

题目：将第二讲的仿真数据集（视觉特征，imu 数据）接入我们的 VINS 代码，并运行出轨迹结果。

仿真数据集无噪声

仿真数据集有噪声（不同噪声设定时，需要配置 vins 中 imu noise 大小。）

第一步：为了将第二章的仿真数据接入 VINS，仿照 run_enroc.cpp 写 simulation-test.cpp。新建 simulation-test.cpp 代码，具体代码注释见源代码，在 run_enroc.cpp 的基础上做了改动，主要改动是 PubImuData()和 PubImageData()。读取仿真数据需要注意仿真数据的格式。

第二步：将第二章的仿真数据放到当前文件夹下，便于 simulation-test.cpp 找到，主要是要读取 imu_pose.txt imu_pose_noise.txt cam_pose.txt，还要利用 cam_pose_tum.txt 和结果作对比

第三步：在 System.cpp 中添加 PubSimImageData()函数。(具体见代码)

第四步：config 文件夹下 euroc_config.yaml 参数修改，改为与 param.h 中参数一样。

第五步：使用 evo 工具评测 SLAM 算法性能并可视化结果

simulation-test 生成的 pose_output.txt 是 TUM 数据集格式，每行有 8 个元素：

timestamp tx ty tz qx qy qz qw

结尾没有空格，时间以秒为单位

注意：

- 1、第二章生成仿真数据要修改噪声设置，同时要修改第七章 config 文件夹下的噪声设置
- 2、第二章每次生成的仿真数据 vio_data_simulation 整个复制到第七章 src 文件夹下
- 3、第二章 camera_pose_tum.txt 数据文件存到第七章 bin 文件夹下
- 4、每次运行 simulation-test 要先删除上一次的 pose_output，否则数据会追加在后面
- 5、生成的 pose_output 文件数据格式不是严格的 TUM，利用 bin 下的 python 文件修改空格问题，生成 pose_output_change.txt，每次也要删除上一次的文件，否则也会追加
- 6、simulation-test.cpp 读取文件路径不要搞错

