Robots Móviles

Tema 3, parte 4.
Mapping con poses conocidas

Otto Colomina Pardo
Sergio Orts Escolano

¿Por qué es necesario un mapa?

- Crear un mapa es uno de los problemas fundamentales de la robótica móvil
- Los mapas permiten a los robots planificar trayectorias, actividades, permiten la localización ...
- No siempre es posible/práctico proporcionar al robot un mapa ya construido

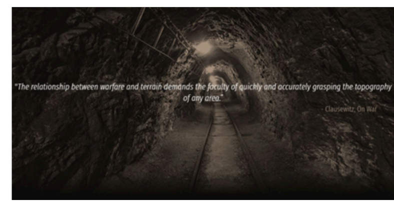
Mapeado automático en Roomba

SUBTERRANEAN (SUBT) CHALLENGE URBAN CIRCUIT

FEBRUARY 2020

DARPA's Subterranean (SubT) Challenge seeks to better equip warfighters and first responders to explore uncharted underground environments that are too dangerous, dark, or deep to risk human lives. In three circuit events and one final event, participating teams will deploy autonomous systems to attempt to map, navigate, and search various underground environments. Teams earn points by correctly identifying artifacts placed within those environments. The Urban Circuit, the second scored event of the SubT Challenge, will take place in February 2020.

More



DARPA subterranean challenge

Mapping

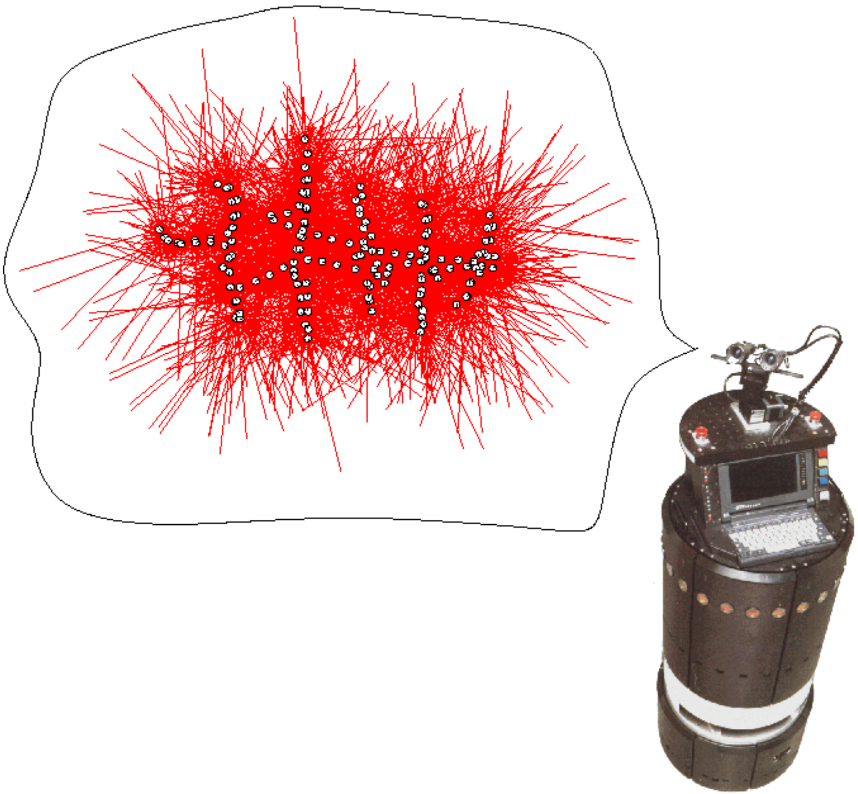
- Formalmente, el mapeo implica, dada la odometría del robot y las lecturas del sensor,

$$d = \{u_1, z_1, u_2, z_2, \ldots, u_n, z_n\}$$

calcular el mapa más probable

$$m^* = \arg\max_m P(m \mid d)$$

El problema del mapeado



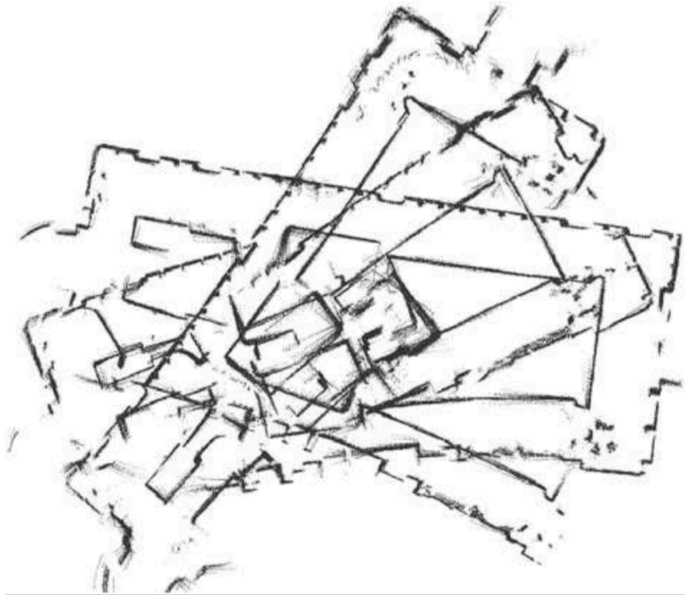
“Desafíos” del mapeado

- El **espacio de hipótesis** (todos los posibles mapas) es extremadamente grande, mucho mayor que en localización
  - No podemos aplicar el filtro de Bayes directamente como hacíamos en localización

