

Turtlebot机器人自主导航及避障实验设计研究

黎凯龙, 涂海燕

四川大学电气工程学院自动化系, 四川 成都
Email: 857012848@qq.com, haiyantu@163.com

收稿日期: 2021年7月2日; 录用日期: 2021年7月23日; 发布日期: 2021年8月4日

摘 要

随着机器人领域研究的深入, 设计一款能够自主移动的家庭服务机器人具有重要的现实意义。本文基于自主导航和避障功能的实现, 设计了一款基于ROS平台的Turtlebot机器人。为了实现自主导航和避障功能, 采用Dijkstra算法进行全局路径规划, 采用动态窗口法进行局部路径规划。通过搭载深度相机Astra和激光雷达EAI F4两款传感器, 进行SLAM和自主导航功能效果的比较。实验过程中使用Gmapping算法来实现SLAM建图。此外, 通过编写脚本, 原本需要人工控制的SLAM建图过程也被改进成了无需人工控制的自主SLAM。实验过程也完成了这款机器人自主导航和避障的功能。

关键词

自主导航, Turtlebot机器人, SLAM, 路径规划

Experimental Design of Autonomous Navigation and Obstacle Avoidance for Turtlebot Robot

Kailong Li, Haiyan Tu

Department of Automation, College of Electrical Engineering, Sichuan University, Chengdu Sichuan
Email: 857012848@qq.com, haiyantu@163.com

Received: Jul. 2nd, 2021; accepted: Jul. 23rd, 2021; published: Aug. 4th, 2021

Abstract

With the research of robotics advancing, it is important to design a home service robot that can move autonomously. The Turtlebot robot based on ROS platform is used to achieve the autonomous navigation and obstacle avoidance functions. In order to realize the autonomous navigation

function, the Dijkstra algorithm is used for global path planning, and the dynamic window algorithm is used for local path planning. A comparison between the results of SLAM and autonomous navigation is carried out by using two sensors, the depth camera Astra and the lidar EAI F4. During the experiment, the Gmapping algorithm was used to implement SLAM mapping. In addition, by writing scripts, the SLAM mapping process, which originally required manual control, has also been improved to autonomous SLAM without manual control, making it more convenient in practical applications. The results of simulation and experiments show that autonomous navigation and obstacle avoidance functions have been implemented.

Keywords

Autonomous Navigation, Turtlebot Robot, SLAM, Path Planning

Copyright © 2021 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

近年来,随着科技的进步,智能化、小型化已成为机器人发展的方向和目标之一。机器人不再是工厂所独有的设备,他们逐渐步入了普通人的生活。与此同时,自主移动机器人的工作环境也变得越来越复杂和多样,例如分发快递、灾难救援以及家庭服务等。这些环境中都存在着多样化的障碍物,这就要求自主移动机器人对环境建模描述的精确性和一致性要更高[1]。为实现这一目标,自主机器人需要具备以下两个能力[2]:第一是 SLAM (simultaneous localization and mapping),也就是在未知环境中的实时定位和地图构建;第二便是路径规划的能力。

SLAM 指自主机器人在未知的环境中自主完成地图构建,并根据该地图进行自身定位的能力[1]。其主要思想是将机器人单独放置在一个未知环境内,从某一起始点开始移动,通过机器人自身搭载的传感器和里程计等设备,让它在移动过程当中建立周围环境的地图,实现机器人定位[3]。

上世纪八十年代末,Cheeseman 和 Crowle 等人运用概率学原理解决了 SLAM 研究中有关设备精度和算法效率的相关问题。由此发展出了 SLAM 的第一个经典的分类,基于卡尔曼滤波器的 KF-SLAM,以及后续衍生出的 EKF-SLAM (扩展卡尔曼滤波)和 UKF-SLAM (无迹卡尔曼滤波)等各种算法[1]。进入 21 世纪后,SLAM 的研究也诞生了新的分支——基于机器视觉的视觉 SLAM,即 VSLAM。VSLAM 通常以相机作为传感器,主要有单目 SLAM、双目 SLAM、以及 RGB-D SLAM 三种类型[4]。

