深蓝学院视觉 SLAM 理论与实践大作业第三题

评价的开源 SLAM 系统

ORB-SLAM2，DSO，VINS-Mono

使用的数据集

TUM，KITTI，EUROC

轨迹精度评价工具

Evo 是一个开源的 Python 脚本，支持 KITTI，EUROC，TUM 三种数据集的 groundtruth 文件，而且提供了轨迹的对齐功能。让我们计算绝对姿态误差和相对姿态误差非常方便。

ORB-SLAM2 简介

ORB-SLAM2 想必大家都很熟悉吧，ORB-SLAM 是一个基于特征点的实时单目 SLAM 系统，ORB-SLAM2 在 ORB-SLAM 接口的基础上增加了双目和 RGBD 相机的接口。ORB-SLAM2 在大规模的、小规模的、室内室外的环境都可以运行。该系统对剧烈运动也很鲁棒，支持宽基线的闭环检测和重定位，包括全自动初始化。该系统包含了所有 SLAM 系统共有的模块：跟踪（Tracking）、建图（Mapping）、重定位（Relocalization）、闭环检测（Loop closing）。由于 ORB-SLAM2 系统是基于特征点的 SLAM 系统，故其能够实时计算出相机的轨线，并生成场景的稀疏三维重建结果。

DSO 简介

DSO（Direct Sparse Odometry），是慕尼黑工业大学计算机视觉实验室的雅各布.恩格尔博士，于 2016 年发布的一个视觉里程计方法。在 SLAM 领域，DSO 属于稀疏直接法，据论文称能达到传统特征点法的五倍速度，并保持同等或更高精度。然而，由于某些历史和个人的原因，DSO 的代码清晰度和可读性，明显弱于其他 SLAM 方案如 ORB、SVO、okvis 等，使得研究人员很难以它为基础，展开后续的研究工作。所以对初学者来讲并不建议阅读 DSO 的源码。

VINS-Mono 简介

Vins-mono 是香港科技大学开源的一个 VIO 算法，前端采用的是光流追踪，IMU 的测量则采用预积分的形式，VINS-Mono 的后端采用滑动窗口优化，滑动窗口的大小通过 yaml 文件指定。回环检测部分则采用的是 DBOW2，提取 FAST 角点并计算 Brief 描述符。由于没有特征点的提取和匹配工作，Vins-Mono 的运算量比 ORB-SLAM 小得多。VINS_MONO 的后端就是融合 IMU 测量的 BA 优化，不过，和大作业第二题提到的 BA 优化相比，VINS 的策略更好一些。

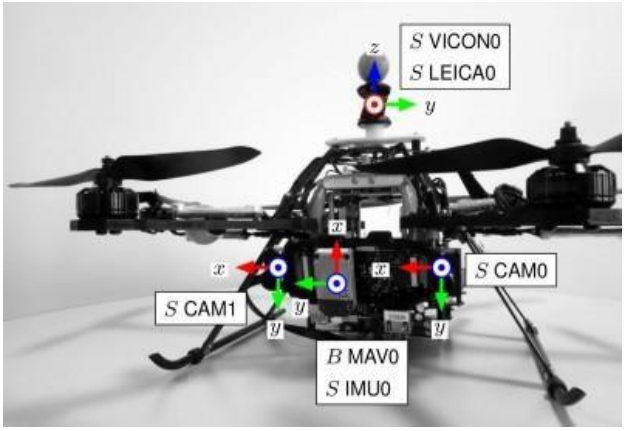
TUM 数据集

TUM 数据集是一个很有名的 RGB-D 数据集，这个数据集是 TUM 的 Computer Vision Lab 公布的。主要是针对纹理丰富的办公室场景，除此之外，这个数据集还有包含很多其他的场景。

他们帧数大小不同，运动有快有慢，有手持的相机。同时，数据集还有不同结构和纹理的数据，运动物体，3D 物体重建。因此这个数据集非常大，更方便的是，这个数据集提供了一些 python 工具，可以轻松的对比 ground truth 和实验结果。

EUROC 数据集

EUROC 提供了两大类数据集，分别是在工厂环境录制的 Machine Hall 系列和在室内录制的 Vicon Room 系列。真实轨迹通过激光跟踪系统捕捉。提供了两个摄像头采集的灰度图像和 IMU 采集的加速度角速率等信息。所有的数据都是对齐好的，官方也给出了精确的传感器标定参数。采集装备如下图所示。两个单目相机的频率是 20Hz，IMU 的频率是 200Hz。IMU 与相机是硬件同步的。位置通过激光跟踪仪获得，精度是 1mm，频率为 20Hz。位姿通过高精度运动捕捉系统捕捉，频率是 100Hz。EUROC 数据集中各个子集的名称、持续时间、平均速度和平均角速度在下面的表格中列出。