第二章仿真数据说明：

运动轨迹是一个 xy 平面椭圆运动，z 正弦运动

house_model 中保存了原始设置的 3d 房屋关键点（一行两个点连成线，保存点前判断一下是否之前获取过）

运行后生成如下数据：

keyframe 文件夹：每一帧相机观测到的特征点，代码中设定相机帧率为 30Hz，仿真 20s，对应一共产生 600 帧观测

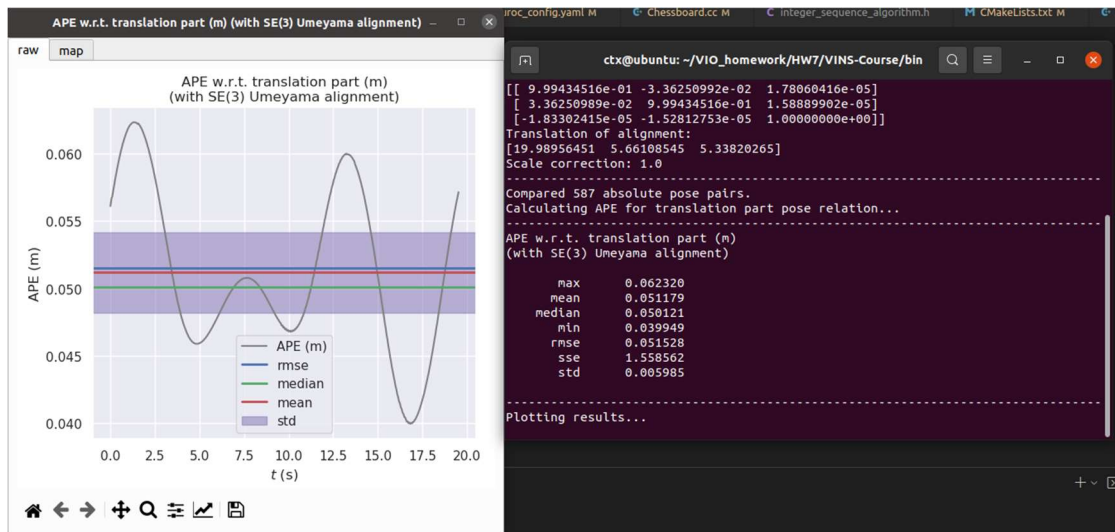
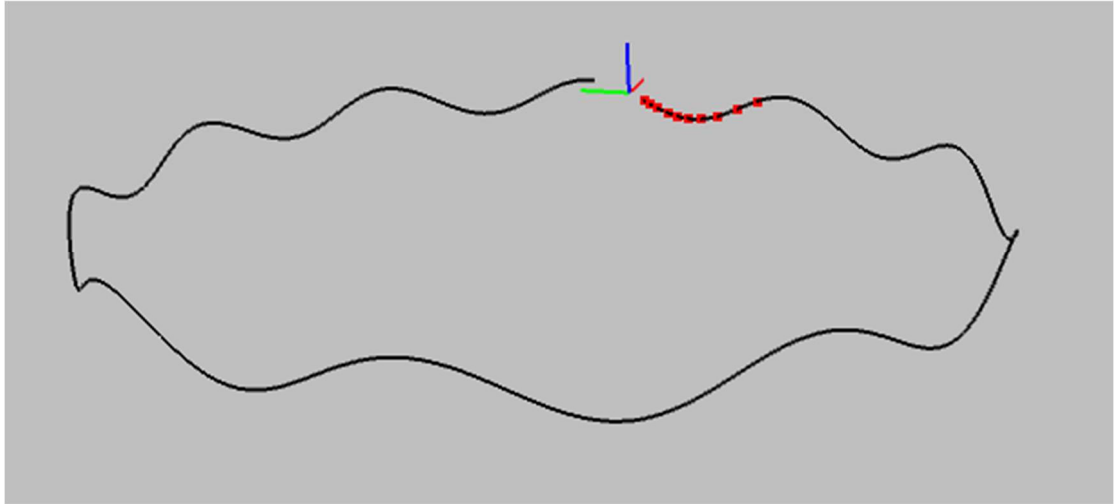
all_points:保存了所有特征点（对比房屋关键点可以有所增加）文件存储的是 house 模型的线特征，每行 4 个数，对应该线两端点在归一化平面的坐标，一共 23 行对应 23 条线段

