深蓝学院
shenlanxueyuan.com

手写VIO第七章作业分享

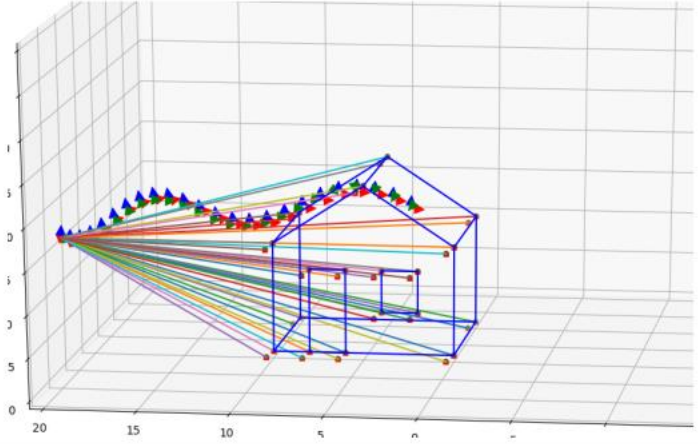
主讲人 啦啦啦

题目

深蓝学院 shenlanxueyuan.com

作业

1. 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。）



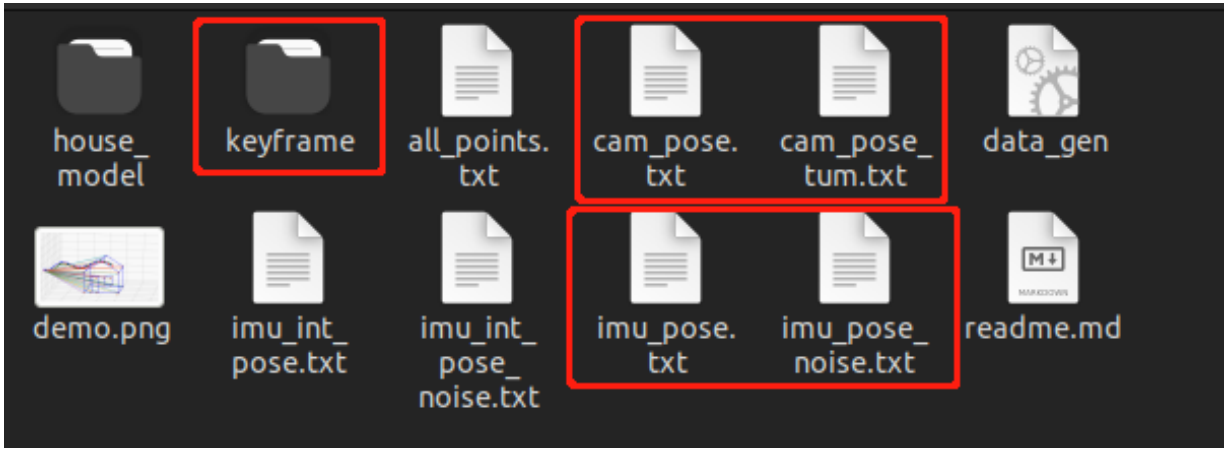
第一题

深蓝学院
shenlanxueyuan.com

方法：

- 第二章代码 - 生成的数据集格式、文件内容，需要确定的参数等等。
- 第七章代码 - 三个线程：每个部分的作用； 本次作业需要用到的部分。
- 进行代码融合。

第一题-第二章



注意 Readme.md 中的文件内容介绍。
本次需要用到的文件：

1. keyframe文件夹 中 all_points_XXX.txt 文件，里面保存了归一化坐标；
2. cam_pose.txt 和 cam_pose_tum.txt 相机位姿真值；
3. imu_pose.txt 无噪声 和 imu_pose_noise.txt 有噪声

第一题-第二章

还有对应的参数：大都在 param.h 头文件下，但 param.cpp 下也有部分。

param.h | param.cpp | draw_trajcory.py | draw_points.py

src > param.cpp > Param()

```cpp
//
// Created by hyj on 17-6-22.
//

#include "param.h"

Param::Param()
{
    Eigen::Matrix3d R;    // 把body坐标系朝向旋转一下,得到相机坐标系，好让它看到
    // 相机朝着轨迹里面看， 特征点在轨迹外部， 这里我们采用这个
    R << 0, 0, -1,
         -1, 0, 0,
         0, 1, 0;
    R_bc = R;
    t_bc = Eigen::Vector3d(0.05,0.04,0.03);

}
```