- En realidad, el mapeado implica estimar **simultáneamente** la posición del robot y el mapa (Simultaneous Localization and Mapping – SLAM)

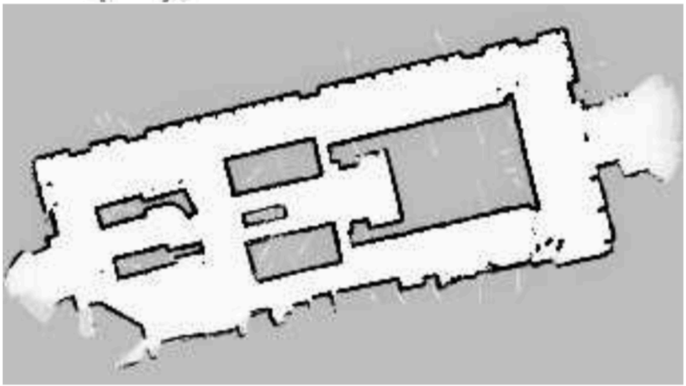
Cuando intentas hacer un mapa con la localización que te da la odometría

(a)



Cuando el mapa se hace conociendo la localización precisa

(b)



Recordemos: tipos de mapas

- Representación espacial
  - Discretización del entorno
  - No representa objetos, si no el entorno
- Representación geométrica
  - Líneas, puntos, polígonos, etc
  - Consumo de memoria bajo
- Representación topológica
  - Parecido a la representación humana
  - Indica lugares y conectividad entre los nodos utilizando una aproximación basado en grafos

Restricción: **poses conocidas**

- Podemos reducirlo a conocidas las posiciones del robot y las lecturas del sensor

$$d = \{x_1, z_1, x_2, z_2, \ldots, x_t, z_t\}$$

- Podemos calcular el mapa más probable

$$m^\star = \mathrm{argmax}_m P(m \mid d)$$

No es una suposición realista pero se puede usar para actualizar los mapas en algoritmos de SLAM

poses map observations & movements

$$p(x_{1:t}, m \mid z_{1:t}, u_{0:t-1}) = p(x_{1:t} \mid z_{1:t}, u_{0:t-1}) \cdot p(m \mid x_{1:t}, z_{1:t})$$

SLAM posterior

Robot path posterior

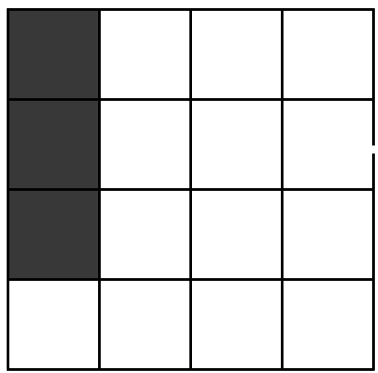
Mapping with known poses

Aquí usaremos **representación espacial**

- Basado en **rejilla de ocupación**
- Tendremos que decidir el **nivel de discretización** (el tamaño de la celdilla en metros)

Representación

- Un mapa m es un conjunto de celdas $\{m_i\}$
- Cada celda es una variable binaria aleatoria que puede valer 0 ó 1
- Para cada celda queremos estimar la prob. de que esté ocupada, $p(m_i = 1)$ o para acortar lo denotaremos con $p(m_i)$



Cada celda o está completamente ocupada o completamente desocupada

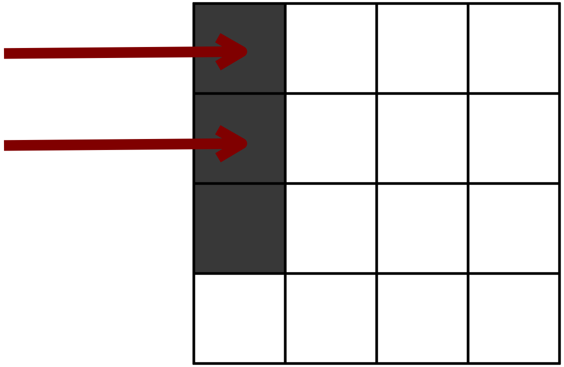
Valores del mapa: $\mathrm{p}(m_i)$

- Valores entre 0 y 1
- Ejemplos
  - 1: Certeza celda ocupada
  - 0: Certeza celda no ocupada
  - 0.5: Desconocido

Simplificación 1

- Supondremos que las celdas son independientes entre ellas
  - Esto no es cierto en el mundo real

