

室外半静态环境下基于快速会话对齐的地图更新方法

方宝富^{1,2}, 谢 艺^{1,2}, 王 浩^{1,2}, 袁晓辉³

(1. 合肥工业大学计算机与信息学院, 安徽 合肥 230009; 2. 大数据知识工程教育部重点实验室(合肥工业大学), 安徽 合肥 230009;
3. 北得克萨斯州大学计算机系, 丹顿 76201, 美国)

摘 要: 目前室外半静态环境下的地图更新算法普遍存在速度较慢, 无法实时运行的问题。为此, 本文提出一种基于会话快速对齐的地图更新方法, 目的是提高会话对齐的速度, 保证地图与环境的一致性。首先引入高斯曲率来稀疏化点云以减小点云规模, 并进行体素化点云配准, 利用所获取的约束构造因子图; 然后, 引入历史约束实现局部会话的因子图修补, 解决因回环检测和点云配准失效导致的因子图构造失败问题; 最后, 通过因子图优化实现会话对齐, 并基于对齐的会话进行地图更新。在 MulRan 数据集和 LT-ParkingLot 数据集上评估了该方法, 其会话对齐频率达到了 13 Hz, 相较于原方法提升了 80%, 可实现在工厂等经典半静态场景下实时对齐会话。通过实验证明了本文方法在地图更新方面的有效性。

关键词: 地图更新; 长期 SLAM (同步定位与地图构建); 会话对齐; 点云配准; 因子图修补

Map Updating Method Based on Fast Session Alignment in Outdoor Semi-static Environment

FANG Baofu^{1,2}, XIE Yi^{1,2}, WANG Hao^{1,2}, YUAN Xiaohui³

(1. School of Computer Science and Information Engineering, Hefei University of Technology, Hefei 230009, China;
2. Key Laboratory of Knowledge Engineering with Big Data (Hefei University of Technology), Ministry of Education, Hefei 230009, China;
3. Computer Science Department, University of North Texas, Denton 76201, USA)

Abstract: Map updating algorithms in outdoor semi-static environments generally suffer from slow speeds and inability to run in real time. Therefore, a map updating method based on fast session alignment is proposed to improve the session alignment speed and ensure consistency between the map and the environment. Firstly, Gaussian curvature is introduced to sparsify point clouds and reduce their size, followed by voxel based point cloud registration, and the obtained constraints are used to construct factor maps. Then, historical constraints are introduced to implement factor graph repair for local sessions, solving the problem of factor graph construction failure caused by loop detection and point cloud registration failure. Finally, session alignment is achieved through factor graph optimization, and map updating is performed based on the aligned sessions. The method is evaluated on MulRan dataset and LT-ParkingLot dataset, and its session alignment frequency reaches 13 Hz, which is 80% higher than the original method, and sufficient for real-time session alignment in classic semi-static scenarios such as factories. The effectiveness of the proposed method in map updating is demonstrated through experiments.

Keywords: map updating; long-term SLAM (simultaneous localization and mapping); session alignment; point cloud registration; factor diagram repair

迄今为止, 自主移动机器人在工业、服务等领域的应用越来越广泛^[1]。在这些应用场景中, 需要依赖同步定位与地图构建 (SLAM) 技术^[2] 构造的地图以支持机器人的导航和定位。经典的 SLAM 算法^[3-5] 基于“场景是静态的”假设构建地图; 近期研究提出的算法^[6-8] 考虑剔除环境中的动态物体, 基于检测连续帧的场景变化进行工作。在半静态环境^[9] 中, 物体并非是完全固定的, 而是缓慢移动或变化的, 如工厂中堆放的物料箱、停车场中停放的车辆和由于季节变化消失或新增的树木等。这些

物体在首次建图时期呈现为静态, 但在一段时间后由于各种因素发生了变化, 它们被称之为“半静态物体”。上述方法无法应对这种情况, 导致构建的地图与环境不一致。避障算法可以在环境发生微小变化时正常工作, 但当静态地图与当前环境存在差异时, 导航性能会下降。若要保证机器人在现实环境中长期运行, 则必须保持静态地图和当前环境一致。目前的相关算法^[10] 主要针对室内半静态环境; 由于室外环境规模较大、噪声较多, 因此在室外运行时需要处理的数据量更大, 现有的算法^[11] 无法

满足实时性要求。

本文提出一种基于快速会话对齐的地图更新方法。所谓“会话”是指对真实场景所构建的地图，即“建图会话”。本文中，“建图会话”简称为“会话”。如果能在两个不同时间段的会话中对同一区域进行对齐，则具备进行变化检测和地图更新的前提条件。然而，在实际运行过程中，由于两次建图的起始位姿和整体轨迹是随机的，导致构建出的会话未对齐，因此需要设计一种方法将两个会话对齐到同一全局坐标系下。本文通过因子图优化实现两个会话对齐，其核心在于计算点云间的约束，本文通过 SVGICP (sparse voxelization generalized iterative closest point) 来实现。

1 相关工作 (Related works)

地图更新算法包括 2 个过程：会话对齐、变化检测和更新。会话对齐模块中点云配准过程是影响对齐速度的决定性因素。

1.1 点云配准

3D 点云配准技术是许多 3 维激光雷达应用的关键，如定位、映射和环境识别等。目前有 2 种流行的点云配准方法：广义迭代最近点 (GICP) 算法^[12]和正态分布变换 (NDT) 算法^[13-14]。GICP 通过 2 个分布之间的距离，扩展了经典的 ICP 算法，从而实现更加准确的配准。而 NDT 利用体素化方法来避免代价高昂的最近邻搜索以提高处理速度。然而，这两种方法都有自己的弱点。其中 GICP 高度依赖于最近邻搜索，如果点的数量很大，有时会很难在计算能力有限的计算机上实时运行。而 NDT 通常对体素分辨率的选择十分敏感，最佳体素分辨率取决于环境和传感器属性，如果没有选择适当的分辨率，NDT 的配准精度会急剧下降。Koide 等^[15]结合上述两种方法提出 VGICP 算法，通过计算点到体素分布的距离完成点云配准，在保证精确度的同时大大提高了配准速度。

1.2 地图更新

在 SLAM 领域，由于栅格地图在室内环境的广泛应用，一些学者将研究重点放在更新占用网格地图上。一方面，通过检测网格占用概率进行地图更新。如 Stefanini 等^[16]提出了一种占用网格更新算法，假设已有环境地图，通过激光雷达检测网格占用概率来更新地图。然而，该方法严重依赖于先前构建地图中机器人准确的初始位姿信息，这些信息由于地图变化很难立即估计，从而会影响占用网格的映射结果；之后该作者又提出了集成局部地图恢

复算法用于解决上述问题^[17]，避免地图退化；另一方面，为保证所构建的地图与实时变化的真实环境保持一致，令机器人不断地动态更新地图是解决问题的有效方法，即增量式建图^[18]。目前，基于图优化的 SLAM 算法^[19-20]已经取得了显著的进展。因子图优化技术是一种用于解决大规模稀疏非线性优化问题的方法，它将 SLAM 中的状态变量和约束表达为图中的节点和边，然后通过图优化技术来优化机器人的轨迹。然而，这类方法存在两个问题：一是在机器人运行过程中节点的不断添加导致优化问题难以处理；二是不断增多的节点数量提高了回环检测环节搜索候选者的维度，同时增加了误判的可能性。因此，许多算法通过删除冗余节点或限制图优化规模来解决上述问题。

Walcott-Bryant 等^[21]提出动态位姿图 (DPG-SLAM) 算法。该算法采用活动图表示环境的当前状态，活动图包括环境中保持不变的部分和新增的部分，但其维护的活动图会随着运行过程不断扩增，无法实现实时优化。Johannsson 等^[22]提出的 SLAM 系统通过简化图来实现系统长期运行。该算法只有在新环境中才会添加新节点，当环境发生变化时将新的节点信息合并到现有图结构中。然而，算法不删除旧节点，没有处理冗余信息。Zhao 等^[23]提出了一种基于 Chow-Liu 树 (CLT)^[24]对子图进行稀疏化的算法。该算法针对冗余信息进行边缘化操作，尽可能保留位姿图信息，降低了计算复杂度，可在环境规模一定的情况下保持图复杂度稳定。将上述方法应用于室内场景时，所构建的栅格地图无法完成室外环境下 3D 点云地图的变化检测和更新。Kim 等^[11]提出一个模块化框架 LT-mapper。该框架基于 LIO-SAM 算法^[25]实现，通过对齐中心会话和查询会话实现 3D 地图更新，并且对大规模场景具有鲁棒性。然而，其在会话对齐模块使用传统的 ICP 方法，无法满足实时性要求。

综上，本文提出一种基于稀疏体素化点云配准的会话对齐方法，使对齐速度得到显著提升。该方法旨在确保会话对齐模块能够实时运行，并保证地图更新的精确性。

2 方法 (Method)

本文方法通过稀疏体素化点云配准实现快速会话对齐，再使用历史约束对未能完成构造的因子图进行修补，最后通过优化因子图对齐两个会话，并进一步实现地图更新，系统框架如图 1 所示。在基于 SVGICP 的因子图构造模块中，针对每一个回环