第一题-第七章

深蓝学院 shenlanxueyuan.com

首先明确代码每部分的作用：（ run_euroc.cpp ）

OK

1. PubImuData：获取imu数据(imu采集的数据)：时间戳、加速度、角速度的数据；以上数据存到imu_buf中。

没有光度怎么进行光流？

2. PubImageData：获取image数据(一帧帧的图像)，对图像进行均衡化、光流匹配、寻找新特征点等；输出xyz_uv_velocity (归一化坐标+像素坐标+xy上速度)

3. ProcessBackEnd：获取后一帧图像+帧间对应的imu数据； imu数据处理；image数据处理；非线性优化。 imu和image的输入数据即是1 2中的输出数据。

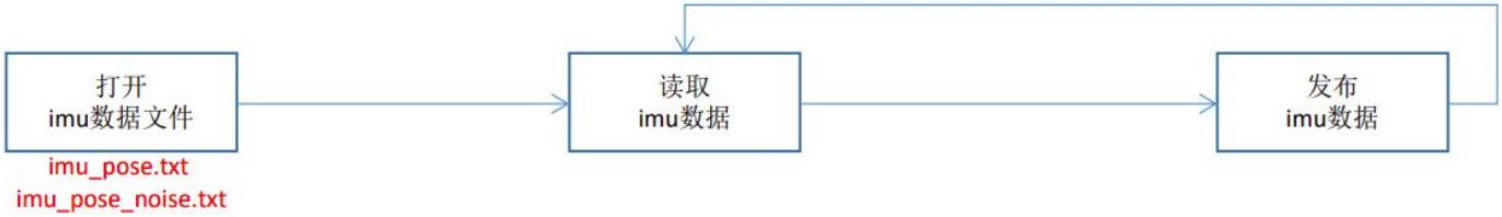
至此，我们大致明白了作业中所需修改的部分。
虽然ProcessBackEnd本次没有用到，但依旧需要仔细读懂，后面也会用得到。

第一题-第七章-流程图

深蓝学院 shenlanxueyuan.com

PubImuData：

主要流程：



fsImu.open(sImu_data_file.c_str());

ssImuData >> dStampNSec >> q.w() >> q.x() >> q.y() >> q.z() >> t(0) >> t(1) >> t(2) >> vGyr.x() >> vGyr.y() >> vGyr.z() >> vAcc.x() >> vAcc.y() >> vAcc.z();

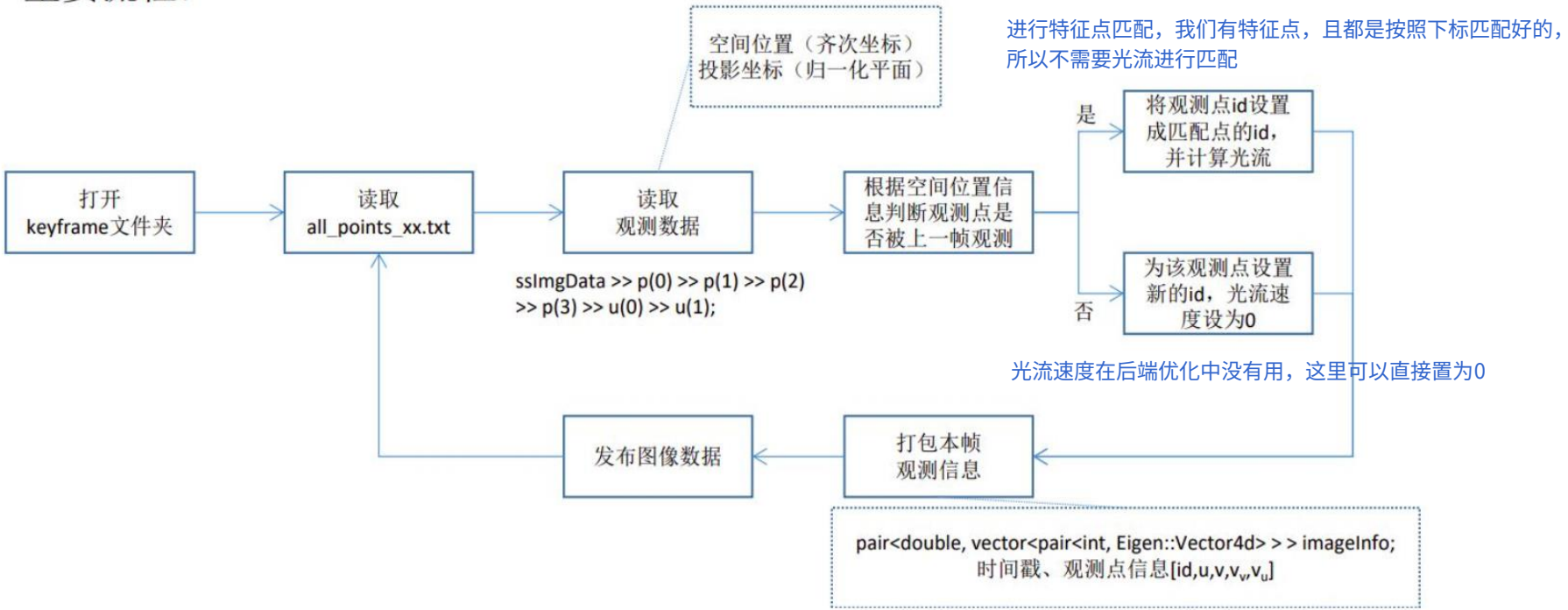
pSystem->PubImuData(dStampNSec, vGyr, vAcc);

