

# 自动驾驶与机器人中的 SLAM技术

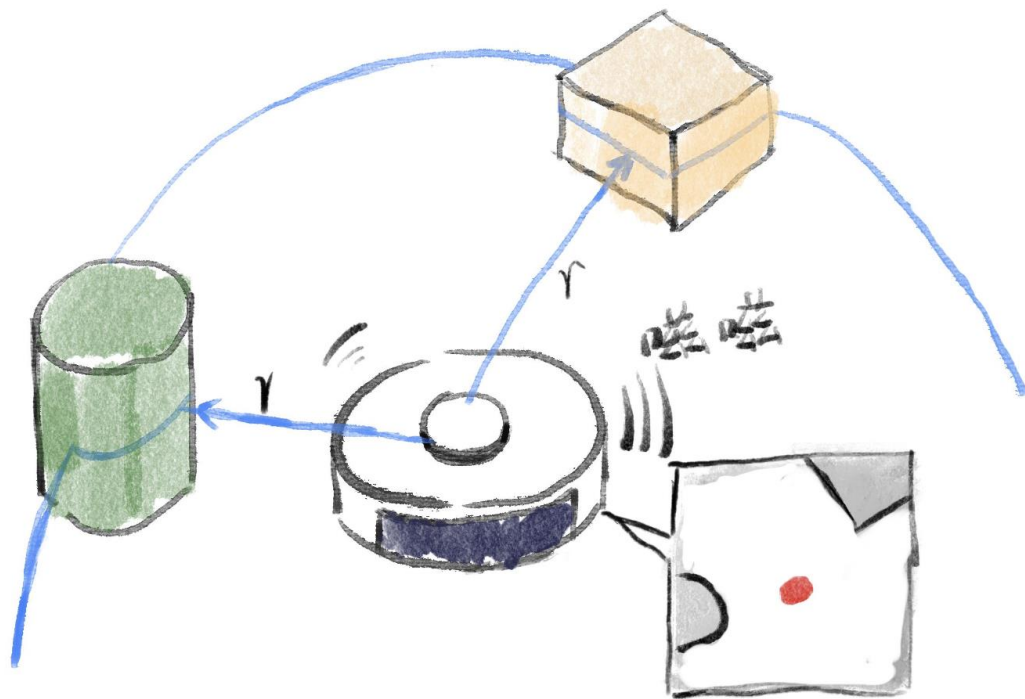
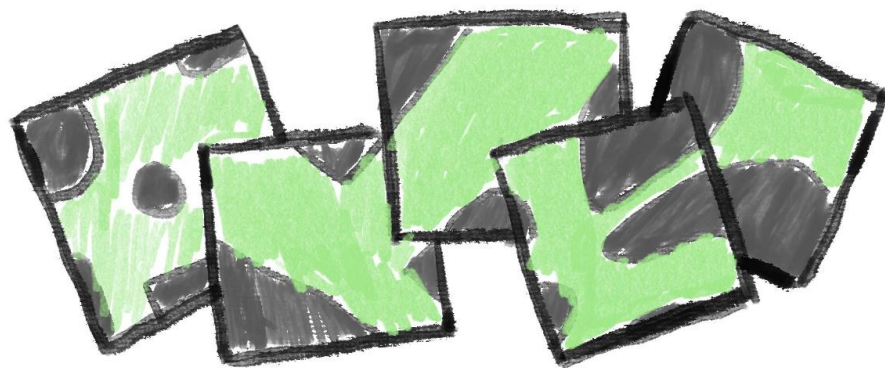
2D激光SLAM





