深蓝学院
shenlanxueyuan.com

视觉SLAM进阶：从零开始手写VIO

第一章：概述与课程介绍
作业讲解

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。

[a] Jianjun 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}.

深蓝学院
shenlanxueyuan.com

1.视觉和IMU进行融合之后有何优势?

视觉和IMU可以相互弥补单个传感器感知定位的不足，从而形成一个更加鲁棒、精准的系统。

视觉定位

优点：1.可以获得较为稠密的环境结构及纹理信息　2.不存在漂移　3. 结构小巧、成本相对低廉、便于装配

缺点：1.以单目相机为代表的视觉无法恢复米制的尺度信息　2. 光照变化、少纹理、运动模糊、摄像头遮挡等会影响跟踪效果

IMU定位

优点：1. 不依赖环境光照特征的变化，能准确计算短时间的运动　2.能给出绝对的米制尺度

缺点：1. 存在噪声和漂移，低速状态下也会很快发散　2. 长时间积分有很大的累计误差

综上，视觉比较依赖环境的信息能很好地感知环境的特征，但在跟踪过程中会尺度丢失、短距离定位不鲁棒；

IMU能很好地计算短距离快速运动，但存在累计误差；两个传感器结合的VIO系统能弥补各自的不足，各取所长形成一个更加鲁棒精准的定位模块。

2.有哪些常见的视觉+IMU融合方案？有没有工业界应用的例子？

深蓝学院
shenlanxueyuan.com

•**松耦合；**IMU、视觉是两个独立的模块，通常使用EKF进行融合。IMU进行预测，视觉用来更新

•**紧耦合；**方法上可以分为基于滤波(filtering-based) 和 基于优化(optimisation-based) 的方法。

1. 滤波框架；MSCKF，SR-ISWF，ROVIO

2.优化框架；VINS-Mono，VINS-Fusion，ORB3，VIORB

VIO框架	耦合方案	后端方案	前端	视觉误差	初始化	回环	精度	效率
MSF	松耦合	滤波EKF	\	\	\	\	最差1	最快5
MSCKF	紧耦合	滤波EKF	fast+光流	重投影	静止	无	较差2	较快4
ROVIO	紧耦合	滤波IEKF	fast+光度	光度	静止	无	一般3	较快4
OKVIS	紧耦合	优化	Harris+BRISK	重投影	静止	无	较好4	最慢1
VINS	紧耦合	优化	Harris+光流	重投影	动态	有	最好5	较慢2
VI-ORB	紧耦合	优化	orb	重投影	动态	有	\	\
ICE-BA	紧耦合	优化	Fast+光流	重投影	静止	无	\	\

2.有哪些常见的视觉+IMU融合方案？有没有工业界应用的例子？

深蓝学院
shenlanxueyuan.com

工业界目前应用较多的领域有：机器人导航、自动驾驶、AR、VR等，其中在自动驾驶中应用较多的还是偏低速的园区物流车，地下车库 AVP等场景；AR、VR用VINS较多。

3.在学术界，VIO研究有哪些新进展？有没有将学习方法用到VIO的例子？

深蓝学院
shenlanxueyuan.com

[1] 基于多平面先验的高效VIO，Jinyu Li, Bangbang Yang, Kai Huang, Guofeng Zhang, and Hujun Bao*. Robust and Efficient Visual-Inertial Odometry with Multi-plane Priors.

[2]基于人工规则结构的VIO，StructVIO : Visual-Inertial Odometry with Structural Regularity of Man-Made Environments

[3] 低纹理顺滑梯度下的VIO:稠密直接滤波法 VIO Uncertainty-Based Adaptive Sensor Fusion for Visual-Inertial Odometry under Various Motion Characteristics Monocular Visual-Inertial Odometry in Low-Textured Environments with Smooth Gradients: A Fully Dense Direct Filtering Approach

应用学习方法：

[1] Lee, Hongyun, Matthew McCrink, and James W. Gregory. “Visual-Inertial Odometry for Unmanned Aerial Vehicle using Deep Learning.” AIAA Scitech 2019 Forum. 2019.

[2] Wang, Chengze, Yuan Yuan, and Qi Wang. “Learning by Inertia: Self-supervised Monocular Visual Odometry for Road Vehicles.” ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2019.

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：

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

请编程验证对于小量 $\omega = [0.01, 0.02, 0.03]^{\mathrm{T}}$，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。

思路

- 旋转矩阵：涉及到李代数与李群的转换，推荐使用Sophus库，当然使用Eigen 把旋转向量转化为矩阵乘法来更新也是可以的
- 四元数：利用Eigen::Quaterniond 对 * 运算符的重载 更新后需要归一化