No hay dependencias



Consecuencia 1

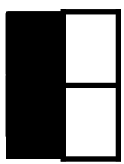
- La distribución de probabilidad del mapa está dada por el producto de las distribuciones de probabilidad de las celdas individuales

$$\underbrace{p(m)}_{\text{mapa}} = \prod_i \underbrace{p(m_i)}_{\text{celda}}$$

Consecuencia 1

- La distribución de probabilidad del mapa está dada por el producto de las distribuciones de probabilidad de las celdas individuales

$$p(m) = \prod_i p(m_i)$$

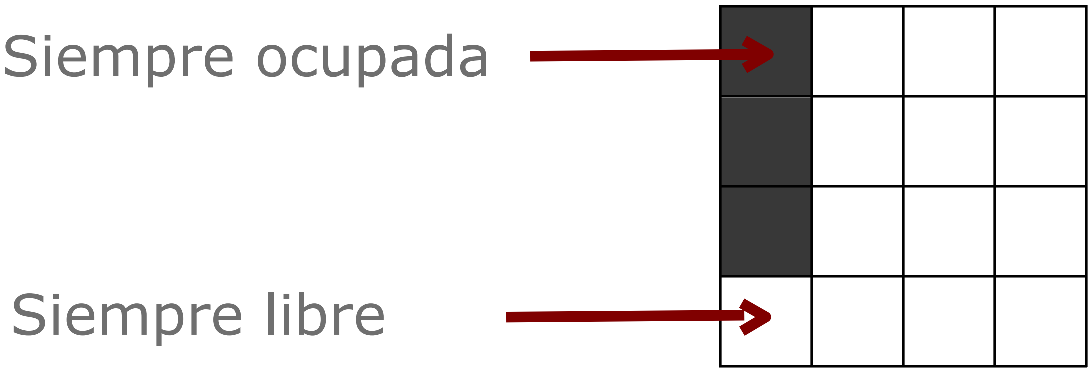


Problema n dimensional

n problemas unidimensionales

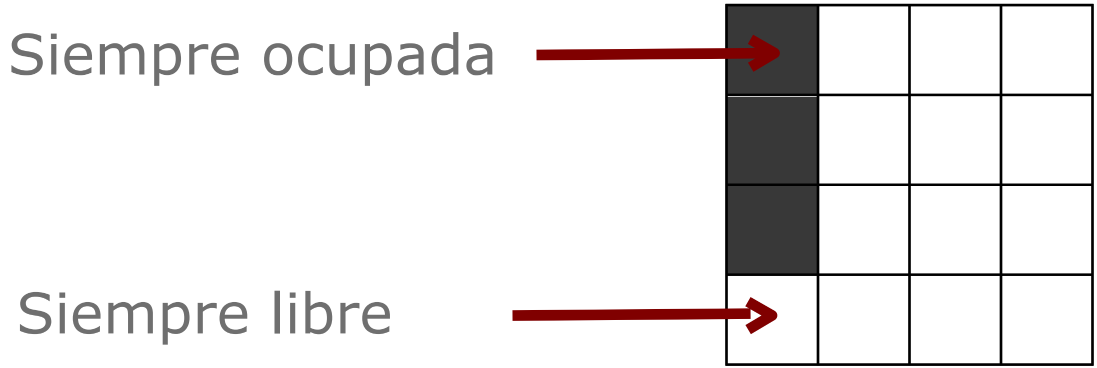
Simplificación 2

- El mundo es estático



Consecuencia 2

- Si aplicamos un filtro de Bayes para estimar las m_i, **no necesitamos fase de predicción**



Estimación del mapa a partir de los datos

Dada la observación $z_{1:t}$ y las poses del sensor, $x_{1:t}$

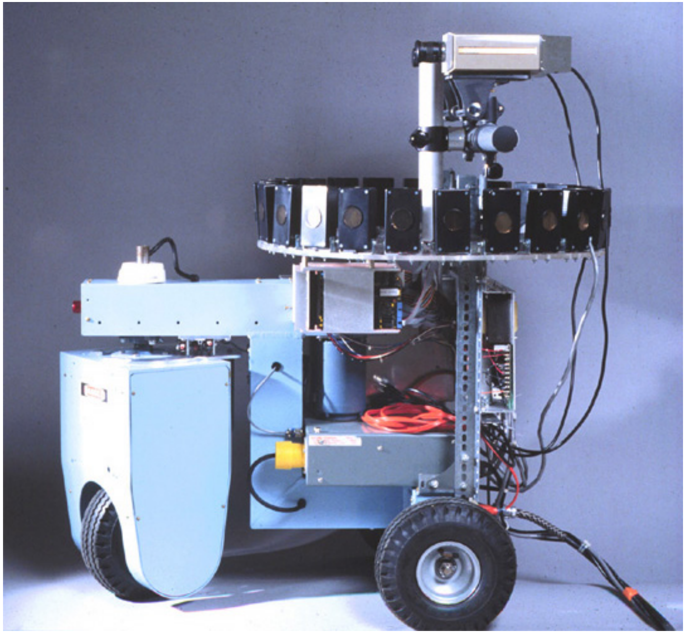
estimamos el mapa

$$p(m \mid z_{1:t}, x_{1:t}) = \prod_i p(m_i \mid z_{1:t}, x_{1:t})$$

Variable aleatoria binaria

Algoritmo: Filtro Bayesiano Binario con estado estático

El algoritmo que vamos a ver es original de Moravec y Elfes, del año 1985:“High Resolution Maps from Wide Angle Sonar“



El robot móvil Neptune, con el que se probaron inicialmente estos métodos

Derivación del filtro bayesiano

No haremos la derivación paso a paso, baste saber que esta fórmula final se obtiene a partir de $p(m^i|z_{1:t}, x_{1:t})$ aplicando el teorema de Bayes, la propiedad de markov y alguna manipulación algebraica…

Más detalles en Thrun, Burgard, Fox. Probabilistic Robotics. MIT Press, 2005. Cap. 4. pp.95,96