EUROC 数据集的采集设备

数据集名称	长度/ 持续时间	平均速度/ 平均角速度
MH_01_easy	80.6m 182s	0.44m/s 0.22rad/s
MH_02_easy	73.5m 150s	0.49m/s 0.21rad/s
MH_03_medium	130.9m 132s	0.99 m/s 0.21 rad/s
MH_04_difficult	91.7m 99s	0.93 m/s 0.24 rad/s
MH_05_difficult	97.6m 111s	0.88 m/s 0.21 rad/s

KITTI 数据集

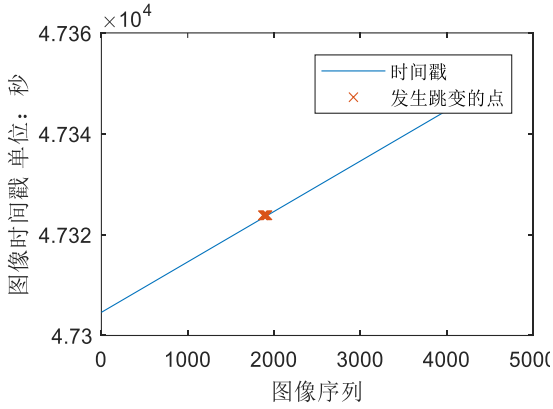
KITTI 是德国卡尔斯鲁厄理工学院利用其自动驾驶平台制作的一个全面的数据集。该平台下图所示，平台包含一个高分辨率的双目相机，可以同时输出彩色图像和灰白图像。一个威力登激光雷达和一个 GPS 定位系统，定位系统使用载波相位技术，提供了 SLAM 系统的真实运动轨迹。KITTI 的制作是为了解决在这之前的很多算法，仅仅在自己的实验的数据集中表现很好，在其他数据集中表现很差的问题。因为 KITTI 数据含量很多，覆盖了街道，校园，住宅区等各种场景，足以模拟真实的各种情况。KITTI 提供的数据集长度达到了 39.2km，这里随机选择了 raw_data 中的 4 个片段进行试验。

KITTI 数据采集车

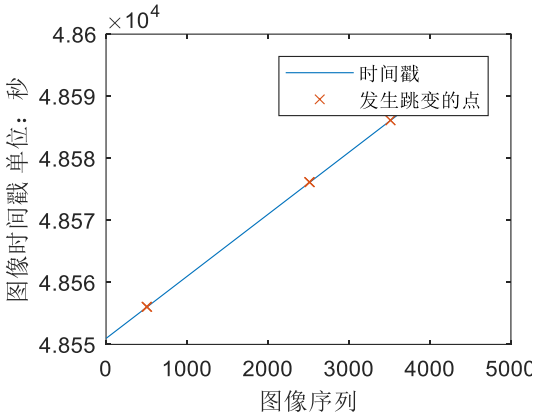
KITTI 的 IMU 时间戳不对齐的处理

博客上说 Kitti 的 IMU 时间戳会跳变，为了检测 KITTI 中 IMU 信息时间戳的连续性，以时间戳的序号为横坐标，时间戳的数值为纵坐标，并标识出前后两个时间戳的差值不在正常

