



点到平面距离公式

$$d = \frac{|Ax_1 + By_1 + Cz_1 + D|}{\sqrt{A^2 + B^2 + C^2}}$$

## FAST\_LIO构建残差函数并进行二次筛选 面特征

FrankDellaert  
酷的像风

已关注

来自专栏 · FAST\_LIO2 &gt;

7 人赞同了该文章 &gt;

### 一、SLAM中的经验参数

下面公式子是什么意思？就是本文第一节要说明的内容！Github有这个issue<sup>[1]</sup>

```
float coefficient= 0.9 * fabs(pdistance) / sqrt(p_body.norm());
```

#### 1.1 对标定参数的看法

这是SLAM比较迷人的地方，但是也是它最没有普世性的地方，当然你说这个地方可以发一个专利，然而对我完全没有吸引力，写论文又不算一个创新点；同时这是SLAM最弱的地方，它到处都是这种经验值，对于这些经验值，我不想详细推导他们如何来的，因为这个东西也推导不了，完全是数据分析后的结果，作者一定是先将数据总结出来，然后设定的经验损失函数的系数，本文准备做个思路梳理；

#### 1.2 当前帧激光点的最近五个地图点如果拟合不到一个平面呢

通过下面的参数进行判断拟合后的点到平面的距离，如果

```
if (esti_plane(pabcd, points_near, 0.1f)) {}
```

#### 1.3 原公式的简化预处理

首先为了简化逻辑,将下面代码中的负号去掉,

```
float s = 1 - 0.9 * fabs(pdistance) / sqrt(p_body.norm());
```

变换成：

```
float coefficient= 0.9 * fabs(pdistance) / sqrt(p_body.norm());
```

if (s > 0.9) 变换成 if(coefficient<0.1111111)

```
Matrix<float, 4, 1> pabcd;          //平面点信息，有效的参数实际是3个，第4个是法向量
point_selected_surf[i] = false;     //将该点设置为无效点，用来判断是否满足条件
//拟合平面方程ax+by+cz+d=0并求解点到平面距离
if (esti_plane(pabcd, points_near, 0.1f))
{
    //当前点到平面的距离
    float pdistance = pabcd(0) * point_world.x + pabcd(1) * point_world.y + pabcd(2)
    float coefficient= 0.9 * fabs(pdistance) / sqrt(p_body.norm());
    //如果残差大于经验阈值，则认为该点是有效点 简言之，距离原点越近的lidar点 要求点到平面的距离越小
    if(coefficient<0.1111111) // 1/9=0.11111
    {
```

赞同 7 添加评论 分享 喜欢 收藏 申请转载



#### 关于作者

FrankDellaert  
酷的像风膝盖借箭诸葛先生、袁博融、王小迪  
MLE 也关注了他

回答 152 文章 259 关注者 912

已关注

发私信

参与 JetBrains 2025开发者生态系统调查，赢取大礼！

#### 预见 2025 开发者趋势

分享你的观点  
有机会获得  
MacBook Pro

广告



```
normvec->points[i].intensity = pdistance;
}
}
```

## 1.4 定性分析和理解

参考<sup>[2]</sup> 其实最大启发我的还是文章<sup>[3]</sup>，但是它的文章不应该是LOAM<sup>+</sup>，而是应该是FAST\_LIO，

s 是个权重，表示s在这个least-square问题中的置信度，每个点的置信度不一样，这个权重，与点到面距离负相关，距离越大，置信度越低，这里相当于是一个在loss之外加了一个鲁棒性函数，用来过减弱离群值的影响

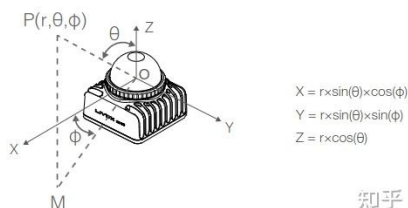
源代码中" $\sqrt{\sqrt{\text{point\_sel.x} * \text{point\_sel.x} + \text{point\_sel.y} * \text{point\_sel.y} + \text{point\_sel.z} * \text{point\_sel.z}}}$ " 这部分，并没有什么逻辑性可言，可以设计一个鲁棒性函数来替代这一行代码；

参考文章<sup>[4]</sup> 可得出最重要的结论：激光点最后得出的残差是激光点到地图中平面的距离，这个距离对应的激光点，可能实际上是不存在的；

```
float coefficient= 0.9 * fabs(pdistance) / sqrt(p_body.norm());
```

