深蓝学院
shenlanxueyuan.com

第7讲作业分享

主讲人 Hoinam

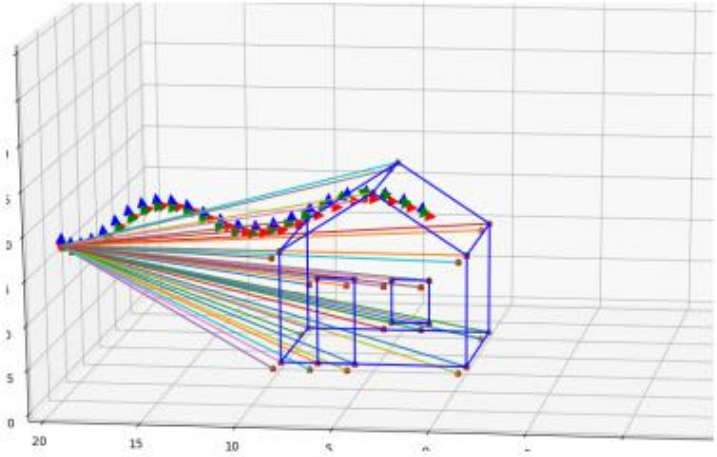
纲要

- 第一部分：实现思路
- 第二部分：代码修改
- 第三部分：实验结果

作业

作业

1. 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（不同噪声设定时，需要配置 vins 中 imu noise 大小。）



实现思路

本章代码主要有三个线程：

IMU线程　　从Euroc数据集的IMU文件读取IMU数据

Tracking线程　　从Euroc数据集的Image文件读取Image数据，并做光流跟踪等特征处理操作

Estimator线程　　后端处理，根据前端提供的IMU数据和特征数据，做预积分和相关后端优化

实现思路

第二章生成的结果有：

IMU数据： imu_pose.txt无噪声 imu_pose_noise.txt有噪声

特征点在归一化平面上的坐标： cam_pose.txt相机时戳

all_points_xx.txt 各帧特征点归一化坐标数据

所以，把实现思路为把以上两部分接入本章代码，IMU数据按格式输入，图像数据直接使用特征点文件代替，省去原先特征跟踪的过程

纲要

- 第一部分：实现思路
- **第二部分：代码修改**
- 第三部分：实验结果

IMU读取

test/run_euroc.cpp　　void PubImuData()

从euroc数据集的imu文件中读取数据，改为从第二章的imu文件中读取。

需要修改读取格式

IMU读取

深蓝学院 shenlanxueyuan.com

```
void PubImuData()
{
    string sImu_data_file = sConfig_path + "MH_05_imu0.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;
    while (std::getline(fsImu, sImu_line) && !sImu_line.empty()) // read imu data
    {
        std::istringstream ssImuData(sImu_line);
        ssImuData >> dStampNSec >> vGyr.x() >> vGyr.y() >> vGyr.z() >> vAcc.x() >> vAc
        // cout << "Imu t: " << fixed << dStampNSec << " gyr: " << vGyr.transpose() <<
        pSystem->PubImuData(dStampNSec / 1e9, vGyr, vAcc);
        usleep(5000*nDelayTimes);
    }
```

```
void PubImuData()
{
    string sImu_data_file = sConfig_path + "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;
        return;
    }
    std::string sImu_line;
    double dStampNSec = 0.0;
    Vector3d vAcc;
    Vector3d vGyr;
    while (std::getline(fsImu, sImu_line) && !sImu_line.empty()) // read imu data
    {
        std::istringstream ssImuData(sImu_line);
        ssImuData >> dStampNSec;
        double tmp;
        for(int i = 0; i < 7 ; i++)
        {
            ssImuData >> tmp;
        }
        ssImuData >> vGyr.x() >> vGyr.y() >> vGyr.z() >> vAcc.x() >> vAcc.y() >> vAcc
        // cout << "Imu t: " << fixed << dStampNSec << " gyr: " << vGyr.transpose() <
        pSystem->PubImuData(dStampNSec, vGyr, vAcc);
        usleep(5000*nDelayTimes);
    }
```

图像读取

test/run_euroc.cpp　　void PubImageData()

从euroc数据集读取图像文件，然后通过 pSystem->PubImageData()，提取和跟踪特征点。

这里改成直接读取特征点坐标文件，获得当前图像帧的所有特征点的数组，送给pSystem->PubImageData(double…, Mat…)。因此需要重载参数，原先是Mat型图像，改为vector<Point2f>型点对数组

