

Fig. 2. Measurement model. A LiDAR point is assumed to lie on a small plane formed by its nearby map points. The $G u_j$ is the normal vector of the plane and $G q_j$ is a point lying on the plane.

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

激光SLAM 拟合平面为什么需要5个点?

FrankDellaert
酷的像风

已关注

来自专栏 · FAST_LIO2 >

佳浩 等 70 人赞同了该文章 >

按照高中数学知识，至少需要3个不共线的点，就可以拟合成一个平面了，为什么需5个点？

没时间看细节的同学，直接看结论：

paper里使用了3个地图点，但代码里使用5个地图点进行拟合平面的法向量；计算残差时使用5个地图点拟合出的平面法向量；我知道使用3个点肯定不合适，因为有噪声的原因，为什么不是4个点或6个点呢？这里就是SLAM的核心问题，太多参数是基于规则进行调优的内容，类似于标定；

为什么我们再回顾LOAM⁺，因为虽然后续有很多新内容，但是基础是按照它这个框架来的，没变的就是复用；

所以除非经由记忆之路不能抵达纵深----- 阿伦特

深刻地感受到SLAM的绝大部分知识点：是涉及到几何，从高中几何到抽象几何和代数，外加编程；

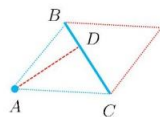
一、空间几何基础

Frank Dellaert：叉积(向量积、外积)+点积(标量积、内积、点乘)
8 赞同 · 0 评论 文章



1) 点到线：2) 点到面：

1) 点到直线距离

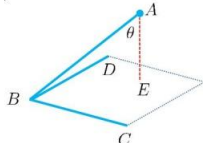


点 A 到直线 BC 的距离为

$$|\vec{AD}| = \frac{|\vec{AB} \times \vec{AC}|}{|\vec{BC}|}$$

即平行四边形面积除以对角线长度

2) 点到平面距离



平面 BCD 的单位法向量为 $\vec{n} = \frac{\vec{BC} \times \vec{BD}}{|\vec{BC} \times \vec{BD}|}$

点 A 到平面 BCD 的距离为

$$|\vec{AE}| = |\vec{AB}| \cos \theta = |\vec{AB}| \cdot |\vec{n}|$$

还有一个说明图也展示出来：图片引用自 @计算机视觉life

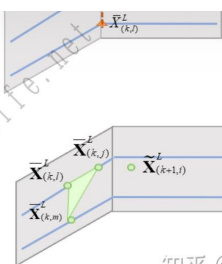


通过优化位姿,使残差接近零

$$d_e = \frac{(\tilde{\mathbf{X}}_{0,k+1,j}^L - \tilde{\mathbf{X}}_{0,k,j}^L) \times (\tilde{\mathbf{X}}_{0,k+1,j}^L - \tilde{\mathbf{X}}_{0,k,j}^L)}{|\tilde{\mathbf{X}}_{0,k,j}^L - \tilde{\mathbf{X}}_{0,k,j}^L|}$$

乘了待优化位姿的坐标点

$$d_H = \frac{\left(\begin{pmatrix} \tilde{\mathbf{X}}_{0,k+1,j}^L - \tilde{\mathbf{X}}_{0,k,j}^L \\ ((\tilde{\mathbf{X}}_{0,k,j}^L - \tilde{\mathbf{X}}_{0,k,j}^L) \times (\tilde{\mathbf{X}}_{0,k,j}^L - \tilde{\mathbf{X}}_{0,k,m}^L)) \end{pmatrix} \right)}{\left| (\tilde{\mathbf{X}}_{0,k,j}^L - \tilde{\mathbf{X}}_{0,k,j}^L) \times (\tilde{\mathbf{X}}_{0,k,j}^L - \tilde{\mathbf{X}}_{0,k,m}^L) \right|}$$



面积/底边

四面体体积
/底面面积

知乎 @Frank Dellaert

Zhang J, Singh S. LOAM: Lidar odometry and mapping in real-time

两个向量互相垂直的公式都是内积为零 $\mathbf{a}1*\mathbf{b}1+\mathbf{a}2*\mathbf{b}2+\mathbf{a}3*\mathbf{b}3=0$, 在更高维度的向量空间中也是类似的;

二、ALOAM 拟合平面

2.1 构建超定方程

平面的一般方程形式: $Ax + By + Cz + 1 = 0$ 平面法向量(A,B,C) [待求]的问题是 已知5个点求解3个未知数, 构建^[1]超定方程 $AX = b$ 求解平面法向量;

