

Table of Contents

1 代码改动.....	1
1.1 相机仿真数据处理.....	1
1.2 IMU 仿真数据处理.....	2
2 配置改动.....	3
3 仿真数据无噪声.....	4
3.1 参考噪声设置及其实验结果.....	4
3.2 错误噪声设置及其实验结果.....	5
4 仿真数据有噪声.....	6
4.1 参考噪声设置及其实验结果.....	6
4.2 错误噪声设置及其实验结果.....	6
4.3 噪声设置实验.....	7
5 总结.....	8

levinjian-第 7 章作业

1 代码改动

为了能够处理仿真数据，我们增加了一个新的可执行文件 run_sim，大部分代码在新增的 rum_sim.cpp 中。

代码的改动主要包括相机数据的读取，封装与发布，IMU 数据的读取与发布。

1.1 相机仿真数据处理

```
void PubImageData()
{
    string sImage_file = sData_path + "cam_pose_tum.txt";

    cout << "1 PubImageData start sImage_file: " << sImage_file << endl;
    ...
    ...

    // cv::namedWindow("SOURCE IMAGE", CV_WINDOW_AUTOSIZE);
    int img_id = 0;
    while (std::getline(fsImage, sImage_line) && !sImage_line.empty())
    {
        std::istringstream ssImuData(sImage_line);
        ssImuData >> dStampNSec;

        std::ostringstream stringstream;
        stringstream << sData_path<<"keyframe/all_points_"<<img_id++<<".txt";
        string imagePath = stringstream.str();
        cout << "Image t : " << fixed << dStampNSec << " Name: " << imagePath
        << endl;
```

```

void System::PubImageData_sim(double dStampSec, std::string img_path){
    ...

    std::string sImage_line;
    vector<int> ids;
    vector<cv::Point2f> cur_un_pts;
    while (std::getline(fsImage, sImage_line) && !sImage_line.empty()){
        double x,y,z,cont;
        double u,v;

        std::istringstream ssImuData(sImage_line);
        ssImuData >> x>>y>>z>>cont>>u>>v;
        std::ostringstream stringStream;
        stringStream << x<<" "<<y<<" "<<z;
        int fid = get_feature_id(stringStream.str());
        ids.push_back(fid);
        cur_un_pts.push_back(Point2f(u,v));
    }

    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 (int i = 0; i < NUM_OF_CAM; i++)
        {
            for (unsigned int j = 0; j < ids.size(); j++)
            {
                int p_id = ids[j];
                hash_ids[i].insert(p_id);
                double x = cur_un_pts[j].x;
                double y = cur_un_pts[j].y;
                double z = 1;
                feature_points->points.push_back(Vector3d(x, y, z));
                feature_points->id_of_point.push_back(p_id * NUM_OF_CAM + i);
                feature_points->u_of_point.push_back(-1);
                feature_points->v_of_point.push_back(-1);
                feature_points->velocity_x_of_point.push_back(-1);
                feature_points->velocity_y_of_point.push_back(-1);
            }
        }
    }
    ...
}

```

1.2 IMU 仿真数据处理

IMU 仿真数据处理在 run_sim.cpp 的 PubImageData 函数中。

```

void PubImuData()
{
    bool use_noisy_data = true;
    string sImu_data_file;
    if(use_noisy_data){
        sImu_data_file = sData_path + "imu_pose_noise.txt";
    }else{
        sImu_data_file = sData_path + "imu_pose.txt";
    }

    cout << "1 PubImuData start sImu_data_file: " << 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;
    Vec3 P;    // pose
    Qd Q;    // Rotation
    while (std::getline(fsImu, sImu_line) && !sImu_line.empty()) // read imu data
    {
        std::istringstream ssImuData(sImu_line);
        ssImuData >> dStampNSec;
        ssImuData>>Q.w()>>Q.x()>>Q.y()>>Q.z();
        ssImuData>>P.x()>>P.y()>>P.z();
        ssImuData >> vGyr.x() >> vGyr.y() >> vGyr.z() >> vAcc.x() >> vAcc.y()
>> vAcc.z();
        pSystem->PubImuData(dStampNSec , vGyr, vAcc);
        usleep(5000*nDelayTimes);
    }
    fsImu.close();
}