图像读取

深蓝学院 shenlanxueyuan.com

```
void PubImageData()
{
    string sImage_file = sConfig_path + "MH_05_cam0.txt";
    cout << "1 PubImageData start sImage_file: " << sImage_file << endl;

    ⋮

        Mat img = imread(imagePath.c_str(), 0);
        if (img.empty())

        {
            cerr << "image is empty! path: " << imagePath << endl;

            return;

        }
        pSystem->PubImageData(dStampNSec / 1e9, img);
```

```
void PubImageData()
{
    string sImage_file = sConfig_path + "cam_pose.txt";
    cout << "1 PubImageData start sImage_file: " << sImage_file << endl;

    ⋮

        //Mat img = imread(imagePath.c_str(), 0);
        //if (img.empty())
        //  cerr << "image is empty! path: " << imagePath << endl;
        //  return;
        //}
        vector<cv::Point2f> FeaturePoints;
        std::ifstream f;
        f.open(imagePath);
        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);
            cout << "cx cy "<< px << py << endl;
            FeaturePoints.push_back(pt);
            }
        }
        //f.close();
        //pSystem->PubImageData(dStampNSec / 1e9, img);
        pSystem->PubImageData(dStampNSec, FeaturePoints);
```

图像处理

src/System.cpp　　void PubImageData(double…)

处理图像文件，主要通过trackData[0].readImage(…)，该接口内部包含 光流跟踪，特征点筛选，去畸变等操作，最终输出特征点

这里上层直接获取了特征点数组，所以可以根据空间位置信息判断观测点是否被上一帧观测。若是，将观测点id设置成匹配点的id,并计算光流。若否，将该观测点设置为新的id，光流速度设为0。（但我实现的时候，未做可见判断操作，直接把特征点数据解析了出来，光流速度始终设为0。）

图像处理

深蓝学院 shenlanxueyuan.com

```
void System::PubImageData(double dStampSec, Mat &img)
{
    if (!init_feature)
    {
        cout << "1 PubImageData skip the first detected feature, which doesn't contain
    ⋮
    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++)
        {
            auto &un_pts = trackerData[i].cur_un_pts;
            auto &cur_pts = trackerData[i].cur_pts;
            auto &ids = trackerData[i].ids;
            auto &pts_velocity = trackerData[i].pts_velocity;
            for (unsigned int j = 0; j < ids.size(); j++)
            {
                if (trackerData[i].track_cnt[j] > 1)
                {
                    int p_id = ids[j];
                    hash_ids[i].insert(p_id);
                    double x = un_pts[j].x;
                    double y = 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(cur_pts[j].x);
                    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);
                }
            }
```

```
void System::PubImageData(double dStampSec, const vector<cv::Point2f> &FeaturePoints)
{
    if (!init_feature)
    {
        cout << "1 PubImageData skip the first detected feature, which doesn't contain
    ⋮
    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++)
        {
            auto &un_pts = trackerData[i].cur_un_pts;
            auto &cur_pts = trackerData[i].cur_pts;
            auto &ids = trackerData[i].ids;
            auto &pts_velocity = trackerData[i].pts_velocity;
            for (unsigned int j = 0; j < FeaturePoints.size(); j++)
            {
                //if (trackerData[i].track_cnt[j] > 1)
                {
                    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(460 * x + 255);
                    feature_points->v_of_point.push_back(460 * y + 255);
                    feature_points->velocity_x_of_point.push_back(0);
                    feature_points->velocity_y_of_point.push_back(0);
                }
            }
```

配置文件

config/euroc_config.yaml

配置文件中设置了相机和IMU的内外参，包括二者的相对位姿以及IMU的噪声参数

这里需要改为第二章生成数据时设置的参数

配置文件

深蓝学院
shenlanxueyuan.com

```
                        # 1  Have an initial guess about extrinsic parameters. We will
                        # 2  Don't know anything about extrinsic parameters. You don't
#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]
#Translation from camera frame to imu frame, imu^T_cam
extrinsicTranslation: !!opencv-matrix
   rows: 3
   cols: 1
   dt: d
   data: [-0.0216401454975,-0.064676986768, 0.00981073058949]

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: 0.08          # accelerometer measurement noise standard deviation. #0.2   0.04
gyr_n: 0.004         # gyroscope measurement noise standard deviation.     #0.05  0.00
acc_w: 0.00004         # accelerometer bias random work noise standard deviation.  #0.
gyr_w: 2.0e-6       # gyroscope bias random work noise standard deviation.     #4.0e-5
g_norm: 9.81007     # gravity magnitude
```

