

• 计算机工程与应用 •



一种室内高动态环境的机器人定位方法

黄 山¹, 黄洪钟^{1*}, 曾 奇², 钱华明¹

(1. 电子科技大学系统可靠性与安全性研究中心 成都 611731; 2. 成都越凡创新科技有限公司 成都 610097)

【摘要】该文提出一种能让机器人在室内动态环境中进行长时间稳定定位的方法。该方法既能实现对高动态物体的过滤又能实现半静态物体的更新, 在去除动态物体对定位性能影响的同时还能利用半静态物体中提供的定位信息提高定位性能。把动态物体的处理分为高动态物体的滤除和半静态物体的更新两部分。对于高动态物体滤除, 考虑到定位系统的特性, 提出延迟对比法和跟踪法相结合的动态物体检测方法; 对于半静态物体的更新, 采用位姿图优化加栅格地图覆盖的方式实现地图的动态更新。两种方法的结合让机器人能实现在动态环境中的长时间稳定定位。经过一系列实验和一年的实际运行表明: 该方法能实现机器人在动态环境中的长时间定位, 克服高动态物体的影响, 同时让机器人的地图始终保持和当前环境一致。

关 键 词 动态环境; 定位算法; 地图更新; 机器人

中图分类号 TP242.2 **文献标志码** A **doi**:10.12178/1001-0548.2021004

A Robot Localization Method in Indoor Dynamic Environment

HUANG Shan¹, HUANG Hong-zhong^{1*}, ZENG Qi², and QIAN Hua-ming¹

(1. Center for System Reliability and Safety, University of Electronic Science and Technology of China Chengdu 611731;

2. EvenTec Co. Ltd. Chengdu 610097)

Abstract Localization is one of the core technologies for mobile robots to achieve full autonomous movement and is a prerequisite for other autonomous tasks. The robot working environment is dynamic in most cases, so the localization algorithm must overcome the effects of dynamic changes in the environment. The paper proposes a localization algorithm that allows the robot to perform robust and life-long localization in dynamic environment. The algorithm can not only filter out high-dynamic objects but also update semi-static object on the map at the same time, and it can also use the information provided in semi-static objects to improve localization performance. In this paper, the processing of dynamic objects is divided into two parts: filtering of high-dynamic objects and updating of semi-static objects. For high dynamic object filtering, a dynamic object detection method combining a delay comparison method and a tracking method is proposed by observing the characteristics of localization system; for the update of semi-static objects, this paper uses the pose graph optimization and occupancy map to implement the dynamic update of the map. The combination of the two methods allows the robot to achieve long-term stable localization in a dynamic environment. The experimental results demonstrate that the proposed method allows the robot to achieve long-term localization, overcome the effects of high-dynamic objects and keep the map always consistent with the environment.

Key words dynamic environment; localization algorithm; map update; robotics

随着移动机器人技术的发展, 全自主移动机器人在仓储物流、野外救援、园区送货和商场服务等场景得到了越来越广泛的应用。移动机器人的定位是移动机器人进行全自主运动的前提, 定位技术是移动机器人的核心技术之一, 因此开展移动机器人定位技术的研究对于实现移动机器人的全自主运动具有重要意义。

移动机器人定位方法可以分为两大类^[1]: 基于

无线电信号的定位 (wireless-based) 和基于地图的定位 (map-based localization)。基于无线电的定位方法主要有卫星定位方法、UWB 定位方法、蓝牙定位方法、wifi 定位方法和 RFID 定位方法等^[2]。这些定位方法的优点是能实现相对于某一个基准坐标系的绝对定位, 定位误差不会累计, 在有信号的情况下可以实现比较准确的定位。然而基于无线电定位的方法在受到遮挡时, 定位质量会急剧下降, 并

收稿日期: 2021-01-05; 修回日期: 2021-03-09

基金项目: 国家重点研发计划 (2017YFB1301300)

作者简介: 黄山 (1992-), 男, 博士生, 主要从事机器人定位技术方面的研究。

通信作者: 黄洪钟, E-mail: hzhuang@uestc.edu.cn

且需要预先设置定位用的基站, 因此严重地限制了其在移动机器人领域的应用。基于地图的定位方法即事先构建机器人运行环境的地图, 然后利用机器人当前传感器的观测数据和预先构建的地图进行匹配, 得到机器人位姿。由于该方法不需要对环境进行改造, 适用于移动机器人的推广, 因此在移动机器人定位领域得到了广泛的应用。按照传感器类型进行分类, 主要可以分为基于激光雷达的方法 (lidar-based) 和基于相机的方法 (camera-based)^[3]。由于激光雷达具有测距精度高、抗干扰能力强等优点, 因此本文使用激光雷达进行定位, 以下所说的定位方法均指激光雷达定位。

机器人定位的最大挑战来自于环境中的动态物体, 环境中动态物体分为高动态物体 (high-dynamic object) 和半静态物体 (semi-static object) 两大类。高动态物体是指环境中位置时刻在发生变化的物体, 特别是机器人观测到该物体时, 该物体处于运动状态, 如行人、手推车等; 半静态物体是指环境中某些位置会发生变动但是变动频率极低的物体, 特别是机器人观测到该物体时, 该物体处于静止状态, 如家具、停车场中的汽车等。基于地图的定位方法的基本原理是用传感器观测到的数据和地图进行匹配, 从而修正机器人的位置。显然高动态物体的位置在实时变动, 因此其不能提供定位信息。而半静态物体的位置在短时间内不发生改变, 在其位置不变的时间段内, 可以提供定位信息。因此对于这两种物体需要有不同的处理方式, 高动态物体直接进行过滤而半静态物体需要更新到地图中以提高机器人定位的稳定性。

