

DOI: 10.13718/j.cnki.xdzk.2020.06.015

基于生长四叉树结构的二维建图算法

姜 晗^{1,2}, 贺付亮^{1,2}, 王世元^{1,2}

1. 西南大学 电子信息工程学院, 重庆 400715; 2. 非线性电路与智能信息处理重庆市重点实验室, 重庆 400715

摘要: 目前, 视觉机器人的同时定位与地图构建(Simultaneous Localization and Mapping, SLAM)算法的建图部分主要采用了三维八叉树地图, 虽然其地图存储容量较大, 但是地图范围却无法实时扩大, 且室内场景中常见的动态事物也因为忽略了大噪声点而难以进行处理. 为此提出一种新的基于生长型四叉树结构的实时网格二维地图构建方法, 将三维体素地图降维为二维网格地图, 增加了对动态特征点轨迹的预测, 在无损失三维空间信息的情况下丰富了导航地图携带的环境信息. 真实室内场景的实验表明: 该算法能够在地图中较为精确地显示障碍物的位置信息, 显著地降低了地图存储空间, 提高了建图速度.

关键词: 同时定位与建图; 场景重建; 自主导航; 网格地图; 四叉树

中图分类号: TP242.6

文献标志码: A

文章编号: 1673-9868(2020)06-0128-12

在机器人感知领域中, 对视觉信息的关注程度越来越高, 其中同时定位与地图构建算法是该领域的重要方向. SLAM 能够在没有任何先验信息的情况下, 通过摄像机或激光雷达等传感器采集环境信息, 来估计自身的运动模型和构建所处的环境地图, 在智能飞行器、智能手机、机器人和 AR/VR 设备中, SLAM 是不可或缺的底层支持算法, 通过快速实时的运算和估计, 不仅能使设备与环境更加深入的交互, 更有利于顶层应用广泛而高效的开发. 所以, 科研技术人员一直在追求更加有效和快速的方法来实现 SLAM. 近年来, 人们在 SLAM 领域取得了重大进展, 例如, 在视觉 SLAM 方面(Visual SLAM), 最著名的 ORB-SLAM 就是一种基于特征点的 SLAM 系统^[1], ORB-SLAM 通过提取采集的图像的特征点进行帧间特征匹配, 进而采取非线性优化完成相机位姿的估计, 从而实现了自定位功能. 进一步, PL-SLAM^[2]在 ORB-SLAM 基础上引入了特征线, 使系统在低纹理场景下更具鲁棒性. 除此之外, 经典 SLAM 还包括基于扩展卡尔曼滤波的 MSCKF^[3], 基于单目相机的 MonoSLAM^[4], 与特征点不同的直接法匹配系统 LSD-SLAM^[5], 以及加入了 IMU 惯性传感器的融合算法 OKVIS^[6], ROVIO^[7], VIORB^[8] 等等. 这些 SLAM 算法主要优化了位姿和稀疏地图点, 因在优化过程中忽略了误差较大的地图点, 所以并不能形成真正用于导航和场景理解的地图. 因此, 从实际情况来看, 仅凭稀疏地图点以及机器人的行动轨迹并不能完整地构建导航地图, 也无法有效地进行机器人避障与自主探索. 在建图方面, 算法 RGBD-SLAM^[9]通过利用较为可靠的深度信息提高了传统迭代最近点算法(Iterative Closest Point, ICP)的精度和鲁棒性. 然而, 以上算法仍然忽略了误差较大的地图点, 而通过实践发现部分特征点来自场景里的动

收稿日期: 2019-11-01

基金项目: 国家自然科学基金项目(61671389); 中央高校基本科研业务费专项资金项目(XDJK2019B011, SWU118060).

作者简介: 姜 晗(1992-), 男, 硕士研究生, 主要从事计算机视觉 SLAM 与智能感知算法的研究.

通信作者: 王世元, 教授, 博士研究生导师.

态事物投影点, 因此这些无法识别的动态障碍物将会影响机器人的导航和避障^[10-12].

在 SLAM 系统的静态场景建图方面, 常采用简单的点云构图^[13], 即通过传感器模型 (Sensor Model, SM) 求得空间点云, 网格化后生成体素, 从而形成点云体素占有地图. 然而该算法消耗的计算资源非常大且不利于地图的维护, 尤其是在大规模场景重建的情况下效率非常低, 为了提高效率, 就需要计算各个网格的占有概率. Dia 和 Mottin 提出了一种应用于一维至三维的基于逆传感器模型的概率网格地图算法^[14], 这种算法为地图的实时更新提供了可能. 进一步, 在概率网格地图的基础上发展出了应用最为广泛的建图方法是 Hornung 提出的八叉树地图 OctoMap^[15], 利用八叉树数据结构作为基础, 压缩了点云地图的存储, 同时进行网格占用概率的估计, 与 WHELANT 等^[16]的点云建图方法相比大范围提高了建图的效率, 减少了存储空间消耗. 然而, 为了能够高效应用于计算能力较差或存储容量较低的终端, 作为底层支持算法的 SLAM 希望构建的是更加小巧便捷的地图. 对于树状数据结构而言, 比八叉树更精简的是四叉树, 四叉树的节点数量接近八叉树的二分之一. 而目前的研究中四叉树常见于激光雷达 SLAM 算法, 由于激光雷达进行的是平面二维扫描, 因此获得的场景信息极为有限, 为此, 本文重点研究了视觉 SLAM 中的四叉树建图算法.

在实际的室内环境中, 视野中的动态事物能够被传感器捕捉, 因此动态事物的运动方向以及运动状态会影响机器人的导航, 被称之为“交通管制问题”. 交通管制问题使机器人的定位与建图功能受到限制, 甚至可能导致无法完成可靠的避障、路径规划与自主探索. 为此, 文献[17]通过深度学习的方法实现了常见室内事物的语义识别, 并赋予其现实含义, 进而提出了基于语义识别的 SLAM 算法. 但是, 该方法的图像多目标识别的效果有限, 且运算复杂度较高, 因此, 较难应用于机器人的基础底层定位建图.