参考代码

```cpp
#include <iostream>
#include <eigen3/Eigen/Dense>
#include <eigen3/Eigen/Geometry>
#include <sophus/so3.h>

int main(int argc, char *argv[])
{
    Eigen::Matrix3d m_rotation = Eigen::AngleAxisd(M_PI/2, Eigen::Vector3d(0,0,1)).toRotationMatrix();
    Eigen::Quaterniond q_rotation(m_rotation);
    Eigen::Vector3d w(0.01,0.02,0.03);

    // Updated by Matrix
    Sophus::SO3 SO3_rotation(m_rotation);
    Sophus::SO3 SO3_rotation_updated = SO3_rotation * Sophus::SO3::exp(w);

    // updated by Quaternion
    Eigen::Quaterniond q_w(1, w[0]/2 ,w[1]/2, w[2]/2);
    Eigen::Quaterniond q_rotation_update = (q_rotation * q_w).normalized(); // need normalized

    // output
    std::cout << "Origin Rotation: " << std::endl;
    std::cout << m_rotation << std::endl;

    std::cout << "Rotation updated by matrix: " << std::endl;
    std::cout << SO3_rotation_updated.matrix() << std::endl;

    std::cout << "Rotation updated by Quaternion: " << std::endl;
    std::cout << q_rotation_update.toRotationMatrix() << std::endl;

    return 0;
}
```

深蓝学院
shenlanxueyuan.com

运行结果

```
Origin Rotation:
6.12323e-17            -1            0
          1 6.12323e-17            0
          0            0            1
Rotation update by matrix:
 -0.030093     -0.9995  0.0096977
   0.99935   -0.029893  0.0201453
-0.0198454   0.0102976    0.99975
Rotation update by Quaternion:
-0.0300895     -0.9995 0.00969661
   0.99935 -0.0298895   0.0201429
-0.0198431   0.0102964    0.99975
```

结论

上述结果可以看出，通过四元数和旋转矩阵两种方式对旋转量进行更新，结果几乎相同，可以认为两种方式是等价的。

深蓝学院
shenlanxueyuan.com

3. 其他导数

使用右乘 $\mathfrak{so}(3)$，推导以下导数：

$$\frac{\mathrm{d}\left(\mathbf{R}^{-1}\mathbf{p}\right)}{\mathrm{d}\mathbf{R}} \tag{21}$$

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

推导思路：对待求导变量右乘一个扰动，再结合泰勒展开、伴随性质、BCH进行推导

注意点：

1. 扰动变量右乘在逆运算符里面
2. 注意第二题是对R2求导，所以保留R1

深蓝学院
shenlanxueyuan.com

(1) $\frac{d(R^{-1}p)}{dR}$

$$\frac{d(R^{-1}p)}{dR} = \lim_{\phi \to 0} \frac{(R \cdot \exp(\phi^\wedge))^{-1} \cdot p - R^{-1}p}{\phi}$$

$(AB)^{-1} = B^{-1}A^{-1}$

$$= \lim_{\phi \to 0} \frac{(\exp(\phi^\wedge))^{-1} R^{-1} \cdot p - R^{-1}p}{\phi}$$

$$= \lim_{\phi \to 0} \frac{\exp(-\phi^\wedge) R^{-1}p - R^{-1}p}{\phi}$$

$\exp(-\phi^\wedge) \approx I - \phi^\wedge$

$$= \lim_{\phi \to 0} \frac{(I - \phi^\wedge) R^{-1}p - R^{-1}p}{\phi}$$

$$= \lim_{\phi \to 0} \frac{-\phi^\wedge R^{-1}p}{\phi}$$

$a^\wedge b = -b^\wedge a$

$$= \lim_{\phi \to 0} \frac{(R^{-1}p)^\wedge \phi}{\phi} = (R^{-1}p)^\wedge$$

深蓝学院
shenlanxueyuan.com

(2) $\dfrac{d(\ln(R_1R_2^{-1})^\vee)}{dR_2}$

$$= \lim_{\phi\to0} \frac{\ln(R_1(R_2\cdot\exp(\phi^\wedge))^{-1})^\vee - \ln(R_1R_2^{-1})^\vee}{\phi}$$

$(R_2\exp(\phi^\wedge))^{-1} = \exp(\phi^\wedge)^{-1}R_2^{-1}$

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

$$= \lim_{\phi\to0} \frac{\ln(R_1\exp(-\phi^\wedge)\cdot R_2^{-1})^\vee - \ln(R_1R_2^{-1})^\vee}{\phi}$$

伴随性质，因为是对R_2导，保留R_1

$R_1\exp(-\phi^\wedge)R_1^T = \exp(-(R_1\phi)^\wedge)$

$$= \lim_{\phi\to0} \frac{\ln(\exp(-(R_1\phi)^\wedge)\cdot R_1\cdot R_2^{-1})^\vee - \ln(R_1R_2^{-1})^\vee}{\phi}$$

BCH 公式：$\ln(\exp(\xi^\wedge)R)^\vee = \ln(R)^\vee + J_l^{-1}(\ln R^\vee)\cdot\xi$

$$= \lim_{\phi\to0} \frac{\cancel{\ln(R_1R_2^{-1})^\vee} + J_l^{-1}\cdot[-R_1\cancel{\phi}] - \cancel{\ln(R_1R_2^{-1})^\vee}}{\cancel{\phi}}$$

$$= -J_l^{-1}\cdot R_1 = -J_l^{-1}(\ln(R_1\cdot R_2^{-1})^\vee)\cdot R_1.$$