对中的点云，首先基于高斯曲率对其进行稀疏化，降低点云规模，再通过稀疏体素化点云配准获取回环对之间的约束，并将其用于会话间因子图构造；在因子图修补模块中，调用保存的历史约束，结合

各会话内的约束信息，修补因子图；在会话对齐模块中，通过优化上述模块构造的因子图得到查询子会话向中心会话对齐的位姿；最后将对齐的会话用于地图更新。

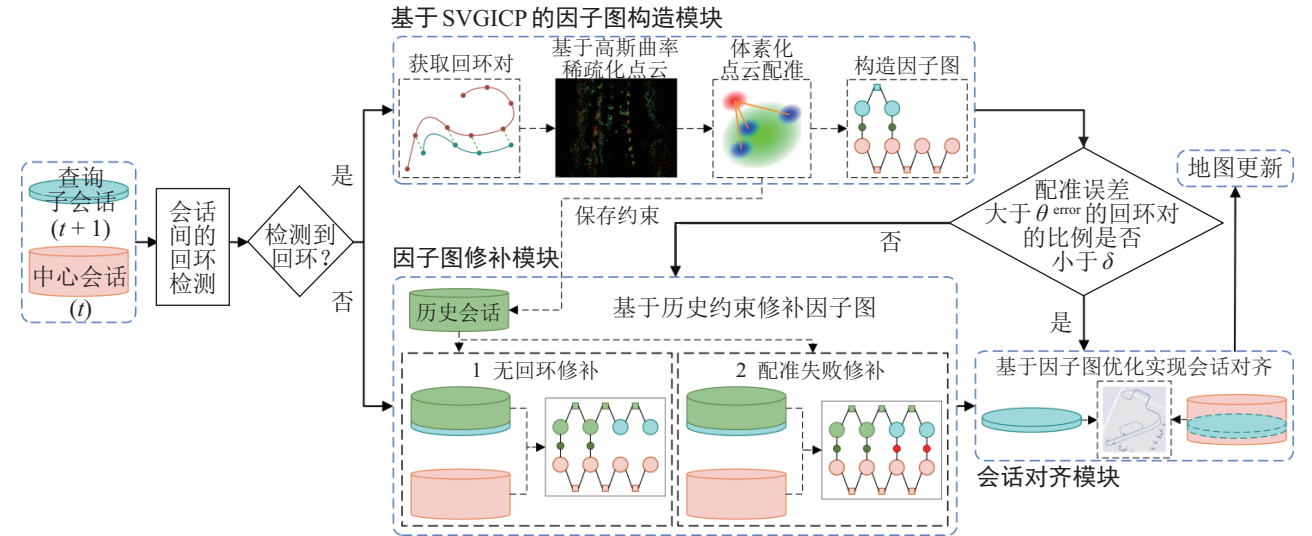


图 1 系统框架图

Fig.1 Frame diagram of the system

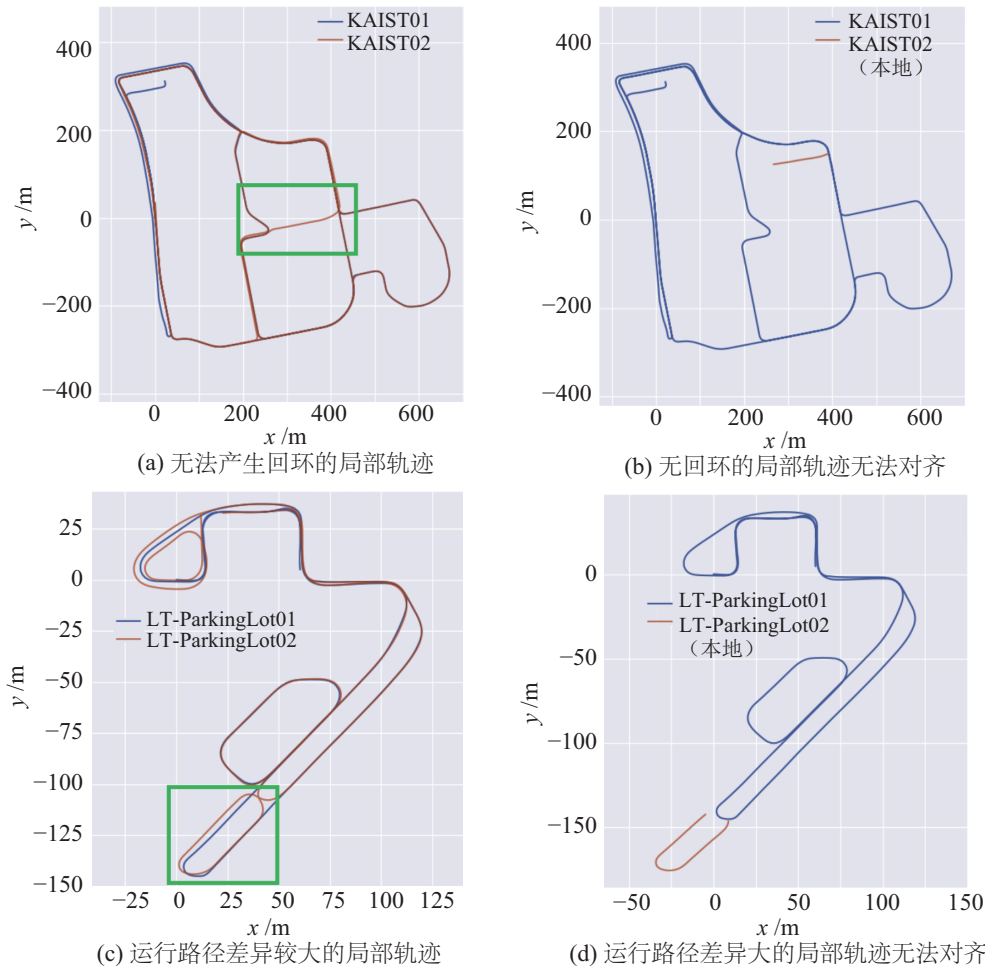


图 2 LT-mapper 无法对齐局部轨迹

Fig.2 LT-mapper cannot align local trajectories

2.1 室外环境下的地图更新算法

本文基于 LT-mapper 开展工作, 重构其中的会话对齐模块。LT-mapper 是一个基于会话对齐的地图更新框架, 其将 t 时刻的会话定义为中心会话, 将 $t+1$ 时刻的会话定义为查询会话, 基于会话间的因子图优化实现对齐。但是, 由于该方法在点云配准部分选择了传统的 ICP 算法, 导致会话对齐速度过慢, 无法实时运行; 并且 LT-mapper 在对齐局部会话时出现了算法失效的情况, 如图 2 所示, 在对齐图 2(a) 和图 2(c) 中的局部会话时会产生如图 2(b) 和图 2(d) 所示无法对齐的情况。这是因为有些局部区域场景变化较大或者两次轨迹差异较大造成因子图构造失败, 影响会话对齐效果。本文分别通过稀疏体素化点云配准和因子图修补解决上述问题。

2.2 基于 SVGICP 的因子图构造模块

解决 LT-SLAM 速度过慢问题的关键是采用了传统的点云配准算法 ICP。本文首先引入高斯曲率对点云进行稀疏化, 在进行点云配准时, 点云的规模大小是影响配准时间的最主要因素, 降低点云规模是提高配准速度最为直接有效的方法; 然后对点云进行体素化, 以此避免极为耗时的最近邻搜索。

基于以上步骤实现点云稀疏体素化配准, 将所得约束信息作为构造因子图的元素。此外, 本文将查询会话进一步划分为查询子会话, 采用流水线方式向中心会话对齐, 定义中心会话为 C , 查询会话为 Q , 查询子会话为 Q^i , 其流程如图 3 所示。

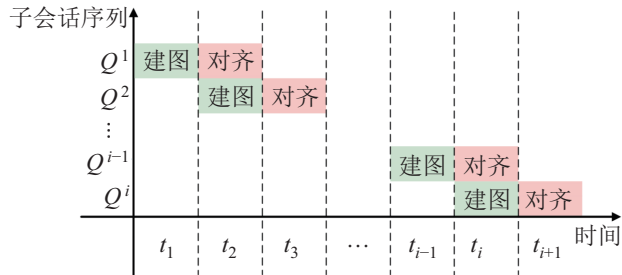


图 3 查询子会话 Q^i 向中心会话流水线式对齐流程

Fig.3 Pipeline alignment process from querying sub-sessions Q^i to the central session

2.2.1 点云稀疏化

在配准过程中, 目的是找到两个点云之间的最佳变换。对于点云特征较少的区域, 点之间的相对位置关系较为一致, 它们无法提供额外的信息来帮助确定最佳变换, 将其定义为配准非关键点。相反, 具有显著特征的区域在配准中应该具有更大的权重, 该区域中的点相对位置关系会发生较大变化, 提供了更多的信息用于匹配和确定变换, 这些点被定义为配准关键点。通过剔除配准非关键点,

