双目视觉 SLAM 方案技术开发阶段性验收报告

1 系统运行

通过在/data/ttt/目录下输入如下命令可运行整个系统：

LD_LIBRARY_PATH=/data/ttt/lib/dep_libs/:/pcl_libs/ ./sweeper_robot ./initial/sweeper_config.yaml。下面分模块进行介绍。

2 定位与建图

2.1 双目视觉 SLAM 方法

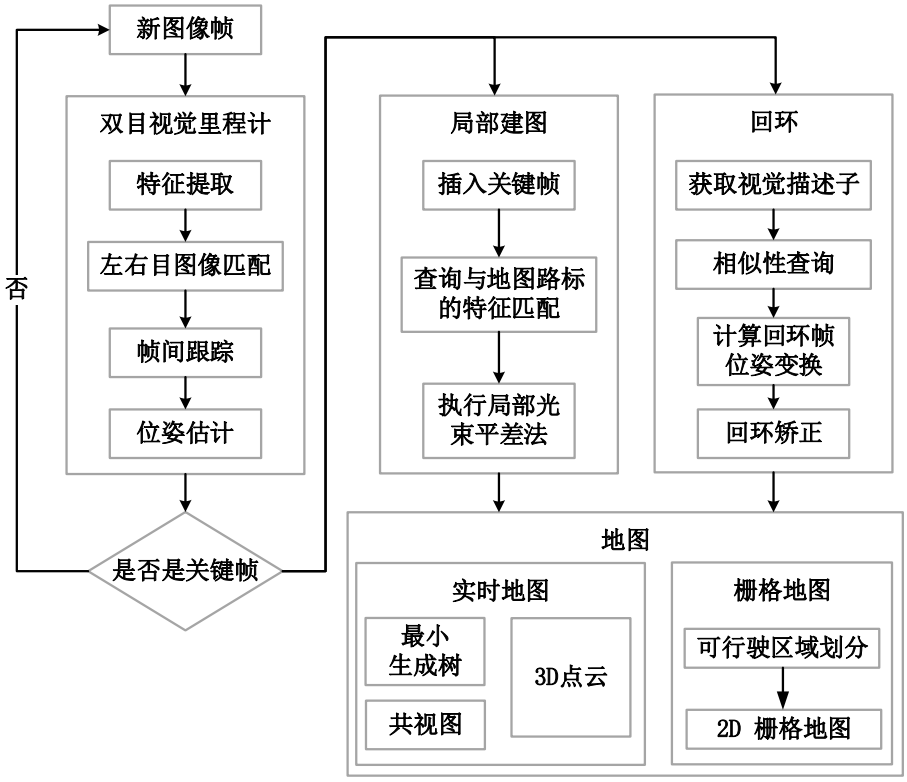


图 2-1 ORB-SLAM3 流程图

2.1.1 主要工作

- 基于 ORB-SLAM3 的双目 VSLAM 实现；
- 代码参考：ORB-SLAM2，ORB-SLAM3。

2.1.2 结论

ORB-SLAM3 有着高效的特征提取，多子图管理模块，并且 SLAM 精度较高，适合基于此

系统开展后续研究。

2.2 基于点线特征的双目视觉 SLAM 方法

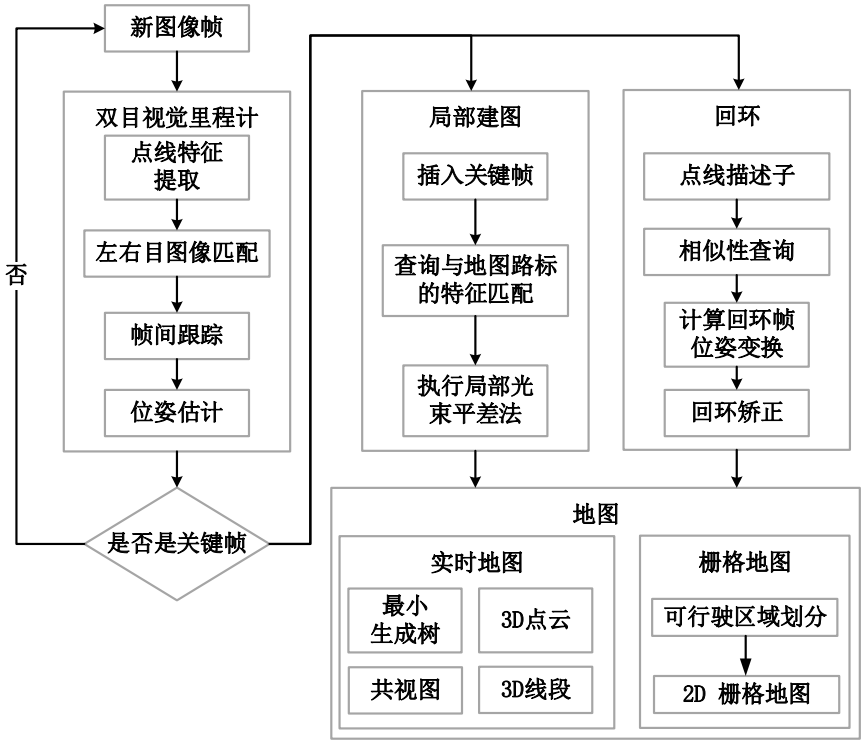


图 2-2 基于点线特征的双目视觉 SLAM 流程图

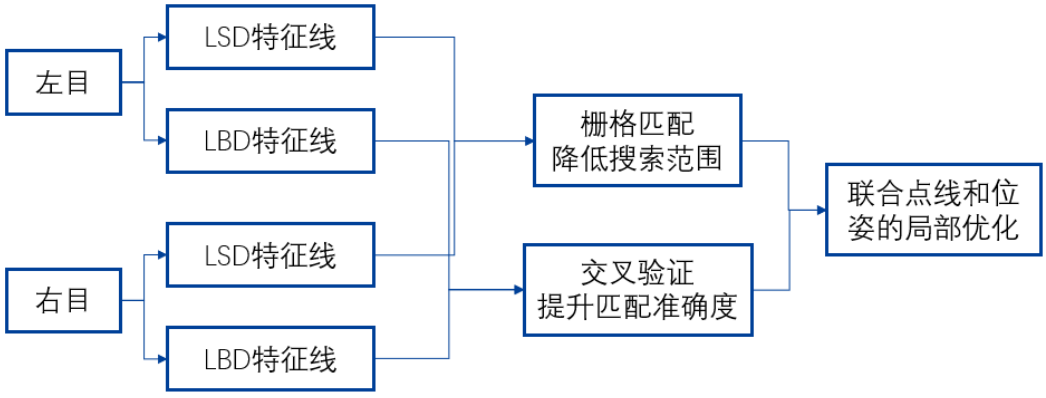


图 2-3 线特征匹配与优化流程图

2.2.1 主要工作

- 线特征提取与描述子提取；
- 基于栅格的线特征匹配；
- 线特征重投影误差构建；
- 基于点线词袋的重定位。

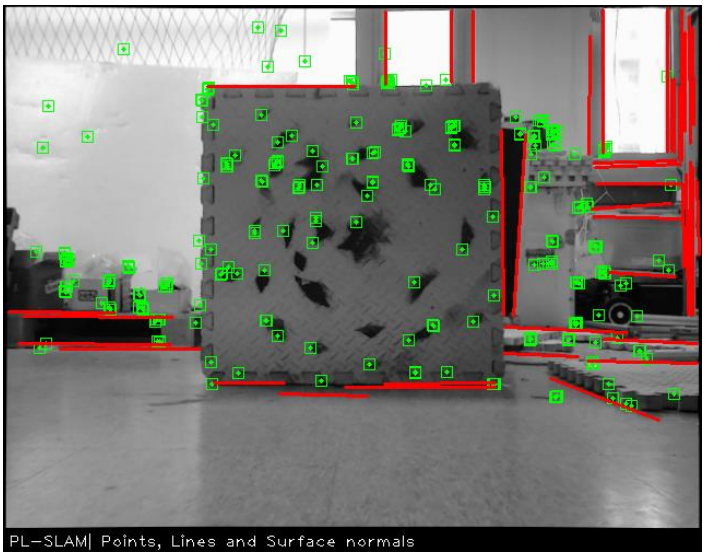


图 2-4 纸箱及其周围环境的点线特征图

图 2-5 桌子及其周围环境的点线特征图

2.2.2 结论

在室内，走廊，弱纹理场景下，加入线特征后，系统鲁棒性没有明显提升；2.线特征提取和优化算法耗时较高，精度没有明显提升，故舍去该方案。

2.3 去除局部地图中的冗余匹配

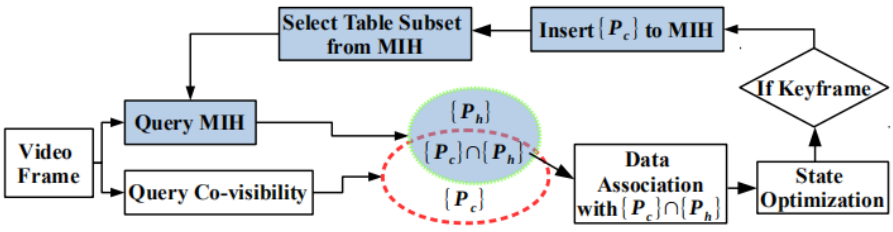


图 2-6 基于 Multi-index Hashing 的局部建图框架

