基于 ORB-SLAM 的视觉惯性单目 SLAM 的复现

庄礼聪

摘要

近年来，在视觉惯性里程计领域的研究取得了许多优秀的成果，致力于精确且鲁棒的计算传感器的增量运动。然而这些里程计方法缺乏闭环和估计累积漂移的能力，即使不断地经过同一个地方。在文章 Visual-Inertial Monocular SLAM With Map Reuse 中，提出了紧耦合的视觉惯性 SLAM(Simultaneous Localization And Mapping) 系统，弥补了视觉惯性里程计无法闭环和地图复用的缺点。尽管该方法可以应用于所有的相机配置，但还是选择会带来尺度不确定性的且最普遍的单目相机配置，测试了由小型无人机采集的公开数据集的 11 个序列，在最好的结果中，尺度误差只有 1%。比较了当时最高水平的视觉惯性里程计，由于增加了地图复用功能，获得了更好的结果。本文根据上述文章的方法，基于 ORB-SLAM2 复现了文章中的内容。

关键词：视觉惯性 SLAM；ORB-SLAM

1 引言

在机器人和计算机视觉领域，传感器运动估计成为一个热门的话题，它为诸如自动驾驶汽车、虚拟显示、增强现实、服务机器人、无人机导航技术提供了基础技术支持。在许多传感器配置中，视觉惯性有着低廉的价格和巨大的潜力。一方面相机提供丰富的环境信息，可以用于建立 3D 模型、定位相机和重复场景检测。在另一方面，IMU(Inertial Measurement Unit) 提供了自身运动的信息，可以恢复单目相机的尺度、估计重力方向、提供绝对的俯仰和选择可观测性。

近几年来，视觉惯性融合的研究工作非常活跃，目前的研究工作主要聚焦于紧耦合的视觉惯性里程计，使用滤波器或者基于关键帧的非线性优化的方法。这些方法往往仅用于计算增量运动，而缺乏地图复用的能力。这意味着即使传感器一直在同一个区域内运动，误差也会无限制的累积。该文章实现了当时第一个基于关键帧的视觉惯性 SLAM 系统，能够实时地进行回环检测和地图复用，基于 ORB-SLAM，该系统的跟踪线程固定一个建好的地图优化当前帧，而后端执行本地窗口化的 BA(Bundle Adjustment)，联合窗口外的固定关键帧优化窗口内的关键帧。相比于其他优化，该系统的优化方法维持一个固定大小的窗口，来维持优化的规模，并且能够复用数据。该系统使用位置识别功能检测大量的回环，并用轻量级的位姿图优化来矫正回环，最后使用了一个单独的线性进行全局 BA 优化，而不会影响实时跟踪。

对该文章的复现工作，能够深入地学习视觉 SLAM 领域中的经典框架，对视觉 SLAM 中的数据处理、编程思路有了更进一步的学习，为以后的研究工作打下了坚实的基础。

2 相关工作

2.1 单目 SLAM

单目 SLAM 最初的解决方法是滤波器方法[1][2]，该方法将所有的帧通过滤波器综合处理估计出地图特征和相机位姿。该方法的缺点是在处理两个几乎没什么变换的帧时浪费大量的计算资源，而且会

累积大量的线性误差。另一方面，基于关键帧的方法[3][4]，该方法通过选择特定的帧估计地图和位姿，可以使用更准确的光速法平差优化来优化位姿。当建图不在受限于帧率，当在相同计算量的情况下，非线性优化方法比滤波器方法更加准确。

PTAM[4]是最具代表性的基于关键帧的 SLAM 系统，这是首次将跟踪和建图分布在不同线程这种想法引入 SLAM 的工作，成功的应用在了小环境中的实时 AR 中，PTAM 使用 FAST 角点和块匹配来建立地图点。这种方法建立的地图点在跟踪过程中很有效，但是在位置识别中效果一般，而且基于低分辨率缩略图进行重定位，视点的不变性很低。

Strasdat[5]等人研发了一个大尺度的单目 SLAM 系统，使用了基于 GPU 的光流法的前端，延续了 FAST 特征点匹配和仅位姿 BA 的方法，使用了基于滑动窗口的 BA 方法，利用了 7 自由度的位姿图优化来实现回环来消除单目 SLAM 中的累积误差。