本文在 ORB-SLAM 系统上进行改进, 首先在补充点云数据进行稠密重建的基础上对三维点云进行完全的地面二维投影, 通过该降维运算, 构建携带三维信息的二维导航地图, 并利用四叉树结构进行储存, 相比于八叉树三维地图, 本文的二维建图方法能够高效实现更大规模的可靠建图. 在机器人的实际运行过程中, 对偏差较大的特征点创建动态矢量, 并对其利用改进的 K-means 算法进行分析判断, 从而识别出动态障碍物矢量; 在两帧之间推测出动态障碍物的行动方向和速度, 进而在概率表示的网格地图中完成标识, 为机器人实时自动导航提供动态障碍物的可靠信息. 最后, 采用安装了 RGBD 相机的室内移动机器人对本文算法进行实际测试, 实验结果表明: 本文的二维建图算法能够在动态室内场景中实现精准的自定位与环境地图重建, 所得的二维导航地图占用极小的空间, 在未使用深度学习的条件下能够提供丰富的环境语义信息, 实现机器人的自主导航和避障.

1 算法数学模型

本节介绍和推导本文采用的数学模型. 主要包括 ORB-SLAM 的建模方法和优化方法, 概率网格地图的构建模型, 地面提取的随机抽样一致 (Random Sample Consensus, RANSAC) 算法, 点云投影变换算法以及四叉树地图储存的相关算法, 利用这些数学模型, 构建基于四叉树的二维网格建图模型.

1.1 ORB-SLAM 的优化模型构建

ORB-SLAM 算法采用光束平差法 (Bundle Adjustment) 优化机器人定位信息. 首先建立像素点与空间点的相机的传感器模型:

$$zP_{u,v} = KTP_w \quad (1)$$

其中 z 指的是像素深度值, $p_{u,v}$ 指代图像像素坐标, K 为相机内参数矩阵, T 为位姿, P_w 为世界坐标.

将公式(1)表示成 $s = h(x, y)$, 这里观测量 s 即公式(1)中的 $p_{u,v}$, (x, y) 分别代表了位姿 T 和地图点 P_w . 引入传感器观测误差 $e = s - h(x, y)$, 即相机观测以及标定误差. ORB-SLAM 因采用非线性

图优化方法因此无需关注噪声类型. 在系统运行时, 图像随着时间不断加入系统, 累积的状态采用 $(x_1, x_2, \dots, x_m, y_1, y_2, \dots, y_n)$ 表示, 因每一帧包含大量观测的地图点, 因此 y 的数量远大于 x 的数量, 即 $m \gg n$. 建立代价函数:

$$\frac{1}{2} \sum_{i=1}^m \sum_{j=1}^n \| e_{i,j} \|^2 = \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^n \| s_{i,j} - h(x_i, y_j) \|^2 \quad (2)$$

对 x_i 和 y_i 进行一阶泰勒展开:

$$\frac{1}{2} \| f(x + \Delta x) \|^2 \approx \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^n \| e_{i,j} + F_{i,j} \Delta \epsilon_i + E_{i,j} \Delta \rho_j \|^2 \quad (3)$$

其中, $F_{i,j}$ 和 $E_{i,j}$ 为雅可比矩阵.

利用高斯牛顿法对上式进行优化. 在优化时, ORB-SLAM 对该代价函数的大误差项用 Huber 回归损失函数进行线性替换, 以消除大误差项对优化造成的影响, 即

$$H(e) = \begin{cases} \frac{1}{2} e^2 & |e| \leq \delta \\ \sigma \left(|e| - \frac{1}{2} \sigma \right) & |e| > \delta \end{cases} \quad (4)$$

其中, δ 为误差阈值. 当误差超过该阈值, 将采用 Huber 函数进行滤波. 从公式(4)可以看到, 将误差较大的项在公式(2)的累加方程中以一次函数替换, 减少了二范数增长过快对优化效果的影响. 而在公式(2)中大误差来源于位姿值和地图点位置的不匹配, 主要包括两帧图像之间特征点的误匹配和从动态障碍物提取了特征点两种情况. 对于机器人自定位来说, 这种误差点均需要消除, 但是对于动态场景的地图构建, 由于需要捕获动态障碍物特征点的移动方向, 大误差点却是非常重要的计算参量.

1.1.1 随机抽样一致算法

与 Huber 函数不同, 随机抽样一致算法^[18]在 ORB-SLAM 中主要用于误匹配消除. 该算法主要在数据集中选取子集并拟合函数, 即计算子集外的元素与子集拟合的函数的相关性, 若相关性高则子集拟合可以近似替代数据集拟合; 若相关性低则再次选取新的子集. 迭代上述过程直到选取的子集最能拟合数据集内大多数元素为止, 即可理解成为最符合整个样本空间的拟合函数.

1.2 空间网格地图的概率模型构建

机器人在进行自定位后, 将位姿与图像数据传入建图程序进行同步建图, 建图算法基于的是贝叶斯后验概率模型. 机器人处在室内静态环境中, 通过视觉传感器数学模型建立外部环境. 考虑到传感器存在噪声的影响, 此类不确定因素的概率分布模型称为概率传感器模型(Sensor Model, SM). 而在检测到外部信息的数据后, 通过外部数据建立地图和动态障碍物的模型称为逆传感器模型(Inverse Sensor Model, ISM)^[14]. 首先定义建立的地图网格之间相互独立. 假定 Ω 表示三维空间, 并定义为空间参考系(Spatial Reference), N 为网格的个数, i 为网格下标, c 为网格基础单元, m_i 为二元随机变量. m_i 表示了 c 的占有概率. $m_i \in (o_i, e_i)$, 即 o_i 和 e_i 分别表示占有和空白两种情况, 由于表示网格单元是否占有的随机变量具有互斥特性, 因此:

$$p(o_i) + p(e_i) = 1 \quad (5)$$

对于地图的概率分布可由全部网格单元概率累乘求出:

$$p(m) = \prod_i p(m_i) \quad (6)$$

利用传感器的输入数据求解的地图概率分布可由公式(6)展开:

$$p(m | s_{1:t}, x_{1:t}) = \prod_i p(m_i | s_{1:t}, x_{1:t}) \quad (7)$$

其中, s 是传感器的观测值, x 是运动模型数据(即机器人的位姿). 在给定传感器测量值 $s_{1:t}$ 和位姿数据 $x_{1:t}$ 时, 每个单元 m_i 的占有概率 o_i 可根据 t 时刻及之前的观测值和运动模型由马尔可夫特性和贝叶斯准则计算得出:

$$p(o_i | s_{1:t}, x_{1:t}) = \frac{p(o_i | s_t, x_t) p(s_t | x_t) p(o_i | s_{1:t-1}, x_{1:t-1})}{p(o_i) p((s_t | s_{1:t-1}, x_{1:t-1}))} \quad (8)$$