目前常见的 SLAM 算法主要有以下几种:Gmapping, Hector SLAM, Cartographer, RGBD SLAM 和 ORB SLAM。Gmapping 由于计算能力的限制,主要用来进行室内小场景的建图。Hector SLAM 可以在崎岖的环境中建图,也可以用于手持设备或无人机。Cartographer 由谷歌发布,主要用于基于激光雷达的二维或三维建图,RGBD SLAM 专门用来 3D 建图,ORB SLAM 适用于获取一组连续的图像,通常用在汽车上。

自主路径规划旨在没有人为因素干扰的情况下,为机器人提供一条从起点到目标位置的最佳无碰撞路径。路径规划主要分为全局路径规划和局部路径规划[1]。全局路径规划是通过先前由机器人储备的环境信息规划出一条从起点到终点的最佳无碰撞路径,它的主要方法有 Dijkstra 算法和 A*算法[3]。局部路径规划则是机器人通过传感器获取当前时刻的环境信息,来达到躲避动态的障碍物的目的,其主要方法有动态窗口法(DWA)、矢量场直方图算法(VFH)和人工势场法[3]。现今还有 Koenig 等人提出的增量式的 D*Lite 算法,以及杜克大学的 Ferrari、北卡州立大学的 Deshpande 等人基于无线传感器的路径规划研究,

都取得了重要的进展[5]。

总之, 机器人是当今最热门的研究领域之一, 而服务型机器人更是热门中的热门。因此, 设计一款家庭服务机器人具有十分重要的意义和价值。而要想实现这个目标, 最关键的两个问题就是 SLAM 和路径规划。本文因此基于已有的研究, 提出一种基于 ROS 平台的 Turtlebot 机器人自主导航和避障算法, 并在实验室予以实现。

2. SLAM 与自主导航的设计

2.1. 设计思路

基于家庭服务的自主移动机器人的设计目的, 本文首先通过搭载深度相机 Astra 和激光雷达 EAI F4 两款传感器, 进行 SLAM 和自主导航的效果的比较, 从而得出两种方法各自的优缺点。在 SLAM 建图方面, 本文重点关注两款传感器的建图速度、准确性和建图的成功率, 而 SLAM 建图采用 Gmapping 算法。在自主导航方面, 重点关注机器人的路径规划能力以及避障能力, 分别用 Dijkstra 算法和动态窗口算法(Dynamic Window Approaches, DWA)实现全局和局部路径规划。其中避障能力又分为静态避障能力和动态避障能力, 并通过实验进行测试, 同时还考虑机器人的安全性。毕竟这是一款家用服务机器人, 需要考虑到家庭环境的复杂程度和多样化, 以及行动迟缓的老人和精力旺盛的小孩。遇到突发情况时, 机器人应当停止运动, 观察当前的环境信息, 以免造成伤害。除此之外, 通过编写脚本, 设置室内环境的关键点, 使原本需要人工控制的 SLAM 过程也将被改进成无需人工控制的自主 SLAM。

2.2. 用于 SLAM 建图的 Gmapping 算法

本文选取 Gmapping 算法来实现 SLAM 建图。Gmapping 算法基于 Rao-Blackwellized 粒子滤波算法, 将定位和建图过程分离, 先进行定位, 然后再进行建图。Gmapping 算法对激光雷达的频率要求比较低。ROS 中的 Gmapping 功能包订阅了机器人的深度信息、IMU 信息和里程计信息, 在完成了必要的参数配置后, 即可创建并输出基于概率的二维栅格地图, 具体有以下四个步骤[6]。

1) 采样: 当前时刻 t 的粒子集合 $\{x_t^{(i)}\}$ 是从上一时刻 $t-1$ 粒子集合 $\{x_{t-1}^{(i)}\}$ 中采样得到的。

2) 计算粒子权重: 每个粒子的权重为 $\{w_t^{(i)}\}$:
$$w_t^{(i)} = \frac{p(x_{t|t}^{(i)} | z_{1:t}, u_{1:t-1})}{\pi(x_{t|t}^{(i)} | z_{1:t}, u_{1:t-1})} \quad (i = 1, 2, \dots, N)$$