Estimación en el instante t de la probabilidad de que la celda m^i esté ocupada

Recordemos que $p(z|x,m)$ es el modelo del sensor. Llamaremos a $p(m|z,x)$ el "modelo inverso del sensor"

$$Bel(m_t^i) = \left[1 + \frac{1 - p(m_t^i \mid z_t, x_t)}{p(m_t^i \mid z_t, x_t)} \, \frac{p(m_t^i)}{1 - p(m_t^i)} \, \frac{1 - Bel(m_{t-1}^i)}{Bel(m_{t-1}^i)}\right]^{-1}$$

Valor que solo depende de la última medida (z_t)

Valor a priori

Término recursivo (valor obtenido en el paso anterior)

Odds

$$Bel(m_t^i) =$$

$$\left[1 + \frac{1 - p(m_t^i \mid z_t, x_t)}{p(m_t^i \mid z_t, x_t)} \; \frac{p(m_t^i)}{1 - p(m_t^i)} \; \frac{1 - Bel(m_{t-1}^i)}{Bel(m_{t-1}^i)}\right]^{-1}$$

- Fijaos en que en todos los términos aparece una probabilidad partida por la probabilidad del "suceso contrario". Matemáticamente esto se llama "**odds**", y varía entre 0 e ∞
- Es fácil pasar de p(x) a odds(x) y viceversa

Log Odds

En lugar de trabajar con los *odds* directamente tomaremos logaritmos.

$$l(x) \quad = \quad \log \frac{p(x)}{1 - p(x)}$$

¿Por qué?: el logaritmo de un producto es la suma de logaritmos. Nos evitamos multiplicar probabilidades muy pequeñas, que daría lugar a errores de redondeo

Cálculo ocupación usando log probabilidades

- El producto se convierte en una suma

$$
\begin{aligned}
& l(m_i \mid z_{1:t}, x_{1:t}) \\
= \quad & \underbrace{l(m_i \mid z_t, x_t)}_{\text{inverse sensor model}} + \underbrace{l(m_i \mid z_{1:t-1}, x_{1:t-1})}_{\text{recursive term}} - \underbrace{l(m_i)}_{\text{prior}}
\end{aligned}
$$

- Notación simplificada

$$
l_{t,i} \quad = \quad \text{inv\_sensor\_model}(m_i, x_t, z_t) + l_{t-1,i} - l_0
$$