同理, 相同网格单元的非占有概率 $p(e_i | s_{1:t}, x_{1:t})$ 与公式(8)的形式一致. 根据公式(5)的关系将单元占有概率和非占有概率的比值进行计算, 可得:

$$\frac{p(o_i | s_{1:t}, x_{1:t})}{1 - p(o_i | s_{1:t}, x_{1:t})} = \frac{p(o_i | s_t, x_t)}{1 - p(o_i | s_t, x_t)} \frac{p(o_i | s_{1:t-1}, x_{1:t-1})}{1 - p(o_i | s_{1:t-1}, x_{1:t-1})} \frac{1 - p(o_i)}{p(o_i)} \quad (9)$$

其中, 等号左边的分母是由式(5)可得. 再利用对数变换公式将式(9)重写为

$$l(o_i | s_{1:t}, x_{1:t}) = l(o_i | s_t, x_t) + l(o_i | s_{1:t-1}, x_{1:t-1}) - l(o_i) \quad (10)$$

$$l(x) = \log \frac{p(x)}{1 - p(x)} \quad (11)$$

利用公式(10)在 SLAM 系统运行时迭代计算每一个网格单元的占有概率, 等式右边的第一项即为逆传感器概率模型, 第二项为迭代项, 第三项为传感器模型. 由于 RGB-D 相机采集的点云是投影于三维空间 $s_i \in \mathbb{R}^3$ 上的, 故需对其进行二维投影操作, 将三维空间点云投影至以地面为基准的二维地图上(本文假设室内环境的地面是理想平面). 首先本文设置了两个高度阈值下高度阈值 h_{down} 和上高度阈值 h_{up} . 对于机器人二维导航而言, 过高的障碍物可以忽略, 因此在 h_{up} 以上的点云在处理中将会剔除, 从而大量减少建图需要处理的地图点. 下高度阈值是为了消除地平面法向量提取误差而设置的缓冲区. 设 $g \in \mathbb{R}^2$ 为地面参考系, 表示为相对于初始相机位姿的变换矩阵 T_g . 设 p_g 为二维网格单元, 则二维地图点观测值投影为

$$p_{g_i} = f(T_g T_{t-1} m_i) \quad (12)$$

其中, f 函数表示从三维点投影到二维点映射. 由公式(12)可知, f 函数在投影过程中可能在二维网格地图上产生重合点, 这是由多个三维单元重叠投影到同一二维地图区域上引起的. 虽然投影得到的二维地图单元混合了传感器噪声, 但是, 本文不对此噪声进行处理, 而是在工程上采用点云的降采样滤波.

进一步, 对三维体素的概率进行投影生成二维网格占有概率:

$$m_{p_{g_i}} = f(l(m_i | s_{1:t}, x_{1:t})) \quad (13)$$

投影后即可生成二维网格的逆传感器占有概率模型. 通过将视觉三维传感器模型转化成二维传感器模型, 减少了大量三维体素的概率计算, 提高了效率.

1.3 地平面投影变换与地面坐标初始化

对于三维空间数据的二维投影需要基准平面, 以地面为基准, 而对于地平面提取与世界坐标系的构建在 SLAM 建图领域一直被忽视. 在地面机器人 SLAM 算法与稠密建图中, 以地平面的视野扩张方向作为相机位姿调整的基准更加适合二维网格建图. 目前, SLAM 算法中通常以相机的初始位姿作为世界参考系的原点和轴向. 本文首先计算地平面相机初始位姿与地平面的姿态矫正矩阵, 该矩阵代表了从相机初始位姿参考系到地面参考系的变换, 并作为二维建图的基准进行系统初始化. 设 T_g 为变换矩阵, P_{mark} 为初始化标志物, 该标志物由一条容易提取特征点的三角形构成, 平置于相机初始视野内的地面上, 利用一致性拟合算法 RANSAC 拟合地平面:

1) 从 RGBD 相机初始帧抓取的点云图中提取地面标志物点云.

2) 设定地面方程公式为 $Ax + By + Cz = D$ 和误差值, 随机抽选 3 个点进行平面构建, 计算其他点云是否在误差内满足该平面公式, 记录内点和外点个数.

3) 重复循环第二步, 直到选出大多数点满足误差内的平面公式, 取出该平面的 A, B, C 值形成地平

面法向量.

可见, RANSAC 算法的迭代次数较多, 因此, 该算法只在第一帧初始化或机器人视野中再次出现 P_{mark} 的情况下求解变换矩阵, 避免大量的迭代计算.

根据地平面法向量计算原相机坐标系(世界坐标系)到地平面坐标系的变换矩阵 T_g :

$$T_{t=1, [0, 0, 1]} = T_{ABC} q_x q_y \quad (14)$$

$$T_g = q_x q_y \quad (15)$$

其中, 四元数 q_x 和 q_y 表示旋转. 将世界坐标系的 z 轴调整至与地平面法向量一致, 则无需进行平移和 x 与 y 轴的变换. 因此, 通过公式(13)和公式(14)即可得出 T_g , 该地平面变换矩阵可将世界坐标系中的地图点转化成为地面坐标系的地图点.

1.4 生长型四叉树数据结构

最后, 对投影产生的二维地图数据采用四叉树数据结构保存, 可提高地图的存储和搜索的效率. 文献[15]的八叉树结构是针对三维空间数据构建的, 所需的节点数是四叉树的两倍. 二维地图由于将三维空间点投影至二维平面点, 且对投影重叠部分进行了数据降采样滤波, 减少了大量的无关地图点, 因此, 更适用于场景的大规模建图. 四叉树结构如图 1 所示. 从图 1 可知对于 64 单元的网格地图存储仅需要 14 个节点, 与点云数据和八叉树相比节省了大量的储存空间.