2.2 视觉惯性 SLAM

视觉传感器与惯性传感器的融合在弱特征、运动模糊等情景下有着很好的鲁棒性，并且可以使单目 SLAM 系统恢复尺度。最早的紧耦合视觉惯性方法可以追溯到多状态约束的卡尔曼滤波器[6]，使用了特征边缘化的方法，避免了 EKF 由于特征数量造成的大量计算代价。第一个紧耦合的基于关键帧的视觉里程计使 OKVIS[7]，当这些系统都依赖于特征点匹配时，ROVIO[8]使用光度误差的直接法应用了 EKF。

VINS-Mono[9]是一个非常准确且鲁棒的单目惯性里程计系统具备了使用 DBoW2 进行回环检测的功能, 还是用了 4 自由度的位姿图优化和地图融合技术。使用 Lucas-Kanade 追踪器跟踪特征，相比于描述子匹配的跟踪方法有稍微更高的鲁棒性，后续的开发实现了双目和双目惯性的 SLAM 系统[10]。

VI-DSO[11]将 DSO 扩展成了视觉惯性里程计，提出了融合惯性观测和高梯度像素点光速误差 BA 的方法，当高梯度的像素信息被成功提取，在无纹理的环境下显著提升了鲁棒性，他们的初始化方法依赖于视觉惯性 BA，使用 20 秒到 30 秒的视觉恢复的尺度误差只有 1%。

2.3 位置识别

Williams[12]等人的综述比较了集中位置识别的方法，对基于外观匹配技术进行了总结，图像和图像匹配在较大的环境条件下比地图和地图匹配或者图像和地图匹配的效果要好。词袋技术例如概率性方法 FAB-MAP，因为其高效率所以处于领先地位。DBoW2 是一个使用 BRIEF 描述子的二进制词袋方法，使用了非常高效的 FAST 特征描述子。这种方法相对于 SURF 和 SIFT 特征在特征提取上减少了一个数量级的时间。尽管整个系统非常高效和鲁棒，但是 BRIEF 描述子在旋转和尺度方面没有较好的不变性，使系统在平面内的轨迹和相同视点中的回环检测受限。[13]基于使用 ORB 特征的 DBoW2 研发了一个非常高效且高不变性的位置识别模块，在多个数据集中都实现了高回召率和高鲁棒性。

3 本文方法

3.1 本文方法概述

本文基于单目配置的 ORB-SLAM，融合了 IMU 传感器，研发了一个视觉惯性 SLAM 系统，在原有的系统基础上，增加了 IMU 预积分、IMU 初始化、视觉关系联合 BA 等功能。在 ORB-SLAM 系统上的三大线程均进行了修改，在跟踪线程上，对每一帧图像帧，读取从上一帧到当前帧的 IMU 数据，

对其进行预积分，计算出上一帧到当前帧的位姿变化、速度变化、IMU 偏置等，为之后的图优化提供原始数据。在对图像帧进行位姿估计后，使当前图像帧和前一图像帧和地图进行联合 BA 进一步优化位姿。在局部建图线程上，增加了对 IMU 进行初始化的功能，使 IMU 能够获得良好的初值，而在局部 BA 上，联合了 IMU 约束，使优化结果更加鲁棒。在闭环线程上，由于 IMU 数据的加入，尺度变得可观测，在回环检测后，使用 6 个自由度的位姿图优化。

3.2 跟踪线程

本文的视觉惯性跟踪需要实时跟踪到传感器的位姿、速度、IMU 偏置，这个可以预估到可靠的相机位姿，而不是像之前的单目 SLAM 系统那样使用点对点运动模型。当位姿被成功估计，地图上的地图点将被投影和帧中的特征点进行匹配，然后我们可以通过最小化特征点的重投影误差和 IMU 误差项对当前帧的位姿进行优化。当地图被局部建图线程或回环线程改变时，跟踪线程对当前帧位姿的优化方式也有所不同，如图 1所示：