ALOAM 拟合平面点需要5个点, 计算残差使用3个点;

The planar patch is represented by 3 points, Similar to the last paragraph;

LOAM paper里计算残差只使用3个平面点:

i, j , and l in $\{L\}$, respectively. Then, for a point $i \in \tilde{\mathcal{H}}_{k+1}$, if (j, l, m) is the corresponding planar patch, $j, l, m \in \tilde{\mathcal{P}}_k$, the point to plane distance is

计算平面时只使用3个地图点

$$d_H = \frac{\left| \frac{(\tilde{\mathbf{X}}_{k+1,i}^L - \tilde{\mathbf{X}}_{k,j}^L) \cdot ((\tilde{\mathbf{X}}_{k,j}^L - \tilde{\mathbf{X}}_{k,l}^L) \times (\tilde{\mathbf{X}}_{k,j}^L - \tilde{\mathbf{X}}_{k,m}^L))}{(\tilde{\mathbf{X}}_{k,j}^L - \tilde{\mathbf{X}}_{k,l}^L) \times (\tilde{\mathbf{X}}_{k,j}^L - \tilde{\mathbf{X}}_{k,m}^L)} \right|}{\left| (\tilde{\mathbf{X}}_{k,j}^L - \tilde{\mathbf{X}}_{k,l}^L) \times (\tilde{\mathbf{X}}_{k,j}^L - \tilde{\mathbf{X}}_{k,m}^L) \right|}, \quad (3)$$

where $\tilde{\mathbf{X}}_{k,m}^L$ is the coordinates of point m in $\tilde{\mathcal{P}}_k$. @Frank Dellaert

To verify that j, l , and m are all planar points, we check again the smoothness of the local surface based on the formula ;

to evaluate the smoothness of the local surface,

$$c = \frac{1}{|S| \cdot \|\mathbf{X}_{k,i}^L\|} \left\| \sum_{j \in S, j \neq i} (\mathbf{X}_{k,i}^L - \mathbf{X}_{k,j}^L) \right\|. \quad (1)$$

知乎 @Frank Dellaert

秦通改写的LOAM, 搜索当前激光点附近的面点数量是 5 个, 用5个点去拟合平面法向量, 下面最关键一步单拿出来分析, 如果需要了解Eigen::Matrix< >::Ones(), 请参考^[2]; $\mathbf{x} = \mathbf{A}.\text{colPivHouseholderQr}().\text{solve}(\mathbf{b})$; 函数, 求解的就是 $\mathbf{Ax}=\mathbf{b}$ 中的 \mathbf{x} 请参考^[3], 为什么求 matB0 的元素全是 -1, 就是求解平面方程 $\mathbf{Ax} + \mathbf{By} + \mathbf{Cz} = -1$;

```
Eigen::Matrix<double, 5, 3> matA0;
Eigen::Matrix<double, 5, 1> matB0 = -1 * Eigen::Matrix<double, 5, 1>::Ones();
Eigen::Vector3d norm = matA0.colPivHouseholderQr().solve(matB0); // find the norm
```

2.2 点到平面距离公式^[4]

ALOAM 里的 `negative_OA_dot_norm`: 表达法向量模的倒数^[5], 它的作用解析如下:

如果矩阵 \mathbf{D} 是单位列向量矩阵, `Eigen::Matrix<double, n, 1>::Ones()`; 那就满足题目要求;

知乎

首发于
FAST_LIO2

...

写文章



赞同 70



添加评论

分享

喜欢

收藏

申请转载



知乎

首发于

FAST_LIO2

...

写文章



赞同 70



添加评论



分享



喜欢



收藏



申请转载

...



点: (x_0, y_0, z_0)

到平面: $Ax + By + Cz + D = 0$

距离为: $d = \frac{|Ax_0 + By_0 + Cz_0 + D|}{\sqrt{A^2 + B^2 + C^2}}$

知乎 @Frank Dellaert

当用点法式求解平面方程时候^[6] 过程如下:

2. 一般式方程

由平面的点法式方程

$$A(x - x_0) + B(y - y_0) + C(z - z_0) = 0$$

$$\Rightarrow Ax + By + Cz - (Ax_0 + By_0 + Cz_0) = 0$$

$= D$

$Ax + By + Cz + D = 0$ 平面的一般式方程

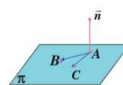
法向量 $\vec{n} = (A, B, C)$.