- 分子：pdistance表达当前降采样的激光点到它附近5个最近的地图点拟合成平面的距离，但是最远那个点距离当前p\_body距离的平方不能超过数值5；当前激光点到拟合地图平面的距离越小，越是我们想要的，因为这个数值代表了匹配关系，如果匹配的距离越大，表示匹配关系就越不合理；
- 分母：p\_body.norm()已经表达了激光点到激光雷达坐标原点的距离，如下图所示<sup>[5]</sup>，为什么越远就越可信呢？只是理解地图中的面点一般都距离当前激光雷达点的距离比较远；

坐标信息：Mid-360的坐标信息可表示为直角坐标系 (x, y, z) 或球坐标系 (r,  $\theta$ ,  $\phi$ )，其直角坐标和球坐标的对应关系如下图所示。如果前方无被探测物体或者被探测物体超出量程范围（例如 >100 m），在直角坐标系下，点云输出为 (0, 0, 0)；在球坐标系下，点云输出为 (0,  $\theta$ ,  $\phi$ )。



## 二、构建残差

### 2.1 如何定义残差

残差是当前激光点到最近的 5 个 地图中平面类型的激光点拟合成平面的距离：

be where the point truly belongs to. That is, the residual is defined as the distance between the feature point's estimated global frame coordinate  ${}^G\hat{\mathbf{p}}_{f_i}^{\kappa}$  and the nearest plane (or edge) in the map. Denoting  $\mathbf{u}_j$  the normal vector (or edge orientation) of the corresponding plane (or edge), on which lying a point  ${}^G\mathbf{q}_j$ , then the residual  $\mathbf{z}_j^{\kappa}$  is computed as.

### 2.2 用约束匹配对 构建残差函数

当前帧点云与局部地图进行匹配 构建残差函数；其实SLAM的问题都可以归结为一个超定方程\*的问题，一般水平的同学会只是估计如何让匹配对形成的方程比较号，包括如何设定匹配中筛选的各种阈值啊等等，但是更加优秀的同学会在此基础上，从数学的角度去剖析如何将这个超定方程求解的较好，比如CMU的zhangji老师防退化的文章<sup>[6]</sup>

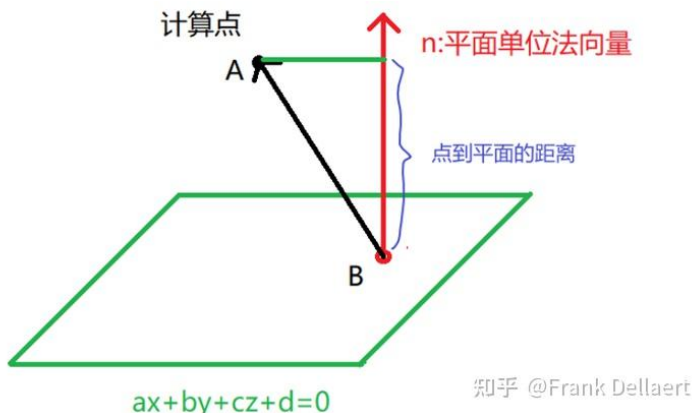




如果尝试谈谈SLAM研究的本质究竟是什么？我会说解决了一个超定非线性方程；

- 对于超定，没有解析解，只能找到近似解；
- 对于非线性，我们需要利用泰勒展开式<sup>+</sup>和微分进而将非线性函数线性化；

下面的点面组成的约束的约束对会有很多，给我们求解SLAM结果提供输入；

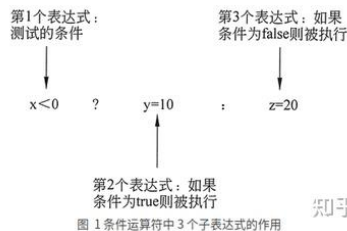


### 2.3 条件运算符(三目运算符)<sup>[7]</sup>

```
(x < 0) ? (y = 10) : (z = 20);
```

等价于：

```
if (x < 0)
    y = 10;
else
    z = 20;
```



### 2.4 筛选 拟合平面的备选5个点

```
bool point_selected_surf[100000] = {1}; //判断当前帧的点云激光点是否是有效的面特征点
```

然后这10万个点的标志位，设定为bool类型，请注意下面的所有判断是针对，**当前帧激光点中的平面点**进行判断是否满足条件，而不是对地图点进行筛选，**注意区分！**