Algoritmo mapping

occupancy_grid_mapping($\{l_{t-1,i}\}, x_t, z_t$)**:**

for all cells m_i *do*

    if m_i *in perceptual field of* z_t *then*

        $l_{t,i} = l_{t-1,i} + \text{inv\_sensor\_model}(m_i, x_t, z_t) - l_0$

    else

        $l_{t,i} = l_{t-1,i}$

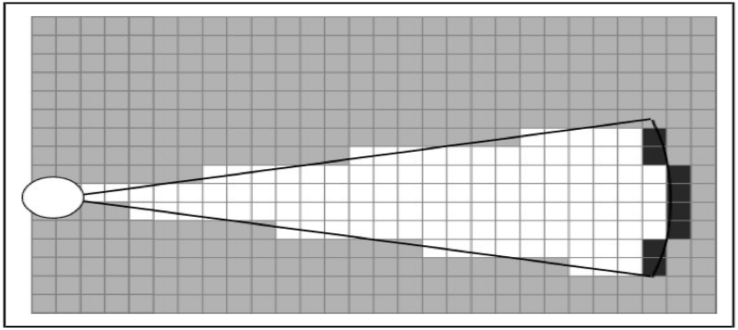
    endif

endfor

return $\{l_{t,i}\}$

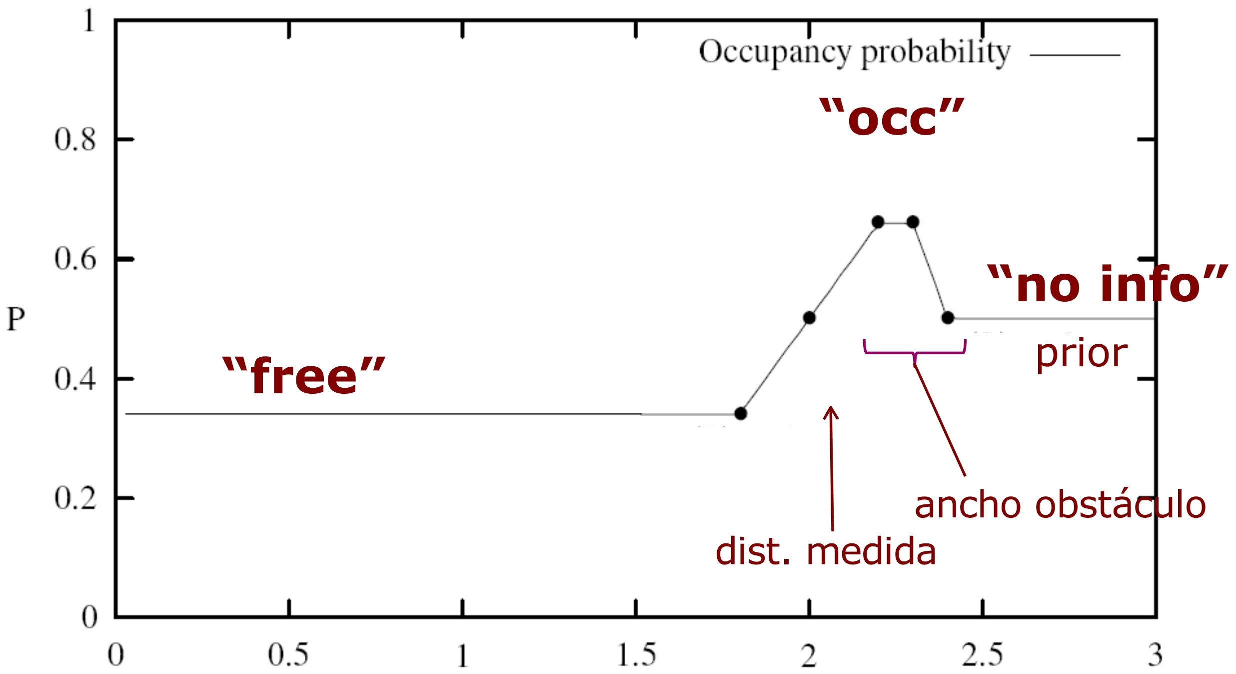
inv_sensor_model **para sonar**

- El haz del sonar es espacialmente como un cono (triángulo en 2D),
- Intuitivamente dada una medida, **en la zona del cono** :
    - La probabilidad de que las celdas por delante de la medida estén ocupadas es baja
    - La probabilidad de que las celdas más o menos a la distancia de la medida estén ocupadas es alta
- La probabilidad de ocupación fuera del cono es desconocida



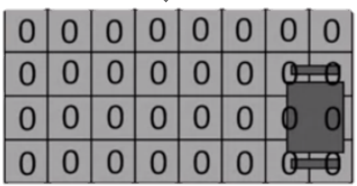
Probabilidad de ocupación

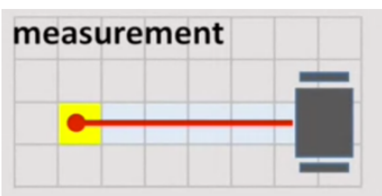
- Lo mismo que antes pero visto de perfil

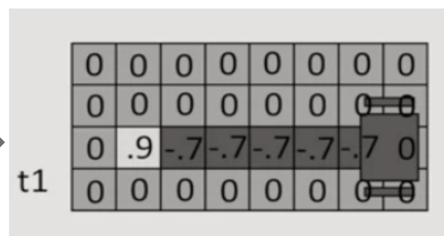


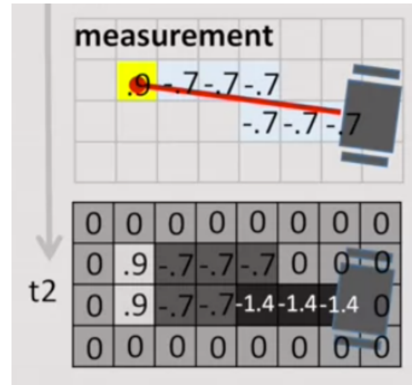
Ejemplo: Occupancy Grid Mapping

Mapa inicial:
- $p(m_i=1) = 0.5$ =>
- $odds(m_i=1) = 1$ => $L(m_i=1) = 0$ en todo el mapa





