



深蓝学院
shenlanxueyuan.com

从零开始手写VIO课程 第一次作业讲评



主讲人 姜新禹



第一题

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}.

第一题

- 1.1 优秀回答 IMU 的特点是存在零偏，使得积分后会产生很大漂移，但是对于短时间内的快速运动可以作出较好的估计。而视觉 SLAM 下，运动过快会产生图像模糊，图像无法匹配，同时也会受到纹理、光照、动态物体干扰等影响，单目情况下更会有尺度不确定性和纯旋转无法估计的问题。

视觉与 IMU 进行融合之后，两者互补，可以有效解决上面提到的问题，因此融合之后的优势主要表现在于以下几个方面：

- 1) 纯视觉 SLAM 容易收到纹理/光照等干扰的影响，此时可以依靠 IMU 进行定位。
- 2) 面临动态障碍物时，视觉 SLAM 单单从图像变化，本质上无法区分是相机自身运动还是外界条件发生变化。融合后可以由 IMU 可以感受到自己的运动信息，减小动态物体的影响。
- 3) 单目与 IMU 的融合，解决了单目尺度不确定性，融合 IMU 可以恢复尺度。
- 4) 单目与 IMU 融合可以解决单目下，纯旋转运动（如快速转弯时）无法估计的问题。
- 5) 视觉与 IMU 融合可解决，视觉 SLAM 下快速运动时图像可能模糊产生丢失的问题。
- 6) 视觉与 IMU 融合，可以借助 IMU 的采样频率，提高系统的输出频率。
- 7) 视觉与 IMU 融合可以消除 IMU 的暂时积分漂移。
- 8) 另外，激光测距太重（无人机上不合适用）、红外传感器对太阳光敏感、声呐测量距离受限，双目相机极限距离太短。相比之下，用视觉与 IMU 融合轻便小巧限制较少。

第一题

● 1.2 优秀回答

MSCKF(https://github.com/KumarRobotics/msckf_vio.git),
(<https://github.com/yuzhou42/MSCKF.git> 中文注释版),
MSCKF_mono(https://github.com/daniilidis-group/msckf_mono.git),
OKVIS(https://github.com/ethz-asl/okvis_ros.git),
ROVIO(<https://github.com/ethz-asl/rovio.git>),
VIORB(<https://github.com/jingpang/LearnVIORB.git>),
VINS-Mono(<https://github.com/HKUST-Aerial-Robotics/VINS-Mono.git>),
VINS-Mobile(<https://github.com/HKUST-Aerial-Robotics/VINS-Mobile.git>),
VINS-Fusion(<https://github.com/HKUST-Aerial-Robotics/VINS-Fusion.git>)

松耦合：SSF、MSF

紧耦合：

基于滤波：MSCKF、ROVIO

基于优化：OKVIS、VIORB、VINS-Mono、VINS-Mobile、VINS-Fusion

第一题

● 1.2 优秀回答

常见视觉+IMU 融合方案：

1. MSCKF，紧耦合，FAST+光流前端，后端用 EKF 滤波，误差采用重投影误差，无回环检测。
2. ROVIO，紧耦合，基于稀疏图像块的 EKF 滤波实现的 VIO，是基于单目开发的。前端使用 fast+光度，后端 IEKF 滤波，无回环检测。
3. VINS Mono，紧耦合，前端使用 Harris+光流，有回环检测。通过非线性优化器优化一个滑动窗口内的关键帧，帧间通过鲁棒的角点关联。系统初始阶段，通过松耦合方式耦合多种传感器；重定位阶段通过紧耦合方式融合传感器，优化前通过 IMU 预积分减少计算量；提供了基于 4DoF 的位姿图优化和回环检测。
4. OKVIS，紧耦合，前端使用多尺度 Harris 提取特征点，并用 BRISK 作为描述子，后端利用 Ceres 通过非线性优化完成状态估计。视觉误差是重投影，无回环检测；它利用非线性优化一个滑窗内的关键帧，其损失函数包括带权重的投影差和带权重的惯导误差。
5. ORBSLAM2+IMU 紧耦合，ORB 稀疏前端、图优化后端、带闭环检测和重定位。