```

2 配置改动

我们增加了一个和仿真数据对应的配置文件，sim_config.yaml. 相对于现有的 euroc_config.yaml, 改动的地方包括外参设定，噪声设定。

```

extrinsicRotation: !!opencv-matrix
  rows: 3
  cols: 3
  dt: d
  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]
acc_n: 0.26870057685088805      # accelerometer measurement noise standard
deviation. #0.2  0.04
gyr_n: 0.21213203435596426    # gyroscope measurement noise standard
deviation. #0.05  0.004
acc_w: 7.0710678118654756e-06  # accelerometer bias random work noise
standard deviation. #0.02
gyr_w: 7.071067811865476e-07  # gyroscope bias random work noise standard
deviation. #4.0e-5
g_norm: 9.81007      # gravity magnitude

```

3 仿真数据无噪声

使用无噪声的数据，我们分别试了参考噪声设置与错误噪声设置，发现两者差不多，与真值轨迹的差别都很小。参考噪声设置是指数据仿真器使用的噪声值，在离散化后的结果。

具体来说，仿真器中用到的噪声是，

acc_n: 0.019, gyr_n:0.015,acc_w:0.0001,gyr_w:1.0e-5

离散化后的参考噪声设置是：

dt = 1/200

acc_n/sqrt(dt), gry_n/sqrt(dt), acc_w*sqrt(dt), gyr_w*sqrt(dt),

也有是

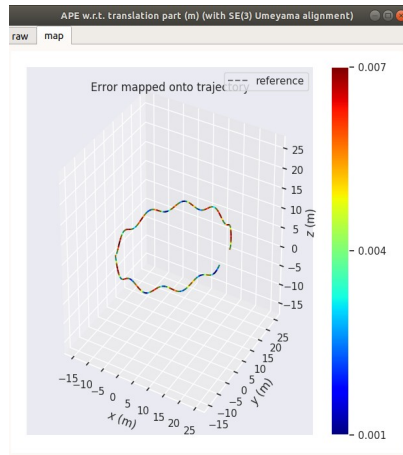
acc_n: 0.26870057685088805, gyr_n:0.21213203435596426,acc_w:7.0710678118654756e-06,gyr_w:7.071067811865476e-07

3.1 参考噪声设置及其实验结果

```

acc_n: 0.26870057685088805      # accelerometer measurement noise standard deviation.
#0.2  0.04
gyr_n: 0.21213203435596426    # gyroscope measurement noise standard deviation.
#0.05  0.004
acc_w: 7.0710678118654756e-06  # accelerometer bias random work noise standard
deviation. #0.02
gyr_w: 7.071067811865476e-07  # gyroscope bias random work noise standard deviation.
#4.0e-5

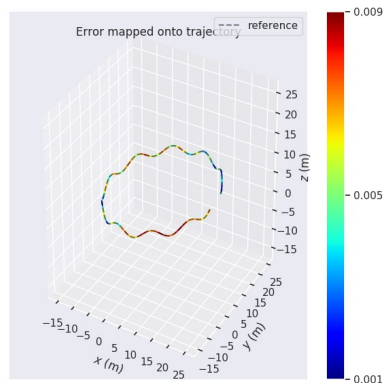
```



APE w.r.t. translation part (m) (with SE(3) Umeyama alignment)	
max	0.006818
mean	0.004558
median	0.004729
min	0.001244
rmse	0.004761
sse	0.004261
std	0.001374

3.2 错误噪声设置及其实验结果

```
#imu parameters
acc_n: 19          # accelerometer measurement noise standard deviation. #0.2  0.04
gyr_n: 15          # gyroscope measurement noise standard deviation.    #0.05 0.004
acc_w: 0.0001      # accelerometer bias random work noise standard deviation. #0.02
gyr_w: 1.0e-5      # gyroscope bias random work noise standard deviation.    #4.0e-5
```



