



深蓝学院
shenlanxueyuan.com

第一章作业讲解

主讲人 会打篮球的猫



第一题

1. VIO 文献阅读

阅读 VIO 相关综述文献如^a，回答以下问题：

- 视觉与 IMU 进行融合之后有何优势？
- 有哪些常见的视觉 + IMU 融合方案？有没有工业界应用的例子？
- 在学术界，VIO 研究有哪些新进展？有没有将学习方法用到 VIO 中的例子？

你也可以对自己感兴趣的方向进行文献调研，阐述你的观点。

^aJianjun Gui et al. "A review of visual inertial odometry from filtering and optimisation perspectives". In: *Advanced Robotics* 29.20 (2015), 1289–1301. ISSN: 0169-1864. DOI: {10.1080/01691864.2015.1057616}.

第二题

2. 四元数和李代数更新

课件提到了可以使用四元数或旋转矩阵存储旋转变量。当我们用计算出来的 ω 对某旋转更新时，有两种不同方式：

$$\begin{aligned} \mathbf{R} &\leftarrow \mathbf{R} \exp(\omega^\wedge) \\ \text{或 } \mathbf{q} &\leftarrow \mathbf{q} \otimes \left[1, \frac{1}{2}\omega\right]^\top \end{aligned} \quad (20)$$

请编程验证对于小量 $\omega = [0.01, 0.02, 0.03]^\top$ ，两种方法得到的结果非常接近，实践当中可视为等同。因此，在后文提到旋转时，我们并不刻意区分旋转本身是 \mathbf{q} 还是 \mathbf{R} ，也不区分其更新方式为上式的哪一种。

第二题

```
4 int main() {
5     // 0. 构造一个旋转向量表示某个旋转, 作为更新前的旋转量
6     TAng3 v(M_PI/3, TVec3(1, 0, 0)); // 绕x轴旋转60度
7
8     // 1. 将旋转向量分别以旋转矩阵和四元数的形式表示
9     TMat3 R = v.toRotationMatrix();
10    TQuat q(v);
11
12    // 2. 构造更新量 w, 并表示为旋转矩阵和四元数的形式
13    TVec3 w(0.01, 0.02, 0.03);
14    TMat3 delta_R = Rodrigues(w); // 定义在头文件 // TODO: Sophus or Eigen vec3-ang3-rot3
15    TQuat delta_q = VecToQuat(w); // 定义在头文件 // TODO: [1, 0.5*w]
16
17    // 3. 分别更新旋转矩阵和四元数
18    TMat3 updated_R = R * delta_R;
19    TQuat updated_q = q * delta_q;
20    updated_q.normalize(); // 四元数归一化处理
21
22    // 4. 将四元数表示为旋转矩阵, 并比较差异
23    // 4-1 矩阵作差 ---- R - R_q
24    TMat3 R1 = updated_R - updated_q.toRotationMatrix();
25    // 4-2 矩阵相乘 ---- Rt * R_q
26    TMat3 R2 = updated_R.transpose() * updated_q.toRotationMatrix();
27
28    // 5. 输出结果
29    // 若两种更新方式等价, 则 R1 应近似为零矩阵, R2 应近似为单位阵
30    std::cout << "origin_R <<" << "\n" << R << "\n";
31    std::cout << "origin_R_q <<" << "\n" << q.toRotationMatrix() << "\n";
32    std::cout << "-----" << "\n";
33    std::cout << "R1 = R - R_q" << "\n" << "R1 <<" << "\n" << R1 << "\n";
34    std::cout << "-----" << "\n";
35    std::cout << "R2 = Rt * R_q" << "\n" << "R2 <<" << "\n" << R2 << "\n";
36 }
```

- 1、构造旋转
- 2、构造更新量
- 3、更新
- 4、结果对比

第二题

```
8  #define TVec3 Eigen::Vector3d
9  #define TQuat Eigen::Quaterniond
10 #define TMat3 Eigen::Matrix3d
11 #define TAng3 Eigen::AngleAxisd
12
13 // 罗德里格斯公式
14 TMat3 Rodrigues(const TVec3 &v) {
15     double theta = v.norm();
16
17     TVec3 nvec = v / theta;
18     TMat3 nHat;
19     nHat << 0.0, -nvec[2], nvec[1], nvec[2], 0.0, -nvec[0], -nvec[1], nvec[0], 0.0;
20     TMat3 R = cos(theta) * TMat3::Identity() + (1.0 - cos(theta)) * nvec * nvec.transpose() + sin(theta) * nHat;
21
22     return R;
23 }
24
25 // 轴角转四元数
26 TQuat VecToQuat(const TVec3 &v) {
27     TQuat q;
28
29     double theta = v.norm();
30     q.w() = cos(0.5 * theta);
31     q.vec() = (sin(0.5 * theta) / theta) * v;
32
33     return q;
34 }
```

