

# Mapping and Localization

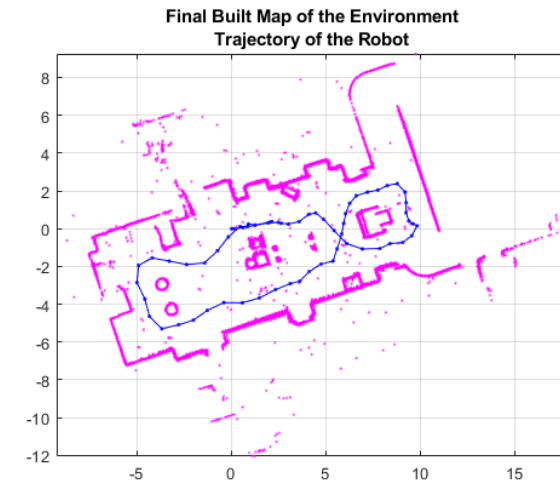
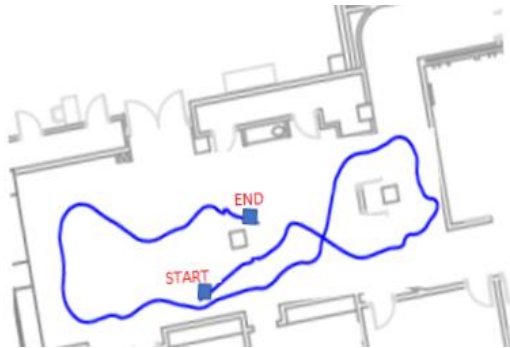
By Liuan  
2018/11/18

# Content

- Intro to SLAM
- Occupancy Grid Maps
- What I do

# SLAM

- simultaneous localization and mapping
- constructing or updating a map of an unknown environment while simultaneously keeping track of an agent's location within it



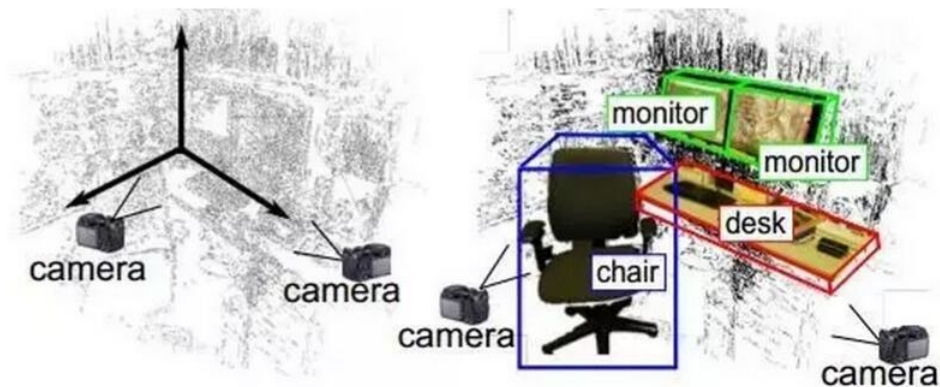
# SLAM

- Core issues
  - What does the world look like?
  - Where I am?
- Methods
  - EKF SLAM(卡尔曼滤波)
  - GraphSLAM(因子图)
  - Occupancy Grid SLAM
  - DP-SLAM



# Map

- Metric Map (尺度地图)
- Topological Map (拓扑地图)
- Semantic Map (语义地图)



# Occupancy Grid Maps

- Occupancy (占据率)