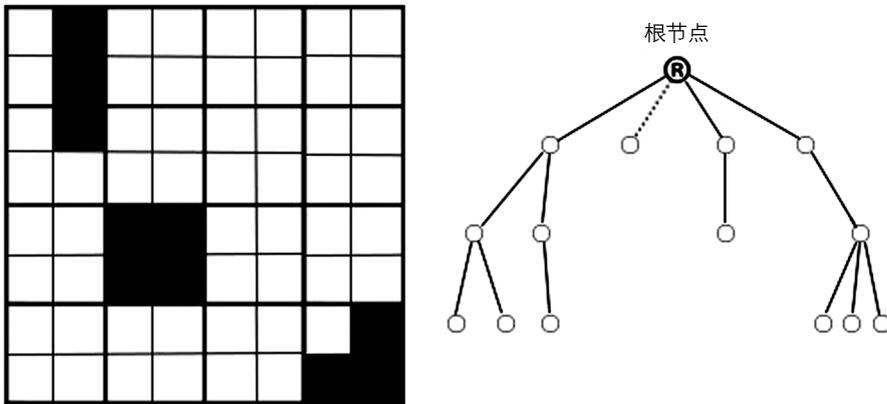


图 1 四叉树结构

本文对地图的四叉树存储采用动态方法, 如图 2, 每一帧都以四叉树形式保存, 当有新的图像帧加入系统, 会生长出新的四叉树. 当 4 个四叉树数据结构匹配后会整合成新的四叉树, 更新根节点和叶子节点, 其中保存的网格占有概率会实时更新. 设定网格分辨率, 以占据网格的初始概率为 0.9, 空闲网格的初始概率为 0.1, 在初始帧情况下建立视野区域并按照网格分辨率分割. 根据相机初始位姿加首帧二维投影观测值, 计算每个网格是否占有, 占有则概率值为 0.9, 否则概率值为 0.1. 进一步, 利用式(10)和式(13)迭代更新其占有、空闲概率值, 如果占有概率值达到 0.9 则视为占有网格, 若空闲概率低于 0.1 则设为空闲概率, 未更新的区域占有概率设为 0.5, 表示其网格的状态为未知情况. 占有概率的运算是将对数值的对数变换进行运算, 保存在节点中的值也是对数值, 将节点值进行反对数变换后即可得到真正的占有概率. 通过对数变换, 可将乘除运算转化为加减运算. 公式(16)为 t 时刻的地图概率状态的对数更新.

$$l(t) = l(t-1) + \log(P_e) - l(0) = \log 9 + \dots \log 9 - \log 4 - \dots \log 4 \quad (16)$$

其中, $\log 9$ 是观测到占有情况加上的量, $\log 4$ 是观测到空闲情况减去的量, 通过对之前的概率进行更新, 即可迭代计算每个网格在观测情况下的概率值.

建立生长型四叉树数据结构对二维网格地图进行储存, 系统每加入一帧图像关键帧, 则更新现有四叉树叶子的节点资料, 并根据新加入的地图点方向构建与上一帧独立的新的四叉树数据结构. 创建栈数

据结构, 其中存在一个初始二叉树, 将新的二叉树压栈, 如图 2 所示, 当匹配到 4 个相关树时将该栈释放, 并设置根节点形成新的二叉树同时更新所有叶节点资料. 将新的二叉树压入新栈, 树的深度加 1. 当形成更深的二叉树时意味着二维地图在生长扩大.

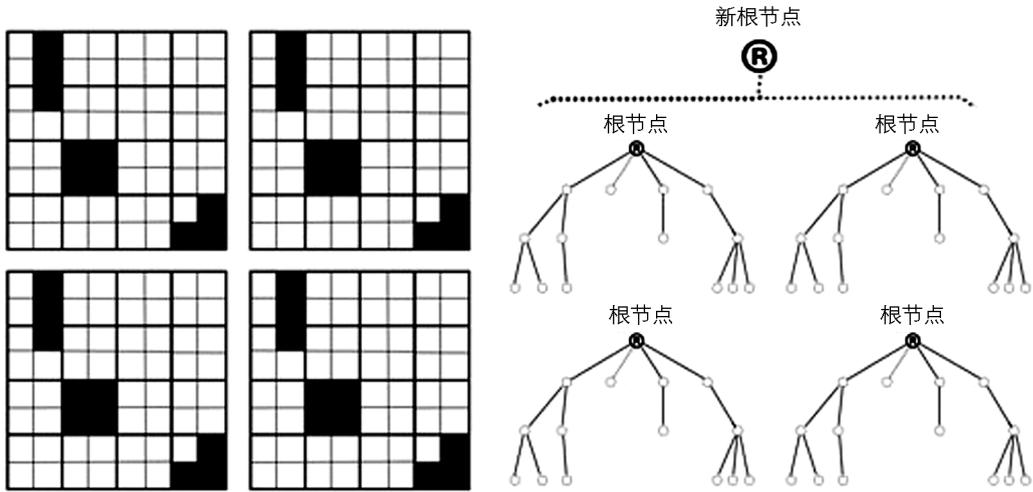


图 2 生长二叉树结构

2 动态场景建图算法

2.1 动态障碍物识别

ORB-SLAM 系统在特征点提取方面划分了视野网格, 将采集的图像设置成 64×48 个区域, 目的在于保证从视野中均匀地提取特征点, 这使得视野中存在动态障碍物时, 容易获取动态障碍物区域的 ORB 特征点. 而在 ORB-SLAM 系统中, 大误差点会被 Huber 函数降幂替换, 除了 Huber 函数, 应用在单目单应矩阵计算的 RANSAC 算法也会将误差点进行剔除, 而被剔除的大误差特征点在实践中有一部分是从动态障碍物物体上提取的.

此外, 大误差点除了来自动态障碍物之外, 还有一个重要的来源是两帧图像的特征点误匹配. Sift 特征提取算法^[19]的特征信息丰富, 对于旋转、视角变换、光线变化等都具有较好的匹配效果, 是目前最准确的特征提取算法. 然而由于 Sift 算法计算量大且计算要求较高, 因此难以在计算能力有限的设备上完成实时运算. 基于此, 本文的方法不改变 ORB-SLAM 原有的提取模式, 首先对可能出现的所有误匹配点进行其动态矢量的计算, 接着重点对动态矢量进行处理, 动态矢量的公式如下:

$$Vec_{obstacle}^i = P_w^{t+1} - P_w^t \quad (17)$$

其中, Vec 为动态障碍物运动方向, i 为动态障碍物索引, P_w 表示特征点的世界坐标, t 为时间帧索引.