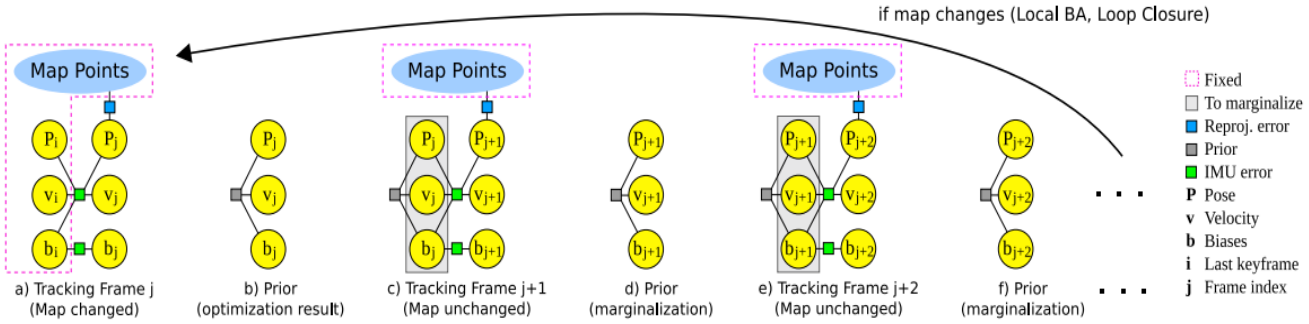

图 1: 当前帧位姿优化[14]

当地图改变时，将上一帧固定，对当前帧进行优化，对于当前帧 j 和上一帧 i 最小二乘公式如下：

$$\theta = \left\{ R_{WB}^{j}, wp_{B}^{j}, wv_{B}^{j}, b_{g}^{j}, b_{a}^{j} \right\} \tag{1}$$

$$\theta^{*} = \underset{\theta}{argmin} \left(\sum_{k} E_{proj}\left(k, j\right) + E_{IMU}\left(i, j\right) \right) \tag{2}$$

其中 R 代表旋转，p 代表位置、v 代表速度，b_g 代表陀螺仪偏置，b_a 代表加速度计偏置,E_{proj} 代表重投影误差，E_{IMU} 代表 IMU 项误差。

当地图没有发生改变时，加入先验项，并对当前帧和上一帧同时进行优化，公式如下：

$$\theta = \left\{ R_{WB}^{j}, wp_{B}^{j}, wv_{B}^{j}, b_{g}^{j}, b_{a}^{j} R_{WB}^{j+1}, wp_{B}^{j+1}, wv_{B}^{j+1}, b_{g}^{j+1}, b_{a}^{j+1} \right\} \tag{3}$$

$$\theta^{*} = \underset{\theta}{argmin} \left(\sum_{k} E_{proj}\left(k, j+1\right) + E_{IMU}\left(j, j+1\right) + E_{prior}\left(j\right) \right) \tag{4}$$

E_{prior} 为先验项定义如下：

$$E_{prior}\left(j\right) = \rho \left(\left[e_{R}^{T} e_{v}^{T} e_{p}^{T} e_{b}^{T} \right] \Sigma_{p} \left[e_{R}^{T} e_{v}^{T} e_{p}^{T} e_{b}^{T} \right]^{T} \right) \tag{5}$$

当没有先验项时，跟踪线程会将当前帧与上一关键帧建立连接，进行优化。

3.3 局部建图线程

局部建图线程在跟踪线程挑选出的关键帧时进行局部 BA，将一个大小为 N 的窗口里的关键帧和所对应的地图点进行优化，其他窗口外的关键帧固定位姿，提供地图点的观测信息，在固定窗口离优化窗口最近的一个关键帧，将会通过 IMU 状态与优化窗口相连如图 2所示：