# Contents

- 2D SLAM的假设与基本原理
- Scan Matching方法
- 占据栅格地图
- 子地图、回环检测与闭环



2D场景可以简化很多问题。



## 2D SLAM的假设与基本原理



## 2D SLAM的假设与基本原理

□ 许多现实中的机器人仅在水平面中运动，于是对环境作2D平面假设

□ 平面假设的优点：

- 节省计算资源；
- 直观地使用2D图像表示地图；
- 更容易判定、更新地图上的障碍物；
- 地图标注更加容易；
- 单线雷达价格低廉（百元级）；

□ 平面假设的缺点：

- 难以表达多个高度上的信息；
- 环境不符合平面假设时，算法容易出现問題。

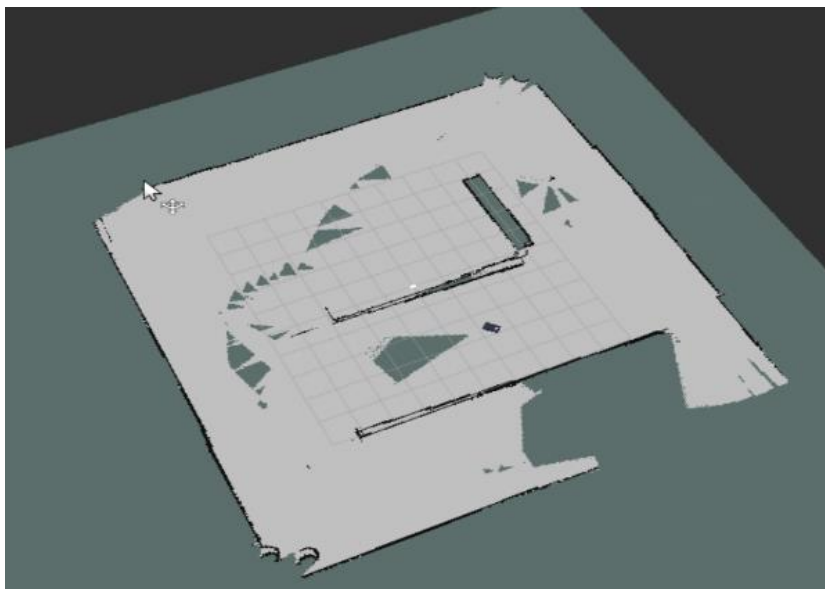


- 一些使用2D SLAM的机器人和传感器
- 一部分使用多线雷达的设备也会转换为2D地图
- 大部分高精地图在使用时只需要2D信息

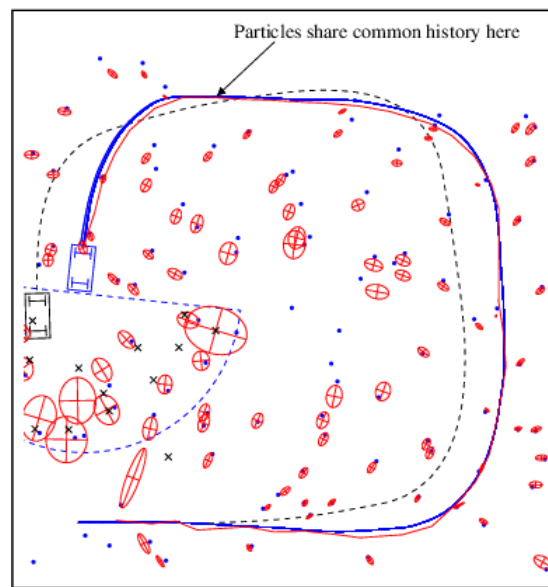


# 2D SLAM的假设与基本原理

## □ 2D激光SLAM是早期SLAM研究焦点



gmapping



fastslam



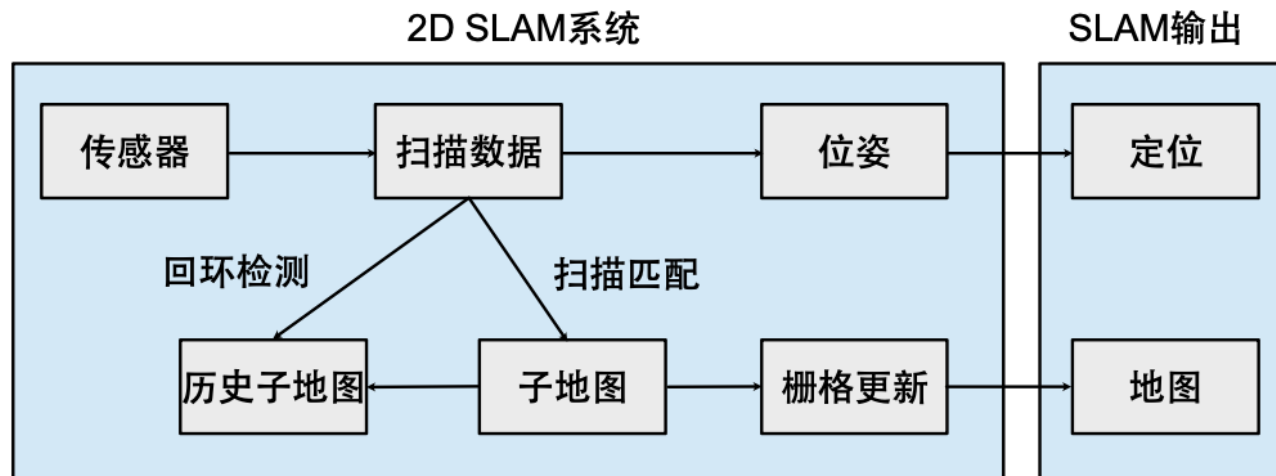
# 2D SLAM的假设与基本原理

## 现代2D SLAM的框图

### 相比早期方案的差异：

- 早期方案不区分前后端（EKF/PF），现代方案通常使用pose graph作为后端；
- 早期方案只使用一整张地图，现代方案以关键帧或子地图作为处理单位；
- 早期方案的回环检测比较粗糙，现代方案一般都带有回环检测，且要求闭环后的地图完整性。

其中最基本的就是scan matching方法



### 主要概念：

- Scan：一次扫描数据；
- Scan matching：将两个scan进行匹配，得出相对运动；
- Submap：一定数量的scan合在一起形成的地图；
- Occupancy grid：以占据概率表达的2D地图。



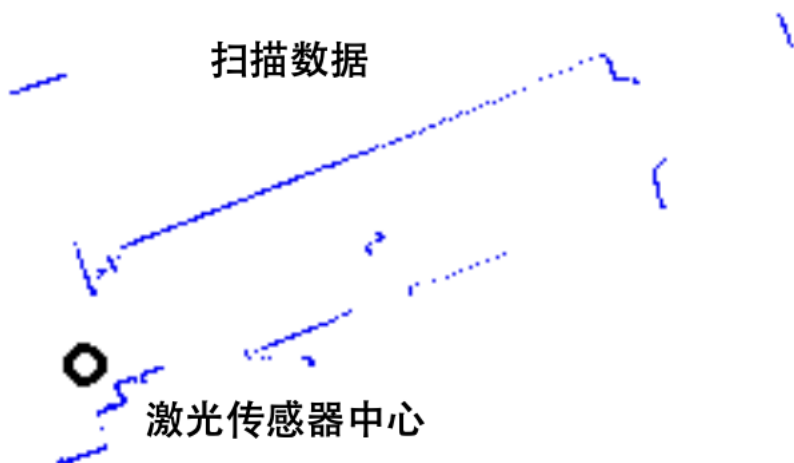
## Scan Matching算法



# Scan Matching算法

## □ 2D激光的Scan数据

## □ 2D激光的测量可以从RAE模型简化为距离和角度读数（极坐标）： $(\rho, r)_i$



单次扫描数据的可视化展示

- 大部分激光是360度的，但具有一定高度的机器人，通常无法让激光具有完整的360度视野（底座、人员会遮挡一部分）。
- 蓝色点是实际激光的测量点，称为末端点（end points）。
- 末端点有两层物理含义：
  1. 末端位置存在一个障碍物；
  2. 从传感器中心到末端位置的连线上没有障碍物。注意：第2条并不是直接测量数据，需要计算（ray casting和rasterization）。
- 激光SLAM需要利用上述两条性质来进行定位建图。
- 如果环境、传感器存在倾斜、透射、反射等情况，那么上述两条性质就不正确。





# Scan Matching算法

- Scan Matching问题：给定两个scan，估计相对运动

观测模型  $\mathbf{z} = \mathbf{h}(\mathbf{x}, \mathbf{m}) + \mathbf{w}$ ,                      最大似然  $\mathbf{x}_{\text{MLE}} = \arg \max p(\mathbf{x}|\mathbf{z}, \mathbf{m}) = \arg \max p(\mathbf{z}|\mathbf{x}, \mathbf{m})$ .

- Scan Matching存在不同的问题模型和解法

- ICP类 (ICP, PL-ICP, ICL, GICP)
- 似然场类 (高斯似然场、CSM等)

- 大部分算法有对应的3D版本，将在下节课的registration方法中介绍。



# Scan Matching算法

## □ Scan Matching中要解决的问题

- 对哪些点进行Scan Matching? ——全部点/采样点/特征点
- 如何确定两个Scan中点和点的对应关系? ——数据关联
- 如何计算残差? ——残差的建模和参数估计

□ 每个部分都存在不同的做法，因此各种算法的排列组合也非常多。



# Scan Matching算法

## □ 2D ICP

核心：交替求解数据关联问题和位姿估计问题

使用这种思路的算法也可以称为类ICP算法。

2D位姿：  $\mathbf{x} = [x, y, \theta]^\top$

注：程序统一使用SE2接口，但公式上展开书写

坐标系：  $\mathbf{x} = \mathbf{T}_{WB}$

和书中其他部分一致（但后文还有一个子地图坐标系）

将一个激光数据  $(\rho, r)_i$  转换到世界坐标系：

$$\mathbf{p}_i^W = [x + r_i \cos(\rho_i + \theta), y + r_i \sin(\rho_i + \theta)]^\top.$$

