

SOUSIC-第7章作业

1. 代码修改

1. 创建文件 run_simulation.cpp，内容基本参照 run_euroc.cpp，修改一下数据路径和读取数据的方法

a. 数据路径

```
test > run_simulation.cpp > PubImageData()
17
18 const int nDelayTimes = 2;
19 string sData_path = "../../ch2/course2_hw_new/vio_data_simulation/bin/";
20 string sConfig_path = "../config/simulation_config.yaml";
21
22 std::shared_ptr<System> pSystem;
```

- b. PubImuData() 修改：注意把前面 7 个不需要的数据先输出到 tmp 中

```
run_simulation.cpp 3, u X System.cpp 4, M run_euroc.cpp 3, M System.h 5, M CMake
test > run_simulation.cpp > PubImageData()
22 std::shared_ptr<System> pSystem;
23
24 void PubImuData() {
25     string sImu_data_file = sData_path + "imu_pose_noise.txt";
26     cout << "1 PubImuData start sImu_data_file: " << sImu_data_file << endl;
27     ifstream fsImu;
28     fsImu.open(sImu_data_file.c_str());
29     if (!fsImu.is_open()) {
30         cerr << "Failed to open imu file! " << sImu_data_file << endl;
31         return;
32     }
33
34     std::string sImu_line;
35     double dStampNSec = 0.0;
36     Vector3d vAcc;
37     Vector3d vGyr;
38     double tmp;
39     while (std::getline(fsImu, sImu_line) &&
40            !sImu_line.empty()) { // read imu data
41         std::istringstream ssImuData(sImu_line);
42         ssImuData >> dStampNSec; // timestamp
43         for (int i = 0; i < 7; i++) {
44             ssImuData >> tmp; // imu quaternion(4), imu position(3)
45         }
46         ssImuData >> vGyr.x() >> vGyr.y() >> vGyr.z() >> vAcc.x() >> vAcc.y() >>
47             vAcc.z();
48         // time is second
49         pSystem->PubImuData(dStampNSec, vGyr, vAcc);
50         usleep(5000 * nDelayTimes);
51     }
52     fsImu.close();
53 }
54
```

- c. PubImageData() 修改：注意 txt 文件的路径；x, y, z, 1 不需要，先输出到 tmp 中；新建一个函数 pSystem->PubFeatureData 来处理 features

```
run_simulation.cpp 3, u x System.cpp 4, M run_euroc.cpp 3, M System.h 5, M CMake
test > run_simulation.cpp > PubImageData()
55 void PubImageData() {
56     string sImage_file = sData_path + "cam_pose.txt";
57
58     cout << "1 PubImageData start sImage_file: " << sImage_file << endl;
59
60     ifstream fsImage;
61     fsImage.open(sImage_file.c_str());
62     if (!fsImage.is_open()) {
63         cerr << "Failed to open image file! " << sImage_file << endl;
64         return;
65     }
66
67     std::string sImage_line;
68     double dStampNSec;
69
70     int n = 0;
71     // cv::namedWindow("SOURCE IMAGE", CV_WINDOW_AUTOSIZE);
72     while (std::getline(fsImage, sImage_line) && !sImage_line.empty()) {
73         std::istringstream ssImuData(sImage_line);
74         ssImuData >> dStampNSec;
75         string pointsFileName
76             sData_path + "keyframe/all_points_" + to_string(n) + ".txt";
77
78         std::ifstream fsPoints;
79         fsPoints.open(pointsFileName);
80         if (!fsPoints.is_open()) {
81             cerr << "Failed to open points file! " << sImage_file << endl;
82             return;
83         }
84
85         std::vector<cv::Point2f> featurePoints;
86         double tmp;
87         std::string sPoints_line;
88         while (std::getline(fsPoints, sPoints_line) && !sPoints_line.empty()) {
89             std::istringstream ssPointsData(sPoints_line);
90             for (size_t i = 0; i < 4; i++) {
91                 ssPointsData >> tmp; // x, y, z, l, u, v
92             }
93             cv::Point2f tmpPoint;
94             ssPointsData >> tmpPoint.x >> tmpPoint.y;
95             featurePoints.emplace_back(tmpPoint);
96         }
97         pSystem->PubFeatureData(dStampNSec, featurePoints);
98
99         usleep(50000 * nDelayTimes);
100         n++;
101     }
102     fsImage.close();
103 }
```

