Cartographer论文带读

Real-Time Loop Closure in 2D LIDAR SLAM

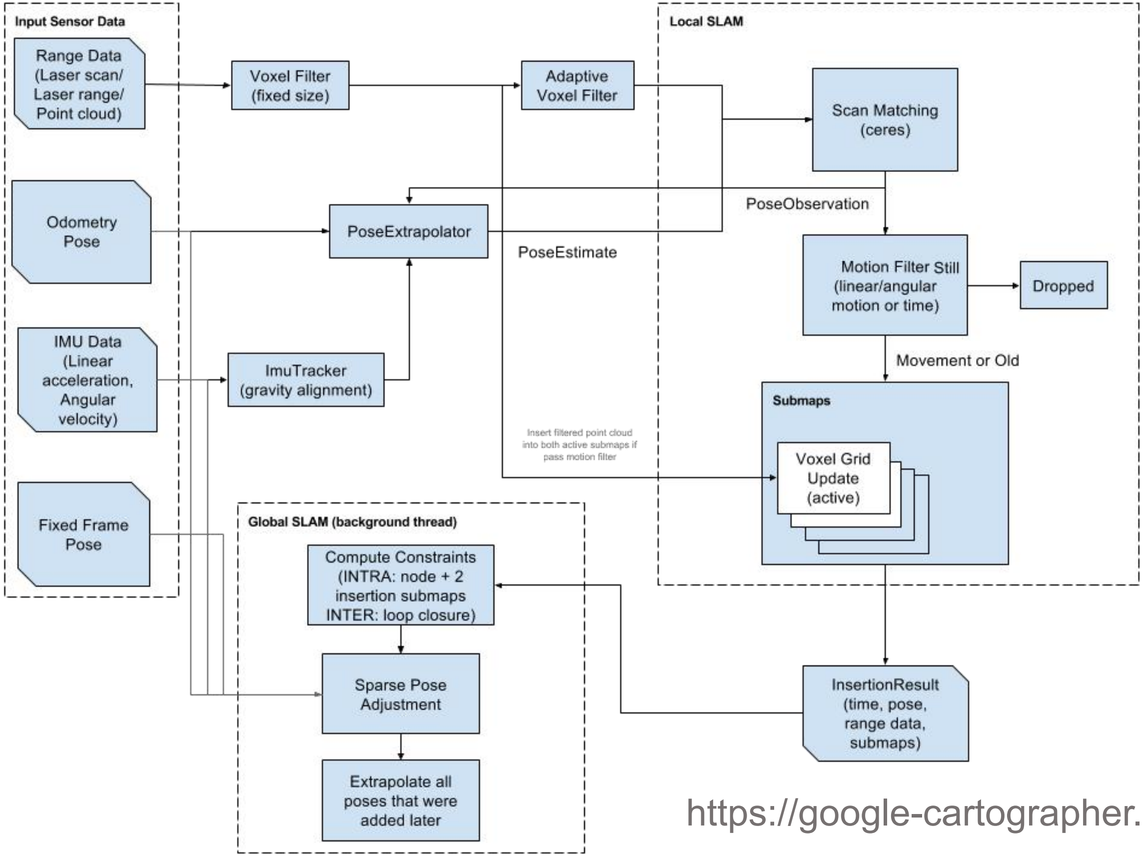
Wolfgang Hess, Damon Kohler, Holger Rapp,Daniel Andor

张涵
2020.04.19

主要内容

- 主要内容
  - Cartographer流程概述
    - Input Sensor Data
    - Local SLAM
    - Global SLAM
  - Real-Time Loop Closure in 2D LIDAR SLAM
    - Abstract
    - Introduction
    - Related work
    - System overview
    - Local 2D SLAM
    - Closing loops (BBA)
    - Conclusions