本文利用了改进后的 K-means 进行动态障碍物矢量的提取, 剔除误匹配点. 从几何上进行分析, 如图 3 所示, 同一障碍物上的多个特征点同时误匹配的概率低于单一特征点的误匹配概率. 因此将公式 (17) 求出的大量动态矢量作为输入, 利用矢量的相似性指标, 通过聚类算法求解动态障碍物矢量集合.

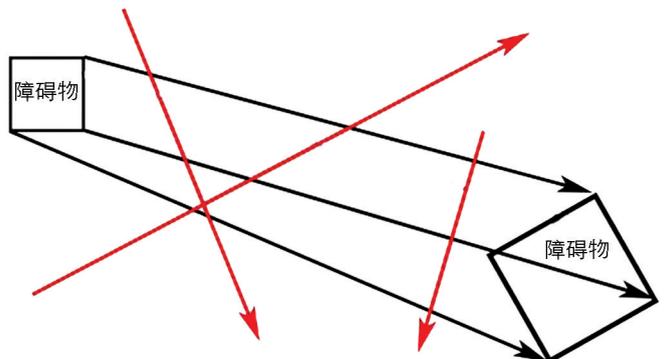


图 3 动态向量示意图

图 3 中的左右方格是同一个障碍物的两

帧位姿,反映到障碍物的特征点矢量即为图中黑色的向量,相对的特征点误匹配矢量则为红色向量.本文利用了 K-means 聚类算法对动态矢量进行聚类提取出动态障碍物矢量,并结合相机对应的位姿数据,求得动态障碍物的速度和移动方向,这里的速度是根据网格地图分辨率来计算的网格速度,即为曼哈顿距离与帧间时差的比值.另需要说明矢量的相似性聚类是通过两步进行的,第一步是聚类矢量的起始点,利用选取欧氏距离作为相似度指标的 K-means 聚类算法将起始点相近的矢量聚成一类;第二步则对第一步中聚类而来的集合再进行以矢量投影作为相似度指标的聚类,这需将 K-means 算法的迭代公式改为

$$SSE = \sum_{k=1}^k \sum_{i=1}^n \left(\frac{|a_i|^2 \cos\theta}{a_i \cdot b_k} - b_k \right)^2 \quad (18)$$

根据公式(18)将矢量聚类:随机选取向量 b_k ,计算剩余向量 a_i 与选取向量之间的投影,再求投影值与选取向量之间的差值.考虑到矢量投影可能大于被投影矢量,因此补充幂运算求该差值的平方.经迭代,从起始点聚类的集合中得到动态障碍物矢量的聚类结果,并计算其平均值,此即为动态障碍物的移动矢量,该移动矢量将被用于导航算法.

2.2 二维网格地图的动态障碍物概率叠加

由于在系统运行中,动态障碍物的信息变动较大,变动速度较快,因此不适合直接对生长四叉树的节点进行添加删除和更新操作.为了避免频繁搜索和读写四叉树节点,新建数组结构,专门存储动态障碍物信息,该数组集合通过索引与四叉树网格地图建立联系,将障碍物矢量起点和终点的位置经过地面坐标系变换以及二维网格地图投影后存入障碍物数组,该矢量信息可用于机器人的自主导航的动态避障工作^[20-21].

最后,将基于生长四叉树结构的二维建图算法流程图总结如图 4 所示.

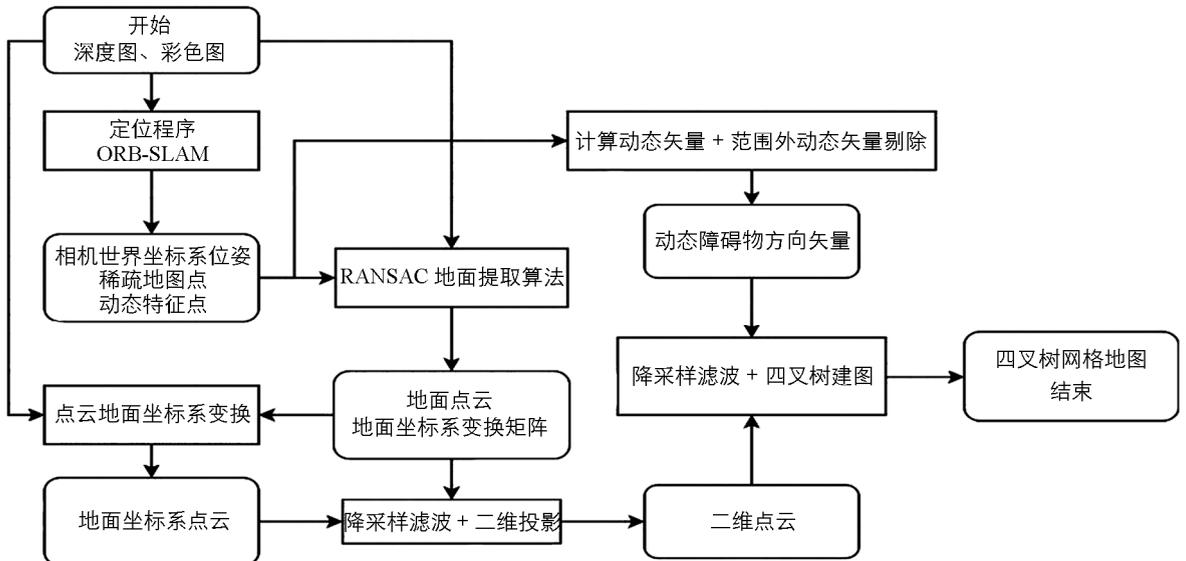


图 4 基于生长四叉树结构的二维建图算法流程图

3 实验结果和分析

建图实验利用了搭载 Kinect.v1 深度摄像机(RGB-D Camera)的小强机器人.小强机器人集成了运动模块和驱动程序,可以进行自主的地图探索,如图 5 所示.深度摄像机由彩色摄像机和一个深度传感器构成,可以发射红外结构光,用于深度信息的测量.代码基于 ORB-SLAMv2 的 RGB-D 版本^[1]开发,无需进

行对极几何和极线搜索的运算.

