Introducción a la Robótica Móvil

Primer cuatrimestre de 2018

Departamento de Computación - FCEyN - UBA

Mapeo del entorno (*mapping*) - clase 17

Mapeo utilizando grillas de ocupación (*occupancy grids*)

¿Por qué construir mapas del entorno?

- No siempre tenemos una representación del espacio de trabajo del robot.
- Construir mapas del ambiente (mapeo) es uno de los problemas fundamentales de la robótica.
- Sin un mapa es imposible que el robot pueda llevar adelante su tarea, no puede localizarse, no puede planificar trayectorias seguras.
- Contra más preciso sea el mapa, mejor va a ser el desempeño del robot.
- Mapear y localizarse es como resolver el problema del huevo y la gallina. El mapeo involucra simultáneamente estimar la pose del robot en el mapa.
- Por eso generalmente se deben abordar ambos problemas (mapear y localizarse) al mismo tiempo, lo que se conoce SLAM (*Simultaneous Localization and Mapping*).
- Para comenzar a tratar el problema del mapeo vamos a considerar que la pose del robot es conocida.

¿Qué problemas tenemos que enfrentar?

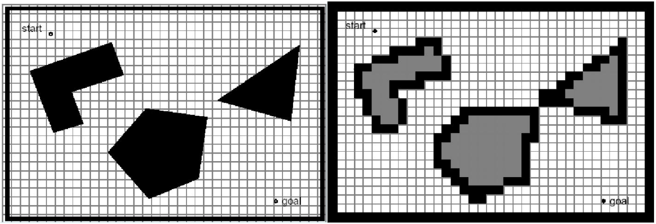
- La interpretación (o modelo) del sensor
  - ¿Cómo extraemos información relevante de los datos crudos que nos arroja el sensor.
  - ¿Cómo representamos e integramos esta información a lo largo del tiempo?
- Las ubicaciones del robot tienen que ser estimadas (o conocidas)
  - ¿Cómo podemos identificar que el robot se encuentra en un lugar que ya visitó? (*loop closure*)
  - ¿Cómo asociamos los datos para poder saber esto? (*data association*)

Representación y modelo del entorno

- Representación del entorno
  - Métrica y continua $\rightarrow x, y, \phi$
  - Métrica y discreta $\rightarrow$ grilla métrica
  - Topológica y discreta $\rightarrow$ grilla topológica
- Modelo (mapa) del entorno
  - Datos crudos del sensor: usamos toda la información capturada, grandes volúmenes de datos, poca distintividad
  - Características de bajo nivel: por ejemplo características geométricas simples, volúmen de datos medio, media distintividad, se puede filtrar la información inutil, pero continúan ambigüedades
  - Características de alto nivel: por ejemplo características visuales, reconocimiento de objetos, poco volúmen de datos, alta distintividad.

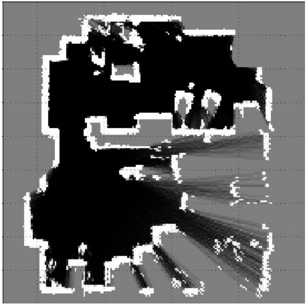
Tenemos que elegir el tipo apropiado de mapa (modelo) de acuerdo con la tarea que queremos resolver.

Descomposición en forma de una grilla de celdas de tamaño fijo.



Grillas de ocupación (*occupancy grids*)

- Introducido por Moravec y Elfes en 1985.
- Dadas las limitaciones intrínsecas de cualquier sensor, es importante componer un modelo del mundo coherente usando la información capturada en múltiples sensados.
- Se representa el entorno con una una grilla de celdas de tamaño fijo.
- Se estima la probabilidad de que cada celda esté ocupada por un obstáculo.
- Se asumen dos hipótesis muy importantes:



1. La probabilidad (o *belief*) de ocupación de cada celda ($m[x, y]$) es independiente:

   $$Bel(m_t) = p(m_t|u_1, z_1, \ldots, u_{t-1}, z_t) = \prod_{x,y} Bel(m_t^{[x,y]})$$

   Por lo tanto podemos calcularla de forma independiente para cada celda.

2. ¡La pose del robot es conocida!

