

# 3D激光SLAM介绍



**主讲人 曾书格**

越凡创新技术负责人  
597457483@qq.com





### 1、3D激光SLAM

3D激光SLAM



### 2、LOAM



## 1、3D激光SLAM

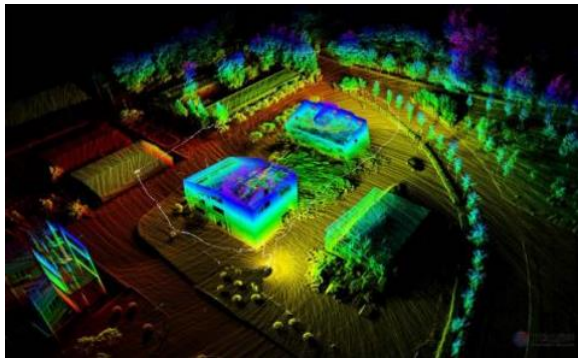
3D激光SLAM



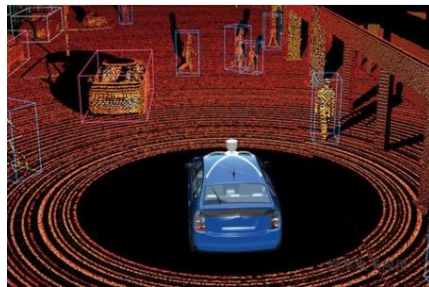
## 2、LOAM

## 介绍

- 地图形式
- 主要应用领域



3D激光地图



无人驾驶汽车



送外卖小车

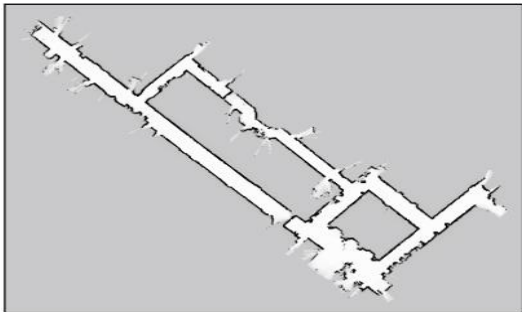


# 3D激光SLAM—对比



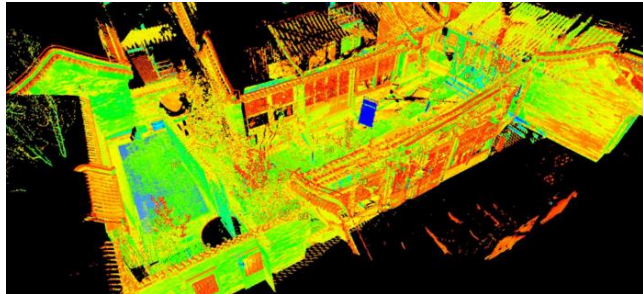
## 2D激光SLAM

- 单线激光雷达
- 二维栅格地图
- 室内移动机器人



## 3D激光SLAM

- 多线激光雷达
- 三维点云地图或者三维栅格地图
- 室外移动机器人





## 3D激光SLAM—帧间匹配

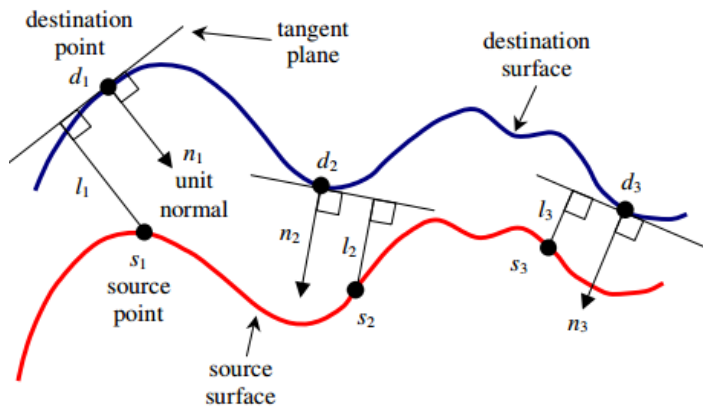


### 帧间匹配算法—Point-to-Plane ICP(点面ICP)

- 目标函数:

$$\mathbf{M}_{\text{opt}} = \arg \min_{\mathbf{M}} \sum_i ((\mathbf{M} \cdot \mathbf{s}_i - \mathbf{d}_i) \cdot \mathbf{n}_i)^2$$

- 示意图:





## 3D激光SLAM—帧间匹配

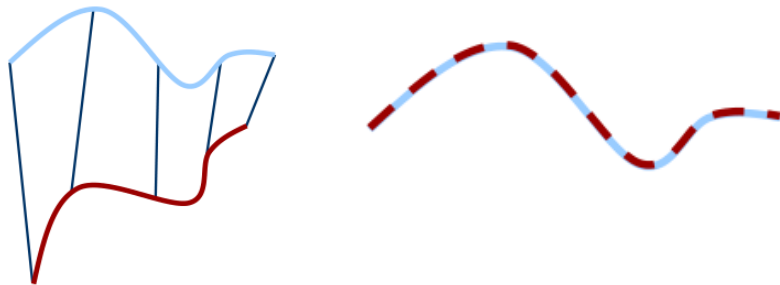


### 帧间匹配算法—Feature-based Method

- 提取特征点：VPF、FHPF、角点、平面点等
- 基于已知对应点的ICP方法：

$$R = VU^T$$

$$t = u_x - Ru_p$$





## 3D激光SLAM—帧间匹配



### 帧间匹配算法—Normal Distribution Transform(NDT方法)

- 把空间划分成小方格
- 点云数据在小方格中服从高斯分布，并根据点云数据计算出该小方格中对应的高斯分布参数：

$$u = \frac{1}{n} \sum x_i \quad \Sigma = \frac{1}{n} \sum (x_i - u)(x_i - u)^T$$

- 落在该方格内的点 $x$ ，其得分为：

$$s \approx e^{-\frac{1}{2}(x-u)^T \Sigma^{-1} (x-u)}$$

- 目标函数为：

$$Score = \sum e^{-\frac{1}{2}(x_i-u)^T \Sigma^{-1} (x_i-u)}$$





## 课程内容



### 1、3D激光SLAM

3D激光SLAM



### 2、LOAM

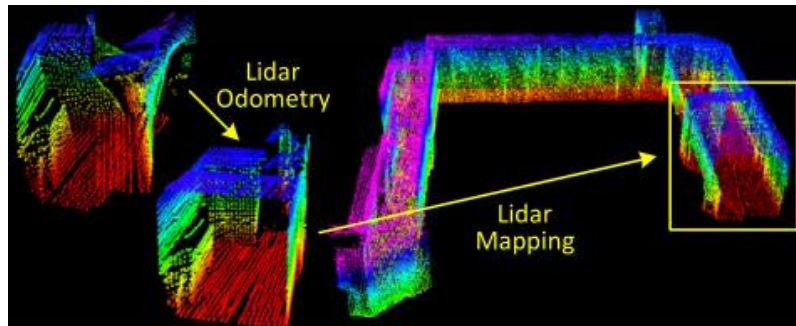


# 3D激光SLAM—Lidar Odometry And Mapping(LOAM)



## 介绍

- 3D激光“SLAM” 算法
- 为数不多的开源3D激光SLAM算法之一
- KITTI数据集Odometry排行榜上长期霸占第一
- Feature-based 匹配方法



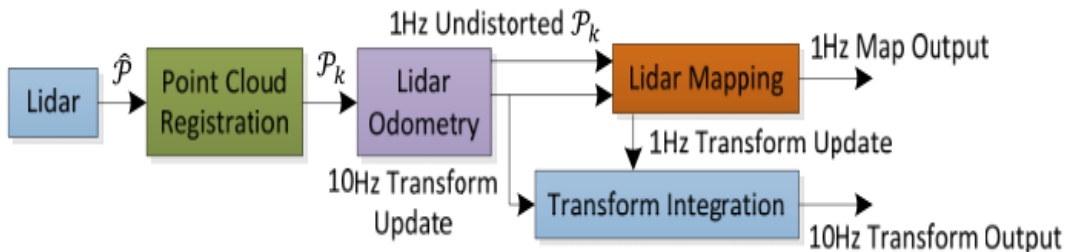


# 3D激光SLAM—Lidar Odometry And Mapping(LOAM)



## 框架

- 激光的帧率为1Hz
- 匀速运动假设
- 激光里程计模型
- 激光建图模块





# 3D激光SLAM—Lidar Odometry And Mapping(LOAM)

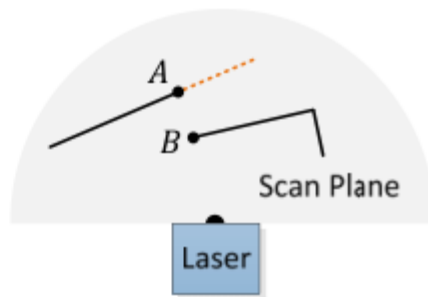
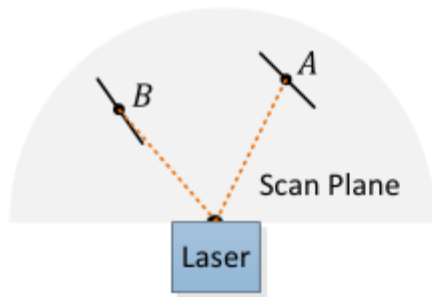


## 激光里程计模块—特征点提取模块

- 计算曲率:

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

- 曲率项最大为Edge Point, 最小为Planar Point。
- 每一个水平激光帧, 分为四个区域; 每个区域最多有2个edge point和4个planar point。
- 去除不稳定特征点:



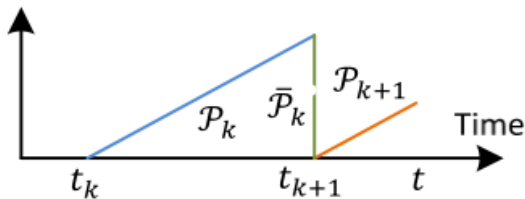


# 3D激光SLAM—Lidar Odometry And Mapping(LOAM)

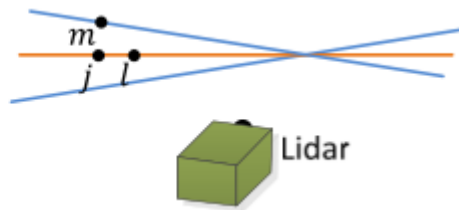
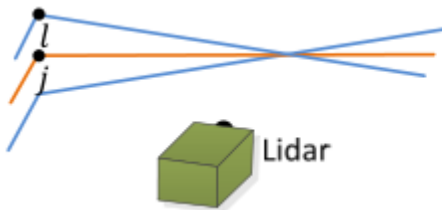


## 激光里程计模块—特征点匹配

- 前一帧的完整数据数据为 $\bar{P}_{k-1}$ ，当前的数据为 $P_k$ 。每一帧数据都由多个scan组成。



- 为当前数据 $P_k$ 中的edge Point和planar Point:





# 3D激光SLAM—Lidar Odometry And Mapping(LOAM)



## 激光里程计模块—特征点匹配

- Edge Point的目标函数:

$$d_{\mathcal{E}} = \frac{\left| (\tilde{\mathbf{X}}_{(k+1,i)}^L - \bar{\mathbf{X}}_{(k,j)}^L) \times (\tilde{\mathbf{X}}_{(k+1,i)}^L - \bar{\mathbf{X}}_{(k,l)}^L) \right|}{\left| \bar{\mathbf{X}}_{(k,j)}^L - \bar{\mathbf{X}}_{(k,l)}^L \right|}$$

- Planar Point的目标函数:

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



# 3D激光SLAM—Lidar Odometry And Mapping(LOAM)

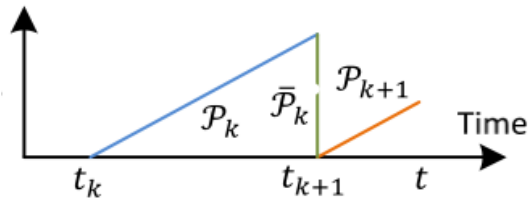


## 激光里程计模块—运动估计

- 认为在整个sweep期间，机器人在匀速运动；令 $T_k^L(t)$ 表示 $t$ 时刻相对于 $t_k$ 时刻的机器人位姿，则对于当前帧的第 $i$ 个激光点，设其对应的时间为 $t_{(k,i)}$ ，则其对应的位姿为：

$$T_{(k,i)}^L = \frac{t_{(k,i)} - t_k}{t - t_k} T_k^L(t)$$

- 所有的点都投影到同一时刻 $t_k$ ：



$$\tilde{X}_{(k,i)}^L = \mathbf{R}_{(k,i)}^L \mathbf{X}_{(k,i)}^L + \tau_{(k,i)}^L$$



## 3D激光SLAM—Lidar Odometry And Mapping(LOAM)



### 激光里程计模块—运动估计

- 构造目标函数：

$$f_{\mathcal{E}}(X_{(k,i)}^L, T_k^L(t)) = d_{\mathcal{E}}, \quad i \in \mathcal{E}_k.$$

$$f_{\mathcal{H}}(X_{(k,i)}^L, T_k^L(t)) = d_{\mathcal{H}}, \quad i \in \mathcal{H}_k.$$

$$f(T_k^L(t)) = d$$

- 非线性优化求解：

$$\mathbf{J} = \partial f / \partial T_k^L(t)$$

$$T_k^L(t) \leftarrow T_k^L(t) - (\mathbf{J}^T \mathbf{J} + \lambda \text{diag}(\mathbf{J}^T \mathbf{J}))^{-1} \mathbf{J}^T d$$





