

从零开始一起学习SLAM | ICP原理及应用

小白：师兄，最近忙什么呢，都见不到你人影，我们的课也好久没更新了呢

师兄：抱歉，抱歉，最近忙于俗事。我后面一起补上，学习劲头得向你们年轻人学习啊！话说，你最近在研究什么呢？

小白：最近在看ICP，十四讲上简单提了下，还有点懵懵哒

师兄：好，那今天就说说ICP算法吧。搬个小板凳做好哈：

ICP全称Iterative Closest Point，翻译过来就是迭代最近点。ICP在点云配准领域应用的非常广泛，因此基于深度相机、激光雷达的算法使用ICP的频率比较高。

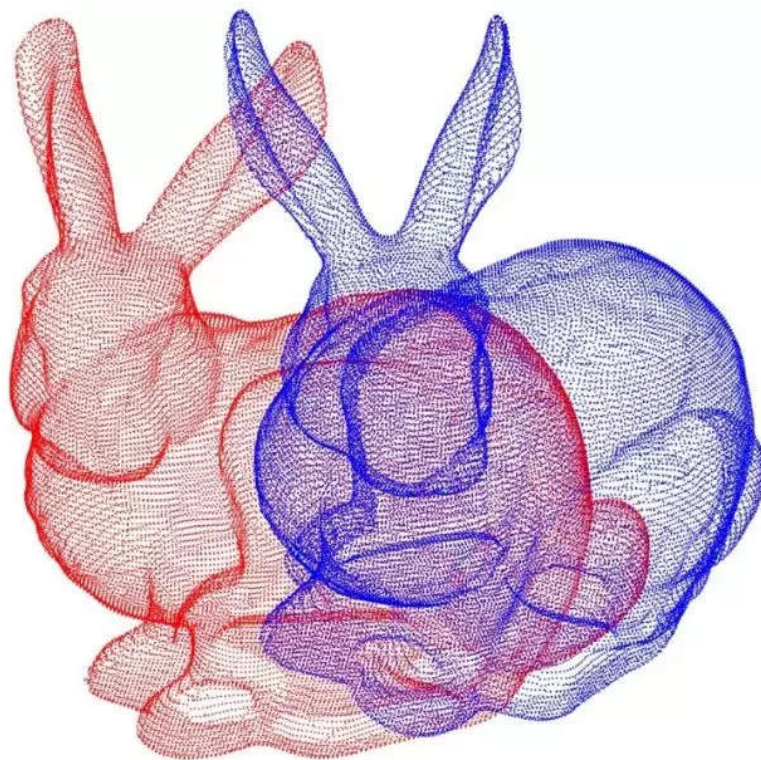
小白：那我是学视觉SLAM的，是不是不用关注了啊（内心OS：毕竟SLAM要学的太多了）

师兄：哈哈，这个懒不能偷的。ICP现在在视觉SLAM中用的也非常多了。我总结了一下ICP的应用场景：

ICP到底有什么用呢？

假设我们在三维空间中有两组点集，我们分别称之为P1, P2好了，P1, P2中都有几千个点，那么可以分为两种情况：

1、我们完全不知道P1, P2中每个点是如何对应的。这样的情况很常见，比如我用只有depth的深度相机或者激光雷达在两个不同的位置对着一只兔儿分别拍摄一张深度图并转化为点云图，我特喵的也不知道哪个点和哪个点对应，但是我想把这两个点云“融合”（专业词汇叫做registration，有些地方翻译为注册，有点让人不知所云，我建议翻译为配准）在一起，变成一个更完整的兔儿。