降低点云规模, 能够有效提高配准速度。本文引入高斯曲率来度量每个点附近区域的弯曲程度, 并以此选取配准关键点。

首先, 求解点的法向量。定义一点 p_0 及根据 KD 树查找到的邻近点 $\{p_0, p_1, \dots, p_n\}$ 构成集合 P^N , 式 (1) 定义了过点 p_0 的平面方程, 其中 a, b, c 分别是平面在 p_0 点处法向量的方向余弦, d 为原点到平面的距离。

$$ax + by + cz = d \quad (1)$$

上述过程将求解法向量问题近似为一个最小二乘平面拟合问题, 只需要求解式 (1) 这一平面方程即可得到点 p_0 处的法线, 其中参数满足 $a^2 + b^2 + c^2 = 1$, 该等式为使用拉格朗日法求最优解创造了条件。在点集 P^N 中, 任意点 $p_i(x_i, y_i, z_i)$ 到平面的距离为 $d_i = ax_i + by_i + cz_i$, 将所有点到平面距离的平均值定义为 $\bar{d}$, 误差函数如式 (2) 所示:

$$e = \sum_{i=1}^n (d_i - \bar{d})^2 \quad (2)$$

构造拉格朗日函数求其最优解:

$$f = \sum_{i=1}^n (d_i - \bar{d})^2 - \lambda(a^2 + b^2 + c^2 - 1) \quad (3)$$

对 $\bar{d}$ 求偏导, 并令其等于 0, 将 $\bar{d}$ 变换为用 a, b, c 表示的形式, 如式 (4) 所示。因此只需求解 a, b, c 即可获取对应 $\bar{d}$ 的值, 分别对 a, b, c 求偏导, 并令偏导值等于 0, 构造方程组, 转换为矩阵形式, 如式 (5) 所示:

$$\bar{d} = \frac{\sum x_i}{n}a + \frac{\sum y_i}{n}b + \frac{\sum z_i}{n}c \quad (4)$$

$$\begin{pmatrix} \sum \Delta x_i \Delta x_i & \sum \Delta x_i \Delta y_i & \sum \Delta x_i \Delta z_i \\ \sum \Delta y_i \Delta x_i & \sum \Delta y_i \Delta y_i & \sum \Delta y_i \Delta z_i \\ \sum \Delta z_i \Delta x_i & \sum \Delta z_i \Delta y_i & \sum \Delta z_i \Delta z_i \end{pmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \lambda \begin{pmatrix} a \\ b \\ c \end{pmatrix} \quad (5)$$

其中, $\Delta x_i = x_i - \bar{x}$, $\Delta y_i = y_i - \bar{y}$, $\Delta z_i = z_i - \bar{z}$ 。

λ 和 $[a, b, c]^T$ 分别为矩阵的特征值和特征向量, 该矩阵为以 p_0 为中心所构建点集的协方差矩阵, 进一步将求解法向量问题转换为求解协方差矩阵的特征值和特征向量的问题。令 a, b, c 等于该矩阵的最小特征值对应的特征向量, 使得所求法向量为最拟合平面在点 p_0 处的法向量。

在获得每一点的法向量之后, 采用 Zhang 等^[26]提出的算法求解高斯曲率。该方法首先构造密接圆求出点的法曲率, 再通过欧拉公式计算该点处的两个主曲率, 高斯曲率即为两个主曲率的乘积。本文

将点 p_i 的高斯曲率定义为 g_i , 点 p_i 的两个主曲率定义为 K_i^1 和 K_i^2 , 则 g_i 的求解如式 (6) 所示。

$$g_i = K_i^1 \times K_i^2 \quad (6)$$

根据式 (7) 选取配准关键点, 在点集 P^N 中, 高斯曲率大于等于阈值 $\theta_{\text{low}}^{\text{curvature}}$ 且小于等于阈值 $\theta_{\text{high}}^{\text{curvature}}$ 的点构成配准关键点集合 P^N 。高斯曲率较低的点代表附近区域的特征较少, 不能提供额外信息帮助配准; 高斯曲率过高的点代表该点极大可能是离群点, 对配准会产生负面影响: 这两类点皆需要剔除。本文参数的设置将在实验部分介绍。

$$P^N = \{p_i \in P^N \mid \theta_{\text{low}}^{\text{curvature}} \leq g_i \leq \theta_{\text{high}}^{\text{curvature}}\} \quad (7)$$

2.2.2 体素化点云配准

在获取配准关键点集合后, 采用体素化点云配准计算两个稀疏化点云之间的最佳变换。记两个点云集合分别为 A (源点云) 和 B (目标点云):

$$A = \{a_i \mid 0 \leq i \leq N\}, \quad B = \{b_j \mid 0 \leq j \leq M\} \quad (8)$$

其中, $a_i(x_{a_i}, y_{a_i}, z_{a_i})$, $b_j(x_{b_j}, y_{b_j}, z_{b_j})$ 。

对每一点进行高斯分布建模: $a_i \sim N(\bar{a}_i, \mathbf{C}_i^A)$, $b_j \sim N(\bar{b}_j, \mathbf{C}_j^B)$, 其中 $\bar{a}_i$ 和 $\bar{b}_j$ 分别为以 a_i 点和 b_j 点为中心所构成点集的均值, $\mathbf{C}_i^A$ 和 $\mathbf{C}_j^B$ 为对应的协方差矩阵。本文对目标点云体素化, 定义体素集合为 V , 定义体素个体为 v , 定义体素分辨率为 r , 定义体素索引为 (X, Y, Z) 。对于目标点云 B , 根据式 (9) 计算其中某一点所在体素的索引。

$$(X, Y, Z) = \left(\left\lfloor \frac{x_{b_j}}{r} \right\rfloor, \left\lfloor \frac{y_{b_j}}{r} \right\rfloor, \left\lfloor \frac{z_{b_j}}{r} \right\rfloor \right) \quad (9)$$

其中, $\lfloor * \rfloor$ 表示对括号内的值向下取整。将索引相同的点放在同一个体素中, 以这种方式构建的体素模型可以保证体素内的点在空间上是邻近的。对目标点云体素化处理后, 通过计算体素索引即可找到点云 B 中与源点云 A 中点 a_i 对应的体素, 基于 a_i 和与其对应的体素 v_{a_i} 构造误差公式:

$$e_i = \sum_{v_{a_i}} \|b_j - \mathbf{T}a_i\| \quad (10)$$

其中, e_i 是构建的误差函数, a_i 是 a_i 的向量表示, b_j 是体素 v_{a_i} 中点的向量表示, $\mathbf{T}$ 为二者之间的变换矩阵。 e_i 的分布如式 (11) 所示:

$$\begin{aligned} e_i &\sim N(\mu^{e_i}, \mathbf{C}^{e_i}) \\ \mu^{e_i} &= \sum_{v_{a_i}} (b_j - \mathbf{T}a_i) = 0 \\ \mathbf{C}^{e_i} &= \sum_{v_{a_i}} (\mathbf{C}_j^B + \mathbf{T}\mathbf{C}_i^A\mathbf{T}^T) \end{aligned} \quad (11)$$

记 e_i 的概率密度函数为 p_{e_i} , a_i 与 v_{a_i} 中每一点的向量误差为 q_i , 概率密度函数最大时的变换矩阵即为所求矩阵, 见式 (12)。为了有效计算该公式, 将其修改为式 (13):

$$\mathbf{T} = \arg \max_{\mathbf{T}} \sum_{v_{a_i}} p_{e_i} = \arg \min_{\mathbf{T}} \sum_{v_{a_i}} q_i^T \mathbf{C}_i^{-1} q_i \quad (12)$$

$$\mathbf{T} = \arg \min_{\mathbf{T}} \sum_{v_{a_i}} N_{v_{a_i}} q_i^T \mathbf{C}_i^{-1} q_i \quad (13)$$

其中, $q_i = \frac{\sum_{v_{a_i}} b_j}{N_{v_{a_i}}} - \mathbf{T}a_i$, $\mathbf{C}_i = \frac{\sum_{v_{a_i}} \mathbf{C}_j^B}{N_{v_{a_i}}} + \mathbf{T}\mathbf{C}_i^A\mathbf{T}^T$ 。

$N_{v_{a_i}}$ 为体素 v_{a_i} 中点的个数, 该过程考虑单点分布到体素分布的马氏距离, $\sum_{v_{a_i}} b_j / N_{v_{a_i}}$ 和 $\sum_{v_{a_i}} \mathbf{C}_j^B / N_{v_{a_i}}$ 分别是该体素的均值和协方差矩阵。通过不断迭代求得两个点云之间的最佳变换矩阵, 稀疏体素化点云配准过程如算法 1 所示。

算法 1 稀疏体素化点云配准算法

输入: 两帧点云 $A = \{a_0, \dots, a_N\}$, $B = \{b_0, \dots, b_M\}$, 两帧点云的协方差矩阵集合 $\mathbf{C}^A = \{\mathbf{C}_0^A, \dots, \mathbf{C}_N^A\}$, $\mathbf{C}^B = \{\mathbf{C}_0^B, \dots, \mathbf{C}_M^B\}$