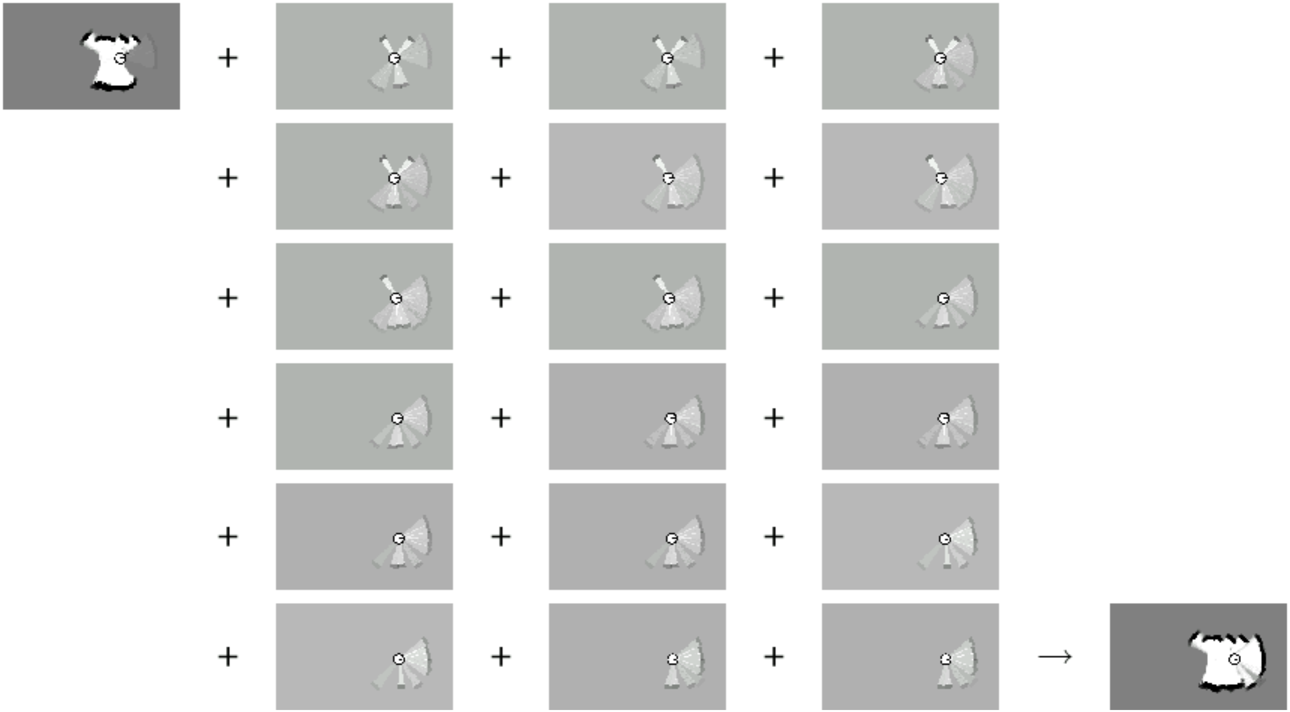




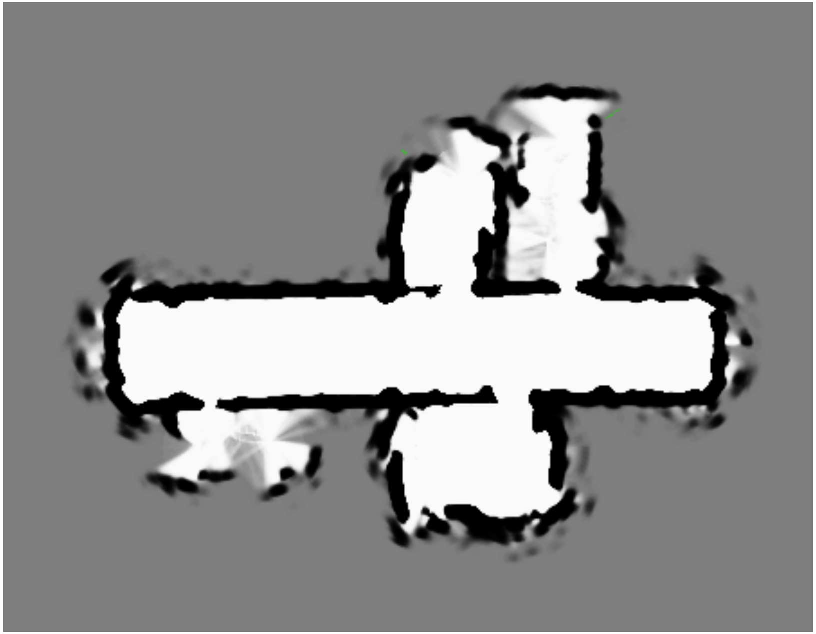
Modelo inverso sensor
- Celda lectura: $L(m_i=1)=0.9$
- Celdas por delante: $L(m_i=1)=-0.7$

https://es.coursera.org/lecture/robotics-learning/3-2-2-log-odd-update-Uh32O

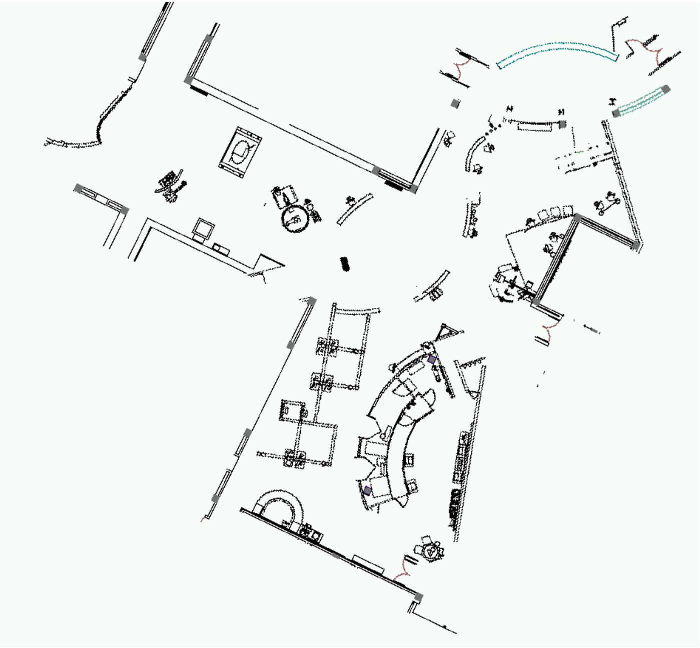
Actualización incremental malla ocupación