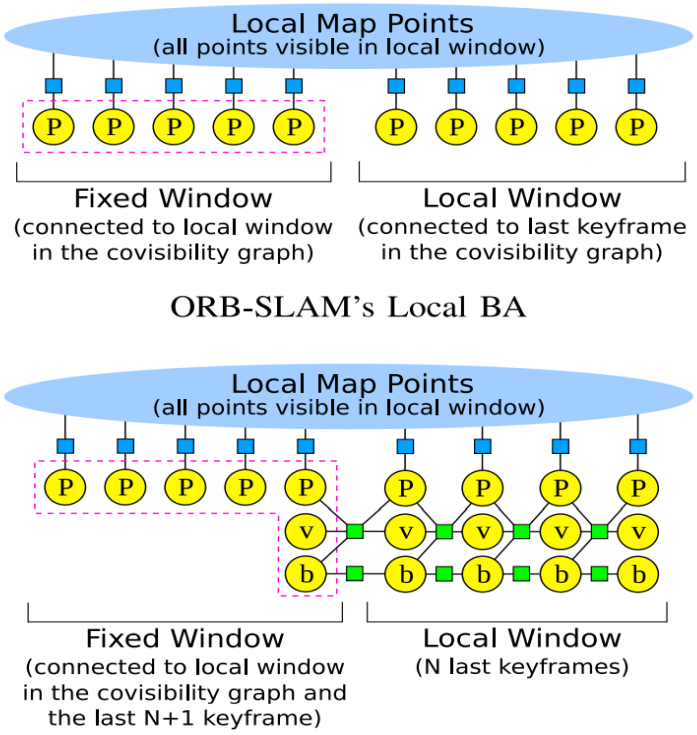

图 2: 局部优化[14]

该局部 BA 与 ORB-SLAM 不同的地方是使用了 IMU 误差项来约束状态，多了速度和偏置误差项，导致优化变量增多，使得优化计算需要更高的代价，所以需要使用合适大小的局部窗口来控制优化规模，保证实时性。由于使用了 IMU 信息，局部建图线程剔除冗余关键帧的策略也有所改变，当两帧的时间差很小，IMU 能够提供的信息非常有限，当两帧的时间差很大，IMU 提供的信息往往没有那么高的准确率，所以在本文中，两个连续关键帧的时间差不小于 0.5 秒和不大于 3 秒，如果不满足这个条件，则将关键帧剔除。

4 复现细节

4.1 与已有开源代码对比

这次复现主要在 ORB-SLAM 中的跟踪线程和局部建图线程中进行了修改，以实现 IMU 预积分、联合 IMU 进行位姿优化、IMU 初始化以及联合 IMU 进行局部优化的功能。其中修改后跟踪线程在原本 ORB-SLAM 的基础上增加了预积分和视觉惯性联合优化功能，其主要功能流程如下所示：

Procedure 1 Tracking

Input: frame F
Output: pose Tcw
if $system\ is\ not\ initialized$ **then**
    Initializer(F)
end
else
    Preintegrated(F) $Tcw = Poseestimate(F)$
    if $IMU\ is\ not\ initialized$ **then**
        $0ptimizeV(F)$
    end
    else
        $0ptimizeVI(F)$
    end
end

首先当系统刚开始时，进行单目初始化，获得初始的地图点和关键帧，初始化后，对每一帧进行预积分，其中 Preintegrated(F) 函数，提取从上一帧到当前帧所有的 IMU 数据，并逐一进行预积分，最终算出两帧之间的旋转、平移以及 IMU 偏置约束，设置当前帧的预积分，大概流程如下所示：

Procedure 2 Preintegrated

Input: imupoint *imu,* frame F
Output: integrate p
for *imu* ***in*** *lastframe to currentframe* **do**
    integrate $p = integrate(imu)$
end
    $setintegrate(F, p)$

算完预积分后，对当前帧进行运动模型估计或者预上一帧进行特征匹配，通过对极几何来计算位姿，最后在通过图优化对位姿进一步优化，0ptimizeV(F) 仅使用视觉信息对位姿进行优化，而在 IMU 初始化完成有了良好的初值之后，使用 0ptimizeVI(F) 进行优化，大概流程如下所示：

Procedure 3 OptimizeVI

Input: frame F
Output: pose Tcw
    createg2osolver()
    setvertex(F.P, F.V, F.ba, F.bg)
for *mappoint* ***in*** *F.matched* **do**
    setvertex(mappoint) setvertexfix(mappoint)
end
    setvertex(lastF.P, lastF.V, lastF.ba, lastF.bg)
if $map\ is\ updated$ **then**
    setvertexfix(lastF)
end
else
    setedge(prior)
end
    setedge()
    pose Tcw = g2osolve()

该非线性优化使用了 G2O 图优化器，将要优化的变量设为节点，在将各变量之间的约束关系作为边加入，利用 G2O 求解优化变量。首先先将当前帧的位姿、速度、IMU 偏置作为节点加入 G2O 优化器，然后将投影到该帧的所有地图点作为节点，并设置为固定状态，表示不进行优化，最后加入上

