

RGB-DSLAM 前端视觉里程计的优化算法

董静薇^{1,2}, 蒋燚铭^{1,2}, 韩知雨^{1,2}

(1. 哈尔滨理工大学 测控技术与仪器黑龙江省高校重点实验室, 哈尔滨 150080;

2. 哈尔滨理工大学 测控技术与通信工程学院, 哈尔滨 150080)

摘 要: 针对视觉同步定位与地图构建领域中基于深度相机传感器的算法前端跟踪耗时长, 精度差的问题, 提出了一种 RGB-D SLAM 算法的前端视觉里程计实现方案。依据传统的视觉里程计实现方案, 采用 ORB 特征提取算法对图像进行特征提取并计算描述子, 与局部地图中地图点进行特征匹配, 最后采用 PnP 算法计算相机位姿。为提高算法跟踪的精准度, 提出在计算相机位姿后, 利用三角测量进行局部地图中地图点深度的更新。实验采用 TUM 数据库中 fr1 系列数据集进行测试, 结果表明, 该算法在保证实时性的同时, 使跟踪过程中的均方根误差平均减小了 9.21%。且 fr1 系列两种数据集通过视觉里程计最后生成点云地图的图像信息熵比传统 RGB-D SLAM 系统平均提高了 11.21%。

关键词: RGB-D SLAM; ORB 特征提取; PnP 算法; 三角测量; 局部地图

DOI: 10.15938/j.jhust.2020.06.023

中图分类号: TP242.6 **文献标志码:** A **文章编号:** 1007-2683(2020)06-0157-08

RGB-DSLAM Visual Odometry Optimization Algorithm

DONG Jing-wei^{1,2}, JIANG Yi-ming^{1,2}, HAN Zhi-yu^{1,2}

(1. School of Measurement and Control Technology and Communications Engineering, Harbin University of Science and Technology, Harbin 150080, China;

2. The Higher Educational Key Laboratory for Measuring and Control Technology and Instrumentations of Heilongjiang Province, Harbin University of Science and Technology, Harbin 150080, China)

Abstract: In view of the long time-consuming and poor precision of RGB-D SLAM algorithm in the field of visual SLAM, an implementation scheme of RGB-D SLAM visual odometry is proposed. Based on the traditional visual odometer implementation scheme, the ORB feature extraction algorithm is used to extract the features of the image and calculate the descriptor to match the features of the map points in the local map. Finally, the PnP algorithm is used to calculate the camera pose. In order to ensure the accuracy of the tracking, it is proposed to use the triangulation algorithm to update the depth of the map points in the local map after calculating the camera pose. The experiment uses fr1 series data set in tum database to test. The experimental results show that the scheme can reduce the root-mean-square error in the tracking process by 9.21% on average while ensuring real-time performance. The image quality of the point cloud map generated is also improved by 11.21%.

Keywords: RGB-D SLAM; ORB feature extraction; PnP algorithm; triangulation algorithm; local map

收稿日期: 2015-05-31

基金项目: 国家自然科学基金(61601149); 黑龙江省自然科学基金(QC2017074)。

作者简介: 董静薇(1972—), 女, 教授, 硕士研究生导师;

韩知雨(1994—), 男, 硕士研究生。

通信作者: 蒋燚铭(1994—), 女, 硕士研究生, E-mail: 2585098146@qq.com.

0 引言

随着人工智能的到来,近年来视觉同步定位与地图构建(Simulation Localization and Mapping, SLAM)技术在现实生活中被广泛应用,如扫地机器人,无人驾驶,VR虚拟现实等。视觉SLAM是指以搭载相机为传感器主体,在没有任何环境先验信息的情况下,在运动过程中建立环境模型,同时估计自己的运动。根据主传感器的不同,视觉SLAM主要分为三类,分别为Monocular SLAM(单目SLAM)^[1]、Stereo SLAM(双目SLAM)、以及RGB-D SLAM(深度相机SLAM)^[2],其中Monocular SLAM目前已实现的方案主要有MonoSLAM、PTAM(parallel tracking and mapping)等,Stereo SLAM主要的实现方案目前较少只有RTAB-MAP(real time appearance-based mapping)方法,而RGB-D SLAM的应用较广,主要的实现方案有DVO(dense visual odometry)、RTAB Map、ORB-SLAM2等。RGB-D SLAM是指把深度相机作为主传感器进行信息获取,深度相机造价低,且具有可以同时获取彩色图和深度图的优点,促使更多的学者把RGB-D相机数据作为研究视觉SLAM的传感器输入数据。