目前移动机器人领域应用最知名、应用最广泛的方法为文献 [4] 提出的蒙特卡洛定位方法 (Monte Carlo localization, MCL)。该方法用粒子 (particle) 来表示机器人的位姿, 然后用里程计数据进行粒子的传播, 用激光雷达观测数据和地图的匹配程度来作为测量值更新粒子的权重, 通过粒子滤波器的迭代更新实现对于机器人位置的估计, MCL 默认环境是静态的, 在动态环境中表现较差。文献 [5] 在 MCL 的基础上, 提出通过比较激光雷达期望测距和实际测距的差来估计该束激光雷达数据是否来自动态物体, 该方法在处理高动态物体时取得了不错的效果。文献 [6] 通过动态物体跟踪来分辨激光雷达数据是否来自动态物体。以上方法虽然能比较好地处理高动态物体, 但是无法处理半静态物体。而实际环境中存在大量的半静态物体, 如商场中店铺

位置的改变、店铺的装修及家具位置的移动等, 因此对于半静态物体的处理也很重要。

对半静态物体进行处理, 基本的思路是实现地图更新, 对半静态物体进行建模。文献 [7-8] 扩展了原本的覆盖栅格地图 (occupancy grid map), 把动态物体也包含到覆盖栅格地图中。文献 [9] 提出了动态覆盖栅格 (dynamic occupancy grid) 的概念, 该方法用一个隐马尔可夫模型来表示二维栅格的占用概率和动态物体的概率。文献 [10] 使用 RBPF (rao-blackwellized particle filter) 来同时估计机器人位姿和环境状态, 实现地图更新。由于以上方法都使用了 RBPF 来进行机器人位姿的估计和地图更新的计算, 因此要么计算量巨大, 不能满足实时性的要求; 要么会消耗巨量的内存, 在资源受限的情况下无法运行。

除了使用粒子滤波之外, 文献 [11-12] 使用位姿图 (pose-graph) 来处理动态物体, 本文工作也受其启发。最大不同在于本文使用栅格地图, 而文献 [11-12] 使用点云地图。点云地图对噪声比较敏感并且容易受到动态物体的影响, 而栅格地图对于这些干扰具有一定的抵抗能力, 因此抗干扰能力更强。

商用服务机器人的运行场景中充斥着大量的高动态物体和半静态物体, 因此商用机器人的长时间稳定应用离不开能同时处理高动态物体和半静态物体的定位系统。同时由于成本的限制, 商用服务机器人的计算资源和内存资源都比较受限, 这决定了其定位系统不能耗费太多的计算和内存资源。

为了克服以上方法的缺点, 同时实现机器人的长时间稳定运行, 本文提出了一种能同时处理高动态物体和半静态物体的定位方法, 同时该方法能在奔腾处理器上实时运行。该方法通过高动态物体的检测与跟踪过滤掉激光雷达数据中的高动态物体数据, 极大提高了机器人在高动态环境中运行的稳定性。同时由于半静态物体能提供丰富的定位信息, 因此本文对于半静态物体的处理不是简单地去除, 而是通过地图更新的方式把半静态物体信息融入到静态地图中, 并提出了一种新的地图更新方法, 该方法能快速实现地图更新。

1 系统框架

本文提出的定位算法整体的系统框架如图 1 所示。

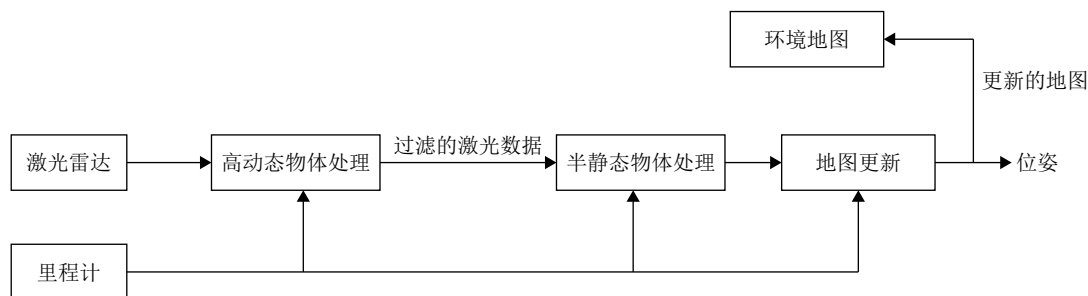


图 1 整体系统框架

从图 1 可以看出，定位算法主要分为 3 个模块：高动态物体处理模块、半静态物体模块和地图更新模块，使用的传感器包括激光雷达和轮式里程计。高动态物体处理模块接收激光雷达数据和轮式里程计数据，以里程计数据为真值，对动态物体进行检测和跟踪。高动态物体处理模块检测出激光数据中包含的高动态物体，并进行过滤，输出一个不带高动态物体的过滤激光帧；过滤激光帧和里程计数据输入半静态物体处理和地图更新模块，半静态物体处理和地图更新模块实现对于机器人的定位和地图更新，把半静态物体更新到地图中，并且输出更新后的地图，同时输出机器人当前的位置。

2 高动态物体的处理

高动态物体检测的目的是过滤掉相对于机器人正在运动的物体，如行人、推车等。因为高动态物体不但无法提供定位信息，反而会降低定位的成功率。本文对高动态物体的处理采用对比法和跟踪法相结合的方式。为了克服对比法的缺点，本文利用定位系统的特点，提出了一种延迟对比方法，大大提高了动态物体检测的性能。

2.1 延迟对比法

对比法是用当前激光帧数据 $z(t)$ 和上一帧 $z(t-1)$ 或者上几帧激光数据 $z(t-k)$ 进行对比，如果当前激光帧 $z(t)$ 中有数据落在过去激光帧的视野范围内，则说明该数据是由动态物体造成的。因为落在上一帧数据的视野范围内，说明上一帧数据在对应位置没有观测到该物体，而当前帧在对应位置观测到了该物体，因此该物体只可能是动态物体。对比法的工作原理如图 2 所示。

图 2 中，坐标系表示机器人当前的位置，黄色点表示上一帧激光数据，洋红色的多边形表示上一帧激光数据对应的视野范围，蓝色点表示当前帧激光数据，红色点表示当前帧中被检测为动态物体的数据，该数据实际上是一个运动的人，红色点