假设此时它的最近邻为： $\mathbf{q}_i^W$ ，那么最简单的残差形式为：

$$\mathbf{e}_i = \mathbf{p}_i^W - \mathbf{q}_i^W,$$



# Scan Matching算法

容易求出该残差对位姿变量的导数（2D位姿可以直接求）：

$$\frac{\partial \mathbf{e}_i}{\partial x} = [1, 0]^\top,$$

$$\frac{\partial \mathbf{e}_i}{\partial y} = [0, 1]^\top,$$

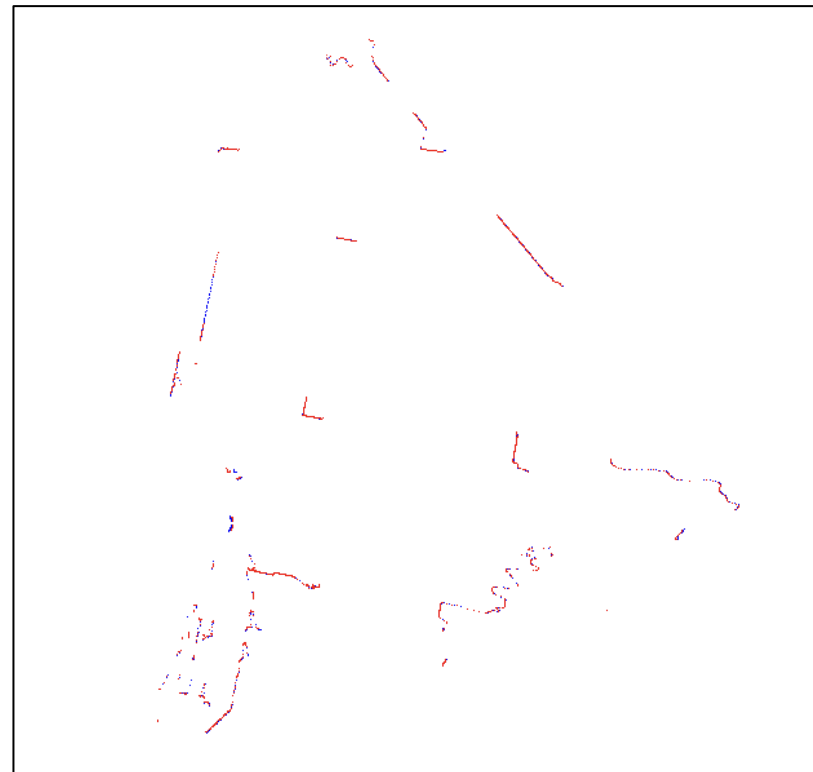
$$\frac{\partial \mathbf{e}_i}{\partial \theta} = [-r_i \sin(\rho_i + \theta), r_i \cos(\rho_i + \theta)]^\top.$$

矩阵形式：

$$\frac{\partial \mathbf{e}_i}{\partial \mathbf{x}} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -r_i \sin(\rho_i + \theta) & r_i \cos(\rho_i + \theta) \end{bmatrix} \in \mathbb{R}^{3 \times 2}.$$

$$\mathbf{p}_i^W = [x + r_i \cos(\rho_i + \theta), y + r_i \sin(\rho_i + \theta)]^\top.$$

$$\mathbf{e}_i = \mathbf{p}_i^W - \mathbf{q}_i^W,$$



Scan Matching可以把原本不同时刻的scan拼到一起



# Scan Matching算法

## □ 点到线的ICP

- 原始ICP使用点到点残差，可能受点的噪声影响；将若干个目标点拟合成某个形状再进行匹配，可以减小单个点噪声的影响；
- 2D直线参数：

$$ax + by + c = 0,$$

- 2D直线拟合：

$$(a, b, c)^* = \arg \min \sum_{i=1}^N \|ax_i + by_i + c\|_2^2.$$

- 直线参数对应 $A$ 的最小奇异向量：

$$A = \begin{bmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ & \dots & \\ x_k & y_k & 1 \end{bmatrix}$$



# Scan Matching算法

□ 点线残差:  $d = \frac{ax + by + c}{\sqrt{a^2 + b^2}},$

分母为常数, 可以省略

残差:  $e = ax + by + c, \quad \frac{\partial e}{\partial x} = a, \quad \frac{\partial e}{\partial y} = b.$

雅可比:  $\frac{\partial e_i}{\partial \mathbf{x}} = \frac{\partial e_i}{\partial \mathbf{p}_i^w} \frac{\partial \mathbf{p}_i^w}{\partial \mathbf{x}},$

易得:  $\frac{\partial e_i}{\partial \mathbf{x}} = [a_i, b_i, -a_i r_i \sin(\rho_i + \theta) + b_i r_i \cos(\rho_i + \theta)]^\top.$

点到点:

$$\frac{\partial e_i}{\partial x} = [1, 0]^\top,$$

$$\frac{\partial e_i}{\partial y} = [0, 1]^\top,$$

$$\frac{\partial e_i}{\partial \theta} = [-r_i \sin(\rho_i + \theta), r_i \cos(\rho_i + \theta)]^\top.$$

- 对比点到点ICP, 可见直线参数也参与了雅可比的计算过程
- 3D的点面ICP亦有类似结论



# Scan Matching算法

## □ ICP类方法最终由最小二乘描述，可直观地视为弹簧——质点模型：

认为每个点和它的最近邻之间存在一个弹簧，弹簧的拉力传递后，将传感器拉到能量最小的位姿上，但是，ICP方法在每次迭代时，需要重新系一遍这些弹簧。能否不需要重新系弹簧，就能产生拉力呢？

## □ 似然场法

- 每个点周围存在一个吸引力场，合并之后产生一个大的吸引力场（似然场，Likelihood Filed）；
- 吸引力随距离平方衰减；
- 可以和栅格地图一起工作，为方便实现，还需要设置分辨率和有效范围。



# Scan Matching算法

□ 一个Scan或一个地图都可以产生似然场

一个点落在似然场中产生读数： $\pi(\mathbf{p}_i^W)$

于是最小化： $\mathbf{x}^* = \arg \min_{\mathbf{x}} \sum_{i=1}^n \|\pi(\mathbf{p}_i^W)\|_2^2$ ,

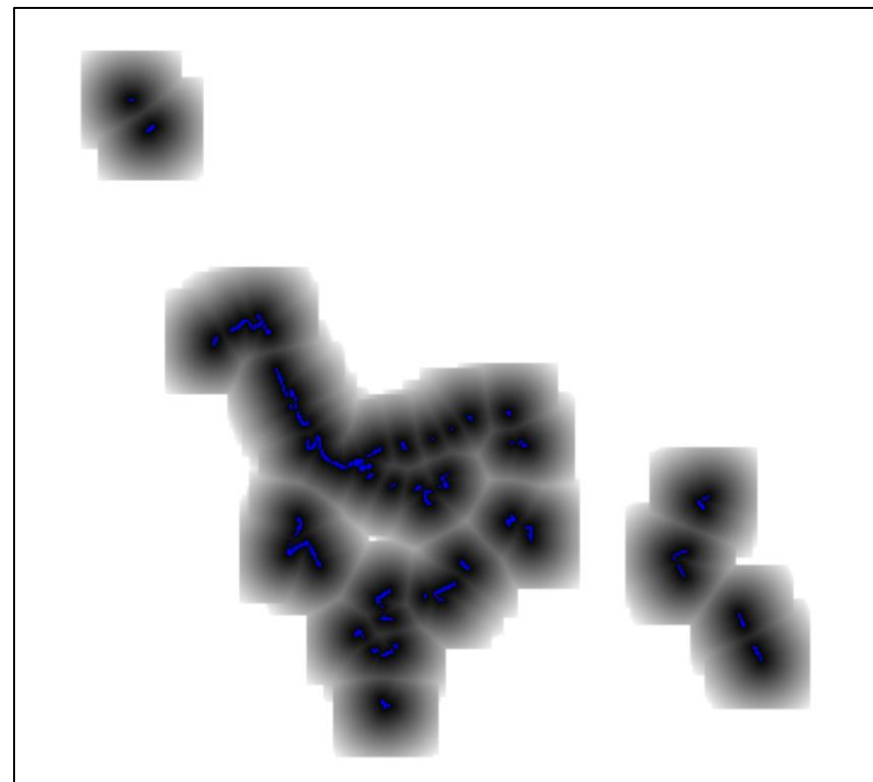
雅可比为： $\frac{\partial \pi}{\partial \mathbf{x}} = \frac{\partial \pi}{\partial \mathbf{p}_i^W} \frac{\partial \mathbf{p}_i^W}{\partial \mathbf{x}}$ .