RGB-D SLAM最早由Henry等^[3]提出。该算法首先使用尺度不变特征转换(scale invariant feature transform, SIFT)提取图像特征点和描述子,后利用迭代最近点(iterative closest point, ICP)方法求解两帧间的位姿,最后利用图优化工具进行全局优化。随后Endres等^[4]于2012年对RGB-D SLAM算法进行评估,同时又使用输出的点云构建了栅格地图,方便了后续的路径规划。Engelhard等^[5]采用SURF(speed-up robust features)特征点提取算法代替SIFT算法进行特征提取,并采用随机采样一致性(random sample consensus, RANSAC)方法对相邻两帧的位姿进行估计。随后在2013年Sturm等^[6]发表DVO SLAM,它是一种基于稠密图像信息的视觉SLAM。在2014年Michaud等^[7]提出RTAB-Map,其使用SURF提取算法、利用回环检测,主要解决大环境的跟踪与建图问题。Murartal等^[8-9]于2016年提出的ORB-SLAM2方案是目前较完善的实现RGB-D SLAM的方案。该方案使用ORB(oriented robust brief)特征提取算法进行特征提取,大大提高了SLAM的实时性。近年来又有诸多关于RGB-D SLAM算法^[10-11]的论文出现,其中的改进大多在关

键帧选取方面,以及后端的回环检测方面。

经典的RGB-D SLAM实现方案,实时性差,很难保证同步跟踪定位以及建图。而近年来提出的RGB-D SLAM的实现方案往往前端的位姿计算结果不准确,优化效果不明显,使后端的回环检测中回环位置很难确定,后端运算难度大。本文研究目的是在保证实时性的同时,实现前端视觉里程计部分的改进,以减小算法在后端运行的压力。

本研究的实现方案为:在前端视觉里程计部分^[12-13],采用ORB特征提取算法,利用关键帧构建局部地图^[14],并采用PnP(perspective-n-point)算法对已经匹配成功的地图点和彩色图像进行位姿估计和优化^[15]。为保证当前帧与地图点特征匹配的准确度,和后续构建的局部地图的质量,本文尝试利用三角化算法,在已知位姿的情况下,计算地图点的深度值(空间点在世界坐标系的 z 值),计算出的结果再与从深度图中采集到的地图点的深度值进行加权数值融合,得到较精准的空间点的深度值结果,最后利用前端输出的地图点构建点云地图。为验证本文提出的RGB-D SLAM前端视觉里程计部分的有效性,使用TUM数据库中的fr1数据集进行实验。

1 ORB特征提取及PnP算法

本文算法流程如图1所示。当获取了深度相机生成的彩色图像后。利用ORB算法对彩色图像进行特征提取,并计算描述子。然后根据描述子与局部地图中的地图点进行匹配,随后采用PnP(perspective-n-point)算法计算相机位姿。最后再更新局部地图点,并且根据局部地图点生成点云地图。

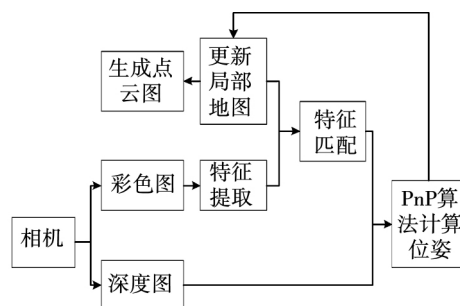


图1 本文算法流程图

Fig. 1 Algorithm flow chart

1.1 ORB特征提取和匹配

现主流的特征提取算法主要分为FAST关键点提取、SIFT关键点提取^[14]、以及ORB关键点提取。其中FAST算法仅提取出关键点而并没有描述子,

因此鲁棒性较低,而 SIFT 算法提取出的特征点虽然稳定性较强,精确度高,但却需要极大的计算量,普通电脑的 CPU 无法实现 SLAM 的实时跟踪建图。而 ORB 特征提取算法属于一种改进的 FAST 角点,但与 FAST 算法不同的是 ORB 算法计算了描述子以及特征点的主方向,其中特征点的主方向为 BRIEF 描述子增加了尺度不变性,所以本文选取的 ORB 算法是折中的特征提取算法,通过适当的降低精度和鲁棒性来提高算法的运行速度。其基本步骤是首先提取 FAST 角点,然后计算 BRIEF 描述子,及对前一步提取出的特征点进行周围图像区域的描述。

特征匹配解决了 SLAM 中数据关联的问题^[17],是计算相机位姿的基础数据,在特征匹配阶段误匹配经常出现,所以在匹配过程中要去除误匹配^[18],其匹配过程如下:

- 1) 采用 ORB 算法提取当前帧彩色图像的特征点,并计算其描述子。
- 2) 定义变量 p ,用于存储能投影在当前帧中的局部地图中的地图点。
- 3) 采用 OpenCV 中 ORB 算法进行特征匹配,参数分别为当前帧描述子以及地图点描述子。
- 4) 设置阈值 min_dis ,只有当描述子距离小于最小距离或小于最小距离的一定倍率时的匹配才算成功匹配,最终得到筛选过的控制距离在阈值内的匹配。

1.2 PnP 算法

PnP 是指在已知空间 3D 点坐标以及其 2D 投影像素坐标的情况下,来求解 3D 点到 2D 点之间运动的方法。采用 ORB 算法进行特征提取时,提取的特征点相对 SIFT 算法较少,而 PnP 算法的优点是两帧图像中仅需要 3 对匹配成功的特征点便可以求得较精准的位姿估计,所以本文选择 PnP 算法进行位姿估计。

本文的 RGB-D SLAM 把 PnP 问题构建成李代数上的非线性最小二乘问题迭代求解位姿。如图 2 所示,通过特征匹配, p_1, p_2 是空间点 P 的投影,由于相机位姿未知, P 在 I_2 上的投影 p_2 与实际位置 $\hat{p}_2$ 有一定误差 e ,所以调整相机位姿使得误差 e 最小,求得每个误差项关于优化变量的导数,其求解过程如下:

现有 n 个三维空间点 P 及其投影点 p_1, p_2 ,目标是求解相机位姿 R, T ,设空间点坐标 $P_i = [X_i, Y_i, Z_i]^T$ 其投影点像素坐标 $\mu_i = [u_i, v_i]^T$ 。根据单孔相

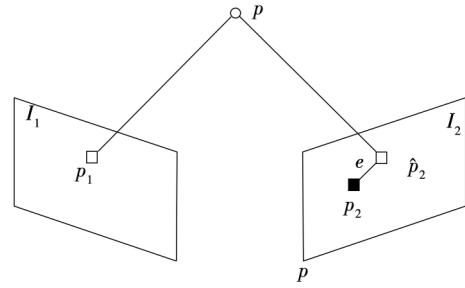


图2 重投影误差示意图

Fig. 2 Schematic diagram of re-projection error

机模型,像素位置与空间点位置的关系如下:

$$s_i \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = K \exp(\zeta^A) \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix} \quad (1)$$

式中: s_i 表示空间点 P_i 在相机坐标系下的 z 轴上的值; K 表示相机参数; ζ 表示相机位姿,式(1)可写成如下形式:

$$s_i u_i = K \exp(\zeta^A) P_i \quad (2)$$

现实情况下因为相机位姿未知以及观测点噪声的原因,所以该等式存在误差,因此我们把误差项求和,构造最小二乘问题,同时找到相机位姿的最优解,使得误差最小化,最小二乘公式如下:

$$\zeta^* = \argmin \frac{1}{2} \sum_{i=1}^n \left\| u_i - \frac{1}{s_i} K \exp(\zeta^A) P_i \right\|_2^2 \quad (3)$$

式(3)是指真实像素点的位置与空间 3D 点根据当前初始位姿投影到像素平面的像素点位置之间的误差。为求解式(3),使用李代数构建无约束的优化问题,并通过高斯牛顿优化算法进行求解,首先对该问题进行线性化:

$$e(X + \Delta x) \approx e(X) + J \Delta x \quad (4)$$

式中: e 为像素坐标误差(2 维); X 为相机位姿(6 维); J 是一个 2×6 的雅克比矩阵,描述了相机误差对相机位姿的一阶变换关系,用李代数的扰动模型进行求解 J :

$$P' = (\exp(\zeta^A) P)_{1:3} = [X', Y', Z']^T \quad (5)$$

式(5)中 P' 表示空间点 P_i 变换到相机坐标系下的坐标,上式表示取该坐标的前 3 维。根据相机针孔模型有:

$$su = KP' \\ \begin{bmatrix} su \\ sv \\ s \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} \quad (6)$$

消去 s 得:

$$u = f_x \frac{X'}{Z'} + c_x, v = f_y \frac{Y'}{Z'} + c_y \quad (7)$$

左乘扰动量并求导得:

$$\frac{\partial e}{\partial \delta \zeta} = \lim_{\delta \zeta \rightarrow 0} \frac{e(\delta \zeta \oplus \zeta)}{\delta \zeta} = \frac{\partial e}{\partial P'} \frac{\partial P'}{\partial \delta \zeta} \quad (8)$$

