

一种自适应的 Graph SLAM 鲁棒闭环算法

张国良, 姚二亮, 汤文俊, 岳亚南

第二炮兵工程大学, 陕西 西安 710025

通信作者: 汤文俊, familyyao915@gmail.com 收稿/录用/修回: 2014-05-28/2014-07-21/2014-12-18

摘要

针对在常规 Graph SLAM(simultaneous location and mapping)算法中后端优化无法高效排除错误闭环影响的问题, 提出一种自适应的 Graph SLAM 鲁棒闭环算法. 通过分析代价函数中尚未确定的参数对优化过程的影响, 根据迭代得到的最新信息, 对这些参数进行更新, 从而加快算法收敛速度, 并对不同的数据集有很好的适应性. 利用公开的数据集对算法进行实验, 通过对比发现, 在添加不同类型、不同数量的异常闭环条件下, 本文算法对不同数据集具有良好适应性且收敛速度较快, 证明了算法的有效性.

关键词

同步定位与地图构建
Graph SLAM
后端优化
鲁棒闭环
中图分类号: TP242
文献标识码: A

An Adaptive Robust Loop Closure Algorithm for Graph SLAM

ZHANG Guoliang, YAO Erliang, TANG Wenjun, YUE Ya'nan

The Second Artillery Engineering University, Xi'an 710025, China

Abstract

We propose an adaptive robust loop closure algorithm for the Graph SLAM to address the problem where the back-end for conventional Graph SLAM obviates the influence of false loops efficiently. The influence of indefinite parameters in the cost function to the optimization procedure is analyzed. The parameters are renewed by the latest information obtained from iterations to speed the convergence rate. The algorithm is adaptive to different datasets. The experiment is performed for the proposed algorithm with public datasets. The comparison results show that the proposed algorithm is adaptive to different datasets with different types and numbers of outliers and the convergence rate is higher, which verifies the efficiency of the algorithm.

Keywords

simultaneous localization and map construction (SLAM);
Graph SLAM;
back-end optimization;
robust loop closure

1 引言

在未知环境中, 机器人的自定位与环境创建是机器人执行其它任务的基础. 机器人要根据带有噪声的传感器观测建立周围环境的地图, 同时根据已建地图估计机器人在环境中的位姿. 多年来基于滤波的 SLAM 方法得到很大发展, 常用的滤波方法包括扩展卡尔曼滤波(EKF)方法^[1]、扩展信息滤波(EIF)方法^[2]及粒子滤波(PF)方法^[3]等. 然而, 基于滤波器的方法存在线性化及更新效率等问题, 使得它难以应用于大规模环境的地图创建^[4].

Lu 与 Milios 首先提出基于图优化的 SLAM 方法(Graph SLAM)^[5], 然而受限于计算方法, 无法满足实时性要求. 随着高效求解方法的出现, Graph SLAM 成为当前 SLAM 研究的热点^[6-9]. 近年来, 涌现了许多优秀的 Graph SLAM 算法, 如 TreeMap^[6]、TORO (tree-based network optimizer)^[7]、HOG-Man^[8]及 g²o^[9]. 然而不可避免地, 尽管有一些学者对 Graph SLAM 前端的闭环产生和闭环确认提出了

改进方法^[10-12], 但仍然会不同概率地出现错误闭环, 这些错误闭环使常规 Graph SLAM 算法不能收敛或者收敛到错误的结果. 在大范围闭合环境中剔除数据关联引入的错误闭环成为了机器人 SLAM 的一个必要功能, 称为鲁棒闭环^[13]. 为此, Latif^[13]提出了利用闭环集群内部及外部的一致性检验来排除错误闭环的 RRR (realizing, reversing, recovering)算法, 但在大量错误闭环环境下算法耗费时间较长, 实效性不好. Olson^[14]采用一种多模态的高斯模型作为鲁棒的代价函数, 为每个闭环添加一个无效假设, 在优化过程中判定闭环是否有效, 但在大量错误闭环环境下算法召回率较低. Sünderhauf^[15]提出给每个闭环约束添加一个转换变量, 控制约束对整个图的影响, 在优化过程中调整机器人位姿和转换变量, 使鲁棒的代价函数最小. 这种算法的关键是将闭环约束看作可转换的约束类型, 因此称为 SC (switchable constrain)算法. 然而此方法需要同时对两种参量进行优化, 并手动设定一个常量, 迭代次数较多.

针对以上问题, 在 Sünderhauf 思想的基础上, 本文提

出一种自适应的鲁棒闭环代价函数. 通过分析鲁棒闭环代价函数中各项参数对优化过程的具体影响, 确定代价函数中的转换变量、原先需要手动设定的参量与其它可计算量的关系和计算方法, 从而可以较好地运用于不同环境下的数据集, 具有较强的自适应性.