$$p(s = 1) \quad \textit{free}$$

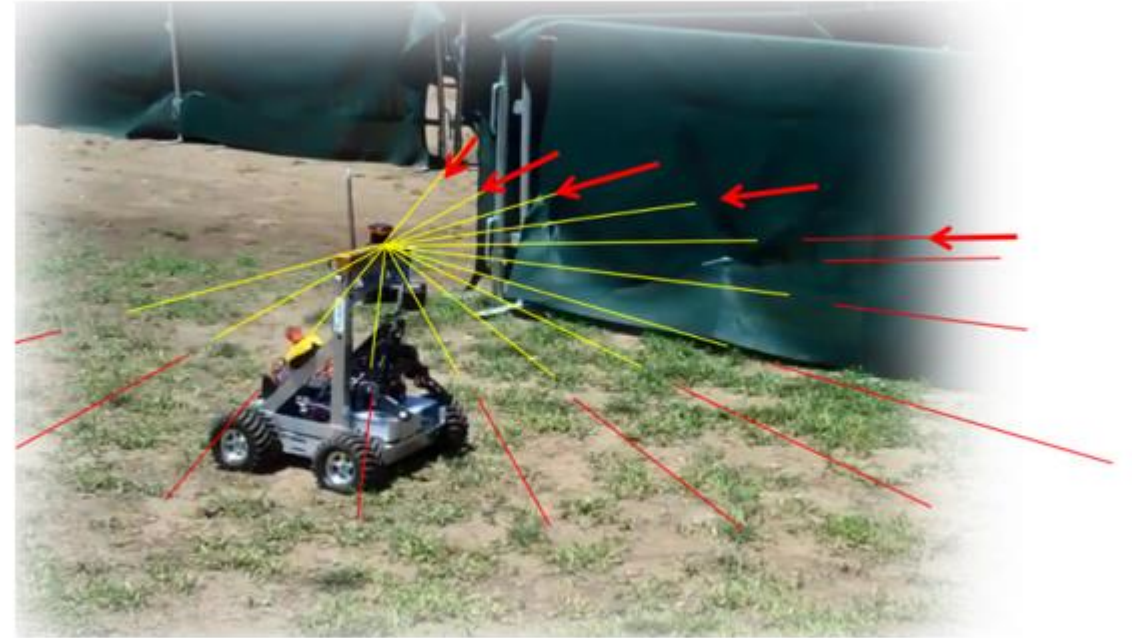
$$p(s = 0) \quad \textit{occupied}$$

$$Odd(s) = \frac{p(s = 1)}{p(s = 0)}$$

$$Odd(s|z) = \frac{p(s = 1|z)}{p(s = 0|z)}$$

$$p(s = 1|z) = \frac{p(z|s = 1)p(s = 1)}{p(z)}$$

$$p(s = 0|z) = \frac{p(z|s = 0)p(s = 0)}{p(z)}$$



# Occupancy Grid Maps

$$\begin{aligned} \text{Odd}(s|z) &= \frac{p(s=1|z)}{p(s=0|z)} \\ &= \frac{p(z|s=1)p(s=1)/p(z)}{p(z|s=0)p(s=0)/p(z)} \\ &= \frac{p(z|s=1)}{p(z|s=0)} \text{Odd}(s) \end{aligned}$$