输出: 两帧点云间的最佳变换矩阵 $\mathbf{T}$

- 1: 初始化变换矩阵 $\mathbf{T}$
- 2: **procedure** 点云配准 ($A, B, \mathbf{C}^A, \mathbf{C}^B, \mathbf{T}$)
- 3: $A' \leftarrow$ 点云稀疏化 ($A, \mathbf{C}^A$)
- 4: $V \leftarrow$ 点云体素化 ($B, \mathbf{C}^B$)
- 5: **for** $a_i \in A'$ **do**
- 6: $v \leftarrow \left\lfloor \frac{a_i}{r} \right\rfloor$
- 7: **if** $v \in V$ **then**
- 8: $\Delta \mathbf{T} \leftarrow \text{cost}(\mathbf{T}, a_i, \mathbf{C}_i^A, v)$ ▷ 求解扰动
- 9: $\mathbf{T} \leftarrow \mathbf{T} \oplus \Delta \mathbf{T}$ ▷ 更新扰动到最新结果中
- 10: **end if**
- 11: **end for**
- 12: **end procedure**
- 13: **return** $\mathbf{T}$

采用该算法求解出两个点云之间的变换矩阵并将其加入到会话因子图构造中, 定义求解出的约束为 $\mathbf{z}_{C_i, Q_j}^{\text{between}}$, 其表示产生回环的中心会话点云 C_i 和查询会话点云 Q_j 之间的最佳变化矩阵。由于本文在建图过程采用 LIO-SAM 算法, 因此会话内有如下约束信息: 雷达里程计因子 $\mathbf{z}^{\text{odom}}$, IMU 预积分因子 $\mathbf{z}^{\text{IMU}}$, 回环因子 $\mathbf{z}^{\text{loop}}$ 。定义中心会话的状态节点为 $\mathbf{x}_{C_i}$, 定义查询会话的状态节点为 $\mathbf{x}_{Q_j}$, 则会话因子图构造如图 4(a) 所示。

2.3 因子图修补

上述模块能完成大多数情况下的因子图构造, 但遇到以下两种情况时将无法正常完成系统功能: 其一, 在局部地图向全局地图对齐时, 若某局部场

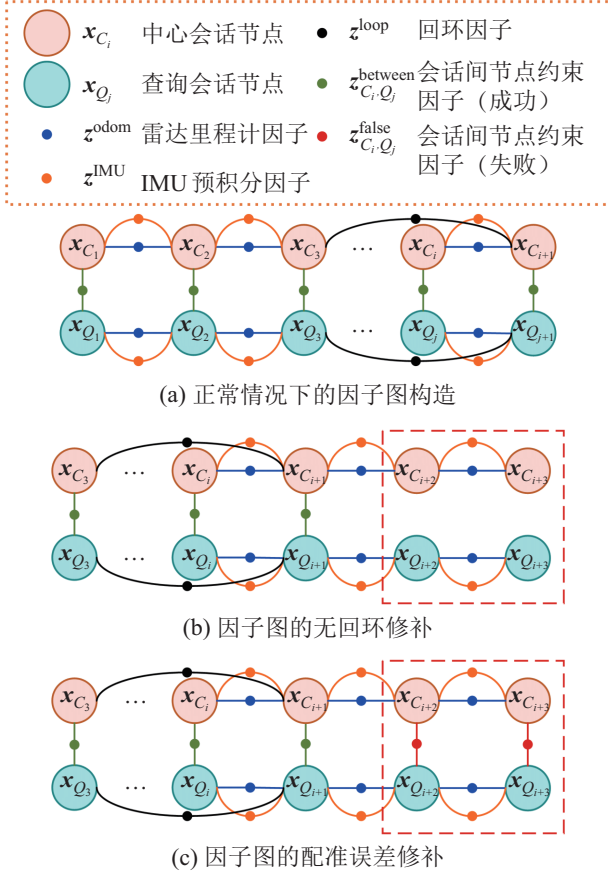


图 4 因子图构造与修补
Fig.4 Construction and repair of the factor graph

景变化较大，或是经过一段全新的场景，将无法检测到回环，导致因子图构造失败，进而无法完成会话对齐，如图 2(b) 所示；其二，经过某局部场景前后两次的轨迹有较大差别，基于坐标计算对应体素索引的算法将会失效，点和体素无法形成正确对应关系，导致配准误差增大，构造的因子图无法满足会话对齐的需求，如图 2(d) 所示。本文设计因子图修补模块以解决上述问题，提高系统的鲁棒性。

2.3.1 无回环修补

本文基于历史约束进行因子图修补，历史约束是已经计算出的会话间约束，能够基于历史约束修补因子图的理论基础是：虽然会话间检测不到回环无法建立约束，但由于采用流水线式对齐，因此各会话内的子会话之间是连续的，具有里程计因子、IMU 预积分因子等约束，间接对未产生回环部分形成约束。

定义历史约束集合为 $z^{history}$ ，其表示先前可以检测到回环的中心会话与查询会话间的约束，如式 (14) 所示。其中 C_i^{index} 表示中心会话点云 C_i 的索引， Q_j^{index} 表示查询会话点云 Q_j 的索引， $z^{history}$ 是计算出的两者间约束， $C_i \propto Q_j$ 代表二者间存在回环。图

4(b) 直观地展示了因子图修补过程，其中红框内表示无法检测到回环的情况，但通过历史约束可以建立起两会话间的因子图。

$$z^{history} = \{(C_i^{index}, Q_j^{index}, z_{C_i, Q_j}^{between}) | C_i \propto Q_j\} \quad (14)$$

2.3.2 配准误差检测与修补

除上述无法检测到回环的情况外，还存在两次运行轨迹差异特别显著的情况，会导致配准误差增大，以至于局部会话对齐失效。因为在进行配准时采用体素索引实现点和体素的关联，而体素索引根据坐标进行计算，所以在运行轨迹差异较大时，无法根据上述方法关联对应的点和体素，图 5 直观地说明了这种情况，图 5(a) 中 5 号框内的点在运行轨迹差异不大的情况下，通过体素索引应与图 5(b) 中的 5 号体素关联；但如果运行轨迹差异较大，则会出现图 5(c) 的情况，点云整体发生偏移，此时图 5(a) 中 5 号框内的点将与图 5(c) 中 6 号体素关联，产生错误关联。

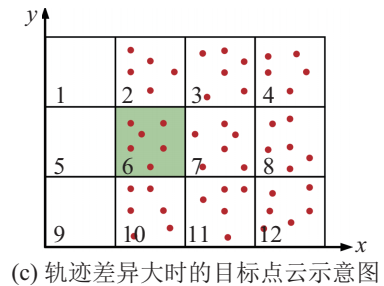
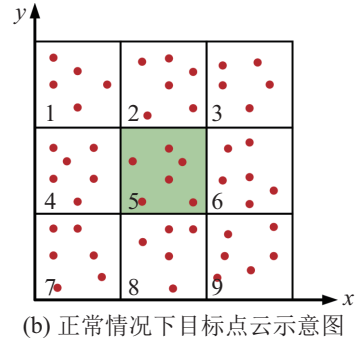
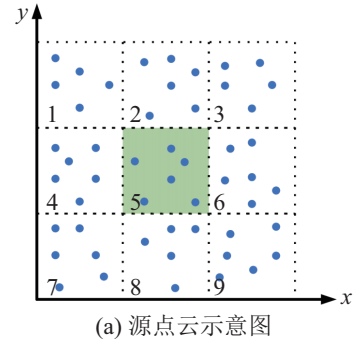


图 5 点和体素错误关联示意图
Fig.5 Schematic diagram of incorrect association between points and voxels

因此本文在点云配准的同时会计算配准误差, 定义配准误差为 E_{C_i, Q_j} , 其计算过程如式 (15) 所示。其中 a_i 和 b_j 分别为配准后点云 C_i 和 Q_j 之间的对应点。将配准后所有对应点之间欧氏距离的均方根误差作为配准误差, 当 E_{C_i, Q_j} 大于 θ^{error} 时认为求解出的约束条件不适合加入因子图构造中, 本文将这种约束条件记为 $\mathbf{z}_{C_i, Q_j}^{\text{false}}$ 。并且规定在上述约束条件数量占有回环对数量的比例小于一定数值时才进行因子图构造, 本文将该值定义为 δ ; 而在不满足上述条件时, 同样需要调用历史约束进行修补, 见图 4(c)。这是因为当可用的会话间约束过少时, 通过因子图优化无法完成局部会话的对齐。

$$E_{C_i, Q_j} = \sqrt{\frac{\sum_{N_{\text{count}}} (a_i - b_j)^2}{N_{\text{count}}}} \quad (15)$$

2.4 会话对齐与地图更新

已对齐的会话为实现地图更新提供了前提条件, 通过上述过程得到中心会话 C 和查询会话 Q 间的因子图。迭代优化式 (16) 得到查询会话向中心会话对齐的位姿, 其中 $\mathbf{z}^{\text{prior}}$ 为各会话内的先验因子。

$$X^* = \arg \max_X \Phi(X) \quad (16)$$

$$\Phi(X) = \prod \mathbf{z}^{\text{prior}} \prod \mathbf{z}^{\text{odom}} \prod \mathbf{z}^{\text{IMU}} \prod \mathbf{z}^{\text{loop}} \prod \mathbf{z}_{C_i, Q_j}^{\text{between}}.$$

在两个会话对齐后, 使用 Removert 算法^[27]进行变化检测和地图更新, 将中心会话中不存在而查询会话中存在的点作为新增部分, 将中心会话中存在而查询会话中不存在的点作为消失部分, 则中心点云的地图更新过程是: 将新增部分添加到地图中, 将消失部分从地图中剔除。

