Course7_homework_vins_data_input

0. Summary

（1）增加关键帧的数量，i.e.加大.yaml 中的 freq，可以提高轨迹的完整性，因为 freq 增加后，处理帧数增加，在开始和结束时间段计算更多的 pose;

对于无噪声的情况，freq 的改变只是改变的轨迹的完整性，而 cam_pose 的结果是相同的；

（2）对于有噪声的情况，发现增加.yaml 中的 freq，可以减小误差；

（3）发现改变 run_euroc.cpp 中的 usleep()，并且 freq=28 时，有时候会导致 imu 和 image 数据之间不同步，结果轨迹误差很大，也有轨迹缺失的情况；后减小 PubImuData()之间的 usleep()，让 imu 数据能在 image 数据之前 publish，就没有这个轨迹缺失现象。

这个现象的原因：

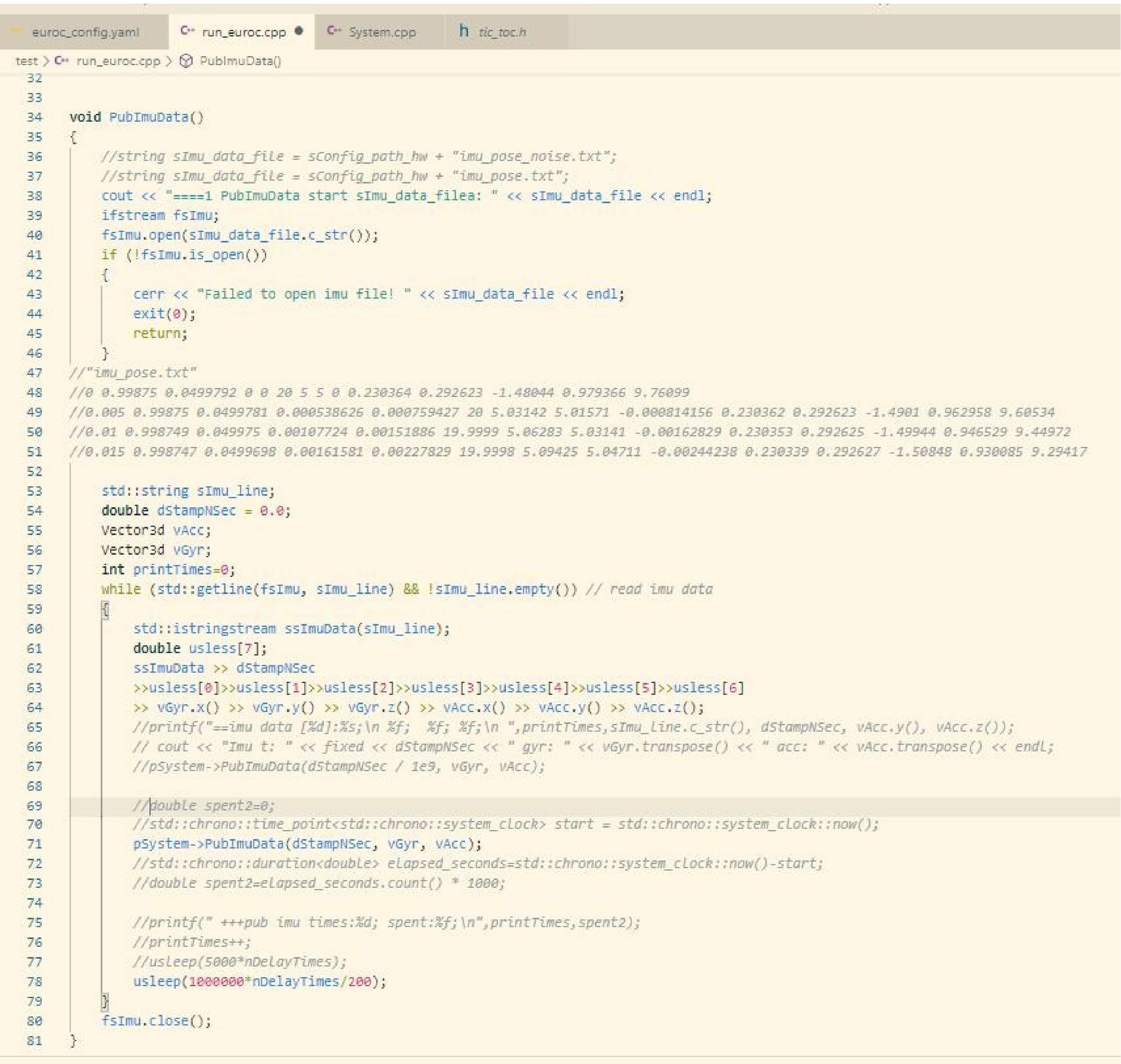
[a] 仿真程序和实际数据的产生时序有区别；

[b] Vins 的程序设计似乎有问题，存在很多锁，感觉可以用很长的循环队列保存 imu 和 image，保证数据时序和程序处理时序不出现重叠，那么可以去掉锁；

[c] Vins 程序的 freq 和 FREQ 判断逻辑也过于复杂，容易出现 imu 和 image 数据之间不同步；

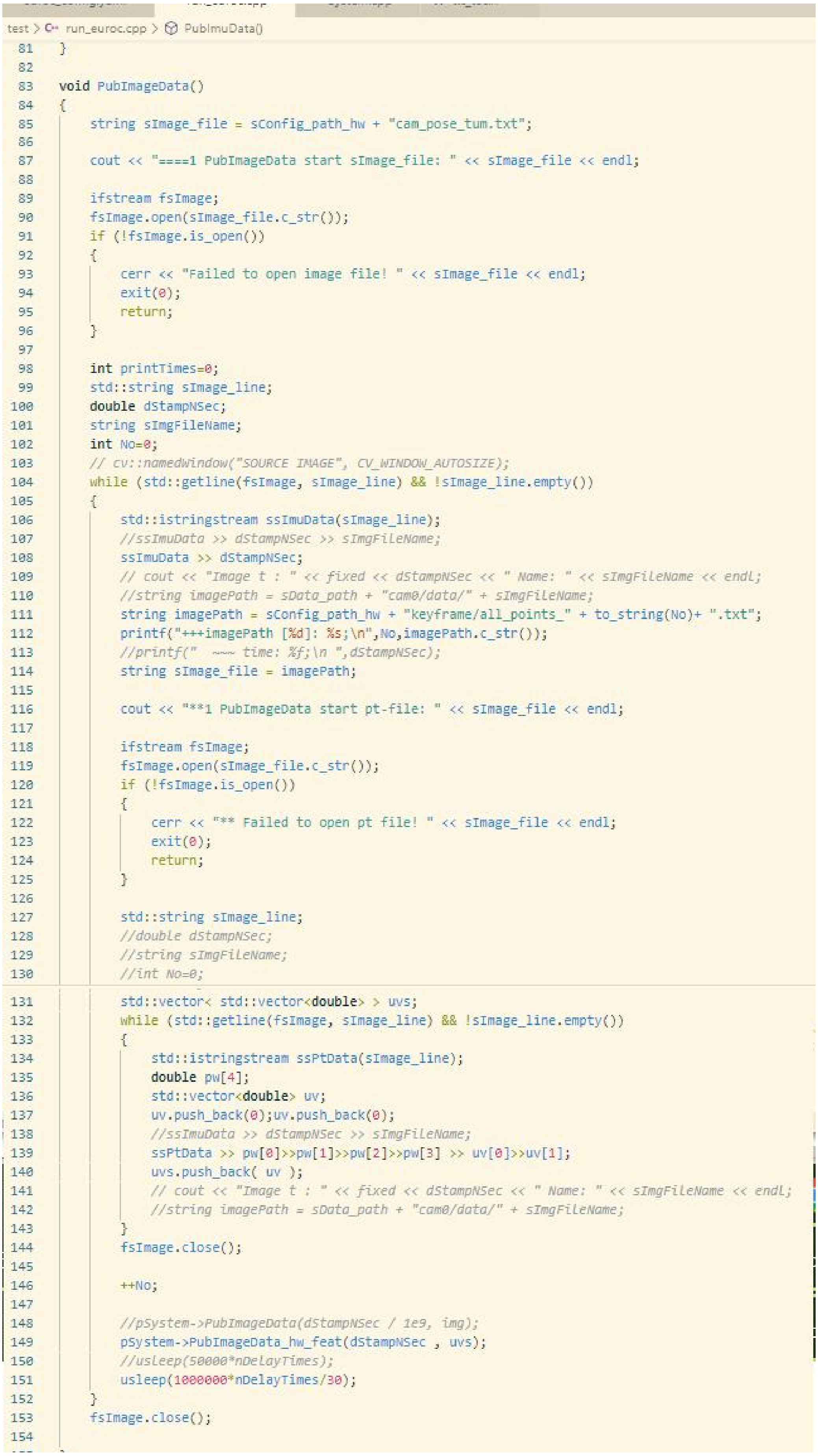
1. Code

(1) 发布 imu 数据



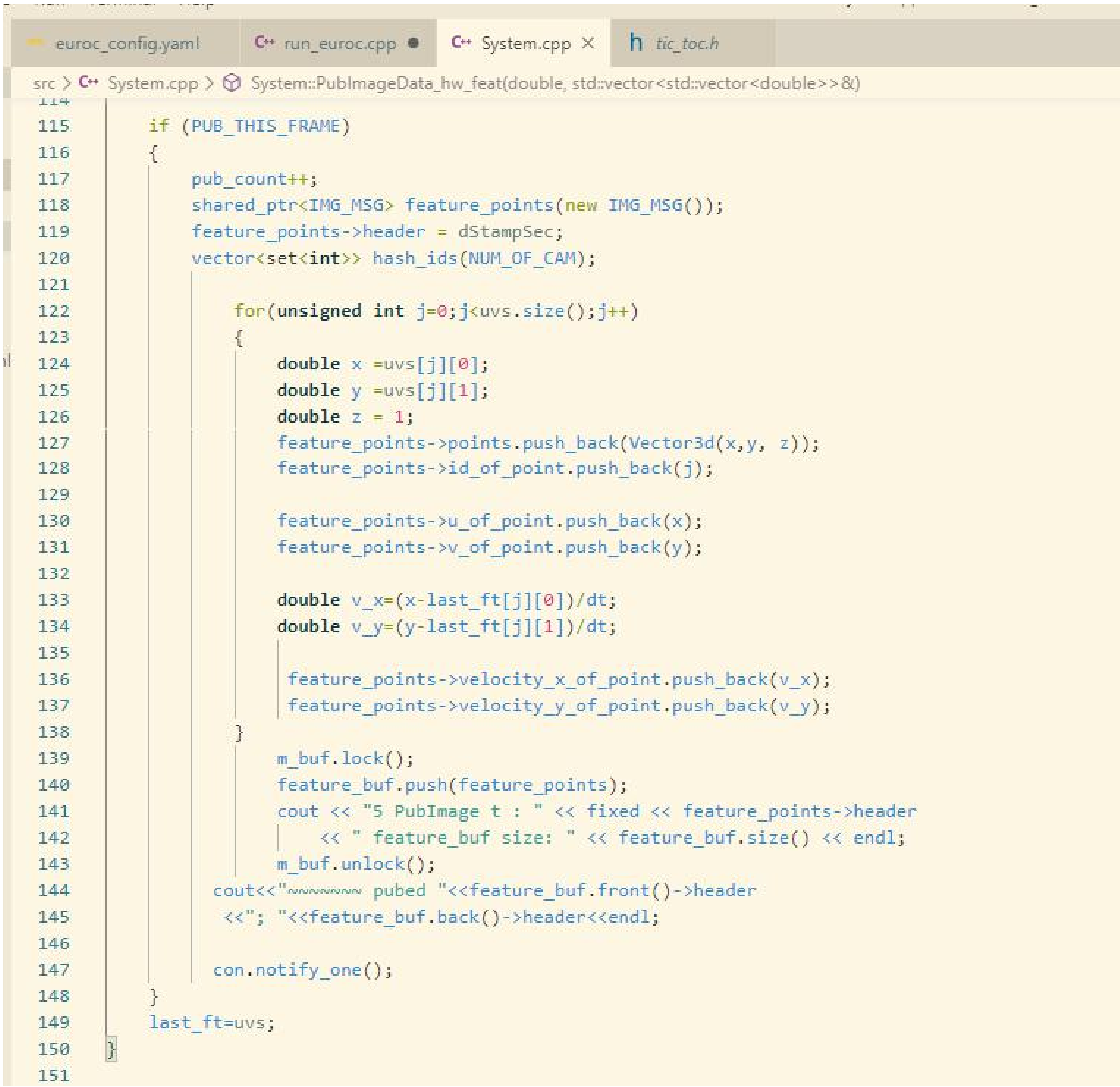
```cpp


void PubImuData()
{
    //string sImu_data_file = sConfig_path_hw + "imu_pose_noise.txt";
    //string sImu_data_file = sConfig_path_hw + "imu_pose.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;