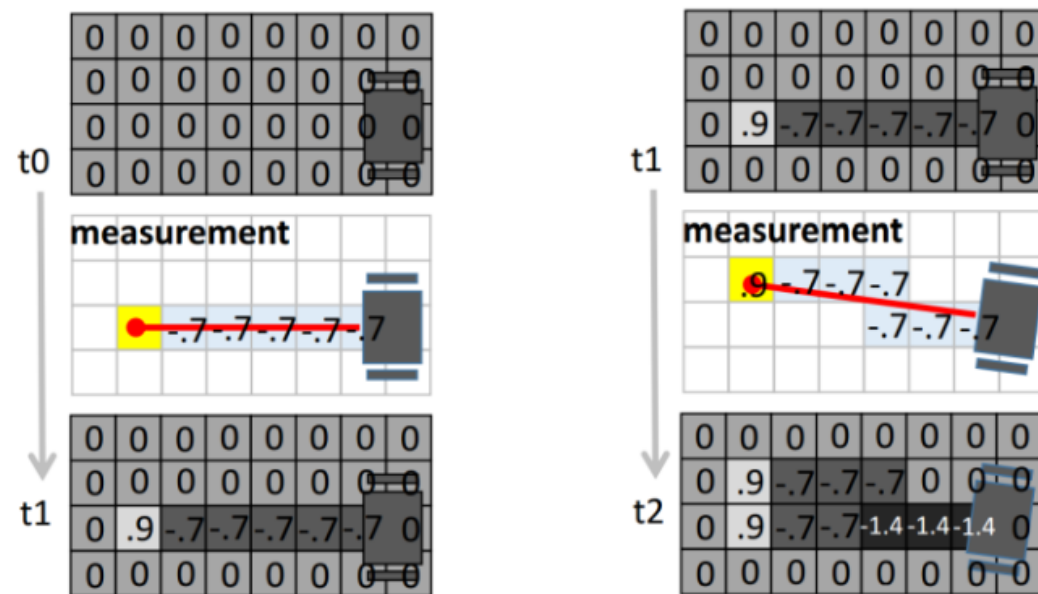
我们对两边取对数得：

$$\log \text{Odd}(s|z) = \log \frac{p(z|s=1)}{p(z|s=0)} + \log \text{Odd}(s)$$

$$\log \text{lofree} = \log \frac{p(z=0|s=1)}{p(z=0|s=0)} \quad \text{和} \quad \log \text{looccu} = \log \frac{p(z=1|s=1)}{p(z=1|s=0)}$$

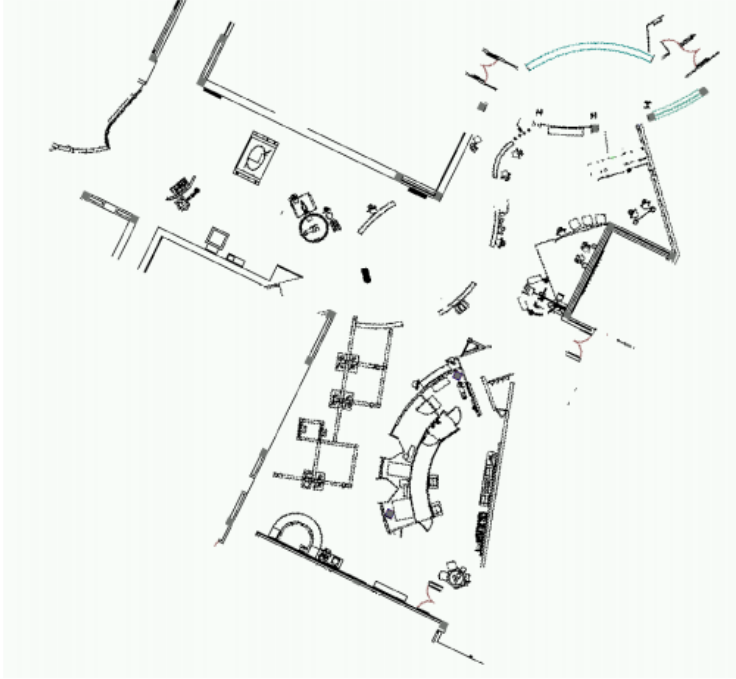
$$S^+ = S^- + \log \text{lofree}$$

例如，假设我们设定  $\log \text{looccu} = 0.9$ ， $\log \text{lofree} = -0.7$ 。那么，一个点状态的数值越大，就表示越肯定它是Occupied状态，相反数值越小，就表示越肯定它是Free状态。