2.3.1 思路解析

- 利用图像中具有共视关系的特征点的描述子构建多个哈希树；
- 关键帧插入时利用之前构建的哈希树对关键帧的特征点进行索引；
- 依照索引结果获得局部地图中关键帧序列的公共特征点；
- 取哈希树得到的特征点与共视关系得到的特征点的交集，达到减小优化特征点数量的目的。

2.3.2 实验结果

- 耗时：统计了不同数据集上的最大耗时和平均耗时，无明显差别（可理解为构建哈希树和索引的时间与该方法节省的优化时间抵消）；
- 精度：在一半数据集上精度得到提升，另一半下降，不好确定该方法的有效性。

2.3.3 结论

该算法对运行速度提升不明显，考虑到该算法与程序中地图点维护有较大关联，可能会影响到系统鲁棒性，因此暂不增加。

2.4 双目立体匹配加速

2.4.1 思路解析

去掉原有的对整幅图片进行矫正，转而只对特征点进行矫正（去畸变）。匹配范围也从极线搜索转为指搜索极线附近的特征点。

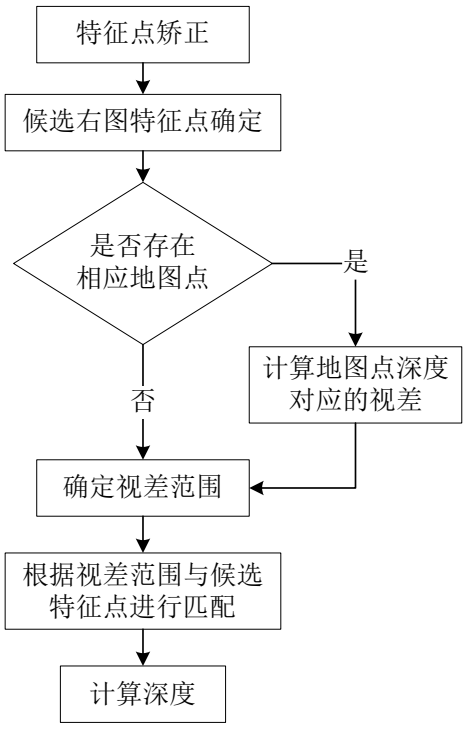


图 2-7 双目立体匹配流程图

2.4.2 主要工作

首先，该算法不对整张图像做矫正，而只对提取的特征点做矫正处理，节省了去畸变过程所使用的时间。其次，由于左图像和右图像的 x 轴原则上处于同一水平线上，考虑到实际误差，该算法依据特征点所在的图像金字塔层数给出误差范围，2*scale，其中 scale 为所在金字塔尺度。接着，该算法判断此特征点是否在局部地图中有对应地图点，如果有，则根据地图点在相机坐标系下的深度 z，求出其对应视差 d，再根据事前设定的视差阈值 threshold，确定该特征点的视差范围为（max（d-thresold，0），min（d+thresold，fx）），其中 fx 为双目相机的焦距；如果没有，视差范围为（0，fx）。接着，结合视差范围，对该特征点的 u 坐标所对应的候选右图像特征点进行遍历，根据特征描述子的距离找到距离最近的匹配点。最后，找到最佳匹配点后，根据双目视差公式计算深度。