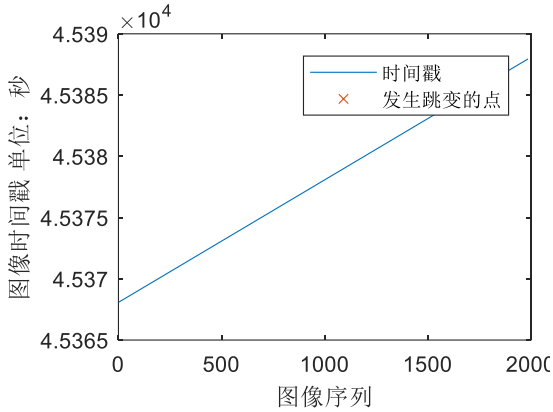
范围内(0.01s)的点。结果如下图所示。



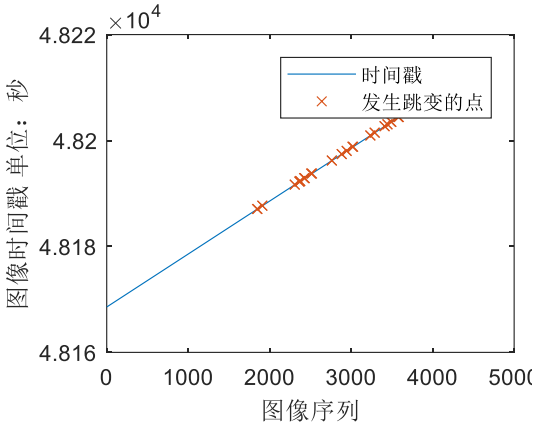
a) 2011_09_26_drive_0009_extract



b) 2011_09_26_drive_0032_extract



c) 2011_09_28_drive_0016_extract



d) 2011_09_26_drive_0028_extract

四个数据集上的时间戳

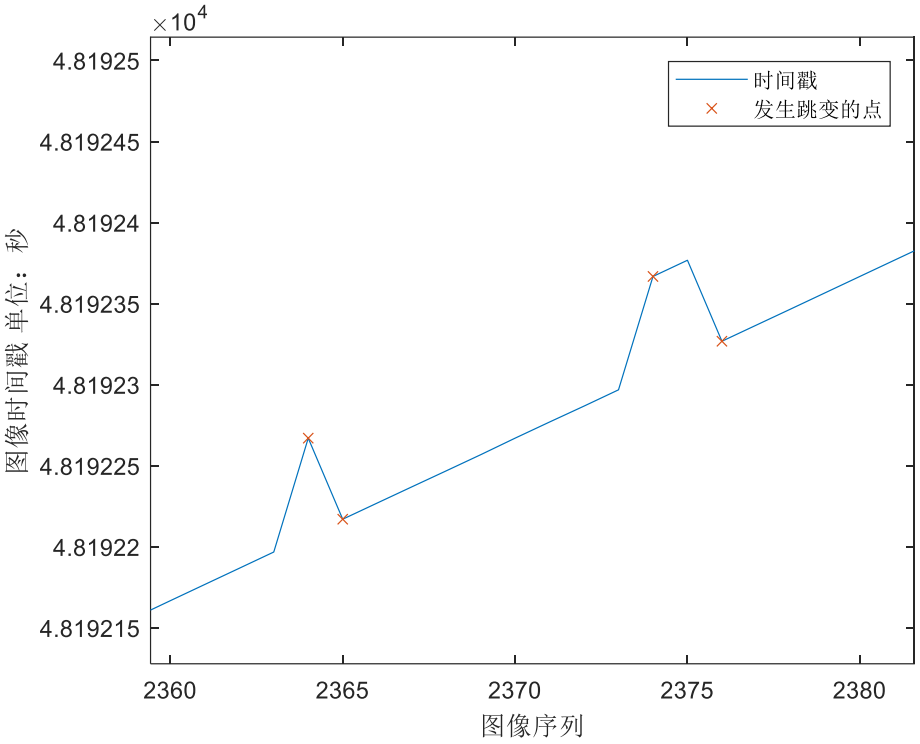
从跳变的放大图可以看出，跳变大致可以分为两种情况：

(1) 时间戳向后跳变一次，然后回复正常；

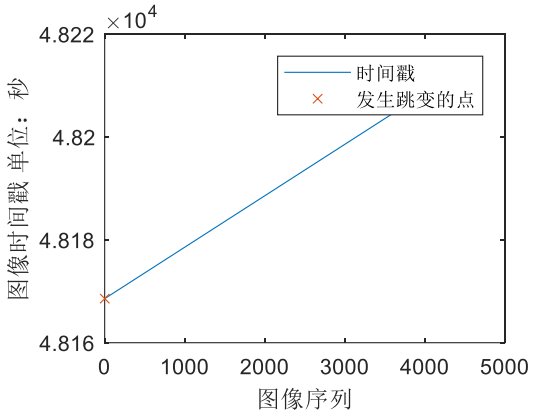
(2) 时间戳向后跳变一段时间，然后回复正常。

针对以上两种情况，采用本文采用的修复跳变的方法为检测相邻两个时间戳是否在（0.005，0.02）区间内，若不在此区间则强制令下一个时间戳为上一个时间戳加 0.01s。为了处理连续跳变的情况，这里‘上一帧’的时间戳是实时更新的，也就是说如果第 i 帧被强制赋值，那么下一个循环则采用强制赋值之后的数据进行判断。

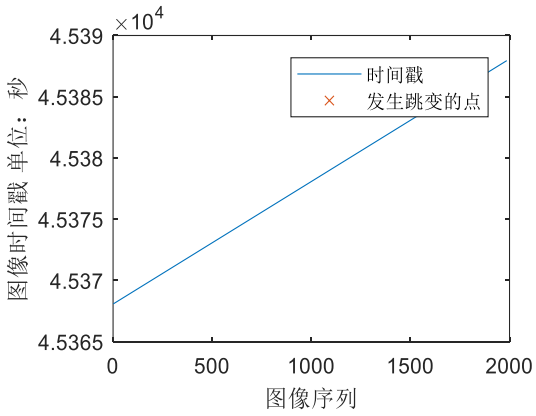
经过处理之后的时间戳都恢复了连续，对于图像的时间戳，用相同的方法进行检查，发现时间的连续性是良好的，没有发生跳变的现象。