2 鲁棒闭环 Graph SLAM 算法

在 Graph SLAM 算法中, 机器人位姿作为图中的节点, 里程计约束以及闭环约束作为图中的边约束. 令 $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_n)^T$ 为机器人位姿矩阵, $\mathbf{x}_i$ 为机器人在时刻 i 的位姿. 约束边为 $U = \{U_{\text{odom}}, U_{\text{loop}}\} = \{u_{mn}\}$, 包括里程计约束 $U_{\text{odom}} = \{u_{kl}\} (l = k + 1)$ 及闭环约束 $U_{\text{loop}} = \{u_{ij}\}$. Graph SLAM 图结构如图 1.

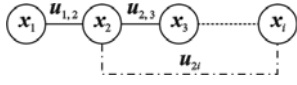


图 1 Graph SLAM 图结构

Fig.1 The structure of Graph SLAM graph

常规 Graph SLAM 算法由前端 (front-end) 和后端 (back-end) 组成, 前端根据外部传感器输入的原始数据来确定图结构的节点位姿和边约束, 后端则根据前端得到的节点和边信息, 利用非线性最小二乘等方法对图结构进行优化, 得到满足约束的最优节点位姿. Graph SLAM 算法框架见图 2.

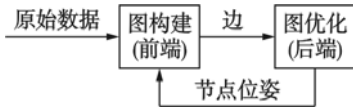


图 2 Graph SLAM 算法框架

Fig.2 The structure of Graph SLAM algorithm

令 γ_{mn} 和 Σ_{mn} 分别为从节点 m 到节点 n 的约束边 u_{mn} 的观测均值向量及协方差阵. 在给定节点状态 $\mathbf{x}$ 和边约束 u_{mn} 下计算约束边引入的剩余误差 e_{mn} :

$$e_{mn}(\mathbf{x}) = \gamma_{mn} - f_{mn}(\mathbf{x}) \quad (1)$$

其中 $f_{mn}(\mathbf{x})$ 为利用节点位姿得到从节点 m 到节点 n 的转移向量. 所以约束 u_{mn} 的引入误差为

$$d_{mn}^2(\mathbf{x}) = e_{mn}^T(\mathbf{x}) \Sigma_{mn}^{-1} e_{mn}(\mathbf{x}) \quad (2)$$

假定所有约束是独立的, 得到所有边约束的引入误差为

$$D^2(\mathbf{x}) = \sum d_{mn}^2(\mathbf{x}) = \sum e_{mn}^T(\mathbf{x}) \Sigma_{mn}^{-1} e_{mn}(\mathbf{x}) \quad (3)$$

对于式(3), 常规的 Graph SLAM 算法通常采用 Gauss-Newton 或 Levenberg-Marquadt 等非线性最小二乘方法, 找到使所有边引入误差最小的节点 $\mathbf{X}^*$:

$$\mathbf{X}^* = \underset{\mathbf{X}}{\operatorname{argmin}} \sum e_{mn}^T(\mathbf{x}) \Sigma_{mn}^{-1} e_{mn}(\mathbf{x}) \quad (4)$$

显而易见, 当约束中存在错误闭环时, 采用常规 Graph SLAM 算法完全考虑错误闭环对整个图优化的影响, 必定会造成算法结果与真实情况相距甚远. 由此, S nderhauf 提出的 SC 算法, 将里程计约束 u_{kl} 与闭环约束 u_{ij} 分开考虑, 从而得到里程计约束项和闭环约束项, 并为

闭环约束附加转换变量 $s_{ij} \in [0, 1]$, 在优化过程中通过调整 s_{ij} 的大小以控制闭环约束对节点的影响, 同时加入附加项 $w \|1 - s_{ij}\|^2$ 以避免闭环约束误差项被完全消除, 无法利用正确闭环修正机器人位姿^[15]. SC 算法思想如下式所示:

$$\begin{aligned} \mathbf{X}^*, \mathbf{S}^* = \underset{\mathbf{X}, \mathbf{S}}{\operatorname{argmin}} & \left(\sum_k \|f(\mathbf{x}_k, \mathbf{u}_{kl}) - \mathbf{x}_l\|_{\Theta_{kl}}^2 + \right. \\ & \sum_{ij} \|s_{ij}(f(\mathbf{x}_i, \mathbf{u}_{ij}) - \mathbf{x}_j)\|_{\Lambda_{ij}}^2 + \\ & \left. \sum_{ij} w \|1 - s_{ij}\|^2 \right) \end{aligned} \quad (5)$$

其中, $\Theta = [\Theta_{ij}]$, $\Lambda = [\Lambda_{ij}]$, 分别为里程计协方差阵和观测协方差阵; $f(\mathbf{x}_i, \mathbf{u}_{kl})$ 为通过节点 $\mathbf{x}_i$ 和约束边 u_{kl} 获得的 $\mathbf{x}_l$ 估计值; w 为手动设定的常量.