似然场以图像存储，于是存在坐标变换： $\mathbf{p}_i^f = \alpha \mathbf{p}_i^W + \mathbf{c}$ ,

分辨率    图像中心

于是： $\frac{\partial \pi}{\partial \mathbf{p}_i^W} = \frac{\partial \pi}{\partial \mathbf{p}_i^f} \frac{\partial \mathbf{p}_i^f}{\partial \mathbf{p}_i^W} = \alpha [\Delta \pi_x, \Delta \pi_y]^\top$ ,

合在一起： $\frac{\partial \pi}{\partial \mathbf{x}} = [\alpha \Delta \pi_x, \alpha \Delta \pi_y, -\alpha \Delta \pi_x r_i \sin(\rho_i + \theta) + \alpha \Delta \pi_y r_i \cos(\rho_i + \theta)]^\top$ .







# Scan Matching算法

□ 算法的程序实现



## 占据栅格地图



## 占据栅格地图

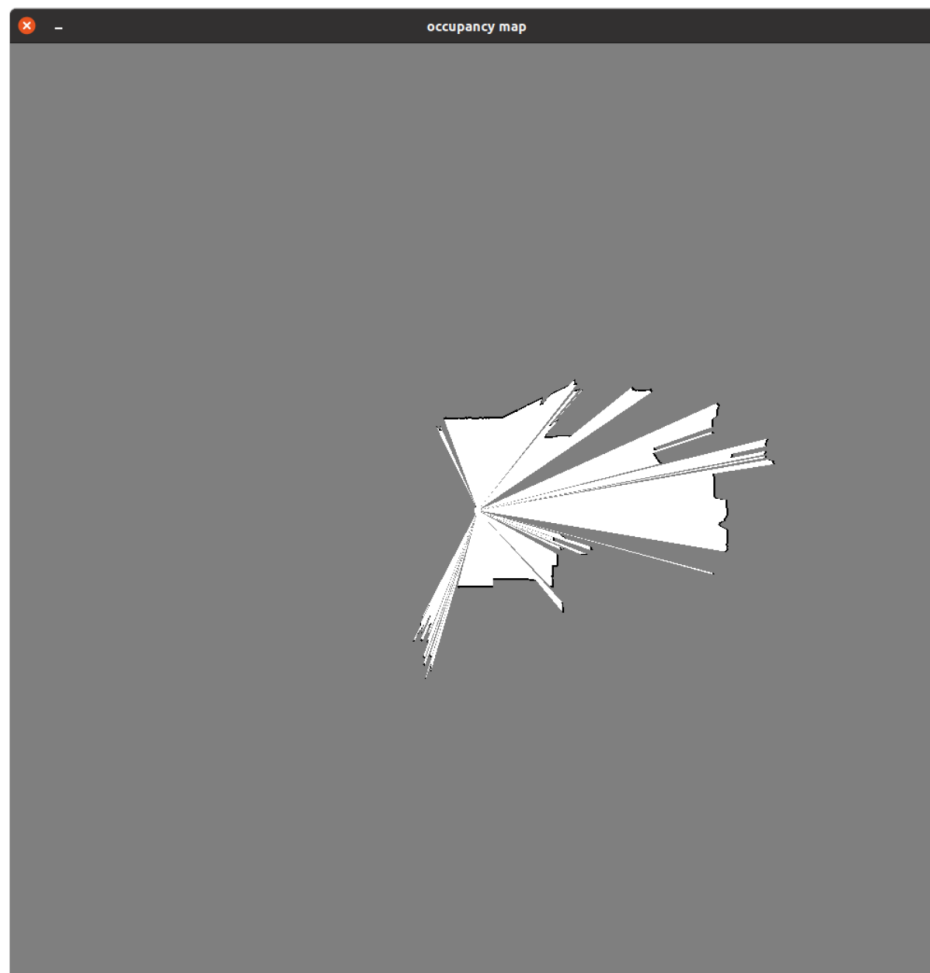
□ 占据栅格是一种非常方便的2D地图表达方式，可以视为一种由0-1浮点数表达的图像。

占据概率：

- 0表示未占据（可通行）；
- 0.5表示未探索；
- 1表示占据（不可通行）。

注意：

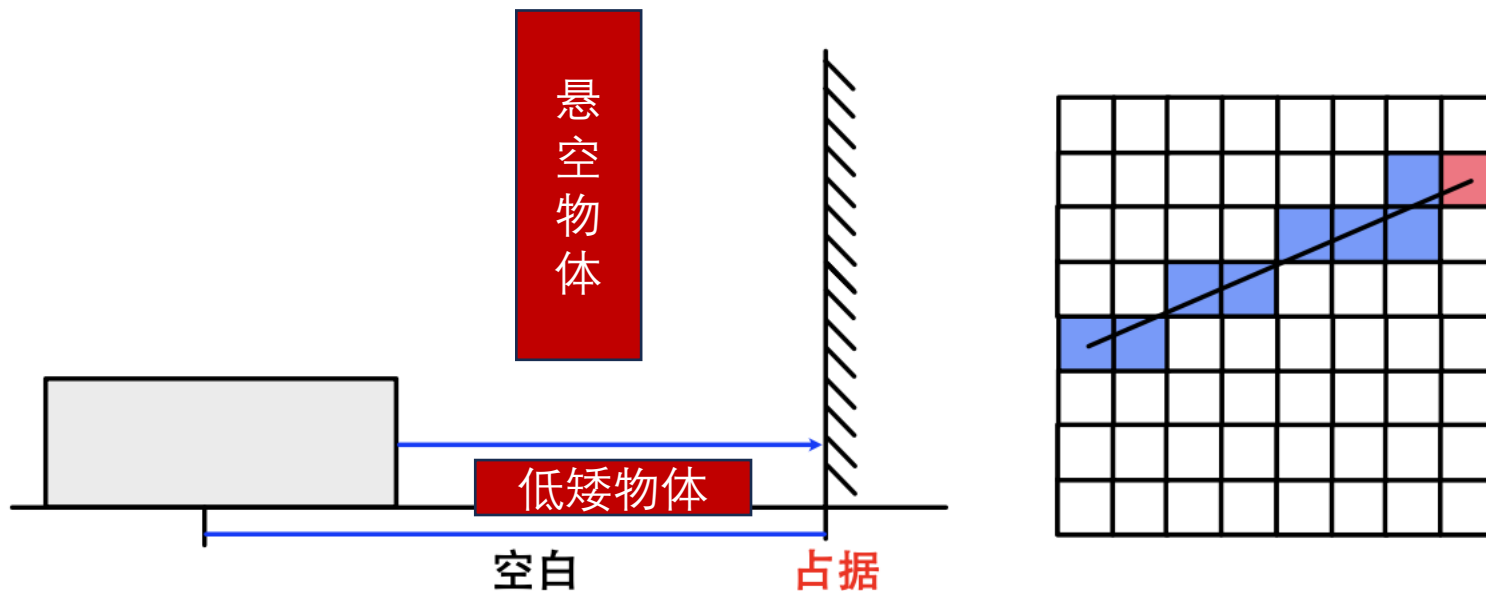
- 可以反过来定义，也可以反过来可视化；
- 可以视为染色问题；
- 工程上也可以不用浮点数实现，使用0-255整数实现；
- 工程上也存在分辨率限制，即一个像素对应物理上多少cm（常见的有5cm, 10cm, 20cm）。