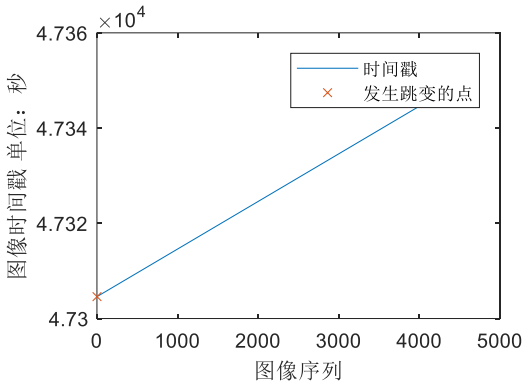
时间戳跳变放大图

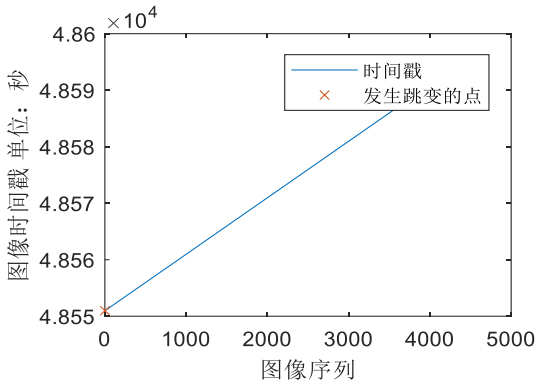


a) 2011_09_26_drive_0028_extract



b) 2011_09_28_drive_0016_extract





c) 2011_09_26_drive_0009_extract　　d) 2011_09_26_drive_0032_extract

经过调整之后四个数据集的时间戳

不过，由于这样处理之后还要把 IMU 和图像录制成 bag 文件才可运行 vins-mono，我录制好的 bag 文件总是不能正常启动 vins，所以并没有成功的在 Kitti 上跑通 vins。

精度评价标准

长期以来，SLAM 一直以相对姿态误差和绝对姿态误差作为评价定位性能的指标。相对姿态误差是指只考虑定位过程中上一时刻与当前时刻位姿变化的误差，这一误差只与上一时刻和当前时刻有关，不会累积。绝对姿态误差在计算时首先要把定位结果和真实定位结果的起点进行对齐，然后计算每一个时刻在世界坐标系下的绝对位置与真实值的差异，绝对姿态误差会累积。但是由于误差漂移的不确定性，积累的误差可能会因为反方向的漂移而减小，所以一般和相对姿态误差一起作为 SLAM 系统的评价指标，本文计算的性能指标包括最大值，平均值和均方根误差(RMSE)，其定义为：