        exit(0);
        return;
    }
//"imu_pose.txt"
//0 0.99875 0.0499792 0 0 20 5 5 0 0.230364 0.292623 -1.48044 0.979366 9.76099
//0.005 0.99875 0.0499781 0.000538626 0.000759427 20 5.03142 5.01571 -0.000814156 0.230362 0.292623 -1.4901 0.962958 9.60534
//0.01 0.998749 0.049975 0.00107724 0.00151886 19.9999 5.06283 5.03141 -0.00162829 0.230353 0.292625 -1.49944 0.946529 9.44972
//0.015 0.998747 0.0499698 0.00161581 0.00227829 19.9998 5.09425 5.04711 -0.00244238 0.230339 0.292627 -1.50848 0.930085 9.29417

    std::string sImu_line;
    double dStampNSec = 0.0;
    Vector3d vAcc;
    Vector3d vGyr;
    int printTimes=0;
    while (std::getline(fsImu, sImu_line) && !sImu_line.empty()) // read imu data
    {
        std::istringstream ssImuData(sImu_line);
        double usless[7];
        ssImuData >> dStampNSec
        >>usless[0]>>usless[1]>>usless[2]>>usless[3]>>usless[4]>>usless[5]>>usless[6]
        >> vGyr.x() >> vGyr.y() >> vGyr.z() >> vAcc.x() >> vAcc.y() >> vAcc.z();
        //printf("==imu data [%d]:%s;\n %f;  %f; %f;\n ",printTimes,sImu_line.c_str(), dStampNSec, vAcc.y(), vAcc.z());
        // cout << "Imu t: " << fixed << dStampNSec << " gyr: " << vGyr.transpose() << " acc: " << vAcc.transpose() << endl;
        //pSystem->PubImuData(dStampNSec / 1e9, vGyr, vAcc);

        //double spent2=0;
        //std::chrono::time_point<std::chrono::system_clock> start = std::chrono::system_clock::now();
        pSystem->PubImuData(dStampNSec, vGyr, vAcc);
        //std::chrono::duration<double> elapsed_seconds=std::chrono::system_clock::now()-start;
        //double spent2=elapsed_seconds.count() * 1000;

        //printf(" +++pub imu times:%d; spent:%f;\n",printTimes,spent2);
        //printTimes++;
        //usleep(5000*nDelayTimes);
        usleep(1000000*nDelayTimes/200);
    }
    fsImu.close();
}
```