第一题-第七章-流程图

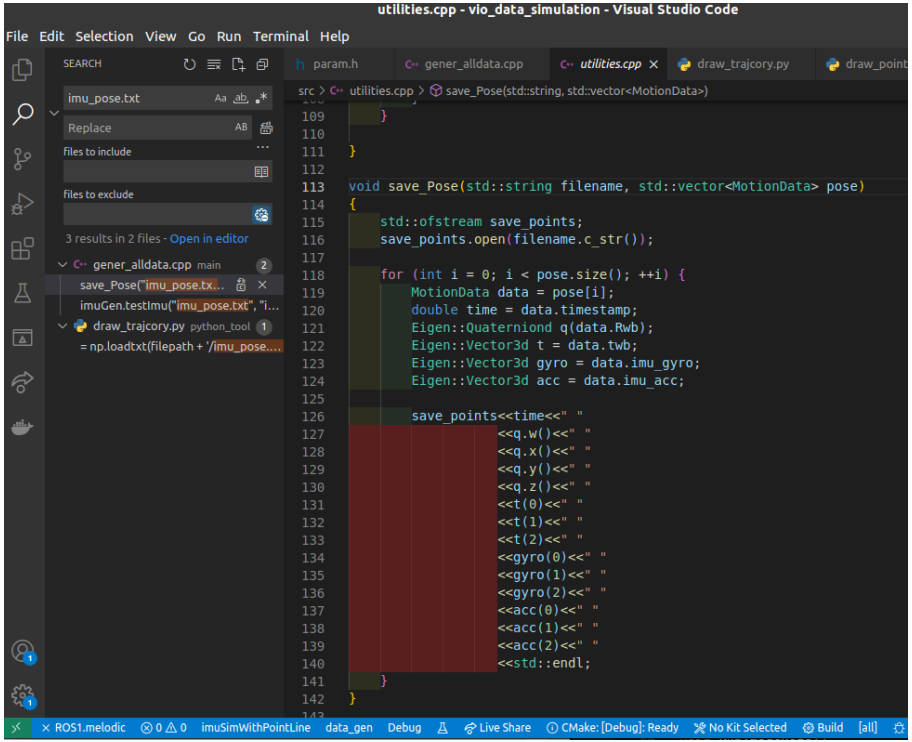
深蓝学院 shenlanxueyuan.com

PubImageData： 只有当前帧与上一帧都观测到空间位置，才能计算速度。

主要流程：



第一题-第七章



所需修改：PubImuData、PubImageData 这两个线程。（一个一个来）

PubImuData：

第二章中imu_pose.txt 和 imu_pose_noise.txt 的格式如图中所示：q和t是不需要的。

从euroc数据集的imu文件中读取数据，改为从第二章的imu文件中读取，修改读取格式即可。

第一题-PubImuData

深蓝学院 shenlanxueyuan.com

```
31  void PubImuData()
32  {
33      // string sImu_data_file = sConfig_path + "MH_05_imu0.txt";
34      string sImu_data_file = sData_path + "imu_pose.txt";
35      cout << "1 PubImuData start sImu_data_filea: " << sImu_data_file << endl;
36      ifstream fsImu;
37      fsImu.open(sImu_data_file.c_str());
38      if (!fsImu.is_open())
39      {
40          cerr << "Failed to open imu file! " << sImu_data_file << endl;
41          return;
42      }
43
44      std::string sImu_line;
45      double dStampNSec = 0.0;
46      Vector3d vAcc;
47      Vector3d vGyr;
48      // [new]
49      Vector4d q_wb;
50      Vector3d t_wb;
51      double imu_q_p;
52      while (std::getline(fsImu, sImu_line) && !sImu_line.empty()) // read imu data
53      {
54          std::istringstream ssImuData(sImu_line);
55          ssImuData >> dStampNSec >> q_wb(0) >> q_wb(1) >> q_wb(2) >> q_wb(3) >> t_wb(0) >> t_wb(1) >> t_wb(2)
56                    >> vGyr.x() >> vGyr.y() >> vGyr.z() >> vAcc.x() >> vAcc.y() >> vAcc.z();
57  //        cout << "Imu t: " << fixed << dStampNSec << " gyr: " << vGyr.transpose() << " acc: " << vAcc.transp
58          pSystem->PubImuData(dStampNSec, vGyr, vAcc);      You, now • Uncommitted changes
59          usleep(5000*nDelayTimes); // usleep-单位-微秒 (us) 10^{-6}次幂 || 200hz
60      }
61      fsImu.close();
62  }
```

ssImuData >> dStampNSec >> q_wb(0) >> q_wb(1) >> q_wb(2) >> q_wb(3) >> t_wb(0) >> t_wb(1) >> t_wb(2)
>> vGyr.x() >> vGyr.y() >> vGyr.z() >> vAcc.x() >> vAcc.y() >> vAcc.z();

第一题-PubImageData

所需修改：PubImuData、PubImageData 这两个线程。（一个一个来）

PubImageData： + System.cpp中的 PubImageData

原文件中输入：euroc 的数据集；输出：xyz_uv_velocity。

输入all_points_XXX.txt文件中的归一化数据：因此，我们需要计算像素uv和velocity。

uv的计算不必多说。 velocity的计算对 undistortedPoints() 中的源代码进行“魔改”即可。【这里也是需要大家对代码流程理解】

对于计算velocity的目的：能够起到优化td时间误差的作用。

第一题-PubImageData

方法很多，达到需求的效果即可。

```
82      std::string sImage_line;
83      double dStampNSec;
84      string sImgFileName;
85      //  [new]
86      u_int points_num = 0;       You, 3 minutes ago • Uncommitted changes
87      // cv::namedWindow("SOURCE IMAGE", CV_WINDOW_AUTOSIZE);