$$Rmse = \sqrt{\frac{\sum_{n} d_i^2}{n}}.$$

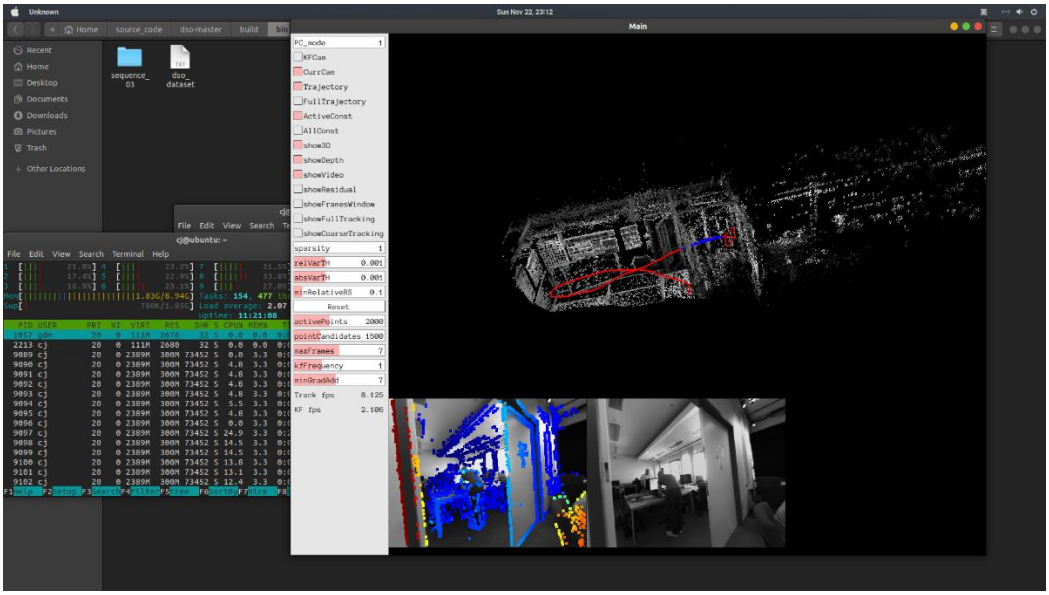
式中 d_i^2 ——测量值与真实值的偏差；

n——测量次数。

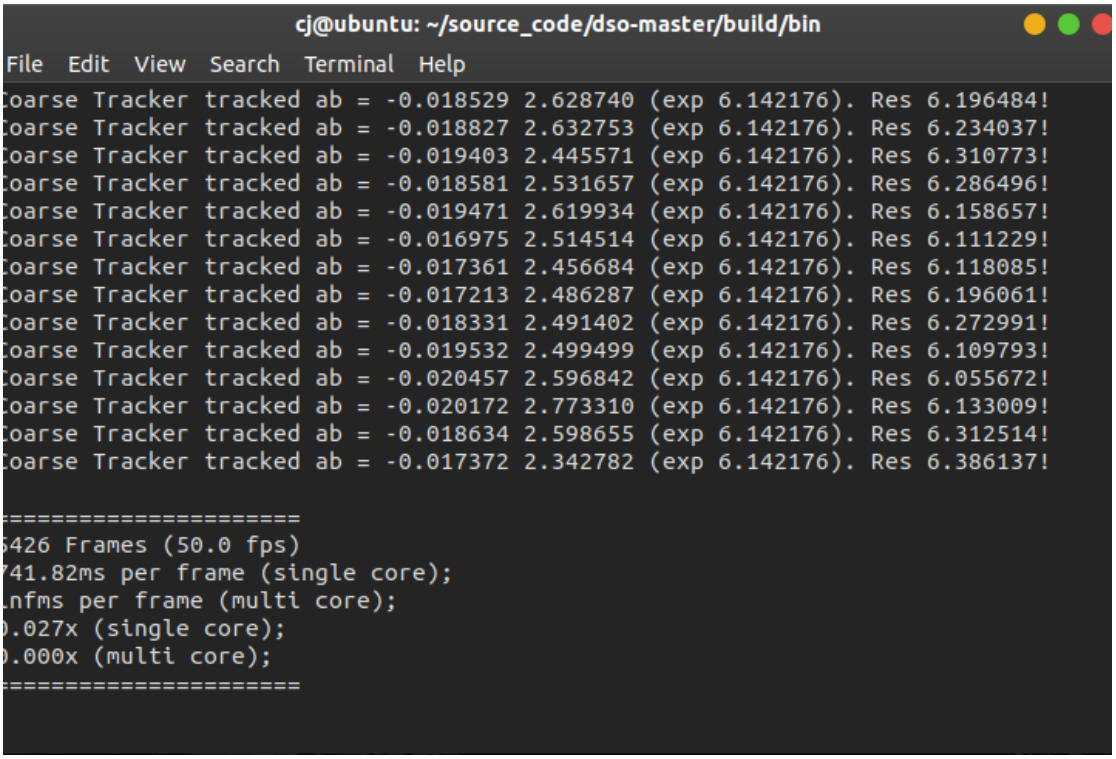
SLAM 系统都带有回环检测功能，能够在很大程度上消除绝对姿态误差，但是这却会使相对姿态误差的表现变得更差，所以相对姿态误差和绝对姿态误差表现得差异体现了回环检测在SLAM 系统中的作用，相对姿态误差和绝对姿态误差虽然在计算方式上有所不同，但是往往都能代表两个不同 SLAM 系统的精度。在 EUROC 数据集中以绝对姿态误差和相对姿态误差同时作为评价系统的标准。