Repaso: Regla de Bayes

Cuando el robot sensa no hace otra cosa que aplicar el teorema de Bayes:

$$P(x|z) = \frac{P(z|x)P(x)}{P(z)}$$

$P(x|z)$: probabilidad a posteriori (*posterior belief*) dado que sensé z
$P(z|x)$: probabilidad de medición dado que estoy en x
$P(x)$: probabilidad a priori de estar en x
$P(z)$: probabilidad de sensar z independientemente de donde esté.
Para hallar $P(z)$ usamos el teorema de la probabilidad total:

$$P(z) = \sum_x P(z|x)P(x)$$

Entonces podemos reescribir la regla de Bayes como:

$$P(x|z) = \frac{P(z|x)P(x)}{P(z)} = \eta P(z|x)P(x)$$

$$\text{donde } \eta = P(z)^{-1} = \frac{1}{\sum_x P(z|x)P(x)}$$

Nota: η es el término de normalización que usamos para que el posterior belief sea una probabilidad bien definida.

Repaso: Filtro Bayesiano

Dado un conjunto de obstervaciones y acciones de control del robot para moverse:

$$d_t = \{u_1, z_1, \ldots, u_t, z_t\}$$

- El modelo de sensado es $p(z_t|x_t)$
- El modelo de movimiento es $p(x_t|u_t, x_{t-1})$
- La probabilidad a priori del estado del sistema es $p(x_t)$

Lo que queremos es estimar el estado de nuestro sistema a posterior de la acción y del sensado, i.e., la probabilidad condicional a posterior es el *posterior belief*

$$Bel(x_t) = p(x_t|u_1, z_1 \ldots, u_t, z_t)$$

Repaso: Filtro Bayesiano

$$
\begin{aligned}
Bel(x_t) &= p(x_t|u_1, z_1 \ldots, u_t, z_t) = p(x_t|z_{1:t}, u_{1:t}) \\
\text{(por Bayes)} &= \eta p(z_t|x_t, z_{1:t-1}, u_{1:t}) p(x_t|z_{1:t-1}, u_{1:t}) \\
\text{(por Markov)} &= \eta p(z_t|x_t) p(x_t|z_{1:t-1}, u_{1:t}) \\
\text{(por Prob. Total)} &= \eta p(z_t|x_t) \int p(x_t|z_{1:t-1}, u_{1:t}, x_{t-1}) p(x_{t-1}|z_{1:t-1}, u_{1:t}) dx_{t-1} \\
\text{(por Markov)} &= \eta p(z_t|x_t) \int p(x_t|u_t, x_{t-1}) p(x_{t-1}|z_{1:t-1}, u_{1:t}) dx_{t-1} \\
\text{(por Markov)} &= \eta p(z_t|x_t) \int p(x_t|u_t, x_{t-1}) p(x_{t-1}|z_{1:t-1}, u_{1:t-1}) dx_{t-1} \\
&= \eta p(z_t|x_t) \int p(x_t|u_t, x_{t-1}) Bel(x_{t-1}) dx_{t-1}
\end{aligned}
$$

Grillas de ocupación: Actualización

Idea: actualizar cada celda usando el Filtro de Bayes. La variable aleatoria x va a ser cada posición del mapa $m^{[x,y]}$:

$$Bel(m_t^{[x,y]}) = \eta p(z_t|m_t^{[x,y]}) \int p(m_t^{[x,y]}|u_t, m_{t-1}^{[x,y]}) Bel(m_{t-1}^{[x,y]}) dm_{t-1}^{[x,y]}$$

Hipótesis adicional: el mapa es estático. Entonces nos queda:

$$Bel(m_t^{[x,y]}) = \eta p(z_t|m_t^{[x,y]}) Bel(m_{t-1}^{[x,y]})$$

También nos queda recordando la definición primera:

$$Bel(m_t^{[x,y]}) = p(m_t^{[x,y]}|z_{1:t}, u_{1:t}) = p(m_t^{[x,y]}|z_{1:t})$$

Grillas de ocupación: Actualización

Vamos a discretizar el mapa en celdas de tamaño fijo:

- La proposición $oc(i, j)$ significa que la celda $C_{i,j}$ está ocupada
- Probabilidad: $p(oc(i, j))$ tiene rango $[0, 1]$.
- Odds: $o(oc(i, j))$ tiene rango $[0, +\infty)$

$$o(A) = \frac{p(A)}{p(\neg A)}$$

- Logaritmo de odds: *log* $o(oc(i, j))$ tiene rango $(-\infty, +\infty)$
- Cada celda $C_{i,j}$ va a mantener el valor de *log* $o(oc(i, j))$ en lugar de mantener directamente el valor $oc(i, j)$.