然而 SC 算法在优化过程中要同时对参量 $\mathbf{X}$ 和 $\mathbf{S}$ 进行迭代计算, 而每次迭代过程中参量并非沿着误差最小方向发展, 就需要多次迭代才能得到较优解. 另外, 算法对于不同数据集中附加项设定的常量 w 也没有给出进一步的说明, 只是采用多次实验的方法确定一个合适的 w 值, 这在真实的未知环境的 SLAM 实验中并不现实. 在 SC 算法实验过程中发现, 不同的 w 值对实验结果的准确性具有一定影响, 甚至会导致算法收敛到错误结果. 因此得出结论, 采用固定 w 值的 SC 算法无法得到最优效果.

3 具有自适应性的鲁棒闭环 Graph SLAM 算法

由于 SC 算法没有利用数据集信息对算法内部的参量进行确定, 算法不能很好地适应不同数据集的计算需求, 导致算法收敛速度较慢, 甚至不能收敛到正确的结果. 为进一步考虑代价函数中的转换变量 s 和常量 w 对算法效果的影响, 单独对一个闭环边 u_{ab} 的约束项及其附加项进行分析, 这里将式(5)改写为

$$\begin{aligned} \mathbf{X}^*, \mathbf{S}^* = \underset{\mathbf{X}, \mathbf{S}}{\operatorname{argmin}} & \left(\sum_k \|f(\mathbf{x}_k, \mathbf{u}_{kl}) - \mathbf{x}_l\|_{\Theta_{kl}}^2 + \right. \\ & \sum_{ij \neq ab} \|s_{ij}(f(\mathbf{x}_i, \mathbf{u}_{ij}) - \mathbf{x}_j)\|_{\Lambda_{ij}}^2 + \\ & \sum_{ij \neq ab} w_{ij} \|1 - s_{ij}\|^2 + \\ & \left. \underbrace{\|s_{ab}(f(\mathbf{x}_a, \mathbf{u}_{ab}) - \mathbf{x}_b)\|_{\Lambda_{ab}}^2 + w_{ab} \|1 - s_{ab}\|^2}_{h(s_{ab}, \chi_{ab}^2, w_{ab})} \right) \end{aligned} \quad (6)$$

得到边 u_{ab} 的闭环约束项及其附加项之和 $h(s_{ab}, \chi_{ab}^2, w_{ab})$ 为

$$h(s_{ab}, \chi_{ab}^2, w_{ab}) = s_{ab}^2 \chi_{ab}^2 + w_{ab} (1 - s_{ab})^2 \quad (7)$$

其中 $\chi_{ab}^2 = \|f(\mathbf{x}_a, \mathbf{u}_{ab}) - \mathbf{x}_b\|_{\Lambda_{ab}}^2$.

当式(6)取得最优解时, 其代价函数关于所有 s 的偏导数为 0, 即对 s_{ab} 的偏导也为 0. 由于代价函数中只有 h 含有 s_{ab} 项, 因此其对转换变量 s_{ab} 的偏导数为 0. 为具有一般性, 下面省去下标 ab , h 对 s 求导, 使其导数为 0, 得到此时 s 值为

$$s = \frac{w}{\chi^2 + w} \quad (8)$$

由此得到参量 s 的计算方法.

在式(8)中, χ^2 可通过计算求得, w 未知. 令 $\chi' =$

$\|f(x_a, u_{ab}) - x_b\|^2$, 考虑到在优化过程中 w 与 χ' 、 Λ 具有一定联系, 令 $w = g(\chi', \Lambda)$. 分析式(8)可知:

$$s \propto g(\chi', \Lambda) \quad (9)$$

对 $g(\chi', \Lambda)$ 中 χ' 的考虑: 当闭环正确时, 此闭环约束应作为图优化的重要因素, 在优化过程中得到重视, 即其 s 值应趋近于最大值 1, 而此时通过计算得到的误差值 χ' 较小, 因此 s 应与 χ' 保持负相关关系, 见式(10), 同样闭环错误时亦是如此:

$$s \propto g(\chi', \Lambda) \propto \frac{1}{\chi'} \quad (10)$$

对 $g(\chi', \Lambda)$ 中 Λ 的考虑: 当闭环正确时, s 值应趋近于最大值 1. 而对于一个正确闭环来说, 其 Λ 值通常来说较小. 因此较大的 Λ 通过函数 $g(\chi', \Lambda)$ 应计算得到较小的 s 值, 即 s 与 Λ 为负相关关系, 见式(11), 同样对于错误闭环中的优化也是如此:

$$s \propto g(\chi', \Lambda) \propto \frac{1}{|\Lambda|} \quad (11)$$

考虑到 s 与 χ' 、 Λ 存在的内在联系, 这里采用以下函数计算 w 值:

$$w = g(\chi', \Lambda) = \frac{1}{\chi' |\Lambda|} \quad (12)$$

将式(12)代入式(8), 可得 s 的计算为

$$s = \frac{1}{\chi'^2 |\Lambda| + 1} \quad (13)$$

所以, 最后具有参数自适应性的 Graph SLAM 鲁棒闭环算法描述为

$$X^* = \operatorname{argmin}_X \left(\sum_k \|f(x_k, u_{kl}) - x_l\|_{\Theta_{kl}}^2 + \sum_{ij} \|s_{ij}(f(x_i, u_{ij}) - x_j)\|_{\Lambda_{ij}}^2 + \sum_{ij} w_{ij} \|1 - s_{ij}\|^2 \right) \quad (14)$$

其中 s_{ij} 和 w_{ij} 分别通过式(13)和式(12)求得.

由此, 得到了对每个闭环约束 u_{ij} 计算 s_{ij} 和 w_{ij} 的方法, 使得算法只需对节点 x_{ij} 进行搜索. 通过在每次迭代中应用 Λ 和最新的 χ^2 、 χ' 对 w_{ij} 和 s_{ij} 的更新, 动态地改变约束边对节点的影响, 使正确闭环能够逐步矫正节点位置, 并大大减弱错误闭环在整个优化过程中的影响力. 因此本文提出的算法可看作一种自适应的 SC 算法(ASC).

4 实验与分析

为进行算法对比, 对公开的数据集 Manhattan3500、City10000、Bicocca 进行仿真. 对于 Manhattan3500 合成数据集, 采用初始位姿精度不同的 Manhattan3500Olson^[16] 数据集和 Manhattan3500g²o^[9] 数据集, 以此检验初始位姿精度对算法收敛时间的影响. 从最后算法运行结果上看, 两种数据集结果相似性较高. City10000 合成数据集从 iSAM 功能包^[17] 中获得. Bicocca 真实数据集从 RRR^[13] 中获得. 4 个数据集节点与边的个数见表 1, 机器人真实节点见图 3. 图 4 为原始数据中机器人节点与边.

为了检验算法对前端产生的不同类型错误闭环的适应性, 采用文[15]的方法对每个数据分别添加 4 种类型的异

常闭环, 即随机(random)闭环、局部(local)闭环、随机组(random grouped)闭环及局部组(local grouped)闭环. 随机闭环在图中随机选取 2 个节点添加约束边. 局部闭环随机在局部区域添加闭环. 组闭环则是添加具有一致性的若干闭环约束. 采用这 4 种闭环用于模拟在真实环境中可能出现的错误闭环. 这里使用 Vertigo 包^[18] 提供的添加异常闭环工具.

表 1 数据集中节点与边个数

Tab.1 The number of nodes and edges in the datasets

数据集	节点数	边个数
City10000	10 000	20 687
Bicocca	8 358	8 405
Manhattan_Olson	3 500	5 598
Manhattan_g2o	3 500	5 598

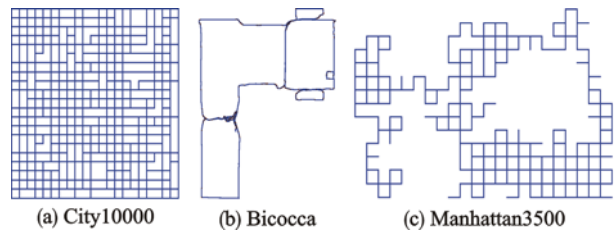


图 3 数据集节点真实位姿

Fig.3 True nodes' poses of the datasets

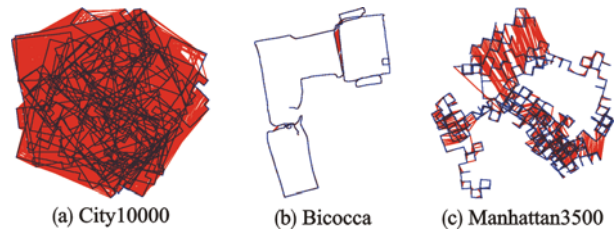


图 4 数据集原始节点(蓝)与边(红)

Fig.4 Raw nodes (blue) and edges (red) of the datasets

对数据集添加 500 个异常闭环约束后, 利用常规 Graph SLAM 算法对含有异常闭环的数据集进行运算, 得到节点和边的配置, 如图 5. 可见常规的 Graph SLAM 算法对闭环依赖性高, 一旦闭环错误就会导致灾难性后果. 因此有必要应用 Graph SLAM 的鲁棒闭环算法.