实验过程截图

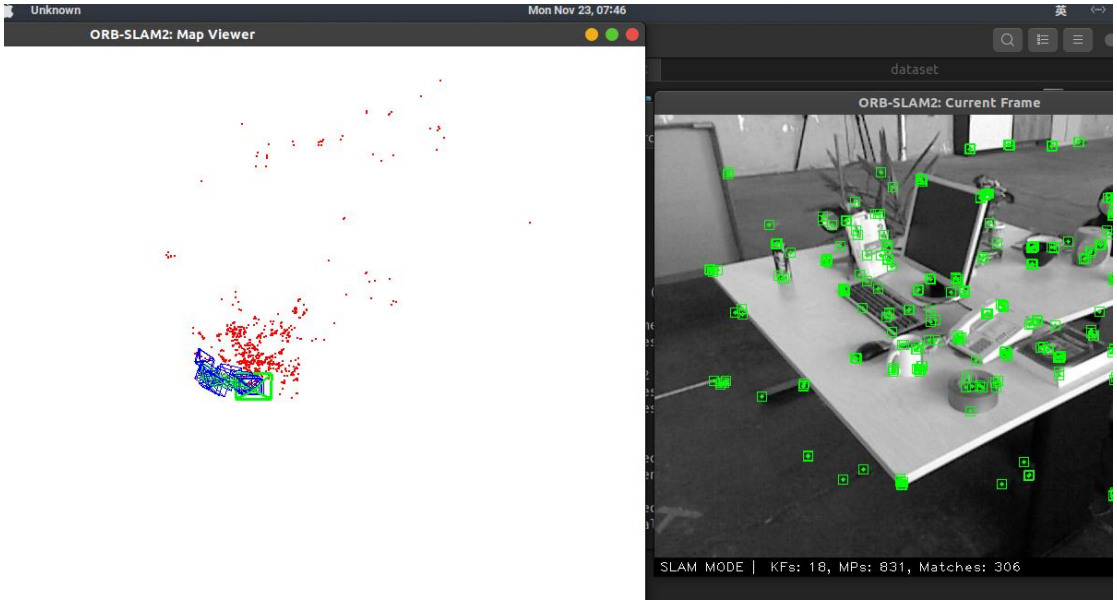
以下是一些实验过程的截图。



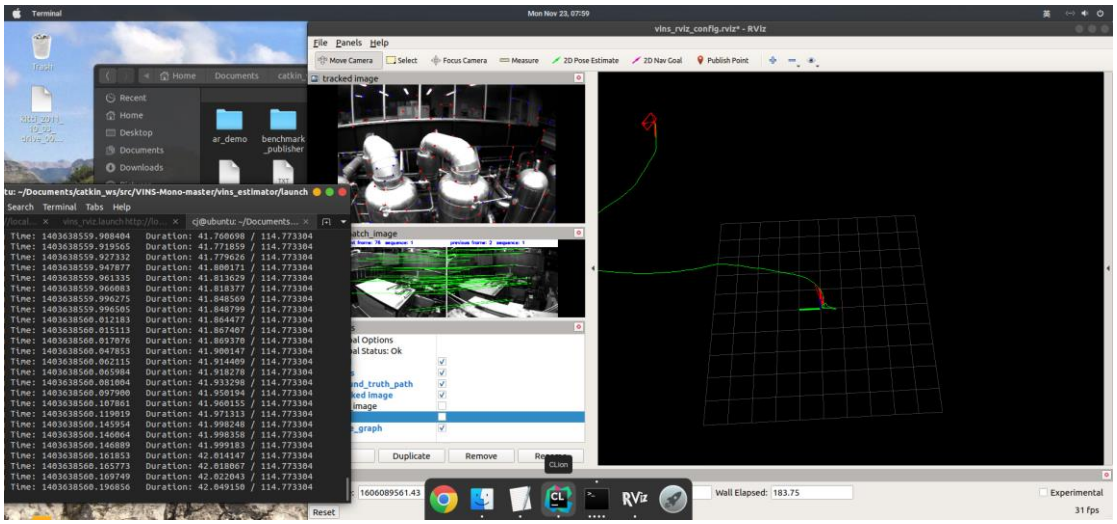
DSO 运行 TUM 的 sequence_03



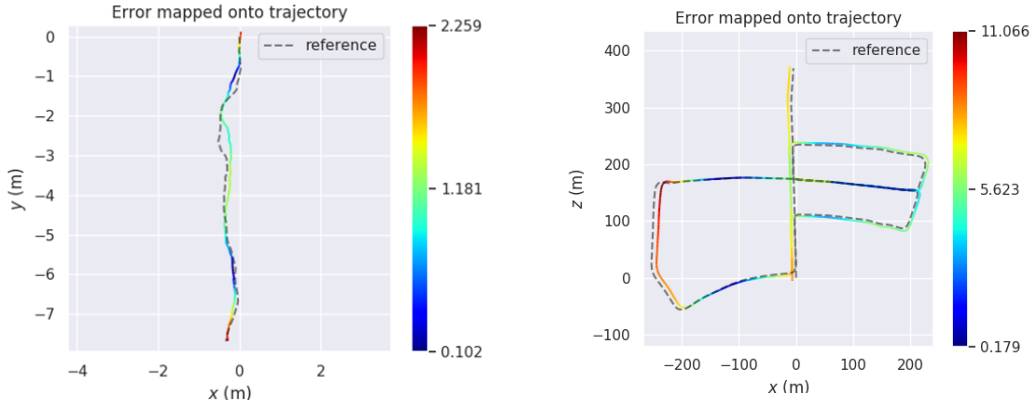
DSO 运行结束后的终端



大家熟悉的 ORB-SLAM2 和 TUM



VIns-Mono 跑 EUROC 数据集



使用 EVO 画出 ORB-SLAM 运行 KITTI 数据集的轨迹

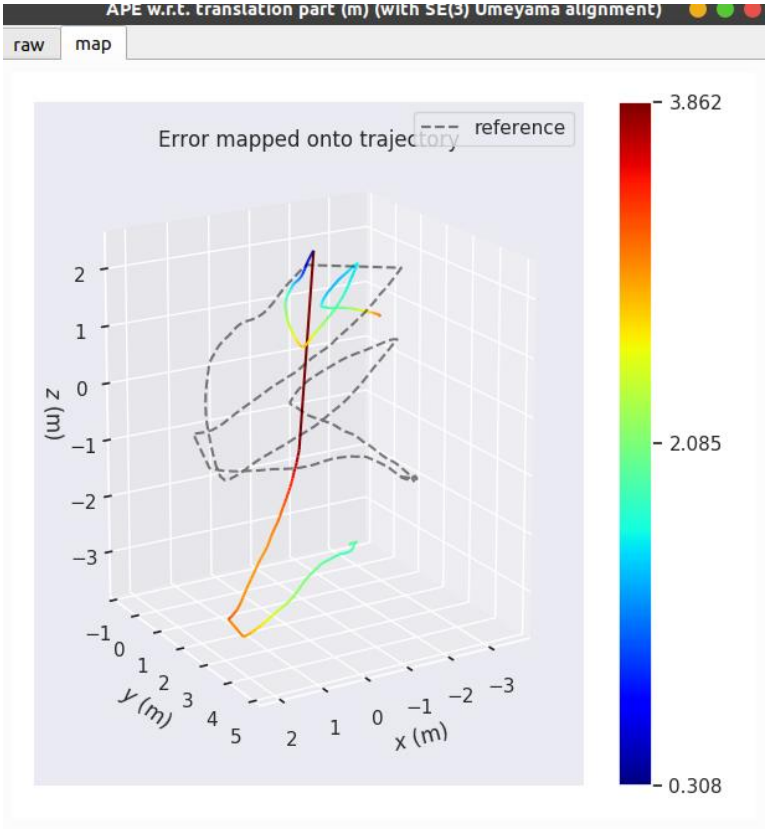
使用 EVO 评价 DSO 轨迹过程中遇到的两个问题

一是 DSO 默认生成的轨迹 result.txt 中包含太多空格，并不是标准的 TUM 格式，需要用以下命令转换一下：

cat result.txt | tr -s [:space:] > DSO.txt

二是 TUM 的 sequence_03 中的 groundtruth 文件夹里面有太多的 Nan，而 evo 中并没有针对这种情况的处理策略。需要把所有 Nan 的行都删除才可以。

可能因为是删除了太多 groundtruth，最终效果很差：

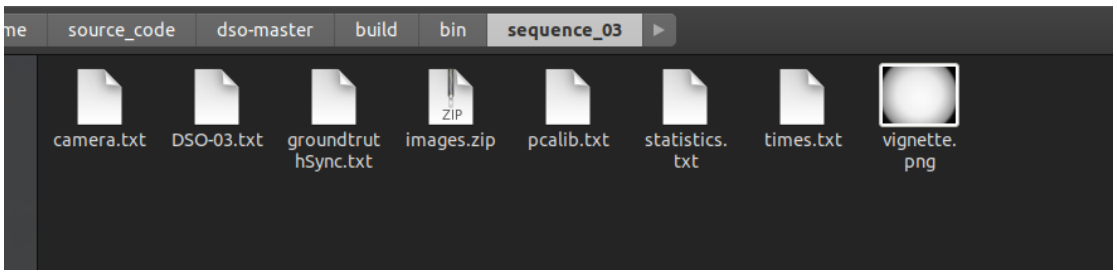


ORB-SLAM 在 TUM 的运行

ORB-SLAM 说明文档很完善，提供了各种数据集的接口，代码清晰易读，这也是为什么很多初学者都把 ORB-SLAM 作为第一个深入研究的 SLAM 系统。

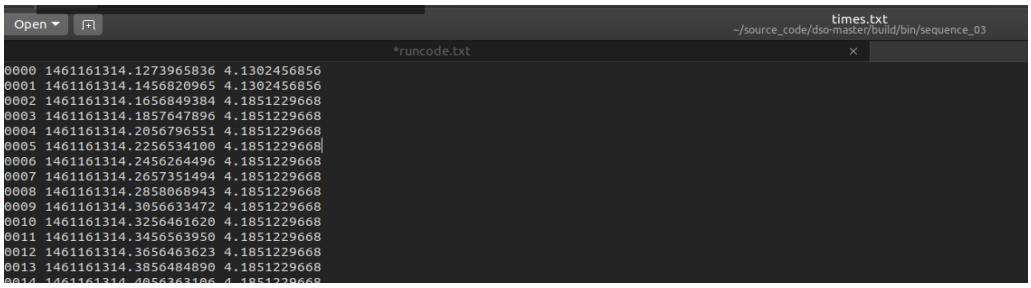