2.4.3 结论

原始的立体匹配算法平均运行时长为 7ms 每帧，占视觉里程计二分之一到三分之一的时长，改进后的立体匹配加速算法利用之前估计的深度信息，缩小查询范围，将运行时长降低至 0.5ms 每帧，极大地保证了实时性要求。

2.5 IMU 与轮速计群空间闭式积分

IMU 与轮速计积分的方式有三种：欧拉积分、二阶 Runge-Kutta 积分、群空间闭式积分。其中欧拉积分比较简单，误差主要来源于平移积分过程中认为角度恒定。这在直线运动时不受影响，以及旋转很小时误差影响也较小。平移二阶 Runge-Kutta 积分过程中认为角度为运动前后的均值，误差比欧拉积分小一点。群空间闭式积分是在李群空间上对码盘进行积分，是无误差的闭式积分。并且，本方案提出的群空间闭式积分是在 SE3 空间上，可以减轻地面不平整、起伏带来的对位姿估计的影响。

记上一时刻位姿为T_{t-1}，通过群空间闭式积分得到当前帧的位姿为T_t，则关键在于如何求得T_t^{t-1}。记 t-1 时刻到 t 时刻之间的平均角速度为 w，时间长度为 dt，则旋转变化量如下公式所示：

$$R(w)=\mathrm{Exp}\left(w*\mathrm{dt}\right)$$

其中，Exp 为李代数 so3 到李群 SO3 的映射函数。

记 t-1 时刻到 t 时刻之间的平均速度为 u，轴角$\alpha=w*\mathrm{dt}$，轴角的模长为θ，旋转轴为a。T_t^{t-1}由以下公式得到：

$$T_t^{t-1}=\begin{bmatrix} R(w) & Ju \\ 0^T & 1 \end{bmatrix}$$

其中：

$$J=\frac{\sin\theta}{\theta}I+\left(1-\frac{\sin\theta}{\theta}\right)aa^T+\frac{1-\cos\theta}{\theta}a^{\wedge}$$

2.5.1 结论

该方案可以获得更高的里程计精度。

2.6 IMU 偏置估计

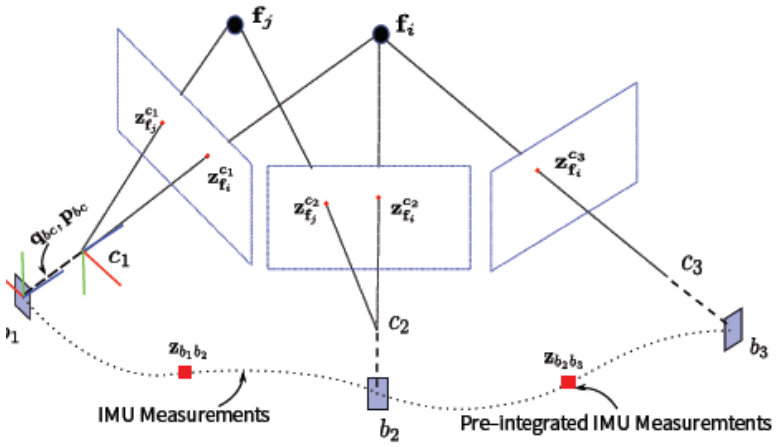


图 2-8 IMU 偏置估计模型

2.6.1 思路解析

因为基础框架 ORB-SLAM 有自己的视觉特征优化框架，因此集成 VINS 系统的紧耦合方法难度较大，但考虑到相机并不会跟着时间漂移，且 ORB-SLAM 通过建立局部地图，连续几帧的位姿变化误差较小，因此采用如上松耦合方案：假想连续几帧的位姿变化误差较小，以此通过最小二乘估计 IMU 偏置。

2.6.2 算法流程

- IMU 预积分，得到关于偏置的雅克比矩阵
- 建立滑动窗口，得到 ORB-SLAM 局部建图优化过后的窗口位姿变化量
- 利用最小二乘进行优化，得到偏置估计
- 设定偏置变化阈值，并基于一阶滤波器对偏置进行平滑
- 采用 ZUPT（零速度更新），检测静止的情况，对 IMU 偏置进行更新，并将此结果与依靠相机得到的 IMU 偏置结果进行加权融合。