一帧的位姿、速度、IMU 偏置，根据地图是否更新，决定是否优化上一帧和是否加入先验边，再将所有约束边加入，使用 G2O 优化求解。

其中局部建图在原有的基础上增加了 IMU 初始化、视觉惯性局部优化的功能，其主要功能流程如下所示：

Procedure 4 LocalMapping

Input: Keyframe *KF*
Output: Map *map*

```
    Process(KF)
if IMU is not initialized then
|       map = LocalBA(KF)
end
else
|       map = LocalinertialBA(KF)
end
if IMU is not initialized then
|       IMUinitialize(map)
end
    Keyframeculling()
```

首先局部建图线程处理从跟踪线程传过来的关键帧计算 BoW 字典、检查地图点、更新共视关系，然后判断 IMU 是否初始化，如果没有初始化就使用纯视觉局部 BA，如果 IMU 已经初始化，则进行视觉惯性联合局部 BA，之后如果 IMU 没有初始化且地图有足够的关键帧则开始 IMU 初始化。最后将冗余的关键帧从地图上删除。其中 LocalinertialBA(KF)，用于视觉惯性联合局部 BA，其主要功能流程如下所示：

Procedure 5 LocalinertialBA

Input: Keyframe *KF*
Output: Map *map*

```
    createg2osolver() for kf in local keyframe window do
|       setvertex( kf.P, kf.V, kf.ba, kf.bg)
end
    setvertex( lkf.P, lkf.V, lkf.ba, lkf.bg)      setvertexfix( lkf )
for kf in outside local keyframe window do
|       setvertex( kf.P, kf.V, kf.ba, kf.bg)      setvertexfix( kf )
end
for mappoint in local map do
|       setvertex( mappoint )
end
    setedge()      Map map = g2osolve()
```

该算法，首先将滑动窗口内的关键帧作为节点加入 g2o 优化器，在找里滑动窗口最近的一个关键帧加入节点并固定，在将地图内的其他关键帧也加入 g2o 优化器但是不参与优化，最后将匹配的地图点加入节点，再添加约束边用 g2o 求解，更新地图。

最后是执行 IMU 初始化的 IMUinitialize(map), 该函数初始化当前地图里的关键帧，估计出尺度大小、重力方向，最后再通过尺度缩放将地图成比例缩放，更新关键帧位姿、速度、IMU 偏置，获得较好的初值，主要流程如下所示：

Procedure 6 IMUinitialize

Input: Map *map*

Output: scale *s,* G *g*

```
    createg2osolver()
for kf in map do
|       setvertex( kf.P, kf.V )
end
    setvertex( s, g, ba, bg )
    setedge()
    scale s, G g = g2osolve()
    UpdateMap( s, g )
```

首先将地图里面所有的关键帧的位姿和速度加入 g2o 节点，然后将尺度和重力加入节点，加入约束边并用 g2o 求解器求解，最后通过尺度和重力方向更新地图中的地图点和位姿。

本次复现的代码均基于 ORB-SLAM 开源代码，对其中的函数和数据结构进行了修改，使其完成以上功能。

4.2 实验环境

本次实验使用了 AMD R7-5800H 型号的 CPU，操作系统为 Ubuntu18.04，依赖库如下所示：

- C++ 11
- Pangolin
- OpenCV
- Eigen3
- DBoW2
- g2o

5 实验结果分析

本文使用单目 ORB-SLAM 和复现的视觉惯性 ORB-SLAM 在 EuRoC 数据集上进行了对比实验，EuRoC 数据集由无人机采集的 11 个序列组成，里面包含了单目、双目和 RGB-D 的图像数据，还收集了 IMU 数据，每个序列根据运动情况、环境情况分成了简单序列、中等序列和困难序列。

5.1 单目 SLAM 尺度恢复

在单目 SLAM 系统中，尺度的不确定性能够使用 IMU 融合的方案来解决，否则只能将尺度固定为某一个数值，使得地图轨迹按比例缩放，无法获得真实的数值，不知道当前相机使移动了 1m 还是 1cm。通过在 EuRoC 数据集中 MH 序列的实验，实验结果如图 3、图 4所示：

虚线轨迹为真实轨迹，彩色轨迹为估计轨迹，彩色线条中颜色越暖表示绝对位姿误差越大，越冷表示误差越小尽管存在一定误差，但还是可以明显看出和恢复出来的轨迹尺度能够接近真值轨迹尺度。