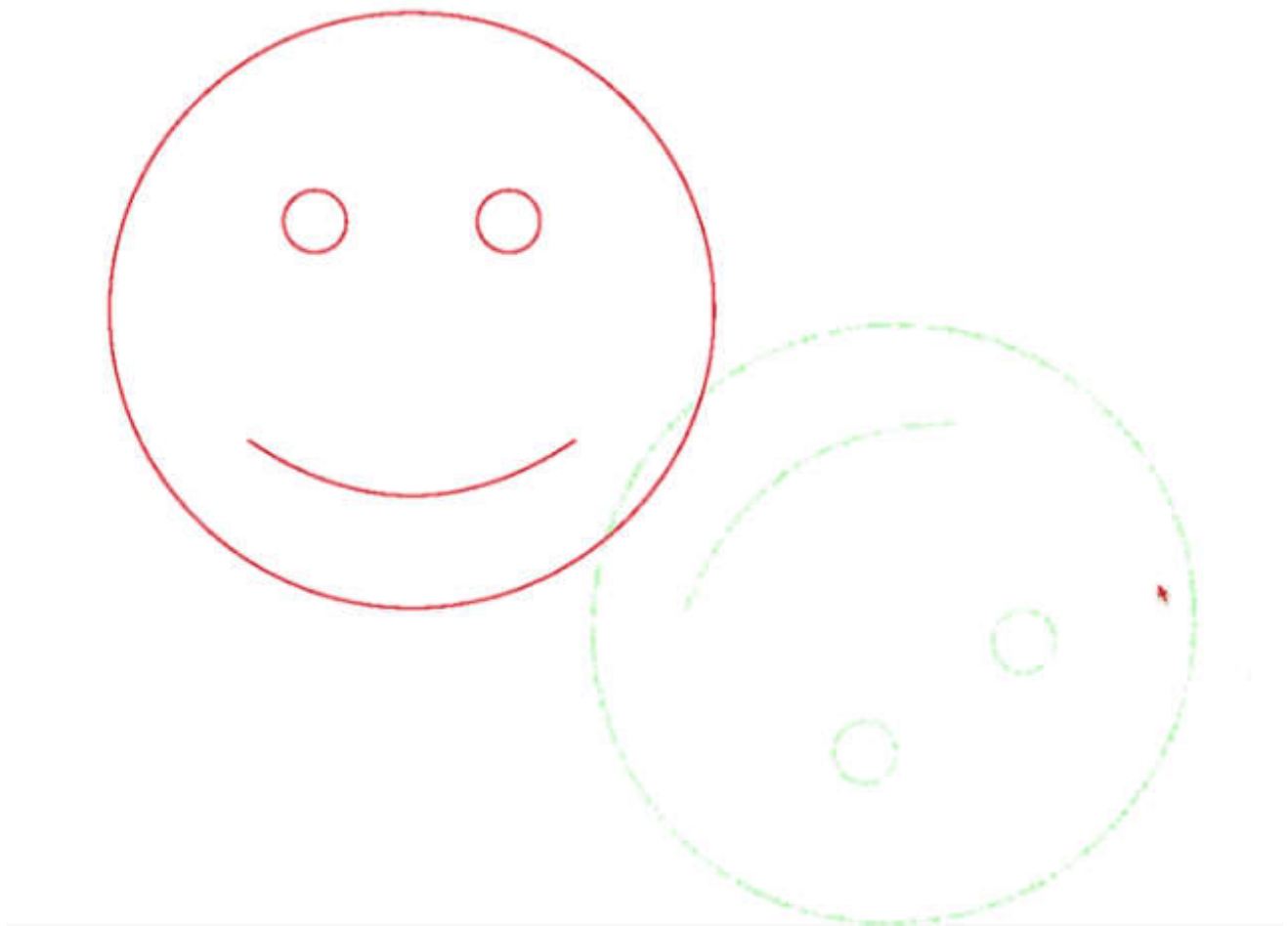


2、我们已经知道 P_1 , P_2 中哪个点对应的哪个点。比如我们使用RGB-D相机分别在两个不同位置拍摄一张 RGB彩色图 + 深度图，而且彩色图和深度图是对齐好的。因为有彩色图我们就可以做特征点匹配了，因为每个特征点都对应一个深度图上的深度值，所以我们能够得到两组对应好的三维点。

小白：原来如此，看起来ICP还挺强大的，对应不对应都能搞定啊，ICP为何这么牛叉？

ICP为何这么牛叉？

师兄：来看看ICP的过程就知道了。为了方便，我们用一个二维的例子来说明吧，初始是两个不同角度下的笑脸（深红色和绿色），下面是红色笑脸如何通过ICP过程和绿色笑脸重合的：