搜索附近点云一定会用到函数ikdtree的成员函数<sup>[8]</sup> Nearest\_Search(),只有下面语句在调用最近邻搜索函数

```
// 寻找point_world的最近邻的平面点
# define NUM_MATCH_POINTS (5)
ikdtree.Nearest_Search(point_world, NUM_MATCH_POINTS, points_near, pointSearchSq
```

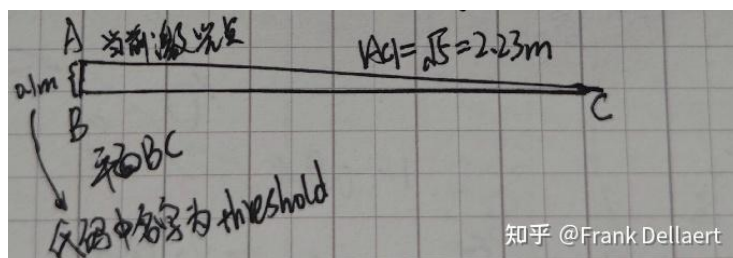
下面语句是对检索最近邻点进行校验,如果最近邻找到的点小于5个，那return false；多数情况，应该是等于5个，但如果小于5个时候是不合理的因为那么多的邻近点，在没有限定阈值的情况下，居然找不到5个点，那么一定是**地图点过于稀疏**了；

```
point_selected_surf[i] = points_near.size() < NUM_MATCH_POINTS ? false : \
pointSearchSqDis[NUM_MATCH_POINTS - 1] > 5 ? false: true;
```





能，原因是：如果这个点特别偏，那么形成的角度  $\angle ACB$  将会很小，对于拟合平面是不利的；



```
if (!point_selected_surf[i])
    continue; // 如果该点不满足条件 不进行下面步骤
```

如果上面的判断条件不满足，就不进行后续的面点提取了；

## 2.5 拟合平面点后进行再次筛选

二次筛选的激光点出现异常值，应在哪里删掉？可是适当参考文章<sup>[9]</sup>，同时下文也将给出回应：

下面进行面点提取，FAST\_LIO1论文中表达阈值参数是0.5m，代码中是 0.1m；原因是：很短时间内，激光雷达和刚性连接的物体不可能走那么远，就将这个筛选点跳过！

Moreover, we only consider residuals whose norm is below certain threshold (e.g., **0.5m**). Residuals exceeding this **threshold** are either outliers or newly observed points<sup>[10]</sup>.

对应代码：

```
// 如果几个点中有距离该平面>threshold的点 认为是不好的平面 返回false
for (int j = 0; j < NUM_MATCH_POINTS; j++)
{
    if (fabs(pca_result(0) * point[j].x + pca_result(1) * point[j].y + pca_result(2)
    {
        return false;
    }
}
return true;
```

FAST\_LIO1还有一个开关设定平面点和角点的设定选项，FAST\_LIO2就较激进，只考虑平面点，不考虑角点；

-----写给未来

1. 针对下面拟合内容不算很难，结合2.4节 和文章<sup>[11]</sup>进行详细剖析：

```
template<typename T>
bool esti_plane(Matrix<T, 4, 1> &pca_result, const PointVector &point, const T &t
```

## 参考

1. ^ 288 [https://github.com/hku-mars/FAST\\_LIO/issues/288](https://github.com/hku-mars/FAST_LIO/issues/288)
2. ^ 2.2 点到平面距离 <https://zhuanlan.zhihu.com/p/5026538391>
3. ^ 系数s的解析 <https://www.cnblogs.com/chenlinchong/p/16133350.html>
4. ^ 3.3 筛选 当前帧激光点中的平面点 <https://zhuanlan.zhihu.com/p/5232561677>
5. ^ [https://terra-1-g.djicdn.com/851d20f7b9f64838a34cd02351370894/Livox\\_Mid-360\\_User\\_Manual\\_CHS.pdf](https://terra-1-g.djicdn.com/851d20f7b9f64838a34cd02351370894/Livox_Mid-360_User_Manual_CHS.pdf)
6. ^ <https://zhuanlan.zhihu.com/p/712698359>
7. ^ 条件运算符 <https://c.biancheng.net/view/1363.html>
8. ^ 2.6 Nearest\_Search <https://zhuanlan.zhihu.com/p/1036324088>
9. ^ FAST\_LIO2中关于平面点和角点的设定选项 <https://zhuanlan.zhihu.com/p/9429660492>
10. ^ <https://arxiv.org/abs/1607.02484>
11. ^ <https://arxiv.org/abs/1607.02484>

赞同 7 添加评论 分享 喜欢 收藏 申请转载