(2) 发布 image 数据



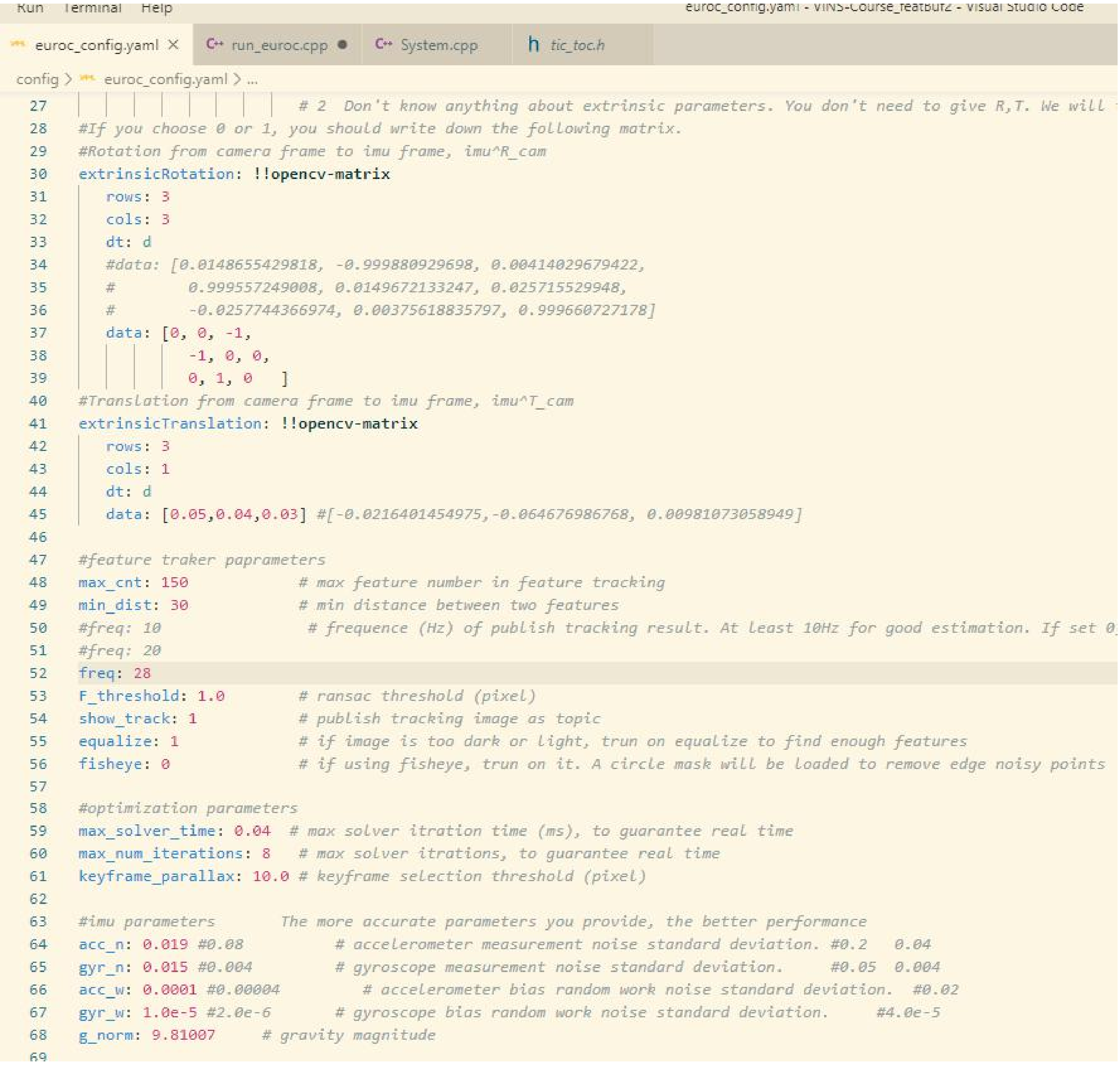
```cpp
}

void PubImageData()
{
    string sImage_file = sConfig_path_hw + "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;
        exit(0);
        return;
    }

    int printTimes=0;
    std::string sImage_line;
    double dStampNSec;
    string sImgFileName;
    int No=0;
    // cv::namedWindow("SOURCE IMAGE", CV_WINDOW_AUTOSIZE);
    while (std::getline(fsImage, sImage_line) && !sImage_line.empty())
    {
        std::istringstream ssImuData(sImage_line);
        //ssImuData >> dStampNSec >> sImgFileName;
        ssImuData >> dStampNSec;
        // cout << "Image t : " << fixed << dStampNSec << " Name: " << sImgFileName << endl;
        //string imagePath = sData_path + "cam0/data/" + sImgFileName;
        string imagePath = sConfig_path_hw + "keyframe/all_points_" + to_string(No)+ ".txt";
        printf("+++imagePath [%d]: %s;\n",No,imagePath.c_str());
        //printf("  ~~~ time: %f;\n ",dStampNSec);
        string sImage_file = imagePath;

        cout << "**1 PubImageData start pt-file: " << sImage_file << endl;

        ifstream fsImage;
        fsImage.open(sImage_file.c_str());
        if (!fsImage.is_open())
        {
            cerr << "** Failed to open pt file! " << sImage_file << endl;
            exit(0);
            return;
        }

        std::string sImage_line;
        //double dStampNSec;
        //string sImgFileName;
        //int No=0;
        std::vector< std::vector<double> > uvs;
        while (std::getline(fsImage, sImage_line) && !sImage_line.empty())
        {
            std::istringstream ssPtData(sImage_line);
            double pw[4];
            std::vector<double> uv;
            uv.push_back(0);uv.push_back(0);
            //ssImuData >> dStampNSec >> sImgFileName;
            ssPtData >> pw[0]>>pw[1]>>pw[2]>>pw[3] >> uv[0]>>uv[1];
            uvs.push_back( uv );
            // cout << "Image t : " << fixed << dStampNSec << " Name: " << sImgFileName << endl;
            //string imagePath = sData_path + "cam0/data/" + sImgFileName;
        }
        fsImage.close();

        ++No;

        //pSystem->PubImageData(dStampNSec / 1e9, img);
        pSystem->PubImageData_hw_feat(dStampNSec , uvs);
        //usleep(50000*nDelayTimes);
        usleep(1000000*nDelayTimes/30);
    }
    fsImage.close();

}
```

