1、程序修改

新建 simulation_test.cpp 文件，修改 PubImuData 如下

```cpp
void PubImuData()
{
    // string sImu_data_file = sConfig_path + "MH_05_imu0.txt";
    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;
    double tmp;
    Vector3d vAcc;
    Vector3d vGyr;
    while (std::getline(fsImu, sImu_line) && !sImu_line.empty()) // read imu data
    {
        std::istringstream ssImuData(sImu_line);
        ssImuData >> dStampNSec;
        for(int i = 0; i < 7; i++)
            ssImuData >> tmp;
        ssImuData >> vGyr.x() >> vGyr.y() >> vGyr.z() >> vAcc.x() >> vAcc.y() >> vAcc.z();
        // cout << "Imu t: " << fixed << dStampNSec << " gyr: " << vGyr.transpose() << " acc
        pSystem->PubImuData(dStampNSec, vGyr, vAcc);
        usleep(5000*nDelayTimes);
    }
    fsImu.close();
}
```

修改 PubImageData 如下

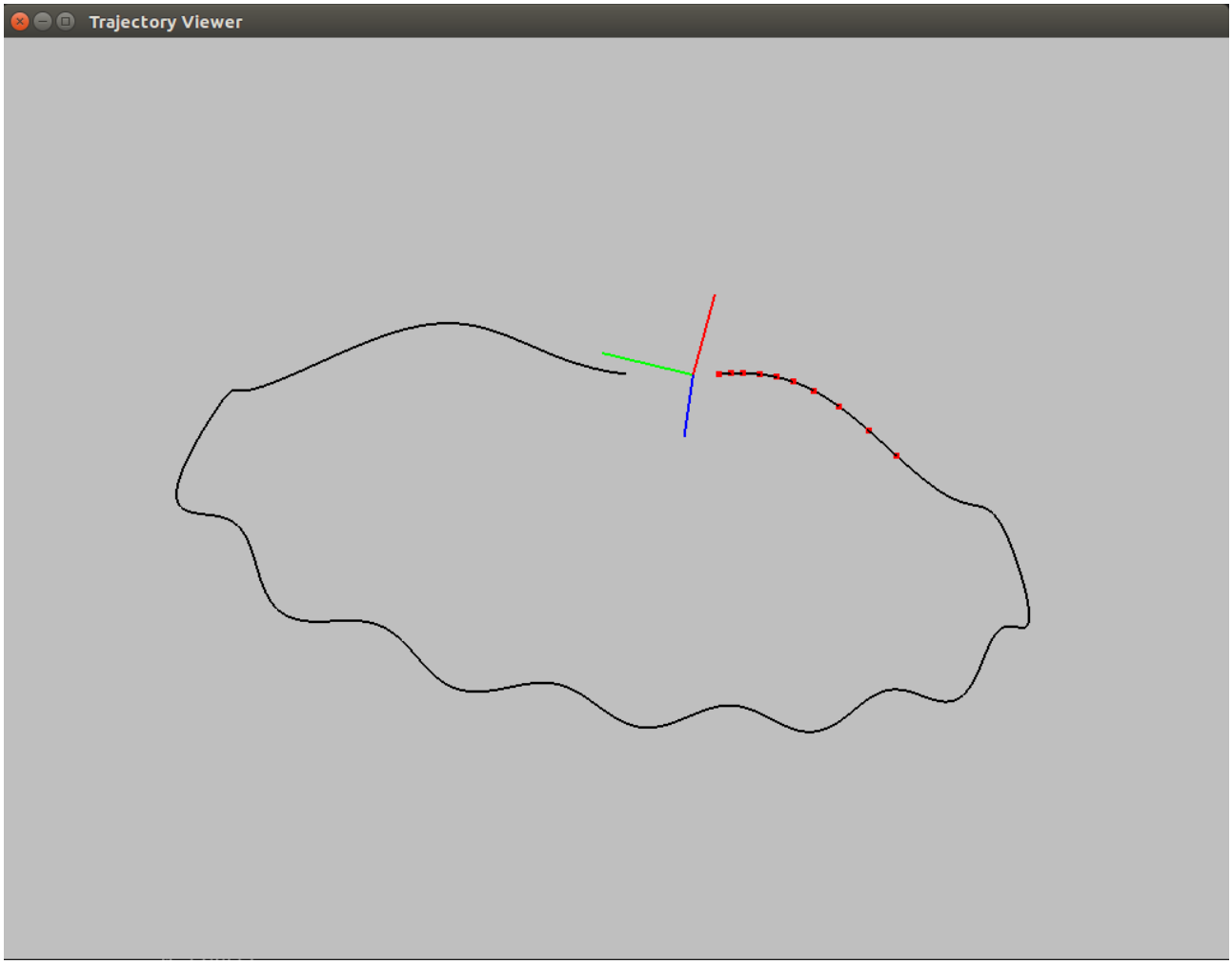
```cpp
void PubImageData()
{
    string sImage_file = sData_path + "cam_pose.txt"; // 含时间戳的文件
    ifstream fsImage;
    fsImage.open(sImage_file.c_str());
    if (!fsImage.is_open())
    {
        cerr << "Failed to open image file! " << sImage_file << endl;
        return;
    }
    std::string sImage_line;
    double dStampNSec;
    string sImgFileName;
    int n = 0;
    while (std::getline(fsImage, sImage_line) && !sImage_line.empty())
    {
        std::istringstream ssImgData(sImage_line);
        ssImgData >> dStampNSec;
        string all_points_file_name = "/home/touchair/vio_data_simulation/bin/keyframe/all_points_" + to_string(n)+ ".txt";
        vector<cv::Point2f> FeaturePoints;
        std::ifstream f;
        f.open(all_points_file_name);
        while(!f.eof())
        {
            std::string s;
            std::getline(f, s);
            if(!s.empty())
            {
                std::stringstream ss;
                ss << s;
                double tmp;
                for(int i = 0; i < 4; i++)
                    ss >> tmp;
                float px, py;
                ss >> px;
                ss >> py;
                cv::Point2f pt(px, py);
                FeaturePoints.push_back(pt);
            }
        }

        pSystem->PubSimImageData(dStampNSec, FeaturePoints);
        usleep(50000*nDelayTimes);
        n++;
    }
    fsImage.close();
}
```