3) 重采样: 重采样是在粒子滤波时, 对于粒子的退化现象提出的解决办法。当有效粒子的数量低于设定的阈值时, 便进行重采样操作。经过重采样操作之后, 所有的粒子拥有一样的权重。

4) 更新地图: 根据机器人的轨迹和传感器的观测数据, 计算地图概率更新地图。

2.3. 自主导航功能

自主导航的关键是机器人的路径规划和定位。前者负责机器人运动线路的设计, 后者用来帮助机器人确定自身的位置, 二者相结合, 就可以实现完整的自主导航功能。路径规划又分为全局路径规划和局部路径规划。前者利用 SLAM 建图得到的全局地图的信息, 后者则依赖于机器人上搭载的传感器提供的实时信息, 以便对环境中事物的变动做出相应的调整。

1) 本文中全局路径规划采用一种典型的广度优先搜索算法——Dijkstra 算法[7]。已知一个带权重的有向图 G , V 是 G 中所有顶点的集合, S 是顶点之一。 E 是顶点之间带权边的集合, (u, v) 表示从顶点 u 到顶点 v 的路径, 则该算法的主要步骤如下:

① 初始状态, 集合 S 仅包含源点 v , 集合 U 包含除了 v 以外的其他所有顶点。如果 v 与集合 U 中的

顶点 u 有边, 则 v 到 u 的距离是边 (v,u) 的权值; 如果没有边则边 (u,v) 的权值为 ∞ 。

② 从集合 U 中选取一个距离 v 最小的顶点 k , 把 k 加入到集合 S 中。

③ 以 k 为中间点, 修改 U 中每个顶点的距离。如果从 v 到 k 再到 u 的距离小于从 v 到 u 的距离, 则修改顶点 u 的距离。那么, 修改后的距离就变为 k 的距离加上边 (k,u) 的权值。

④ 重复步骤②和③, 直到 S 中包含所有的顶点。

2) 局部路径规划则采用了动态窗口法算法 DWA (Dynamic Window Approaches)。其核心思想是创建一个由直线速度 v 和旋转速度 ω 组成的速度对 (v,ω) 的速度矢量空间, 将路径规划问题转化成速度矢量空间上的约束优化问题。对于地面机器人, 速度矢量空间 V_r 由速度矢量集合 V_s 、可允许速度矢量 V_a 和速度动态窗口 V_d 的交集构成[8]: $V_r = V_s \cap V_a \cap V_d$ 。其中可允许速度矢量 V_a 表示在当前位置下机器人的加速度不会导致碰撞[8]:

$$V_a = \left\{ (v, \omega) \mid v \leq \sqrt{2 \text{dis} \tan t(v, \omega) \cdot a_v} \wedge \omega \leq \sqrt{2 \text{dis} \tan t(v, \omega) \cdot a_\omega} \right\}$$

式中 $\text{dis} \tan t(v, \omega)$ 表示机器人障碍物的最小距离, a_v 和 a_ω 分别是机器人的平移加速度和旋转加速度。

从速度矢量空间 V_r 选择机器人在 $k+1$ 时刻的最优运动速度矢量, 需要依靠目标函数[8]:

$$G(v, \omega) = \sigma(\alpha \cdot \text{heading}(v, \omega) + \beta \cdot \text{dist}(v, \omega) + \gamma \cdot \text{velocity}(v, \omega))$$

其中 $\text{heading}(v, \omega)$ 是方位角评价函数, 表示当前速度下, 机器人运动方向和目标方向的一致程度。

$\text{dist}(v, \omega)$ 用来评价机器人当前的运动轨迹和障碍物的最近距离。 $\text{velocity}(v, \omega)$ 是速度评价函数, 表示当前时刻机器人的速度大小。 α 、 β 、 γ 为三个评价函数的权重比, 可根据场合的不同调整他们的大小, 以此来调整机器人的动作。