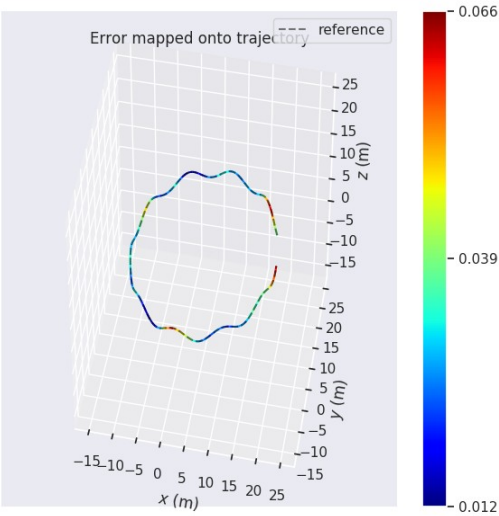
APE w.r.t. translation part (m) (with SE(3) Umeyama alignment)	
max	0.009270
mean	0.005805
median	0.006089
min	0.001282
rmse	0.006095
sse	0.006984
std	0.001859

4 仿真数据有噪声

4.1 参考噪声设置及其实验结果

在噪声设置正确的情况下，一段长达 119 米的 VIO 轨迹 RMSE 误差仅为 3cm.

```
acc_n: 0.26870057685088805      # accelerometer measurement noise
standard deviation. #0.2  0.04
gyr_n: 0.21213203435596426      # gyroscope measurement noise standard
deviation. #0.05  0.004
acc_w: 7.0710678118654756e-06    # accelerometer bias random work
noise standard deviation. #0.02
gyr_w: 7.071067811865476e-07    # gyroscope bias random work noise
standard deviation. #4.0e-5
```

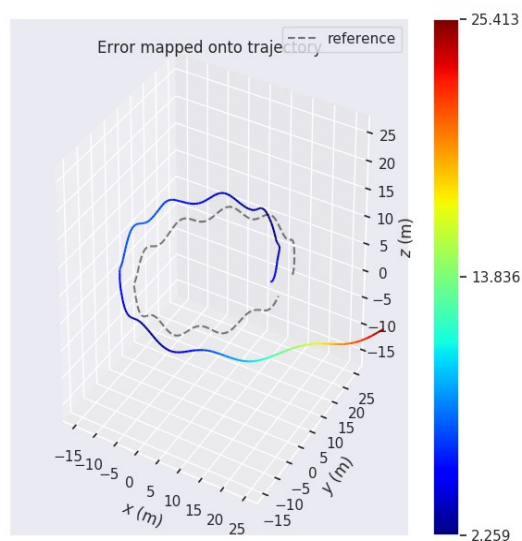


APE w.r.t. translation part (m) (with SE(3) Umeyama alignment)	
max	0.066250
mean	0.032357
median	0.029164
min	0.011569
rmse	0.034444
sse	0.223045
std	0.011808

4.2 错误噪声设置及其实验结果

在噪声设置错误的情况下，一段 119 米的 VIO 轨迹 RMSE 误差为 8m.

```
acc_n: 0.0019      # accelerometer measurement noise standard deviation. #0.2  0.04
gyr_n: 0.0015      # gyroscope measurement noise standard deviation. #0.05  0.004
acc_w: 0.0001      # accelerometer bias random work noise standard deviation. #0.02
gyr_w: 1.0e-05     # gyroscope bias random work noise standard deviation. #4.0e-5
```



APE w.r.t. translation part (m) (with SE(3) Umeyama alignment)	
max	25.412674
mean	6.457839
median	4.949057
min	2.258735
rmse	8.133958
sse	12438.318267
std	4.945461

4.3 噪声设置实验

为了进一步研究 IMU 噪声设置与 VIO 轨迹精度的关系，我们在参考噪声值的基础上，每次只改变其中的一个参数，做了一些交叉实验。主要的观察与分析如下：

- 1) 噪声设置对 VIO 精度有很大的影响。

在本次实验中，当噪声设置合适时，VIO 精度是 cm 级别的，但不合适时，精度是米级别的，VIO 基本不可用。

- 2) 角速度白噪声的设置对精度影响最大，其次是加速度白噪声。角速度偏置随机游走噪声与加速度偏置随机游走噪声影响不大。
- 3) 相对来说，取比参考噪声稍大一点的值时得到的精度最好。