Grillas de ocupación probabilísticas

- Aplicamos la regla de Bayes:

$$P(x|z) = \frac{p(z|x)p(x)}{p(z)}$$

- donde x es $oc(i,j)$ y z es una observación $r = D$.
- Sabemos que $Bel(m_t^{[x,y]}) = p(m_t^{[x,y]}|z_{1:t}, u_{1:t}) = p(m_t^{[x,y]}|z_{1:t})$,
- Si llamamos $\eta = P(z)^{-1}$, $p(z_t|m_t^{[x,y]}) = P(z|x)$ y $Bel(m_{t-1}^{[x,y]}) = P(x)$ nos queda:

$$Bel(m_t^{[x,y]}) = \eta p(z_t|m_t^{[x,y]}) Bel(m_{t-1}^{[x,y]})$$

- Podemos simplificar esto usando la representación de *log odds*.

Grillas de ocupación: deducción

- Regla de Bayes:

$$p(x|z) = \frac{p(z|x)p(x)}{p(z)}$$

- De igual forma:

$$p(\neg x|z) = \frac{p(z|\neg x)p(\neg x)}{p(z)}$$

- Luego:

$$o(x|z) = \frac{p(x|z)}{p(\neg x|z)} = \frac{p(z|x)p(x)}{p(z|\neg x)p(\neg x)} = \lambda(z|x)o(x)$$

- donde:

$$o(x|z) = \frac{p(x|z)}{p(\neg x|z)}$$

$$\lambda(z|x) = \frac{p(z|x)}{p(z|\neg x)}$$

Grillas de ocupación: Actualización usando Bayes

- La Regla de Bayes puede ser reescrita:

$$o(x|z) = \lambda(z|x)o(x)$$

- Tomando *log odds* hacemos que las multiplicaciones sean sumas:

$$log\ o(x|z) = log\ \lambda(x|z) + \log\ o(x)$$

- De esta forma tenemos una forma sencilla de actualizar el contenido de la celda.

Grillas de ocupación: Actualización de cada celda

- Cada celda $C_{i,j}$ mantiene $log\ o(oc(i,j))$
- La evidencia $r = D$ significa que el sensado r retornó D
- Para cada celda $C_{i,j}$ se acumula la evidencia de cada lectura del sensor:

$$\begin{aligned} log\ o(x|z) =& log\ \lambda(z|x) + log\ o(x) \\ log\ o(oc(i,j)|r = D) =& log\ \lambda(r = D|oc(i,j)) + log\ o(oc(i,j)) \end{aligned}$$

- Esta última ecuación es la regla de actualización de cada celda $C_{i,j}$
- Falta dar el modelo de sensado para poder calcular $\lambda(r = D|oc(i,j))$.

Modelo de sensado

El modelo de sensado depende de cada sensor y es un modelo (aproximación) que se propone que debe ajustarse lo más fielmente al comportamiento que tiene el sensor.

Modelo de sensado para un Láser 2D

La función de probabilidad $p(z_t|m_t^{[xy]})$ se define como:

$$p(z_t|m_t^{[xy]}) = \frac{1 + model_o^{z_t}(\alpha, d) - model_l^{z_t}(\alpha, d)}{2}$$

Donde:

- (α, d) son las coordenadas polares de la celda $m_t^{[xy]}$ respecto al marco de referencia del Láser.
- z_t es la distancia medida.