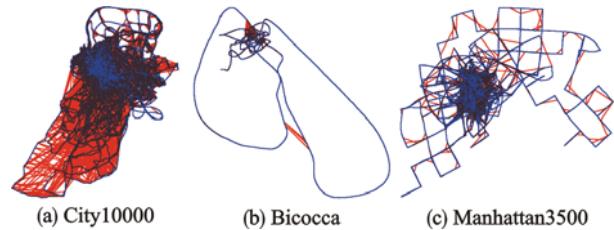


图 5 异常闭环结果节点(蓝)和边(红)

Fig.5 Nodes (red) and edges (blue) of the outliers result

本文对 4 个数据集分别添加 4 种不同类型(R、L、RG、LG)、不同数量(1 000, 2 000, 3 000, 4 000)的异常闭环. 采用 SC 算法和本文的 ASC 算法对数据集进行计算.

在 SC 算法中设定 $w = 1^{[15]}$. 迭代优化时, 代价函数变化量小于阈值时算法停止. 得到的两种算法的收敛时间见表 2.

由于两种算法得到的结果类似, 这里给出在 1 000 个随机异常闭环下 ASC 算法得到的节点与边的结果, 见图 6.

表 2 实验收敛时间对比(单位: s)

Tab.2 Convergence time comparison among the experiments (unit: s)

数据集	1 000	2 000	3 000	4 000
算法	R, L, RG, LG	R, L, RG, LG	R, L, RG, LG	R, L, RG, LG
City10000_SC	40.5, 32.4, 50.4, 25.2	95.6, 33.8, 84.1, 31.4	218.4, 45.1, 123.4, 41.8	320.2, 47.6, 240.6, 53.4
City10000_ASC	15.6, 8.5, 9.4, 4.2	40.1, 9.2, 23.5, 5.3	65.1, 12.4, 31.1, 8.2	206.9, 11.7, 180.1, 10.8
Bicocca_SC	-, -, -, -	-, -, -, -	-, -, -, -	-, -, -, -
Bicocca_ASC	2.1, 0.9, 1.7, 1.0	2.5, 1.1, 2.4, 1.4	3.7, 1.7, 3.4, 1.3	5.1, 2.7, 6.8, 2.4
Manhattan_Olson_SC	12.6, 2.3, 9.7, 2.4	19.7, 2.4, 16.5, 2.1	28.6, 3.4, 32.6, 3.6	45.6, 4.5, 38.2, 4.8
Manhattan_Olson_ASC	6.4, 1.2, 3.9, 1.6	7.5, 2.1, 8.4, 1.9	19.6, 2.3, 21.9, 3.2	24.8, 3.1, 20.7, 4.4
Manhattan_g2o_SC	3.8, 1.7, 3.5, 1.2	7.6, 2.6, 4.5, 2.8	9.4, 3.2, 11.2, 5.1	12.6, 3.4, 12.1, 6.7
Manhattan_g2o_ASC	2.1, 1.1, 2.3, 0.9	4.5, 1.7, 3.2, 1.6	6.5, 1.9, 5.7, 2.6	8.4, 2.2, 7.5, 4.1

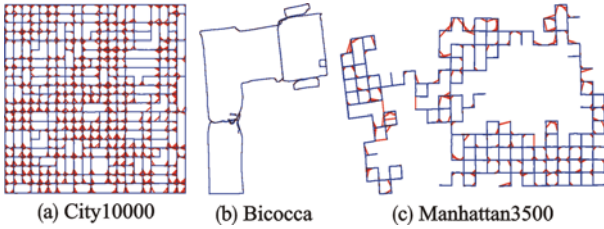


图 6 ASC 算法结果节点(蓝)和边(红)

Fig.6 Nodes (red) and edges (blue) of the ASC result

对比图 3 与图 6 可发现, City10000 节点拓扑正确性较高, 而 Bicocca 和 Manhattan3500 则发生一定程度的拓扑倾斜现象. 这是由于原始数据中部分节点误差较大, 且缺少足够有效的闭环约束来纠正里程计误差. 如图 4 中 Bicocca 数据集, 可发现其存在“绑架”现象, 机器人轨迹发生跳变. 而对于 Manhattan3500 数据集, 观察图 7 发现, 原始数据集中只有 A 区、B 区和 C 区存在有效的闭环约束, 它们之间的连接部分 D 区却没有足够有效的闭环约束. 从算法的结果来看, 正是 D 区中的节点偏差导致节点拓扑产生倾斜.