2.6.3 实验结果

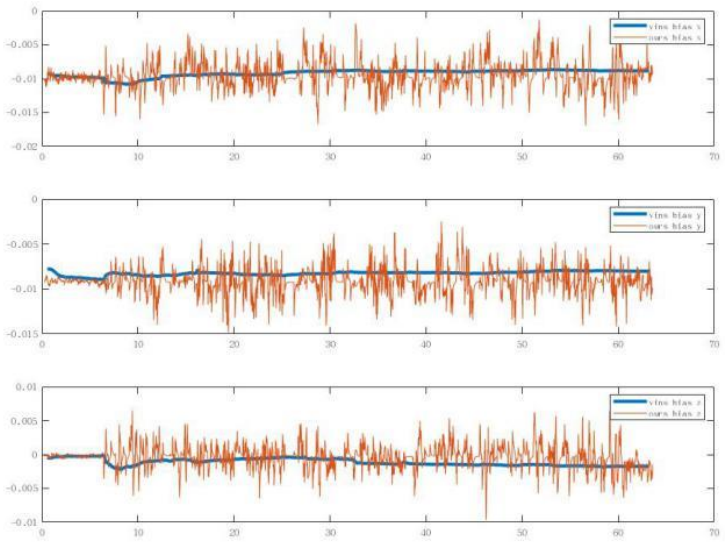


图 2-9 IMU 偏置估计实验结果图

2.6.4 结论

蓝色是 VINS 系统估计的结果，橘红色是我们方案的结果，最大估计误差在 0.005°/s，耗时 1.7ms，可以使用。

2.7 多子图连接与平滑本质图构建

2.7.1 思路解析

- 检测一个条件发生（例如机器人被抱起，轮子空转），则新建一个地图
- 新地图的关键帧放入数据库，寻找与旧地图的重合点，若发现则拼接地图。
- 在一个地图中，检测“丢失条件”发生（例如局部地图中没有足够多的特征点），则根据里程计距离和角度变化，插入不需要维护特征点的关键帧，并将此关键帧的父节点设定为上一帧能正常跟踪的关键帧。

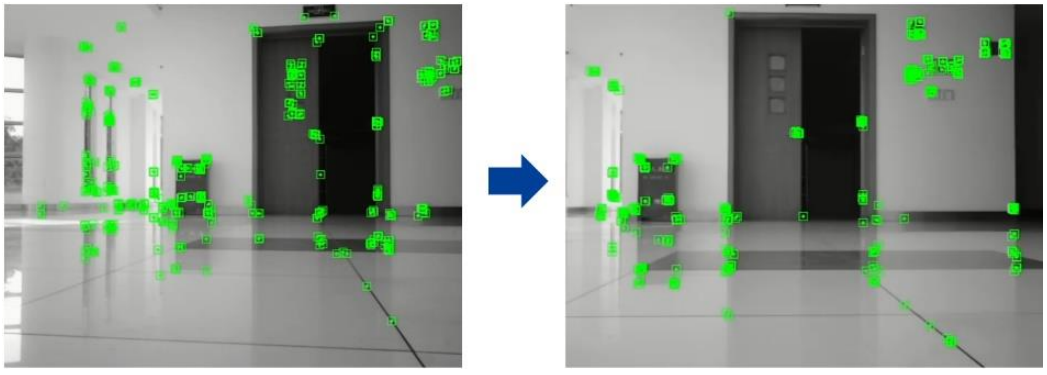


图 2-10 回环检测

图 2-11 多子图下回环矫正

2.8 鲁棒多传感器 SLAM 系统构建

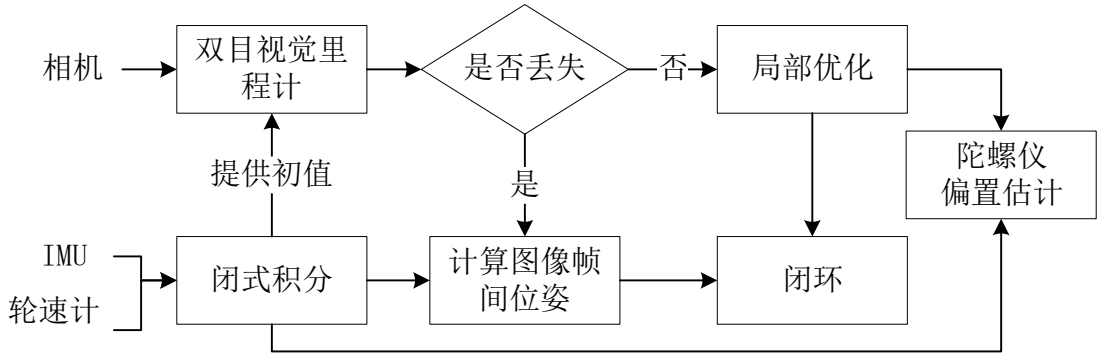


图 2-12 基于多传感器构建 SLAM 系统

视觉里程计不可避免地因为光照变化，快速运动而丢失，本方案使用 IMU 与轮速计群空间闭式积分计算得到的短时相对位姿对系统位姿进行实时更新，并在系统丢失时，用此方法继续给机器人提供实时位姿，并尝试基于此位姿重新对视觉里程计进行初始化。另外，为了保证系统长时间运行，本方案基于 IMU 偏置估计算法，利用滑动窗口内的 vslam 解算的位姿结果对陀螺仪的偏置进行实时估计。最后，为了构建连续的位姿图，本方案基于多子图连接与平滑本质图构建，分别针对视觉里程计和 IMU 与轮速计群空间闭式积分两种位姿更新方式，设定位姿节点之间的边，以用于回环矫正。

如上图所示，在相机采集到的左右图像输入到视觉里程计模块，系统通过判断跟踪到的地图点的个数判断是否丢失。若没有丢失，则正常进入关键帧局部优化；若丢失，则利用 IMU 和轮速计闭式积分得到的多个连续相对位姿量，插值计算得到两帧图像之间的位姿，用于对视觉里程计丢失期间内的位姿补充。并且在下一帧图像到来时，尝试对视觉里程计进行重新初始化，初始化位姿为当前时刻利用 IMU 和轮速计闭式积分插值更新得到的位姿。

在陀螺仪配置估计中，记陀螺仪偏置为 $\delta \mathbf{b}^{g}$，B 为滑动窗口中图像帧的集合，c0 为滑动窗口第一帧所代表的坐标系。假设滑动窗口内，相机视觉里程计未丢失且位姿足够准确，用其对陀螺仪的零飘进行估计，遂构建以下优化方程：

$$\arg\min_{\delta \mathbf{b}^g} \sum_{k \in B} \left\| 2 \left\lfloor \mathbf{q}_{c_0 b_{k+1}}^{-1} \otimes \mathbf{q}_{c_0 b_k} \otimes \mathbf{q}_{b_k b_{k+1}} \right\rfloor_{xyz} \right\|^2$$

