

# 双目相机和激光雷达的融合 SLAM 研究

刘鸿勋, 王 伟

(南京信息工程大学自动化学院, 江苏 南京 210006)

**[摘要]** 针对基于图优化的激光 SLAM 算法在高相似度的场景中闭环检测出错的问题, 提出使用双目相机进行闭环检测的方法. 使用加入旋转不变性的 FAST 特征点和 BRIEF 描述子进行双目深度估计; 引入局部地图的概念, 使用单帧激光雷达数据与局部地图进行匹配, 提高 SLAM 前端的精度. 使用基于词袋(bag of words, BOW)模型的 k 叉树字典评估图片相似度从而完成闭环检测, 最后构建全局优化问题并求解. 与主流开源激光雷达 SLAM 算法的对比实验表明, 研究内容改善了只使用激光雷达数据进行闭环检测的方法在相似度较高场景下失效的问题, 并且在较大面积场景运行效果明显优于基于滤波的 SLAM 算法.

**[关键词]** FAST 特征点, 图优化, SLAM, 双目视觉, 词袋

**[中图分类号]** TP242.6 **[文献标志码]** A **[文章编号]** 1672-1292(2021)01-0064-08

## Research on Fusion SLAM of Binocular Camera and Lidar

Liu Hongxun, Wang Wei

(College of Automation, Nanjing University of Information Science and Technology, Nanjing 210006, China)

**Abstract:** Aiming at the problem of closed-loop detection errors in the graph-optimized laser SLAM algorithm in high-similarity scenes, the method of using binocular cameras for closed-loop detection is presented in this paper. We use FAST feature points and BRIEF descriptors with rotation invariance for binocular depth estimation. We introduce the concept of local maps, and use single-frame lidar data to match with local maps to improve the accuracy of the SLAM front-end. A k-ary tree dictionary based on the BOW model is used to evaluate the similarity of the pictures to complete the closed-loop detection, and finally, the global optimization problem is constructed and solved. The contrastive experiment with the mainstream open source lidar SLAM algorithms shows that the method improves the problem which the closed-loop detection method that only uses lidar data for closed-loop detection fails in a scene with high similarity, and that the operation effect in a larger area is significantly better than that based on Filtered SLAM algorithm.

**Key words:** FAST feature points, graph optimized, SLAM, binocular vision, bag of words

SLAM 问题可以描述为机器人在未知环境中从一个未知位置开始移动, 在移动过程中根据位置估计和地图进行自身定位, 同时在自身定位的基础上建造增量式地图, 实现机器人的自主定位和导航. 此课题在学术界已经有二三十年的历史, 根据获取环境信息的传感器不同可以分为视觉 SLAM、激光 SLAM 两个大类. 视觉 SLAM 又分为单目、双目、深度(RGBD)相机 3 种. 激光雷达主流的有单线激光、16、32、64、128 线激光等, 价格基本呈指数增长.

激光雷达 SLAM 算法是目前产业界落地使用最多、实际运行表现最稳定的算法, 主要分为两种, 一种是传统的以扩展卡尔曼、粒子滤波为代表的贝叶斯滤波方法<sup>[1]</sup>, 另一种是基于图优化的全局优化处理方法<sup>[2]</sup>. 由于近年来对 SLAM 算法的深入研究, 其应用场景逐步向大规模环境发展. 图优化方法由于支持全局优化、方便消除累积误差的特点, 渐渐成为主流. 但是单线激光雷达获取的点云数据只是在单一的平面内, 近些年来, 经过研究改进, 基于图优化的激光 SLAM 算法中的闭环检测有 ICP、PL-ICP<sup>[3]</sup>、NDT 等方法, 这些方法在结构化、重复度较高的场景容易失效, 导致全局优化失败或得到错误结果.

本文从激光 SLAM 实际应用场景出发,研究了视觉双目相机提取特征及匹配算法<sup>[4]</sup>,采用视觉回环检测,用双目相机的图像提取出改进的 FAST 特征点<sup>[5]</sup>,并使用 BOW 方法进行特征点匹配及闭环的检测,配合基于图优化方法的激光 SLAM 算法,有效提高了激光 SLAM 的回环准确率。

## 1 双目—激光融合 SLAM 框架

### 1.1 运动和观测

把离散时刻  $t=1,2,\dots,k$  机器人的位置记为  $X=\{\mathbf{x}_1,\mathbf{x}_2,\dots,\mathbf{x}_k\}$ . 路标点集合  $Y=\{\mathbf{y}_1,\mathbf{y}_2,\dots,\mathbf{y}_k\}$ ,当机器人在  $\mathbf{x}_k$  位置上观测到某个路标点  $\mathbf{y}_j$ ,就产生了一个观测数据,得到运动方程和观测方程分别为:

$$\begin{cases} \mathbf{x}_k = f(\mathbf{x}_{k-1}, \mathbf{u}_k, \mathbf{w}_k), \\ \mathbf{z}_{k,j} = h(\mathbf{y}_j, \mathbf{x}_k, \mathbf{v}_{k,j}). \end{cases} \quad (1)$$