第一题

● 1.3

参考: <https://www.zhihu.com/people/li-wen-qiang-96-17/posts>

Struct SLAM:

除了使用点特征外, 结合线, 曲线, 平面等结构特征来提高定位精度并提供更多的环境几何结构信息。

基于深度学习的SLAM:

具体研究方向可以分为三块:

- 用深度学习方法替换传统SLAM中的一个/几个模块。
- 在传统SLAM之上加入语义信息。
- 端到端的SLAM。

第二题

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：

$$\begin{aligned} \mathbf{R} &\leftarrow \mathbf{R} \exp(\omega^\wedge) \\ \text{或 } \mathbf{q} &\leftarrow \mathbf{q} \otimes \left[1, \frac{1}{2}\omega\right]^T \end{aligned} \quad (18)$$

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

1. 旋转矩阵更新时可以使用Sophus(或Eigen)中exp，或者Eigen写出罗德里格斯公式
2. 注意最后 \mathbf{q} 需要归一化，以及四元数旋转小量之间的0.5倍关系

第二题

```
#include <iostream>
#include <Eigen/Core>
#include <Eigen/Geometry>
#include <sophus/so3.h>
using namespace std;

int main(int argc, char** argv)
{
    // 1. 设置随机 q 和 R
    srand((unsigned int) time(0));
    int R_theta = rand()%360;
    Eigen::Vector3d R_axis = Eigen::Vector3d::Random().normalized();
    Eigen::AngleAxisd R_aa(M_PI*R_theta/180.0, R_axis);
    Eigen::Matrix3d R = R_aa.toRotationMatrix();
    Eigen::Quaterniond q(R);

    Eigen::Vector3d w(0.01, 0.02, 0.03);
    // 2. 计算 dR, 并更新 R
    // 2.1 使用sophus的exp
    Sophus::SO3 S03_R(R);
    Sophus::SO3 S03_dR = Sophus::SO3::exp(w);
    Sophus::SO3 S03_updated = S03_R * S03_dR;
    Eigen::Matrix3d R_S03_updated = S03_updated.matrix();
    std::cout << "R_S03_updated:\n" << R_S03_updated << std::endl;
```

```
quanmeixiang@BJ-DZ0102968:~/work/update/build$ ./update
R S03 updated:
0.0690548 -0.14914 -0.986402
0.570197 -0.805438 0.161697
-0.818601 -0.573609 0.0294198
R Rodrigues updated:
0.0690548 -0.14914 -0.986402
0.570197 -0.805438 0.161697
-0.818601 -0.573609 0.0294198
R Quaterniond updated1:
0.0690531 -0.149139 -0.986402
0.5702 -0.805436 0.161694
-0.818598 -0.573612 0.029421
R Quaterniond updated2:
0.0690531 -0.149139 -0.986402
0.5702 -0.805436 0.161694
-0.818598 -0.573612 0.029421
R diff:
1.77924e-06 -1.3922e-06 3.35051e-07
-3.19565e-06 -1.80667e-06 2.26966e-06
-2.07585e-06 2.89881e-06 -1.24059e-06
```