# 3D激光SLAM—Lidar Odometry And Mapping(LOAM)



## 激光里程计模块—总结

### Algorithm 1: Lidar Odometry

```
1 input :  $\bar{\mathcal{P}}_{k-1}, \mathcal{P}_k, T_k^L(t)$  from the last recursion at initial guess
2 output :  $\bar{\mathcal{P}}_k$ , newly computed  $T_k^L(t)$ 
3 begin
4   if at the beginning of a sweep then
5     |  $T_k^L(t) \leftarrow 0$ ;
6   end
7   Detect edge points and planar points in  $\mathcal{P}_k$ , put the points in
    $\mathcal{E}_k$  and  $\mathcal{H}_k$ , respectively;
8   for a number of iterations do
9     | for each edge point in  $\mathcal{E}_k$  do
10      | Find an edge line as the correspondence, then
        | compute point to line distance based on (7) and stack
        | the equation to (9);
11    | end
12    | for each planar point in  $\mathcal{H}_k$  do
13      | Find a planar patch as the correspondence, then
        | compute point to plane distance based on (8) and
        | stack the equation to (9);
14    | end
15    | Compute a bisquare weight for each row of (9);
16    | Update  $T_k^L(t)$  for a nonlinear iteration based on (10);
17    | if the nonlinear optimization converges then
18      | Break;
19    | end
20  | end
21  if at the end of a sweep then
22    | Project each point in  $\mathcal{P}_k$  to  $t_{k+1}$  and form  $\bar{\mathcal{P}}_k$ ;
23    | Return  $T_k^L(t)$  and  $\bar{\mathcal{P}}_k$ ;
24  | end
25  else
26    | Return  $T_k^L(t)$ ;
27  | end
28 end
```

- 特征点检测
- 寻找匹配点
- 构建非线性方程组
- 求解非线性方程组
- 如果到了当前帧的末尾，则投影到当前帧的时刻，开始进行下一帧的求解

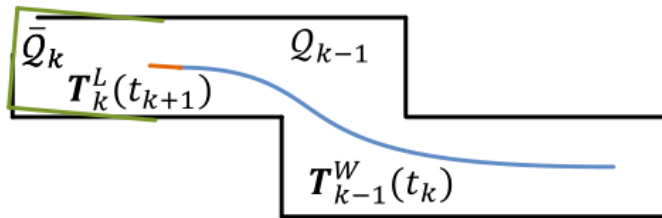


# 3D激光SLAM—Lidar Odometry And Mapping(LOAM)



## 激光建图模块

- Lidar Odometry的输出跟地图进行匹配:



- Mapping过程提取的特征点数量，是Lidar Odometry的10倍。
- 特征点的匹配不是寻找对应的2个或者三个特征点。而是对于当前帧的特征点，在 $Q_{k-1}$ 中的对应位置附近10cm\*10cm\*10cm的cubic中提取所有的特征点。



## 3D激光SLAM—Lidar Odometry And Mapping(LOAM)



### 激光建图模块—直线匹配

- 对于Edge Point, cubic中的所有Edge Point会按照直线进行分布。根据这些特征点求解出直线的方程
- 根据Cubic中的Edge Point, 计算位姿协方差矩阵。
- 对协方差矩阵进行特征值分解。
- 最大特征值对应的特征向量即为直线的方向向量。
- 该直线通过所有Edge Point的几何中心



## 3D激光SLAM—Lidar Odometry And Mapping(LOAM)



### 激光建图模块—平面匹配

- 对于Planar Point, cubic中的所有Planar Point会按照平面进行分布。根据这些特征点求解出平面的方程。
- 根据Cubic中的Planar Point, 计算位姿协方差矩阵。
- 对协方差矩阵进行特征值分解。
- 最小特征值对应的特征向量, 即为平面的法向量。
- 该平面通过所有Planar point的几何中心

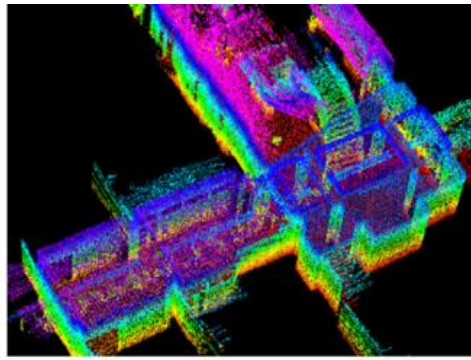
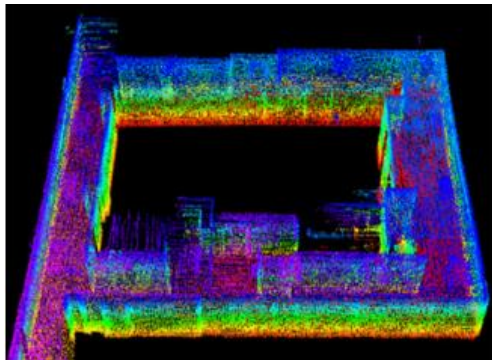


# 3D激光SLAM—Lidar Odometry And Mapping(LOAM)



## 激光建图模块

- 按照Lidar Odometry模块的估计方式进行运动估计即可，最终的结果如下：





结语

感谢各位聆听!

Thanks for Listening