3 实验与分析 (Experiment and analysis)

为验证所提出方法的有效性, 本文在 MulRan 数据集^[28]和 LT-ParkingLot 数据集上进行评估。MulRan 数据集主要包含 4 种场景, 市区、校园、河边、发展中的城市, 不同场景有多个序列。本文选取其中规模适中的校园环境, 轨迹长度大约为 6 km。该场景包含 3 个序列, 分别是 KAIST01、KAIST02、KAIST03, 每个序列对同一场景建图, 但起始位置和建图路径不同, 同时整体轨迹也未对齐, 本文使用该场景验证所提会话对齐方法的有效性。LT-ParkingLot 包含 6 个序列, 轨迹长度大约为 1 km, 是在 6 个不同的时间段对某停车场环境进行建图, 停车场是半静态环境的经典场景, 其内经常会有新增和驶离的车辆, 而这些变化在建图时期可能无法检测。本文使用该数据集展示在会话对齐后

的半静态物体检测及地图更新效果。本文的实验平台采用 Intel i7 处理器, 24 GB 内存。

3.1 参数设置

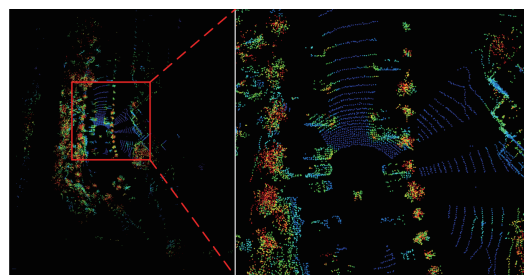
所提出方法需要设定一些阈值参数, 参数的设置经过上述数据集验证调优得到, 实验部分所使用的参数值如表 1 所示。

表 1 参数设置
Tab.1 Parameter setting

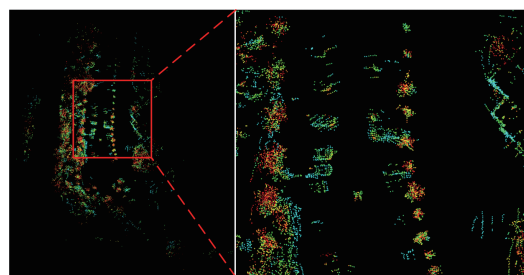
$\theta_{\text{curvature low}}$	$\theta_{\text{curvature high}}$	θ^{error}	δ	r
5×10^{-7}	5×10^{-3}	1.3	0.4	1.0

3.2 点云稀疏化分析

为验证点云稀疏化的有效性, 本文取一帧点云进行稀疏化处理并对其展开分析。图 6(a) 展示的是某一帧点云稀疏化处理之前的点云, 图 6(b) 为之后的, 两者都对中心区域进行了放大处理, 能够更加直观地看出稀疏化前后的效果。本文用色温反映高斯曲率的高低, 其中蓝色点表示其高斯曲率最低, 红色点表示其高斯曲率最高, 其余颜色的点介于二者之间。根据图示可以直观看出来地面和墙壁等比较平坦、特征较少的部分展现为蓝色, 树木和车辆等表面弯曲程度较大、特征较显著的部分大多展现为红色和黄色, 经过稀疏化处理大部分地面点等非配准关键点被剔除, 保留下来的配准关键点则是车辆、树木等部分。此外, 稀疏化处理之前该点云中点的个数为 23 370 个, 之后为 15 096 个, 相较于前者点云规模减小了 35.4%。



(a) 稀疏化之前点云示意图



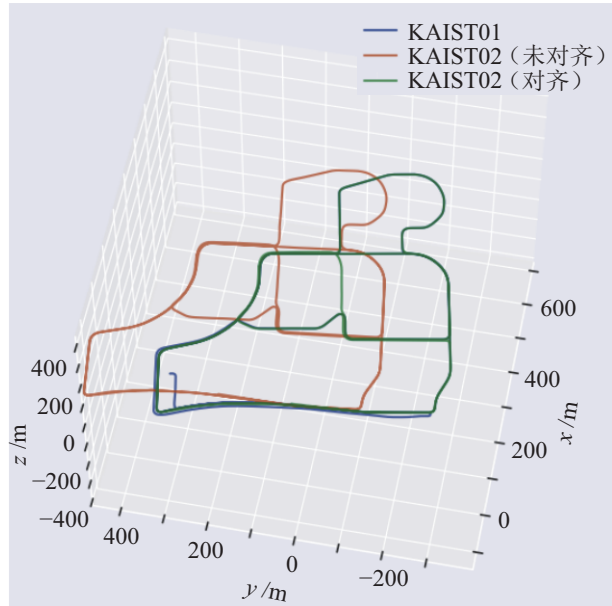
(b) 稀疏化之后点云示意图

图 6 点云稀疏化处理前后对比

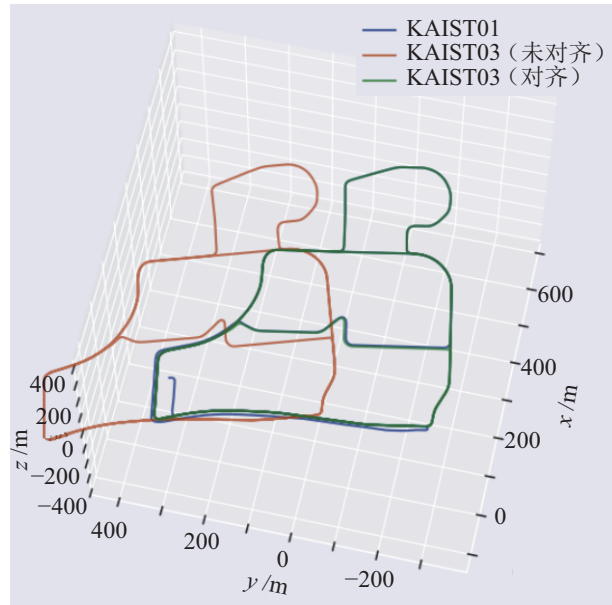
Fig.6 Comparison of point cloud before and after the sparsification

3.3 会话对齐效果分析

在 KAIST01、KAIST02、KAIST03 序列上进行会话对齐实验，如图 7 所示。分别将 KAIST02 序列和 KAIST03 序列向 KAIST01 序列对齐，其中蓝色轨迹是 KAIST01 序列，称为中心轨迹，红色轨迹是未对齐的序列，绿色轨迹是经过本文方法对齐后的序列，可以直观看出本文方法在不同序列之间会话对齐的有效性。



(a) KAIST02 序列向 KAIST01 序列对齐



(b) KAIST03 序列向 KAIST01 序列对齐

图 7 不同序列的对齐轨迹和未对齐轨迹与中心轨迹对比
Fig.7 Comparison of the aligned and unaligned trajectories with the central trajectories on different sequences

由于不同序列的轨迹不同，在对比平移误差和

旋转误差时不具备参考性，因此本文在同一序列下进行两次建图操作，用于保证其整体轨迹一致，将其中一条轨迹随机进行变换，作为中心轨迹，再使用本文方法将未经变换的轨迹向其对齐，如图 8 所示。其中图 8(a)(c)(e) 分别是未对齐的轨迹、平移误差和旋转误差，图 8(b)(d)(f) 分别为对齐后的上述指标，可以看出对齐后的轨迹与真实轨迹相差无几。

3.4 半静态环境下长期运行分析

为了模拟机器人在半静态环境中长期运行的情景，一方面，本文通过重复播放 KAIST01 序列让算法持续建图来模拟机器人长期运行，另一方面手工变换数据集中的若干点云数据来模拟半静态环境。表 2 比较了本文方法与其余 3 个经典的激光惯性融合的 SLAM 算法：LeGO-LOAM^[29]、LIO-SAM^[25]、FAST-LIO2^[30]，其中 N1 到 N5 表示运行的轮次，建图失败用“-”表示。可以看出，本文方法在轮次递增时绝对轨迹误差未发生太大变化，而随着轮次的增加，其余 3 个 SLAM 算法的绝对轨迹误差也在增加。一方面是因为随着运行时间的增长，其累积误差在不断增大，而本文方法每个轮次之间的建图都是相互独立的，累积误差可以保持在单轮建图的水平，相较于其余 3 个算法小很多；另一方面是因为变化的环境对其余 3 个算法会造成负面影响，使机器人的定位误差增大，进而影响机器人的运行轨迹。另外可以发现，LIO-SAM 和 LeGO-LOAM 在第 4 轮开始建图失败，原因是随着运行时间的增长，其需要处理的数据量在不断增大，到达第 4 轮运行时，算法需要的计算资源无法得到满足，导致建图失败。而 FAST-LIO2 由于没有回环检测模块，相较于其他两个算法更加轻量级，因此其顺利地运行到了第 5 轮次，但与此同时，它的绝对轨迹误差也较大。相比其余 3 个方法，本文方法通过将长期运行任务拆解为若干独立的短期运行任务来保证系统精确度。

表 2 其他 SLAM 方法与本文方法的绝对轨迹误差
Tab.2 Absolute trajectory error between other SLAM methods and the method proposed