d. 在 System.cpp 中新建 `pSystem->PubFeatureData` 函数

```

src > System.cpp > PubFeatureData(double dStampSec, const vector<cv::Point2f> &featurePoints) {
50 void System::PubFeatureData(double dStampSec, const vector<cv::Point2f> &featurePoints) {
51     if (!init_feature) {
52         cout << "1 PubImageData skip the first detected feature, which doesn't contain optical flow speed" << endl;
53         init_feature = 1;
54         return;
55     }
56
57     if (first_image_flag) {
58         cout << "2 PubImageData first_image_flag" << endl;
59         first_image_flag = false;
60         first_image_time = dStampSec;
61         last_image_time = dStampSec;
62         return;
63     }
64     // detect unstable camera stream 发现时间戳不连续甚至倒退, 提示重新输入
65     if (dStampSec - last_image_time > 1.0 || dStampSec < last_image_time) {
66         cerr << "3 PubImageData image discontinue! reset the feature tracker!" << endl;
67         first_image_flag = true;
68         last_image_time = 0;
69         pub_count = 1;
70         return;
71     }
72     last_image_time = dStampSec;
73
74     PUB_THIS_FRAME = true;
75
76     TicToc t_r;
77
78     if (PUB_THIS_FRAME) {
79         pub_count++;
80         shared_ptr<IMG_MSG> feature_points(new IMG_MSG());
81         feature_points->header = dStampSec;
82         vector<set<int>> hash_ids(NUM_OF_CAM);
83         for (int i = 0; i < NUM_OF_CAM; i++) {
84             for (unsigned int j = 0; j < featurePoints.size(); j++) {
85                 int p_id = j;
86                 hash_ids[i].insert(p_id);
87                 double x = featurePoints[j].x;
88                 double y = featurePoints[j].y;
89                 double z = 1;
90                 feature_points->points.push_back(Vector3d(x, y, z));
91                 feature_points->id_of_point.push_back(p_id * NUM_OF_CAM + i);
92
93                 float u = 460 * x + 255;
94                 float v = 460 * y + 255;
95                 feature_points->u_of_point.push_back(u);
96                 feature_points->v_of_point.push_back(v);
97                 feature_points->velocity_x_of_point.push_back(0);
98                 feature_points->velocity_y_of_point.push_back(0);
99             }
100
101             // skip the first image; since no optical speed on frist image
102             if (!init_pub) {
103                 cout << "4 PubImage init_pub skip the first image!" << endl;
104                 init_pub = 1;
105             } else {
106                 m_buf.lock();
107                 feature_buf.push(feature_points);
108                 // cout << "5 PubImage t : " << fixed << feature_points->header
109                 // << " feature_buf size: " << feature_buf.size() << endl;
110                 m_buf.unlock();
111                 con.notify_one();
112             }
113         }
114     }
115 }

```

2. 修改 CmakeLists.txt

```

M CMakeLists.txt M X run_simulation.cpp 3, U System.cpp 4, M
M CMakeLists.txt
84
85 add_executable(run_simulation test/run_simulation.cpp)
86 target_link_libraries(run_simulation
87     MyVio
88     -lpthread)
89