式中,  $\mathbf{u}_k$  是 imu、里程计等运动传感器的读数(也可以称为运动方程的输入),  $\mathbf{w}_k$  为噪声. 观测方程中的  $\mathbf{y}_i$  为路标点,  $\mathbf{v}_{k,j}$  为此次观测的噪声. 对于在平面内运动的机器人位姿,  $\mathbf{x}_k = [x, y, \theta]^T$ , 同时,运动传感器能测量出机器人在  $\Delta t$  时间内位置和转角的变化量  $\mathbf{u}_k = [\Delta x, \Delta y, \Delta \theta]^T$ , 所以运动方程参数化为:

$$\begin{bmatrix} x \\ y \\ \theta \end{bmatrix}_k = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}_{k-1} + \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta \theta \end{bmatrix}_k + \mathbf{w}_k. \quad (2)$$

对于观测方程,激光雷达观测一个路标点时,测量到的有两个量:路标点与机器人之间的距离  $r$  和夹角  $\varphi$ ,记路标点为  $\mathbf{y} = [p_x, p_y]^T$ ,观测数据为  $\mathbf{Z} = [r, \varphi]^T$ ,激光雷达观测方程的形式为:

$$\begin{bmatrix} r \\ \varphi \end{bmatrix} = \begin{bmatrix} \sqrt{(p_x - x)^2 + (p_y - y)^2} \\ \arctan\left(\frac{p_y - y}{p_x - x}\right) \end{bmatrix} + \mathbf{v}. \quad (3)$$

对于相机,使用小孔成像模型,记路标点坐标为  $\mathbf{y} = [p_x, p_y, p_z]^T$ ,观测数据为  $\mathbf{Z} = [u, v]^T$ ,则观测方程的形式为:

$$\mathbf{Z} \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} p_x \\ p_y \\ p_z \end{pmatrix}. \quad (4)$$

其中  $z$  为路标点与相机成像平面的垂直距离,  $u, v$  为路标点在图像中的像素坐标,  $f_x, f_y, c_x, c_y$  是相机内参,可通过相机标定。

### 1.2 非线性优化求解机器人状态估计

上文建立了一个机器人 SLAM 系统的模型:已知从零时刻开始的机器人运动(imu、里程计)和测量传感器(激光雷达、相机)的读数,求解定位问题和建图问题,图 1 是系统状态的描述图。

使用非线性优化的方法求解这个状态估计问题,设从 1 到  $n$  所有时刻,有  $m$  个路标点。

$$\mathbf{x} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}, \mathbf{y} = \{\mathbf{y}_1, \dots, \mathbf{y}_m\}.$$

那么这个状态估计就是已知输入数据  $u$  和观测数据  $\mathbf{Z}$  的条件下计算状态  $\mathbf{x}, \mathbf{y}$  的条件概率分布:

$$P(\mathbf{x}, \mathbf{y} | \mathbf{Z}, u).$$

利用贝叶斯法则,有:

$$P(\mathbf{x}, \mathbf{y} | \mathbf{Z}, u) = \frac{p(\mathbf{Z}, u | \mathbf{x}, \mathbf{y}) p(\mathbf{x}, \mathbf{y})}{p(\mathbf{Z} | u)} \propto \underbrace{P(\mathbf{Z}, u | \mathbf{x}, \mathbf{y})}_{\text{似然}} \underbrace{P(\mathbf{x}, \mathbf{y})}_{\text{先验}}. \quad (5)$$

式中,  $P(\mathbf{Z} | u)$  与待估计的状态  $\mathbf{x}, \mathbf{y}$  无关,可忽略. 左侧为后验概率,正比于右侧似然和先验的乘积. 无需

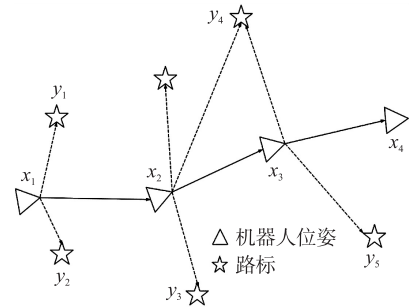


图 1 机器人 SLAM 系统状态

Fig. 1 State of robot SLAM system

求解完整的后验分布,只求一个最优的状态,使该状态下后验概率最大化,

$$(\mathbf{x}, \mathbf{y})_{\text{MAP}}^* = \operatorname{argmax} P(\mathbf{x}, \mathbf{y} | \mathbf{Z}, u) = \operatorname{argmax} P(\mathbf{Z}, u | \mathbf{x}, \mathbf{y}) P(\mathbf{x}, \mathbf{y}). \quad (6)$$