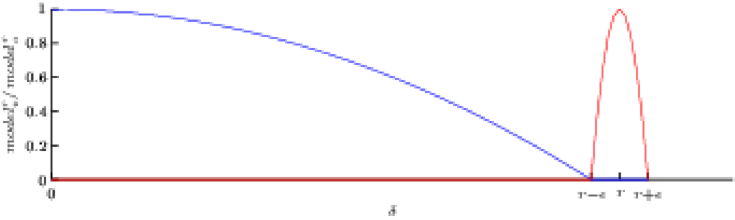
Modelo de sensado para un Láser 2D

El modelo filtra mediciones más allá de un X:

- $model_o^r(\delta) = \begin{cases} 1 - (\frac{\delta - r}{\epsilon})^2, & r < X \wedge \delta \in [r-\epsilon, r+\epsilon] \\ 0, & \text{en otro caso} \end{cases}$
- $model_l^r(\delta) = \begin{cases} 1 - (\frac{\delta}{r - \epsilon})^2, & \delta \in [0, r-\epsilon] \\ 0, & \text{en otro caso} \end{cases}$

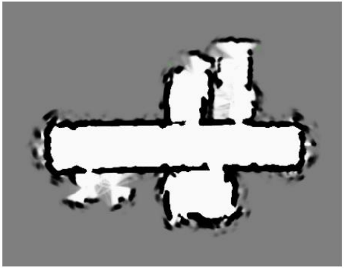
Donde:

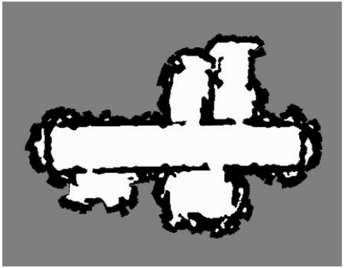
- δ es la distancia a considerar por sobre el vector determinado por el haz del láser.



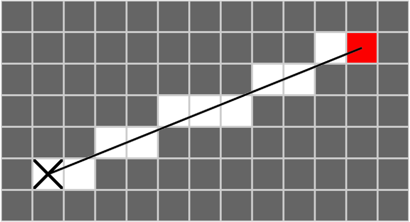
¿Cómo obtenemos el mapa de ocupación final?

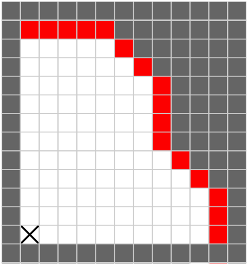
Entonces el mapa se obtiene considerando que están ocupadas todas las celdas que tienen valor $C_{i,j} > 0{,}5$, que están libres aquellas que tienen valor $C_{i,j} < 0{,}5$ y las celdas que tienen valor $C_{i,j} = 0{,}5$ son aquellas que sobre las que tenemos información todavía.





Enfoque práctico



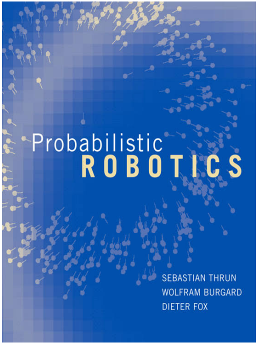


- Conectar la celda origen del sensor con la celda correspondiente a donde impactó el láser.
- Definir todas las celdas sobre la línea como desocupadas.
- Definir la celda impactada como ocupada.
- Aplicar la regla de Bayes para actualizar la grilla.
- Utilizar algún algoritmo de gráficado de rectas (Bresenham).
- Mejora: Utilizar algún algoritmo de relleno por difusión (flood fill) para graficar el escaneo láser completo.

Grillas de ocupación: resumen

- Las grillas de ocupación son un enfoque muy utilizado para representar el entorno de un robot móvil cuando la pose del robot en cada instante es conocida.
- En este enfoque cada celda se considera independientemente del resto.
- Cada celda almacena la probabilidad posterior de que la correspondiente área en el entorno esté ocupada por un obstáculo.
- Las grillas de ocupación pueden ser eficientemente construidas usando un enfoque probabilístico.
- Usando la regla de Bayes y un modelo del sensor podemos obtener una regla de actualización sencilla del contenido de cada celda.
- Para obtener el mapa de mayor verosimilitud utilizamos un umbral (por ejemplo, 0.5)

Más sobre Mapping



"Probabilistics Robotics", Sebastian Thrun, Wolfram Burgard, Dieter Fox. MIT press, 2006. **Capítulo 3**