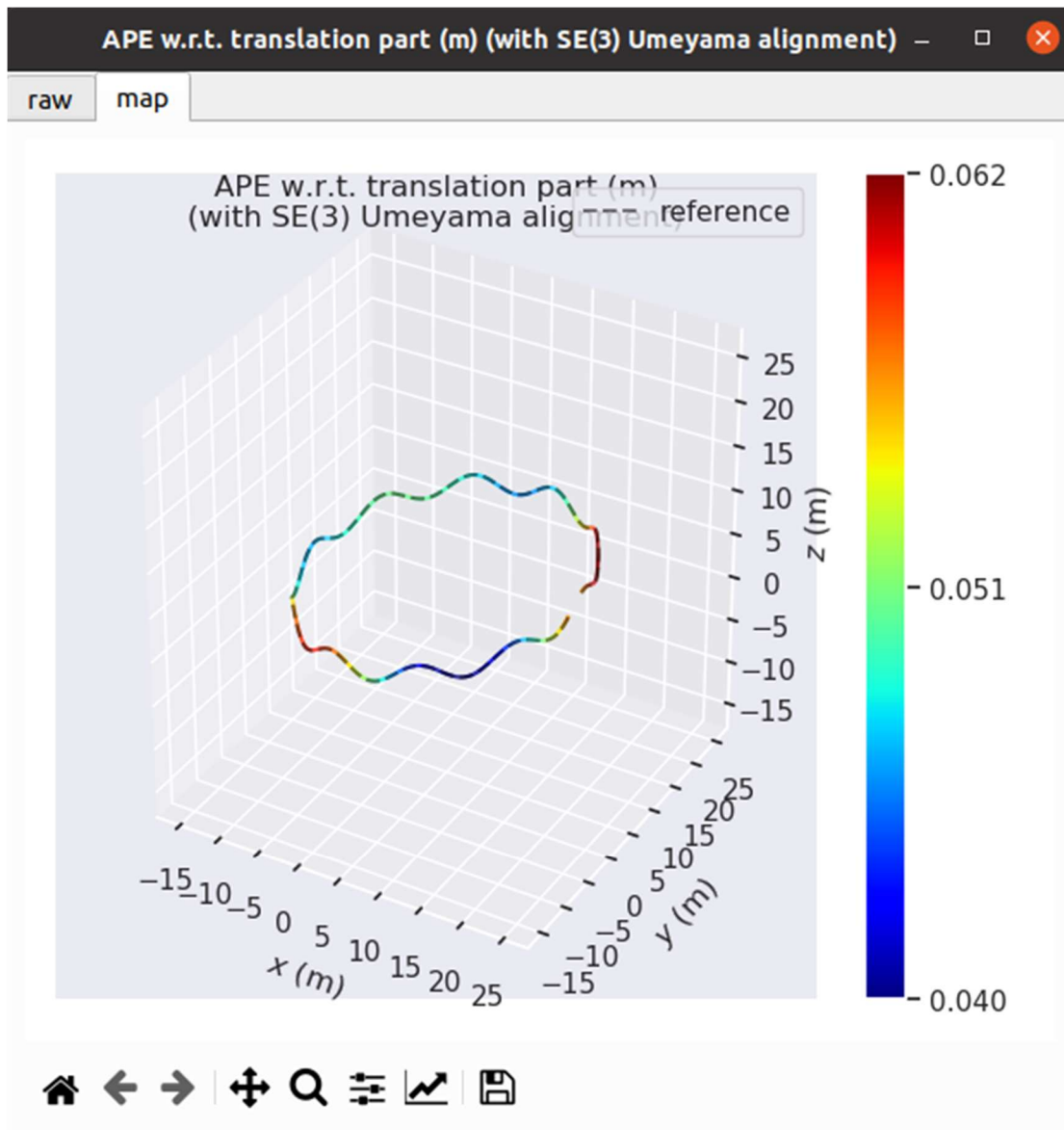
cam_pose:所有相机位姿， cam_pose_tum 只是存储顺序和变量个数有所不同

