代码主要做以下改动：

run\_euroc.cpp中：

1. **void** PubImuData()
2. {
3. //  string sImu\_data\_file = sConfig\_path + "MH\_05\_imu0.txt";
4. string sImu\_data\_file = "/media/yxt/storage/github\_useful\_tools/vio\_data\_simulation/bin/imu\_pose.txt";
5. cout << "1 PubImuData start sImu\_data\_file: " << sImu\_data\_file << endl;
6. ifstream fsImu;
7. fsImu.open(sImu\_data\_file.c\_str());
8. **if** (!fsImu.is\_open())
9. {
10. cerr << "Failed to open imu file! " << sImu\_data\_file << endl;
11. **return**;
12. }
14. std::string sImu\_line;
15. **double** dStampNSec = 0.0;
16. Vector3d vAcc;
17. Vector3d vGyr;
18. **double** tmp;
19. **while** (std::getline(fsImu, sImu\_line) && !sImu\_line.empty()) // read imu data
20. {
21. std::istringstream ssImuData(sImu\_line);
22. ssImuData >> dStampNSec;//时间戳
23. **for**(**int** i=0;i<7;i++) //利用循环跳过imu quaternion(4)和imu position(3)
24. ssImuData>>tmp;
25. ssImuData >> vGyr.x() >> vGyr.y() >> vGyr.z() >> vAcc.x() >> vAcc.y() >> vAcc.z();
26. // cout << "Imu t: " << fixed << dStampNSec << " gyr: " << vGyr.transpose() << " acc: " << vAcc.transpose() << endl;
27. pSystem->PubImuData(dStampNSec, vGyr, vAcc);
28. usleep(5000\*nDelayTimes);//10000um = 0.01s
29. }
30. fsImu.close();
31. }
32. **void** PubSimImageData(){
33. string sImage\_file = "/media/yxt/storage/github\_useful\_tools/vio\_data\_simulation/bin/cam\_pose.txt";
35. cout << "1 PubImageData start sImage\_file: " << sImage\_file << endl;
37. ifstream fsImage;
38. fsImage.open(sImage\_file.c\_str());
39. **if** (!fsImage.is\_open())
40. {
41. cerr << "Failed to open image file! " << sImage\_file << endl;
42. **return**;
43. }
45. std::string sImage\_line;
46. **double** dStampNSec;
47. //    string sImgFileName;
48. **int** n = 0;
50. **while** (std::getline(fsImage, sImage\_line) && !sImage\_line.empty()){
51. std::istringstream ssImuData(sImage\_line);
52. ssImuData >> dStampNSec;
53. cout<<"cam time: "<<fixed<<dStampNSec<<endl;
54. //cam\_pose.txt中相机数与keyframe中每一帧一一对应，从all\_points\_0.txt到all\_points\_600
55. string imagePath = "/media/yxt/storage/github\_useful\_tools/vio\_data\_simulation/bin/keyframe/all\_points\_"
56. + std::to\_string(n) + ".txt";
57. cout<<"points\_file: "<<imagePath<<endl;
58. vector<cv::Point2f> FeaturePoints;//容器FeaturePoints存放一个相机的特征点(归一化坐标)
59. ifstream f;
60. f.open(imagePath.c\_str());
62. **if** (!f.is\_open())
63. {
64. cerr << "Failed to open image file! " << imagePath << endl;
65. **return**;
66. }
68. std::string s;
69. **while** (std::getline(f, s) && !s.empty()){
70. std::istringstream ss(s);
71. **double** tmp;
72. **for** (**int** i = 0; i < 4; i++)
73. ss >> tmp;
74. **float** px, py;
75. ss >。。 px;
76. ss >> py;
77. cv::Point2f pt(px, py);
78. FeaturePoints.push\_back(pt);
80. }
81. pSystem->PubSimImageData(FeaturePoints, dStampNSec);
82. usleep(100000\*nDelayTimes);//usleep延时时间单位为微秒，百万分之一,200000us = 0.2s,5Hz
83. n++;
84. }
85. fsImage.close();
87. }

System.cpp中

1. **void** System::PubSimImageData(**const** vector<cv::Point2f> &feature, **const** **double** &dStampSec) {
3. last\_image\_time = dStampSec;
5. PUB\_THIS\_FRAME = **true**;
6. TicToc t\_r;
8. trackerData[0].loaddata(feature, dStampSec);
10. **for** (unsigned **int** i = 0;; i++)
11. {
12. **bool** completed = **false**;
13. completed |= trackerData[0].updateID(i); //a|=b;就是a=a|b
15. **if** (!completed)
16. **break**;
17. }
19. **if** (PUB\_THIS\_FRAME) {
20. pub\_count++;
21. shared\_ptr<IMG\_MSG> feature\_points(**new** IMG\_MSG());
22. feature\_points->header = dStampSec;
24. **for** (**int** i = 0; i < NUM\_OF\_CAM; i++) {
25. auto &pts\_velocity = trackerData[i].pts\_velocity;
26. **for** (**int** j = 0; j < feature.size(); j++) {
27. **int** p\_id = j;
28. feature\_points->points.push\_back(Vector3d(feature[j].x, feature[j].y, 1));
29. feature\_points->id\_of\_point.push\_back(p\_id \* NUM\_OF\_CAM + i);
30. cv::Point2f pixel\_point;
31. pixel\_point.x = 460 \* feature[j].x + 255;
32. pixel\_point.y = 460 \* feature[j].y + 255;
33. feature\_points->u\_of\_point.push\_back(pixel\_point.x);
34. feature\_points->v\_of\_point.push\_back(pixel\_point.y);
35. feature\_points->velocity\_x\_of\_point.push\_back(pts\_velocity[j].x);
36. feature\_points->velocity\_y\_of\_point.push\_back(pts\_velocity[j].y);
37. //                feature\_points->velocity\_x\_of\_point.push\_back(0);
38. //                feature\_points->velocity\_y\_of\_point.push\_back(0);
39. }
40. **if** (!init\_pub)
41. {
42. cout << "4 PubImage init\_pub skip the first image!" << endl;
43. init\_pub = 1;
44. }
45. **else** {
46. m\_buf.lock();
47. feature\_buf.push(feature\_points);
48. m\_buf.unlock();
49. con.notify\_one();
50. }
51. }
52. }
53. // cout << "5 PubImage t : " << fixed << feature\_points->header
54. //     << " feature\_buf size: " << feature\_buf.size() << endl;
55. }

IMU噪声与轨迹精度如下表所示

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | acc\_n | gyr\_n | acc\_w | gyr\_w | max | mean | min | rmse |
| 0 | 0 | 0 | 0 | 0 | 0.20 | 0.08 | 0.013 | 0.094 |
| 1 | 0.0019 | 0.0015 | 1e-5 | 1e-6 | 0.22 | 0.11 | 0.026 | 0.11 |
| 2 | 0.0038 | 0.0030 | 2e-5 | 2e-6 | 0.57 | 0.25 | 0.069 | 0.28 |
| 3 | 0.0076 | 0.0060 | 4e-5 | 4e-6 | 1.3 | 0.81 | 0.9 | 0.86 |
| 4 | 0.019 | 0.015 | 0.0001 | 1e-5 | 4.95 | 2.20 | 0.66 | 2.41 |

下面四张图分别对应上表1、2、3、4四种情况

|  |  |
| --- | --- |
| 1 | 2 |
| 3 | 4 |