修改 System.cpp 中 PubImageData 如下

```
for (int i = 0; i < NUM_OF_CAM; i++)
{
      auto &un_pts = trackerData[i].cur_un_pts;// 去畸变的归一化图像坐标
      auto &cur_pts = trackerData[i].cur_pts;// 当前追踪到的特征点
      auto &ids = trackerData[i].ids;
      auto &pts_velocity = trackerData[i].pts_velocity;
    //遍历相机的所有特征点
    for (unsigned int j = 0; j < FeaturePoints.size(); j++)
    {
          if (trackerData[i].track_cnt[j] > 1)
          {
              int p_id = ids[j];
        int p_id = j;
        hash_ids[i].insert(p_id);
        double x = FeaturePoints[j].x;
        double y = FeaturePoints[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(cur_pts[j].x); // 像素坐标
              feature_points->v_of_point.push_back(cur_pts[j].y);

              feature_points->velocity_x_of_point.push_back(pts_velocity[j].x);
              feature_points->velocity_y_of_point.push_back(pts_velocity[j].y);

        cv::Point2f pixel_point;//特征点对应的像素坐标
        pixel_point.x = 460 * x + 255;
        pixel_point.y = 460 * y + 255;

        feature_points->u_of_point.push_back(pixel_point.x); // 像素坐标
        feature_points->v_of_point.push_back(pixel_point.y);
        // 这里默认速度为0不考虑
        feature_points->velocity_x_of_point.push_back(0);
        feature_points->velocity_y_of_point.push_back(0);
          }
    }
```