单位：m				
轮次	LeGO-LOAM	LIO-SAM	FAST-LIO2	本文方法
N1	4.3	2.7	6.7	2.5
N2	7.4	4.2	8.1	2.7
N3	9.1	7.3	14.3	2.9
N4	—	—	20.6	2.5
N5	—	—	33.5	2.4

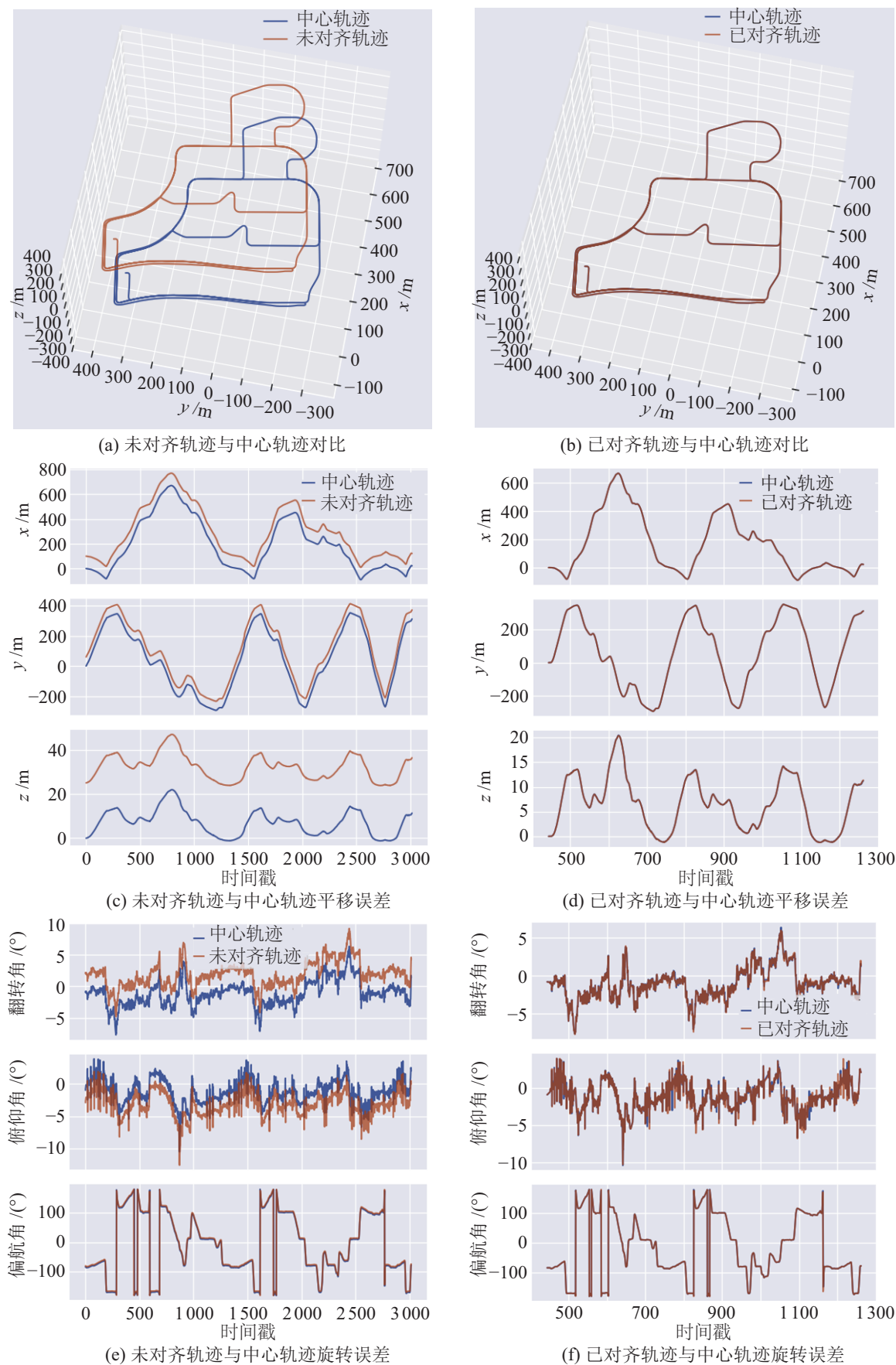


图 8 LT-SLAM 与本文方法在 KAIST01 序列上的轨迹对比

Fig.8 Comparison of trajectories between LT-SLAM and the proposed method on KAIST01 sequence

3.5 因子图修补实验分析

为检验因子图修补模块的有效性,在图 2 展示的两对序列上进行局部会话对齐实验,实验结果见图 9。对比图 2(b)(d)可直观地看出,在历史约束的作用下图 9(a)(b)能够成功向中心轨迹对齐。通过实验可发现,在因子图修补模块的作用下,一些本不能对齐的局部轨迹也能够对齐,因子图修补模块能够处理由于检测不到回环和配准误差较大所引起局部会话无法对齐的问题。并且值得注意的一点是,图 9(a)中的红色轨迹在 KAIST01 序列中没有出现,但是在经过对齐之后能够很好地拟合到其应在的位置,这与地图拼接实现的效果无二,采用本文方法能够以较小的代价对地图进行扩展。

3.6 地图更新分析

本文在 LT-ParkingLot01 和 LT-ParkingLot02 两个序列上进行地图更新实验。首先采用本文方法对齐上述两个序列,然后将已对齐的会话作为输入,使用 Kim 等^[27]提出的一种基于视点可见的动态物体检测和剔除方法实现地图更新。

图 10 展示了地图更新的效果,第 3 列为中心会话和查询会话的融合点云。选取了完整点云地图中的 4 个区域来说明地图更新情况,即图 10(a)~(d)对应图 11(a)中标注的 4 个区域,分别展示了 4 种不同的情况:区域 1,在中心会话中出现的部分在查询会话中并未出现,说明该部分在最新的地图中消失了,该区域在融合点云中展现的状态是全红色;区域 2,在中心会话中出现的部分在查询会话中也出现了,说明该部分未发生变化,该区域在融合点云中展现的状态是红蓝交替;区域 3,在中心

会话和查询会话中均未出现物体,说明该区域没有新增部分也没有消失部分,该区域在融合点云中展现的状态也为红蓝交替,即为地面点;区域 4,中心会话中不存在的部分在查询会话中出现了,说明该部分是在最新的地图中新增的,该区域在融合点云中展现的状态为全蓝色。综上所述,在融合点云中保留蓝色点云和红蓝交替的点云即为最新地图,即展现为图中的绿色点云。

此外,本文还引入倒角距离^[31]度量点云相似度,点云间倒角距离越小表示点云越相似。通过计算更新前后点云的倒角距离可定量检验地图更新的效果,如图 11 所示。将每个序列建立的点云地图以及更新后的点云地图划分为 $10\text{ m} \times 10\text{ m} \times z$ 的长方体, z 是点云的高度,然后分别计算对应方块之间的倒角距离。由倒角距离可视化效果图可以直观地看出,经由本文方法更新后的地图在绝大部分区域下能够和最新环境保持一致。另外,将本文方法与 LT-mapper 方法进行了比较,见表 3。由表 3 可知,本文方法与最新环境的一致性更好。

表 3 地图更新前后的倒角距离对比

Tab.3 Comparison of chamfer distance before and after the map updating

一对点云地图	倒角距离		
	最大值	平均值	方差
未更新的 01 和 02	15.67	1.66	2.83
使用 LT-mapper 更新的 01 和 02	1.17	0.21	0.05
使用本文方法更新的 01 和 02	1.08	0.17	0.03

注: 01 和 02 为 LT-ParkingLot 数据集下的序列号。

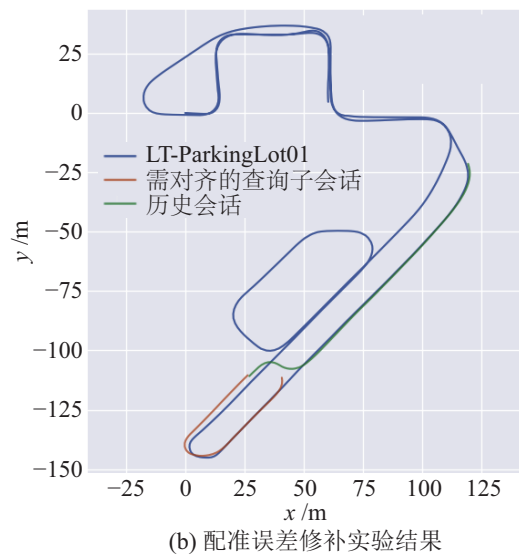
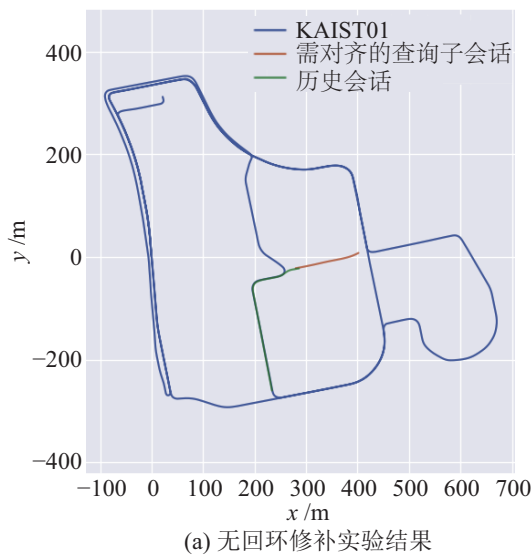