<https://zhuanlan.zhihu.com/p/5026538391>

例 1 求过三点 $A(2, -1, 4)$ 、 $B(-1, 3, -2)$ 和 $C(0, 2, 3)$ 的平面方程.

解 $\vec{AB} = (-3, 4, -6)$

$\vec{AC} = (-2, 3, -1)$



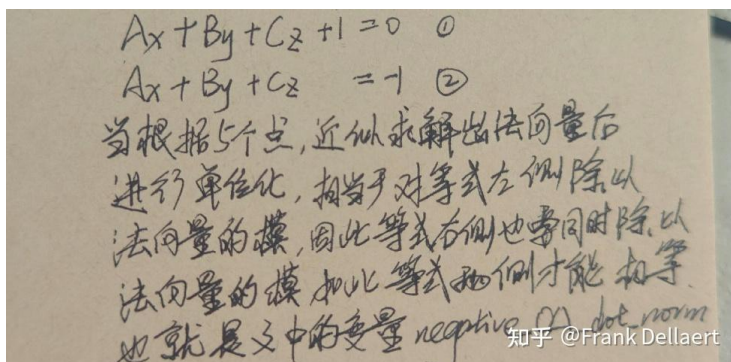
取 $\vec{n} = \vec{AB} \times \vec{AC} = (14, 9, -1)$,

所求平面方程为 $14(x - 2) + 9(y + 1) - (z - 4) = 0$,

化简得 $14x + 9y - z - 15 = 0$.

知乎 @Frank Dellaert

进一步分析如下^[7]:



知乎 @Frank Dellaert



```
kdtreeSurfFromMap->nearestKSearch(pointSel, 5, pointSearchInd, pointSearchSqDis);
Eigen::Matrix<double, 5, 3> matA0;
Eigen::Matrix<double, 5, 1> matB0 = -1 * Eigen::Matrix<double, 5, 1>::Ones();
.....
for (int j = 0; j < 5; j++)
{
    matA0(j, 0) = laserCloudSurfFromMap->points[pointSearchInd[j]].x;
    matA0(j, 1) = laserCloudSurfFromMap->points[pointSearchInd[j]].y;
    matA0(j, 2) = laserCloudSurfFromMap->points[pointSearchInd[j]].z;
    //printf(" pts %f %f %f ", matA0(j, 0), matA0(j, 1), matA0(j, 2));
}
// find the norm of plane
Eigen::Vector3d norm = matA0.colPivHouseholderQr().solve(matB0);
double negative_OA_dot_norm = 1 / norm.norm();
norm.normalize();
```

如果5个点中如果有一个偏差较大, 就是break这个循环

```
bool planeValid = true;
for (int j = 0; j < 5; j++)
{
    // if OX * n > 0.2, then plane is not fit well
    if (fabs(norm(0) * laserCloudSurfFromMap->points[pointSearchInd[j]].x +
            norm(1) * laserCloudSurfFromMap->points[pointSearchInd[j]].y +
            norm(2) * laserCloudSurfFromMap->points[pointSearchInd[j]].z + negative.
    {
        planeValid = false;
        break;
    }
}
```

三、FAST_LIO2 拟合平面和构建残差

3.1 FAST_LIO2 搜索当前点的最近邻点时, 为什么没区分角点和平面点?

搜索最近邻的原点 表述的就是FAST_LIO2代码中变量 **point_world**的含义, 在定位时读取到原始的当前帧点云, 再进行降采样后的点云, 这个过程完全没有区分是corner点还是plane点, 在实际应用中, 却没有加以区分, 但当我们搜索到当前激光点附近的5个点然后进行拟合平面时, 就是对当前激光点进行了一次**check**, (这里有一个前提是: 当前机器人的全局位置与地图中的偏差是在LiDAR Odom或IMU一个预测环节的里程计范围之内, 偏差过大就会出现为题, 定位飘了), 括号内的距离是对激光是否是平面类型的激光点的**二次check**, 虽然没直接对雷达当前激光点进行区分, 但间接也相当于进行了区分。