式(8)中 $\oplus$ 指李代数上的左乘扰动,等式右端第一项是指误差项关于投影点的导数:

$$\frac{\partial e}{\partial P'} = - \begin{bmatrix} \frac{\partial u}{\partial X'} & \frac{\partial u}{\partial Y'} & \frac{\partial u}{\partial Z'} \\ \frac{\partial v}{\partial X'} & \frac{\partial v}{\partial Y'} & \frac{\partial v}{\partial Z'} \end{bmatrix} = - \begin{bmatrix} \frac{f_x}{Z'} & 0 & -\frac{f_x X'}{Z'^2} \\ 0 & \frac{f_y}{Z'} & -\frac{f_y Y'}{Z'^2} \end{bmatrix} \quad (9)$$

式(8)等式右端第二项是指变换成空间坐标后对李代数的导数:

$$\frac{\partial (TP)}{\partial \delta \zeta} = \begin{bmatrix} I & -P'^A \\ 0^T & 0^T \end{bmatrix} \quad (10)$$

取前3维得:

$$\frac{\partial P'}{\partial \delta \zeta} = [I \quad -P'^A] \quad (11)$$

将两项相乘得到了 2×6 的雅克比矩阵:

$$\frac{\partial e}{\partial \delta \zeta} = - \begin{bmatrix} \frac{f_x}{Z'} & 0 & -\frac{f_x X'}{Z'^2} & -\frac{f_x X' Y'}{Z'^2} & f_x + \frac{f_x X'^2}{Z'^2} & -\frac{f_x Y'}{Z'} \\ 0 & \frac{f_y}{Z'} & \frac{f_y Y'}{Z'^2} & -\frac{f_y Y'^2}{Z'^2} & \frac{f_y X' Y'}{Z'^2} & \frac{f_y X'}{Z'} \end{bmatrix} \quad (12)$$

至此推导出了观测相机方程关于相机位姿的导数矩阵,为优化过程提供梯度的方向,指导优化迭代。利用高斯牛顿法优化出精度更高的相机位姿。

2 改进算法

本文采用 Monocular SLAM 中常用的构建局部地图的方法。使当前帧只与局部地图中的地图点进行匹配,并计算位姿,这样可以大大减小由于相邻帧之间位姿误差的积累而产生的漂移现象。而后为减少从深度图中采集到的地图点对应的空间点深度值的误差,本文提出采用三角化算法来计算出空间点的深度值,并与采集到的空间点深度值进行加权数值融合,争取达到提高构建的局部地图质量的目的。

局部地图仅包含与当前帧有相同特征点的地图点和关键帧,所以计算量小,不会大幅度增加算法的运行时间,但是却提高了算法的精确度,同时也为后

端的回环检测提供了更加精准的地图点深度的数据。

2.1 构建局部地图

构建局部地图是指向地图中插入关键帧以及地图点等信息^[9],局部地图描述了附近的特征点信息,只保留距离相机当前位置较近的特征点,把远的视野外的特征点信息删掉。如图3显示了构建局部地图的具体过程。

在正确的计算 PnP 相机位姿后,要经过地图点的更新,局部地图构建过程如下:

1) 从第一帧彩色图提取其 ORB 特征点,并根据深度图计算对应特征点深度,并把第一帧当做关键帧,取全部的特征点为地图点,加入到局部地图中。

2) 正常工作模式下,利用 PnP 求解位姿后,更新地图点,把彩色图中没有匹配上的特征点求取其深度信息,添加到地图点中,把地图点中匹配成功率低的点删掉。

3) 关键帧检测,设置阈值,当 PnP 求取的位姿结果大于阈值时,则把当前帧添加进关键帧,其上所有特征点取其深度以及描述子加入到局部地图中。这样完成了整个局部地图的构建的过程。

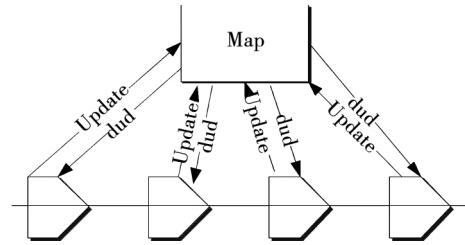


图3 局部地图构建

Fig. 3 Local map construction

2.2 三角化局部地图点深度

在采用了 PnP 算法估计出了相机位姿的基础上,对 RGB-D SLAM 算法的视觉里程计前端提出进一步的改进,通过估计出的位姿对局部地图中地图点的深度进行进一步优化,这样可以提高下一帧匹配的准确度以及减小重投影的误差^[20-21]。