88      while (std::getline(fsImage, sImage_line) && !sImage_line.empty())
89      {
90          std::istringstream ssImgData(sImage_line);
91          ssImgData >> dStampNSec ;
92          // ****** [new] ******
93          string key_frame_PointsFiles = sData_path + "keyframe/all_points_" + to_string(points_num) + ".txt";
94  //      cout << "Image t : " << fixed << dStampNSec << " Name: " << key_frame_PointsFiles << endl;
95
96          ifstream fin(key_frame_PointsFiles);
97          if(!fin){
98              cerr << "keyframe/all_points_XX" << key_frame_PointsFiles << "not found" << endl;
99          }
100
101         // Mat img = imread(imagePath.c_str(), 0);
102         vector<Matrix<double,6,1>> xyz_uv_sum;
103         Matrix<double,6,1> xyz_uv;
104         while (!fin.eof())
105         {
106             /* code */ // xyz:4 uv:2
107             fin >> xyz_uv(0) >> xyz_uv(1) >> xyz_uv(2) >> xyz_uv(3) >> xyz_uv(4) >> xyz_uv(5);
108             xyz_uv_sum.push_back(xyz_uv);
109         }
110
111         pSystem->PubImageData(dStampNSec , xyz_uv_sum);
```

第一题-System::PubImageData

深蓝学院 shenlanxueyuan.com

```
161            prev_un_pts_map = cur_un_pts_map; //计算速度 前一帧更新
162
163    //              auto &un_pts = trackerData[i].cur_un_pts;
164    //              auto &cur_pts = trackerData[i].cur_pts;
165    //              auto &ids = trackerData[i].ids;
166    //              auto &pts_velocity = trackerData[i].pts_velocity;
167            for (unsigned int j = 0; j < img.size(); j++)
168            {
169
170                int p_id = j;
171                hash_ids[i].insert(p_id);
172                double x = img[j][4];
173                double y = img[j][5];
174                double z = 1;
175                double u = 460 * x + 255;
176                double v = 460 * y + 255;
177                feature_points->points.push_back(Vector3d(x, y, z));
178                feature_points->id_of_point.push_back(p_id * NUM_OF_CAM + i);
179                feature_points->u_of_point.push_back(u);
180                feature_points->v_of_point.push_back(v);
181                feature_points->velocity_x_of_point.push_back(0);
182                feature_points->velocity_y_of_point.push_back(0);      You, now • Uncommi
183                // feature_points->velocity_x_of_point.push_back(pts_velocity[j].x);
184                // feature_points->velocity_y_of_point.push_back(pts_velocity[j].y);
185            }
186
```