5.2 轨迹误差分析

在纯视觉系统中，位姿信息仅仅用图像特征约束，当在无纹理、运动模糊、强光条件下，图像特征难以提取，将会导致位姿无法估计，通过融合 IMU 数据，为图像帧增加约束，使得位姿轨迹估计更加鲁棒，本次复现在 EuRoC 数据集上跑了 7 个序列，实验结果如表 1所示：

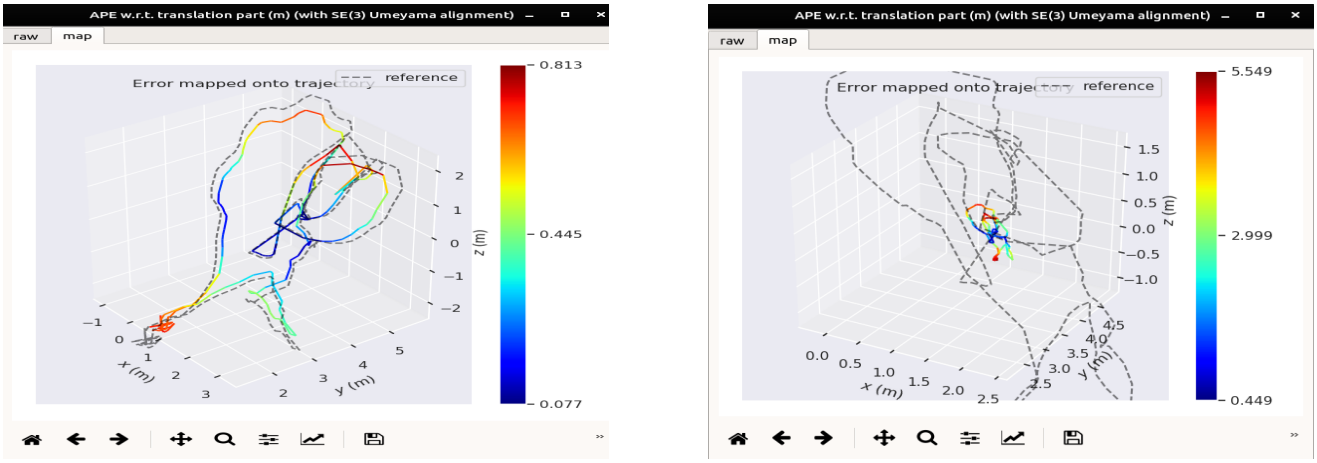

图 3: MH01 序列，左图为 VI-ORBSLAM 结果，右图为 ORB-SLAM 结果

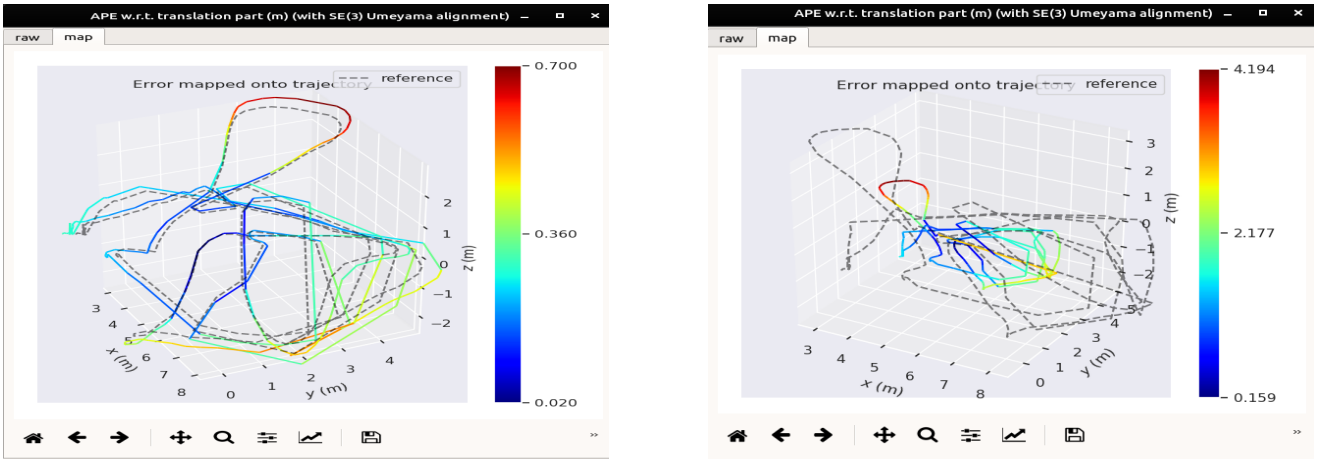

图 4: MH03 序列，左图为 VI-ORBSLAM 结果，右图为 ORB-SLAM 结果

上表展示了在单目方案和视觉惯性方案下在 EuRoC 数据集上绝对轨迹误差的表现，评估指标为平移方均根误差 (RMSE), 该指标用于衡量真实轨迹和估计轨迹的误差，通过实验数据，将 IMU 数据与视觉数据融合后，总体误差有所下降，但是并没有很大的提升，这是因为增加了 IMU 的误差项进行优化，导致计算代价提升，为了保证 SLAM 系统的实时性，需要缩小优化规模，所以对轨迹误差的提升效果不是很大。MH01 序列随着时间变化绝对轨迹误差变化图如图 5所示：

表 1: EuRoC 平移方均根误差

	Monocular ORB-SLAM	VI-ORBSLAM
MH_01_easy	0.046	0.040
MH_02_easy	0.038	0.032
MH_03_medium	0.039	0.034
MH_04_difficult	0.061	0.056
MH_05_difficult	0.051	0.046
V1_01_easy	0.096	0.091
V1_02_medium	0.063	0.062
V1_03_difficult	0.071	0.067
V2_01_easy	0.058	0.054
V2_02_medium	0.056	0.045
V2_03_difficult	0.086	0.081