## 占据栅格地图

- 当我们得到一个scan的位姿时，可以用来更新占据栅格地图。将scan数据转换为栅格地图中，需要做一些几何假设：



注意：当机器人有一定高度时，2D地图无法表达悬空物体和低矮物体。



# 占据栅格地图

## □ Bresenham算法

- 如何计算一条几何直线通过地图当中哪些栅格？

记直线上两点为 $p_1, p_2$ ，那么：

1. 记  $[dx, dy] = p_2 - p_1$ ，表示坐标增长的方向。
2. 比较  $|dx|$  和  $|dy|$ ，取大的那个为主要增长方向。不妨记为  $x$  轴。
3. 取初始的  $(x, y)$  从  $p_1$  出发。因为直线的斜率为  $dy/dx$ ，所以每当  $x$  自增 1，该点坐标与真实直线的误差就自增  $dy/dx$ 。当这个误差值大于 0.5 以后，让  $y$  增加 1，同时误差减 1。
4. 重复上述过程直到  $x, y$  到达  $p_2$  点。

上述过程中的 $dy/dx$ 是浮点数；

为避免浮点数，将第3步斜率乘 $2dx$ ，误差再减 $dx$ 。



## 占据栅格地图

1. 记  $[dx, dy] = p_2 - p_1$ ，表示坐标增长的方向。
2. 比较  $|dx|$  和  $|dy|$ ，取大的那个为主要增长方向。不妨记为  $x$  轴。
3. 取初始的  $(x, y)$  从  $p_1$  出发。因为直线的斜率为  $dy/dx$ ，所以每当  $x$  自增 1, 该点坐标与真实直线的误差就自增  $dy/dx$ 。当这个误差值大于 0.5 以后，让  $y$  增加 1, 同时误差减 1。
4. 重复上述过程直到  $x, y$  到达  $p_2$  点。

整数形式：

1. 记  $[dx, dy] = p_2 - p_1$ ，表示坐标增长的方向。
2. 比较  $|dx|$  和  $|dy|$ ，取大的那个为主要增长方向。不妨记为  $x$  轴。
3. 取初始的  $(x, y)$  从  $p_1$  出发。取初始误差为  $e = -dx$ 。每当  $x$  自增 1,  $e$  增加  $2dy$ 。若  $e > 0$ ,  $y$  自增 1,  $e$  减去  $2dx$ 。
4. 重复上述过程直到  $x, y$  到达  $p_2$  点。

若主要增长轴为  $y$  轴，则交换  $x$  和  $y$  的符号即可。

	浮点e	整数e
初始	0	$-dx$
增长	$dy/dx$	$2dy$
修正	-1	$-2dx$
判定	$e > 0.5$	$e > 0$

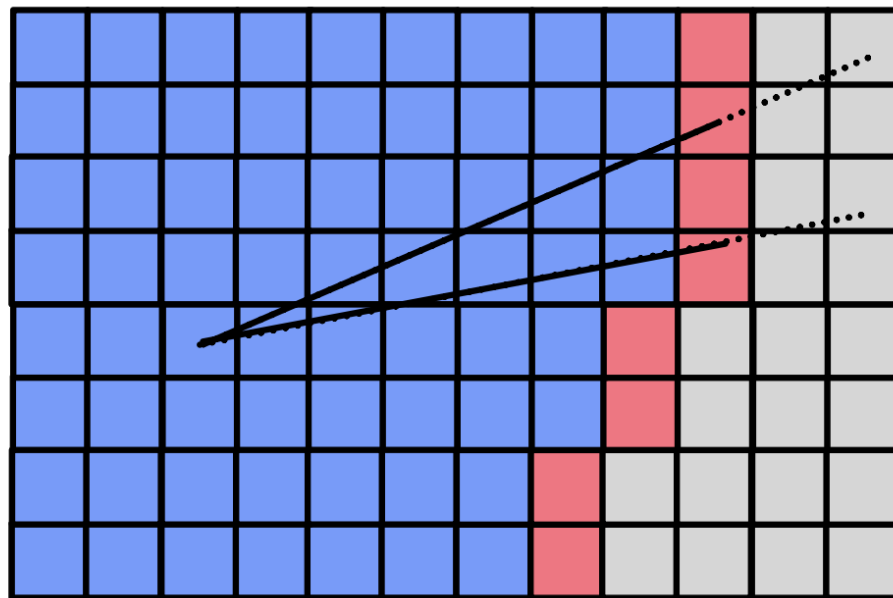


## 占据栅格地图

### □ 基于模板的填充

- 预先计算一个以机器人为中心，固定大小的栅格；
- 计算每个栅格对应的极坐标（角度和距离）；
- 比较该模板中每个格子所在角度的距离 $d$ 和scan对应角度的距离：
  - 若 $d$ 比测到的距离远，则不更新；
  - 若 $d$ 比测到的距离近，则涂白；
  - 若 $d$ 等于测到的距离，则涂黑；

该方法不用计算每条线的ray-casting，但需要按指定角度查找距离，且需要计算的栅格数较多，随距离平方增长。

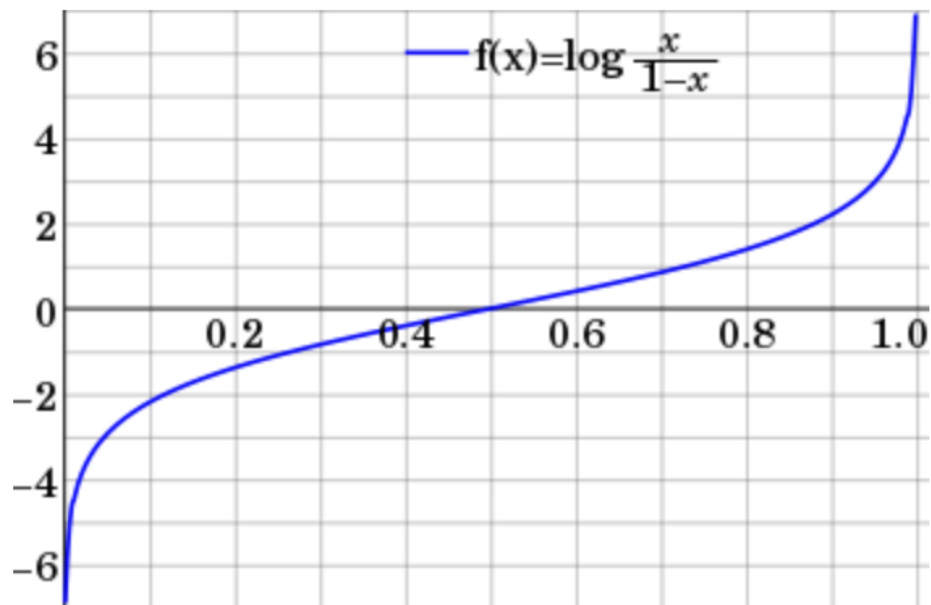




# 占据栅格地图

## □ 栅格的更新

- 如果按0-1浮点数表示栅格占据概率，那么概率更新需要转换到logit处理，工程上会引入额外的对数计算，不划算；
- 可以简单地对栅格进行加减1的操作，改为统计次数，最后对栅格地图取二值化。