Cartographer流程概述



https://google-cartographer.readthedocs.io/en/latest/

Real-Time Loop Closure in 2D LIDAR SLAM

计算机视觉life

ABSTRACT

I INTRODUCTION

II RELATED WORK

III SYSTEM OVERVIEW

IV LOCAL 2D SLAM

Cartographer中的前端、扫描匹配，子图的构建

V CLOSING LOOPS

后端优化，BBS

~~VI EXPERIMENTAL RESULTS~~

VII CONCLUSIONS

满足实时、大场景的需求

Real-Time Loop Closure in 2D LIDAR SLAM

Abstract

关键词：便携式平台、实时、闭环、分枝定界加速

Portable laser range-finders, further referred to as LIDAR, and simultaneous localization and mapping (SLAM) are an efficient method of acquiring as-built floor plans. Generating and visualizing floor plans in real-time helps the operator assess the quality and coverage of capture data. **Building a portable capture platform necessitates operating under limited computational resources**. We present the approach used in our backpack mapping platform which achieves **real-time mapping and loop closure at a 5cm resolution**. To achieve real-time loop closure, we use **a branch-and-bound approach** for computing **scan-to-submap matches as constraints**. We provide experimental results and comparisons to other well known approaches which show that, in terms of quality, our approach is competitive with established techniques.

I Introduction

The contribution of this paper is a novel method for **reducing the computational requirements of computing loop closure constraints from laser range data.**

提高实时回环检测的速度 → 分枝定界加速

Real-Time Loop Closure in 2D LIDAR SLAM

II Related work

Scan Matching

scan to scan

ICP　PL-ICP

accumulate error quickly

不适合大场景应用

scan to map（本文）

CSM

G-N需良好的初始位姿

高频LiDAR数据

IMU

Pixel-accurate scan

精度高

计算量大

address the remaining local error accumulation

NEXT

~~粒子滤波~~

图优化

III System overview *funny points*

By completing the optimization every few seconds, the experience of an operator is that loops are closed immediately when a location is revisited. This leads to the **soft real-time constraint★** that the loop closure scan matching has to happen quicker than new scans are added, otherwise it falls behind noticeably. We achieve this by using **a branch-and-bound approach** and ★**several precomputed grids per finished submap.**

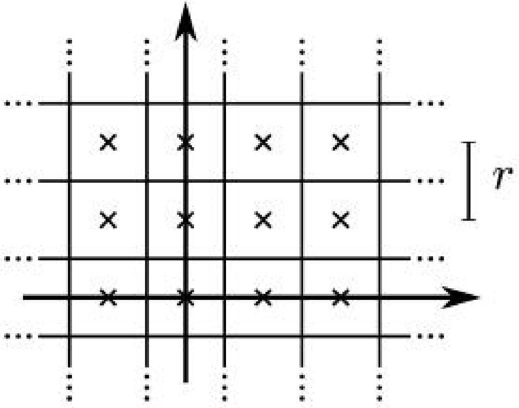
Real-Time Loop Closure in 2D LIDAR SLAM

计算机视觉life

IV Local 2D SLAM

scans

$$T_\xi p = \underbrace{\begin{pmatrix} \cos\xi_\theta & -\sin\xi_\theta \\ \sin\xi_\theta & \cos\xi_\theta \end{pmatrix}}_{R_\xi} p + \underbrace{\begin{pmatrix} \xi_x \\ \xi_y \end{pmatrix}}_{t_\xi}$$



submaps

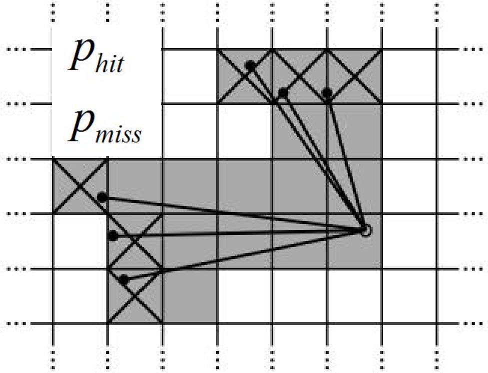
概率网格M

precomputed grids

区间限定函数

$$\text{odds}(p) = \frac{p}{1-p}$$

$$M_{\text{new}}(x) = \boxed{\text{clamp}}(\text{odds}^{-1}(\text{odds}(M_{\text{old}}(x)) \cdot \text{odds}(p_{\text{hit}})))$$



ceres scan matching

$$\underset{\xi}{\text{argmin}} \sum_{k=1}^{K} (1 - \boxed{M_{\text{smooth}}}(T_\xi h_k))^2$$

a smooth version of the probability values

use bicubic interpolation

result [0,1]

require good initial estimates

NEXT CLOSING LOOPS

V Closing loops

A. Optimization problem

Loop closure optimization, like scan matching, is also formulated as **a nonlinear least squares problem** which allows easily adding residuals to take additional data into account.