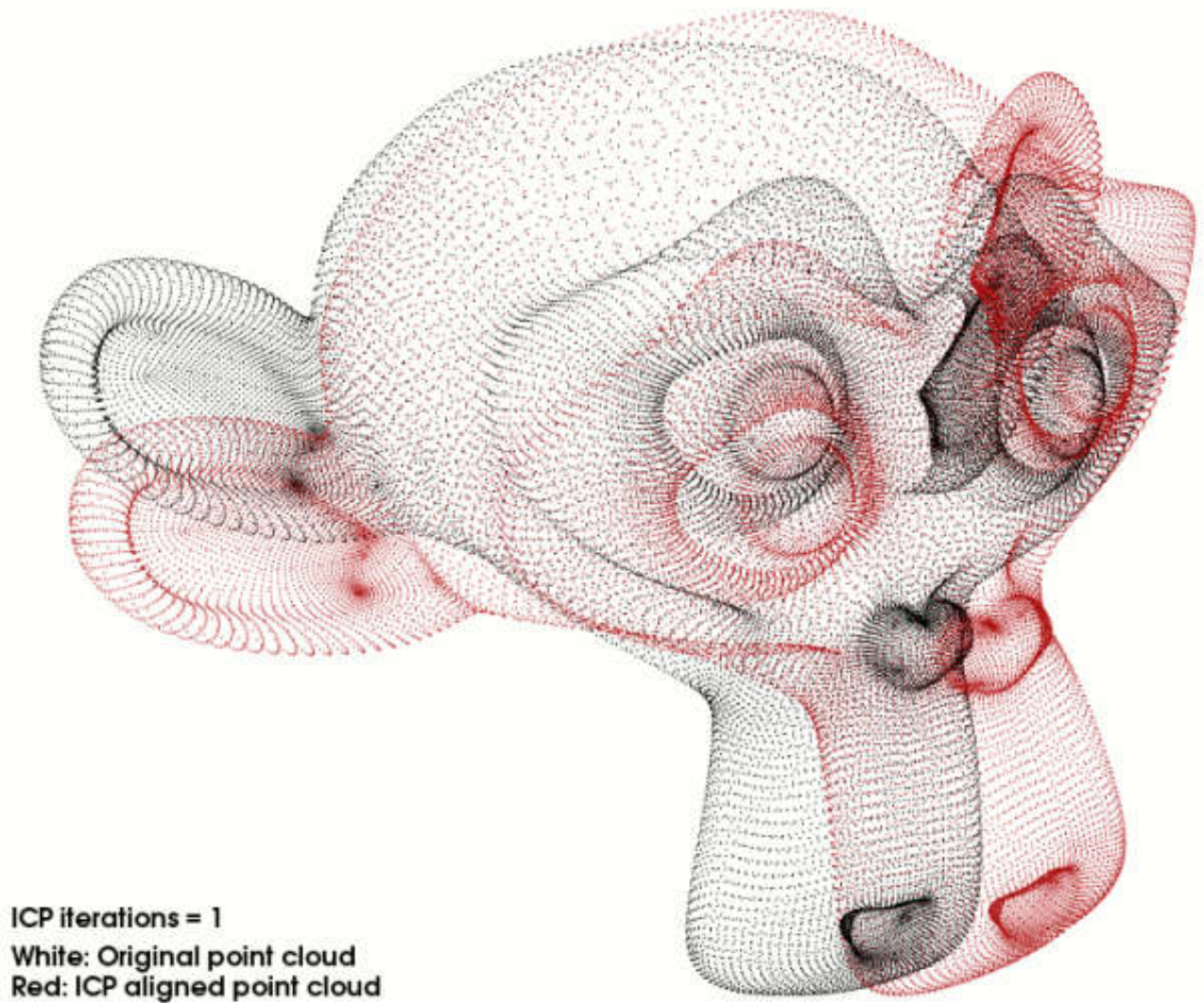
小白：好神奇啊

师兄：这个只是一个示例，通常情况下我们在SLAM中不会遇到这种情况，都是三维空间的点。我再以点云配准为例进行说明。假设现在有两幅待配准的点云（比如上面的小兔子），ICP算法是这样配准两幅点云的：

ICP算法流程

1. 首先对于一幅点云中的每个点，在另一幅点云中计算匹配点（最近点）
2. 极小化匹配点间的匹配误差，计算位姿
3. 然后将计算的位姿作用于点云
4. 再重新计算匹配点
5. 如此迭代，直到迭代次数达到阈值，或者极小化的能量函数变化量小于设定阈值