原因可能是在生成仿真数据时，我们的图像数据是没有噪声的。适当加大 IMU 噪声可以较少 IMU 参差在成本函数中的权重，更加依赖图形数据，从而取得了更好的效果。

具体的实验结果如下，其中，所有表格的第一行是参考噪声，绿色的行表示精度最好的组合，

与噪声参考值的倍数关系	测试参数	max	mean	median	rmse	std
1	acc_n, 0.26870057685088805	0.06625	0.032357	0.029164	0.034444	0.011808
0.1	acc_n, 0.026870057685088805	0.14559	0.062551	0.059148	0.06854	0.028018
0.01	acc_n, 0.0026870057685088805	0.170923	0.068579	0.062001	0.076149	0.0331
10	acc_n, 2.6870057685088805	0.065533	0.021683	0.0198	0.024614	0.011648
100	acc_n, 26.870057685088805	0.463865	0.25247	0.217428	0.265492	0.082126
与噪声参考值的倍数关系	测试参数	max	mean	median	rmse	std
1	gyr_n, 0.21213203435596426	0.06625	0.032357	0.029164	0.034444	0.011808
0.1	gyr_n, 0.021213203435596426	1.289007	0.588574	0.599336	0.67815	0.336851
0.01	gyr_n, 0.0021213203435596426	13.100765	3.291967	2.956284	4.114657	2.468472
10	gyr_n, 2.1213203435596426	0.0663	0.027632	0.022596	0.031582	0.015293
100	gyr_n, 21.213203435596426	0.10221	0.040772	0.035613	0.046762	0.022898
与噪声参考值的倍数关系	测试参数	max	mean	median	rmse	std
1	acc_w, 7.0710678118654756e-06	0.06625	0.032357	0.029164	0.034444	0.011808
0.1	acc_w, 7.0710678118654756e-07	0.066156	0.032239	0.029179	0.034313	0.011748
0.01	acc_w, 7.0710678118654756e-08	0.072833	0.03447	0.032386	0.036477	0.011932
10	acc_w, 7.0710678118654756e-05	0.066506	0.032407	0.029268	0.034505	0.011848
100	acc_w, 7.0710678118654756e-04	0.066416	0.032409	0.029295	0.034499	0.011827
1000	acc_w, 7.0710678118654756e-03	0.06601	0.032296	0.029147	0.034377	0.011778
10000	acc_w, 7.0710678118654756e-02	0.067389	0.027777	0.025047	0.029966	0.011242
100000	acc_w, 7.0710678118654756e-01	0.069192	0.026364	0.023951	0.030387	0.015111
1000000	acc_w, 7.0710678118654756e-0	0.15089	0.052706	0.045264	0.062302	0.03322
与噪声参考值的倍数关系	测试参数	max	mean	median	rmse	std
1	gyr_w, 7.071067811865476e-07	0.06625	0.032357	0.029164	0.034444	0.011808
0.1	gyr_w, 7.071067811865476e-08	0.065128	0.031851	0.028969	0.033798	0.011308
10	gyr_w, 7.071067811865476e-06	0.066351	0.032397	0.029319	0.034488	0.011828
100	gyr_w, 7.071067811865476e-05	0.066352	0.032397	0.029321	0.034487	0.011826
1000	gyr_w, 7.071067811865476e-04	0.066381	0.032394	0.029299	0.034486	0.011829
10000	gyr_w, 7.071067811865476e-03	0.066308	0.032273	0.029131	0.034362	0.011796
100000	gyr_w, 7.071067811865476e-02	0.06468	0.031153	0.027863	0.033047	0.011026
1000000	gyr_w, 7.071067811865476e-01	0.0649	0.029759	0.027705	0.031438	0.010138
10000000	gyr_w, 7.071067811865476e0	0.057116	0.024294	0.022553	0.026071	0.00946

5 总结

利用仿真的相机特征数据和 IMU 数据，来调试和分析 VIO 运行是一件非常有意义的事情。从这个项目的实验结果来看，IMU 噪声设置对 VIO 有非常大的影响。

在实际 VIO 项目中，IMU 噪声调优是一项十分重要的工作，它很大程度上决定了系统的定位精度。受到这个项目实验结果的启发，打算在实际工作中曾试以下步骤做 IMU 噪声参数调优。

- 1) 利用 Allan 方差标定方法得到 IMU 噪声的参考值
- 2) 在 IMU 噪声的参考值附近，设计一系列参数组合，其中，角速度白噪声可选值最多，加速度白噪声次之，偏置随机游走噪声可选值可比较少。
- 3) 测试所有的参数组合，选取最佳组合