imu_pose:所有 imu 位姿， imu_pose_noise 带噪声

imu_int_pose:积分后得到的 imu 轨迹， imu_int_pose_noise 带噪声

仿真数据集无噪声运行结果：（接入 imu_pose.txt）





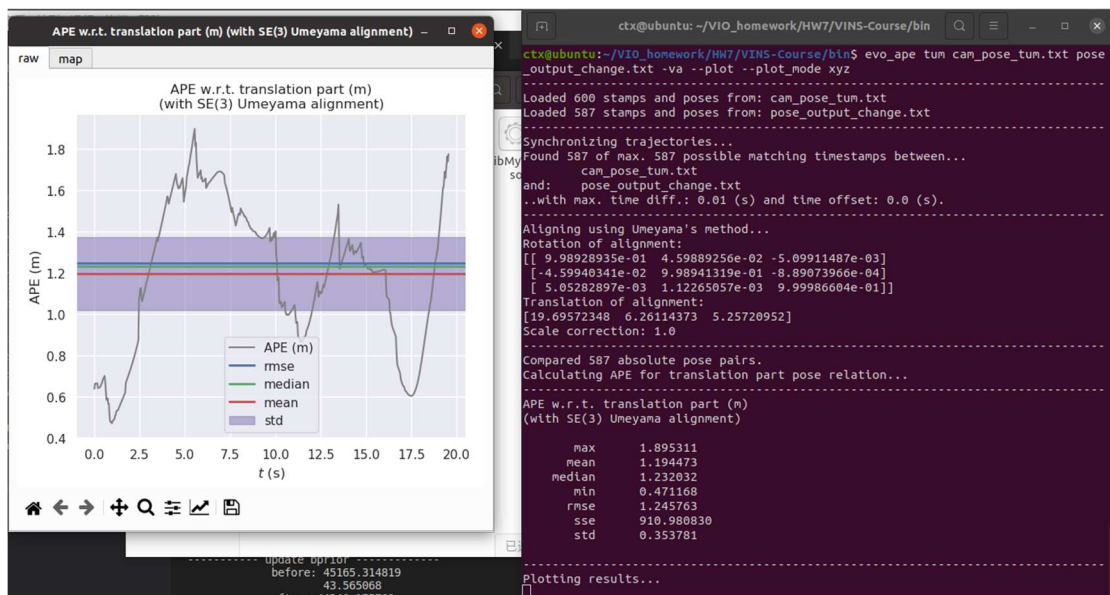
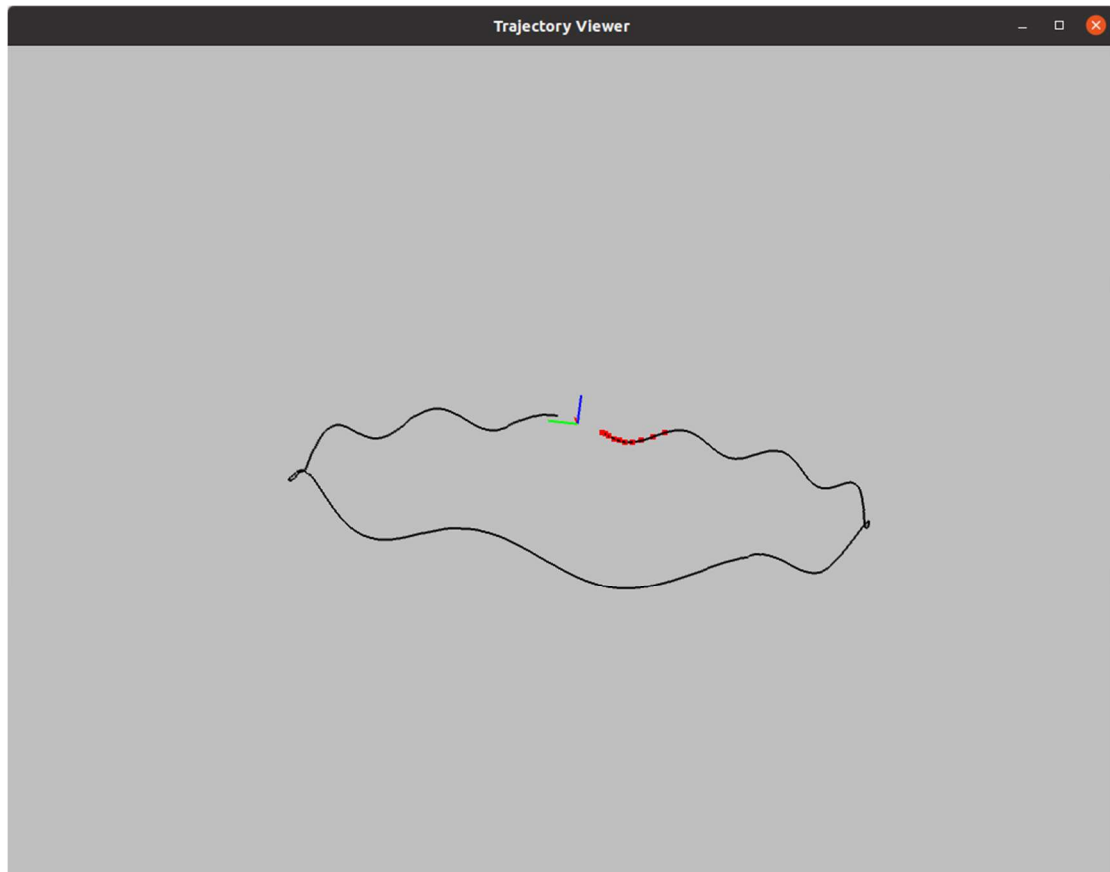
小噪声：(接入 imu_pose_noise.txt)