2、对不同大小噪声的 IMU 数据和相机数据仿真

无噪声

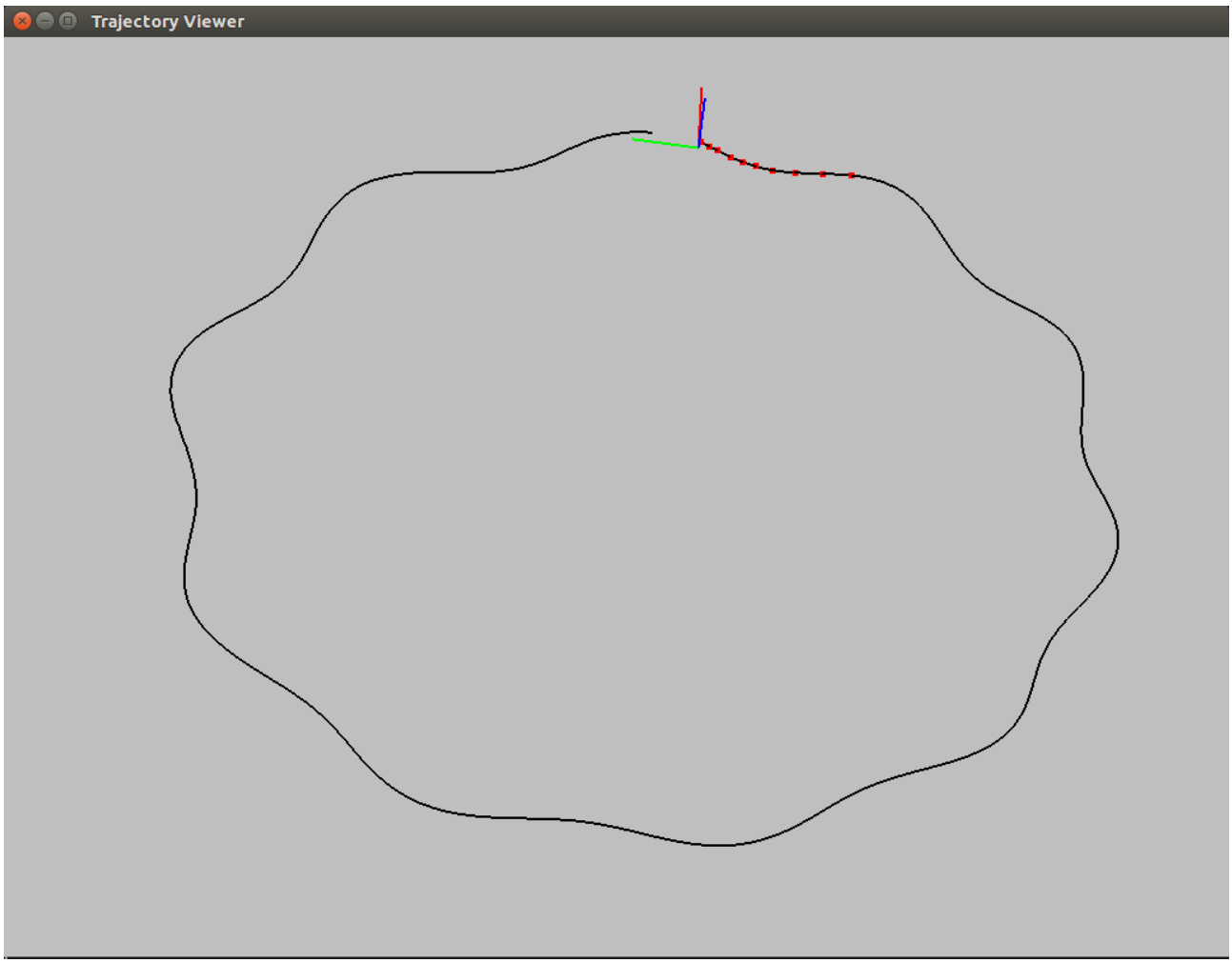


低噪声

```
double gyro_bias_sigma = 1.0e-7;
double acc_bias_sigma = 0.000001;

double gyro_noise_sigma = 0.00015;
double acc_noise_sigma = 0.00019;
```

加大噪声

```
double gyro_bias_sigma = 1.0e-6;
double acc_bias_sigma = 0.00001;

double gyro_noise_sigma = 0.0015;
double acc_noise_sigma = 0.0019;
```



继续加大噪声

```
double gyro_bias_sigma = 1.0e-5;
double acc_bias_sigma = 0.0001;

double gyro_noise_sigma = 0.015;
double acc_noise_sigma = 0.019;
```

可以看出，噪声越大，效果越差