(3) 处理 imu 和 image 数据


euroc_config.yaml
run_euroc.cpp
System.cpp
tic_toc.h
src > System.cpp > System::PubImageData_hw_feat(double, std::vector<std::vector<double>>&)

```
    if (PUB_THIS_FRAME)
    {
        pub_count++;
        shared_ptr<IMG_MSG> feature_points(new IMG_MSG());
        feature_points->header = dStampSec;
        vector<set<int>> hash_ids(NUM_OF_CAM);

            for(unsigned int j=0;j<uvs.size();j++)
            {
                double x =uvs[j][0];
                double y =uvs[j][1];
                double z = 1;
                feature_points->points.push_back(Vector3d(x,y, z));
                feature_points->id_of_point.push_back(j);

                feature_points->u_of_point.push_back(x);
                feature_points->v_of_point.push_back(y);

                double v_x=(x-last_ft[j][0])/dt;
                double v_y=(y-last_ft[j][1])/dt;

                 feature_points->velocity_x_of_point.push_back(v_x);
                 feature_points->velocity_y_of_point.push_back(v_y);
            }
                m_buf.lock();
                feature_buf.push(feature_points);
                cout << "5 PubImage t : " << fixed << feature_points->header
                    << " feature_buf size: " << feature_buf.size() << endl;
                m_buf.unlock();
          cout<<"~~~~~~~~ pubed "<<feature_buf.front()->header
           <<"; "<<feature_buf.back()->header<<endl;

          con.notify_one();
    }
    last_ft=uvs;
}
```

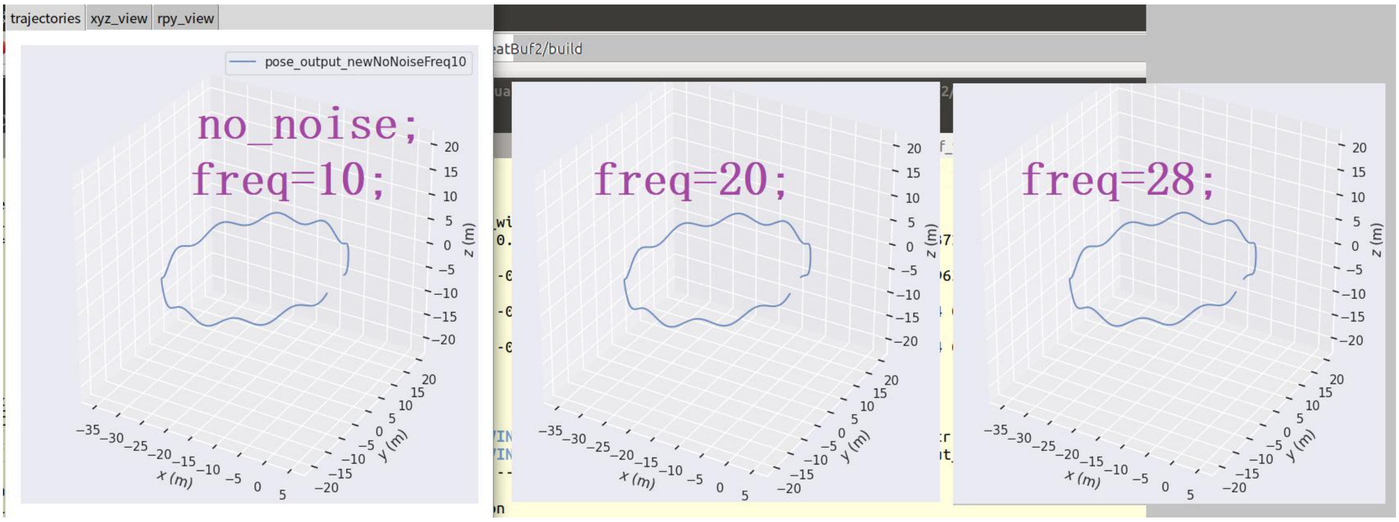
（4）.yaml


Run   Terminal   Help
euroc_config.yaml - VINS-Course_featBuf2 - Visual Studio Code
euroc_config.yaml
run_euroc.cpp
System.cpp
tic_toc.h
config > euroc_config.yaml > ...