$$\underset{\Xi^m, \Xi^s}{\operatorname{argmin}} \frac{1}{2} \sum_{ij} \rho\left(E^2\left(\xi_i^{\mathrm{m}}, \xi_j^{\mathrm{s}}; \Sigma_{ij}, \xi_{ij}\right)\right)$$

- submap pose
- scan pose

世界坐标系下

- associated covariance matrices
- relative poses

$$E^2\left(\xi_i^{\mathrm{m}}, \xi_j^{\mathrm{s}}; \Sigma_{ij}, \xi_{ij}\right) = e\left(\xi_i^{\mathrm{m}}, \xi_j^{\mathrm{s}}; \xi_{ij}\right)^T \Sigma_{ij}^{-1} e\left(\xi_i^{\mathrm{m}}, \xi_j^{\mathrm{s}}; \xi_{ij}\right)$$

$$e\left(\xi_i^{\mathrm{m}}, \xi_j^{\mathrm{s}}; \xi_{ij}\right) = \xi_{ij} - \begin{pmatrix} R_{\xi_i^{\mathrm{m}}}^{-1}\left(t_{\xi_i^{\mathrm{m}}} - t_{\xi_j^{\mathrm{s}}}\right) \\ \xi_{i;\theta}^{\mathrm{m}} - \xi_{j;\theta}^{\mathrm{s}} \end{pmatrix}$$

Huber robust error function

$$L_\delta(a) = \begin{cases} \frac{1}{2}a^2 & \text{for } |a| \le \delta \\ \delta\left(|a| - \frac{1}{2}\delta\right), & \text{otherwise} \end{cases}$$

a表述residuals，亦可用y-f(x)表示

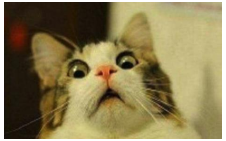
V Closing loops

B. Branch-and-bound scan matching

We are interested in the **optimal**, **pixel-accurate match** where $\mathcal{W}$ is the search window and M_{nearest} is M extended to all of $\mathbb{R}^2$ by rounding its arguments **to the nearest grid point first**, that is **extending the value of a grid points to the corresponding pixel**. The quality of the match can be **improved further** using (CS).

$$\xi^\star = \underset{\boxed{\xi \in \mathcal{W}}}{\operatorname{argmax}} \sum_{k=1}^{K} \boxed{M_{\text{nearest}}}(T_\xi h_k) \quad \text{(BBS)}$$

怎么理解？



Efficiency is improved by carefully choosing step sizes. We choose the angular step size δ_θ so that scan points at the maximum range $d_{\max}$ do not move more than r, the width of one pixel. 计算步长

$$d_{\max} = \max_{k=1,\ldots,K} \|h_k\| \qquad \delta_\theta = \arccos\left(1 - \frac{r^2}{2d_{\max}^2}\right)$$

NEXT

take Naive algorithm 1 for example

V Closing loops

B. Branch-and-bound scan matching

Algorithm 1 Naive algorithm for (BBS)

```
best_score ← −∞
for j_x = −w_x to w_x do
  for j_y = −w_y to w_y do
    for j_θ = −w_θ to w_θ do
      score ← Σ_{k=1}^{K} M_nearest(T_{ξ0+(r j_x, r j_y, δ_θ j_θ)} h_k)
      if score > best_score then
        match ← ξ0 + (r j_x, r j_y, δ_θ j_θ)
        best_score ← score
      end if
    end for
  end for
end for
return best_score and match when set.
```

暴力搜索

We compute an integral number of steps covering given linear and angular search window sizes.