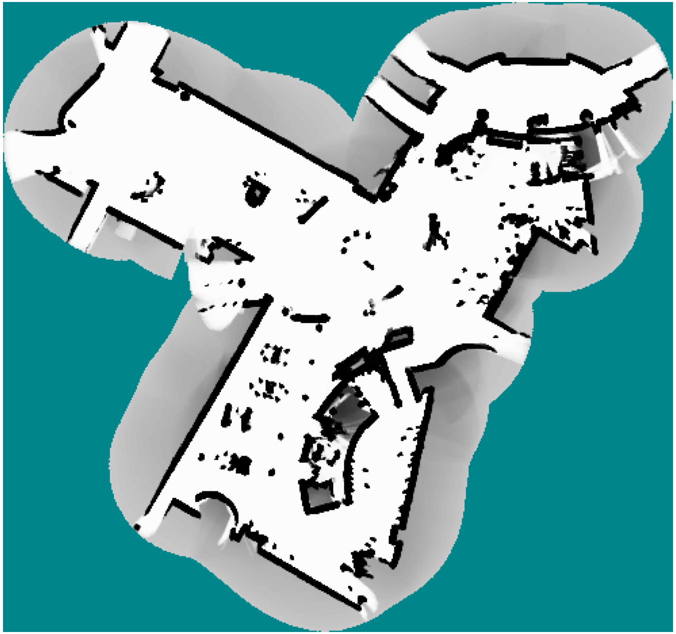
Mapa generado



Tech Museum, San Jose

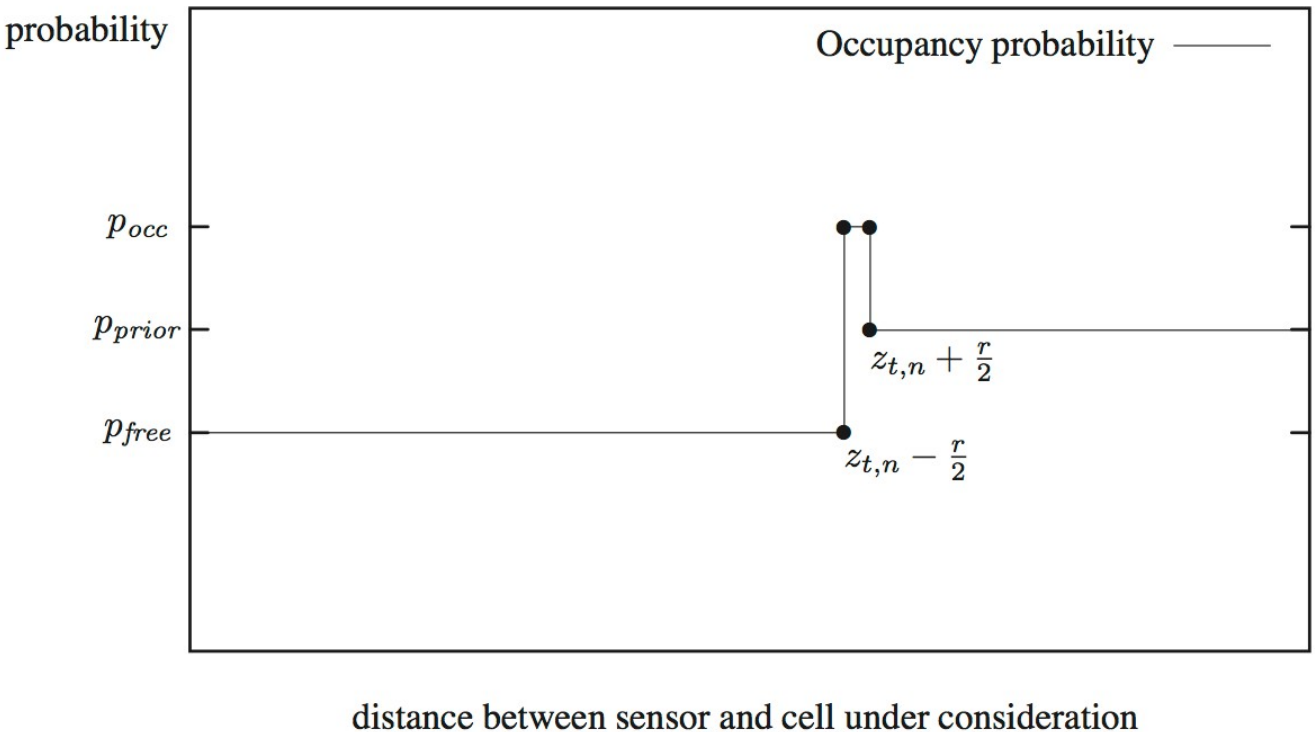


Mapa CAD

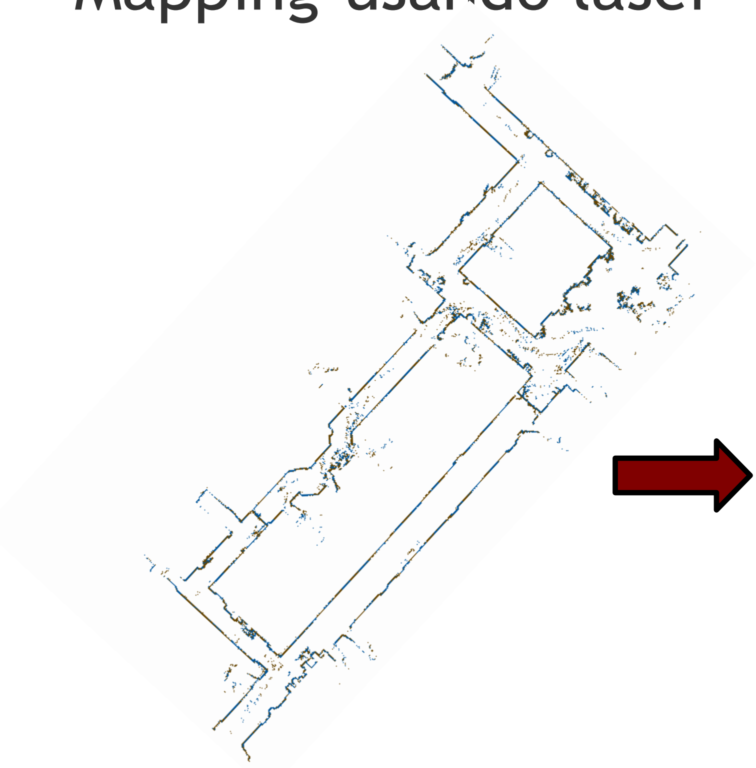


Malla de ocupación

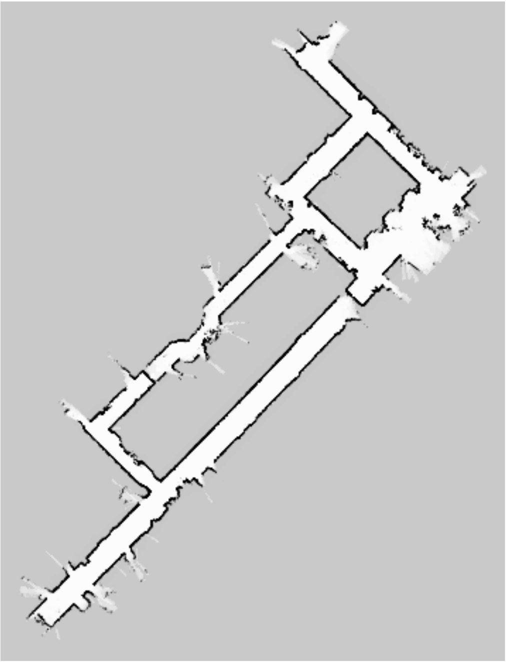