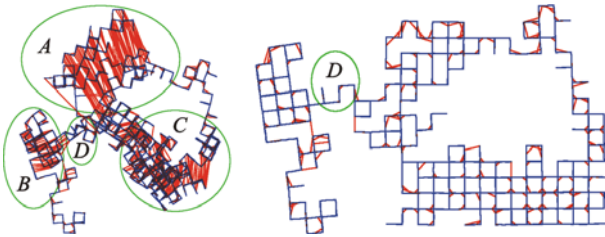


图 7 Manhattan 原始数据集与优化数据集图结构

Fig.7 The graph structure of the raw Manhattan dataset and the optimized dataset

从表 2 可见, 在不同类型、不同数量的闭环异常下, ASC 算法的收敛时间明显优于 SC 算法. 对于初始节点位姿精度不同的两种 Manhattan3500 数据集, 相比 Olson 版本, g²o 版本收敛时间较短, 可见对于 Graph SLAM 来说, 其前端的精确度是必须考虑的因素. 两种算法在处理随机性的异常闭环(R, RG)时, 收敛时间随异常闭环数量上升

都有很大增长. 这是由于随机性错误闭环对整个图结构产生较大影响, 使节点位姿几乎完全偏离真实位姿. 因此需要 SC 算法和 ASC 算法通过较多次数的迭代优化过程才能达到收敛. 而局部性错误闭环只对局部区域的节点位姿产生负面影响, 虽然使节点偏离真实情况, 但仍处在其附近区域, 相对来说需要的迭代次数较少. 但总体说, 无论对于随机性错误闭环还是局部性错误闭环, ASC 算法收敛更快. 而对于 Bicocca 数据集, 在标准参数设置下 ($w = 1$), SC 算法不能得到有效的图结构.

由于一个节点的位姿误差会对接下来的所有节点的位姿都产生影响, 采用 RMSE (root mean square error) 值在评价节点图时不能有效评价其局部一致性, 因此本文使用 RPE (relative pose error) 值衡量算法结果与真实数据的差距. 其计算方法如下:

$$RPE = \frac{1}{n} \sum_{kl} (\delta_{kl} - \hat{\delta}_{kl})^2 \quad (15)$$

其中 δ_{kl} 为通过计算得到的约束边两个端点 $\mathbf{x}_k$ 与 $\mathbf{x}_l$ 之间的相对转移量, $\hat{\delta}_{kl}$ 为真实节点 $\hat{\mathbf{x}}_k$ 与 $\hat{\mathbf{x}}_l$ 之间的相对转移量.

由于 ASC 和 SC 算法结果类似(除 Bicocca 数据集), 这里只给出在不同类型、不同数量的异常闭环下, ASC 算法的 RPE 值, 见图 8. 对于 City10000, Manhattan3500 和 Bicocca 数据集, ASC 算法都可以较好地得到比较精确的图结构. 通过实验发现, 对于随机性的异常闭环(R, RG), SC 和 ASC 算法的 RPE 值随异常闭环数量上升都有一定增长, 但相比常规 Graph SLAM 算法依然能够保持节点拓扑的一致性. 而在对局部性闭环(L, LG)的实验中, SC 和 ASC 算法都可以达到非常好的效果, RPE 值在不同异常闭环数量下都能够保持相对稳定(10^{-4} m ~ 10^{-5} m), 可见算法对局部性异常闭环的影响有很好的消除作用, 体现了算法的有效性.

5 总结

本文在 SC 算法的基础上, 对代价函数中的参量进行定性和定量分析, 利用最新迭代信息更新这些参量, 动态

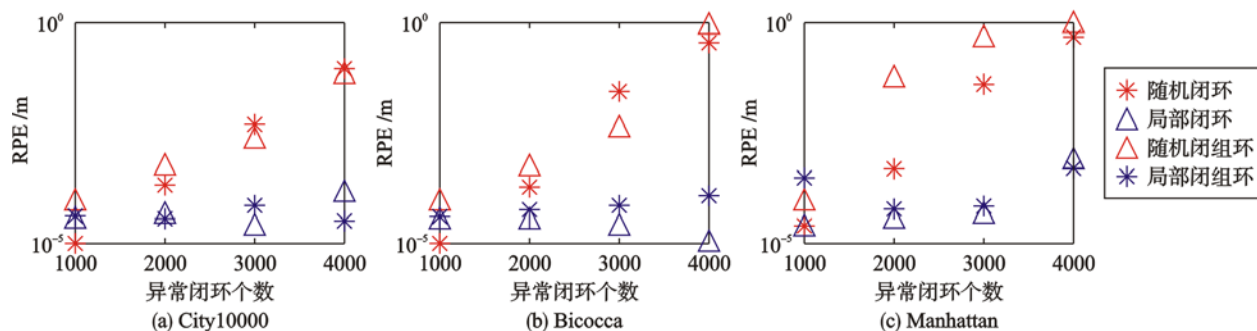


图8 ASC算法的RPE值

Fig.8 RPE of the ASC algorithm

调整闭环对节点位姿的影响,将错误闭环的影响力大大降低,从而加快算法收敛速度,且对不同的数据集有良好的适应性.另外,实验表明ASC算法在处理局部性异常闭环时具有较大的优越性,对随机性异常闭环的数据集在收敛

速度上有一定的优势.而Bicocca数据集的对比实验表明,ASC算法更具鲁棒性,适应性更强.下一步将继续研究在随机性异常闭环条件下如何进一步提高算法实效性和精确度,从而加强算法的有效性.