# 占据栅格地图

## ▣ 栅格地图程序实现



子地图



# 子地图

□ 子地图是一种更加方便管理扫描数据的方式。

- 大部分SLAM系统需要用关键帧来存储不同区域的扫描数据;
- 多个关键帧可以组成子地图;



- 频率太高
- 数据量大



- 表达单一
- 难以调整

- 激光SLAM的扫描数据相对稳定, 同时单个Scan数据量较少, 可以认为一个子地图内部的数据是不变的 (而VSLAM通常使用关键帧+路标来描述);
- 子地图对应的Scan、栅格地图、似然场可以统一存储。



## 子地图

### □ 子地图坐标系与位姿

每个子地图有独立坐标系 $S$ ，于是每个scan在世界系中的位姿为： $\boldsymbol{T}_{WC} = \boldsymbol{T}_{WS}\boldsymbol{T}_{SC}$ 。

我们允许改变子地图位姿 $\boldsymbol{T}_{WS}$ ，而scan相对子地图位姿 $\boldsymbol{T}_{SC}$ ，估计完之后就不必再改变；

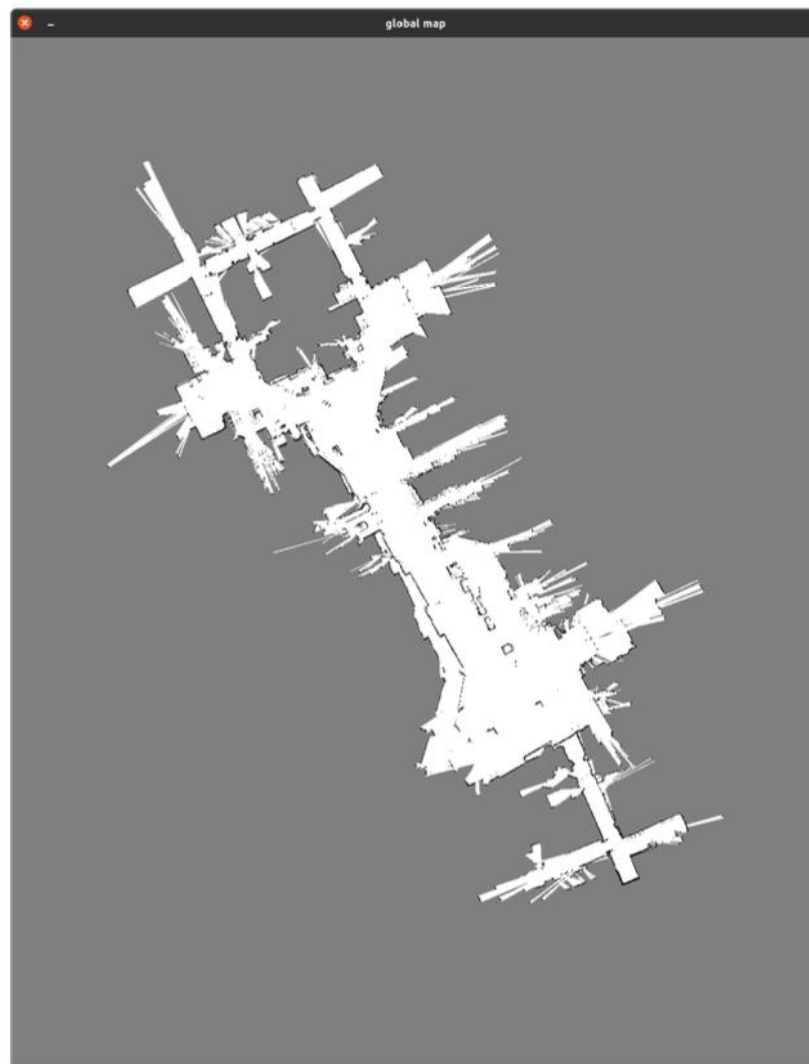
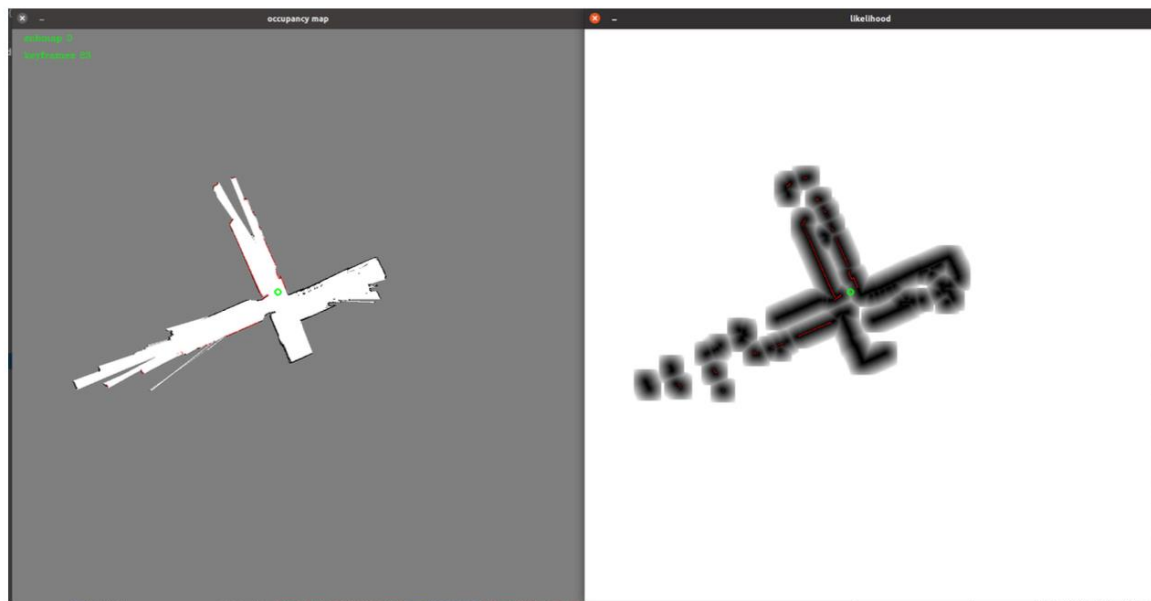
每次Scan Matching计算的是 $\boldsymbol{T}_{SC}$ ，称为scan-to-map。



## 子地图

在SLAM过程中，不断创建新的子地图，新的子地图可以使用一部分旧地图的数据，子地图拼在一起形成全局地图。

- 由于子地图位姿可以调整，我们可以灵活地修正全局地图





# 子地图

## □ 回环检测模块可以基于子地图来设计

- 在关键帧系统中，回环检测也可以基于关键帧来设计；
- 回环模块使用基于子地图位姿的pose graph；
- 首先要解决回环的检测和计算问题。

## □ 回环检测可以看作初始解较差的Scan-to-map

- 核心是如何消除初始解较差的问题

## □ 主要方法

- 网格搜索 (grid search)
- 粒子滤波 (particle filter)
- 分枝定界 (branch and bound)
- 由粗至精/金字塔 (coarse to fine)