为了比较本文二维投影二叉树建图与二叉树建图的效率和存储空间消耗, 设置一项对比试验, 选取 nyud 数据集 dining_room 序列中的 5 帧图像, 对其进行二叉树三维重建和二叉树二维重建, 对比两者的运行时间、树结构节点数和存储空间消耗数据, 从而比较二者算法的性能优劣. 对不同分辨率选取进行了实验, 讨论分辨率选择对建图效果的影响. 对地面提取进行误差实验, 讨论地面法向量提取误差对下高度阈值选择的影响.

最后应用小强机器人在实际室内场景中运行 ORB-SLAM2 和三维稠密点云的二维投影二叉树建图程序, 验证本文方法的实际运行效果.



图 5 小强机器人

3.1 二叉树建图与投影二叉树建图

图 6(a)显示了 ORB-SLAM 在室内环境下的稀疏建图, 从该图中可知稀疏地图点信息量较少, 不能获得准确的障碍物信息, 因此无法利用稀疏点地图进行有效导航. 图 6(b)显示了环境的稠密三维点云图, 从该图可知该地图缺乏地面提取的信息. 因此, 将图 6(b)经过地面坐标变换投影后生成二维点云图, 如图 6(c)所示. 由图 6(c)可知, 二维投影点云图能够较好地提取真实场景中障碍物的信息和机器人导航中所需的地面信息, 因此, 基于二维点云图生成的地图能够有效地应用于机器人导航.

图 7 所示为体素地图和网格地图. 网格地图相较于体素地图缺少了高度维, 但是在实际 SLAM 程序中, 可以通过设置阈值来使一定高度范围内的障碍物投影, 因此即使地图缺少高度维数据, 二维网格地图对于地面机器人导航仍然具备理论上的可靠性.

表 1 显示了二叉树与四叉树在不同分辨率下的速度与容量对比. 分辨率越高的意味着在 SLAM 系统中建图范围越广, 地图容量就会越大. 从表 1 可以看出四叉树建图的地图容量远小于二叉树地图, 建图速度也同样远高于二叉树建图. 表 2 显示了不同场景下 3 种地图模式的性能比较, 从表中可知四叉树投影地图的性能要优于二叉树地图, 树的深度一致是由于树深度取决于地图范围, 而四叉树的节点数是二叉树的一半. 对于室内地面机器人而言, 本文的三维视觉投影二维四叉树地图更加适用于视觉 SLAM 系统.

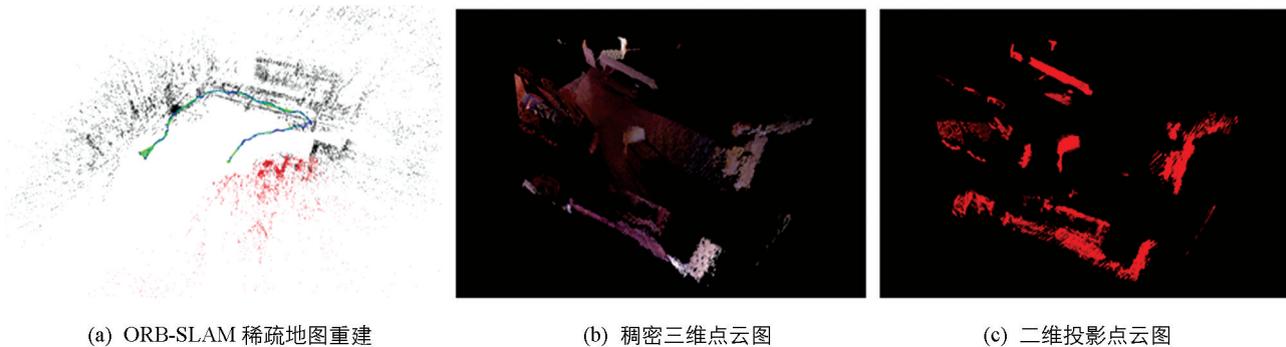


图 6 稀疏建图、三维地图和二维投影点云图对比

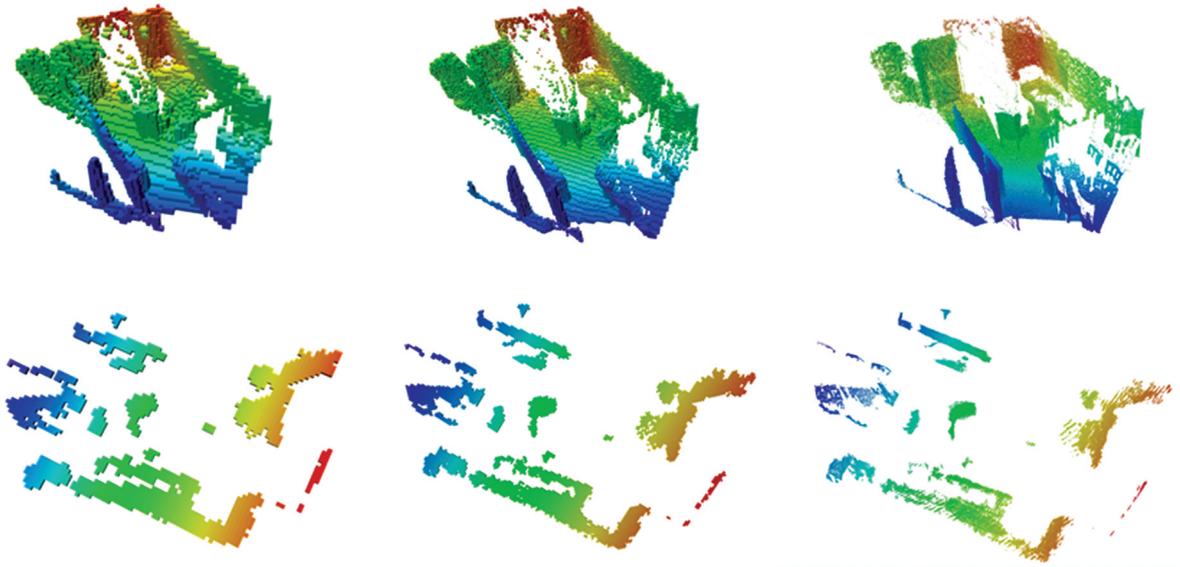


图 7 体素地图和网格地图

表 1 八叉树与四叉树不同分辨率的性能

分辨率	0.1	0.05	0.01
建图时间/ms	3.569 36	45.176 2	3166.17
	0.500 523	1.036 26	63.461 9
存储容量/KB	11.9	56.5	2018.7
	2.8	9.9	219.4
建图效果			

