

Cartographer解读--基于分枝定界闭环检测的原理

2021-05-16 15:01 639阅读 · 8喜欢 · 3评论



无境智能

粉丝：7 文章：1

+ 关注

前言

在Cartographer中，相关性扫描匹配可是重点中的重点，而分支限界加速更是大大提高了它的实时性。本文就来详细解读一下其中的奥秘。

本文提纲：

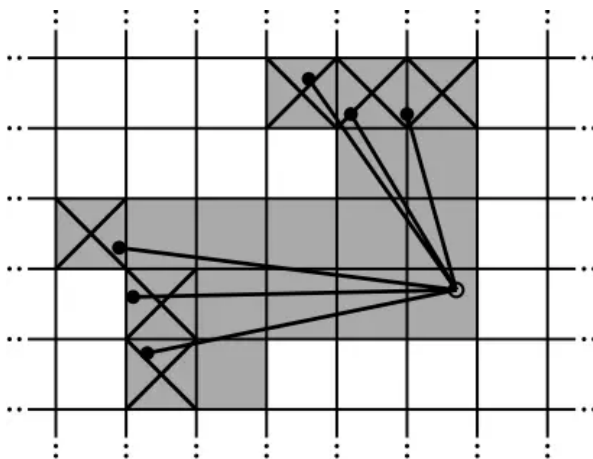
1，前置知识 2，分枝定界闭环检测的原理

1 前置知识

1.1 概率栅格地图

概率栅格地图是一种非常有用地图形式，请记住其中一点，概率栅格地图通常使用对数形式的概率值。

如何基于一帧激光scan来构造概率栅格地图？首先按照一定的分辨率把世界栅格化，然后每个激光点都会落在栅格中，该栅格称为hit点，如图所示。被hit栅格的占据概率是概率分布曲线中均值对应的概率，而对于没有被击中的栅格，将按照与最近被击中栅格的距离在正态分布曲线中做概率值查询，查出的概率就是这个栅格的占据概率。对所有的栅格都按这种方式处理，我们就得到了这个scan的概率栅格地图。（这个类似概率机器人学中的--似然域模型，但似然域模型考虑了更多的因素）。



1.2 扫描匹配

扫描匹配的实现方式有两种，第一种是相关性扫描匹配，第二种是基于优化的扫描匹配。第一种思路简单，但有一些复杂的细节，而第二种则是构建非线性最小二乘优化问题并求解。实际中，通常是两种方法同时用，先用相关性扫描匹配求初值，再用基于优化的扫描匹配进一步优化。本文只讨论第一种扫描匹配方法，其思路是：想象给定一个栅格地图，再给定当前帧的点云，怎样才能知道激光雷达所在的位置呢？最简单的办法，所有激光点云投影到栅格地图上，看在这个位姿时，点云是否与地图重合，重合程度最高的位置就是激光雷达的真实位姿，换句话说就是找到一个位姿使激光点云的重投影误差最小。



8



1



5



3

1.3 闭环检测

Cartographer通过创建大量的submap来实现大场景建图，submap在短时间内的准确度是可靠的，但长时间会存在累积误差，为了消除累积误差，需要通过闭环检测来优化所有submap的位姿。

1.4 分枝定界

分枝界限法是由三栖学者查理德·卡普（Richard M.Karp）在20世纪60年代发明，成功求解含有65个城市的旅行商问题，创当时的记录。“分枝界限法”把问题的可行解展开如树的分枝，再经由各个分枝中寻找最佳解。

其主要思想是：把全部可行的解空间不断分割为越来越小的子集（称为分支），并为每个子集计算一个下界或上界（称为定界）。在每次分支后，对凡是界限超出已知可行解值那些子集不再做进一步分支。这样，解的许多子集（即搜索树上的许多结点）就可以不予考虑了，从而缩小了搜索范围。

2 分枝定界闭环检测的原理

2.1 Cartographer闭环检测

Cartographer闭环检测使用了相关性扫描匹配方法，即当获得新的scan时，在其附近一定范围搜索最优匹配帧，若该最优匹配帧符合要求，则认为是一个闭环。

记为传感器扫描到的K个hit点集合，是第k个hit点在机器人坐标系下的位置坐标。那么在地图坐标系下的坐标可以表示为：

其中，表示位姿估计，分别是机器人在地图坐标系下的坐标，是机是机器人的方向角。用表示地图坐标系到机器人坐标系的坐标变换。那么这个Pixel-accurate扫描匹配问题可以用下式来描述：

上式中，是一个搜索窗口，是离最近的栅格单元的占用概率。上式可以解释为，在搜索窗口中找到一个最优的位姿，使得所有hit点出现的概率最大化。

论文中给出了一个暴力的搜索方法，该方法也是一种枚举的方法。给定搜索步长和，该搜索过程以为中心，通过三层循环遍历所有的Pixel并选出得分最高的位姿作为输出。

对于暴力匹配法来说，和方向搜索数量计算为

假设步长，因此

则搜索空间表示为

暴力匹配的 algorithm 如下所示：

2.2 深度优先的分枝定界加速搜索

显然如果搜索窗口过大或者搜索步长太小，都将导致整个搜索过程耗时过长。为了提高算法的效率，Cartographer选择使用分支定界的方法来搜索位姿。该算法的基本思想就是：把整个解空间用一棵树来表示，其根节点代表整个搜索窗口W。树中的每一个节点的孩子节点都是对该节点所代表的搜索空间的一个划分，每个叶子节点都对应着一个解。