三角测量是指通过在两处同时观察到同一个点的夹角,从而确定该点的距离,如图4所示。

左右两图分别为图像 I_1, I_2 以左图为参考帧,右图的位姿为 T, O_1, O_2 分别为两个位置的相机光心,点 P 为空间点, p_1, p_2 分别为空间点在两帧图像中对应的特征点,由图中可以看到由于噪声原因,直线 O_1, p_1 与直线 O_2, p_2 并没有交于点 P ,按照对极几何定义:

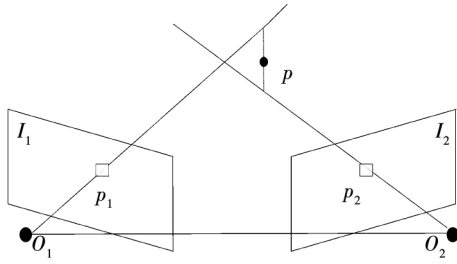


图4 三角化获得地图点深度

Fig. 4 Triangulation to obtain map points depth

$$s_1 x_1 = s_2 R x_2 + t \quad (13)$$

式中: s_1, s_2 分别为特征点的深度; x_1, x_2 为特征点的归一化坐标; R, t 为相机位姿, 上式左乘 x_1^A 得:

$$s_1 x_1^A x_1 = 0 = s_2 x_1^A R x_2 + x_1^A t \quad (14)$$

因为估计的相机位姿存在误差, 现构建最小二乘问题求解 s_2 , 这里的优化变量为 s_2 , 即求得当 s_2 为何值时, 等式最小, 最后利用式 s_2 求得 s_1 。

$$\theta = \operatorname{argmin} \frac{1}{2} \sum_{i=0}^n \| s_2 x_1^A R x_2 + x_1^A t \|^2 \quad (15)$$

由于是前端的位姿估计, 此处并没有选择计算量较大的深度滤波进行深度估计, 数据融合算法采取普通的加权数据融合算法。将上一步求得的深度数据 s_1 与从深度图中采集得到的数据融合:

$$s_1 = x + v_1 \quad (16)$$

$$s_0 = x + v_0 \quad (17)$$

式中 s 表示深度数据; v 表示两深度估计值存在的随机误差, 且设 $v_i (i=0, 1)$ 为存在的误差, $v_i \sim N(0, \sigma_i^2)$, 且两估计值分别独立。

设 x 的估计值 $\hat{x}$ 与观测值 s 成线性关系, $\hat{x}$ 为 x 的无偏估计则:

$$\hat{x} = w_1 s_0 + w_2 s_1 \quad (18)$$

式中 $\Omega = (w_1, w_2)$ 为两个估计值的权值。设估计误差 $\tilde{x} = x - \hat{x}$ 取代价函数为 $\tilde{x}$ 的均方误差, 则:

$$D = E(\tilde{x}^2) = E\{ [x - w_1(x + s_0) - w_2(x + s_1)]^2 \} \quad (19)$$

$$E(\tilde{x}) = E[x - w_1(x + s_0) - w_2(x + s_1)] = 0 \quad (20)$$

式中 $E(v_1) = E(v_2) = 0, E(x) = E(\hat{x})$, 则有:

$$w_2 = 1 - w_1 \quad (21)$$

则代价函数可写为

$$D = E(\tilde{x}^2) = E[w_1^2 v_1^2 + (1 - w_1)^2 v_2^2 + 2w_1(1 - w_1)v_1 v_2] \quad (22)$$

$$D = E(\tilde{x}^2) = w_1^2 \sigma_1^2 + (1 - w_1)^2 \sigma_2^2 \quad (23)$$

为使 J 最小, 求导有

$$\frac{\partial D}{\partial \Omega} = 0 \quad (24)$$

最优权值为

$$w_1^* = \frac{\sigma_2^2}{\sigma_2^2 + \sigma_1^2}, w_2^* = \frac{\sigma_1^2}{\sigma_2^2 + \sigma_1^2} \quad (25)$$

则最优估计量为

$$\hat{x} = \frac{\sigma_2^2 s_0}{\sigma_2^2 + \sigma_1^2} + \frac{\sigma_1^2 s_1}{\sigma_2^2 + \sigma_1^2} \quad (26)$$

3 仿真实验结果与分析

为验证算法的有效性, 本章选取 TUM 数据集中的 fr1 系列数据集进行测试, 首先进行特征匹配实验, 然后通过定位误差与运行速度实验和点云建图实验, 对传统算法和本文算法进行评估, 最后得出实验结论。