注：单元格中上方数据为八叉树建图，下方为四叉树建图。

表 2 相同场景下 3 种地图模式的性能比较

	处理点云	叶节点/深度	地图容量/KB	建图时间/ms
点云地图	448 175	448 175/1	7 372.8	—
八叉树压缩	448 175	24 585/16	56.5	45.176 2
四叉树投影	49 060	13 515/16	9.9	1.036 26

注：分辨率=0.05，基础点云数据均经过降采样滤波处理；点云地图不是树结构，所以节点数为点云数，深度为 1。

在投影过程中，上高度阈值是投影障碍物的上限，在阈值以上的障碍物假定为对自主导航无影响障碍物，因此不对其进行投影。下高度阈值是对地面提取误差的修正，在地面提取有误差的情况下，下高度阈值太低会造成将部分地面也投影成障碍物，影响机器人的导航判断，高度太高则会忽略实际障碍物的投影，法向量对高度阈值的影响如图 8 所示，这里假设场景地面范围为 300，法向量偏差值为 $0-\pi/4$ ($0-0.79$)，下高度阈值根据偏差来进行设置，当偏差过大，需要重新验证地面的提取是否可靠，偏差较

大需要对法向量提取过程进行噪声滤波。

图 9 显示了小强机器人结合 ORB-SLAM 和三维重建程序的实际室内场景运行效果, 建图采用了 ROS 的 Rviz 三维可视化平台^[22]显示。建图程序将视野中提取的三维障碍物投影至二维网格地图建立了可导航地图。白色代表了可行区域, 黑色代表了障碍物, 灰色是未知网格单元。从图(9)左中可见, 本文提出的二叉树建图算法具有较高场景环境的重建精度, 但由于缺乏对二叉树概率值进行动态调整, 导致概率地图的误差累积, 因此, 图像中间和右边仍然存在少部分的错误障碍物投影。

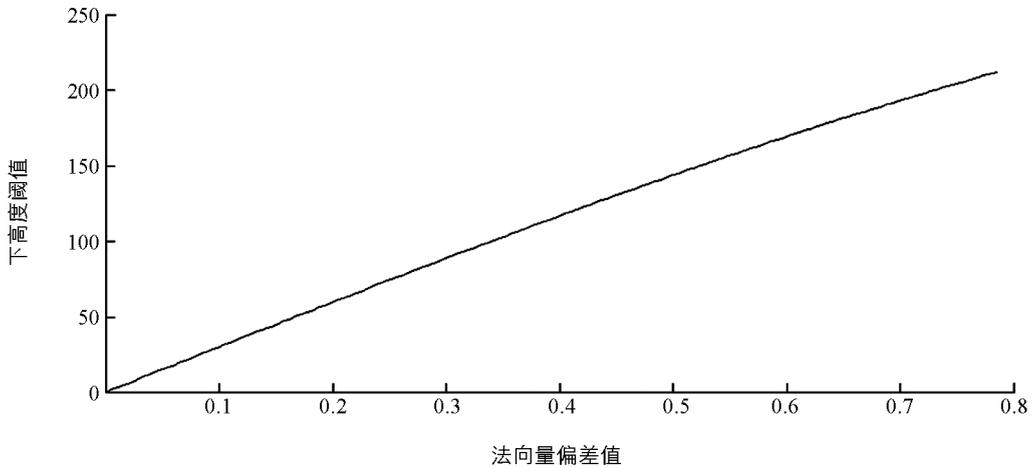


图 8 地面法向量噪声对下高度阈值选择的影响

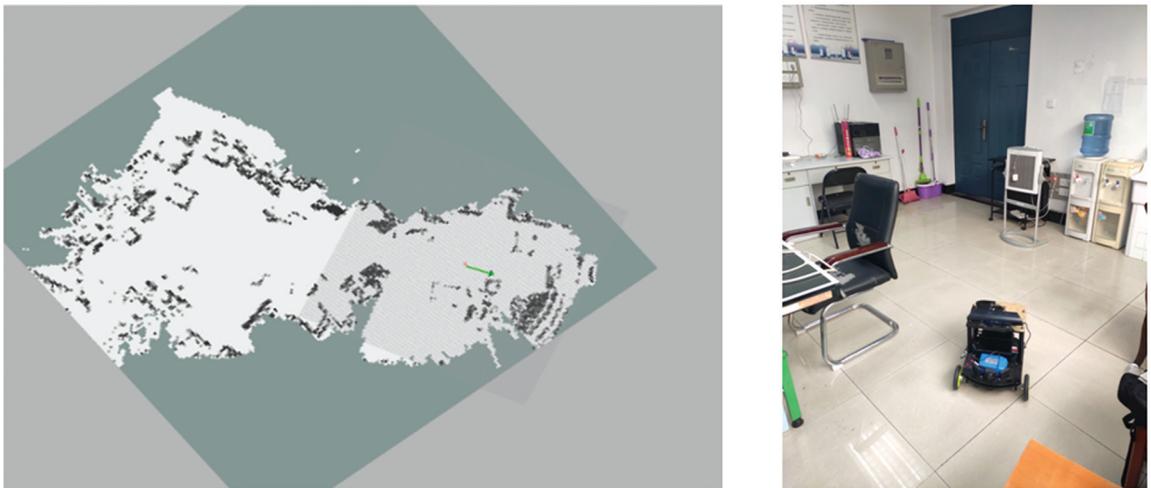


图 9 室内真实场景三维投影二维网格地图重建实验效果

4 结 论

针对传统 SLAM 无法建立高效可导航地图的缺点, 本文在 ORB-SLAM 和二叉树建图的基础上, 建立了基于二叉树的二维投影网格地图算法, 与二叉树地图和点云地图相比, 二叉树建图算法大量降低了地图存储空间, 提高了建图速度, 适合地图的大规模重建。实验结果表明: 基于二叉树的二维网格建图算法能够高效地应用于室内地面机器人的场景地图重建和自主导航, 由于本文算法有效地处理了动态障碍物, 因此能够为机器人自主导航提供更加丰富的语义信息, 具有较强的实用性, 能够实时生成用于机器人自主导航和路径规划的地图。