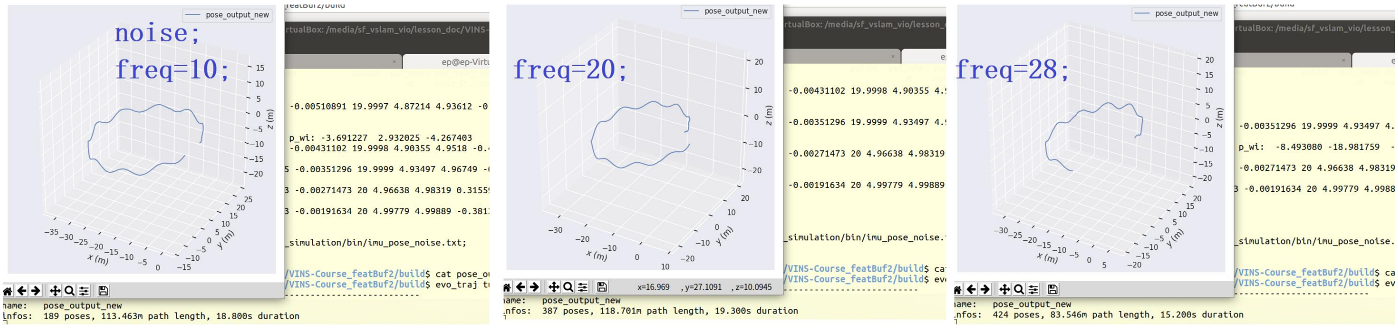
```yaml
                        # 2  Don't know anything about extrinsic parameters. You don't need to give R,T. We will
#If you choose 0 or 1, you should write down the following matrix.
#Rotation from camera frame to imu frame, imu^R_cam
extrinsicRotation: !!opencv-matrix
   rows: 3
   cols: 3
   dt: d
   #data: [0.0148655429818, -0.999880929698, 0.00414029679422,
   #        0.999557249008, 0.0149672133247, 0.025715529948,
   #        -0.0257744366974, 0.00375618835797, 0.999660727178]
   data: [0, 0, -1,
            -1, 0, 0,
            0, 1, 0   ]
#Translation from camera frame to imu frame, imu^T_cam
extrinsicTranslation: !!opencv-matrix
   rows: 3
   cols: 1
   dt: d
   data: [0.05,0.04,0.03] #[-0.0216401454975,-0.064676986768, 0.00981073058949]

#feature traker paprameters
max_cnt: 150            # max feature number in feature tracking
min_dist: 30            # min distance between two features
#freq: 10                # frequence (Hz) of publish tracking result. At least 10Hz for good estimation. If set 0
#freq: 20
freq: 28
F_threshold: 1.0        # ransac threshold (pixel)
show_track: 1           # publish tracking image as topic
equalize: 1             # if image is too dark or light, trun on equalize to find enough features
fisheye: 0              # if using fisheye, trun on it. A circle mask will be loaded to remove edge noisy points

#optimization parameters
max_solver_time: 0.04  # max solver itration time (ms), to guarantee real time
max_num_iterations: 8   # max solver itrations, to guarantee real time
keyframe_parallax: 10.0 # keyframe selection threshold (pixel)

#imu parameters       The more accurate parameters you provide, the better performance
acc_n: 0.019 #0.08          # accelerometer measurement noise standard deviation. #0.2   0.04
gyr_n: 0.015 #0.004         # gyroscope measurement noise standard deviation.     #0.05  0.004
acc_w: 0.0001 #0.00004         # accelerometer bias random work noise standard deviation.  #0.02
gyr_w: 1.0e-5 #2.0e-6       # gyroscope bias random work noise standard deviation.     #4.0e-5
g_norm: 9.81007     # gravity magnitude