# 子地图

## □ 本章实现coarse to fine思路

1. 生成多个分辨率下的似然场图像（更低分辨率下的似然场，吸引范围更远）；
2. 由粗至精进行Scan Matching，上一层的解用于下一层的初始值；
3. 根据匹配的重合度判定是否成功。

问题：

当激光不是360度视野时，很可能存在一部分扫描数据在历史地图中并未出现的情形。

因此，如何分辨匹配的正确性，会影响到回环检测模块的灵敏度。





# 子地图

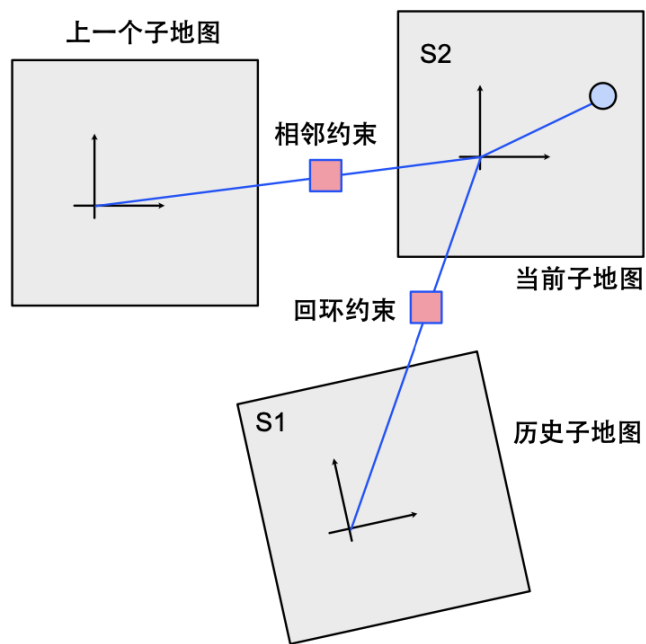
## 回环修正

- 假设回环算法计算了当前Scan ( $C$ 系) 和某个历史子地图 ( $S_1$ ) 之间的变换  $T_{S_1 C}$
- 那么当前子地图 ( $S_2$ ) 和历史子地图之间的关系为:

$$T_{S_1 S_2} = T_{S_1 C} T_{WC}^{-1} T_{WS_2}$$

- 它构成了子地图pose graph中的一个约束

残差:  $e = \text{Log}(T_{WS_1}^{-1} T_{WS_2} T_{S_1 S_2}^{-1}) \in \mathbb{R}^3$ .



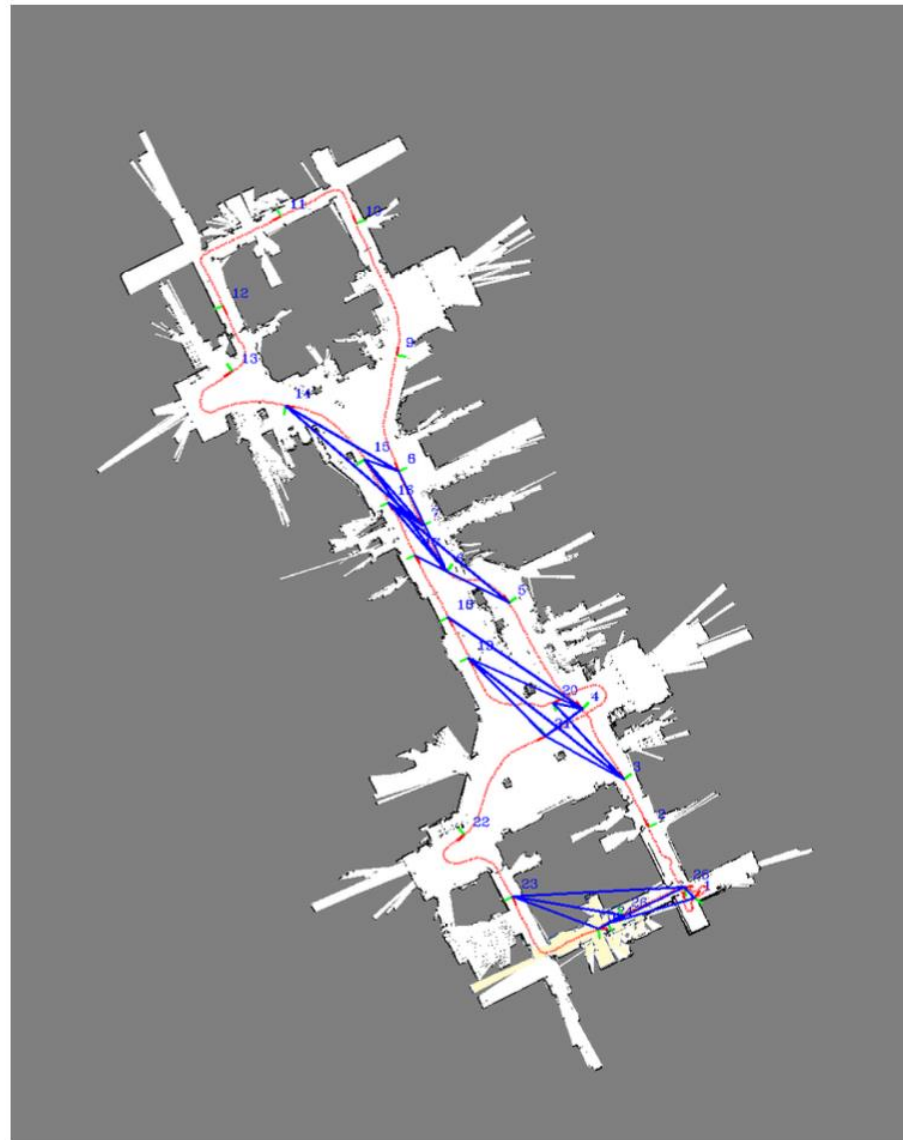




# 子地图

## □ 回环检测修复后的地图

- 红线：轨迹
- 蓝线：子地图间回环
- 黄色区域：当前子地图

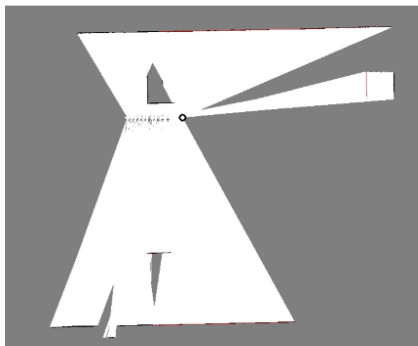
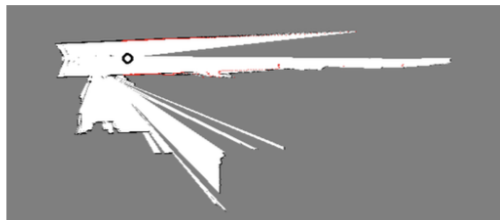