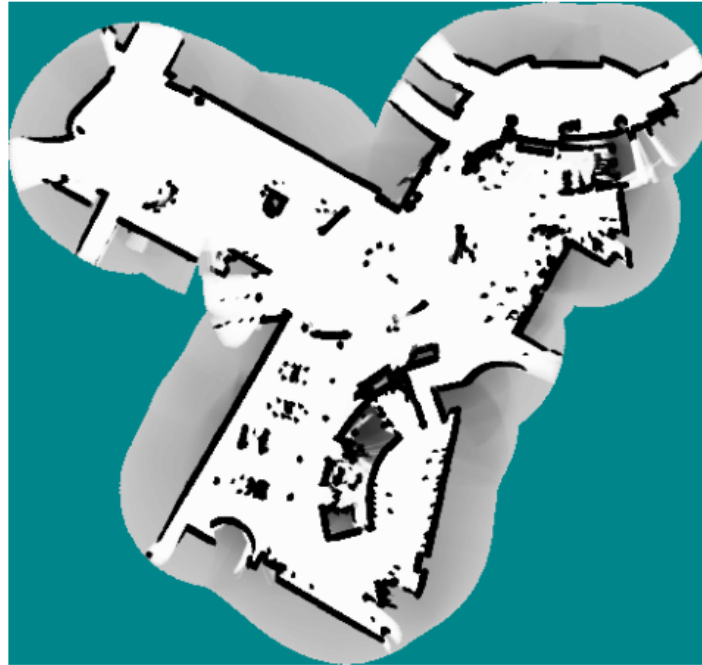


上图就展示了用两个激光传感器的数据更新地图的过程。在结果中，一个点颜色越深表示越肯定它是Free的，颜色越浅表示越肯定它是Occupied的。

# Occupancy Grid Maps



**CAD map**



**occupancy grid map**



# Clean-robot simplified version

- Ultrasonic sensor
- LEGO EV3
- DFS

---

```
function DFS(positionx, positiony)
    scanning(positionx, positiony)
    for  $i = 0 \rightarrow 4$  do
         $next_x \leftarrow x + dir[i][0]$ 
         $next_y \leftarrow y + dir[i][1]$ 
        if map[x, y] is not empty and vis[x, y] is not visit then
            move(nextx, nexty)
            DFS(nextx, nexty)
        end if
        moveback(currentx, currenty)
    end for
```

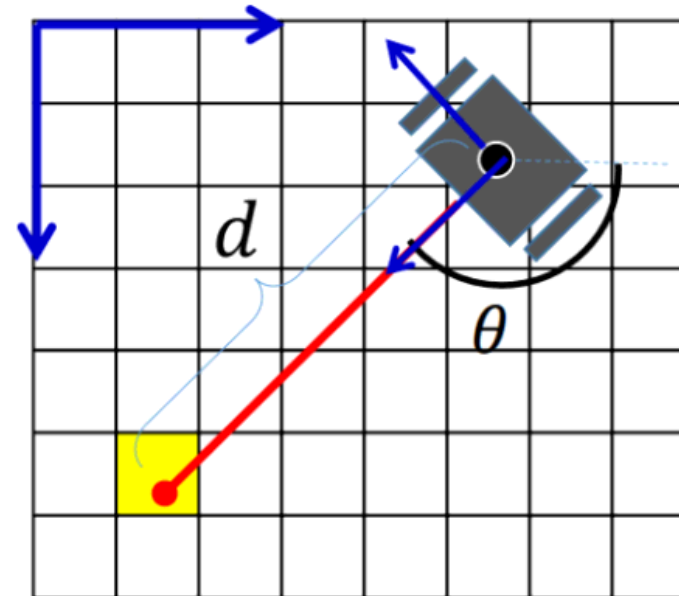
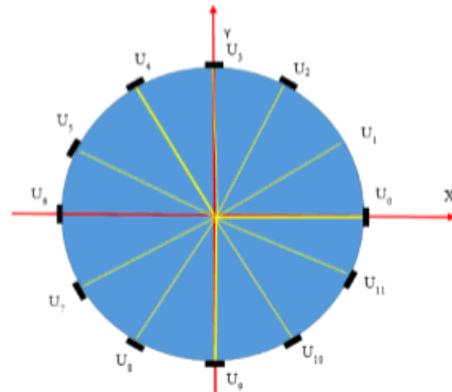
---