图 9 因子图修补实验结果展示

Fig.9 Display of experimental results for factor graph repair

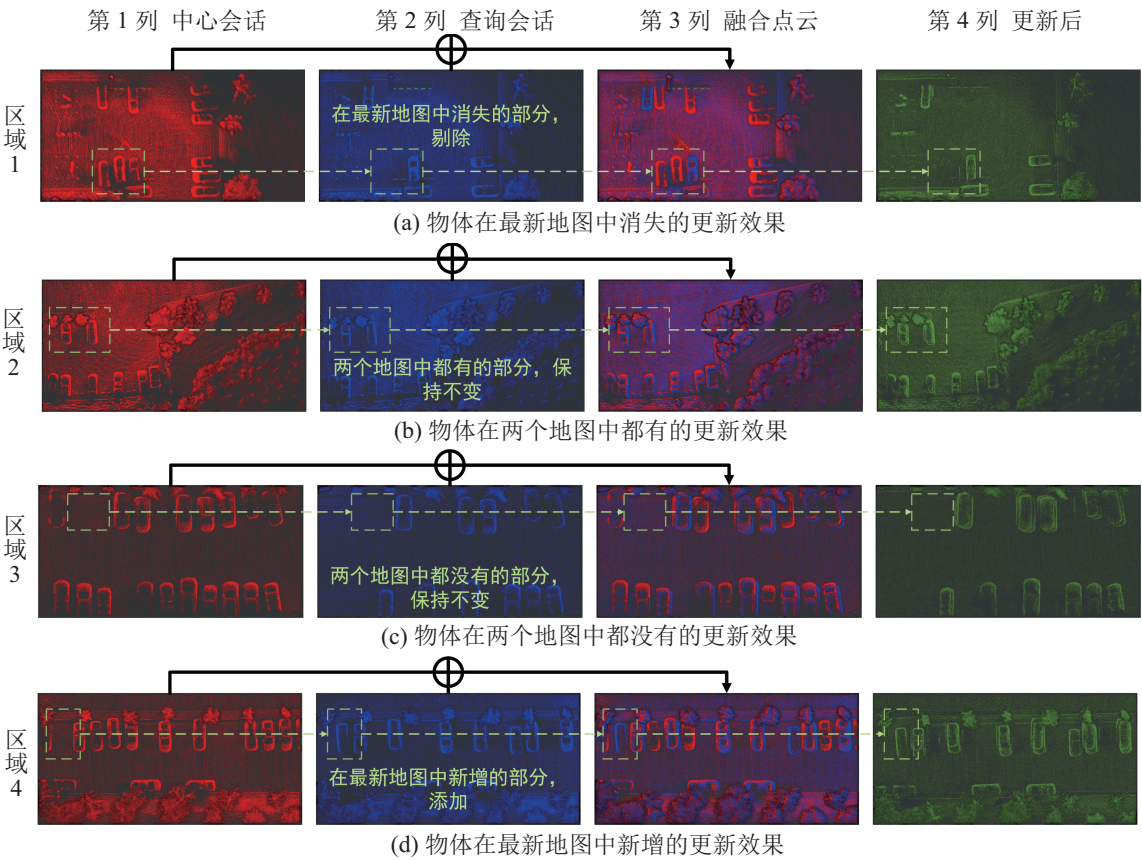


图 10 地图更新效果

Fig.10 Effect of the map updated

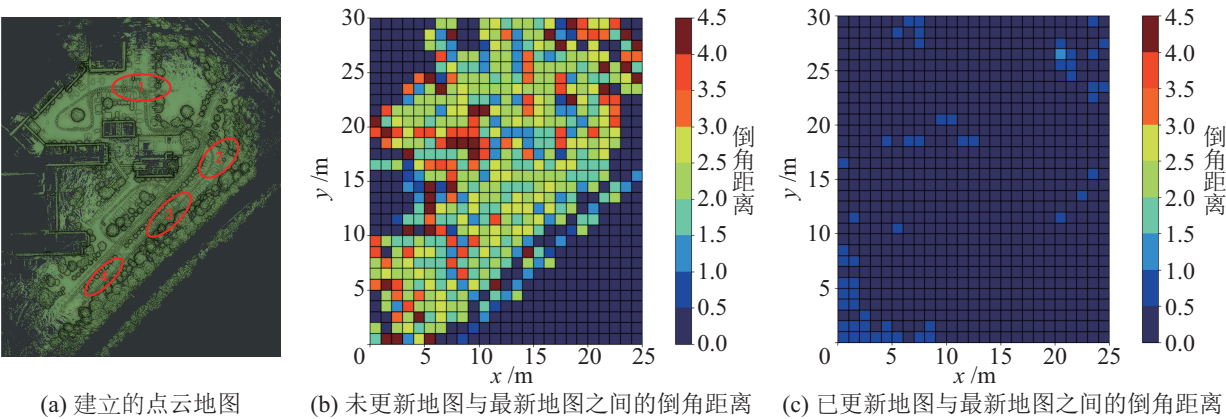


图 11 倒角距离可视化效果图

Fig.11 Visualization diagram of chamfer distance

表 4 3 种点云配准方法的配准时间和结果相似度比较

Tab.4 Comparison of registration time and result similarity among the 3 point cloud registration methods

点云序列	ICP		SICP (仅稀疏化)			SVGICP (稀疏体素化)			
	点的个数	配准时间 /ms	点的个数	配准时间 /ms	I /%	点 / 体素个数	配准时间 /ms	I /%	p /%
中心帧 1	24 199	380	14 581	233	38.7	14 581 (点)	77	79.7	99.93
查询帧 1	25 402		15 073			2 051 (体素)			
中心帧 2	24 407	378	15 267	243	35.7	15 267 (点)	79	79.1	99.89
查询帧 2	25 818		15 503			2282 (体素)			
中心帧 3	24 496	381	14 224	231	39.4	14 224 (点)	74	80.6	99.90
查询帧 3	24 518		14 331			2 271 (体素)			

3.7 运行时间分析

3.7.1 点云配准时间分析

随机抽取 3 对产生回环的点云分别采用传统的 ICP 方法、稀疏化点云配准方法 (SICP) 和稀疏体素化点云配准 (SVGICP) 方法进行计算, 结果统计如表 4 所示。其中 I 为后两种方法相较于 ICP 方法的配准时间下降百分比, ρ 为 ICP 和 SVGICP 所得配准结果的相似度, 若上述方法所得变换矩阵分别为 $\mathbf{M}$ 和 $\mathbf{N}$, 则 ρ 的定义为

$$\rho = \frac{1}{1 + \|\mathbf{M} - \mathbf{N}\|_F} \quad (17)$$

$$\|\mathbf{M} - \mathbf{N}\|_F = \sqrt{\sum_{i,j} |\mathbf{M}_{i,j} - \mathbf{N}_{i,j}|^2} \quad (18)$$

其中 $\|\mathbf{M} - \mathbf{N}\|_F$ 是两个矩阵的 Frobenius 范数, 是用于度量矩阵相似性的一种常用指标。两个矩阵越相似, 其 Frobenius 范数越小; 反之, 则越大。根据上述性质, 两个矩阵越接近, 则 ρ 的值越接近 1。

由表 4 可见, 稀疏化后的点云, 其点的数量平均下降了 40% 左右, 配准时间明显缩短, 但其配准频率只能达到 4 Hz 左右, 无法满足实时性的要求。进一步对点云进行体素化处理后, 配准时间 (包含了对点云作预处理的时间) 大大缩短, 相较于 ICP 算法下降了约 80%; 同时, 观察 ρ 的值能够发现, 本文方法在提高配准速度的同时可以保证结果的准确性。

本文方法点云配准速度的性能提升基于以下 3 个方面: 一是对点云进行了稀疏化, 选取其中的配准关键点进行运算, 降低了点云规模, 配准速度有了初步提升; 二是对点云进行了体素化, 每一个体素的分布是基于其内包含的点计算得到的, 而不是通过 NDT 等基于直方图的计算方式, 这样即使体素内只有 1 个点, 也能够得到合适的体素分布, 并且通过体素索引将相关的点和体素进行关联, 避免了极为耗时的最邻近搜索, 进一步提升了配准速度; 三是经过体素化后, 在优化过程中不需要进行最近邻搜索, 不存在大量的条件分支, 可以引入 GPU 进行加速。

3.7.2 主要模块运行时间分析

在 LT-ParkingLot 数据集上针对所提出的会话对齐算法的主要模块运行时间进行了实验验证。选取该数据集的 01 序列和 02 序列, 实现 02 序列向 01 序列对齐, 分别对两个序列进行建图后, 01 序列有 629 个节点, 02 序列有 600 个节点, 二者之间产生了 533 个回环。运行时间的对比涉及回环检测、点

云配准、图优化 3 个主要过程。

将基于 SICP 的本文方法和基于 SVGICP 的本文方法与 LT-SLAM 方法的时间开销进行对比, 如表 5 所示。可直观看出, 系统的主要时间开销集中于点云配准模块, 提高点云配准的速度可以有效提高会话对齐的速度, 其频率达到 13 Hz, 可以满足实时性需求。