位于上一帧的视野范围之内，因此其被认为是动态物体。

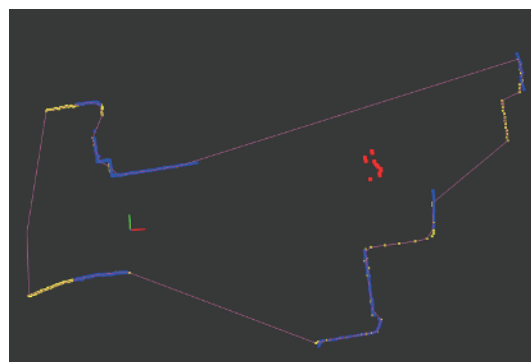


图 2 对比法工作原理图

相对于机器人来说，动态物体分为向机器人靠近和远离机器人两种。从对比法的原理来说，只能处理向机器人靠近的动态物体，因为如果动态物体正在远离机器人，那么该动态物体并不在上一帧数据的视野范围，所以对比法无法检测出远离机器人的动态物体。

为了克服对比法的问题，本文提出了延迟对比法。对于机器人来说，里程计数据的精度在较短的时间内可以认为是比较高的，因此定位系统实际上只需要定期修正里程计的漂移即可，不需要进行实时修正。即用激光数据进行定位时，并不需要用当前最新的激光数据 $z(t)$ ，而可以用过去一段时间的激光数据 $z(t-1)$ ，比如几秒钟之前的数据。如果用过去的激光数据 $z(t-1)$ 来定位，那么用定位的数据 $z(t-1)$ 不但可以和以前的数据 $z(t-2)$ 进行对比来识别动态物体，同时也可以和未来的数据 $z(t)$ 进行对比来识别动态物体，延迟对比法的工作原理如图 3 所示。图中洋红色的多边形表示 $z(t-1)$ 的视野范围，绿色多边形表示 $z(t+1)$ 的视野范围，红色的点表示一个正在远离机器人的行人。显然红色的点在洋红色多边形外，因此对比法无法识别出其是一个动态物体，然而红色的点在绿色多边形的内部，因此用延迟对比法可以识别出该行人是一个动态物体。

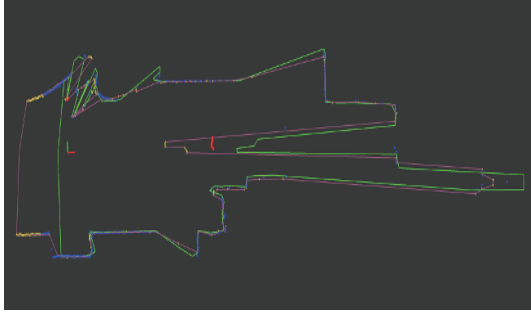


图3 延迟对比法工作原理图

显然, 和过去数据进行对比可以识别出向机器人靠近的动态物体, 而和未来数据进行对比则可以识别出远离机器人的物体, 因此延迟对比法能识别两种类型的动态物体, 克服了对比法只能识别向机器人靠近的物体的缺点, 能大幅提高机器人对于动态物体的处理, 增强机器人在高动态环境定位的稳定性。

2.2 跟踪法

与对比法不同, 跟踪法^[13]不利用激光帧视野范围形成的空白区域, 只利用激光帧本身观测到的数据。跟踪法首先把一帧激光雷达数据根据欧式距离进行聚类, 从而把一帧数据分解成一个个不同的类, 这些不同的类跟上一帧的数据进行数据关联, 确定是否为同一个物体, 如果确定为同一个物体, 则可以认为是跟踪成功; 如果没有匹配上, 则可以认为是新出现的物体, 分配一个新的 ID 进行跟踪。判断两个类是否为同一个物体可以根据两个类的距离以及相似度等。每新出现一个物体, 则分配一个卡尔曼滤波器进行跟踪来计算该物体的位置和速度等参数, 如果连续成功地跟踪了好几帧, 则根据物体的速度判断其是否为动态物体, 速度超过某一个阈值则认为是动态物体。

2.3 延迟对比法和跟踪法的结合

延迟对比法的优势是在看到动态物体的瞬间就能进行分辨, 但是没有对不同帧之间的动态物体进行数据关联, 缺失上下文信息, 在传感器存在噪声的情况下, 存在漏检的可能; 而跟踪法对不同帧之间的动态物体进行数据关联让动态物体跟踪比较稳定, 但是必须跟踪物体一定时间之后才能判断该物体是否为动态物体。

为了克服彼此的缺点, 本文将延迟对比法和跟踪法两种方法结合起来, 对于每一个延迟对比法检测出来的动态障碍物都分配一个滤波器进行跟踪, 并且在不同帧之间进行数据关联, 实现动态障碍物

的实时检测和稳定跟踪的效果。

3 半静态物体的处理

跟高动态物体不同, 半静态物体能提供丰富的定位信息, 因此不能直接视为噪声进行滤除, 而是需要充分利用半静态物体提供的信息来实现机器人的长时间稳定定位^[14]。本文在进行半静态物体处理时, 假设激光数据中的高动态物体已经被滤除, 激光数据中只包含有静态物体(跟当前地图吻合的物体)和半静态物体(跟当前地图不吻合的物体)。本文在处理半静态物体时, 采用位姿图优化^[15]的形式更新地图。

