

# AI\_12\_09

## Localization

### #Markov

- 可以任何地方都可以開始
- 當在更新位置的時候就會更新整張地圖
- 可以恢復模擬兩可的地方
- 如果格子太多，記憶體跟運算能力需要比較好

### #Kalman

- 會跟蹤機器人
- 不確定性會越來越高，ex. 與物體碰撞
- 會丟失位置，除非到了到過的點

## Markov

- 對狀態空間所有位置的機率使用明確的離散表示
- 通過有限數量的可能位置的拓樸來表示環境
- 每次更新，更新整個空間的每個狀態 ( 元素 ) 的機率

會因為環境而改變自己位置的機率

$P(A)$

$r$  robot 機器人

$l$  position 位置

$t$  time 時間

$p(r_t = l)$  在  $t$  時間在  $l$ 位置的機率

$i_t$  the sensor input 感測器

$p(r_t = l, i_t)$  再有傳感器  $i_t$  的情況下，在  $l$  位置的機率

$$p(A|B) = p(B|A)p(A)/p(B) \rightarrow p(l|i) = p(i|l)p(l)/p(i)$$

Map from a belief state and a sensor input to a refined belief state

$p(l)$  更新之前的 belief state

$p(i|l)$  在位置  $l$  時，獲得  $i$  的機率

$o_t$  action 動作、走一步

$$p(l_t|o_t) = \int p(l_t|l'_{t-1}, o_t)p(l'_{t-1})dl'_{t-1}$$

- 從 belief state 經過一個動作後到 new belief state
- 總結機器人可能會到達  $l$  的所有可能機率

## Kalman filter Localization

只在意自己在地圖的地方

- 必須要有資訊進來
- 會有增加雜訊
- 動態模組的不確定性增加真實度