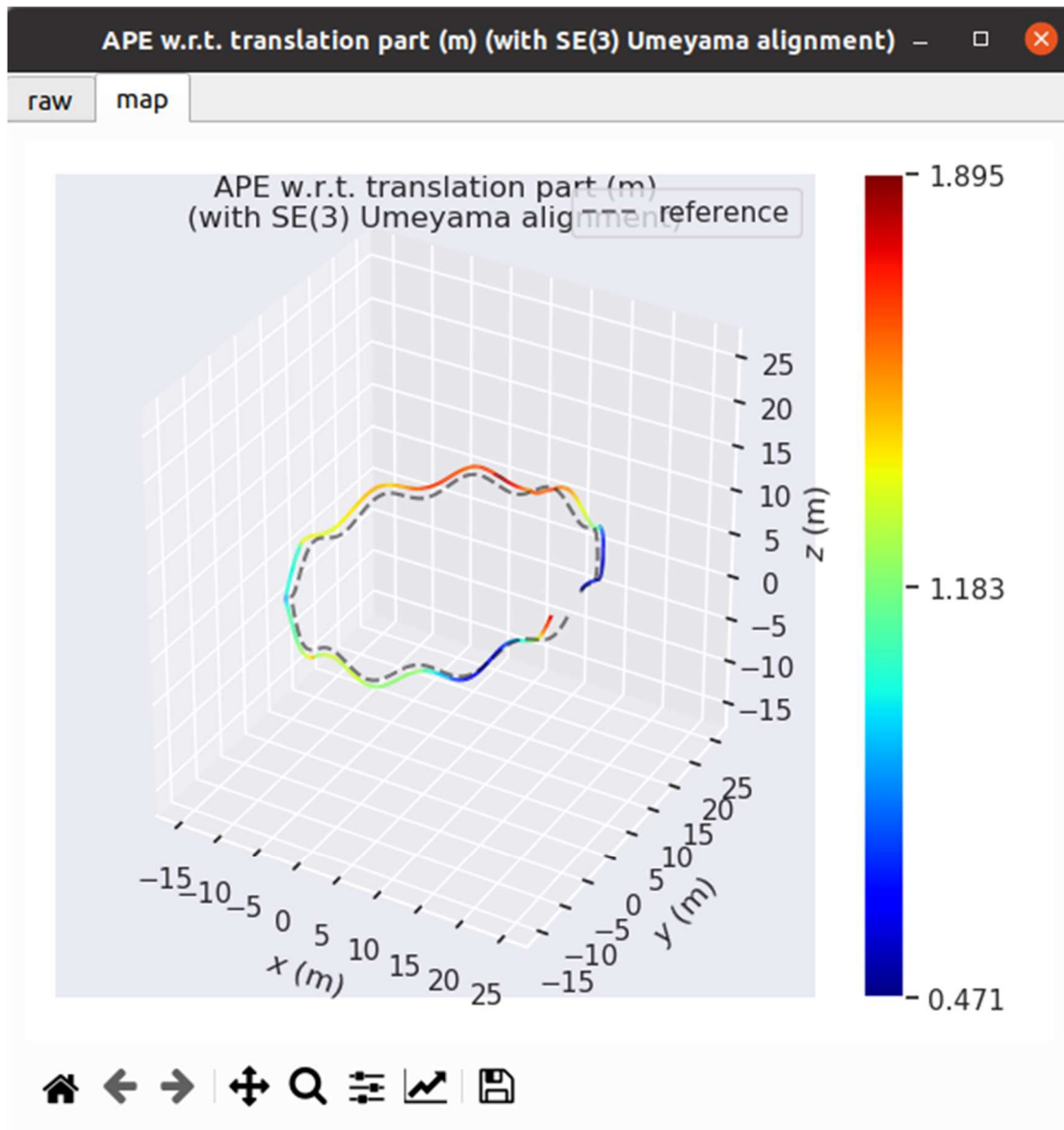
在利用第二章代码生成 imu_pose_noise.txt 时的配置：(param.h)

```
// noise
double gyro_bias_sigma = 1.0e-6;
double acc_bias_sigma = 0.0001;

double gyro_noise_sigma = 0.015; // rad/s * 1/sqrt(hz)
double acc_noise_sigma = 0.019; // m/(s^2) * 1/sqrt(hz)

double pixel_noise = 1; // 1 pixel noise
```