3) 定位。机器人的定位则是机器人通过传感器获取当前自身周围的环境信息, 从而推算出自己目前所处的位置。ROS 中的 `amcl` 功能包为用户提供了一种基于概率统计的自适应的蒙特卡洛定位方法, 使机器人可以随时随地地计算出自己在地图中所处的位置。自适应蒙特卡洛定位是在对粒子进行重采样的过程中引入自适应机制, 即设定重采样粒子数量的阈值, 并判断当粒子数小于阈值时, 执行重采样操作, 否则, 跳过重采样过程, 直接输出定位结果, 再循环执行预测过程与更新过程。

3. 实验过程及分析

3.1. 各部件的调试

1) Turtlebot 机器人的调试: Turtlebot 指底盘 + kinect + 托盘 + 支架的合体(如图 1 所示)。跟 Turtlebot 相连的电脑称为上网本, 远程到上网本的称为工作站或工作机。调试时在上网本打开新的终端输入 `roscore`, 如果 ROS 安装则正确显示 `started core service [/rosout]`。开启 Turtlebot 开关, 连接上网本。输入检测命令 `ls/dev/kobuki`, 如果显示对应的设备 `/dev/kobuki`, 则说明设备已成功连接, 否则就需要添加新的设备名称。在工作机打开新的终端, 启动 Turtlebot 键盘控制。如果能通过键盘使 Turtlebot 正常运动, 则证明 Turtlebot 功能完善。

2) 深度相机 Orbbec Astra [9]的调试: 首先, 安装依赖, 再从网站下载对应版本的驱动并进行安装。然后安装 `astra_camera` 和 `astra_launch`, 最后更新环境变量配置。将 Astra 连接到电脑上, 运行 `astra_launch` 文件, 若没有发生错误则说明深度相机正常工作, 使用 `rqt_image_view` 命令打开 `ImageView` 即可查看图像。在 `ImageView` 中选择对应话题, 即可显示对应类型的图像。如图 2 所示, 此时显示的就是和普通相机一样的彩色图像。如图 3 所示, 此时显示的就是深度相机所独有的深度图像。