参考上文中的观测模型,对于某次观测  $\mathbf{Z}_{k,j} = \mathbf{h}(\mathbf{y}_j, \mathbf{x}_k) + \mathbf{v}_{k,j}$ ,假设噪声项  $\mathbf{v}_k \sim N(0, Q_{k,j})$ ,则观测数据的条件概率为  $P(\mathbf{Z}_{k,j} | \mathbf{x}_k, \mathbf{y}_j) = N(\mathbf{h}(\mathbf{y}_j, \mathbf{x}_k), Q_{k,j})$ ,它是一个高斯分布. 取其负对数,

$$-\ln(P(\mathbf{x})) = \frac{1}{2} \ln((2\pi)^N \det(\sum)) + \frac{1}{2} (\mathbf{x} - u)^T \sum^{-1} (\mathbf{x} - u). \quad (7)$$

第一项与  $x$  无关可以省略,代入观测方程,相当于求

$$\operatorname{argmin}((\mathbf{Z}_{k,j} - \mathbf{h}(\mathbf{x}_k, \mathbf{y}_j))^T Q_{k,j}^{-1} (\mathbf{Z}_{k,j} - \mathbf{h}(\mathbf{x}_k, \mathbf{y}_j))). \quad (8)$$

定义各次输入和观测数据与模型之间的误差:

$$\begin{aligned} \mathbf{e}_{u,k} &= \mathbf{x}_k - f(\mathbf{x}_{k-1}, \mathbf{u}_k), \\ \mathbf{e}_{z,k,j} &= \mathbf{Z}_{k,j} - \mathbf{h}(\mathbf{x}_k, \mathbf{y}_j). \end{aligned} \quad (9)$$

最小化所有时刻的误差等价于求最大似然估计,转化为一个最小二乘问题,

$$\min J(\mathbf{x}, \mathbf{y}) = \sum_k \mathbf{e}_{u,k}^T \mathbf{R}_k^{-1} \mathbf{e}_{u,k} + \sum_k \sum_j \mathbf{e}_{z,k,j}^T \mathbf{Q}_{k,j}^{-1} \mathbf{e}_{z,k,j}. \quad (10)$$

式中,  $\mathbf{e}_{u,k}$  是运动误差,只与  $\mathbf{x}_k$  和  $\mathbf{x}_{k-1}$  有关,  $\mathbf{e}_{z,k,j}$  是观测误差,只与  $\mathbf{x}_k, \mathbf{y}_j$  有关.

## 2 双目相机前端处理

由式(4)可知,单个相机的观测方程中缺少了图像点的深度信息  $Z$ ,现使用双目相机和改进的 FAST 角点检测匹配算法进行深度信息恢复,并选择关键帧,为后端闭环检测做准备.

### 2.1 改进的 FAST 角点检测算法

FAST 特征点的原理:检测图像中任意一个像素  $P_{ix}$ ,它的亮度为  $L_{P_{ix}}$ ,以此像素为中心,选取半径为 3 的圆上的 16 个像素点,如果有连续 11 个点的亮度大于  $L_{P_{ix}} * 1.2$  或小于  $L_{P_{ix}} * 0.8$ ,那么认为像素  $P_{ix}$  是一个特征点,如图 2 所示.

为了比较各个特征点提取算法的性能,对 SIFT、SURF、ORB、FAST 特征点提取算法在 intel-i5-8250U 下用 5 张测试图片提取特征点数量和用时.

由表 1 可见,FAST 算法可提取大量图像特征点,且检测速率比 SIFT 提高了两个数量级. 因此,本研究采用 FAST 角点作为特征点检测算法. SIFT、SURF、ORB、FAST 特征点提取用时分别为 5.42 s、4.35 s、0.24 s、0.06 s.

表 1 4 种特征点提取数量

| 图片编号 | SIFT  | SURF  | ORB | FAST   |
|------|-------|-------|-----|--------|
| 1    | 2 582 | 3 651 | 500 | 1 2147 |
| 2    | 1 311 | 2 606 | 500 | 4 293  |
| 3    | 5 446 | 6 611 | 500 | 16 553 |
| 4    | 4 592 | 6 033 | 500 | 706    |
| 5    | 1 979 | 4 212 | 500 | 10 676 |

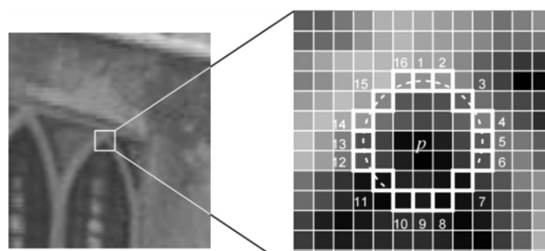


图 2 FAST 特征点

Fig. 2 FAST feature points

但是机器人在运动过程中的视角变化较大, FAST 角点过于简单,需要添加旋转不变性. 在  $P_{ix}$  点的邻域  $S$  内计算灰度质心,定义邻域  $S$  的矩为:

$$m_{pq} = \sum_{u,v} x^p y^q I(u, v). \quad (11)$$

式中,  $I(u, v)$  为像素坐标  $(u, v)$  点的灰度值,  $(u, v) \in S$ , 令  $p, q$  分别为 0 和 1, 则质心位置:

$$C = \left( \frac{M_{10}}{M_{00}}, \frac{M_{01}}{M_{00}} \right). \quad (12)$$

定义 FAST 特征点的方向为:

$$A = (\operatorname{atan} 2(M_{01}, M_{10})). \quad (13)$$

## 2.2 特征描述子及双目特征匹配

使用 BRIEF-128 描述子,选取特征点附近的 128 对像素点,选取方式如图 3 所示.