第一题-第七章

深蓝学院 shenlanxueyuan.com

主要内容修改完毕，但还要符合题目要求：1. 参数设置 2.输出轨迹结果

1.参数修改：修改config下的配置文件yaml ： 和第二章一一对应好即可。

2.输出轨迹结果：程序已经写好了一个输出文件 pose_output.txt - ofs_pose 找到如下文件输出内容，改成对应格式即可 (包含q的虚实部的顺序，t，时间戳等的顺序)

```
426
427         // 非线性优化
428         if (estimator.solver_flag == Estimator::SolverFlag::NON_LINEAR)
429         {
430             Vector3d p_wi;
431             Quaterniond q_wi;
432             q_wi = Quaterniond(estimator.Rs[WINDOW_SIZE]);
433             p_wi = estimator.Ps[WINDOW_SIZE];
434             vPath_to_draw.push_back(p_wi);
435             double dStamp = estimator.Headers[WINDOW_SIZE];
436             cout << "1 BackEnd processImage dt: " << fixed << t_processImage.toc() << " stamp: " <<  dStamp << " p_w
437             ofs_pose << fixed << dStamp << " " << p_wi(0) << " " << p_wi(1) << " " << p_wi(2) << " "
438                      << q_wi.w() << " " << q_wi.x() << " " << q_wi.y() << " " << q_wi.z() << endl;
```

第一题-第七章-配置文件

深蓝学院
shenlanxueyuan.com

```
25  # 【修改如下】
26  image_width: 640
27  image_height: 640
28  distortion_parameters: # 用不到畸变矫正
29      k1: 0
30      k2: 0
31      p1: 0
32      p2: 0
33  projection_parameters:
34      fx: 460
35      fy: 460
36      cx: 255
37      cy: 255
38
```

```
61  # 【修改如下】
62  #Rotation from camera frame to imu frame, imu^R_cam
63  extrinsicRotation: !!opencv-matrix
64     rows: 3
65     cols: 3
66     dt: d
67     data: [0, 0, -1,
68            -1, 0, 0,
69            0, 1, 0]
70  #Translation from camera frame to imu frame, imu^T_cam
71  extrinsicTranslation: !!opencv-matrix
72     rows: 3
73     cols: 1
74     dt: d
75     data: [0.05,0.04,0.03]
```

```
108   acc_n: 0.019
109   gyr_n: 0.015
110   acc_w: 0.0001
111   gyr_w: 1.0e-5
```

第一题-第七章-输出轨迹结果

深蓝学院 shenlanxueyuan.com

```cpp
void save_Pose_asTUM2(std::string filename, std::vector<MotionData> pose)
{
    std::ofstream save_points;
    save_points.setf(std::ios::fixed, std::ios::floatfield);
    save_points.open(filename.c_str());

    for (int i = 0; i < pose.size(); ++i) {
        MotionData data = pose[i];
        double time = data.timestamp;
        Eigen::Quaterniond q(data.q);
        Eigen::Vector3d t = data.p;

        save_points.precision(9);
        save_points <<time<<" ";
        save_points.precision(5);
        save_points <<t(0)<<" "
                    <<t(1)<<" "
                    <<t(2)<<" "
                    <<q.x()<<" "
                    <<q.y()<<" "
                    <<q.z()<<" "
                    <<q.w() <<std::endl;
    }

}
```