3) 激光雷达 EAI F4 的调试: 激光雷达 EAI F4 的调试过程与深度相机类似, 需要先下载相应的驱动

并进行安装。安装完成后, 将激光雷达连接到电脑, 打开可视化工具 rviz 即可查看通过激光雷达得到的点云图(如图 4 所示)。



Figure 1. Turtlebot robot
图 1. Turtlebot 机器人

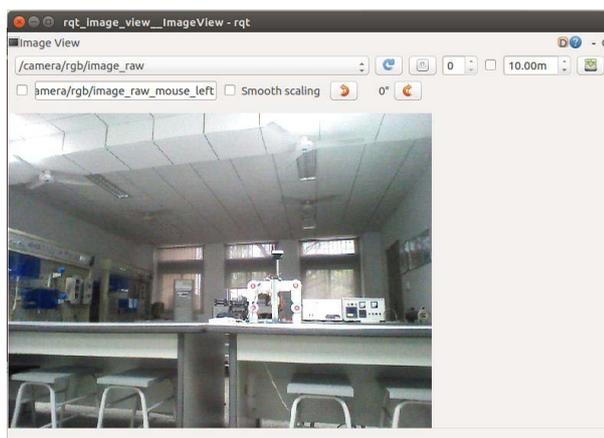


Figure 2. RGB image of depth camera Astra
图 2. 深度相机 Astra 的彩色图像

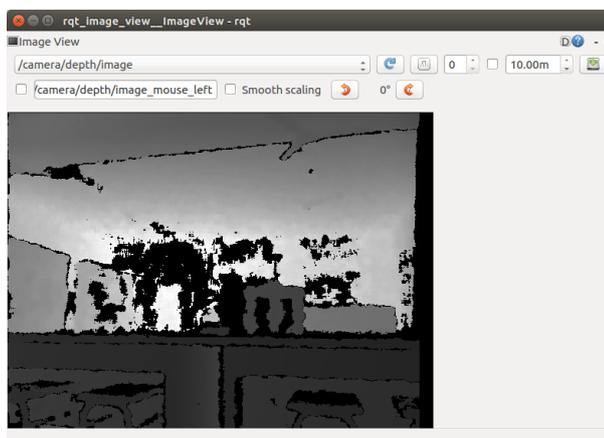


Figure 3. Depth image of depth camera Astra
图 3. 深度相机 Astra 的深度图像

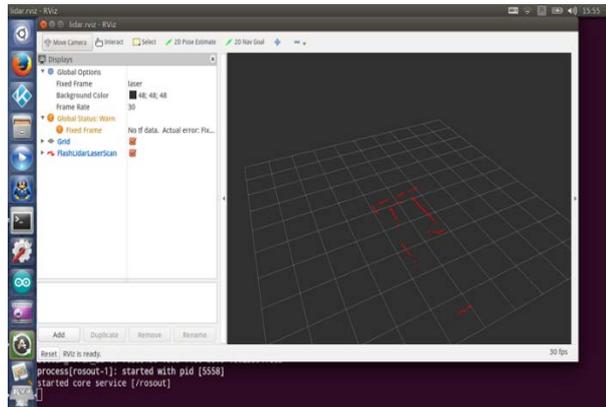


Figure 4. Point cloud image of lidar
图 4. 激光雷达的点云图

3.2. 搭载深度相机 Astra 的实验

在 gazebo 中进行过仿真实验后, 接着在实际的 Turtlebot 机器人上搭载深度相机 Astra 来进行实验。首先对实验室进行建图, 图 5 为实验室实景图, 实验室里面的主要设施有四排桌子、一排实验设备以及凳子若干。通过 Gmapping 建图得到的实验室地图如图 6 所示。可以看到, 最远的边界发生了一点偏移。这就是前文所说的 Gmapping 有时会出现建图错位的情况。同样的, 笔者还是进行了多次建图实验, 并记录了实验室的长度和宽度进行数据分析。

数据分析的结果如表 1 所示。从表中可以看出, 和仿真实验相比, 实验室长度的建图误差在 0.8% 到 2.3% 的范围内, 与仿真环境的长度的建图误差相差不大。但在实验室宽度的建图误差方面变化较大。除一次误差是 3.895%, 其他几次的误差都超过了 5%, 这就是建图错位所带来的影响。随后进行自主导航的实验。如图 7 所示, 机器人依照指令到达指定目标点, 完成自主导航。在运动过程中, 机器人也并不是完全根据已有地图的信息进行路径规划。机器人也会根据实际情况, 做出实时调整, 图中紫色部分便是机器人接收到的实时信息。



Figure 5. Real picture of the laboratory
图 5. 实验室实景图

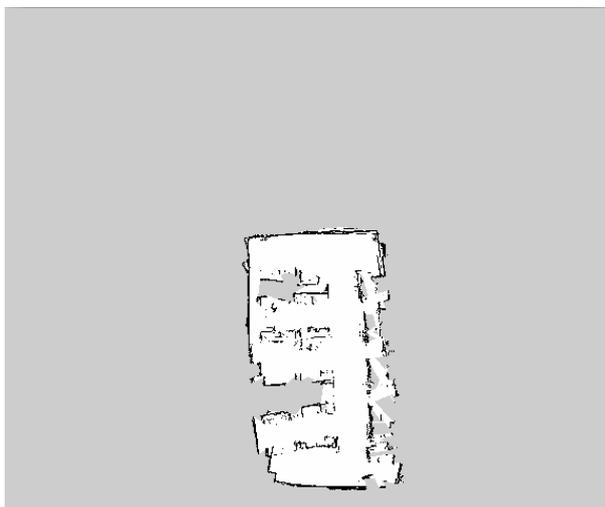


Figure 6. Map of the laboratory
图 6. 实验室地图

Table 1. Experiment data analysis of depth camera Astra
表 1. 搭载深度相机 Astra 的实验数据分析