取出点集  $\mathbf{P}_S = \begin{pmatrix} u_1 & u_2 & \cdots & u_{128} \\ v_1 & v_2 & \cdots & v_{128} \end{pmatrix}$ ,上节中得到的特征点方向角度为  $A$ ,化为旋转矩阵:

$$\mathbf{R}_A = \begin{bmatrix} \cos A & \sin A \\ -\sin A & \cos A \end{bmatrix}.$$

则旋转后的点集  $\mathbf{S}_A = \mathbf{R}_A \mathbf{P}_S$ ,特征描述子为:

$$g_{128}(i, A) := f_{128}(I) I(u_i, v_i) \in \mathbf{S}_A. \quad (14)$$

使用 Hamming 距离判定它们的匹配程度,我们认为当前特征点与最邻近点的 Hamming 距离越小且与次临近点的 Hamming 距离越大,匹配的效果越好.对双目相机的图像分别做特征提取和描述子计算,经过匹配可以得到一组从两个角度同时拍摄到的图像点,如图 4 所示.

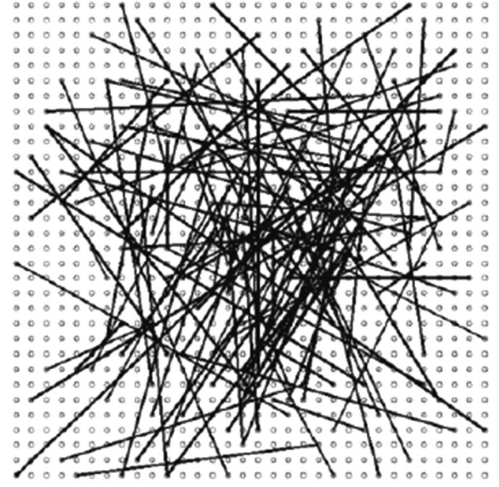


图 3 特征点周围的 128 个点

Fig. 3 128 pairs of points near feature point

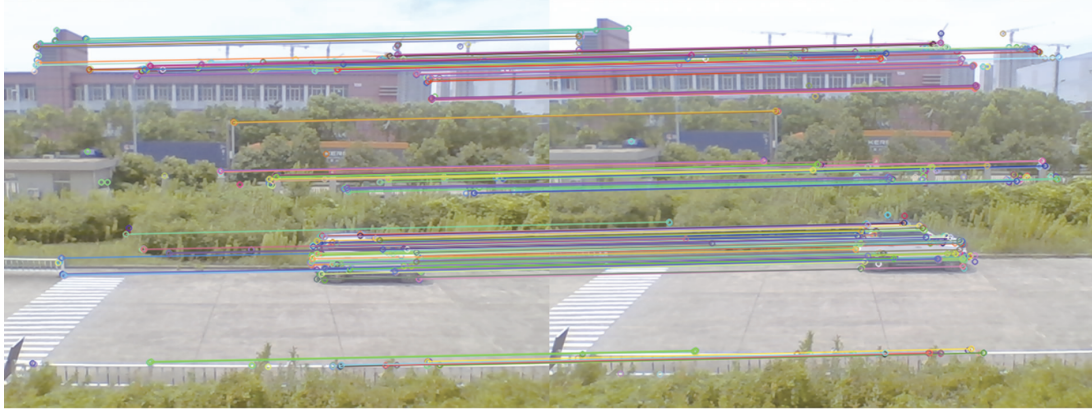


图 4 特征点匹配结果

Fig. 4 Feature points match

## 2.3 双目相机模型和特征点深度计算

经过特征匹配,得到物理世界中  $B$  点在两个相机图像中的投影点  $P_L$  和  $P_R$ ,左眼相机和右眼相机的中心距离为基线的长度  $b$ ,相机焦距为  $f$ ,建立如图 5 所示的双目相机几何模型.