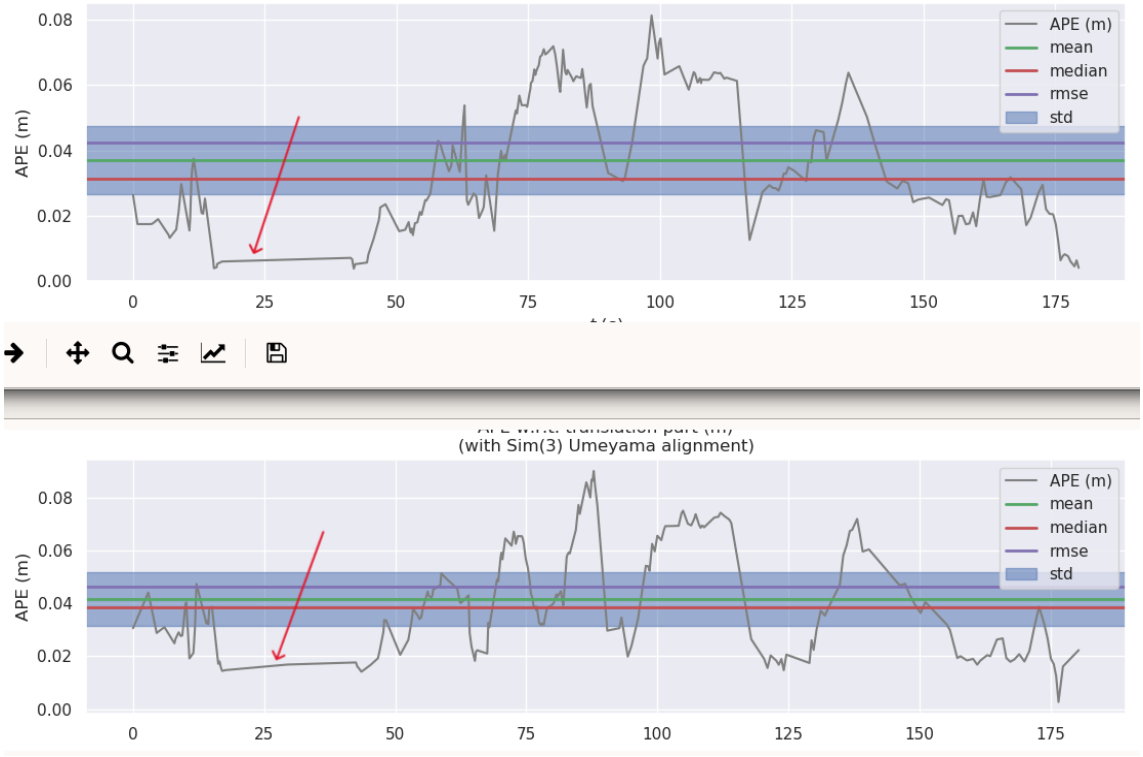

图 5: 绝对轨迹误差

上图为 VI-ORBSLAM 的结果，下图为 Monocular-ORBSLAM 的结果，在红色箭头处可以明显的看到 VI-ORBSLAM 的误差更低。

6 总结与展望

通过融合 IMU 数据，单目 ORB-SLAM 获得了更好的轨迹估计，前端跟踪通过融合 IMU 获得更准确的位姿估计，后端优化融合了 IMU 提升了优化结果。并且通过 IMU 数据，可以解决单目 SLAM 系统中尺度不确定性的问题，通过 IMU 的约束，SLAM 系统可以准确的恢复出真实世界的尺度。

通过这次复现工作，使我对视觉 SLAM 领域有了更深刻的理解，为以后的研究工作打下一定的基础，通过对代码的阅读和改写，使我的编程能力有所提升，对工程文件的结构、写法有了更近一步的学习，在这次复现中，我深入理解了 ORB-SLAM 中的数据处理、程序流程以及各种细节，对我以后的研究、开发工作提供了灵感。在这次复现中，还有许多工作可以继续研究，例如将 ORB-SLAM 中的特征提取，换成更加稳定的特征，以提升 SLAM 的系统的鲁棒性，将 ORB-SLAM 中的优化过程进行改进，使位姿估计效果更佳准确等等。这些都是值得继续深入研究的工作。

参考文献

[1] DAVISON A J, REID I D, MOLTON N D, et al. MonoSLAM: Real-time single camera SLAM[J]. IEEE transactions on pattern analysis and machine intelligence, 2007, 29(6): 1052-1067.

[2] CIVERA J, DAVISON A J, MONTIEL J M. Inverse depth parametrization for monocular SLAM[J]. IEEE transactions on robotics, 2008, 24(5): 932-945.

[3] MOURAGNON E, LHUILLIER M, DHOME M, et al. Real time localization and 3d reconstruction[C]//2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06): vol. 1. 2006: 363-370.

[4] KLEIN G, MURRAY D. Parallel tracking and mapping for small AR workspaces[C]//2007 6th IEEE and ACM international symposium on mixed and augmented reality. 2007: 225-234.

[5] STRASDAT H, MONTIEL J, DAVISON A J. Scale drift-aware large scale monocular SLAM[J]. Robotics: Science and Systems VI, 2010, 2(3): 7.

[6] MOURIKIS A I, ROUMELIOTIS S I, et al. A Multi-State Constraint Kalman Filter for Vision-aided Inertial Navigation.[C]//ICRA: vol. 2. 2007: 6.