第二题

```
// 2.2 使用eigen实现罗德里格斯公式来更新R
double theta = w.norm();
Eigen::Vector3d w_axis = w / theta;
Eigen::Matrix3d w_hat;
w_hat << 0, -w_axis(2), w_axis(1),
        w_axis(2), 0, -w_axis(0),
        -w_axis(1), w_axis(0), 0;
Eigen::Matrix3d I3x3 = Eigen::Matrix3d::Identity();
Eigen::Matrix3d Rodrigues_dR = cos(theta)*I3x3 + (1-cos(theta))*w_axis*w_axis.transpose()
                                + sin(theta)*w_hat;
Eigen::Matrix3d R_Rodrigues_updated = R * Rodrigues_dR;
std::cout << "R_Rodrigues_updated:\n" << R_Rodrigues_updated << std::endl;

//3. 计算 dq, 并更新 q
Eigen::Quaterniond dq(1, w(0)/2, w(1)/2, w(2)/2);
Eigen::Quaterniond q_updated1 = (q * dq).normalized();
dq.normalize();
Eigen::Quaterniond q_updated2 = q * dq;
std::cout << "R_Quaterniond_updated1:\n" << q_updated1.toRotationMatrix() << endl;
std::cout << "R_Quaterniond_updated2:\n" << q_updated2.toRotationMatrix() << endl;

//4. 计算exp和四元数更新R的差值
Eigen::Matrix3d R_diff = R_S03_updated - q_updated1.toRotationMatrix();
std::cout << "R_diff:\n" << R_diff << std::endl;

return 0;
```

第三题

使用右乘 $\mathfrak{so}(3)$ ，推导以下导数：

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

$$\frac{d \ln(\mathbf{R}_1 \mathbf{R}_2^{-1})}{d\mathbf{R}_2}$$

- 公式规范，**粗体**
- 右扰动： $\mathbf{R}^{-1} \rightarrow (\mathbf{R} \exp(\phi^\wedge))^{-1} = \exp(\phi^\wedge)^{-1} \mathbf{R}^{-1}$
- 叉乘交换： $\mathbf{a} \times \mathbf{b} = -\mathbf{b} \times \mathbf{a}$
- 近似： $\exp(\phi^\wedge)^{-1} = \exp(-\phi^\wedge) \approx \mathbf{I} - \phi^\wedge$
- 近似： $\ln(\mathbf{R} \exp(\phi^\wedge))^\vee \approx \ln(\mathbf{R})^\vee + \mathbf{J}_r^{-1}(\ln(\mathbf{R})^\vee)\phi$
- 近似： $\ln(\exp(\phi^\wedge)\mathbf{R})^\vee \approx \mathbf{J}_l^{-1}(\ln(\mathbf{R})^\vee)\phi + \ln(\mathbf{R})^\vee$
- 伴随： $\mathbf{R} \exp(\phi^\wedge) \mathbf{R}^T = \exp((\mathbf{R}\phi)^\wedge)$

第三题

● 对不同类型函数的右雅可比定义

	Functions from vector space to vector space	Functions from vector space to SO(3)
定义:	$\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} \triangleq \lim_{\delta \mathbf{x} \rightarrow 0} \frac{f(\mathbf{x} + \delta \mathbf{x}) - f(\mathbf{x})}{\delta \mathbf{x}}$	$\begin{aligned} \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} &\triangleq \lim_{\delta \mathbf{x} \rightarrow 0} \frac{f(\mathbf{x} + \delta \mathbf{x}) \ominus f(\mathbf{x})}{\delta \mathbf{x}} \\ &= \lim_{\delta \mathbf{x} \rightarrow 0} \frac{\text{Log}(f^{-1}(\mathbf{x})f(\mathbf{x} + \delta \mathbf{x}))}{\delta \mathbf{x}} \end{aligned}$
	Functions from SO(3) to vector space	Functions from SO(3) to SO(3)
定义:	$\begin{aligned} \frac{\partial f(\mathbf{R})}{\partial \boldsymbol{\theta}} &\triangleq \lim_{\delta \boldsymbol{\theta} \rightarrow 0} \frac{f(\mathbf{R} \oplus \delta \boldsymbol{\theta}) - f(\mathbf{R})}{\delta \boldsymbol{\theta}} \\ &= \lim_{\delta \boldsymbol{\theta} \rightarrow 0} \frac{f(\mathbf{R} \text{Exp}(\delta \boldsymbol{\theta})) - f(\mathbf{R})}{\delta \boldsymbol{\theta}} \end{aligned}$	$\begin{aligned} \frac{\partial f(\mathbf{R})}{\partial \boldsymbol{\theta}} &\triangleq \lim_{\delta \boldsymbol{\theta} \rightarrow 0} \frac{f(\mathbf{R} \oplus \delta \boldsymbol{\theta}) \ominus f(\mathbf{R})}{\delta \boldsymbol{\theta}} \\ &= \lim_{\delta \boldsymbol{\theta} \rightarrow 0} \frac{\text{Log}(f^{-1}(\mathbf{R})f(\mathbf{R} \text{Exp}(\delta \boldsymbol{\theta})))}{\delta \boldsymbol{\theta}} \end{aligned}$