图 5 中, $z$  为  $B$  点与相机的垂直距离,是我们想要恢复的深度信息, $u_L$  为  $P_L$  的像素横坐标, $u_R$  为  $P_R$  的像素横坐标.由相似三角形原理可得:

$$\frac{z-f}{z} = \frac{b-u_L+u_R}{b}. \quad (15)$$

整理可得:

$$z = \frac{f * b}{u_L - u_R}. \quad (16)$$

以左目相机中心为原点,建立相机坐标系  $O-x-y-z$ ,如图 6、图 7 所示.设  $B$  点在相机坐标系中的坐标为  $(x_c, y_c, z_c)$ ,此时  $B$  点的  $z$  坐标、在左目图像中的像素坐标  $(u_L, v_L)$  均已知,由右图中的相似三角形关系可得:

$$\frac{x_c}{u_L} = \frac{z_c}{f}, \frac{y_c}{v_L} = \frac{z_c}{f}.$$

整理可得:

$$x_c = \frac{z_c * u_L}{f}, y_c = \frac{z_c * v_L}{f}. \quad (17)$$



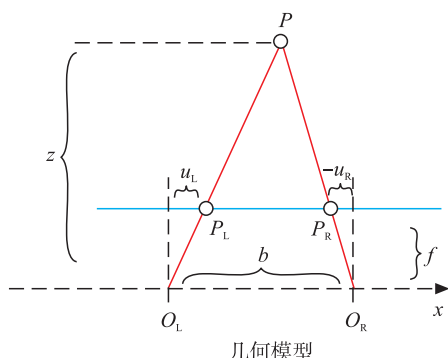


图 5 双目相机视差模型

Fig. 5 Binocular camera parallax model

结合式(16)和式(17),可以计算出所有特征点在相机坐标系下的三维坐标,每一组图像算出的特征点作为一帧点云数据.同时,为了保证运算速度和节省内存资源,根据以下几个原则选取关键帧进行保存:

- (1)当前帧至少提取了 50 个有效的特征点.
- (2)当前帧中的特征点与上一帧关键帧相同点的数量少于 90%.
- (3)局部建图处于空闲状态,或者超过 20 帧图像未保存关键帧.

### 3 地图构建和全局优化

#### 3.1 激光雷达点云帧间匹配

激光雷达点云数据  $H = \{h_k\}$ ,  $k = 1, \dots, K, h_k \in \mathbf{R}^2$ . 设激光雷达的位姿为  $T_\xi$ ,则点云中每个数据点  $B_L$  的全局坐标为:

$$T_\xi p = \begin{pmatrix} \cos \xi_\theta & -\sin \xi_\theta \\ \sin \xi_\theta & \cos \xi_\theta \end{pmatrix} B_L + \begin{pmatrix} \xi_x \\ \xi_y \end{pmatrix}. \quad (18)$$

构建一个概率网格  $G: rZ \times rZ \rightarrow [p_{\min}, p_{\max}]$ ,如下图所示,这些分立的网格点可以设置分辨率大小,比如我们设置为 5 cm. 每个网格点的数值代表它是障碍物点的概率,我们定义每个网格点包含了它所在的方形区域内所有的点,如图 8 所示.

为了提高帧间匹配的稳定性,我们把连续的多帧点云数据累积起来,形成“局部地图”,然后用点云与这个“局部地图”进行匹配. 设置两个点集“hit”和“miss”(如图 9 所示),同时设置初始概率  $p_{\text{hit}}$  和  $p_{\text{miss}}$ ,插入点云时,把点云所在的网格点记为“hit”,把点云与机器人之间连线的所有网格点记为“miss”,如果某网格点是第一次被观测到,就把它的概率  $P_m$  相应地赋值为  $p_{\text{hit}}$  或  $p_{\text{miss}}$ ,如果以前已经观测到过,则以下列公式更新它的概率值:

$$\text{odds}(P_m) = \frac{P_m}{1 - P_m}. \quad (19)$$

$$G_{\text{new}}(x) = \text{clamp}(\text{odds}^{-1}(\text{odds}(G_{\text{old}}(x)) \cdot \text{odds}(p_{\text{hit}}))). \quad (20)$$

式(10)是计算最优插入位置  $\xi$  的最小二乘问题:

$$\arg \min_{\xi} \sum_{k=1}^K (1 - G_{\text{smooth}}(T_\xi h_k))^2. \quad (21)$$

式中,  $T_\xi h_k$  表示把点云坐标转换到局部地图坐标系的过程,  $G_{\text{smooth}}$  函数是更新概率值函数的平滑版本,使用 ceres 和双三次插值方法. 实验结果显示,概率值计算结果有可能超过 1,但是由于求最小二乘的过程不受全概率理论约束,故不会影响优化结果. 求解这个最小二乘问题的方法可以参考文献[6-7].