本算法的实验平台为 Intel i5 6200U 处理器 (2.3 GHz), 4 GB RAM, 64 位 Ubuntu14.04 操作系统。

3.1 ORB 特征匹配实验结果

特征匹配实验选取 fr1/xyz 中的部分数据进行实验。ORB 特征匹配结果如图 5、6 所示, 其中图 5 为直接匹配的结果, 图 6 为去除误匹配之后的匹配效果。图 6 的特征匹配相比于图 5 减小了 67%, 也就是正确匹配率为 33%。这里设置的筛选依据是, 汉明距离小于最小距离的 2 倍。



图5 直接匹配

Fig. 5 Direct matching



图6 去除误匹配

Fig. 6 Removes mismatches

3.2 定位误差与运行速度分析

本次选取 fr1/room 数据集以及 fr1/xyz 数据集进行测试, 进行 5 次重复试验, 在实验过程中记录算法的运行时间以便对算法的实时性进行评估,

表1、2为本文算法与其他两种相关算法的跟踪精度和运行时间的对比,均方根误差体现了数据的真实轨迹和算法跟踪的轨迹之间的误差。

表1 fr1/room 定位误差与运行速度对比

Tab. 1 fr1/room positioning error and running speed comparison

算法	均方根误差/m	运行速度/s
SIFT + ICP	0.219	0.318
ORB + PnP	0.227	0.018
ORB + PnP + tri (本文算法)	0.209	0.020

表2 fr1/xyz 定位误差与运行速度对比

Tab. 2 fr1/xyz positioning error and running speed comparison

算法	均方根误差/m	运行速度/s
SIFT + ICP	0.209	0.457
ORB + PnP	0.250	0.023
ORB + PnP + tri (本文算法)	0.223	0.026

从表1、2可以看出:

1) 针对前两行,SIFT算法与ORB算法的比较,SIFT算法的均方根误差比ORB算法平均降低了9.962%,而对运算速度而言,ORB算法却较SIFT算法提升了90%,相较于均方根误差的变化,运算速上的差异则更加明显,由此证明本文采用的ORB特征提取算法极大提高了算法的实时性。

2) 针对后两行,同样使用ORB特征提取的两个算法,本文算法相较于ORB + PnP算法,均方根误差平均降低了9.21%,而算法的运行时间却仅增加了0.002 s,对于fr1这样每秒取30帧的数据集,该算法的处理帧速度可满足SLAM算法的实时同步跟踪定位的要求。

综上所述,在满足处理帧速度的条件下,本文算法减小了均方根误差,提高了跟踪的精度。

3.3 局部地图构建实验结果

同步定位与地图构建技术的建图方式有很多种,其中主流的建图方式有点云建图、八叉树建图、网格建图。点云建图是指用一组离散的点来表示的地图,其中每个点都包含 x, y, z 三维坐标信息。相比于点云建图,八叉树建图法以及网格建图法作为底层技术更加方便向上层应用来提供信息。但是建图结果不清晰,导致实验对比困难,所以为验证本文算法的有效性,本文采取最基本的点云建图方式进

行实验对比分析。

本文的点云建图实验选取fr1系列的fr1/desk数据集以及fr1/floor数据集中的全部数据进行点云建图,建图结果如图7~10所示。图7、8为采用fr1/desk数据集的建图结果,图9、10为采用fr1/floor数据集的建图结果。其中图7、9为前端没有三角测量的RGB-D SLAM算法生成的点云图,图8、10则为使用本文算法最后生成的点云图。获得的数据结果如表3、4所示。