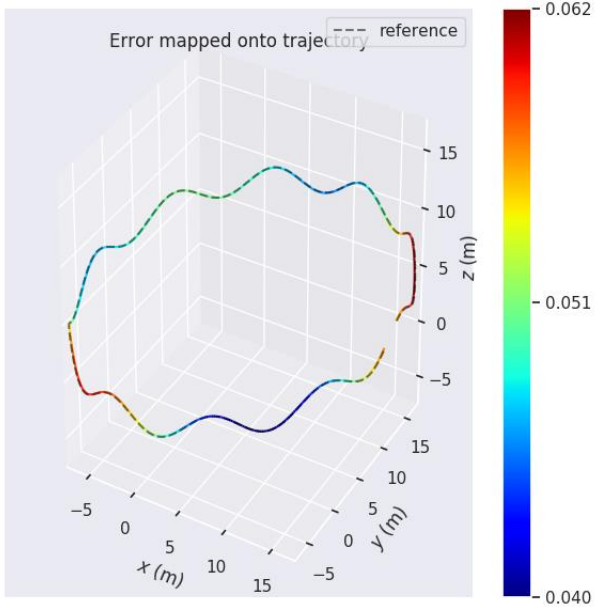
第一题-结果

结果的展示：

问题

实际上，IMU 传感器获取的数据为离散采样，离散和连续高斯白噪声存在何种关系？

无噪声



获取的imu数据为离散数据。这里的噪声我用的离散化之后的结果。

[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]

连续的

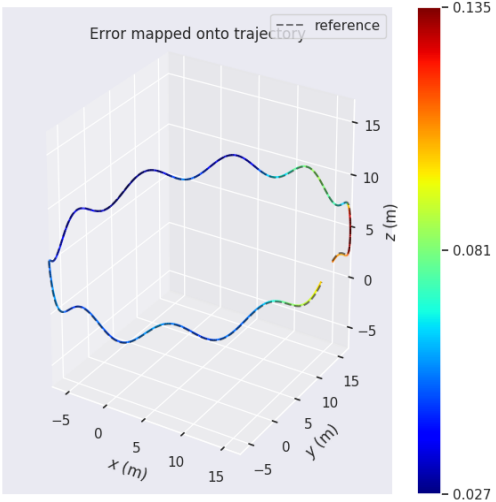
```
//*****************************************
    double times_exp = 1; //  【new】

    ACC_N *= times_exp ; //  【连续噪声】
    ACC_W *= times_exp ;
    GYR_N *= times_exp ;
    GYR_W *= times_exp ;

    ACC_N *= sqrt(200); //  【离散噪声】
    ACC_W /= sqrt(200);
    GYR_N *= sqrt(200);
    GYR_W /= sqrt(200);
//*****************************************
```

第一题-结果：不同噪声

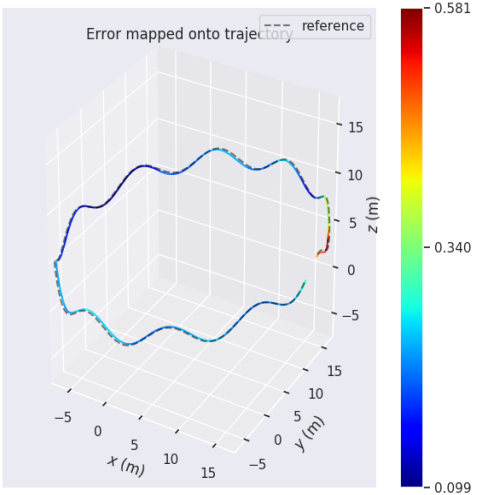
深蓝学院 shenlanxueyuan.com

	max	mean	min	rmse
nosie-no	0.061820	0.050894	0.039767	0.051236
nosie*1	0.135345	0.060929	0.026851	0.066198
noise * 10	0.581123	0.239262	0.099104	0.256938
noise * 30	0.905109	0.474739	0.113542	0.502929

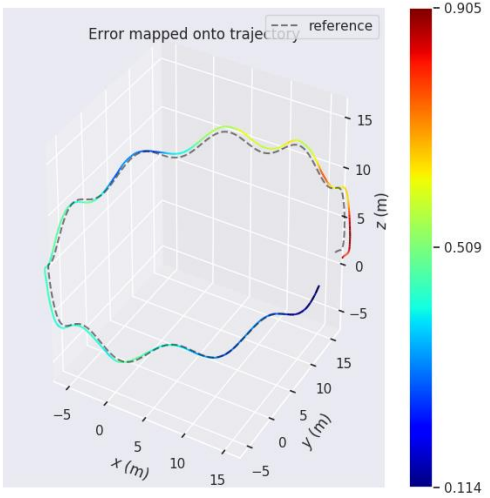
噪声*1



噪声*10



噪声*30



第一题-结果

`noise * 100`：噪声太大，导致曲线形状都变了，也就没了比较的意义。

在线问答

深蓝学院
shenlanxueyuan.com

深蓝学院
shenlanxueyuan.com

感谢各位聆听

Thanks for Listening