$$W_x = W_y = 7m \qquad W_\theta = 30^\circ$$

$$w_x = \left\lceil \frac{W_x}{r} \right\rceil, \quad w_y = \left\lceil \frac{W_y}{r} \right\rceil, \quad w_\theta = \left\lceil \frac{W_\theta}{\delta_\theta} \right\rceil$$

$$\overline{\mathcal{W}} = \{-w_x, \ldots, w_x\} \times \{-w_y, \ldots, w_y\} \times \{-w_\theta, \ldots, w_\theta\}$$

$$\mathcal{W} = \left\{ \boxed{\xi_0} + \boxed{(r j_x, r j_y, \delta_\theta j_\theta)} : (j_x, j_y, j_\theta) \in \overline{\mathcal{W}} \right\}$$

初始位姿　搜索集中的当前位姿

V Closing loops

B. Branch-and-bound scan matching-main idea

主要思想是将可能性子集表示为树中的节点，其中根节点表示所有可能的解决方案。每个节点的子节点构成其父节点的一个分区，因此它们一起表示相同的可能性集。通过得分进行判断，为了得到一个具体的算法，我们必须确定选择节点、分支和上界计算的方法

NEXT NODE SELECTION

Real-Time Loop Closure in 2D LIDAR SLAM

V Closing loops

B. Branch-and-bound scan matching- Node selection

与广度优先搜索算法不同，深度优先搜索算法类似与树的先序遍历。这种搜索算法所遵循的搜索策略是尽可能“深”地搜索一个图。快速计算多个叶节点。

abdehcfg

算法的效率

好的解决方案

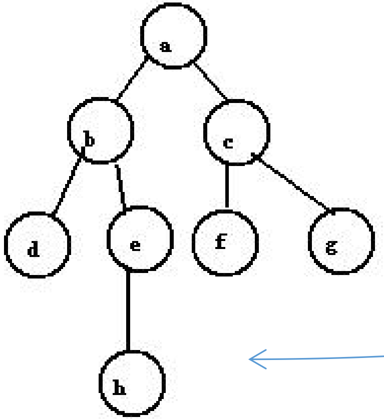
好的上界

去除匹配中比较差的最优解

设定一个分数阈值

实践中，通常不会被超越

算法3



计算得分上界，访问最有希望的子节点

DFS 深度优先搜索

https://blog.csdn.net/m0_37316917/article/details/70880521

V Closing loops

B. Branch-and-bound scan matching- Branching rule

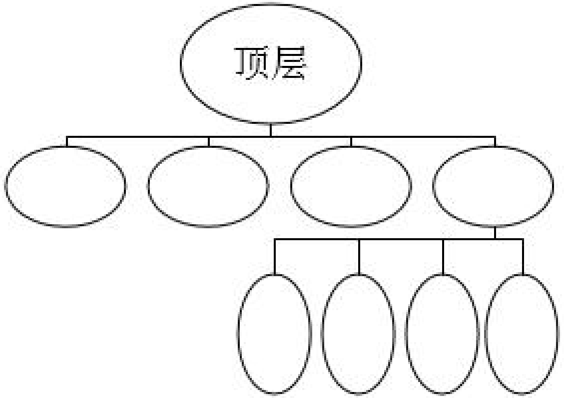
Each node in the tree is described by a tuple of integers $c = (c_x, c_y, c_\theta, c_h) \in \mathbb{Z}^4$ Nodes at height c_h combine up to $2^{c_h} \times 2^{c_h}$ possible translations but represent a specific rotation.Leaf nodes have height $c_h = 0$, and correspond to feasible solutions $\mathcal{W} \ni \xi_c = \xi_0 + (rc_x, rc_y, \delta_\theta c_\theta)$

x_index_offset
y_index_offset
加入新的偏移量

$$\overline{\overline{\mathcal{W}}}_c = \left(\left\{(j_x, j_y) \in \mathbb{Z}^2 : \begin{array}{l} c_x \le j_x < c_x + 2^{c_h} \\ c_y \le j_y < c_y + 2^{c_h} \end{array}\right\} \times \{c_\theta\}\right)$$

角度不变
int scan_index

$$\overline{\mathcal{W}}_c = \overline{\overline{\mathcal{W}}}_c \cap \mathcal{W}.$$