```

3. 在 config 文件夹下创建 simluation_config.yaml，复制 euroc_config.yaml 中需要的部分，并做如下修改：

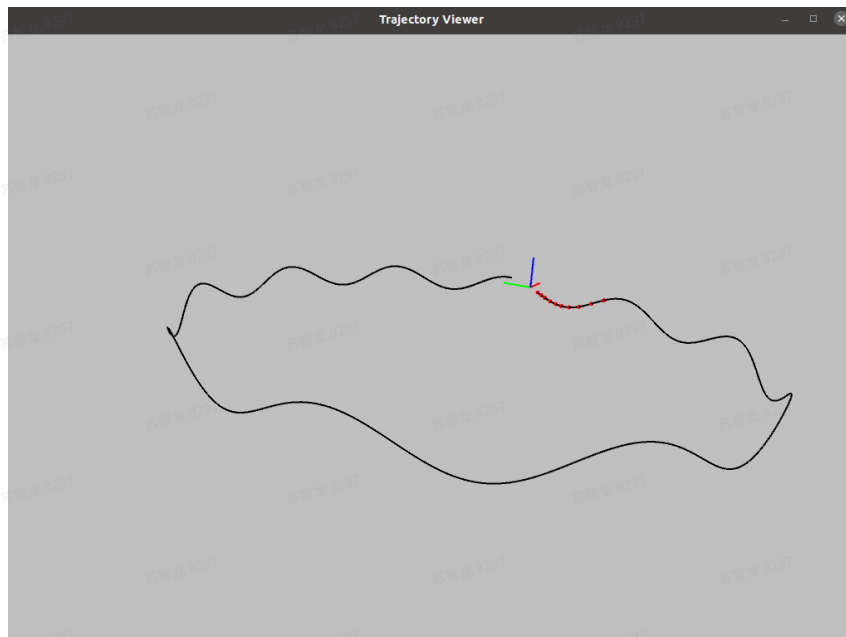
```
config > ! simulation_config.yaml
1  %YAML:1.0
2
3  image_width: 640
4  image_height: 640
5  distortion_parameters:
6    k1: 0
7    k2: 0
8    p1: 0
9    p2: 0
10 projection_parameters:
11   fx: 460
12   fy: 460
13   cx: 255
14   cy: 255
15
16 extrinsicRotation: !!opencv-matrix
17   rows: 3
18   cols: 3
19   dt: d
20   data: [0, 0, -1,
21         -1, 0, 0,
22         0, 1, 0]
23 #Translation from camera frame to imu frame, imu^T_cam
24 extrinsicTranslation: !!opencv-matrix
25   rows: 3
26   cols: 1
27   dt: d
28   data: [0.05,0.04,0.03]
29
30 #imu parameters      The more accurate parameters you provide, the better performance
31 acc_n: 0.019         # accelerometer measurement noise standard deviation. #0.2  0.04
32 gyr_n: 0.015         # gyroscope measurement noise standard deviation.   #0.05  0.004
33 acc_w: 0.0001        # accelerometer bias random work noise standard deviation. #0.02
34 gyr_w: 1.0e-5        # gyroscope bias random work noise standard deviation.  #4.0e-5
35 g_norm: 9.81007      # gravity magnitude
```

2. 无噪声

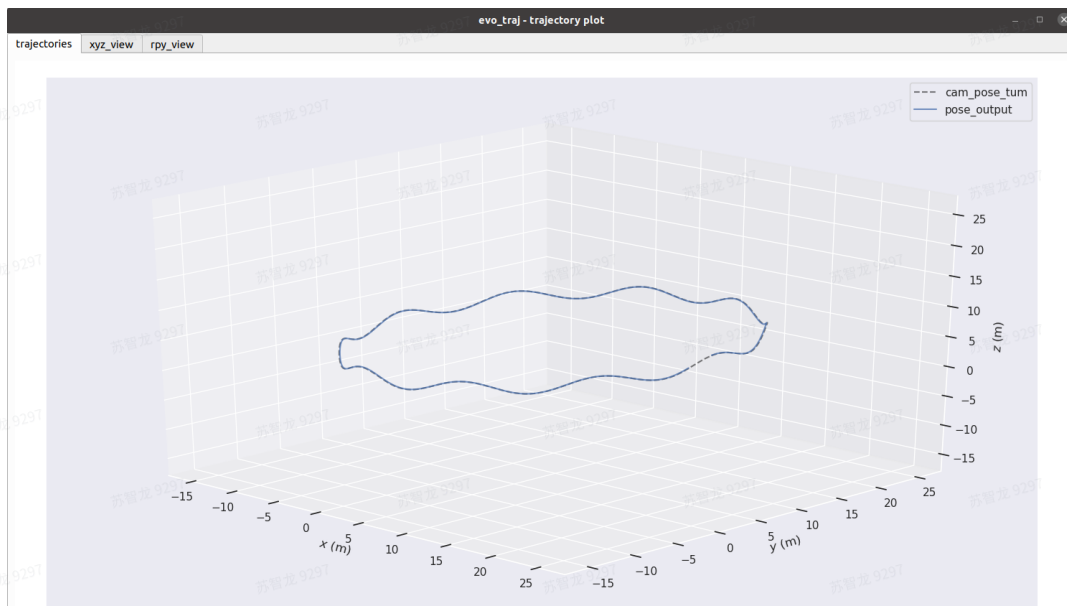
2.1 代码修改

```
run_simulation.cpp 3, U × System.cpp 4, M run_euroc.cpp 3, M System.h 5, M CMake
test > run_simulation.cpp > PubImageData()
17
18 const int nDelayTimes = 2;
19 string sData_path = "../ch2/course2_hw_new/vio_data_simulation/bin/";
20 string sConfig_path = "../config/simluation_config.yaml";
21
22 std::shared_ptr<System> pSystem;
23
24 void PubImuData() {
25   string sImu_data_file = sData_path + "imu_pose.txt";
26   cout << "1 PubImuData start sImu_data_file: " << sImu_data_file << endl;
27   ifstream fsImu;
28   fsImu.open(sImu_data_file.c_str());
```

