

1 视觉与IMU融合之后有何优势？

|  |  |  |
| --- | --- | --- |
|  | 视觉 | IMU |
| 静止状态 | 初始化后，数据稳定 | 通常会产生较大偏移 |
| 缓慢运动 | 特征点数量稳定状态下，数据输出稳定 | IMU可以准确计算出运动 |
| 快速运动 | 可能导致特征点丢失或者两帧之间重叠区域太少以至于无法进行特征匹配 | IMU估计较为准确 |

1. 当图像发生变化时，本质上我们没法知道是相机自身发生了运动，还是外界条件发生了变化，所以VSLAM 难以处理动态的障碍物。而IMU能够检测到自身运动信息，从某种程度上减轻动态物体的影响。
2. 对于单目视觉SLAM，存在尺度不确定性，融合IMU odmetry后可以恢复尺度。
3. 纯视觉SLAM在容易受弱纹理场景和光照变化的影响，在定位失败时，可以依靠IMU进行短暂的定位。

综上，Visual与IMU融合之后会弥补各自的劣势，可利用Visual来估计IMU的零偏，减少IMU由零偏导致的发散和累积误差。IMU可以为Visual提供快速运动时的定位，以及因为某种因素（场景特征点较少，光照变化较大等）定位失败时的状态估计。

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？

（1）常见的视觉+IMU融合方案

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> )

# [Visual-Odometry-Review](https://github.com/MichaelBeechan/Visual-Odometry-Review)

<https://github.com/MichaelBeechan/Visual-Odometry-Review>

（2）工业界应用

Google：Tango，ARCore

Apple：ARKit

Microsoft：HoloLens

百度：DuMix AR

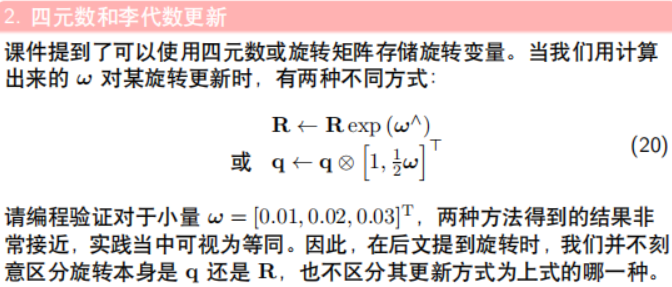
3 在学术界，VIO研究有哪些新进展？有没有将学习方法应用到VIO的例子？

Jianjun Gui, Dongbing Gu. A review of visual inertial odometry from filtering and optimization perspectives. [J] Advanced Robotics, 2015

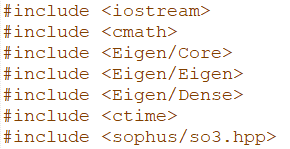
Chang Chen, Hua Zhu. A Review of Visual-Inertial Simultaneous Localization and Mapping from Filtering-Based and Optimization-Based Perspectives. [J] Robotics, 2018

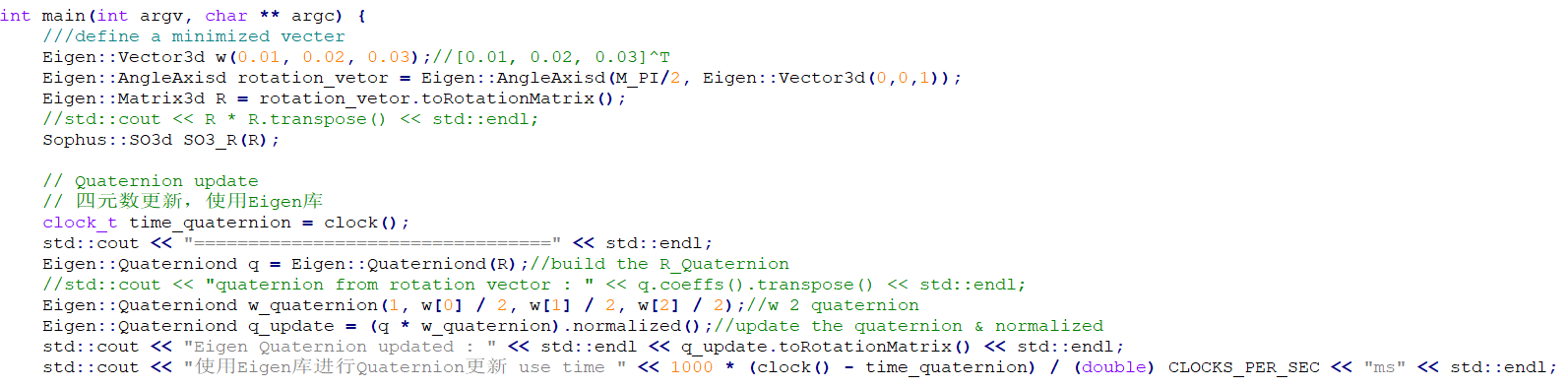
Chen, Changhao, et al. "Selective Sensor Fusion for Neural Visual-Inertial Odometry." arXiv preprint arXiv:1903.01534 (2019).