NEXT algorithm 2、3

V Closing loops

B. Branch-and-bound scan matching- Computing upper bounds

$$\begin{aligned} score(c) &= \sum_{k=1}^{K} \max_{j \in \overline{\overline{\mathcal{W}}}_c} M_{\text{nearest}}(T_{\xi_j} h_k) \\ &\geq \sum_{k=1}^{K} \max_{j \in \overline{\mathcal{W}}_c} M_{\text{nearest}}(T_{\xi_j} h_k) \\ &\geq \max_{j \in \overline{\mathcal{W}}_c} \sum_{k=1}^{K} M_{\text{nearest}}(T_{\xi_j} h_k). \end{aligned}$$

Note that, to be able to do this, we also compute the maximum over $\overline{\overline{\mathcal{W}}}_c$ which can be larger than $\overline{\mathcal{W}}_c$ near the boundary of our search space.

$$score(c) = \sum_{k=1}^{K} M_{\text{precomp}}^{c_h}(T_{\xi_c} h_k)$$

$$M_{\text{precomp}}^{h}(x, y) = \max_{\substack{x' \in [x, x+r(2^h-1)] \\ y' \in [y, y+r(2^h-1)]}} M_{\text{nearest}}(x', y')$$

为了有效计算最大值 → precomputed grids $M_{\text{precomp}}^{c_h}$

Real-Time Loop Closure in 2D LIDAR SLAM

V Closing loops

B. Branch-and-bound scan matching- Branching rule

Algorithm 2 Generic branch and bound

$best\_score \leftarrow -\infty$
$\mathcal{C} \leftarrow \mathcal{C}_0$
while $\mathcal{C} \neq \emptyset$ **do**
  Select a node $c \in \mathcal{C}$ and remove it from the set.
  if c is a leaf node **then**
    if $score(c) > best\_score$ **then**
      $solution \leftarrow n$
      $best\_score \leftarrow score(c)$
    end if
  else
    if $score(c) > best\_score$ **then**
      Branch: Split c into nodes $\mathcal{C}_c$.
      $\mathcal{C} \leftarrow \mathcal{C} \cup \mathcal{C}_c$
    else
      Bound.
    end if
  end if
end while
return $best\_score$ and $solution$ when set.

Algorithm 3 DFS branch and bound scan matcher for (BBS)

$best\_score \leftarrow score\_threshold$
Compute and memorize a score for each element in $\mathcal{C}_0$.
Initialize a stack $\mathcal{C}$ with $\mathcal{C}_0$ sorted by score, the maximum score at the top.
while $\mathcal{C}$ is not empty **do**
  Pop c from the stack $\mathcal{C}$.
  if $score(c) > best\_score$ **then**
    if c is a leaf node **then**
      $match \leftarrow \xi_c$
      $best\_score \leftarrow score(c)$
    else
      Branch: Split c into nodes $\mathcal{C}_c$.
      Compute and memorize a score for each element in $\mathcal{C}_c$.
      Push $\mathcal{C}_c$ onto the stack $\mathcal{C}$, sorted by score, the maximum score last.
    end if
  end if
end while
return $best\_score$ and $match$ when set.

看代码

VII Conclusion

In this paper, we presented and experimentally validated **a 2D SLAM system** that combines **scan-to-submap matching** with **loop closure detection** and **graph optimization**. Individual submap trajectories are created using our local, grid-based SLAM approach. In the background, **all scans are matched to nearby submaps using pixel-accurate scan matching to create loop closure constraints.** The constraint graph of submap and scan poses is **periodically optimized** in the background. The operator is presented with an **up-to-date preview** of the final map as a **GPU accelerated** combination of finished submaps and the current submap. We demonstrated that it is possible to run our algorithms on modest hardware **in real-time**.

总结

Thanks

什么是「 SLAM研习社」？

计算机视觉life

- 计算机视觉life读者自发组织，专注于SLAM的开源学习组织
- 宗旨：SLAM技术从基础到应用，懂原理会应用，完全搞透彻，不留死角
- 角色：主讲、联络、宣传
- 报名：simiter@126.com
- 报名主讲/嘉宾不需要你是大牛，只要熟悉SLAM某个知识点即可
- SLAM研习社详细介绍、直播预告、视频回放链接
- https://github.com/electech6/LearnSLAM/blob/master/README.md