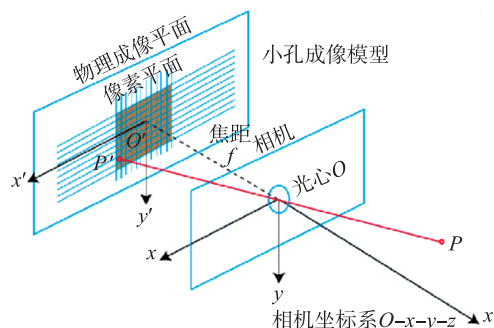


图 6 相机坐标系

Fig. 6 Camera coordinate system

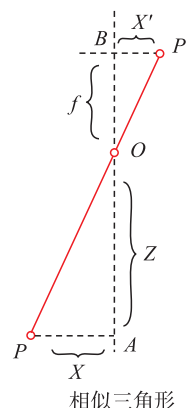


图 7 相机坐标系俯视图

Fig. 7 Top view of camera coordinate system

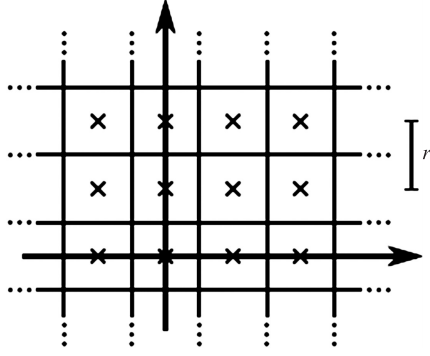


图8 概率网格

Fig. 8 Probability grid

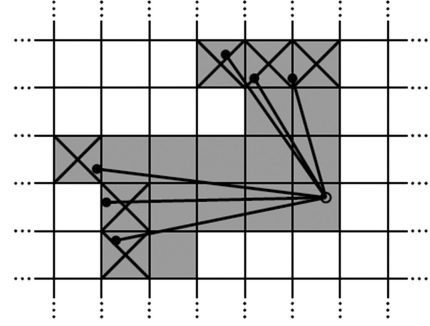


图9 miss 和 hit 的标记规则

Fig. 9 Rule of miss and hit

### 3.2 闭环检测

图像闭环检测使用视觉词袋<sup>[8]</sup>度量两幅图像的相似程度. 经离线采集大量图像, 提取出  $N$  个特征点, 使用  $K$ -means 方法将它们归成  $k$  个单词的词典:

- (1) 随机选取  $k$  个中心点:  $c_1, \dots, c_k$ .
- (2) 计算每个样本与每个中心点的距离, 取最小的作为它的归类.
- (3) 重新计算每个类的中心点.
- (4) 如果每个中心点都变化很小, 则算法收敛结束, 否则返回第二步.

使用 TF-IDF 模型<sup>[9-10]</sup> 计算相似度, 在建立字典时计算 IDF, 统计某个节点  $w_i$  中特征点数量占所有特征点数量的比例. 设一共有  $n$  个特征,  $n_i$  个  $w_i$ , 该单词的  $I$  为:

$$I_i = \lg \frac{n}{n_i}. \quad (22)$$

设图像  $A$  中单词  $w_i$  出现了  $l_i$  次, 而单词一共出现了  $l$  次,  $F$  即为:

$$F_i = l_i / l. \quad (23)$$

则  $w_i$  的权重等于  $F$  与 IDF 的乘积:

$$\eta_i = F_i \times I_i. \quad (24)$$

用一个向量  $\mathbf{v}_A$  描述图像  $A$ :

$$A = \{(w_1, \eta_1), (w_2, \eta_2), \dots, (w_N, \eta_N)\} \stackrel{\text{def}}{=} \mathbf{v}_A. \quad (25)$$

参考文献[11]中的  $L_1$  范数形式, 描述两个向量  $\mathbf{v}_A, \mathbf{v}_B$  的差异程度:

$$s(\mathbf{v}_A - \mathbf{v}_B) = 2 \sum_{i=1}^N |\mathbf{v}_{Ai}| + |\mathbf{v}_{Bi}| - |\mathbf{v}_{Ai} - \mathbf{v}_{Bi}|. \quad (26)$$

### 3.3 全局优化

经过闭环检测后, 判断检测结果, 比如分数大于 70 (总分 100), 则认为找到了足够相似的图像匹配, 然后把这个匹配结果作为一个全局“约束”加入到优化问题中, 根据第 2 章节中的问题模型, 将其中的观测误差具体化为:

$$\arg \min_{I_m, I_s} \frac{1}{2} \sum_{ij} \rho(E^2(\xi_i^m, \xi_j^s; \Sigma_{ij}, \xi_{ij})). \quad (27)$$

式中, 为了保证残差最小化时的收敛速度和稳定性, 使用 Huber Loss 函数  $\rho$  对观测误差进行封装,  $I_m$  是关键帧  $A$  在全局坐标系下的位姿,  $I_s$  是与  $A$  建立闭环的关键帧  $B$  在全局坐标系下的位姿, 它们之间的约束通过相对位姿  $\xi_{ij}$  和协方差  $\Sigma_{ij}$  来描述, 对于一组关键帧  $i$  和  $j$ , 相对位姿  $\xi_{ij}$  描述的是分别采集到该帧的时刻机器人所处的位置, 这个位置由 3.1 中激光雷达的连续的帧间匹配给出, 协方差  $\Sigma_{ij}$  的由闭环检测算法给出.  $E$  函数是残差函数, 在这个约束问题中具体形式为:

$$E^2(\xi_i^m, \xi_j^s; \Sigma_{ij}, \xi_{ij}) = e(\xi_i^m, \xi_j^s; \xi_{ij})^T \Sigma_{ij}^{-1} e(\xi_i^m, \xi_j^s; \xi_{ij}). \quad (28)$$

$$e(\xi_i^m, \xi_j^s; \xi_{ij}) = \xi_{ij} - \begin{pmatrix} R_{\xi_i^m}^{-1}(t_{\xi_i^m} - t_{\xi_j^s}) \\ \xi_{i;\theta}^m - \xi_{j;\theta}^s \end{pmatrix}. \quad (29)$$

