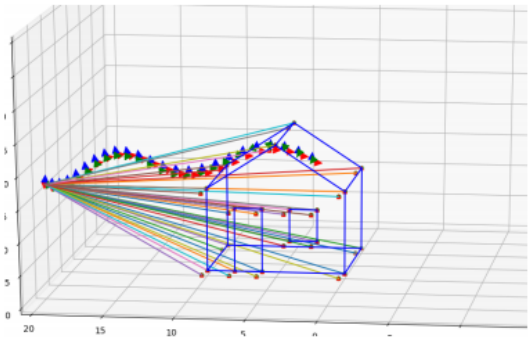
作业

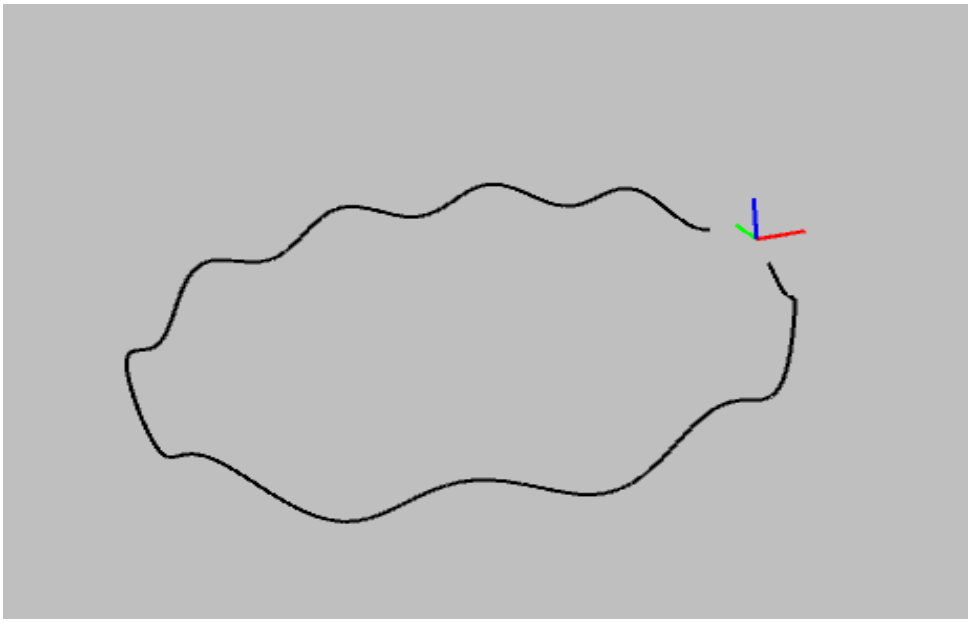
❶ 将第二讲的仿真数据集（视觉特征，imu 数据）接入我们的 VINS 代码，并运行出轨迹结果。

- 仿真数据集无噪声
- 仿真数据集有噪声（对于同一噪声数据，请比较不同配置文件对定位精度的影响，即修改 config 中 yaml 文件里 imu noise 的大小。）

注意：评估不同参数配置对精度影响时，请给出数据对比．建议采用 evo 工具（https://github.com/MichaelGrupp/evo）对轨迹精度进行评估，仿真数据的轨迹真值已给出。