表 5 主要模块运行时间

Tab.5 Running time of the main modules

单位: ms				
方法	回环检测	点云配准	图优化	总时间
LT-SLAM	210	221 504	100	221 814
本文方法 (基于 SICP)	217	117 183	91	117 491
本文方法 (基于 SVGICP)	196	32 970	95	33 261

4 总结 (Conclusion)

针对目前室外半静态环境下地图更新算法无法实时运行的问题, 提出一个基于快速会话对齐的地图更新方法。首先, 提出一种稀疏体素化点云配准方法, 从降低点云规模和避免最近邻搜索两个方面提高配准速度; 然后, 基于上述点云配准方法, 设计了一种流水线式会话对齐方法, 实现两个会话实时精确的对齐; 同时, 设计因子图修补方法, 用于应对局部会话对齐失效情况, 同时其在地图扩展方面展现出显著效果; 最后, 基于对齐的会话实现地图更新。与经典的地图更新算法和激光 SLAM 算法进行对比, 证明了本文方法在长期 SLAM 任务下具有一定优势。

本文的工作仍存在一些值得改进的地方, 比如进一步提高会话对齐的精度和速度。此外, 还计划将对齐的会话应用到后续研究中, 如地图拼接等。

参考文献 (References)

- [1] 陶永, 刘海涛, 王田苗, 等. 我国服务机器人技术研究进展与产业化发展趋势[J]. 机械工程学报, 2022, 58(18): 56-74.
TAO Y, LIU H T, WANG T M, et al. Research progress and industrialization development trend of Chinese service robot[J]. Journal of Mechanical Engineering, 2022, 58(18): 56-74.
- [2] JIA Y J, YAN X Y, XU Y H. A survey of simultaneous localization and mapping for robot[C]//IEEE 4th Advanced Information Technology, Electronic and Automation Control Conference. Piscataway, USA: IEEE, 2019: 857-861.
- [3] HESS W, KOHLER D, RAPP H, et al. Real-time loop closure in 2D LIDAR SLAM[C]//IEEE International Conference on Robotics and Automation. Piscataway, USA: IEEE, 2016: 1271-1278.

- [4] MUR-ARTAL R, TARDOS J D. ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras[J]//IEEE Transactions on Robotics, 2017, 33(5): 1255-1262.
- [5] CAMPOS C, ELVIRA R, RODRÍGUEZ J J G, et al. ORB-SLAM3: An accurate open-source library for visual, visual-inertial, and multimap SLAM[J]//IEEE Transactions on Robotics, 2021, 37(6): 1874-1890.
- [6] LIM H, HWANG S, MYUNG H. ERASOR: Egocentric ratio of pseudo occupancy-based dynamic object removal for static 3D point cloud map building[J]. IEEE Robotics and Automation Letters, 2021, 6(2): 2272-2279.
- [7] QIAN C L, XIANG Z H, WU Z R, et al. RF-LIO: Removal-first tightly-coupled Lidar inertial odometry in high dynamic environments[C]//IEEE/RSJ International Conference on Intelligent Robots and Systems. Piscataway, USA: IEEE, 2021: 4421-4428.
- [8] FANG B F, MEI G F, YUAN X H, et al. Visual SLAM for robot navigation in healthcare facility[J]//Pattern Recognition, 2021, 113. DOI: 10.1016/j.patcog.2021.107822.
- [9] QIAN J X, CHATRATH V, SERVOS J, et al. POV-SLAM: Probabilistic object-aware variational SLAM in semi-static environments[DB/OL]. (2023-07-02) [2024-09-23]. <https://arxiv.org/abs/2307.00488>.
- [10] LABBE M, MICHAUD F. Long-term online multi-session graph-based SPLAM with memory management[DB/OL]. (2022-12-30) [2024-09-23]. <https://arxiv.org/abs/2301.00050>.
- [11] KIM G, KIM A. LT-mapper: A modular framework for LiDAR-based lifelong mapping[C]//International Conference on Robotics and Automation. Piscataway, USA: IEEE, 2022: 7995-8002.
- [12] SEGAL A, HAEHNEL D, THRUN S. Generalized-ICP[M]//Robotics: Science and Systems V. Cambridge, USA: MIT Press, 2010.
- [13] DAS A, WASLANDER S L. Scan registration using segmented region growing NDT[J]. International Journal of Robotics Research, 2014, 33(13): 1645-1663.
- [14] DAS A, SERVOS J, WASLANDER S L. 3D scan registration using the normal distributions transform with ground segmentation and point cloud clustering[C]//IEEE International Conference on Robotics and Automation. Piscataway, USA: IEEE, 2013: 2207-2212.
- [15] KOIDE K, YOKOZUKA M, OISHI S, et al. Voxelized GICP for fast and accurate 3D point cloud registration[C]//IEEE International Conference on Robotics and Automation. Piscataway, USA: IEEE, 2021: 11054-11059.
- [16] STEFANINI E, CIANCOLINI E, SETTIMI A, et al. Efficient 2D LIDAR-based map updating for long-term operations in dynamic environments[C]//IEEE/RSJ International Conference on Intelligent Robots and Systems. Piscataway, USA: IEEE, 2022: 832-839.
- [17] STEFANINI E, CIANCOLINI E, SETTIMI A, et al. Safe and robust map updating for long-term operations in dynamic environments[J]. Sensors, 2023, 23(13). DOI: 10.3390/s23136066.
- [18] KAESE M, RANGANATHAN A, DELLAERT F. iSAM: Incremental smoothing and mapping[J]. IEEE Transactions on Robotics, 2008, 24(6): 1365-1378.
- [19] 路春晓, 钟焕, 刘威, 等. 复杂地形环境下的多传感器融合 SLAM 技术[J]. 机器人, 2024, 46(4): 425-435.
- LU C X, ZHONG H, LIU W, et al. Multi-sensor fusion SLAM in complex terrain environments[J]. Robot, 2024, 46(4): 425-435.
- [20] 姚二亮, 宋海涛, 赵婧, 等. 考虑点云结构和表面信息的激光雷达-惯性 SLAM 算法[J]. 机器人, 2024, 46(4): 436-449.
- YAO E L, SONG H T, ZHAO J, et al. LiDAR-inertial SLAM algorithm based on point cloud structure and appearance[J]. Robot, 2024, 46(4): 436-449.
- [21] WALCOTT-BRYANT A, KAESE M, JOHANSSON H, et al. Dynamic pose graph SLAM: Long-term mapping in low dynamic environments[C]//IEEE/RSJ International Conference on Intelligent Robots and Systems. Piscataway, USA: IEEE, 2012: 1871-1878.
- [22] JOHANSSON H, KAESE M, FALLON M, et al. Temporally scalable visual SLAM using a reduced pose graph[C]//IEEE International Conference on Robotics and Automation. Piscataway, USA: IEEE, 2013: 54-61.
- [23] ZHAO M, GUO X, SONG L, et al. A general framework for lifelong localization and mapping in changing environment[C]//IEEE/RSJ International Conference on Intelligent Robots and Systems. Piscataway, USA: IEEE, 2021: 3305-3312.
- [24] CARLEVARIS-BIANCO N, KAESE M, EUSTICE R M. Generic node removal for factor-graph SLAM[J]//IEEE Transactions on Robotics, 2014, 30(6): 1371-1385.
- [25] SHAN T X, ENGLLOT B, MEYERS D, et al. LIO-SAM: Tightly-coupled Lidar inertial odometry via smoothing and mapping [C]//IEEE/RSJ International Conference on Intelligent Robots and Systems. Piscataway, USA: IEEE, 2020: 5135-5142.
- [26] ZHANG X P, CHENG Z L. Curvature estimation of 3D point cloud surfaces through the fitting of normal section curvatures [DB/OL]. (2018-12-30) [2024-09-23]. <https://api.semanticscholar.org/CorpusID:10468338>.
- [27] KIM G, KIM A. Remove, then revert: Static point cloud map construction using multiresolution range images[C]//IEEE/RSJ International Conference on Intelligent Robots and Systems. Piscataway, USA: IEEE, 2020: 10758-10765.
- [28] KIM G, PARK Y S, CHO Y, et al. MulRan: Multimodal range dataset for urban place recognition[C]//IEEE International Conference on Robotics and Automation. Piscataway, USA: IEEE, 2020: 6246-6253.
- [29] SHAN T X, ENGLLOT B. LeGO-LOAM: Lightweight and ground-optimized lidar odometry and mapping on variable terrain[C]//IEEE/RSJ International Conference on Intelligent Robots and Systems. Piscataway, USA: IEEE, 2018: 4758-4765.
- [30] XU W, CAI Y X, HE D J, et al. FAST-LIO2: Fast direct LiDAR-inertial odometry[J]. IEEE Transactions on Robotics, 2022, 38(4): 2053-2073.
- [31] FAN H Q, SU H, GUIBAS L. A point set generation network for 3D object reconstruction from a single image[C]//IEEE Conference on Computer Vision and Pattern Recognition. Piscataway, USA: IEEE, 2017: 2463-2471.

作者简介:

方宝富 (1978-), 男, 博士, 副教授。研究领域: 智能机器人和多智能体系统。