其中预积分的一阶泰勒近似，如下公式:

$$\mathbf{q}_{b_k b_{k+1}} \approx \mathbf{q}_{b_k b_{k+1}} \otimes \begin{bmatrix} 1 \\ \frac{1}{2} \mathbf{J}_{bg}^{\mathbf{q}} \delta \mathbf{b}^g \end{bmatrix}$$

因此，求取预积分量对偏置的雅克比矩阵，构建正定方程 HX = b 即可以完成该优化问题的求解。

系统中位姿图的构建分为视觉里程计丢失和成功两种情况。成功情况下，当前关键帧的父节点为与其共视点数量最多的关键帧。丢失情况下，设定关键帧选取条件 $\beta = \Delta t/0.1 + \Delta\phi/10$，其中 Δt 为与上一帧关键帧之间的位移，$\Delta\phi$ 为与上一帧关键帧的旋转轴角的模长（角度制）。当 β 大于 1 时，建立新一帧关键帧，其父节点为最老的一帧视觉里程计跟踪成功的关键帧。按照上述方式，即可构建一个连续的位姿图用于回环矫正。

2.9 算法评测

平移误差百分比 = 估计值与真实值的欧氏距离 / 已走路程

旋转误差百分比 = 估计值与真实值相对旋转的欧拉角的模长 / 已旋转的角度

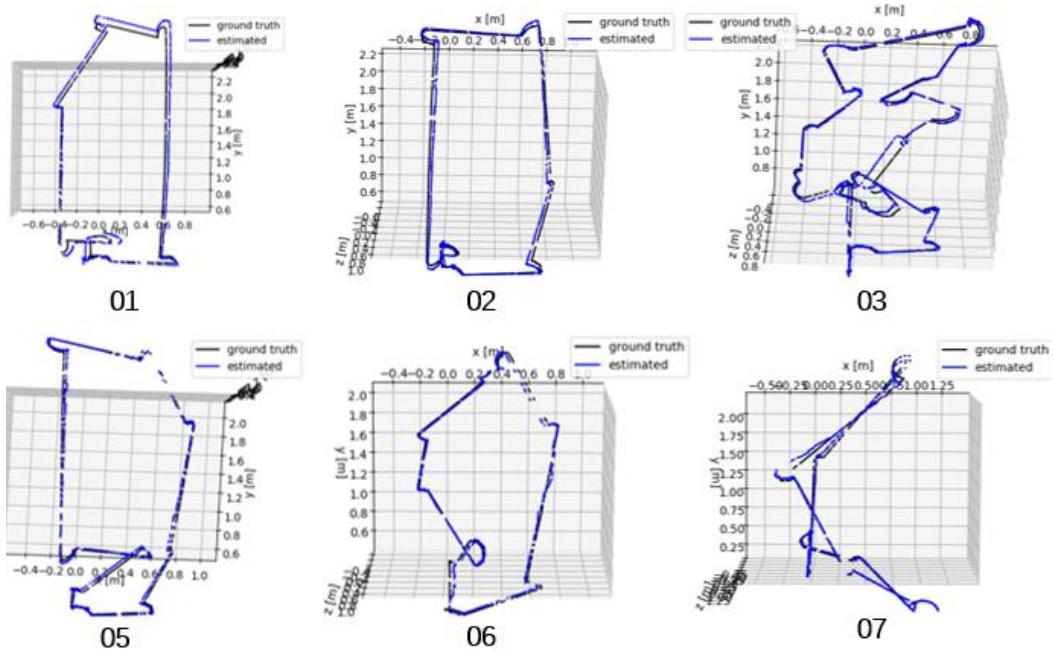


图 2-13 不同运动轨迹下的算法测试结果图

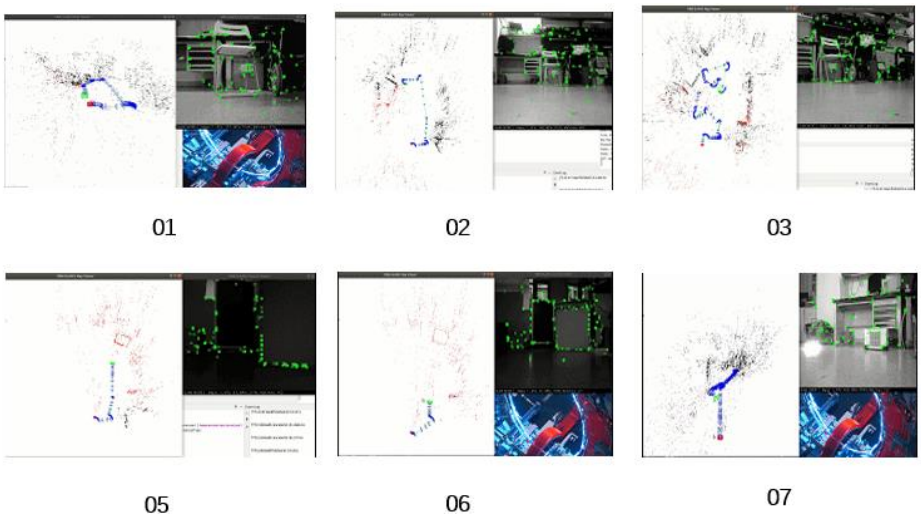


图 2-14 不同测试环境下的算法测试结果图

2.9.1 实验结果

数据集介绍：10 数据集

表 1 ORB-SLAM 测试结果

ORB-SLAM	非实时运行结果（逐帧运行）	实时运行结果
Median Translation Error	0.4655%	0.8592%
Median Rotation Error	0.0009%	0.0830%