整个搜索过程的基本思想，就是不断地分割搜索空间，这个过程称为**分支**。为每次分支之后的孩子节点确定一个上界，这个过程称为**定界**。如果一个节点的定界超出了已知最优解的值，这意味着该节点下的所有解都不可能比已知解更优，将不再分支该节点。这样很多分支就不会继续搜索了，可以缩小搜索范围，提高算法效率。

Cartographer使用一种深度优先(DFS, Depth-First Search,)的搜索方法，其算法的示意图如下图所示。该算法设定了一个阈值score_threshold，预先约定了下界，如果一个节点的上界比它还小，说明这个解空间实在太糟糕了，直接就不用考虑它了。

基于深度优先分枝定界搜索过程如下：

a、计算顶层线性搜索空间



8



1



5



3

假设，则，其中depth是自己定义的，如果是7，则顶层步长，那么顶层的线性搜索空间大小就是1000/128，大约为8，如果角度搜索个数300，则候选点节点层数目是： $4 \times 4 \times 300$ 。

论文中用下面两个公式计算了角度步长：

b、计算其余层结构

我们知道在二维空间中，机器人的位姿实际上是由三个量来表示的。那么搜索窗口的栅格索引集合可以通过笛卡尔积来枚举。其中分别是x和y方向上最大的搜索索引，则是搜索角度的最大索引。那么搜索窗口就可以用集合来表示，其中，是搜索中心，也是机器人位姿的初始估计。和分别是位移和角度的搜索步长。

搜索树中的每个节点都可以用四个整数来表示，其中分别是其搜索空间所对应的x,y轴的起始索引，是搜索角度，该搜索空间下一共有个可能的解，它们具有相同的搜索角度，但位置坐标不同。这些解的组合方式可以用如下的笛卡尔积来表示：

该节点所对应的搜索空间的栅格索引集合为

c、搜索步骤

1. 遍历Candidates所有Candidates，计算每个Candidate得分，从大到小排序；
2. 最低分辨率层 次低分辨率层计算这四新的Candidate得分，从大到小排序；
3. 重复步骤2，直到depth=Max，得到最底层的四个Candidate并计算其得分，得分最高的作为best_score
4. 返回倒数第二层，将best_score与剩下三个Candidate的得分比较，若best_score大于任何一个得分，则无需进入其他分支，继续返回best_score。
5. 重复4直到遍历整个分支树，返回的结果为最优结果。

深度优先的分枝定界 algorithm 下图所示：

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.

2.3 多分辨率概率栅格地图

下图是四张多分辨率概率栅格地图（也就是论文中的Precomputed grids），其尺寸分别为1、4、16、64，分支定价方法具体实现的时候，先从根节点开始遍历，把它拆分成4个子格子。这4个子格子分别计算评分，并由高到低排序。从中选出分数最高的格子，进



8



1



5



3

一步拆分成4个子格子。假设这4个子格子已经到了叶子节点，也就是达到了真实地图的分辨率。此时计算出的4个子格子的评分代表了真实的位姿评分（只有叶子节点的评分是真实评分，其它格子的评分都是上界），找出其最大值，记作best_score。以上便是分支的实现过程。接下来，该轮到限界出场了。第二层的格子还有3个未曾探索，我们当然是选择最大的那个开始分支。但别急，先看一下它的评分是否大于best_score，如果是，继续分支，如果否，就可以直接剪枝了，抛弃这个格子及其所有子格子。因为格子的评分代表了其子格子评分的上界，如果上界都小于best_score，就不可能再有子格子的评分大于它了。按照如此方法，大刀阔斧地剪枝即可。注意，当遇到评分大于best_score的叶子节点时，记得更新best_score，这样可以更快地缩小搜索空间。

搜索树上的每个节点的上界可以通过下式计算得到：

对于第n层的节点，在预算图中x,y的占用上界可以表示成下式，即以考察点x,y为起点，尺寸为的窗口内栅格的最高占用概率作为上界。

那么节点c的上界就可以直接查表得到：

所以，我们可以认为整个闭环检测的业务逻辑就是，根据当前的子图构建一个占用栅格地图，然后为该地图计算多分辨率地图，接着通过深度优先的分枝定界搜索算法估计机器人的位姿，最后就可以建立机器人位姿与子图之间的约束关系。

结语

Cartographer使用深度优先的分枝定界搜索算法进行闭环检测计算，该算法能够高效地确定激光点云与子图的匹配度，并估计出机器人位姿。为了能够高效的对搜索空间进行分割并计算上界，Cartographer还为每个子图预先计算了不同尺度下的占用概率，以后的搜索过程只需要简单的查表就可以完成。

如果本文对您有帮助，请不要吝惜点赞哦！

本文为我原创 本文禁止转载或摘编

分享到：

[投诉或建议](#)



8



1



5



3

推荐文章

更多精彩内容 >

关于我辞掉8K+工作去咖啡店打工的事.....

家里今年遇到很多事情，因为实在忙碌也糟心，在自己没理顺之前不想提起。到今天感觉差不多缓过神了，也算是人生的一个大...

浅间 日常 2363 59 12

知名男星刘维父亲去世！半年前妈妈癌症病逝，36岁父母双亡成孤儿

10月21日，知名男星刘维在社交平台上发文称自己的父亲已经于凌晨去世了，不过短短半年时间而已，刘维已经失去了自己的一...

会火-娱乐先知 日常 1754 8 1

高呼环保的正义人士，成了欧美的荒诞暴徒

原创 抓绒 X博士 欧洲的各大美术馆和博物馆的馆长，最近都比较紧张。因为谁也不知道，自家的艺术品会不会突然遭到一群疯...

X博士官方频道 日常 1056 113 5

评论

全部评论

按时间排序

8

1

5

3