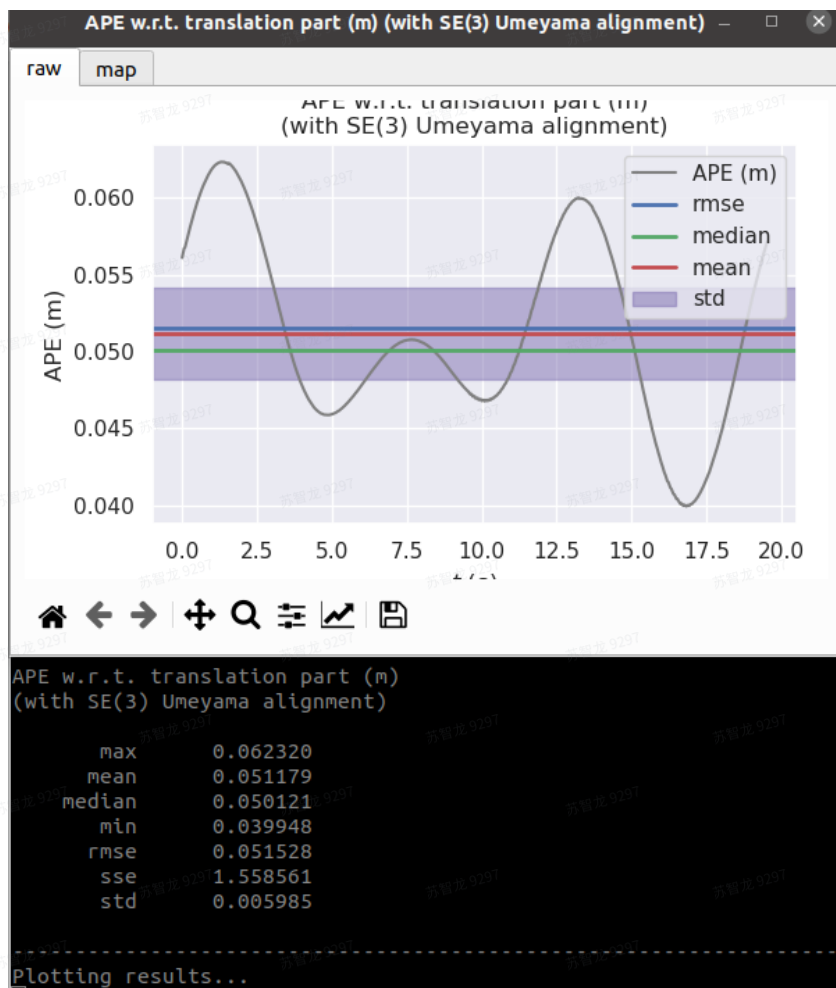
2.2 结果



2.3 evo traj



2.2 evo ape



3. 加噪声

- 修改 ch2 中数据生成中的 noise，并修改输入数据路径：

```
run_simulation.cpp 3, U × System.cpp 4, M run_euroc.cpp 3, M System.h 5, M
test > run_simulation.cpp > PubImuData()
19 string sData_path = "../ch2/court32_hw_new/vio_data_simulation/bin/";
20 string sConfig_path = "../config/simulation_config.yaml";
21
22 std::shared_ptr<System> pSystem;
23
24 void PubImuData() {
25     string sImu_data_file = sData_path + "imu_pose_noise.txt";
26     cout << "1 PubImuData start sImu_data_file: " << sImu_data_file << endl;
27     ifstream fsImu;
28     fsImu.open(sImu_data_file.c_str());
```