3.1 位姿图优化

位姿图是一种机器人状态估计的描述方法, 用节点表示机器人的位姿, 用边表示两个节点之间的空间约束, 如图4所示。

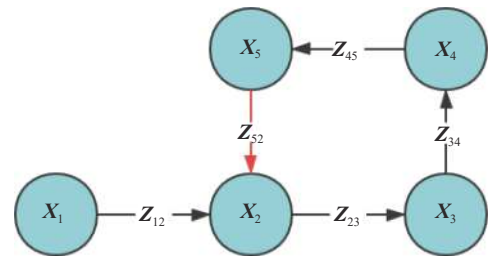


图4 位姿示意图

图4中, X_i 表示移动机器人位姿, 本文中机器人在平面上运动, 因此 $X_i = (x_i, y_i, \theta_i)$ 。 Z_{ij} 表示两个节点之间的相对位姿关系, Z_{ij} 可以通过里程计或者扫描匹配得到, 黑色的边表示帧间约束, 红色的边表示回环约束。本文使用文献^[16]的算法进行粗搜索, 使用文献^[17]的算法进行细搜索的方式来计算两个节点的相对位姿关系。在给定帧间约束和回环约束的前提下, 通过后端优化能得到机器人轨迹的极大似然估计。已知机器人整条轨迹的位姿, 可以通过覆盖栅格构图算法^[18]生成对应的覆盖栅格地图。

无论是里程计测量还是帧间匹配都有误差, 因此随着机器人的行走, 误差会不断累积。只有当机器人进入已知区域, 形成回环检测时才能消除误差, 图4中红色的边表示回环约束, X_5 和 X_2 观测到环境中同一个特征, 因此可以通过扫描匹配得到相对位姿 Z_{52} , 从而构建出回环约束。当回环约束构建完毕之后, 即可以通过优化求解机器人轨迹的极大似然估计。

节点之间相对位姿的观测值用 Z_{ij} 和 Λ_{ij} 表示,

其中 \mathbf{Z}_{ij} 表示节点 i 和节点 j 之间的相对位姿关系, \mathbf{A}_{ij} 表示 \mathbf{Z}_{ij} 对应的信息矩阵。节点之间相对位姿的预测值即当前两个节点之间的相对位姿为:

$$\mathbf{h}(\mathbf{X}_i, \mathbf{X}_j) = \begin{bmatrix} \mathbf{R}^T(\theta_i) \left(\begin{bmatrix} x_j \\ y_j \end{bmatrix} - \begin{bmatrix} x_i \\ y_i \end{bmatrix} \right) \\ \theta_j - \theta_i \end{bmatrix} \quad (1)$$

式中, $\mathbf{R}(\theta)$ 表示旋转角度是 θ 时的旋转矩阵。显然误差函数即为预测值和观测值的差, 其表达式为:

$$\mathbf{e}_{ij}(\mathbf{X}) = \mathbf{e}(\mathbf{X}_i, \mathbf{X}_j) = \mathbf{h}(\mathbf{X}_i, \mathbf{X}_j) - \mathbf{Z}_{ij} \quad (2)$$

求解机器人位姿的极大似然估计, 等价于求解式 (3) 所示的目标函数:

$$\mathbf{X}^* = \arg \min_{\mathbf{X}} \sum \mathbf{e}_{ij}^T(\mathbf{X}) \mathbf{A}_{ij} \mathbf{e}_{ij}(\mathbf{X}) \quad (3)$$

由于预测函数是一个非线性函数, 因此机器人位姿估计问题是一个非线性最小二乘 (nonlinear least square) 问题。对于式 (3) 的求解, 可以通过局部线性化, 然后迭代更新。方程的非线性来自于误差函数, 因此可以把误差函数进行泰勒展开, 然后取一阶近似实现整个目标函数的线性化:

$$\mathbf{e}_{ij}(\mathbf{X} + \Delta \mathbf{X}) \approx \mathbf{e}_{ij}(\mathbf{X}) + \mathbf{J}_{ij} \Delta \mathbf{X} \quad (4)$$

式中, \mathbf{J}_{ij} 表示误差函数对于机器人位姿 \mathbf{X} 的 Jacobian 矩阵。由式 (1) 和式 (2) 可知, $\mathbf{e}_{ij}(\mathbf{X})$ 只与 \mathbf{X}_i 和 \mathbf{X}_j 有关, 与其他项无关, 因此 \mathbf{J}_{ij} 只有 \mathbf{X}_i 和 \mathbf{X}_j 对应的两项为非 0, 其他项均为 0。因此有:

$$\mathbf{J}_{ij} = \begin{pmatrix} 0 & \cdots & \frac{\partial \mathbf{e}_{ij}(\mathbf{X})}{\partial \mathbf{X}_i} & \cdots & \frac{\partial \mathbf{e}_{ij}(\mathbf{X})}{\partial \mathbf{X}_j} & \cdots & 0 \end{pmatrix} \quad (5)$$

式中, $\mathbf{e}_{ij}(\mathbf{X})$ 对于 \mathbf{X}_i 的偏导数为:

$$\frac{\partial \mathbf{e}_{ij}(\mathbf{X})}{\partial \mathbf{X}_i} = \mathbf{J}_i = \begin{bmatrix} -\mathbf{R}^T(\theta_i) & \frac{\partial \mathbf{R}^T(\theta_i)}{\partial \theta_i} \left(\begin{bmatrix} x_j \\ y_j \end{bmatrix} - \begin{bmatrix} x_i \\ y_i \end{bmatrix} \right) \\ 0 & -1 \end{bmatrix} \quad (6)$$

$\mathbf{e}_{ij}(\mathbf{X})$ 对于 \mathbf{X}_j 的偏导数为:

$$\frac{\partial \mathbf{e}_{ij}(\mathbf{X})}{\partial \mathbf{X}_j} = \mathbf{J}_j = \begin{bmatrix} \mathbf{R}^T(\theta_i) & 0 \\ 0 & 1 \end{bmatrix} \quad (7)$$