运行单帧平均时间为 mean tracking time: 0.5893 s

2.10 地图可视化

2.10.1 2D 栅格地图构建造

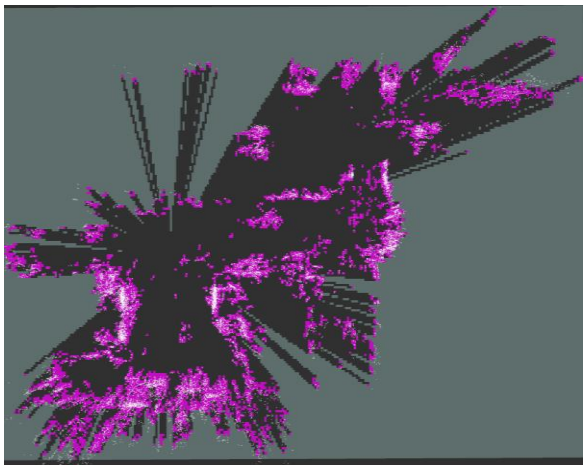


图 2-15 2D 栅格地图

2.10.2 依照线特征与深度图进行半稠密地图构建

图 2-16 在实验室拍摄的原图像

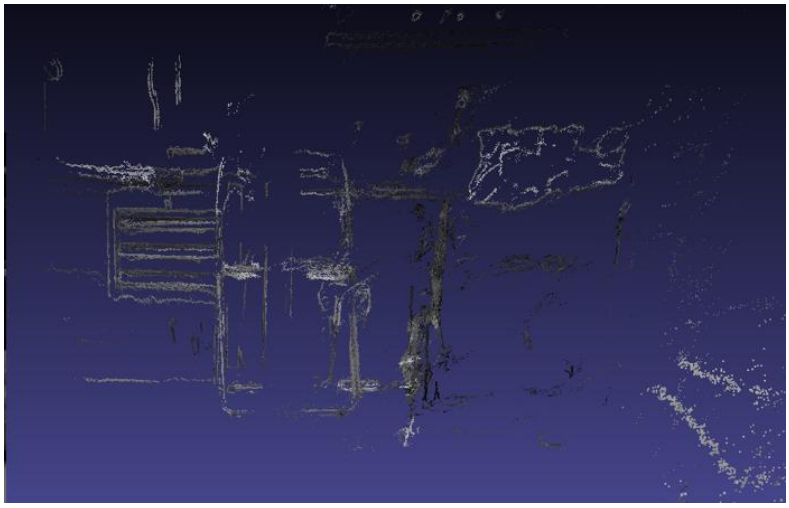


图 2-17 基于线特征和深度构建的半稠密地图

2.10.3 结论

从图 2-17 可知，算法展示了效果较好的一个结果，但由于双目深度估计的准确度仍然不是很理想，导致线段拟合有时会出现较多的错误，因此这个建图方法并不鲁棒。

3 重定位

3.1 点线词典构建

3.1.1 思路解析

- 由 ORB 特征提取器获得点特征描述子集合；
- 由 LSD 检测线特征，LBD 提取线特征描述子集合；
- 基于点线特征集合，实现小规模词典构建。

3.1.2 评价指标

- 训练与测试数据集为同一场景的不同序列；

- Recall_N： 对于一张图片，找到最相似的 N 张，若该 N 张找对了 M 张，则计算真值占比 M/N，统计平均召回率；
- 真值：VICON 获得的轨迹位移小于 20cm，角度小于 10° 认为是属于同一场景。

3.1.3 实验结果

表 2 重定位结果

	ORB	ORB+LBD
Recall_1	99.6516783	99.6516783
Recall_20	99.3263141	99.3587714
Recall_40	98.4456143	98.5366529
Recall_60	95.356502	95.6449229

3.2 词袋法加速

- 已实现接口：FBoW，DBoW2，DBoW3；
- 已实现功能：词典读取，词典保存，词典训练，词袋向量转换，词袋向量评分。

3.2.3 实验结果

表 3 词袋法加速结果

K=10, L=3	加载	保存	训练	词典向量转换
DBoW2	4.9ms	4.2ms	114.0ms	2.7ms
DBoW3	870.9μs	300.4μs	152.1ms	1.6ms
FBoW	24.3μs	217.2μs	116.8ms	2.0ms

4 主方向识别

4.1 方案

主方向识别基于房间的物体大都垂直于墙体摆放的假设，通过检测图像上的灭点得到房间主方向相对于相机的旋转角。主方向识别的过程如图 4-1 所示。对于每一帧新的双目图像，分别对左右两张图像进行 LSD 直线检测、灭点检测，并利用相机投影公式分别得到主方向相对于两个相机的旋转角。然后对双目的检测结果进行双目验证：如果两个相机的旋转角结果的差小于阈值则认为通过双目验证。如果通过了双目验证，则通过轮式里程计得到的偏航角将当前帧的主方向结果转换到第一帧下，并将结果存入容器中，并且等待新一帧图像。如果距离第一帧间隔超过了 N（我们取 50）帧，则利用容器内的所有第一帧检测结果，利用 RANSAC 算法得到最终第一帧主方向检测结果和置信度。如果置信度小于 0.5，那么重新进行主方向识别，直到得到置信度大于 0.5 的结果，并且将此时的主方向识别结果利用里程计的值转换到全局初始帧下。