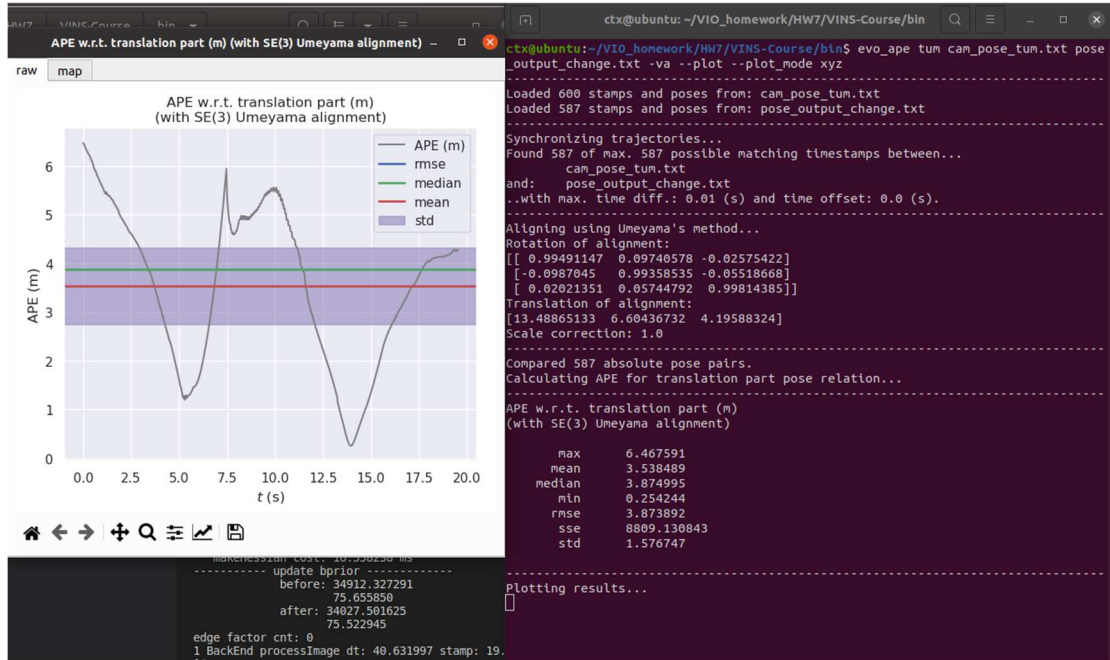
大噪声: (接入 imu_pose_noise.txt)