[7] LEUTENEGGER S, FURGALE P, RABAUD V, et al. Keyframe-based visual-inertial slam using non-linear optimization[J]. Proceedings of Robotis Science and Systems (RSS) 2013, 2013.

[8] BLOESCH M, OMARI S, HUTTER M, et al. Robust visual inertial odometry using a direct EKF-based approach[C]//2015 IEEE/RSJ international conference on intelligent robots and systems (IROS). 2015: 298-304.

[9] QIN T, LI P, SHEN S. Vins-mono: A robust and versatile monocular visual-inertial state estimator[J]. IEEE Transactions on Robotics, 2018, 34(4): 1004-1020.

[10] QIN T, PAN J, CAO S, et al. A general optimization-based framework for local odometry estimation with multiple sensors[J]. arXiv preprint arXiv:1901.03638, 2019.

[11] VON STUMBERG L, USENKO V, CREMERS D. Direct sparse visual-inertial odometry using dynamic marginalization[C]//2018 IEEE International Conference on Robotics and Automation (ICRA). 2018: 2510-2517.

[12] WILLIAMS B, CUMMINS M, NEIRA J, et al. A comparison of loop closing techniques in monocular SLAM[J]. Robotics and Autonomous Systems, 2009, 57(12): 1188-1197.

[13] MUR-ARTAL R, TARDÓS J D. Fast relocalisation and loop closing in keyframe-based SLAM[C]// 2014 IEEE International Conference on Robotics and Automation (ICRA). 2014: 846-853.

[14] MUR-ARTAL R, TARDÓS J D. Visual-inertial monocular SLAM with map reuse[J]. IEEE Robotics and Automation Letters, 2017, 2(2): 796-803.