联合式 (5)~式 (7) 可以得到误差函数 $\mathbf{e}_{ij}(\mathbf{X})$ 的一阶近似函数式 (4) 的准确表达式, 把式 (4) 代入式 (3) 中, 则目标函数变成:

$$\begin{aligned} F(\mathbf{X} + \Delta \mathbf{X}) &= \sum (\mathbf{e}_{ij}(\mathbf{X}) + \mathbf{J}_{ij} \Delta \mathbf{X})^T \mathbf{A}_{ij} (\mathbf{e}_{ij}(\mathbf{X}) + \mathbf{J}_{ij} \Delta \mathbf{X}) = \\ &= \sum \mathbf{e}_{ij}^T \mathbf{A}_{ij} \mathbf{e}_{ij} + 2 \mathbf{e}_{ij}^T \mathbf{A}_{ij} \mathbf{J}_{ij} \Delta \mathbf{X} + \Delta \mathbf{X}^T \mathbf{J}_{ij}^T \mathbf{A}_{ij} \mathbf{J}_{ij} \Delta \mathbf{X} = \\ &= \sum c_{ij} + 2 \mathbf{b}_{ij}^T \Delta \mathbf{X} + \Delta \mathbf{X}^T \mathbf{H}_{ij} \Delta \mathbf{X} = \\ &= c + 2 \mathbf{b}^T \Delta \mathbf{X} + \Delta \mathbf{X}^T \mathbf{H} \Delta \mathbf{X} \end{aligned} \quad (8)$$

从式 (8) 可得, 线性化后的目标函数是关于待求解变量的二次函数, 因此目标函数的极小值可以通过对自变量求导, 然后令导数等于 0 的方式进行求解:

$$\frac{\partial F(\mathbf{X} + \Delta \mathbf{X})}{\partial \Delta \mathbf{X}} = 2 \mathbf{H} \Delta \mathbf{X} + 2 \mathbf{b} = 0 \quad (9)$$

显然式 (9) 是一个比较容易求解的线性方程组。这里得到的解是线性化目标函数的最小值, 只在当前解的邻域内成立, 因此需要不断迭代, 直到解收敛。解收敛则可得到式 (3) 所示非线性方程组的解。为了简单起见, 本文使用 G2o^[19] 求解该问题。

3.2 地图更新

机器人进行定位时, 假设已经具备环境的静态地图。当环境中不存在半静态物体, 即环境没有发生变化时, 直接用静态地图进行匹配即可以实现机器人的稳定定位。当环境发生变化时, 静态地图与实际环境不匹配, 会导致定位失败。因此本文用激光雷达数据和地图匹配是否失败作为判断环境是否发生变化的指示。如果当前数据和地图匹配失败, 则说明环境发生变化, 因此本文把匹配失败作为触发地图更新的条件。本文地图更新的流程如图 5 所示。

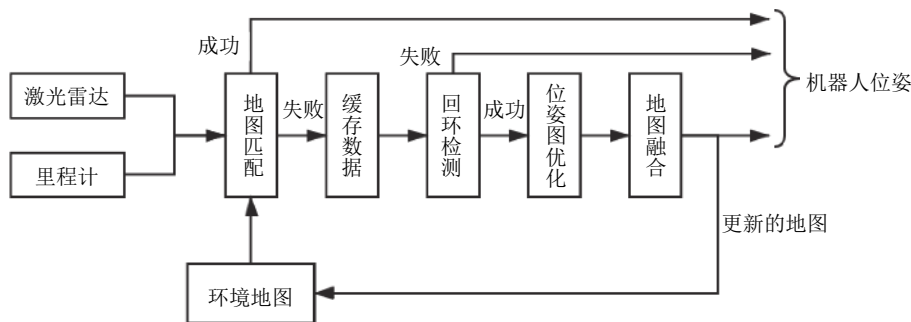


图 5 地图更新流程图

从图5中可以看出, 当环境没有发生变化时, 用当前数据跟静态地图进行匹配得到位姿。当环境发生变化时, 会触发匹配失败, 此时位姿只能依赖激光雷达数据的帧间匹配, 而无法被地图修正。地图匹配失败说明环境已经改变, 静态地图跟真实环境不再匹配, 此时需要用当前的观测数据对地图进行更新, 因此定位系统会缓存匹配失败的数据, 形成一个单链的位姿图。当机器人重新进入已知区域时, 回环检测成功, 形成一个全局的回环约束, 然后通过位姿图优化来修正机器人的轨迹, 最后进行地图的融合。

在位姿图中, 位姿误差不断累积, 因此位姿图中除了首尾节点之外, 其他节点位姿都存在累积误差。位姿图优化就是为了消除帧间匹配的累积误差, 位姿图优化的结果如图6所示。



图6 位姿图矫正前后的误差

从图6中可以看出, 机器人在一个办公室场景中运行, 该环境的静态地图被擦除一部分, 因此机器人行走在被擦除部分对应的环境时, 地图匹配会失败, 地图匹配失败的轨迹为图6中蓝色的轨迹。蓝色轨迹的位姿误差是逐步累积的, 最后一个节点的累积误差最大。当机器人进入已知区域后, 回环检测程序会让其最后一个节点和地图匹配成功, 最后一个位姿得到修正并且形成回环, 图6中黑色的边即为回环约束。最后进行位姿图优化, 整条轨迹的误差都得到了修正, 绿色轨迹即为修正之后的轨迹。

由于位姿图第一个节点位姿和最后一个节点位姿都是跟静态地图匹配得到, 因此优化后的位姿图坐标系跟静态地图坐标系是对齐的。本文把位姿图理解成一个由半静态物体构成的局部子图, 静态地图则表示全局地图。地图更新的过程即把局部子图

与全局地图进行融合的过程。为了最大限度地减少地图融合过程中出现歧义的情况, 本文把融合分为清除和增加两个部分, 即先清除那些地图中不存在的物体, 然后再增加新出现的物体。

首先利用位姿图的位姿和激光数据, 按照覆盖栅格建图算法生成一个栅格地图, 该地图标识了局部子图中的空白区域和占用区域。对于全局地图中的占用区域, 如果它位于局部子图中的空白区域中, 则说明该物体在全局地图中是存在的, 但是并不存在于局部子图中, 其对应的情况是环境中的物体被移走。通过这一步骤可以清除全局地图中那些已经不存在的物体。清除之后, 把局部子图中的物体加入到全局地图中, 为了防止由于定位误差导致的融合地图出现歧义的情况, 如果局部子图中的物体离全局地图中的物体小于一个阈值, 则认为该物体是由于定位误差导致, 并不是真实发生的改变, 因此该物体直接忽略, 不加入到全局地图中。