参考文献:

- [1] MUR-ARTAL R, MONTIEL J M M, TARDOS J D. ORB-SLAM: A Versatile and Accurate Monocular SLAM System [J]. *IEEE Transactions on Robotics*, 2015, 31(5): 1147-1163.
- [2] GOMEZ-OJEDA R, ZUNIGA-NOEL D, SCARAMUZZA D, et al. PL-SLAM: a Stereo SLAM System Through the Combination of Points and Line Segments [J]. *IEEE Transactions on Robotics*, 2019, 35(3): 734-746.
- [3] HEO, SEJONG, PARK C G. Consistent EKF-Based Visual-Inertial Odometry on Matrix Lie Group [J]. *IEEE Sensors Journal*, 2018, 18(9): 3780-3788.
- [4] ANDREW J D, IAN D R, NICHOLAS D M. MonoSLAM: Real-Time Single Camera SLAM [J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2007, 29(6): 1052-1067.
- [5] ENGEL J, SCHOPS T, CREMERS D. LSD-SLAM: Large-Scale Direct Monocular SLAM [M] // *Computer Vision-ECV 2014*. Cham: Springer International Publishing, 2014: 834-849.
- [6] LEUTENEGGER S, LYNEN S, BOSSE M, et al. Keyframe-Based Visual-Inertial Odometry Using Nonlinear Optimization [J]. *The International Journal of Robotics Research*, 2015, 34(3): 314-334.
- [7] BLOESCH M, OMARY S, HUTTER M, et al. Robust Visual Inertial Odometry Using a Direct EKF-Based Approach [C] // *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, September 28-October 2, 2015. Hamburg, Germany. IEEE, 2015: 298-304.
- [8] FORSTER C, CARLONE L, DELLAERT F, et al. On-Manifold Preintegration for Real-Time Visual-Inertial Odometry [J]. *IEEE Transactions on Robotics*, 2017, 33(1): 1-21.
- [9] ENDRES F, HESS J, STURM J, et al. 3-D Mapping with an RGB-D Camera [J]. *IEEE Transactions on Robotics*, 2014, 30(1): 177-187.
- [10] 余成波, 孙梦娜, 杨如民. 运用 iBeacon 技术的室内导航系统的设计与实现 [J]. *重庆理工大学学报(自然科学版)*, 2018, 32(5): 162-168.
- [11] 盛昫瑶, 张福泉, 任 艳. 应用 MapReduce 与视觉描述符的图像检索算法 [J]. *重庆理工大学学报(自然科学版)*, 2018, 32(12): 149-156.
- [12] 李 波, 易 洁. 基于时间的多机器人协调避碰算法研究 [J]. *重庆理工大学学报(自然科学版)*, 2019, 33(3): 91-97.
- [13] LEVINSON J, MONTEMERLO M, THRUN S. Map-Based Precision Vehicle Localization in Urban Environments [C] // *Robotics; Science and Systems III*, Georgia Institute of Technology, Atlanta, Georgia, USA. DBLP, 2007, June 27-30.
- [14] DIA R, MOTTIN J, RAKOTOVAO T, et al. Evaluation of Occupancy Grid Resolution through a Novel Approach for Inverse Sensor Modeling [J]. *IFAC-PapersOnLine*, 2017, 50(1): 13841-13847.
- [15] HORNING A, WURM K M, BENNEWITZ M, et al. OctoMap: An Efficient Probabilistic 3D Mapping Framework Based on Octrees [J]. *Autonomous Robots Journal*, 2013, 34(3): 189-206.
- [16] WHELAN T, SALAS-MORENO R F, GLOCKER B, et al. Elastic Fusion [J]. *International Journal of Robotics Research*, 2016, 35(14): 1697-1716.
- [17] YU C, LIU Z X, LIU X J, et al. DS-SLAM: A Semantic Visual SLAM towards Dynamic Environments [C] // *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, October 1-5, 2018. Madrid. IEEE, 2018: 1-5.
- [18] MATAS J, CHUM O. Randomized RANSAC with Td, d test [J]. *Image and Vision Computing*, 2004, 22(10): 837-842.
- [19] LOWE D G. Object Recognition from Local Scale-Invariant Features [C] // *Proceedings of the Seventh IEEE International Conference on Computer Vision*, September 20-27, 1999. Kerkyra, Greece. IEEE, 1999.
- [20] 周 亮, 李 静, 杨 飞. 基于跟踪—关联模块的多目标跟踪方法研究 [J]. *西南大学学报(自然科学版)*, 2018, 40(4): 151-161.

- [21] 侯荣波, 魏 武, 黄 婷, 邓超锋. 基于 ORB-SLAM 的室内机器人定位和三维稠密地图构建 [J]. 计算机应用, 2017, 37(5): 1439-1444.
- [22] MARTINEZA, FERNANDEZE. Learning ROS for Robotics Programming [M]. Birmingham: Packt Publishing Ltd, 2013: 45-48.

Two-Dimensional Mapping Algorithm Based on Growing Quadtree Structure

JIANG Han^{1,2}, HE Fu-liang^{1,2}, WANG Shi-yuan^{1,2}

1. School of Electronic Information Engineering, Southwest University, Chongqing 400715, China;

2. Chongqing Key Laboratory of Nonlinear Circuits and Intelligent Information Processing, Chongqing 400715, China

Abstract: The mapping part in the simultaneous localization and mapping (SLAM) algorithm for visual robot positioning and environmental map construction mainly uses the three-dimensional octree map with a large map storage capacity. As a result, the range of mapping cannot be extended adaptively in real time. In the positioning part, it is difficult to deal with the common dynamic things in the indoor scene because of neglecting the large noise points. To address these problems, a novel real-time grid two-dimensional map construction method based on the growth quadtree structure is proposed in this paper to reduce the dimension of the three-dimensional voxel map to the two-dimensional grid map. The prediction of the dynamic feature point trajectory is improved by the proposed method, and the environmental information carried by navigation map is enriched without loss of the three-dimensional spatial information. An experiment in the real indoor scene shows that the proposed algorithm can display the position information of obstacles accurately in the map, greatly reduce the map storage space and improve the drawing speed.

Key words: simultaneous localization and mapping (SLAM); scene re-construction; autonomous navigation; grid map; quadtree

责任编辑 汤振金