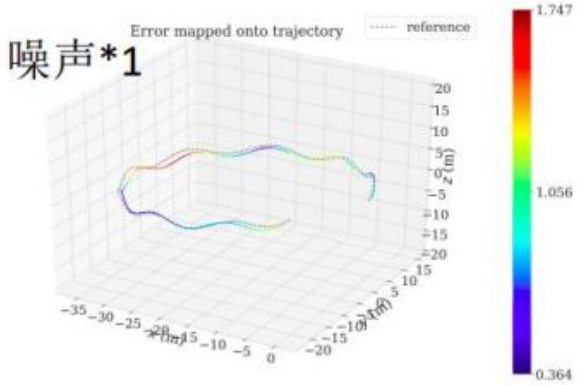
```
                        # 1  Have an initial guess about extrinsic parameters. We will
                        # 2  Don't know anything about extrinsic parameters. You don't
#If you choose 0 or 1, you should write down the following matrix.
#Rotation from camera frame to imu frame, imu^R_cam
extrinsicRotation: !!opencv-matrix
   rows: 3
   cols: 3
   dt: d
   data: [0, 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]

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          # accelerometer measurement noise standard deviation.  #0.2   0.0
gyr_n: 0.015         # gyroscope measurement noise standard deviation.     #0.05  0.00
acc_w: 0.0005         # accelerometer bias random work noise standard deviation.  #0.0
gyr_w: 5.0e-5       # gyroscope bias random work noise standard deviation.     #4.0e-5
g_norm: 9.81007     # gravity magnitude
```

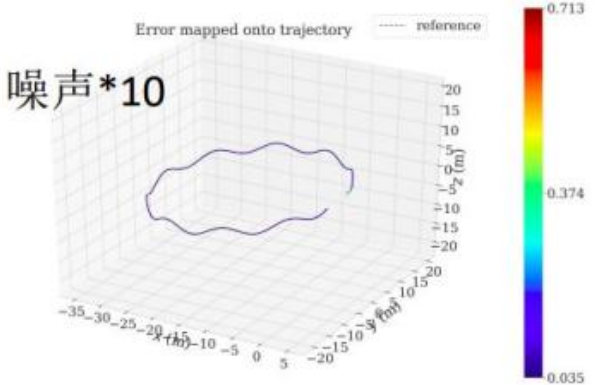
纲要

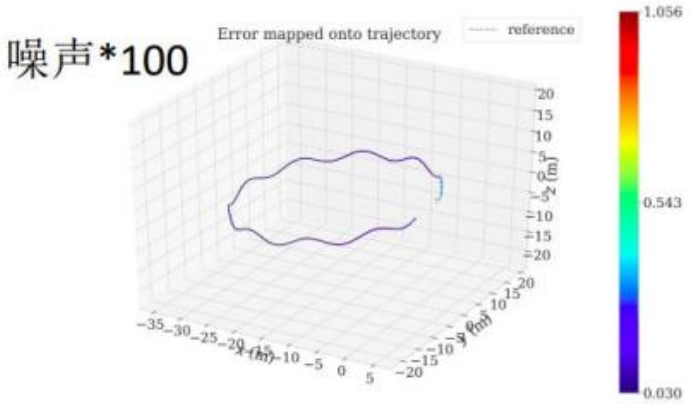
深蓝学院
shenlanxueyuan.com

- 第一部分：实现思路
- 第二部分：代码修改
- **第三部分：实验结果**

噪声影响







	max	mean	min	rmse
噪声*1	1.746751	0.944979	0.364448	1.004950
噪声*10	0.712855	0.072721	0.035496	0.095686
噪声*100	1.055822	0.124855	0.029540	0.172392

改进

图像加噪声：

由于目前的噪声都是加在IMU里的，可以给图像处理部分也模拟出噪声。比如在特征点的位置加上高斯噪声等。

系统鲁棒性：

特征点加入随机漏检，虚警，误匹配等策略，模拟实际环境下特征提取会出现的问题，验证系统鲁棒性。

在线问答

深蓝学院
shenlanxueyuan.com

Q&A

深蓝学院
shenlanxueyuan.com

感谢各位聆听！

Thanks for Listening