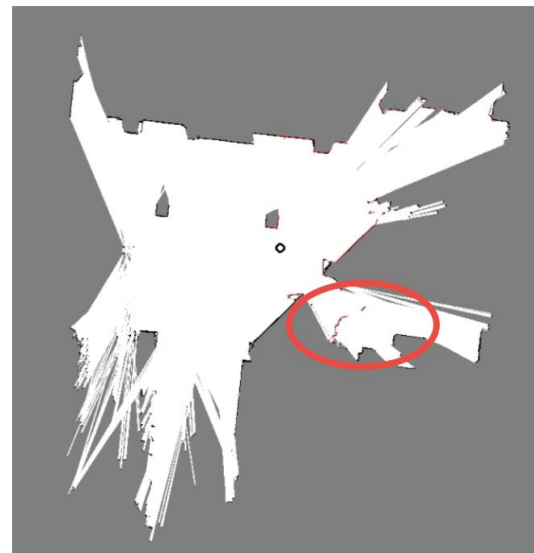
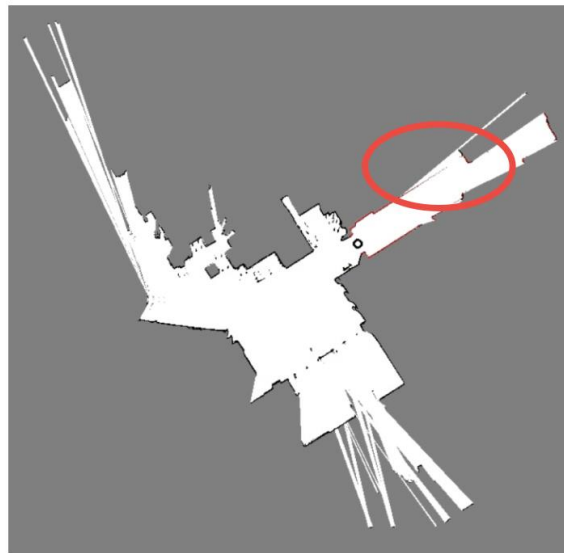
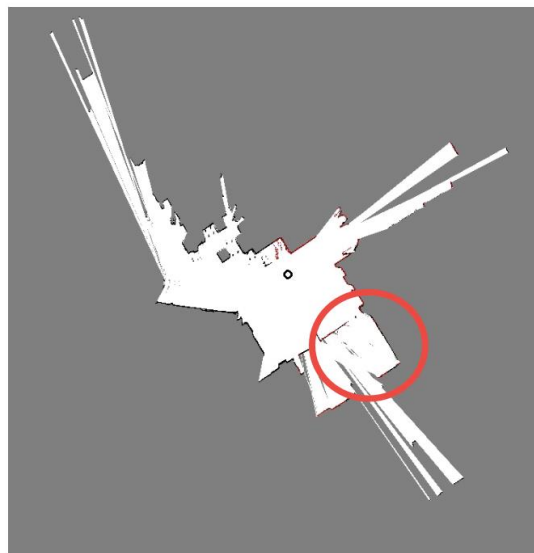
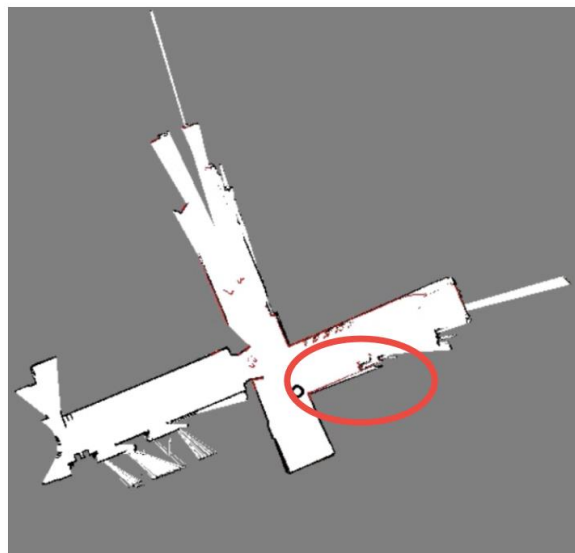
# 子地图

## □ 激光SLAM其他工程问题讨论

- 左上：运动补偿（去畸变）
- 右上：透射现象
- 左下：反光现象
- 右下：机器倾斜



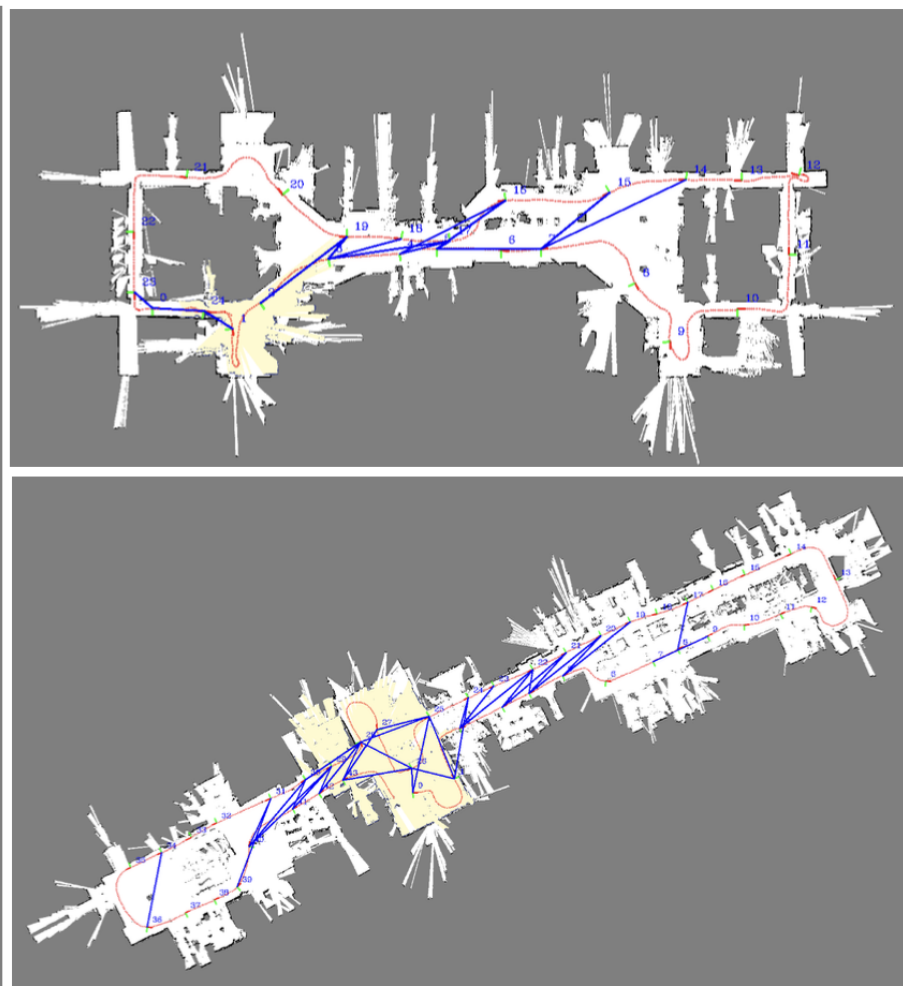
退化场景和空旷区域  
两者可以同时出现





# 子地图

## □ 程序实现和讨论





## 习题

1. 实现基于优化器的点到点ICP、点到线ICP。
2. 实现对似然场图像进行插值后进行scan matching的算法。
3. 基于直线拟合方法，讨论并实现对单个Scan退化检测的算法。
4. 讨论在实际机器人中，如何处理悬空物体和低矮物体（开放问题）。

感谢聆听！  
Thanks for Listening