需要说明的是，主方向识别仅在扫地机器人系统运行初进行，一般 10s 左右可以完成。

主方向成功识别的前提是基本假设的满足，即要求房间中的物体都垂直于墙体摆放。为了得到更好的主方向结果，要求主方向识别过程中扫地机器人缓慢平稳的运动。

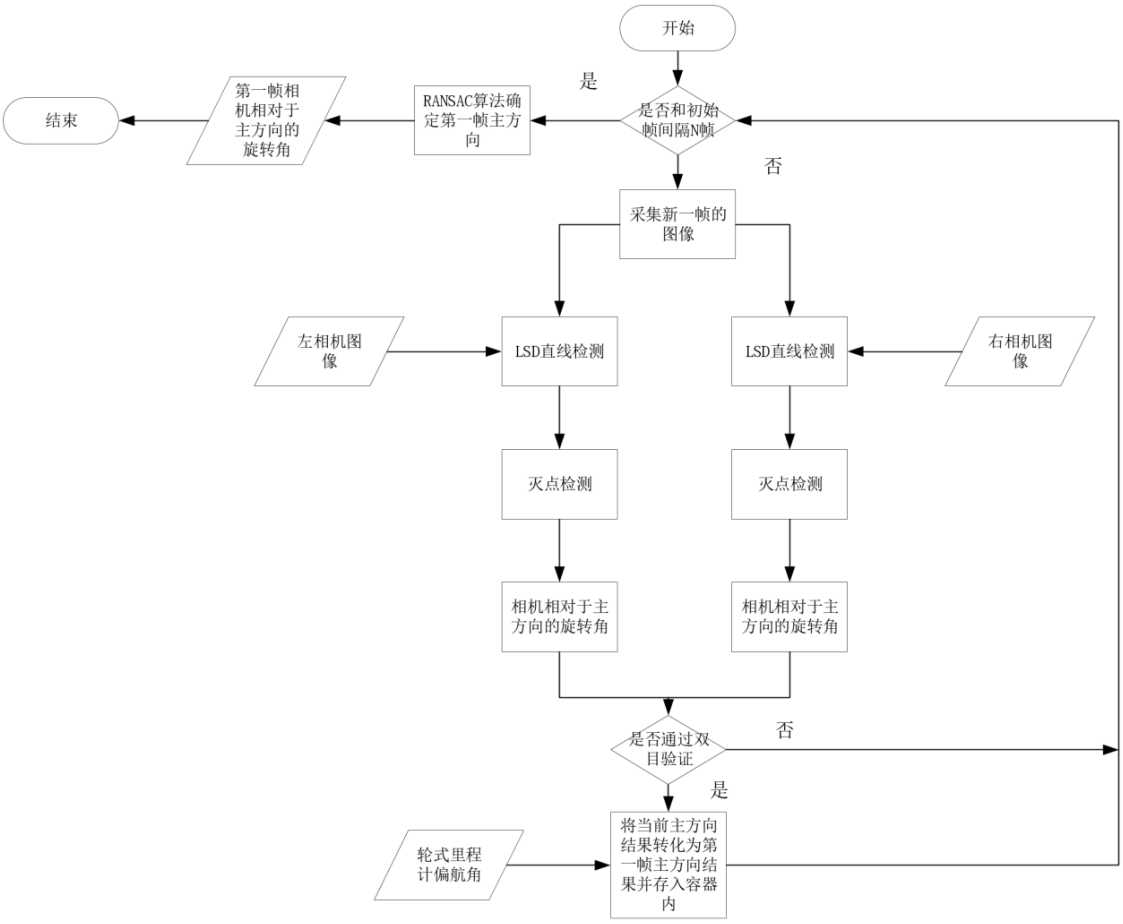


图 4-1 主方向识别流程图

4.2 程序运行

需要修改 data_path save_path

cd /data/ttt/

LD_LIBRARY_PATH=/data/ttt/lib/dep_libs/:/pcl_libs/ ./main_direction_detecti on ./initial/main_direction_detector.yaml

4.3 精度检测

4.3.1 检测方法

在 dataset/10/提供的 2020-09-25-13-30-56.bag 数据集初始的 0-15s 内随机开始进行主方向识别，识别结果保存在

save_path /main_direction_gt.txt 中，各列表示：

(1) 用于主方向识别的 50 帧中第一帧的时间戳；

(2) 房间主方向相对于相机的角度值；

(3) 置信度；

(4) 程序开始运行的全局第一帧时间戳；

(5) 全局第一帧房间主方向相对于相机的角度。

真实值通过 VICON 测量得到，保存在如下文件中：

/main_direction_result/main_direction_gt.txt

其中文档里各列表示：

(1) 时间戳；

(2) 房间主方向相对于相机的角度值。

通过时间戳将主方向检测结果和真实值进行对应和比较。

4.3.2 精度检测结果

主方向检测结果如下，表示 1601011864.18562 时间戳下的主方向偏转角为 2.31592 度，检测置信度为 0.83333，查询真实值文件，得到改时间戳下真实值为 1.0002386054950292 度，所以误差约为 1.3 度。

表 4 主方向识别结果

时间戳	主方向偏转角	检测置信度
1601011864.18562	2.31592°	0.83333

5 障碍物检测和目标识别

5.1 方案

障碍物检测和目标识别模块既需要对障碍物进行检测以实现避障，也需要对羽毛球、麻将、线、狗屎和毛毯这五类物体的位置、大小和类别进行目标识别，为更上层的任务服务。整体过程如图 5-1 所示。

障碍物检测在相机距离地面高度已知的前提下，通过将同一像素纵坐标的视差和地面视差进行比较得到障碍物掩模，然后将障碍物掩模作用与视差图上，利用双目模型和相机投影模型从视差图得到障碍物点云。

物体识别部分首先通过 yolo 网络得到二维检测结果，然后对每一个二维检测框内的区域的像素点根据 3D 图映射到三维空间，对框内点取 x 轴、y 轴和 z 轴各自的平均值作为三维物体检测的结果。

需要注意的是，为了提高 yolo 网络的推理速度，我们使用了 mobile-yolo5 网络，同时在原有网络的基础上进行了剪枝和量化，对模型进行了压缩，最终基于 RKNN 框架在 NPU 上进行模型推理，提高了网络推理速度。