```

2.轨迹对比

2.1 无噪声情况的不同 freq 轨迹对比



2.2 有噪声情况的不同 freq 的对比

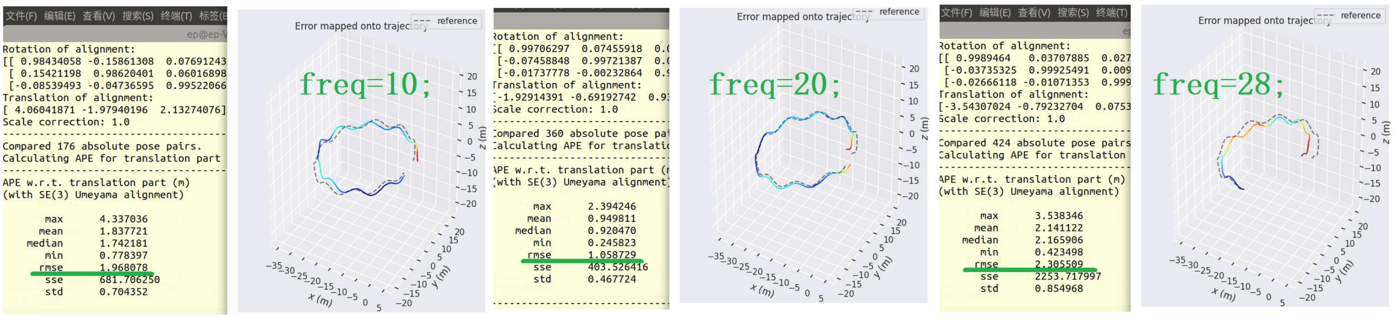

pose_output_new
noise;
freq=10;
freq=20;
freq=28;
x (m)
y (m)
z (m)
rtualBox: /media/sf_vslam_vio/lesson_doc/VINS-
ep@ep-Virtu
x=16.969 , y=27.1091 , z=10.0945

```
-0.00510891 19.9997 4.87214 4.93612 -0

p_wi: -3.691227  2.932025 -4.267403
-0.00431102 19.9998 4.90355 4.9518 -0.

5 -0.00351296 19.9999 4.93497 4.96749 -

3 -0.00271473 20 4.96638 4.98319 0.3155

3 -0.00191634 20 4.99779 4.99889 -0.381

_simulation/bin/imu_pose_noise.txt;

/VINS-Course_featBuf2/build$ cat pose_ou
/VINS-Course_featBuf2/build$ evo_traj t
--------------------------
name:    pose_output_new
infos:   189 poses, 113.463m path length, 18.800s duration
```

```
-0.00431102 19.9998 4.90355 4.

-0.00351296 19.9999 4.93497 4.

-0.00271473 20 4.96638 4.98319

-0.00191634 20 4.99779 4.99889

_simulation/bin/imu_pose_noise.

/VINS-Course_featBuf2/build$ ca
/VINS-Course_featBuf2/build$ ev
--------------------------
name:    pose_output_new
infos:   387 poses, 118.701m path length, 19.300s duration
```

```
-0.00351296 19.9999 4.93497 4.

p_wi:  -8.493080 -18.981759  -

-0.00271473 20 4.96638 4.98319

3 -0.00191634 20 4.99779 4.9988

_simulation/bin/imu_pose_noise.

/VINS-Course_featBuf2/build$ ca
/VINS-Course_featBuf2/build$ ev
--------------------------
name:    pose_output_new
infos:   424 poses, 83.546m path length, 15.200s duration
```

2.3 误差对比


文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 标签(B
ep@ep-V
Error mapped onto trajectory
reference
freq=10;
freq=20;
freq=28;
x (m)
y (m)
z (m)

```
Rotation of alignment:
[[ 0.98434058 -0.15861308  0.07691243
 [ 0.15421198  0.98620401  0.06016898
 [-0.08539493 -0.04736595  0.99522066
Translation of alignment:
[ 4.06041871 -1.97940196  2.13274076]
Scale correction: 1.0
-------------------------------------
Compared 176 absolute pose pairs.
Calculating APE for translation part
-------------------------------------
APE w.r.t. translation part (m)
(with SE(3) Umeyama alignment)

       max      4.337036
      mean      1.837721
    median      1.742181
       min      0.778397
      rmse      1.968078
       sse      681.706250
       std      0.704352
```

```
Rotation of alignment:
[[ 0.99706297  0.07455918  0.0
 [-0.07458848  0.99721387  0.0
 [-0.01737778 -0.00232864  0.9
Translation of alignment:
[-1.92914391 -0.69192742  0.93
Scale correction: 1.0
-----------------------------
Compared 360 absolute pose pai
Calculating APE for translatio
-----------------------------
APE w.r.t. translation part (m
(with SE(3) Umeyama alignment)

       max      2.394246
      mean      0.949811
    median      0.920470
       min      0.245823
      rmse      1.058729
       sse      403.526416
       std      0.467724

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

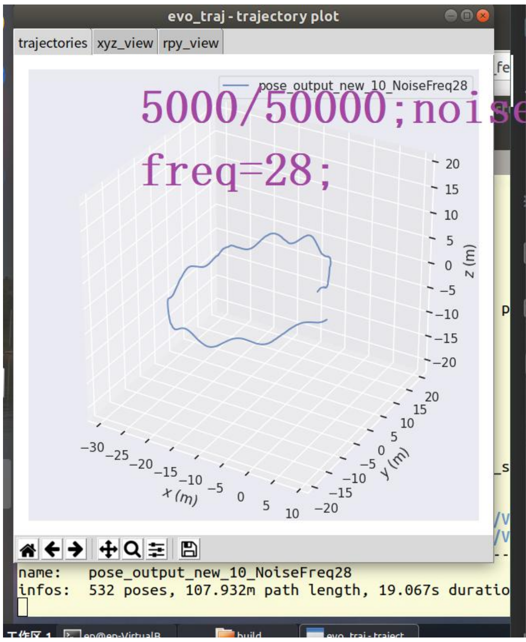
```
Rotation of alignment:
[[ 0.9989464   0.03707885  0.027
 [-0.03735325  0.99925491  0.009
 [-0.02666118 -0.01071353  0.999
Translation of alignment:
[-3.54307024 -0.79232704  0.0753
Scale correction: 1.0
-----------------------------
Compared 424 absolute pose pairs
Calculating APE for translation
-----------------------------
APE w.r.t. translation part (m)
(with SE(3) Umeyama alignment)