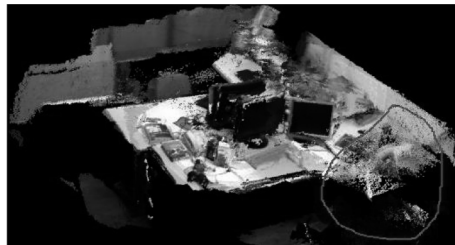


图7 ORB + PnP 算法实现的效果图

Fig. 7 ORB + PnP algorithm



图8 本文算法实现的效果图

Fig. 8 The algorithm of improved



图9 ORB + PnP 算法实现的效果图

Fig. 9 ORB + PnP algorithm

如图7~10所示,图8相较于图7图像轮廓更加清晰,图像质量更好。而图10相较于图9圆圈处建图范围更广(如图中红线处圈出的地板所示)。



图10 ORB + PnP 算法实现的效果图

Fig. 10 ORB + PnP algorithm

下面采用图像质量评判的方法对以上三组对比图进行定量分析。本文采取的图像质量评判标准分别为信息熵,亮度均值以及标准差。

1) 图像信息熵通过统计每个像素点灰度值的总期望得出,信息熵值大说明该图像的轮廓更加清楚。

2) 亮度均值是衡量一个图像中像素的灰度值的平均数,亮度均值大的图像说明该图像亮度越大,点云建图结果更加符合真实场景。

3) 图像标准差反应一个图像高频部分的大小,标准差大的图像对比明显,图像更加清晰。

以上两组对比图像的像素分别为 522×300 , 440×390 。量化分析的实验结果如表3、4所示:

表3 fr1/desk 图像数据统计分析

Tab. 3 Statistical analysis of fr1/desk image data

	信息熵	亮度均值	标准差
图7 ORB + PnP	5.07	50.48	29.34
图8 本文算法	5.15	50.48	29.34

表4 fr1/floor 图像数据统计分析

Tab. 4 Statistical analysis of fr1/floor image data

	信息熵	亮度均值	标准差
图9 ORB + PnP	4.67	50.48	29.34
图10 本文算法	4.98	50.49	29.35

从表3、4可以看出,对于 fr1/desk、fr1/floor 数据集,本文算法相较于 ORB + PnP 算法生成点的点云图,虽然亮度均值及标准差相差无几,但是信息熵

都有明显提高,分别提高了 15.78%、6.64%。可见图8、10的建图结果更加清晰,图像信息含量更高。

综上所述,两种数据集的测试结果都表明,使用本文算法构建的点云地图更加清晰,本文算法确实通过三角化地图点深度提高了构建的点云地图的质量。

4 结 论

本文主要把传统的 RGB-D SLAM 算法的前端视觉里程计部分进行改进,在保证算法处理帧速度的同时,在前端视觉里程计部分加入了三角测量算法来提高定位的稳定性。最后采用 TUM 数据集集中的 fr1 系列数据集进行了均方根误差,运行速度以及生成点云图的实验。实验结果证明本方案在定位精度上有明显提高,其中均方根误差平均减少 9.21%。改进后算法生成的点云图相较于传统算法生成的点云图信息熵提高了 11.21%,可以为后端的闭环检测提供良好的基础数据。

fr1 数据集使用的 Kinect 相机的运行速度是 0.41 m/s,旋转速度是 23.33 deg/s,运行相对较慢。下一步的研究是如何在运行速度较快时实现准确的定位与建图。

参 考 文 献:

- [1] LEES H, CIVERA J. Loosely-coupled Semi-direct Monocular SLAM [J]. IEEE Robotices & Automation Letters, 2018, 4 (2): 399.
- [2] KERL C, STURM J, CREMERS D. Dense Visual SLAM for Rgb-d Cameras [C] // IEEE/RSJ International Conference on Intelligent Robots and Systems. USA Piscataway: IEEE, 2014: 2100.
- [3] HENRY P, KRAININ M, HERBST E, et al. Rgb-d mapping: Using Depth Cameras for Dense 3D Modeling of Indoor Environments [C] // The 12th International Symposium on Experimental Robotics. 2010: 22.
- [4] ENDRESS F, HESS J, STURM J, et al. 3-D Mapping with an Rgb-d Camera [J]. IEEE Transactions on Robotics, 2014, 30 (1): 177.
- [5] ENGELHARD N, ENDRESS F, HESS J, et al. Real-time 3D Visual SLAM with a Hand-held Rgb-d Camera [C] // Proceedings of the RGB-D Workshop on 3D Perception in Robotics at the European Robotics Forum, Sweden Vasteras, 2011: 1.
- [6] KERL C, STURM J, CREMERS D. Dense Visual SLAM for Rgb-d Cameras [C] // In 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2013: 2100.
- [7] LABBE M, MICHAUD F. Online Global Loop Closure Detection for Large-scale Multi-session Graph-based SLAM [C] // In 2014