本文的地图更新方法首先利用位姿图对机器人位姿进行优化, 实现两个坐标系的对齐, 然后生成覆盖栅格地图, 直接在覆盖栅格地图层面进行融合。位姿图优化的复杂度跟节点的数量成正比, 而实际环境中, 环境都是缓慢变化的, 不会发生大规模的突然变化, 因此位姿图中节点的个数一般小于100个。对于这个数量级的位姿图优化, 在奔腾处理器上的时间小于10ms。已知位姿进行覆盖栅格地图生成基本上小于5ms。由于局部子图和全局地图的坐标系对齐, 因此进行栅格地图融合只需要遍历一遍局部子图即可, 在栅格地图的分辨率为5cm的情况下, 栅格地图融合的耗时在10~20ms之间。因此本文的地图更新方法总耗时在25~35ms之间, 完全能满足实际运行的需要。同时地图更新之后, 存储的是新的栅格地图而不是激光的原始数据, 内存的消耗仅仅是一张图片的大小, 不会超过10Mb。因此本算法适用于计算资源和存储资源都比较受限制的商用机器人。

4 实验

为了验证本文提出的定位算法的有效性, 本文用C++实现了全部的代码, 并且把本文的定位模块和建图模块、规划模块以及控制模块进行整合, 形成一个完整的导航系统, 对本文提出的定位算法在动态环境中的定位性能进行了充分测试。测试平台为成都越凡创新公司的全自动售货机器人FANBOT的原型机, 如图7所示。



图7 FANBOT 原型机

图7所示的原型机是一个差分驱动的轮式机器人，该机器人装备有多种传感器，包括RGBD传感器、双目相机和2D激光雷达等，本文只使用2D激光雷达数据。该原型机的处理器为一个奔腾四核处理器。

4.1 高动态物体测试

高动态物体测试场景为机器人在环境中行走，同时有行人在机器人前方来回走动，既有远离机器人的运动，也有靠近机器人的运动。在不进行动态物体滤除的情况下，机器人生成的点云地图如图8所示。

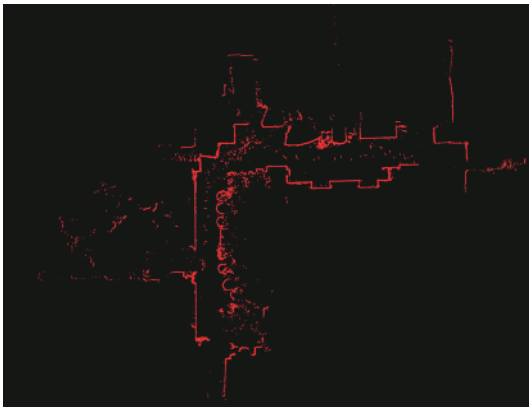


图8 无动态物体滤除

从图8可以看出，行人对机器人生成的点云地图产生了严重的干扰，点云地图中包含大量动态物体造成的数据，让点云地图变得比较杂乱无章，会严重影响机器人的定位性能。经过本文动态物体滤除算法过滤后的点云地图如图9所示。

从图9可以看出，本文的高动态物体去除方法基本上过滤掉了动态物体，对生成的点云地图质量有极大的提高。值得注意的是，除了过滤掉行人之外，有一些不属于行人的激光点也被过滤掉了，这

是因为FANBOT使用低成本激光雷达数据质量较差，正常数据中也会夹杂很多的数据噪点，因此被动态物体滤除算法当成动态物体一并过滤了。从图8和图9的对比可知，本文方法能成功地过滤掉高动态物体，极大地提高点云质量。

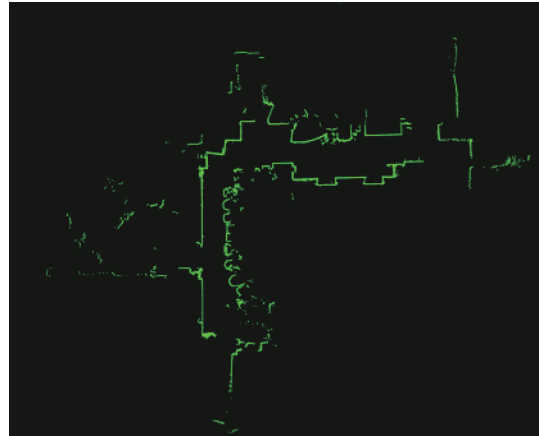


图9 动态物体滤除

4.2 地图更新测试

本文在一个办公室场景进行半静态物体的测试。首先构建该办公室场景的静态地图，然后人为移动环境中的物体来模拟环境变化，让机器人在变化后的环境中进行定位。为了让对比更加明显，特地使用点云地图来进行显示。构建的静态地图如图10所示。

