ICP 算法简述

Version	Authors	Date	Details
V1.0	宫婉婷	2015 年 12 月 14 日星期一	文档创建

目 录

1 ICP 算法简介

机器人探索未知环境对于自身定位没有绝对的参考坐标，除了通过传感器可以发现的特征，使用可分辨的标记是一种可行方法，但是需要解决物体识别问题，尤其是物体如果使用二维激光范围扫描用于定位，很难精确探测以及定位范围扫描中的环境标记例如角落或者闭塞遮挡。ICP 算法是一种三维点集配准算法它是基于纯粹几何模型的三维物体对准算法由于它的强大功能以及高的精确度，很快就成为了曲面配准中的主流算法。算法原理：主要用于三维物体的配准问题。给定来自来自不同坐标系的数据点集，找到两个点集之间的空间变换使它们能进行空间匹配

2 基于原始代码对算法的理解

首先要清楚所具有的数据，这些数据是怎样获取的。是传感器在每个时间点对自己前方（一个平角的范围）所测出来的距离值。从后面的学习与思考，得知这些点很多，采集频率非常高。
针对这些数据的个数，在代码中进行提取，进行第一层 for 循环（循环的是每一对的点的 ICP 过程）这里是对，所以所迭代的次数是总数据减去一。

2.1 ICP 算法核心

输入参数有点多，但个人觉得某些固定值可以在 ICP 函数中直接定义，比函数更简洁点。
这里分析下输入参数：参考点的数据（所测所有距离）与预测点的数据（所测所有距离，初始化的 invDelta，门限值 gate 以及算法为了更精确所设置的迭代次数 nit）。算法核心针对基于激光束匹配的移动机器人相对运动，如图 1

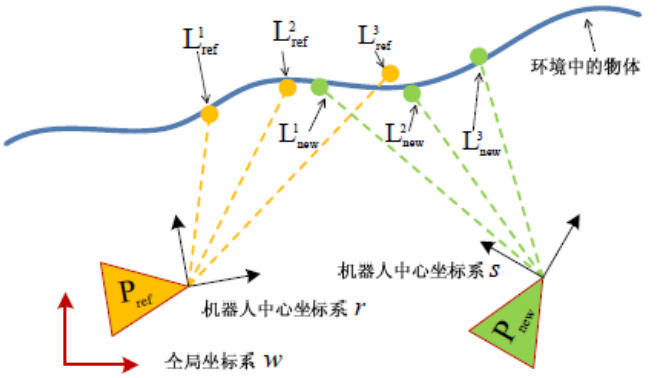

图 1: 基于激光束匹配的移动机器人相对运动示意图

2.1.1 距离转换为坐标（get_laser_points）:

这里要通过传感器是如何测出这些距离来计算坐标，这里的坐标是基于该点的坐标。但是该过程有个限制值，就是测算出的数据值过大的点是舍去的。所要理解的是函数结果输出的坐标都是基于每个点的坐标。参考点与预测点都需要坐标转化。

2.1.2 invDelta 参数简介

自己曾经在 invDelta 这个参数到底是什么，到底是怎么出来的，以及为什么参考点需要和它进行 compond。后来思考才发现 invDelta 是参考点 A 在预测点 B 的坐标，预测点相对于参考点的运动表现在位置和角度变化上，数学表达式可以看做 Tba，但是这个值在刚开始是我们先假设出来的，所以很明显需要对这个值进行优化和改进。后来在修改代码时将其重新命名为 rel_pos_ba。更好的理解为该值是参考点在预测点下的坐标。

2.1.3 预测点值的优化

通过参考点与预测点 laser_points 进行优化，优化原理：基于相邻激光束中激光点的关联关系并通过迭代优化欧式距离误差函数。在高的时间频率下，很多点会被相邻位置的激光束所采集。

2.1.4 不同坐标系下欧氏距离计算

已知 Xba，Yba 等，这个时候可以把参考点 A 的激光点的坐标与其进行 compound，这样就可以得出参考点 A 的所有激光点在预测点 B 下的坐标，这样的话，参考点 A 与预测点 B 所有的激光点便表示在预测点 B 的坐标系下。这样的话，可以通过函数将所有的点的距离算出来形成一个方阵，因为每一个点都会和对应激光束下的所有点进行距离计算，这样就形成了一个方阵。