增加噪声后的运行结果如下：

3.1 小噪声

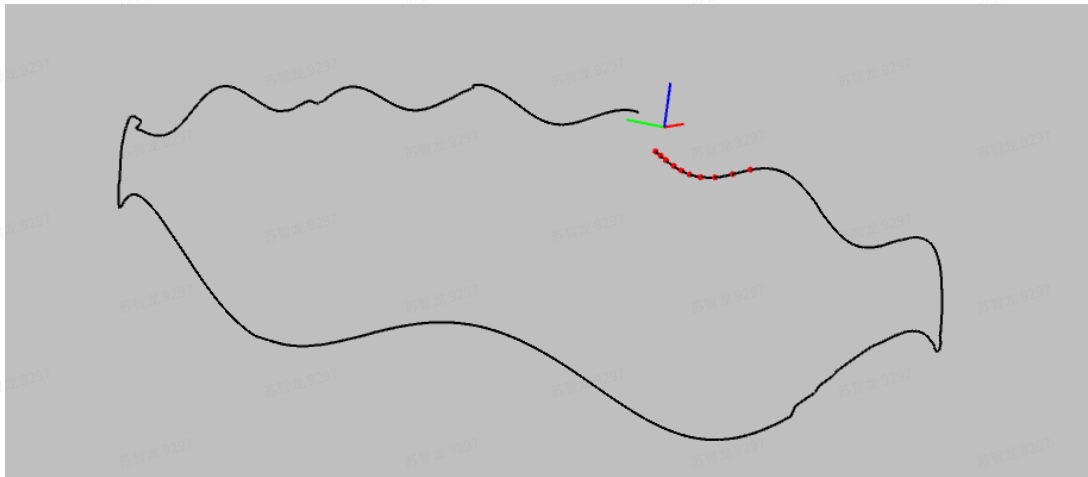
3.1.1 代码修改

```

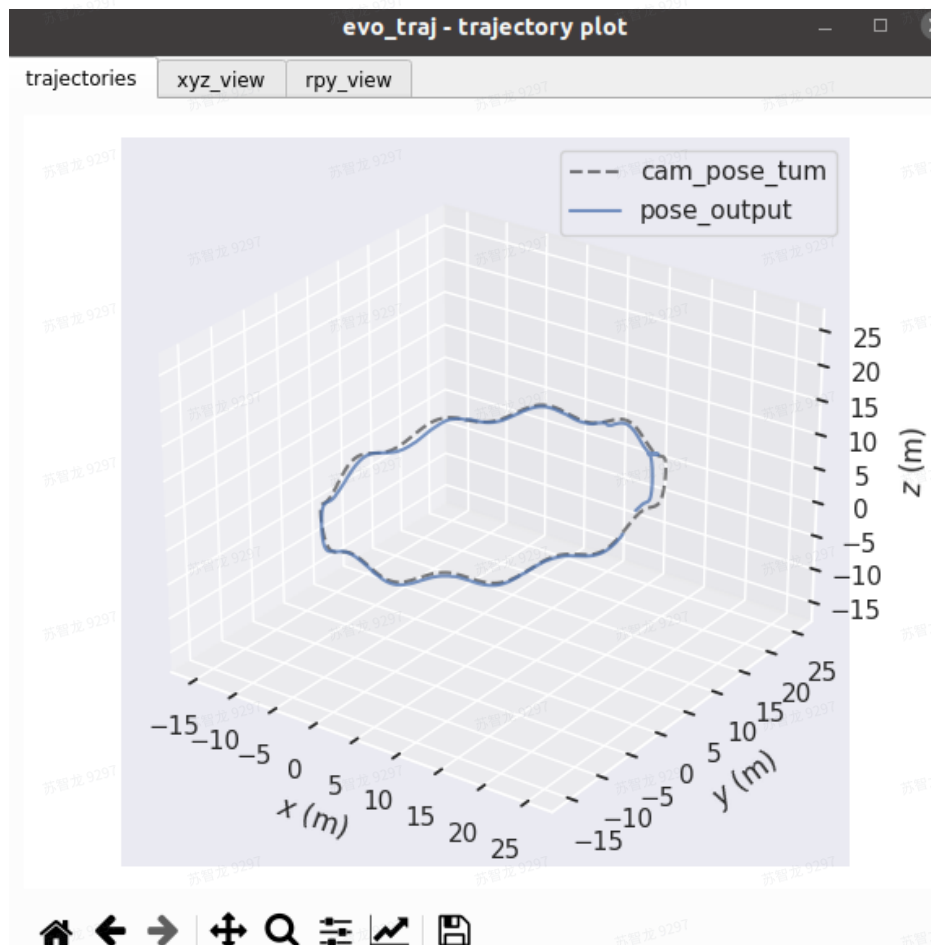
C param.h × CMakeLists.txt
src > C param.h > ...
23
24 // noise
25 double gyro_bias_sigma = 1.0e-6;
26 double acc_bias_sigma = 0.00001;
27
28 double gyro_noise_sigma = 0.0015; // rad/s
29 double acc_noise_sigma = 0.0019; // m/(
30
31 double pixel_noise = 1; // 1 pix