不过虽然都是 TUM 数据集，由于 DSO 需要光度标定，所以 DSO 的数据集和 ORB-SLAM 所有的数据集并不是同一种类型，而且这两种类型互相不兼容，如果想要两个 SLAM 系统同时运行同一个数据集需要对数据集的格式进行修改。

比如在 DSO 种，我们用的是单目数据集，里面文件是这样的：



图像和时间戳的对应关系保存在 times.txt 文件夹中。

Times.txt 文件的格式是这样的，第三行表示曝光时间:



而 ORB-SLAM 用的数据集是 RGBD 数据集，如下如所示：

这个数据集的图像时间戳对应关系记录在 rgb.txt 中：

用 python 或者 matlab 脚本转换一下格式就可以在 DSO 的数据集里面运行 ORB-SLAM 了。

实验结果

以下表格中所有的单位均为 m。统计值为变量为误差。所有实验的参数都是开源代码的默认参数，除了路径外，没有对代码进行任何修改。

Vins-Mono 和 orb-slam 在 Euroc 中的结果

绝对姿态误差

	Vins_Mono			ORB-SLAM		
	最大值	均值	方差	最大值	均值	方差
MH_01	**0.60**	**0.11**	**0.15**	**0.07**	**0.04**	**0.04**
MH_02	**1.19**	**0.09**	**0.16**	**0.08**	**0.03**	**0.03**
MH_03	**9.38**	**0.85**	**1.67**	**0.10**	**0.03**	**0.04**
MH_04	**6.77**	**0.36**	**0.98**	**1.35**	**0.51**	**0.58**
MH_05	**0.56**	**0.14**	**0.16**	**0.11**	**0.05**	**0.05**