4 实验

实验部分,使用双目相机和一台 peppfuches 扫描范围为 30 m 的单线激光雷达,将相机和激光雷达检测到的闭环约束及优化效果对比,然后给出约束的数量和准确率,并验证最终的建图效果.

4.1 激光雷达的局部定位和建图

首先是点云与局部地图的匹配,这个过程在 SLAM 问题中被称为“前端”,衡量它的精度方法与里程计的评估方法相同,表 2 是采用 5 cm 分辨率局部地图,通过扫描匹配得到的局部运动误差.可见,前端匹配精度在 0.7%左右,也就是机器人运动 100 m 约产生 0.7 m 的误差,图 10 是两张实验场地内构建的局部地图.

表 2 局部误差

Table 2 Local error

| 实际运动<br>距离/m | 扫描匹配计<br>算结果/m | 绝对误<br>差/% | 百分比<br>误差/% |
|--------------|----------------|------------|-------------|
| 4.09         | 4.08           | -0.01      | -0.2        |
| 5.40         | 5.43           | 0.03       | 0.6         |
| 8.67         | 8.74           | 0.07       | 0.8         |
| 15.09        | 15.20          | 0.11       | 0.7         |
| 15.12        | 15.23          | 0.11       | 0.7         |

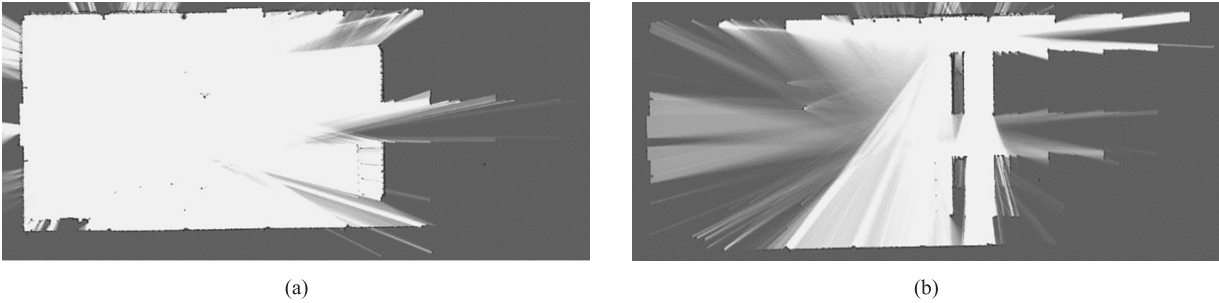


图 10 局部地图

Fig. 10 Submap

4.2 视觉与激光闭环检测对比

目前学术界流行的使用激光雷达进行闭环检测的方法是 Real-Time Correlative Scan Matching<sup>[12]</sup>,图 11 是在实验场景中的闭环检测效果,蓝色线连接的是使用激光雷达 scan-match 匹配的相似度较高的位置,它们明显不是相同位置,而是出现了误检测,从而导致地图出现偏差,如图 12 所示,使用图像进行闭环检测的建图结果如图 13 所示.