参考文献

- [1] Bailey T, Durrant-Whyte H. Simultaneous localization and mapping (SLAM): Part I[J]. IEEE Robotics & Automation Magazine, 2006, 13(2): 99-108.
- [2] Smith R C, Cheeseman P. On the representation and estimation of spatial uncertainty[J]. The International Journal of Robotics Research, 1986, 5(4): 56-68.
- [3] 祝继华, 郑南宁, 袁泽剑, 等. 基于中心差分粒子滤波的SLAM算法[J]. 自动化学报, 2010, 36(2): 249-257.
Zhu J H, Zheng N N, Yuan Z J, et al. A SLAM algorithm based on central difference particle filter[J]. Acta Automatic Sinica, 2010, 36(2): 249-257.
- [4] Huang S, Dissanayake G. Convergence and consistency analysis for extended Kalman filter based SLAM[J]. IEEE Transactions on Robotics, 2007, 23(5): 1036-1049.
- [5] Lu F, Milios E. Globally consistent range scan alignment for environment mapping[J]. Autonomous Robots, 1997, 4(4): 333-349.
- [6] Frese U, Schroder L. Closing a million-landmarks loop[C]//2006 IEEE/RSJ International Conference on Intelligent Robots and Systems. Piscataway, NJ, USA: IEEE, 2006: 5032-5039.
- [7] Grisetti G, Stachniss C, Burgard W. Nonlinear constraint network optimization for efficient map learning[J]. IEEE Transactions on Intelligent Transportation Systems, 2009, 10(3): 428-439.
- [8] Grisetti G, Kummerle R, Stachniss C, et al. Hierarchical optimization on manifolds for online 2D and 3D mapping[C]//2010 IEEE International Conference on Robotics and Automation. Piscataway, NJ, USA: IEEE, 2010: 273-278.
- [9] Kummerle R, Grisetti G, Strasdat H, et al. g²o: A general framework for graph optimization[C]//2011 IEEE International Conference on Robotics and Automation. Piscataway, NJ, USA: IEEE, 2011: 3607-3613.
- [10] Tipaldi G D, Braun M, Arras K O. FLIRT: Interest regions for 2D range data with applications to robot navigation[C]//Experimental Robotics. Berlin, Germany: Springer, 2014: 695-710.
- [11] Olson E. Recognizing places using spectrally clustered local matches[J]. Robotics and Autonomous Systems, 2009, 57(12): 1157-1172.
- [12] Cummins M, Newman P. FAB-MAP: Probabilistic localization and mapping in the space of appearance[J]. The International Journal of Robotics Research, 2008, 27(6): 647-665.
- [13] Latif Y, Cadena C, Neira J. Robust loop closing over time for pose graph SLAM[J]. The International Journal of Robotics Research, 2013, 32(14): 1611-1626.
- [14] Olson E, Agarwal P. Inference on networks of mixtures for robust robot mapping[J]. The International Journal of Robotics Research, 2013, 32(7): 826-840.
- [15] Sinderhauf N, Protzel P. Switchable constraints for robust pose graph slam[C]//2012 IEEE/RSJ International Conference on Intelligent Robots and Systems. Piscataway, NJ, USA: IEEE, 2012: 1879-1884.
- [16] Olson E, Leonard J, Teller S. Fast iterative alignment of pose graphs with poor initial estimates[C]//2006 IEEE International Conference on Robotics and Automation. Piscataway, NJ, USA: IEEE, 2006: 2262-2269.
- [17] Kaess M, Ranganathan A, Dellaert F. iSAM: Fast incremental smoothing and mapping with efficient data association[C]//2007 IEEE International Conference on Robotics and Automation. Piscataway, NJ, USA: IEEE, 2007: 1670-1677.