1、 仿真数据集无噪声：



2、 仿真数据集加入噪声：

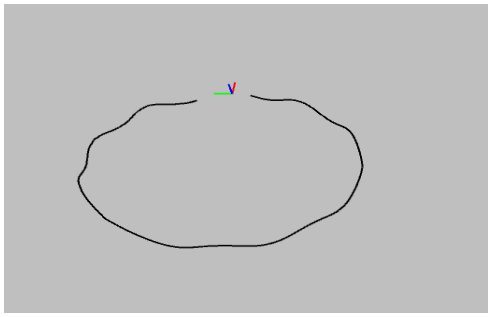
改 pubimudata 函数：

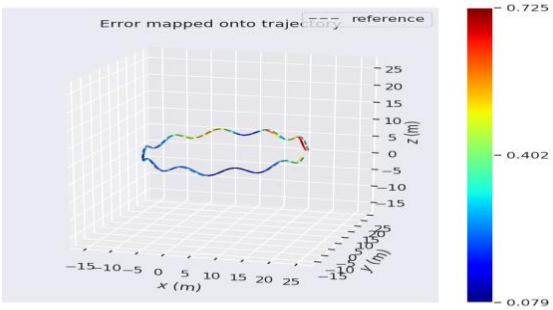
```
void PubImuData()
{
    string sImu_data_file = sData_path + "imu_pose_noise.txt";
    cout << "1 PubImuData start sImu_data_filea: " << sImu_data_file << endl;
    ifstream fsImu;
    fsImu.open(sImu_data_file.c_str());
    if (!fsImu.is_open()){
        cerr << "Failed to open imu file! " << sImu_data_file << endl;
        return;
    }

    std::string sImu_line;
    double dStampNSec = 0.0;
    Vector3d vAcc;
    Vector3d vGyr;
    Eigen::Quaterniond Qwb;
    Vector3d position;

    while (std::getline(fsImu, sImu_line) && !sImu_line.empty()) // read imu data
    {
        std::istringstream ssImuData(sImu_line);
        ssImuData >> dStampNSec >>Qwb.w()>>Qwb.x()>>Qwb.y()>>Qwb.z()>>position.x()>>position.y()>>position.z() >>vGyr.x() >> vGyr.y() >> vGyr.z() >> vAcc.x() >> vAcc.y() >> vAcc.z();
        // cout << "Imu t: " << fixed << dStampNSec << " gyr: " << vGyr.transpose() << " acc: " << vAcc.transpose() << endl;
        pSystem->PubImuData(dStampNSec, vGyr, vAcc);
        usleep(5000 * nDelayTimes);
    }
    fsImu.close();
}
```

改 pubimagedata 函数：

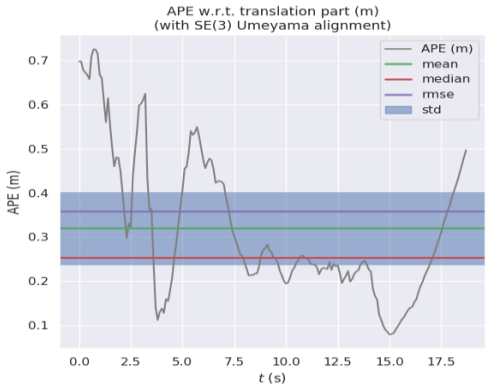
```
void PubImageData()
{
    string sImage_file = sData_path + "cam_pose_tum.txt";
    cout << "1 PubImageData start sImage_file: " << sImage_file << endl;
    ifstream fsImage;
    fsImage.open(sImage_file.c_str());
    if (!fsImage.is_open()){
        cerr << "Failed to open image file! " << sImage_file << endl;
        return;
    }

    std::string sImage_line;
    double dStampNSec;
    int n = 0;
    // cv::namedWindow("SOURCE IMAGE", CV_WINDOW_AUTOSIZE);
    while (std::getline(fsImage, sImage_line) && !sImage_line.empty())
    {
        vector<cv::Point2f> feacturePoint;
        std::istringstream ssImageData(sImage_line);
        ssImageData >> dStampNSec ;
        // cout << "Image t : " << fixed << dStampNSec <<endl;

        std::stringstream filename1;
        filename1<<"keyframe/all_points_"<<n<<".txt";
        n = n+1;
        string featurePath = sData_path + filename1.str();
        // cout<<"featurePath: "<<featurePath<<endl;

        ifstream fsFeature;
        fsFeature.open(featurePath.c_str());
        if (!fsFeature.is_open()){
            cerr << "Failed to open image file! " << featurePath << endl;
            return;
        }
        std::string sFeature_line;
        while(std::getline(fsFeature,sFeature_line) && !sFeature_line.empty()){
            cv::Point2f featureTemp;
            Vector4d temp;
            std::istringstream ssFeatureData(sFeature_line);
            double u = 0.0;
            double v = 0.0;
            ssFeatureData>>temp.x()>>temp.y()>>temp.z()>>temp.w()>>u>>v;
            featureTemp.x = u;featureTemp.y = v;
            feacturePoint.push_back(featureTemp);
        }
        // cout<<"feacturePoint.size(): "<<feacturePoint.size()<<endl;
        pSystem->PubImageData(dStampNSec, feacturePoint);
        usleep(50000*nDelayTimes);
    }
    fsImage.close();
}
```