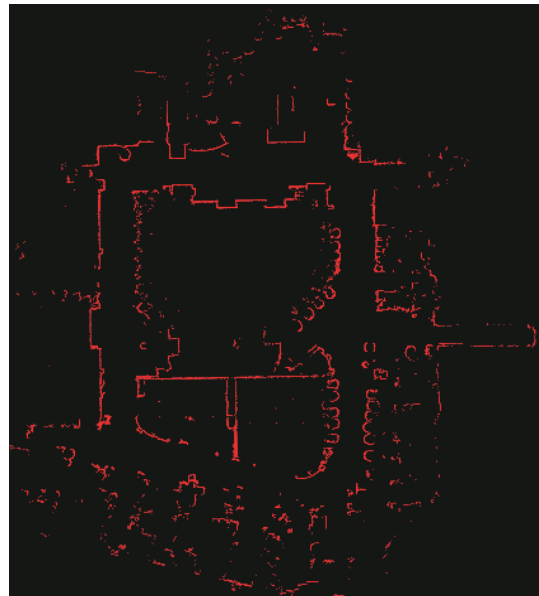


图10 静态地图

为了让机器人在变化后的环境中运行，需定位系统能够构建出环境变动的部分来更新地图。让机器人在办公室中走5圈，最终构建出的地图和静态地图的对比图如图11所示。

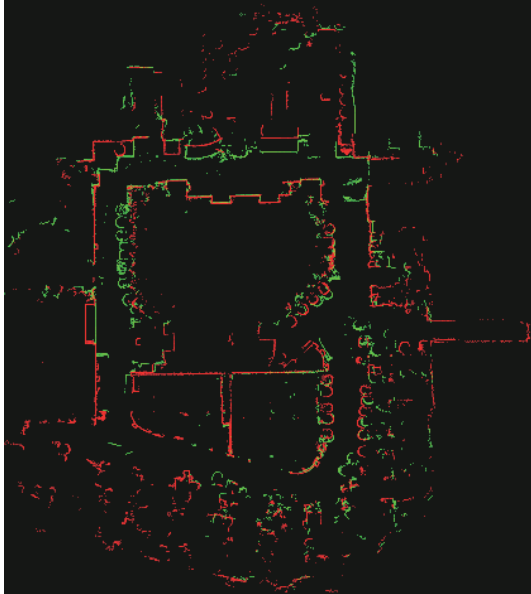
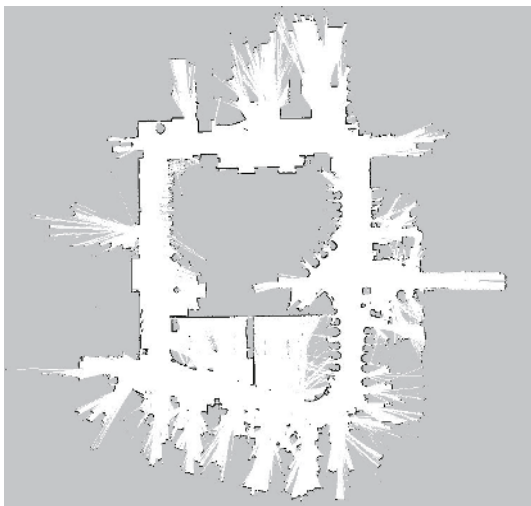


图 11 对比图

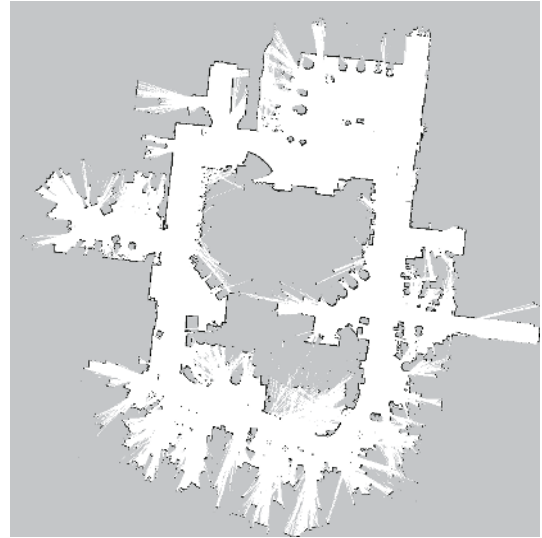
图 11 中红色点云为原始点云, 与图 10 是重合的; 绿色点云为定位系统构建出来的点云。从图 11 可以看出, 在环境发生巨大变化的情况下, 定位系统依然可以进行稳定的定位, 并且自动把环境中变化的部分更新到地图中。证明了该算法在环境发生改变之后, 能自适应的更新地图, 让地图始终跟当前环境保持一致。因此可以处理动态环境中的低动态物体, 让机器人实现长时间稳定的定位。

4.3 长时间定位测试

为了验证本论文定位系统的长时间定位性能, 本文让测试样机在图 12a 所示的环境中进行长达一个月的不间断运行, 机器人运动的过程中环境在不断的发生变化, 在运行一个月之后, 机器人的地图更新为图 12b 所示。



a. 原始地图



b. 更新地图

图 12 机器人运动的地图

从图 12a 和图 12b 的对比中可以看出, 尽管在机器人的运行过程中, 环境发生了巨大改变, 本文的定位方法依然可以实现机器人的稳定定位, 同时可以重建出环境的变化。该实验表明本文的定位算法对于动态环境的适应性, 能在动态环境中实现长时间稳定的定位。

4.4 机器人定位可靠性分析

经过在商场环境下长时间的运行, 发现对于机器人定位稳定性影响最大的 3 个因素为机器人轮胎磨损、机器人被人推动和激光雷达传感器感知能力限制。

机器人轮胎直径在出厂时会进行标定。然而随着机器人运行时间的增长, 轮胎的磨损加剧, 导致机器人轮胎半径发生变化, 和出厂时的误差越来越大, 导致机器人的里程计精度越来越差。而里程计精度对于机器人的定位系统来说是一个重要的传感器, 其不但影响机器人在环境变化时的定位精度, 也会影响用位姿图优化时初始解的准确度, 如果初始解太差, 可能会导致地图更新出错。因此防止机器人轮胎磨损, 或者在机器人轮胎磨损之后能自动标定轮胎直径的功能对于机器人定位稳定性具有重要的意义。

机器人被人推动, 又叫机器人绑架, 是实际运行中影响机器人定位稳定性的另一个重要因素。由于行人对于机器人的好奇, 经常会有人把机器人从一个地方推动到另外一个地方, 导致机器人定位失败。同时激光雷达数据信息量不够丰富, 导致机器人不能快速地进行全局定位, 所以这时需要人为地介入才能解决这个问题。因此如何将激光雷达数据和信息量丰富的传感器数据, 比如视觉传感器, 融

合起来解决机器人绑架问题也值得研究。

最后一个因素是激光雷达属于光学传感器,因此它看不到透明玻璃。然而商场环境中具有大量的透明玻璃,因此在商场中总是会出现激光雷达检测不到物体的情况,此时相当于纯靠机器人的里程计进行航迹推算,从而导致机器人定位漂移。因此融合多个传感器数据,让至少有一个传感器能观测到数据,对于机器人定位的稳定性具有积极影响。