测量序号	实验室长度(m)	误差(%)	实验室宽度(m)	误差(%)
1	12.130	2.256	6.376	7.326
2	12.167	1.958	6.532	5.058
3	12.249	1.297	6.612	3.895
4	12.229	1.459	6.474	5.901
5	12.306	0.838	6.395	7.049
真实值	12.410	-	6.880	-
平均值	12.216	1.563	6.478	5.843

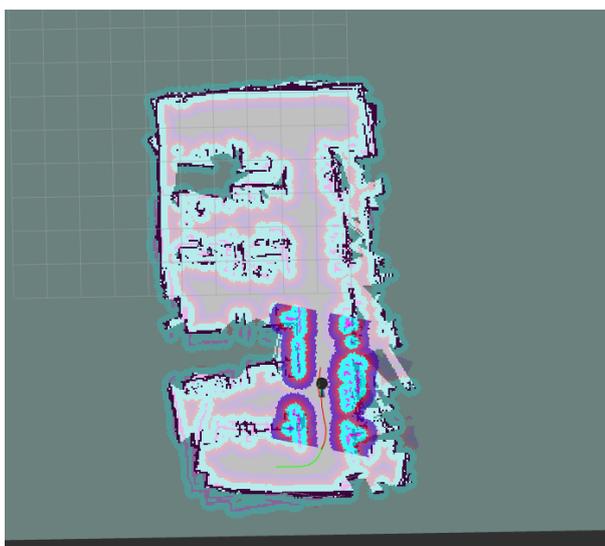


Figure 7. Autonomous navigation experiment
图 7. 自主导航实验

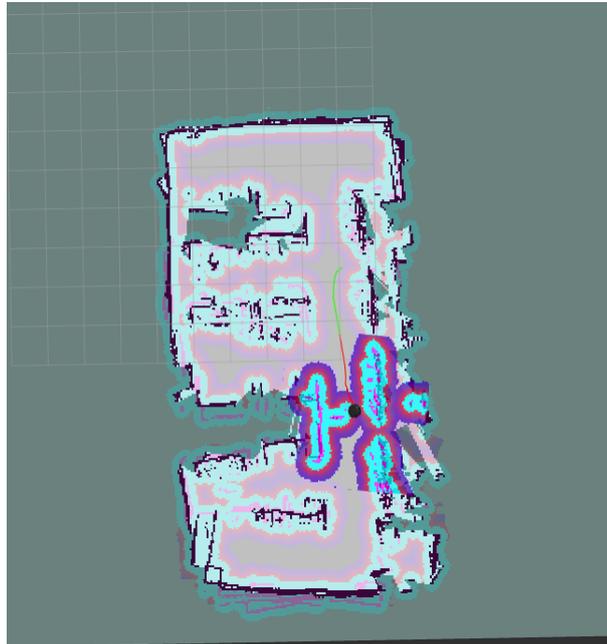


Figure 8. Obstacle avoidance experiment
图 8. 避障实验

接下来是避障实验。在机器人的原定路线上添加新的障碍物。给机器人指定一目标点, 使其在前往目标点的过程中一定会经过障碍物, 然后让机器人开始自主导航。如图 8 所示, 机器人规划了一条新的路径, 避开了障碍物, 实现了避障功能。

最后进行的是实时障碍物的避障测试。首先, 在主机上给机器人指定一目标点, 使机器人进行自主导航。然后在机器人前往目标点的过程中故意出现在机器人的前进路线上, 观察机器人的行为。经过多次测试后发现, 和仿真一样, 机器人可以避开 1.5 m 以外的障碍物。

3.3. 搭载激光雷达 EAI F4 的实验

作为对比, 我们还进行了在实际的 Turtlebot 机器人上搭载激光雷达 EAI F4 的测试。具体过程与搭载 Astra 的实验类似, 首先还是对实验室进行建图, 建图结果如图 9 所示。可以发现, 通过激光雷达建图所得的地图边界要比通过深度相机建图所得的地图边界清晰一些, 整张地图也没有发生错位。同搭载深度相机 Astra 时的实验一样, 本文还是进行了多次建图实验, 并记录了实验室的长度和宽度进行数据分析, 数据分析的结果如表 2 所示。由表中可见, 和搭载深度相机 Astra 的实验相比, 实验室长度和宽度的建图误差都小了很多。实验室长度的建图误差只有一次超过了 2%, 其余几次都没有超过 1.3%。实验室宽度的建图误差情况类似, 只有一次超过了 2%, 其他几次都没有超过 1.6%。这就证明在建图方面激光雷达比深度相机精度更高。