下面是用三维点云进行ICP的一个效果



小白：第一次计算的误差估计很大，后面会慢慢变小，直到收敛？

师兄：嗯，这是一个由粗放到精细的过程，在视觉算法中由粗到精的方法很常用。。

小白：比如图像金字塔！哈哈

师兄：嗯，熟悉的配方，熟悉的味道（/笑脸）

ICP的精髓其实就是迭代，一次次的修正错误，最后获得一个还不错的结果。

不过话是这样说，但是一般情况下数据都有不少噪音，如果噪音较大，第一种情况不一定能完全配准正确。

小白：听起来合情合理，也不麻烦，书上为啥推导那么复杂？

师兄：嗯，这就是感性到理性的过程。视觉SLAM十四讲书上从数学上推导了具体如何实现。我这里引用一下求解ICP的三个步骤：

1. 计算两组点的质心位置 p, p' ，然后计算每个点的去质心坐标：

$$q_i = p_i - p, \quad q'_i = p'_i - p'.$$

2. 根据以下优化问题计算旋转矩阵：

$$R^* = \arg \min_R \frac{1}{2} \sum_{i=1}^n \|q_i - Rq'_i\|^2.$$

3. 根据第二步的 R ，计算 t ：

$$t^* = p - Rp'.$$

小白：原理推导大概是能看明白了

师兄：嗯嗯，推导部分给你留了一个小小的课后作业了

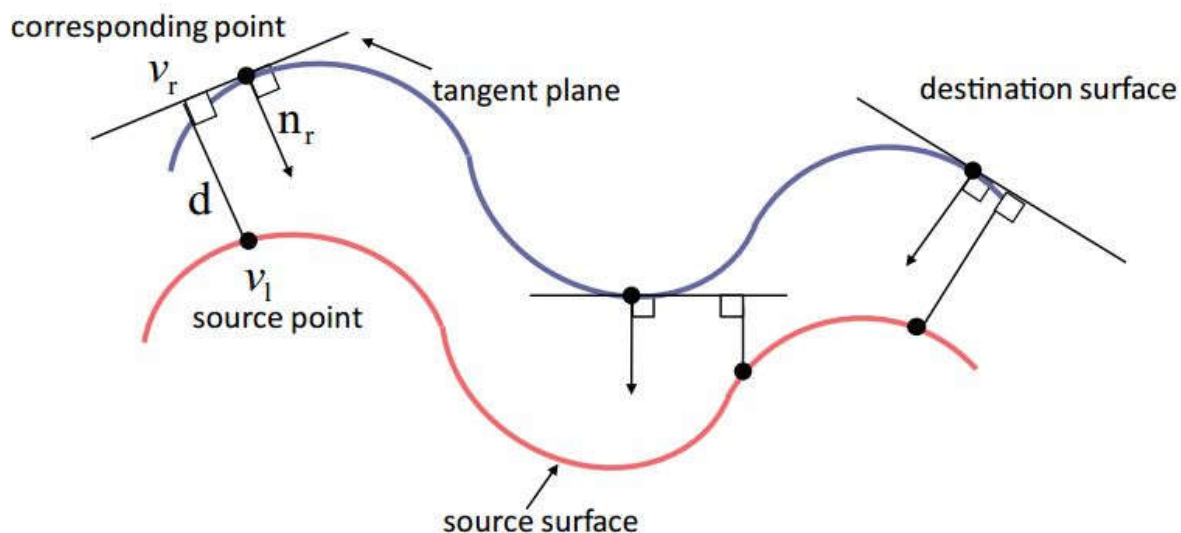
小白：师兄想的周到，这样才能真正掌握呢（内心OS：真狠啊。。）

师兄：上面介绍是最简单的点和点匹配的ICP算法，实际应用中为了使得算法鲁棒，也就是在有不少噪声的情况下仍然能够得到正确的结果，很多研究者提出了不同的匹配思路：例如，极小化的误差项包括对应点的点到点的欧式距离，和对应点的点到平面距离,以及极小化对应点的颜色值误差等

小白：这么多方法，用哪个好呢？

师兄：如果你应用场景简单，数据干净基本没有噪声，最原始的点点匹配就行。如果有噪声，则最后考虑一下复杂点的方法。2003年的时候，pottman 和Hofer两位大牛的论文中证明了当两幅点云比较接近时，极小化对应点的点到平面距离比点到点距离更接近两个平面之间的真实距离，也就是说计算点到平面的距离更靠谱！