在这里有个细节处理：就是在之前的舍去过长点的时候可能会让参考点与预测点的激光点不同，算法采用了取少的边界来计算，保证每个激光点都得到了相应的匹配。之后对这个方阵进行处理，这个方阵的每一行都是参考点 A 的激光点的某一个与预测点 B 所有的激光点的距离值。再通过 min 函数，将每一行的最小值以及最小值所在的列数，输出出来。得到了一个 M*2 的矩阵。第一列是每一个参考点 A 的激光点与其对应预测点 B 的激光点的最小距离值，第 2 列是该激光点对应的预测点 B 的激光点的第几个。

2.1.5 筛选匹配点

在这个计算过程中可能会产生一个参考点的激光点对应了 2 个或 2 个以上的预测点的激光点，这样的情况，通过双层 for 循环，舍去那些多个匹配的激光点。最终得到了一个新的 M*2 的矩阵，

第一列是参考点 A 激光点的序列号，第二列是对应预测点的激光点的序列号这个矩阵是参考点 A 的激光点分别对应的预测点的激光点（完成了激光点的匹配）。

这样的话可以通过这个矩阵来找到我们真正需要的激光点。在经过相应的数学计算得出来一个列向量，自己分析：通过分配好的对应的激光点去优化先前相对位置（平移与旋转）。通过计算之后的相对位置（这里还是参考点对于预测点的坐标，自己之前一直认为是预测点相对参考点的坐标，后来通过单步执行，才发现的，原因就在于那段没有理解好的数学公式计算）。通过误差分析来判断是否满足条件，不满足在这个基础上再去迭代一次。满足的话则 break 跳出 for 循环。

这是一对点的 ICP 过程，整个过程得出来的其实是$\begin{bmatrix} X_{ba} \\ Y_{ba} \\ \theta_{ba} \end{bmatrix}$。然后进行 inverse。这样的话与 pos（参考点的世界坐标）进行 compound。可以得到预测点的世界坐标。这个坐标将会成为下次迭代（下一对激光束的匹配）的参考点坐标。就这样将每一对进行 ICP。得到每个点的世界坐标。

算法在每次迭代的过程中，对数据点的每一点，在模型点中寻找欧式距离最近点作为对应点，

$$R = \min \sum_{1}^{N} \| Q_i - (RP_i + t) \|^2$$

通过这组对应点使目标函数最小化：来得到最优的平移向量 t 和旋转矩阵 R。将平移向量 t 和旋转矩阵 R 作用到数据点上，得到新的数据点云带入下次迭代过程。

3 算法流程:

1. 获取数据，计算数据数量决定外层循环次数。
2. 从前往后获取相邻激光束的距离数据，将数据距离转为自身坐标系的坐标。
3. 将参考点与预测点的所有激光点的距离计算得出距离矩阵。
4. 通过该矩阵得到最终的对应匹配点。
5. 利用对应匹配点去优化相对位姿。迭代出最好的相对位姿
6. 通过计算的相对位姿，compound 成世界坐标，就是预测点的世界坐标。
7. 下一次 for 取下一对激光束，上一次的预测点是这一次的参考点，过程一样
8. 直到遍历完所有的激光点

4 算法修改思想

再看懂了算法的流程之后，发现自己在 ICP 流程中的计算参考点的激光点在预测点的坐标停留了很久，本来自己觉得应该是基于参考点来计算激光点的距离。源代码中先计算参考点 A 在预测点 B 下的坐标，在进行 inverse。

后来自己理解了之后想到可不可以就按照自己先前感觉应该在参考点下计算距离呢，这样会直接可以计算出预测点基于参考点的坐标，在进行 compound，得出预测点的世界坐标。这样的话就可以不用 inverse 去转化了。代码就可修改。

5 结束语

花了很长时间才改出来的代码，在文档上自己还是不是很规范，但是了解基本原理思想和从代码了解算法的能力大概形成，很感谢老师的耐心和体谅，经过这几天的我会更加专心的看代码，将自己的方向按照时间节点完成。

参考文献

[1] F. Lu and E. Milios, “Robot pose estimation in unknown environments by matching2d range scans,” Journal of Intelligent and Robotic Systems, vol. 18, no. 3,pp. 249–275, 1997.

[2] Zuolei Sun Localization and Mapping for Mobile Robotin Large-Scale Unstructured Environment 2010

[3] R. Smith, M. Self, and P. Cheeseman, “Estimating uncertain spatial relationshipsin robotics,” in Autonomous Robot Vehicles (I. Cox and G. Wilfong, eds.),pp. 167–193, Springer-Verlag, 1990.

[4] Zuolei Sun Quantify the Uncertainty inCRF-Laser Scan Matching

[5] 3D shape modeling using a self-developed hand-held 3D laser scanner