随后也是自主导航的实验。通过图 10 可见, Turtlebot 机器人可以正确抵达指定地点。通过图 11 可见, 在导航过程中, 对于在原定路线上新添加的障碍物, Turtlebot 机器人也能够及时避开。但是有的时候, 激光雷达会径直撞上障碍物, 并没有及时避开。这与激光雷达获取的实时环境信息较少有关, 因为激光雷达只能扫描它自身所在的平面。

通过实验可以发现, SLAM 建图过程中激光雷达 EAI F4 的效果要比较好, 速度也比较快, 这是因为激光雷达在建图时是在进行 360°扫描。这个特点也使得机器人在搭载激光雷达进行自主导航时运动更为流畅, 很少停下来进行再次定位和扫描。而且, 通过激光雷达建图得到的地图很少发生错位。而深度相

机的建图速度就比较慢了。由于深度相机只能获得正前方的环境信息，在建图过程中，机器人有时会停下来原地 360°转圈以便获得四周的环境信息。另外，在面对特征不明显的障碍物例如空白墙壁的时候，它有可能不会获得任何数据，需要变换角度进行多次扫描。

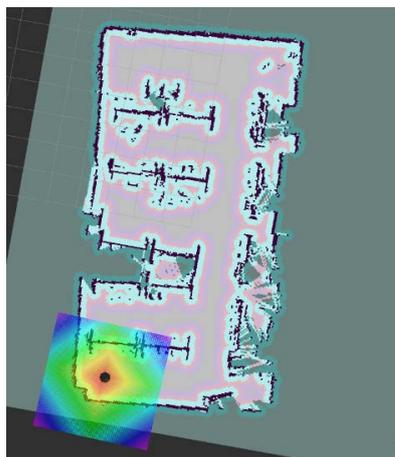


Figure 9. Map drawn by lidar

图 9. 激光雷达建图

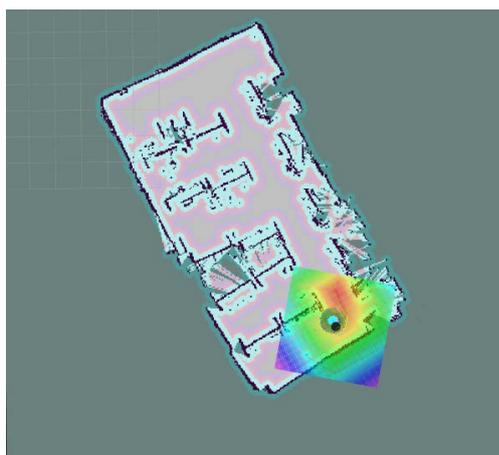


Figure 10. Autonomous navigation experiment

图 10. 自主导航实验

Table 2. Experiment data analysis of EAI F4 lidar

表 2. 搭载激光雷达 EAI F4 的实验数据分析

测量序号	实验室长度(m)	误差(%)	实验室宽度(m)	误差(%)
1	12.320	0.725	6.914	0.494
2	12.340	0.564	6.890	0.145
3	12.360	0.403	6.733	2.137
4	12.126	2.288	6.773	1.555
5	12.261	1.201	6.973	1.352
真实值	12.410	-	6.880	-
平均值	12.281	1.039	6.857	0.334