```

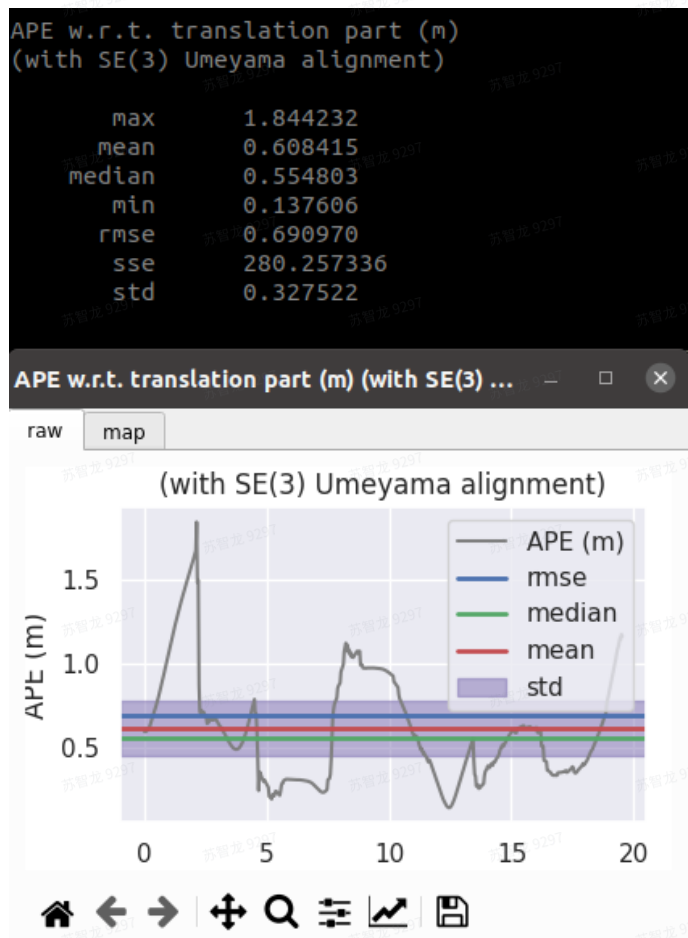
3.1.2 结果



3.1.3 evo traj



3.1.4 evo ape

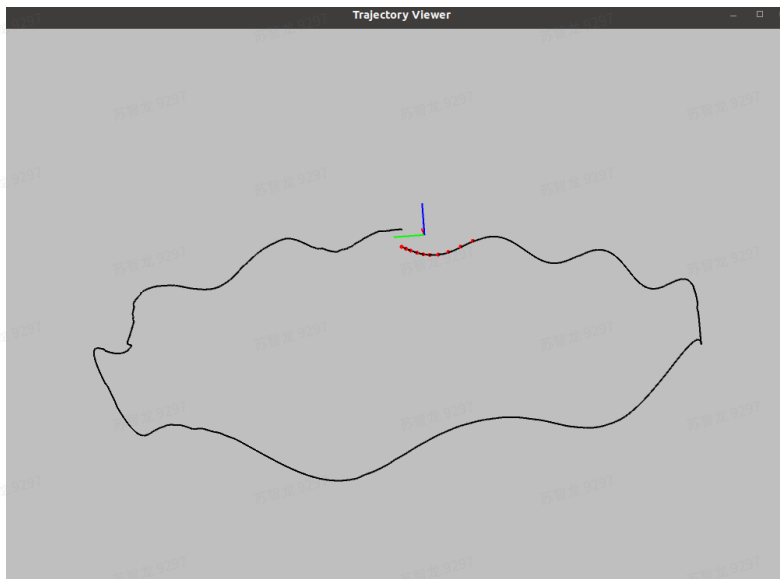


3.2 大噪声

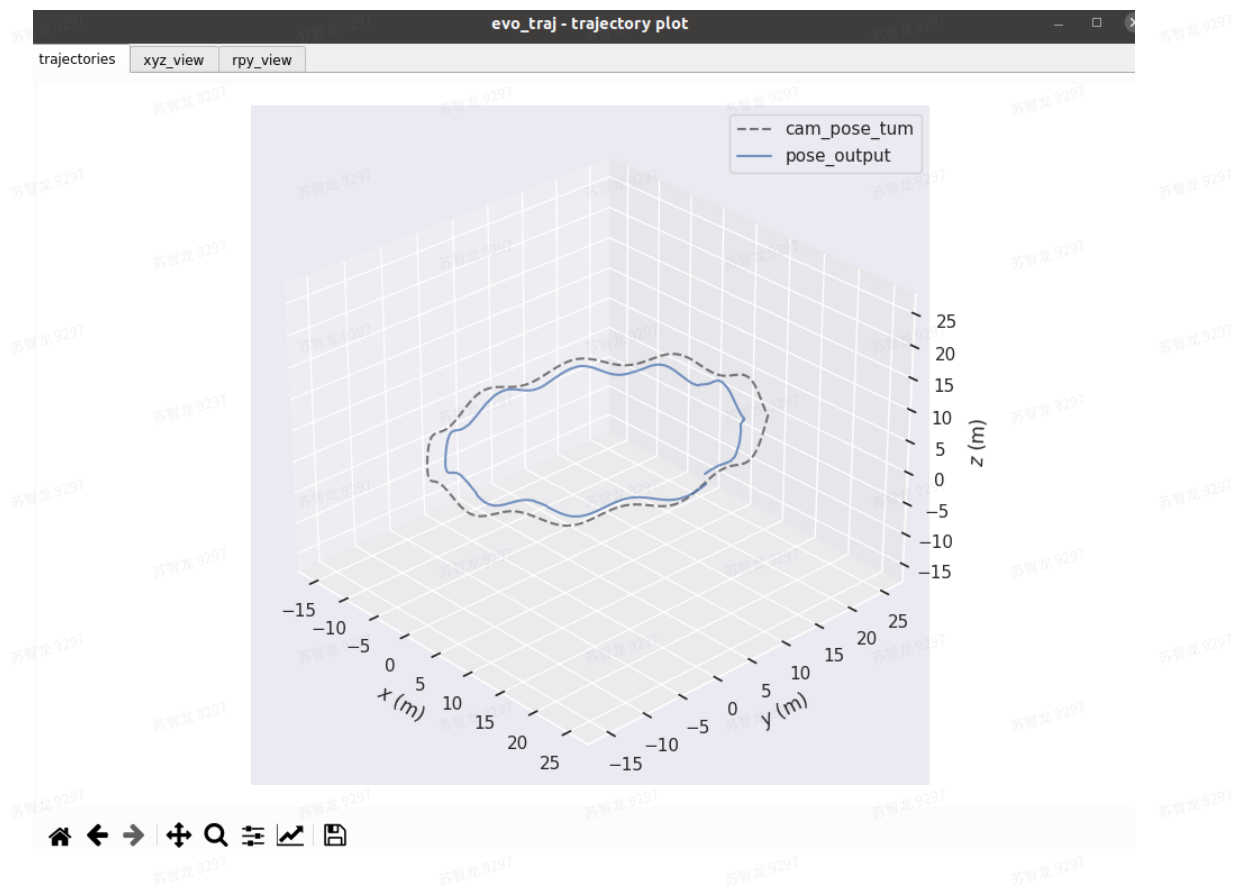
3.2.1 代码修改

```
src > C param.h > ...  
23  
24 // noise  
25 double gyro_bias_sigma = 1.0e-5;  
26 double acc_bias_sigma = 0.0001;  
27  
28 double gyro_noise_sigma = 0.015;  
29 double acc_noise_sigma = 0.019;
```

3.2.2 结果



3.2.3 evo traj



3.2.4 evo ape