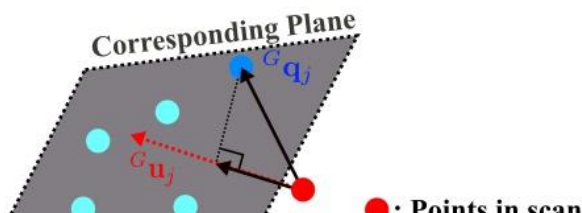
```
ikdtree.Nearest_Search(ikdtree.Nearest_Search(point_world, NUM_MATCH_POINTS, poin
```

3.2 FAST_LIO2 拟合平面

以下文章如果没有特别说明, 提及FAST_LIO都是指FAST_LIO2;

fast lio2比较特别的一点是删除了特征提取模块, 而是直接将点云配准到地图^[8];

paper显示: 使用5个地图点, 下面的代码内容和它的内容是一致的, 所以这也是FAST_LIO2 paper比LOAM paper好的地方, 文章和代码内容一致;





G_{q_j} is a point lying on the plane.

知乎 @Frank Dellaert

代码显示: 寻找point_world最近邻的 5 个平面点, 具体详细过程可参考^[9]

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

找到最近的5个点, 然后再经过必要的筛选后, 拟合平面方程 $ax+by+cz+d=0$ 并求解点到平面距离float pd2 ;

```
if (esti_plane(pabcd, points_near, 0.1f))
{
    //当前点到平面的距离
    float pd2 = pabcd(0) * point_world.x + pabcd(1) * point_world.y+ pabcd(2)*point_world.z + pabcd(3);
    float s = 1 - 0.9 * fabs(pd2) / 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 要求点到平面的残差越小
    if (s > 0.9) //如果残差大于阈值, 则认为该点是有效点
    {
        point_selected_surf[i] = true;
        normvec->points[i].x = pabcd(0); //存储平面的单位法向量 以及当前点到平面的距离
        normvec->points[i].y = pabcd(1);
        normvec->points[i].z = pabcd(2);
        normvec->points[i].intensity = pd2;
    }
}
```

3.3 FAST_LIO2 如何构建残差函数

之前需要知道残差是什么, fastlio2是怎样构造点云和地图之间的残差函数:

求观测函数的偏导本质就是为了线性化观测方程: 可需要再写一篇文档;

-----写给未来

1. 可将高翔关于激光的那本书中论述拟合平面的内容融合到这个里面;

参考

1. ^ 点拟合平面 https://blog.csdn.net/qq_37611824/article/details/128341199
2. ^ 2.4 MatrixXd全部元素为固定值的初始化 <https://zhuanlan.zhihu.com/p/704663536>
3. ^ https://blog.csdn.net/weixin_42156097/article/details/107702367
4. ^ 点到平面的距离 <https://www.cnblogs.com/chenlinchong/p/16133350.html>
5. ^ <https://bbs.huaweicloud.com/blogs/367156>
6. ^ 平面方程 (一) <https://blog.csdn.net/hpdlzu80100/article/details/100087733>
7. ^ https://math.fudan.edu.cn/_upload/article/files/8a/c7/b6bc60b64933acee041a351adda9/d0bb8522-a61d-4d6c-ab50-0051ba33a403.pdf
8. ^ <https://www.zhihu.com/question/552003749>
9. ^ 3.3 筛选 当前帧激光点中的平面点 <https://zhuanlan.zhihu.com/p/5232561677>

送礼物

还没有人送礼物, 鼓励一下作者吧

所属专栏 · 2024-11-27 22:14 更新

赞同 70

添加评论

分享

喜欢

收藏

申请转载

...



编辑于 2024-12-03 15:38 · 中国香港

FAST-LIO 激光SLAM LOAM



理性发言，友善互动



还没有评论，发表第一个评论吧

推荐阅读

SLAM14讲第六章 非线性优化

前言：经典SLAM模型的运动方程可以有变换矩阵描述，并由李代数进行优化；观测方程由相机模型给出，根据针孔成像原理，内参由相机标定后得到，外参是相机的位姿。但是由于噪声的存在，使得...

宁宁



SLAM系统地图表示| SLAM: 现在，未来和鲁棒年代（四）...

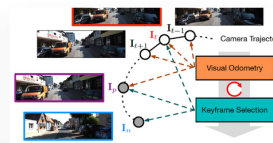
泡泡机器人



几何vs 目标，哪种SLAM方法更主流？

深蓝学院

发表于自动驾驶技...



推荐一些视觉SLAM的深度学习方法（上）

黄浴

发表于深度学习在...