak{sim}(3) = \left\{ \boldsymbol{\zeta} | \boldsymbol{\zeta} = \begin{bmatrix} \boldsymbol{\rho} \\ \boldsymbol{\phi} \\ \sigma \end{bmatrix} \in \mathbb{R}^7, \boldsymbol{\zeta}^\wedge = \begin{bmatrix} \sigma\mathbf{I} + \boldsymbol{\phi}^\wedge & \boldsymbol{\rho} \\ \mathbf{0}^{\mathrm{T}} & 0 \end{bmatrix} \in \mathbb{R}^{4\times4} \right\}. \tag{4.50}$$

유사 변환 군과 그의 리 대수

Exponential mapping

$$\exp\left(\zeta^{\wedge}\right)=\begin{bmatrix} \mathrm{e}^{\sigma}\exp\left(\boldsymbol{\phi}^{\wedge}\right) & \mathbf{J}_{s}\boldsymbol{\rho} \\ \mathbf{0}^{\mathrm{T}} & 1 \end{bmatrix}. \tag{4.51}$$

Where $\mathbf{J}_s$ is:

$$\begin{aligned}\mathbf{J}_{s}=&\frac{\mathrm{e}^{\sigma}-1}{\sigma}\mathbf{I}+\frac{\sigma\mathrm{e}^{\sigma}\sin\theta+\left(1-\mathrm{e}^{\sigma}\cos\theta\right)\theta}{\sigma^{2}+\theta^{2}}\mathbf{a}^{\wedge}\\&+\left(\frac{\mathrm{e}^{\sigma}-1}{\sigma}-\frac{\left(\mathrm{e}^{\sigma}\cos\ Theta-1\right)\sigma+\left(\mathrm{e}^{\sigma}\sin\theta\right)\theta}{\sigma^{2}+\theta^{2}}\right)\mathbf{a}^{\wedge}\mathbf{a}^{\wedge}.\end{aligned}$$

$$s=\mathrm{e}^{\sigma},\ \mathbf{R}=\exp(\boldsymbol{\phi}^{\wedge}),\ \mathbf{t}=\mathbf{J}_{s}\boldsymbol{\rho}. \tag{4.52}$$

유사 변환 군과 그의 리 대수 미분

- 섭동을 부여하고 섭동에 대한 미분을 구한다.
- 4 x 7 jacobian 행렬을 갖는 7차원 벡터이다.

SE(3)

$$\begin{bmatrix} \mathbf{I} & -(\mathbf{R}\mathbf{p}+\mathbf{t})^\wedge \\ \mathbf{0}^T & \mathbf{0}^T \end{bmatrix} \triangleq (\mathbf{T}\mathbf{p})^\odot$$

Sim(3)

$$\frac{\partial \mathbf{S}\mathbf{p}}{\partial \boldsymbol{\zeta}} = \begin{bmatrix} \mathbf{I} & -\mathbf{q}^\wedge & \mathbf{q} \\ \mathbf{0}^T & \mathbf{0}^T & 0 \end{bmatrix}$$

유사 변환 군과 그의 리 대수 미분

- 섭동을 부여하고 섭동에 대한 미분을 구한다.
- 4 x 7 jacobian 행렬을 갖는 7차원 벡터이다.

SE(3)

$$\begin{bmatrix} \mathbf{I} & -(\mathbf{Rp}+\mathbf{t})^{\wedge} \\ \mathbf{0}^{\mathrm{T}} & \mathbf{0}^{\mathrm{T}} \end{bmatrix} \triangleq (\mathbf{Tp})^{\odot}$$

Sim(3)

$$\frac{\partial \mathbf{Sp}}{\partial \zeta} = \begin{bmatrix} \mathbf{I} & -\mathbf{q}^{\wedge} & \boxed{\begin{matrix}\mathbf{q} \\ 0\end{matrix}} \\ \mathbf{0}^{\mathrm{T}} & \mathbf{0}^{\mathrm{T}} & \end{bmatrix}$$

scale

Summary

- 리군 SO(3), SE(3) 및 이에 대응하는 리 대수에 대한 이해
- Exponential mapping 및 Logarithm Mapping에 대한 이해
- 리 대수를 사용하는 이유와 미분가능성
- 섭동모델을 통한 간소화
- Sophus 기본 사용법
- Sim(3)에 대한 간단한 이해