- IEEE /RSJ International Conference on Intelligent Robots and Systems, 2014: 2661.
- [8] MURARTAL R, TARDOS J D. Orb-SLAM: A Versatile and Accurate Monocular SLAM System [C] //IEEE Transactions on Robotics. IEEE, 2015: 1147.
- [9] MURARTAL R, TARDOS J D. Orb-SLAM2: an Open-source SLAM System for Monocular, Stereo, and Rgb-d Cameras [J]. IEEE Transactions on Robotics, 2017, 35(5): 1255.
- [10] 陈邵, 郭宇翔, 高天啸, 等. 移动机器人 RGB-D 视觉 SLAM 算法 [J]. 农业机械学报, 2018, 49(10): 38.
CHEN Shao, GUO Yuxiang, GAO Tianxiao, et al. Rgb-d Visual SLAM Algorithm for Mobile Robots [J]. Transactions of the Chinese Society for Agricultural Machinery, 2018, 49(10): 38.
- [11] 李戈星, 刘士荣, 仲朝亮, 等. 基于改进关键帧选择的 RGB-D SLAM 算法 [J]. 大连理工大学学报, 2017, 57(4): 411.
LI Yixing, LIU Shirong, ZHONG Chaoliang, et al. Rgb-d SLAM Algorithm Based on Improved Key-frame Selection [J]. Journal of Dalian University of Technology, 2017, 57(4): 411.
- [12] HENRY P, KRAININ M, HERBST E, et al. Rgb-d Mapping: Using Kinect-style Depth Cameras for Dense 3D Modeling of Indoor Environments [J]. International Journal of Robotics Research, 2012, 31(5): 647.
- [13] 丁文东, 徐德, 刘希龙, 等. 移动机器人视觉里程计综述 [J]. 自动化学报, 2018, 44(3): 385.
DING Wendong, XU De, LIU Xilong, et al. Review on Visual Odometry for Mobile Robots [J]. Acta Automatica Sinica, 2018, 44(3): 385.
- [14] 龚学锐, 闵华松. 基于 ROS 的惯性和视觉里程计的机器人室内定位 [J]. 组合机床与自动化加工技术, 2019, 5(5): 106.
GONG Xuerui, MIN Huasong. Indoor Positioning by Visual-odometry Based on Ros [J]. Modular Machine Tool & Automatic Manufacturing Technique, 2019, 5(5): 106.
- [15] 张震, 郑宏. 一种结合 ORB 特征和视觉词典的 RGB-D SLAM 算法 [J]. 计算机工程与应用, 2018, 54(12): 170.
ZHANG Zhen, ZHENG Hong. Rgb-d SLAM Algorithm based on Orb Features and Visual Dictionary [J], Computer Engineering and Applications, 2018, 54(12): 170.
- [16] LEPETIT V, MORENO F, FUA P, et al. Epnp: an Accurate and Efficient Solution to the PnP Problem [J]. International Journal of Computer Vision, 2008, 81(2): 155.
- [17] RUBLEE E, RABAU D V, KONOLIGE K, et al. Orb: an Efficient Alternative to Sift or Surf [C] //International Conference on Computer Vision, 2011: 2564.
- [18] 董静薇, 赵春丽, 海博. 融合同态滤波和小波变换的图像去雾算法研究 [J]. 哈尔滨理工大学学报, 2019, 24(1): 66.
DONG Jingwei, ZHAO Chunli, HAI Bo. Research on Image Defog Algorithm based on Fusion Homomorphic Filtering and Wavelet Transform. Journal of Harbin University of Science and Technology, 2019, 24(1): 66.
- [19] 刘志斌, 吴显亮, 徐文丽, 等. 视觉 SLAM 中的基于误匹配风险预测的特征选择 [J]. 机器人, 2010, 32(5): 635.
LIU Zhibin, WU Xianliang, XU Wenli, et al. Erroneous Matching Risk Prediction based Feature Selection for Visual SLAM [J]. Robot, 2010, 32(5): 635.
- [20] PRAKHIA SM, LIU B, LIN W, et al. Sparse Depth Odometry: 3D Keypoint based Pose Estimation from Dense Depth Data [C] // IEEE International Conference on Robotics and Automation, 2015: 4216.
- [21] 彭清漪, 夏林元, 吴东金. 基于光度与点线特征融合的半直接单目视觉定位算法 [J]. 传感器与微系统, 2020, 39(4): 110.
PENG Qingyi, XIA Linyuan, WU Dongjin. Semi-direct Monocular Visual Localization Algorithm based on Photometry and Point/line Feature Fusion [J]. Transducer and Microsystem Technologies, 2020, 39(4): 110.
- [22] 方正, 赵世博, 李昊来. 一种融合稀疏几何特征与深度流的深度视觉 SLAM 算法 [J]. 机器人, 2019, 41(2): 185.
FANG Zheng, ZHAO Shibao, LI Haolai. A Depth Vision SLAM Algorithm Combining Sparse Geometric Features with Range Flow [J]. Robot, 2019, 41(2): 185.

(编辑: 温泽宇)