使用带噪声的 IMU 数据，配置文件参数 acc_n:0.8 gyr_n:0.4 时，生成轨迹和真值轨迹的比对如下：

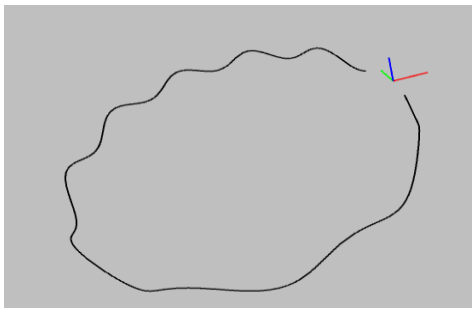


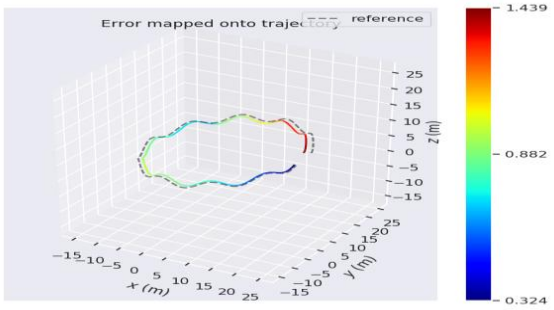



```
Loaded 600 stamps and poses from: cam_pose_tum.txt
Loaded 188 stamps and poses from: pose_output_noise1.txt
Synchronizing trajectories...
Found 188 of max. 600 possible matching timestamps between...
       cam_pose_tum.txt
and:    pose_output_noise1.txt
..with max. time diff.: 0.01 (s) and time offset: 0.0 (s).
-------------------------------------------------------------------------------
Aligning using Umeyama's method...
Rotation of alignment:
[[ 9.96283394e-01 -8.61326272e-02  7.54904662e-04]
 [ 8.61287051e-02  9.96275060e-01  4.22514739e-03]
 [-1.11601573e-03 -4.14442522e-03  9.99990789e-01]]
Translation of alignment:
[19.50551371  7.10679275  5.49373889]
Scale correction: 1.0
-------------------------------------------------------------------------------
Compared 188 absolute pose pairs.
Calculating APE for translation part pose relation...
-------------------------------------------------------------------------------
APE w.r.t. translation part (m)
(with SE(3) Umeyama alignment)

       max      0.725407
      mean      0.319667
    median      0.253968
       min      0.078633
      rmse      0.358653
       sse      24.182825
       std      0.162620

-------------------------------------------------------------------------------
```



使用带噪声的 IMU 数据，配置文件参数 acc_n:1.0,gyr_n:1.0 时，生成轨迹和真值轨迹的比对如下：





```
Loaded 600 stamps and poses from: cam_pose_tum.txt
Loaded 188 stamps and poses from: pose_output_noise.txt
Synchronizing trajectories...
Found 188 of max. 600 possible matching timestamps between...
        cam_pose_tum.txt
and:    pose_output_noise.txt
..with max. time diff.: 0.01 (s) and time offset: 0.0 (s).
-------------------------------------------------------------------------------
Aligning using Umeyama's method...
Rotation of alignment:
[[ 0.99805855 -0.06227235 -0.00113119]
 [ 0.06227411  0.99805781  0.0015975 ]
 [ 0.00102951 -0.00166484  0.99999808]]
Translation of alignment:
[18.68878962  5.96102271  5.71172069]
Scale correction: 1.0
-------------------------------------------------------------------------------
Compared 188 absolute pose pairs.
Calculating APE for translation part pose relation...
-------------------------------------------------------------------------------
APE w.r.t. translation part (m)
(with SE(3) Umeyama alignment)

       max      1.438891
      mean      0.896164
    median      0.889697
       min      0.324319
      rmse      0.934663
       sse      164.236011
       std      0.265491

-------------------------------------------------------------------------------
```