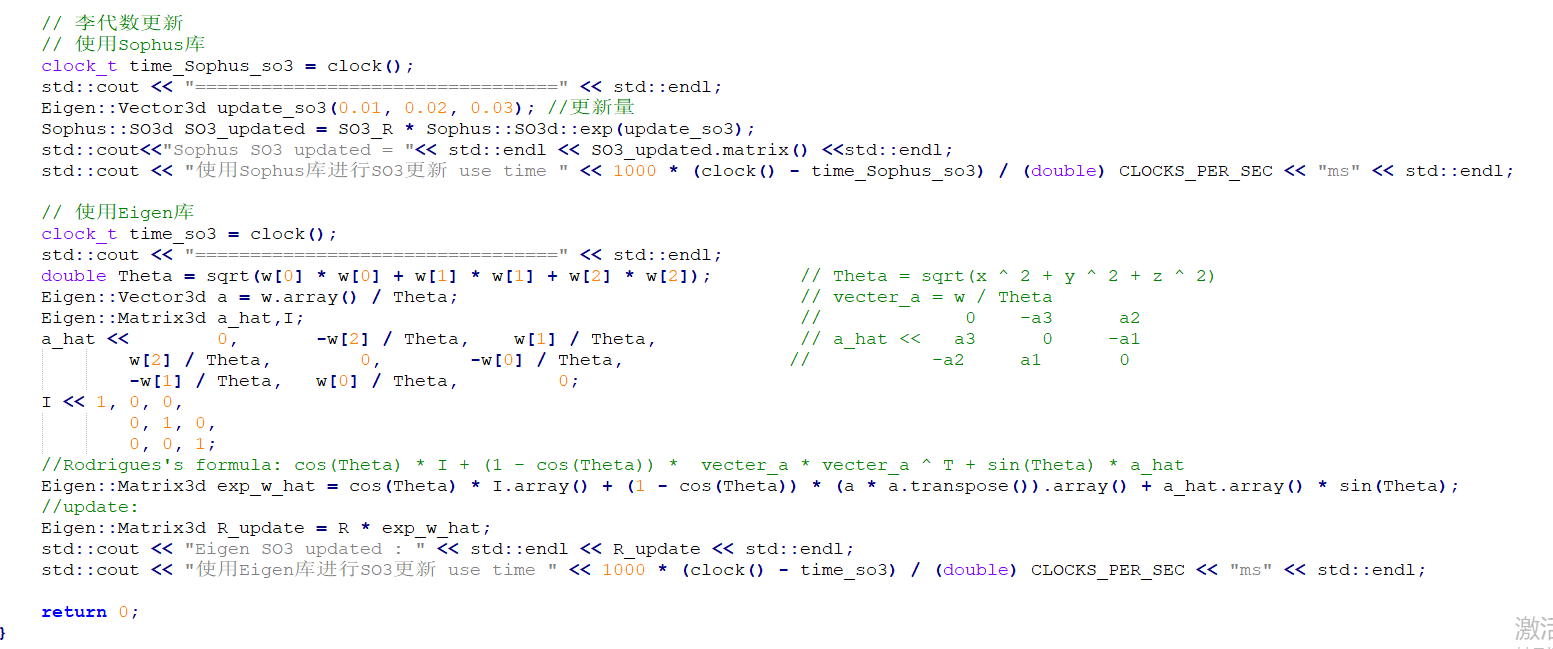
Lee, Hongyun, Matthew McCrink, and James W. Gregory. "Visual-Inertial Odometry for Unmanned Aerial Vehicle using Deep Learning." AIAA Scitech 2019 Forum. 2019.



代码如下，库文件引用：







结果可以看出，在小量的前提下，不管用旋转矩阵或者四元数进行更新差别不大，且四元数在Eigen库中的实现可将速度提升5倍左右。