第二题

1、作差的形式: $R1 = \text{updated_R} - \text{updated_R_q}$

2、相乘的形式: $R2 = \text{updated_R}$ 的转置 * updated_R_q

若两种更新方式等价, 则 $R1$ 应近似为零矩阵, $R2$ 应近似为单位阵。

```
● lcx@lcx:~/Desktop/vio_homework/ch1/ch1_code$ ./bin/rotaton
origin_R <<
      1      0      0
      0      0.5 -0.866025
      0 0.866025      0.5
origin_R_q <<
      1      0      0
      0      0.5 -0.866025
      0 0.866025      0.5
-----
R1 = R - R_q
R1 <<
      0 -3.46945e-18 -3.46945e-18
      0  5.55112e-17      0
-3.46945e-18 -1.11022e-16      0
-----
R2 = Rt * R_q
R2 <<
      1      0 -1.73472e-18
-1.73472e-18      1      0
-3.46945e-18 1.11022e-16      1
```

第三题

3. 其他导数

使用右乘 $so(3)$, 推导以下导数:

$$\frac{d(\mathbf{R}^{-1}\mathbf{p})}{d\mathbf{R}} \quad (21)$$

$$\frac{d \ln \left(\mathbf{R}_1 \mathbf{R}_2^{-1} \right)^{\vee}}{d\mathbf{R}_2} \quad (22)$$

第三题

$$\begin{aligned}\frac{d(R^{-1}p)}{dR} &= \lim_{\varphi \rightarrow 0} \frac{(\text{Rexp}(\varphi^\wedge))^{-1}p - R^{-1}p}{\varphi} \\ &= \lim_{\varphi \rightarrow 0} \frac{\exp(-\varphi^\wedge)R^{-1}p - R^{-1}p}{\varphi} \\ &= \lim_{\varphi \rightarrow 0} \frac{(I - \varphi^\wedge)R^{-1}p - R^{-1}p}{\varphi} \\ &= \lim_{\varphi \rightarrow 0} \frac{-\varphi^\wedge R^{-1}p}{\varphi} \\ &= \lim_{\varphi \rightarrow 0} \frac{(R^{-1}p)^\wedge \varphi}{\varphi} \\ &= (R^{-1}p)^\wedge\end{aligned}$$

第三题

$$\begin{aligned} & \frac{d \ln(\mathbf{R}_1 \mathbf{R}_2^{-1})^\vee}{d \mathbf{R}_2} \\ &= \lim_{\phi \rightarrow 0} \frac{\ln(\mathbf{R}_1 (\mathbf{R}_2 \exp(\phi^\wedge))^{-1})^\vee - \ln(\mathbf{R}_1 \mathbf{R}_2^{-1})^\vee}{\phi} \\ &= \lim_{\phi \rightarrow 0} \frac{\ln(\mathbf{R}_1 (\exp(\phi^\wedge))^{-1} \mathbf{R}_2^{-1})^\vee - \ln(\mathbf{R}_1 \mathbf{R}_2^{-1})^\vee}{\phi} \end{aligned}$$

旋转矩阵有: $\mathbf{R}^T = \mathbf{R}^{-1}$, $\mathbf{R} \exp(\phi^\wedge) \mathbf{R}^T = \exp((\mathbf{R} \phi)^\wedge)$

$$\begin{aligned} &= \lim_{\phi \rightarrow 0} \frac{\ln(\mathbf{R}_1 \mathbf{R}_2^{-1} \mathbf{R}_2 \exp(-\phi^\wedge) \mathbf{R}_2^T)^\vee - \ln(\mathbf{R}_1 \mathbf{R}_2^{-1})^\vee}{\phi} \\ &= \lim_{\phi \rightarrow 0} \frac{\ln(\mathbf{R}_1 \mathbf{R}_2^{-1} \exp((- \mathbf{R}_2 \phi)^\wedge))^\vee - \ln(\mathbf{R}_1 \mathbf{R}_2^{-1})^\vee}{\phi} \end{aligned}$$

BCH 线性近似表达为:

$$\ln((\exp(\phi_1^\wedge) \exp(\phi_2^\wedge))^\vee) = \begin{cases} \mathbf{J}_l(\phi_2)^{-1} \phi_1 + \phi_2 & \text{当 } \phi_1 \text{ 为小量} \\ \mathbf{J}_r(\phi_1)^{-1} \phi_2 + \phi_1 & \text{当 } \phi_2 \text{ 为小量} \end{cases}$$

以及:

$$\begin{aligned} \ln(\mathbf{R} \exp(\phi^\wedge))^\vee &= \ln(\mathbf{R})^\vee + \mathbf{J}_r^{-1}(\ln(\mathbf{R})^\vee) \phi \\ &= \lim_{\phi \rightarrow 0} \frac{\ln(\mathbf{R}_1 \mathbf{R}_2^{-1})^\vee + \mathbf{J}_r^{-1}(\ln(\mathbf{R}_1 \mathbf{R}_2^{-1})^\vee) (-\mathbf{R}_2 \phi) - \ln(\mathbf{R}_1 \mathbf{R}_2^{-1})^\vee}{\phi} \\ &= -\mathbf{J}_r^{-1}(\ln(\mathbf{R}_1 \mathbf{R}_2^{-1})^\vee) \mathbf{R}_2 \end{aligned}$$



感谢各位聆听 !
Thanks for Listening