第三题

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

$$\begin{aligned}\frac{\partial(\exp(\phi^\wedge)p)}{\partial \phi} &= \lim_{\delta \phi \rightarrow 0} \frac{[\exp((\phi + \delta \phi)^\wedge)]^{-1} p - [\exp(\phi^\wedge)]^{-1} p}{\delta \phi} \\&= \lim_{\delta \phi \rightarrow 0} \frac{[\exp(\phi^\wedge) \exp((J_r \delta \phi)^\wedge)]^{-1} p - [\exp(\phi^\wedge)]^{-1} p}{\delta \phi} \\&= \lim_{\delta \phi \rightarrow 0} \frac{[\exp((J_r \delta \phi)^\wedge)]^{-1} [\exp(\phi^\wedge)]^{-1} p - [\exp(\phi^\wedge)]^{-1} p}{\delta \phi} \\&\approx \lim_{\delta \phi \rightarrow 0} \frac{(I - (J_r \delta \phi)^\wedge) \exp(-\phi^\wedge) p - \exp(-\phi^\wedge) p}{\delta \phi} \\&= \lim_{\delta \phi \rightarrow 0} \frac{(-J_r \delta \phi)^\wedge \exp(-\phi^\wedge) p}{\delta \phi} \\&= \lim_{\delta \phi \rightarrow 0} \frac{[\exp(-\phi^\wedge) p]^\wedge (J_r \delta \phi)}{\delta \phi} \\&= (R^{-1} p)^\wedge J_r\end{aligned}$$

第三题

$$\begin{aligned}\frac{d\ln(R_1 R_2^{-1})^\vee}{dR_2} &= \lim_{\psi \rightarrow 0} \frac{\ln(R_1 (R_2 \exp(\psi^\wedge))^{-1})^\vee - \ln(R_1 R_2^{-1})^\vee}{\psi} \\&= \lim_{\psi \rightarrow 0} \frac{\ln(R_1 \exp(-\psi^\wedge) R_2^{-1})^\vee - \ln(R_1 R_2^{-1})^\vee}{\psi} \\&= \lim_{\psi \rightarrow 0} \frac{\ln(R_1 R_2^{-1} R_2 \exp(-\psi^\wedge) R_2^{-1})^\vee - \ln(R_1 R_2^{-1})^\vee}{\psi} \\&= \lim_{\psi \rightarrow 0} \frac{\ln(R_1 R_2^{-1} \exp(-R_2 \psi^\wedge))^\vee - \ln(R_1 R_2^{-1})^\vee}{\psi} \\&= \lim_{\psi \rightarrow 0} \frac{\ln(R_1 R_2^{-1})^\vee - J_r^{-1} R_2 \psi - \ln(R_1 R_2^{-1})^\vee}{\psi} \\&= \lim_{\psi \rightarrow 0} \frac{-J_r^{-1} R_2 \psi}{\psi} \\&= -J_r^{-1} (\ln(R_1 R_2^{-1})^\vee) \cdot R_2\end{aligned}$$

$$= -\mathbf{J}_l^{-1} (\ln(\mathbf{R}_1 \mathbf{R}_2^{-1})^\vee) \mathbf{R}_1$$

感谢各位聆听 !
Thanks for Listening