下面是个示意图



ICP 算法通过极小化最近点的点到平面的距离计算位姿

这就是ICP的原理啦！

下面是实践环节

练习

1、推导ICP中的一个步骤

证明：《视觉SLAM十四讲》第174页公式7.55中的

$$q_i^T R q'_i = \text{tr}(R q'_i q_i^T)$$

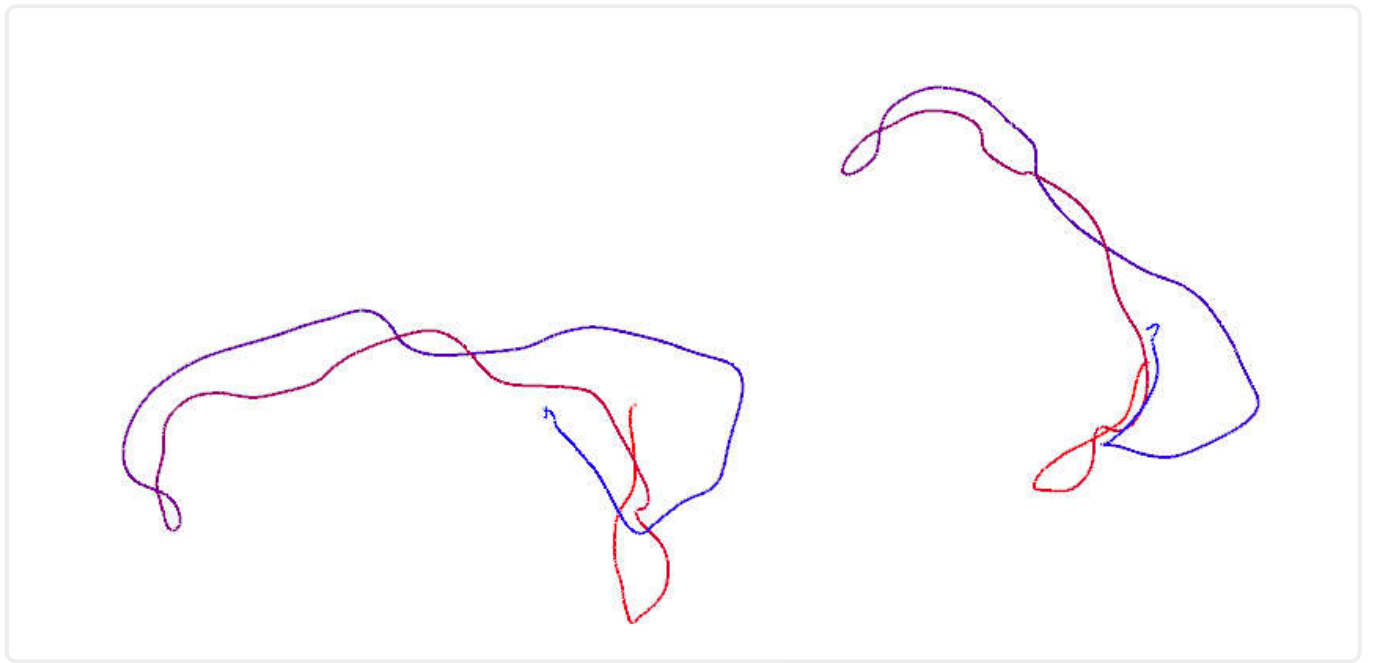
各符号定义见书上，其中，tr表示矩阵的迹。

2、精心设计的ICP编程实践

给定一个轨迹1，数据格式：timestamp tx ty tz qx qy qz qw，自定义一个任意的旋转矩阵和平移向量（可以尝试不同的值，甚至加一些噪声看看结果有什么变化），对轨迹1进行变换，得到一个新的轨迹2，使用ICP算法（提示：取平移作为三维空间点）估计轨迹1，2之间的位姿，然后将该位姿作用在轨迹2。

验证：ICP算法估计的旋转矩阵和平移向量是否准确；轨迹1，2是否重合。

如下是我加了一个旋转平移量后的两个轨迹，经过ICP计算好位姿后再反作用在变换后的轨迹，最终两个轨迹是重合滴！



本文参考：高翔《视觉SLAM十四讲》

推荐阅读