       max      3.538346
      mean      2.141122
    median      2.165906
       min      0.423498
      rmse      2.305509
       sse      2253.717997
       std      0.854968
```

可见，增加 freq，误差减小；但是 freq 比较大时，有时候出现时序不同步的情况，导致误差很大；

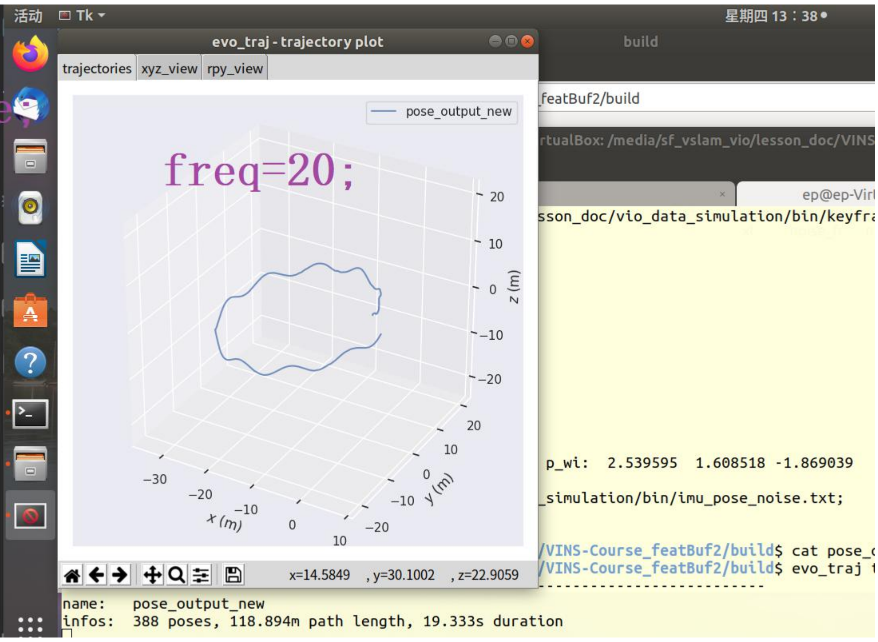
2.4 改变主程序的 usleep()的影响

以下是 2 种 usleep()的结果，当 PubImuData()的速度较慢时，有时候程序结果的轨迹会缺失，因为程序中 image 已经全部 publish 完毕，但是 imu 还没有 publish 完，而 vins 需要 image 和 imu 才能计算，所以轨迹出现缺失；

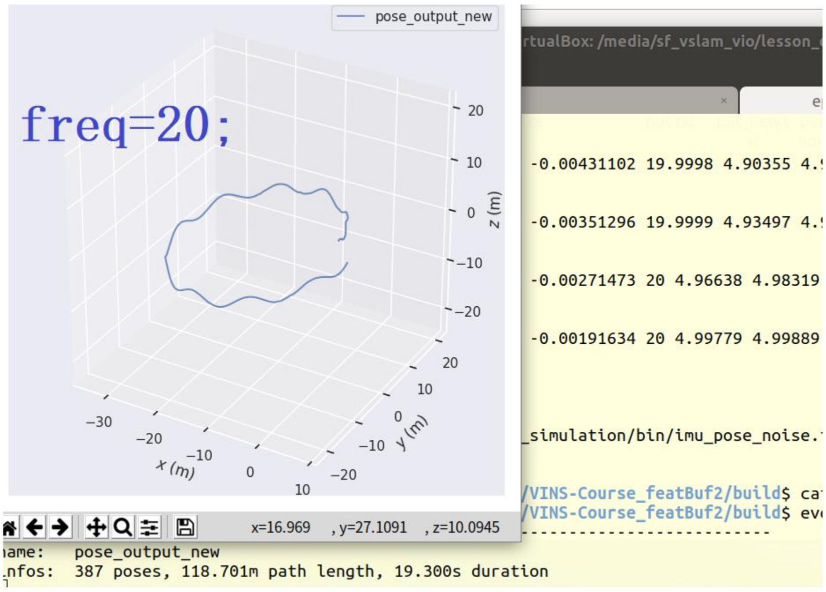
当 PubImuData()的速度较快时，结果轨迹正常。



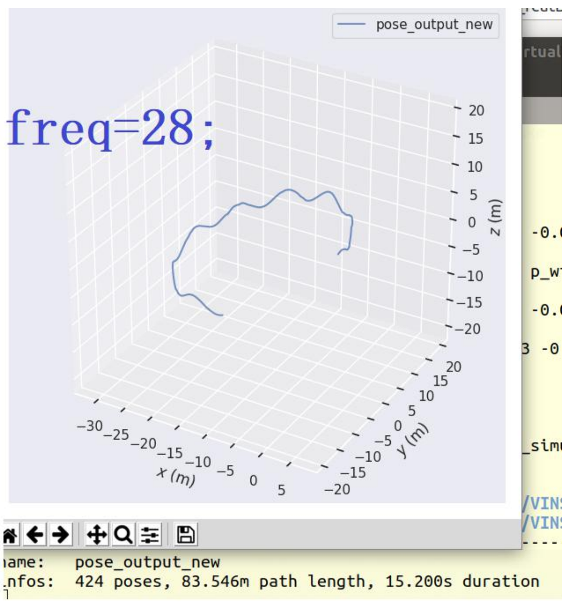
```
name:   pose_output_new_10_NoiseFreq28
infos:  532 poses, 107.932m path length, 19.067s duratio
```



```
featBuf2/build
rtualBox: /media/sf_vslam_vio/lesson_doc/VINS
ep@ep-Virt
sson_doc/vio_data_simulation/bin/keyfra

p_wi:  2.539595  1.608518 -1.869039
_simulation/bin/imu_pose_noise.txt;
/VINS-Course_featBuf2/build$ cat pose_c
/VINS-Course_featBuf2/build$ evo_traj t
------------------------
name:   pose_output_new
infos:  388 poses, 118.894m path length, 19.333s duration
```



```
rtualBox: /media/sf_vslam_vio/lesson_
-0.00431102 19.9998 4.90355 4.9
-0.00351296 19.9999 4.93497 4.9
-0.00271473 20 4.96638 4.98319
-0.00191634 20 4.99779 4.99889
_simulation/bin/imu_pose_noise.t
/VINS-Course_featBuf2/build$ ca
/VINS-Course_featBuf2/build$ ev
------------------------
ame:   pose_output_new
nfos:  387 poses, 118.701m path length, 19.300s duration
```



```
rtual
-0.0
p_w
-0.0
3 -0
_sim
/VIN
/VIN
----
ame:   pose_output_new
nfos:  424 poses, 83.546m path length, 15.200s duration
```

3.无 imu 数据的情况

vins 必须同时有 imu 和 image 才能计算轨迹；