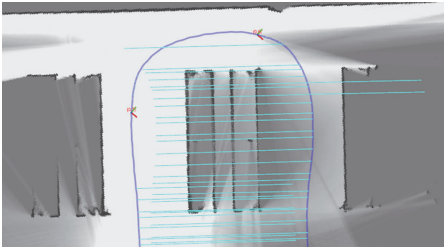


图 11 错误的闭环约束

Fig. 11 Wrong loop closure

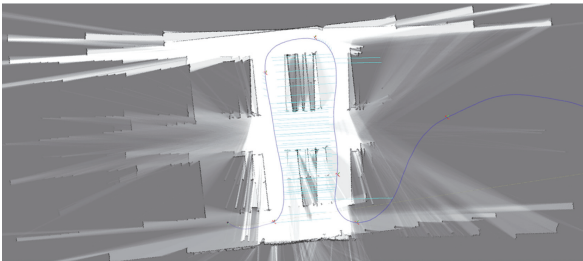


图 12 激光闭环的地图

Fig. 12 Map by laser-loop closure

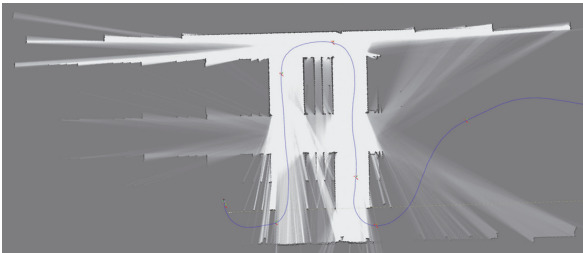


图 13 视觉闭环的地图

Fig. 13 Map by visual-loop closure

表 3 是在不同场景中分别使用激光和视觉闭环检测测试出的约束数量及准确率. 可见,使用视觉进行检测闭环约束的方法明显提高了准确率,在大部分场景下可保证 99% 以上,即使出现了一两个错误的闭环约束,由于损失函数的加入,最小化误差算法仍然能给出正确的优化结果,图 14 是使用视觉闭环构建的完整的实验场地地图.

表 3 全局约束

Table 3 Global constraints

| 场景编号 | 约束数量(激光/视觉) | 准确率(激光/视觉)  |
|------|-------------|-------------|
| 1    | 971/532     | 98.1%/100%  |
| 2    | 5786/1000   | 75.6%/99.9% |
| 3    | 916/486     | 97.2%/100%  |
| 4    | 1857/894    | 82.1%/99.8% |
| 5    | 554/201     | 99.8%/100%  |

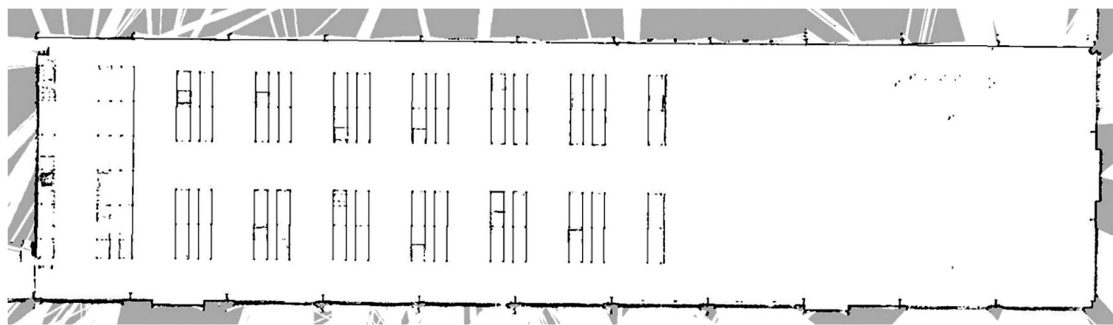


图 14 完整地图

Fig. 14 Full map

## 5 结论

激光雷达传感器的优势是工作稳定,对环境、光照影响不敏感,但是数据量小,感知能力有限;而相机的特点则是数据量非常大,双目相机又可以解决图像深度问题. 使用双目相机替代激光的回环检测很好地解决了平面内相似度较高的场景下的误匹配问题.

## [参考文献] (References)

- [1] SEBASTIAN T. Probabilistic robotics[M]. Beijing:China Machine Press,2017.
- [2] Aisha W B. Dynamic pose graph SLAM:long-term mapping in low dynamic environments[C]//Intelligent Robots and Systems. Vilamoura,Portugal,2012.
- [3] 张恒,徐万红,张禹. 激光雷达与深度相机融合的 SLAM 技术研究[J]. 机械工程师,2020,47(7):22-24.
- [4] 陈庆伟,李民东,罗川,等. 视觉 SLAM 中图像特征点提取与匹配算法研究[J]. 现代制造工程,2019,41(6):44-45.
- [5] 高翔,张涛. 视觉 SLAM 十四讲[M]. 北京:电子工业出版社,2017:132-138.
- [6] 刘国林,姜岩,陶华学. 非线性最小二乘参数平差[J]. 测绘学报,1998,27(3):224-230.
- [7] 吴玲,刘忠,卢发兴. 全局收敛高斯-牛顿法解非线性最小二乘定位问题[J]. 火控雷达技术,2003,32(1):3-4.
- [8] KENSUKE K,KANJI T. Multi-scale bag-of-features for large-size map retrieval[C]//Robotics and Biomimetics. Tianjin: IEEE Computer Press,2010:961-966.
- [9] SIVIC J,ZISSERMAN A. Video google:a text retrieval approach to object matching in videos[C]//Computer Vision:IEEE, 2003:1470-1477.
- [10] ROBERTSON S. Understanding inverse document frequency:on theoretical arguments for IDF[J]. Journal of Documentation, 2004,60(5):503-520.
- [11] NISTER D,STEWENIUS H. Scalable recognition with a vocabulary tree[C]//Computer Vision and Pattern Recognition: IEEE,2006:2161-2168.
- [12] OLSON E B. Real-time correlative scan matching[C]//IEEE International Conference on Robotics and Automation:IEEE, 2009.

[责任编辑:陈 庆]