5 结束语

本文提出了一种移动机器人的定位算法,该算法能同时处理环境中的高动态物体和半静态物体,实现机器人在动态环境中长时间的稳定定位。本文利用定位系统不需要使用最新数据的特点,提出能同时处理远离机器人和靠近机器人这两种动态物体的延迟对比法。同时提出一个结合位姿图优化和栅格地图覆盖的方法,实验证明该方法能很好地处理环境中的半静态物体,让机器人的地图保持跟当前环境的一致性。同时总结了实际运行过程中影响机器人定位可靠性的三大因素,后续工作主要是对这3个因素进行进一步的优化。

参 考 文 献

- [1] 高云峰,周伦,吕明睿,等. 自主移动机器人室内定位方法研究综述[J]. 传感器与微系统, 2013, 32(12): 1-5.
GAO Yun-feng, ZHOU Lun, LÜ Ming-rui, et al. Research overview of indoor localization methods for autonomous mobile robots[J]. *Transducer and Microsystem Technologies*, 2013, 32(12): 1-5.
- [2] 张立立,钟耳顺. 无线室内定位技术[C]//中国地理信息系统协会第八届年会论文集,北京,中国: [s.n.], 2004: 966-972.
ZHANG Li-li, ZHONG Er-shun. Wireless indoor location technology[C]//The 8th Annual Conference of China Geographic Information System Association. Beijing, China: [s.n.], 2004: 966-972.
- [3] KUUTTI S, FALLAH S, KATSAROS K, et al. A survey of the state-of-the-art localization techniques and their potentials for autonomous vehicle applications[J]. *IEEE Internet of Things Journal*, 2018, 5(2): 829-846.
- [4] DELLAERT F, FOX D, BURGARD W, et al. Monte Carlo localization for mobile robots[C]//IEEE International Conference on Robotics and Automation. [S.l.]: IEEE, 1999: 1322-1328.
- [5] FOX D, WOLFRAM B, SEBASTIAN T. Markov localization for mobile robots in dynamic environments[J]. *Journal of Artificial Intelligence Research*, 1999, 11: 391-427.
- [6] HAHNEL D, TRIEBEL R, BURGARD W, et al. Map building with mobile robots in dynamic environments[C]//2003 IEEE International Conference on Robotics and Automation. Taipei, Taiwan, China: IEEE, 2003, 2: 1557-1563.
- [7] CHENG C, TAY C, MEKHACHA K. Dynamic environment modeling with gridmap: a multiple-object tracking application[C]//IEEE International Conference on Control, Automation, Robotics and Vision. Singapore: IEEE, 2006: DOI: [10.1109/ICARCV.2006.345399](https://doi.org/10.1109/ICARCV.2006.345399).
- [8] BRECHTEL S, GINDELE T, DILLMANN R. Recursive importance sampling for efficient grid-based occupancy filtering in dynamic environments[C]//IEEE International Conference on Robotics and Automation. [S.l.]: IEEE, 2010: DOI: [10.1109/ROBOT.2010.5509931](https://doi.org/10.1109/ROBOT.2010.5509931).
- [9] MEYER-DELIUS D, HESS J, GRISETTI G, et al. Temporary maps for robust localization in semi-static environments[C]//IEEE International Conference on Intelligent Robots and Systems. [S.l.]: IEEE, 2010: DOI: [10.1109/IROS.2010.5648920](https://doi.org/10.1109/IROS.2010.5648920).
- [10] TIPALDI G D, MEYER-DELIUS D, BURGARD W. Lifelong localization in changing environments[J]. *The International Journal of Robotics Research*, 2013, 34: 1622-1678.
- [11] LAZARO M T, CAPOBIANCO R, GRISETTI G. Efficient long-term mapping in dynamic environments[C]//IEEE International Conference on Intelligent Robots and Systems. [S.l.]: IEEE, 2018: DOI: [10.1109/IROS.2018.8594310](https://doi.org/10.1109/IROS.2018.8594310).
- [12] BONIARDI F, TIM C, KÜMMERLE R. A pose graph-based localization system for long-term navigation in CAD floor plans[J]. *Robotics and Autonomous Systems*, 2019, 112: 84-97.
- [13] SUALEH M, KIM G. Dynamic multi-LiDAR based multiple object detection and tracking[J]. *Sensors*, 2019, 19: 1474.
- [14] VALENCIA R, SAARINEN J, ANDREASSON H. Localization in highly dynamic environments using dual-timescale NDT-MCL[C]//IEEE International Conference on Robotics and Automation. Hong Kong, China: IEEE, 2014: DOI: [10.1109/ICRA.2014.6907433](https://doi.org/10.1109/ICRA.2014.6907433).
- [15] GRISETTI G, KU M R, STACHNISS C, et al. A tutorial on graph-based SLAM[J]. *IEEE Intelligent Transportation Systems Magazine*, 2010, 2(4): 31-43.
- [16] OLSON E B. Real-time correlative scan matching[C]//IEEE International Conference on Robotics and Automation. Kobe, Japan: IEEE, 2009: DOI: [10.1109/ROBOT.2009.5152375](https://doi.org/10.1109/ROBOT.2009.5152375).
- [17] SERAFIN J, GRISETTI G. Using extended measurements and scene merging for efficient and robust point cloud registration[J]. *Robotics & Autonomous Systems*, 2017, 92: 91-106.
- [18] COLLINS T, COLLINS J J, RYAN C. Occupancy grid mapping: An empirical evaluation[C]//IEEE International Conference on Control & Automation. Athens, Greece: IEEE, 2007: 1-6.
- [19] KUMMERLE R, GRISETTI G, STRASDAT H, et al. G2o: A general framework for graph optimization[C]//IEEE International Conference on Robotics and Automation Shanghai, China: IEEE, 2011: DOI: [10.1109/ICRA.2011.5979949](https://doi.org/10.1109/ICRA.2011.5979949).