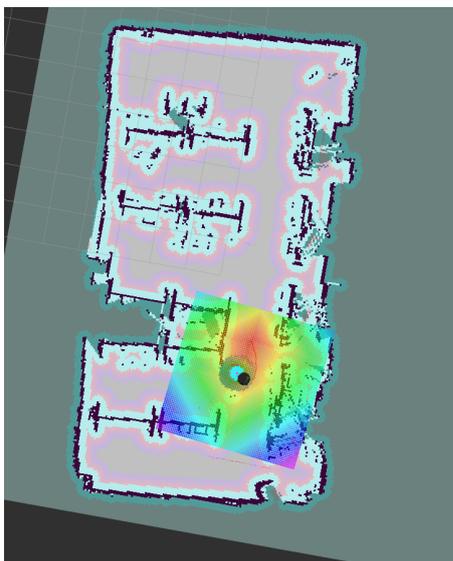


Figure 11. Obstacle avoidance experiment

图 11. 避障实验

不过, 在自主导航和避障方面, 深度相机的表现要比激光雷达好。在同一方向上, 深度相机可以获得更多的信息。在进行自主导航的时候, 深度相机能观测到更多的障碍物, 因此它的避障效果也就更好。相反的, 激光雷达只能在一个平面上进行扫描, 这使得它获得的信息有限, 往往会忽略一些过高或过低的障碍物, 进而影响自主导航和避障时的效果。

4. 结束语

本文虽然初步实现了机器人的自主导航和避障功能, 但距离投入生产和使用还有很长的路要走。尤其是安全性和准确性方面, 还有着很大的进步空间。当机器人搭载激光雷达时, 面对实时障碍物时所展现的避障能力不是很理想。而深度相机面对实时障碍物时也需要一定的预处理空间。除此之外, 通过实验可以发现, 不管是深度相机还是激光雷达, 单一传感器在进行室内环境的 SLAM 和自主导航时都有着各自的问题。相比而言, 深度相机 Astra 在建图时虽然速度较慢, 但它可以获得更加全面的信息。因此在进行自主导航的时, 其效果要好一些。当然, 深度相机也有一个缺点, 即在面对特征不明显的障碍物例如空白墙壁的时候, 它有可能不会获得任何数据, 需要变换角度进行多次扫描。因此, 要想设计出能在实际生活中使用的自主移动机器人, 可以尝试同时使用两款传感器, 结合他们的优点, 完善现有功能, 同时使机器人的安全性进一步提升[10], 这是今后要做的工作。

基金项目

本文得到四川省科技厅重点研发项目(2020YFG0119)的部分资助。

参考文献

- [1] 许玉龙. 基于 ROS 的工业机器人建图导航技术研究[D]: [硕士学位论文]. 哈尔滨: 哈尔滨工业大学, 2017.
- [2] 陈卓, 苏卫华, 安慰宁, 秦晓丽. 移动机器人 SLAM 与路径规划在 ROS 框架下的实现[J]. 医疗卫生装备, 2017, 38(2): 109-113.
- [3] 祝甜一. 基于 ROS 的自主式救援机器人 SLAM 和导航系统研究[D]: [硕士学位论文]. 南京: 东南大学, 2015.
- [4] 应宏钟. 基于 ROS 的工业机器人系统 SLAM 技术研究[D]: [硕士学位论文]. 成都: 西南交通大学, 2018.

-
- [5] 田亮亮. 基于 ROS 的移动机器人未知区域探索与环境建模[D]: [硕士学位论文]. 沈阳: 东北大学, 2015.
- [6] 李红娜. 基于 Turtlebot 的 AGV 自主导航控制方法研究[D]: [硕士学位论文]. 大连: 大连理工大学, 2017.
- [7] 田凯乔. 基于 ROS 及深度摄像机的智能避障机器人的研究与实现[D]: [硕士学位论文]. 长春: 长春工业大学, 2018.
- [8] 韩轻. 基于 ROS 的室内移动服务机器人定位与导航系统的研究与开发[D]: [硕士学位论文]. 成都: 电子科技大学, 2017.
- [9] Product Introduction of Astra, Astra S & Astra Pro. <https://orbbe3d.com/product-astra-pro/>
- [10] 蒋林, 方东君, 雷斌, 李维刚. 单目视觉移动机器人导航算法研究现状及趋势[J]. 计算机工程与应用, 2021, 57(5): 1-9.