# Clean-robot simplified version

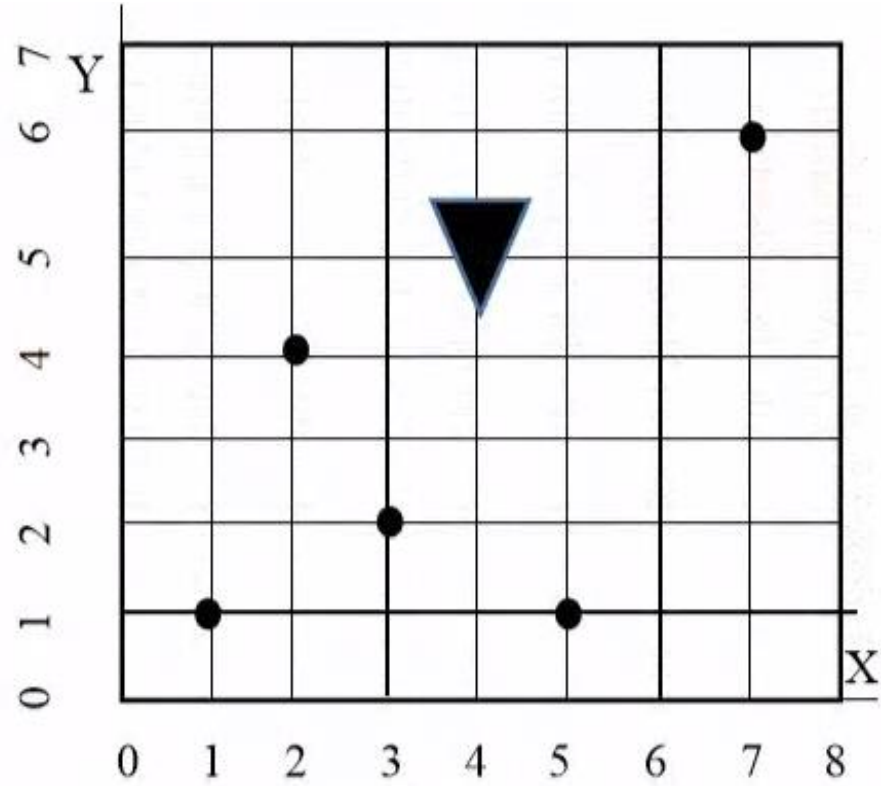
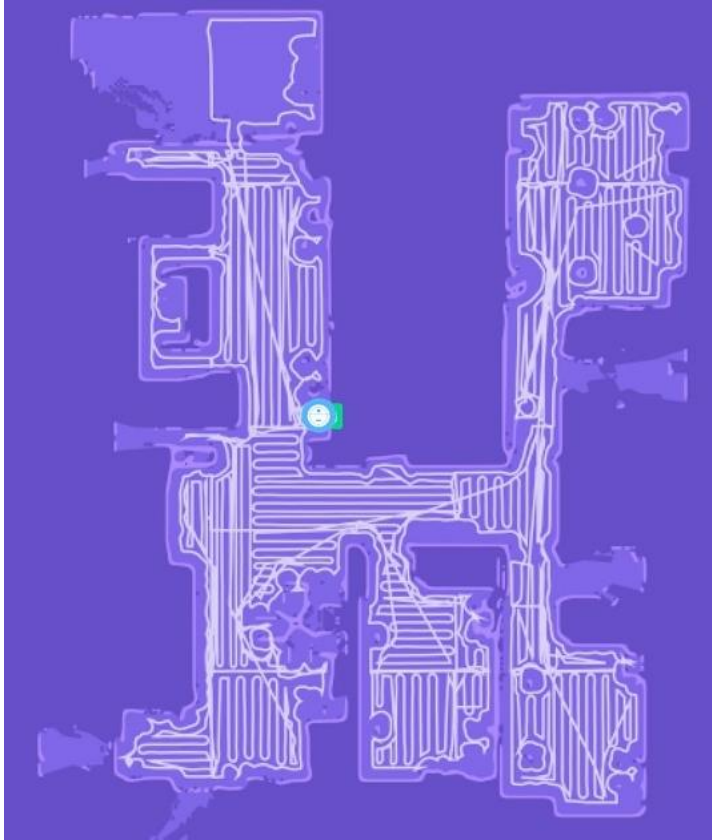
- `move(current_x,current_y,next_x,next_y)`
- `scanning()`
- `move_back()`



Figure 2: The NXT, sonar unit, and sonar testbed



# Result



# Improve

- How to move back
- Record how far have gone
- Self-correcting

# Reference

- [https://en.wikipedia.org/wiki/Simultaneous\\_localization\\_and\\_mapping](https://en.wikipedia.org/wiki/Simultaneous_localization_and_mapping)
- <https://zhuanlan.zhihu.com/p/21738718>
- <https://www.cnblogs.com/cwyblogs/p/8335064.html>
- Sonar sensor virtualization for object detection and localization
- Exploiting MindStorms NXT: Mapping and Localization Projects for the AI Course
- Localization and Mapping in Local Occupancy Grid Maps: Simulation in Ackerman model mobile robot

# Thanks

Q&A