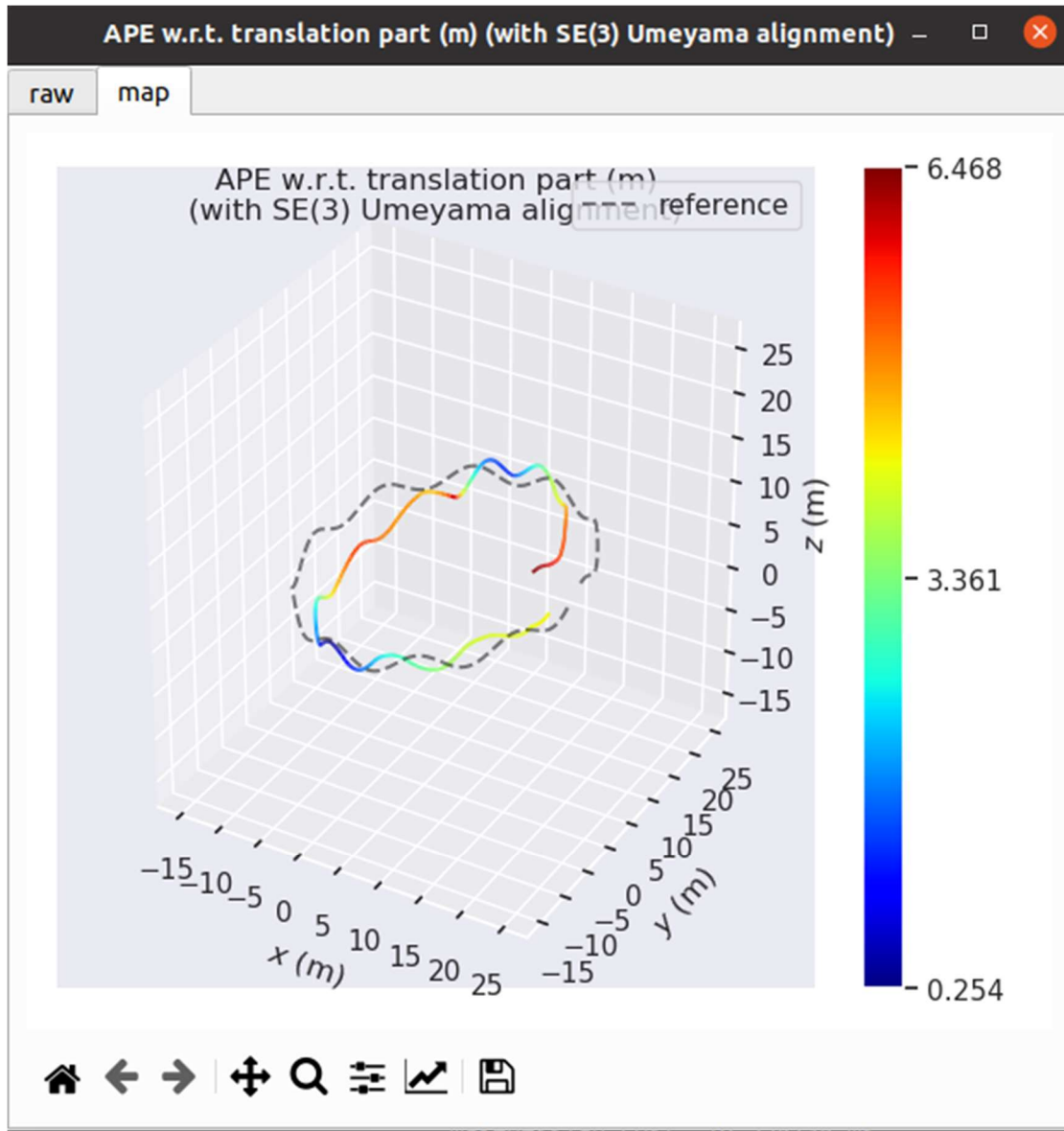
噪声配置(param.h)

```
// noise
double gyro_bias_sigma = 1.0e-5;
double acc_bias_sigma = 0.0001;

double gyro_noise_sigma = 0.015; // rad/s * 1/sqrt(hz)
double acc_noise_sigma = 0.019; // m/(s^2) * 1/sqrt(hz)

double pixel_noise = 1; // 1 pixel noise
```





分析：

每种情况运行结果和生成的数据集有很大关系，大噪声有时候偏的离谱。

可以看到设置的 imu 噪声越大，通过 VINS 估计出来的轨迹结果越差。