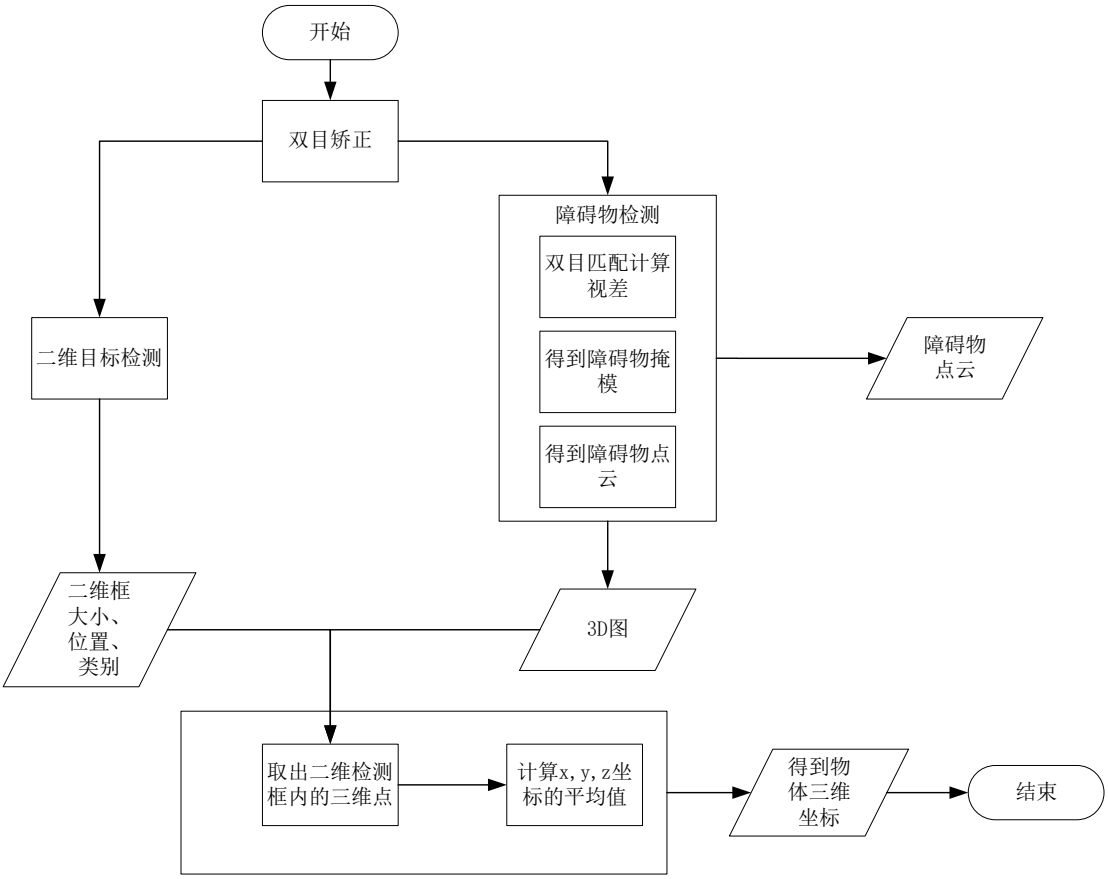
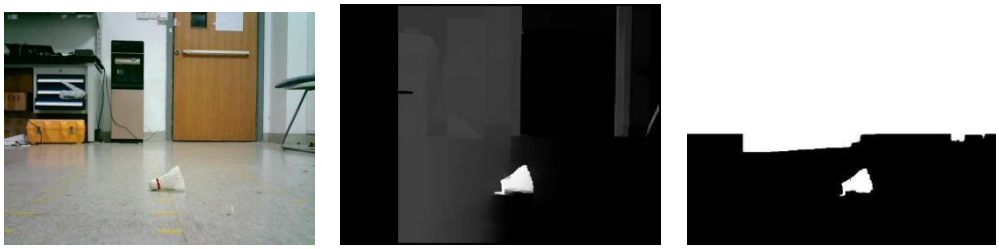


图 5-1 障碍物检测和物体识别流程图

5.2 程序运行

```
# 仅进行障碍物识别
# 需要修改 data_path save_path
cd /data/ttt/
LD_LIBRARY_PATH=/data/ttt/lib/dep_libs/:/pcl_libs/ ./obstacle_detector ./in
i
tial/obstacle_detector.yaml
```

5.3 障碍物检测结果



(a)原始 RGB 图　　(b)视差图　　(c)障碍物掩膜

图 5-2 障碍检测效果图

5.4 精度检测

5.4.1 检测方法

为了检测精度，单独录制了四类物体距离相机深度 0.5m 到 1.4m，横向距离-0.15m,0m，0.15m，包含 240 张图片的数据集，保存在 exp0702 文件夹中。

物体检测结果保存在 obj_det_dataset/object_det_matlab_sgbm.txt 中。各列分别表示：

(1) 帧序号；

(2) 该帧 yolo 检测到的物体数量；

(3) 该物体类别序号（0：羽毛球；1：线；2：麻将；3：狗屎； 4：毛毯）；

(4) 该物体世界坐标系下 x、y、z 坐标(m)。

真实值通过尺子测量得到，保存在 object_det_result/gt.txt 中，其中各列表示：

(1) 类别序号；

(2) 世界坐标系下 x、y、z 坐标(m)。

5.4.2 检测结果

运行指令：

python evaluation/object_det_eval/eval.py path_to_obj_det_dataset/object_det_matlab_sgbm.txt

(1)前四类

表 5 前四类二维度检测结果

Object	True Positive	Total Images	Recall
Badminton	59	59	100%
Wire	60	60	100%
Mahjong	59	60	98.3%
Dog Shit	62	62	100%
Total Average	240	241	99.6%

表 6 前四类三维检测结果

Object	True Positive	Total Images	Recall
Badminton	38	38	100%
Wire	36	40	90%
Mahjong	35	40	87.5%
Dog Shit	34	41	82.9%
Total Average	143	159	89.9%

(2) 毛毯类

从总数据集中抽取 100 张厚度大于 1cm 毛毯的图片作为毛毯的数据集

carpet_test_img，

用于评价毛毯检测精度,结果如下表所示。

表 7 前四类三维检测结果

Object	True Positive	Total Images	Recall
Carpet	98	100	98%

(3) 时间

嵌入式平台上平均单帧运行时间 average_time: 1.29276 s