(下转第327页)

- [3] Chebouba A, Yalaoui F. Optimization of natural gas pipeline transportation using ant colony optimization[J]. Computer & Operations Research, 2009, 36(6): 1916–1923.
- [4] 李博, 何森, 李江飞, 等. 求解天然气管网运行优化问题的 TDDP 算法[J]. 石油工程建设, 2014, 40(2): 6–10.
Li B, He M, Li J F, et al. A TDDP algorithm for optimizing natural gas pipeline network operation[J]. Petroleum Engineering Construction, 2014, 40(2): 6–10.
- [5] Borraz-Sanchez C, Rios-Mercado R Z. A non-sequential dynamic programming approach for natural gas network optimization[J]. WSEAS Transactions on Systems, 2004, 3(4): 1384–1389.
- [6] 张化光, 张欣, 罗艳红, 等. 自适应动态规划综述[J]. 自动化学报, 2013, 39(4): 303–311.
Zhang H G, Zhang X, Luo Y H, et al. An overview of research on adaptive dynamic programming[J]. Acta Automatica Sinica, 2013, 39(4): 303–311.
- [7] Lall H S, Percell P B. A dynamic programming based gas pipeline optimizer[M]//Analysis and Optimization of Systems. Berlin, Germany: Springer, 1990: 123–132.
- [8] Borraz-Sanchez C. Optimization methods for pipeline transportation of natural gas[D]. Norway: University of Bergen, 2010: 56–57.
- [9] Marc C S. On PDE solution in transient optimization of gas networks[J]. Journal of Computational and Applied Mathematics, 2007(203): 345–361.
- [10] 李长俊. 天然气管道输送[M]. 北京: 石油工业出版社, 2008: 61–62.
Li C J. Transportation of natural gas pipeline[M]. Beijing: Petroleum Industry Press, 2008: 61–62.
- [11] Pia D, Bjorn G, Alexander M, et al. Combination of nonlinear and linear optimization of transient gas network[J]. Inform Journal on Computing, 2011, 23(4): 605–617.
- [12] 杨毅. 天然气管道运行优化模型及其寻优方法研究[D]. 成都: 西南石油学院, 2006: 62–67.
Yang Y. Optimization model and solution researches of natural gas transportation networks[D]. Chengdu: Southwest Petroleum University, 2006: 62–67.
- [13] Wu S, Rios-Mercado R Z, Boyd E A, et al. Model relaxations for the fuel cost minimization of steady-state gas pipeline networks[J]. Mathematical and Computer Modeling, 2000, 31(2): 197–220.
- [14] 刘武, 张鹏, 刘稀, 等. 天然气管网优化调度方法研究[J]. 西南石油大学学报, 2009, 31(3): 146–148.
Liu W, Zhang P, Liu X, et al. Algorithm research of natural gas transportation networks optimization[J]. Journal of Southwest Petroleum University, 2009, 31(3): 146–148.
- [15] Lei Y, Jennie S, Konstantinos S, et al. Performance evaluation of direct heuristic dynamic programming using control-theoretic measures[J]. Journal of Intelligent Robotic Systems, 2009, 55(2): 177–201.
- [16] 孙健, 刘锋, 郭文涛, 等. 一种改进的近似动态规划方法及其在 SVC 的应用[J]. 电机与控制学报, 2011, 15(5): 95–102.
Sun J, Liu F, Guo W T, et al. An improved approximate dynamic programming and its application in SVC control[J]. Electric Machines and Control, 2011, 15(5): 95–102.
- [17] 厉洋峰. 动态规划及其在数学模型中的应用[J]. 中国新技术产品, 2009, 15(16): 244–245.
Li Y F. Dynamic programming and its application in mathematic models[J]. China New Technologies and Products, 2009, 15(16): 244–245.
- [18] 胡寿松, 王执铨, 胡维礼. 最优控制理论与系统[M]. 北京: 科学出版社, 2005: 130–157.
Hu S S, Wang Z Q, Hu W L. Optimal control theory and system[M]. Beijing: Science Press, 2005: 130–157.
- [19] Borraz-Sanchez C, Haugland D. Minimizing fuel cost in gas transmission networks by dynamic programming and adaptive discretization[J]. Computers & Industrial Engineering, 2011, 61(2): 364–372.
- [20] Boyd E A, Kim S. Efficient operation of natural gas transmission systems: A network-based heuristic for cyclic structures[J]. Computers and Operations Research, 2006, 33(8): 2323–2351.

作者简介

李立刚(1976–), 副教授. 研究领域为天然气管网运行优化与系统工程.

(上接第 320 页)

- [18] S nderhauf N. Vertigo: Versatile extensions for robust inference using graphical models[DB/OL]. (2013–01–08)[2014–04–12].
<http://openslam.org/vertigo.html>.

作者简介

张国良(1970–), 男, 博士, 教授. 研究领域为智能机器人, 先进控制理论与应用, 组合导航.

姚二亮(1991–), 男, 硕士生. 研究领域为移动机器人 SLAM.

汤文俊(1986–), 男, 博士生. 研究领域为多机器人 SLAM.