相对姿态误差

	Vins_Mono			**ORB-SLAM**		
	最大值	**均值**	**方差**	**最大值**	**均值**	**方差**
MH_01	0.65	**0.003**	**0.016**	**3.75**	**0.35**	**0.532**
MH_02	1.21	**0.004**	**0.042**	**2.14**	**0.33**	**0.451**
MH_03	9.19	**0.032**	**0.357**	**6.23**	**0.82**	**1.305**
MH_04	7.11	**0.020**	**0.315**	**2.00**	**0.44**	**0.533**
MH_05	0.51	**0.006**	**0.020**	**1.98**	**0.49**	**0.63**

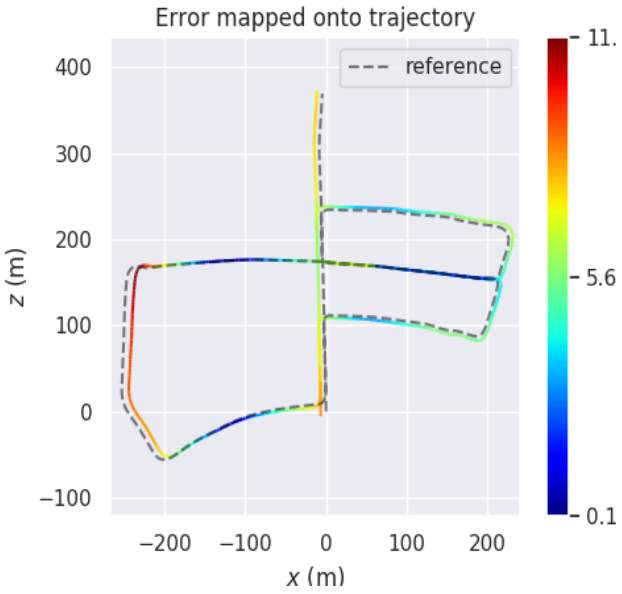
分析：VINS_Mono 有部分步骤跟丢了，所以才会造成误差的最大值比较大。

ORB-SLAM 在 EUROC 中的结果和论文中比较相似。Vins_Mono 则与论文有较大出入。

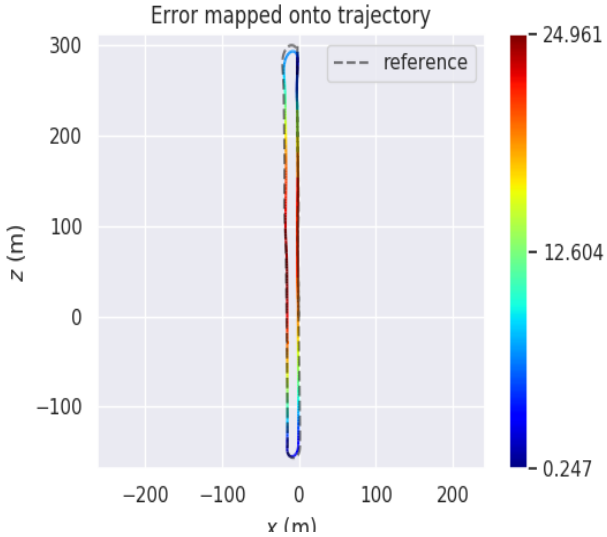
ORB-SLAM 在 KITTI 中的结果

	ORB-SLAM 的相对姿态误差		
	最大值	均值	方差
2011_09_26_drive_0009_extract	25.0	14.04	16.02
2011_09_26_drive_0028_extract	3.90	0.864	1.07
2011_09_26_drive_0032_extract	2.26	1.06	1.18
2011_09_28_drive_0016_extract	11.06	5.70	6.08

部分数据集轨迹与差分式 gps 给出的定位结果的对比：



2011_09_26_drive_0009_extract



2011_09_28_drive_0016_extract

由于 KITTI 是在室外采集的数据，地图的尺度很大，所以定位的误差在数值上会比较大，但是考虑到地图实际的面积，能有现在的精度已经很好了。

ORB-SLAM 和 DSO 在 TUM 数据集上运行的实验结果

由于网络原因，只下载到了数据集中的一个序列，sequence03,运行结果如下：

	最大值	平均值	均方根误差
ORB-SLAM	3.86	2.17	2.30
DSO	4.14	3.05	3.31

可以看出，ORB 在这个序列中由 EVO 给出的定位精度也很差，但是 ORB-SLAM 在追踪过程可以看出追踪的效果并没有那么差，可以验证我们当初的猜想，这里给出的定位结果很差是因为 groundturth 被删除的太多了。