Inverse Sensor Model para Laser



Mapping usando laser

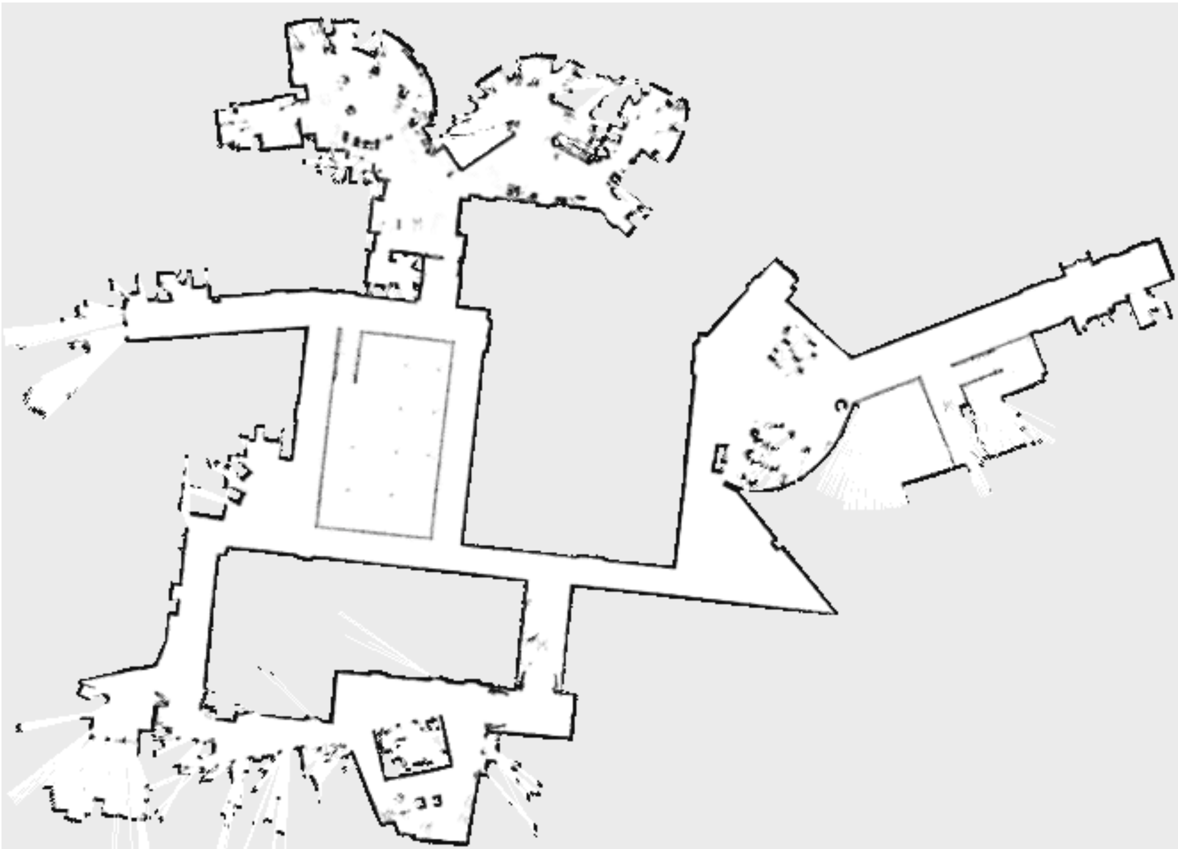


Datos del sensor “en bruto”. Se puede observar ruido causado por obstáculos móviles (personas, normalmente)

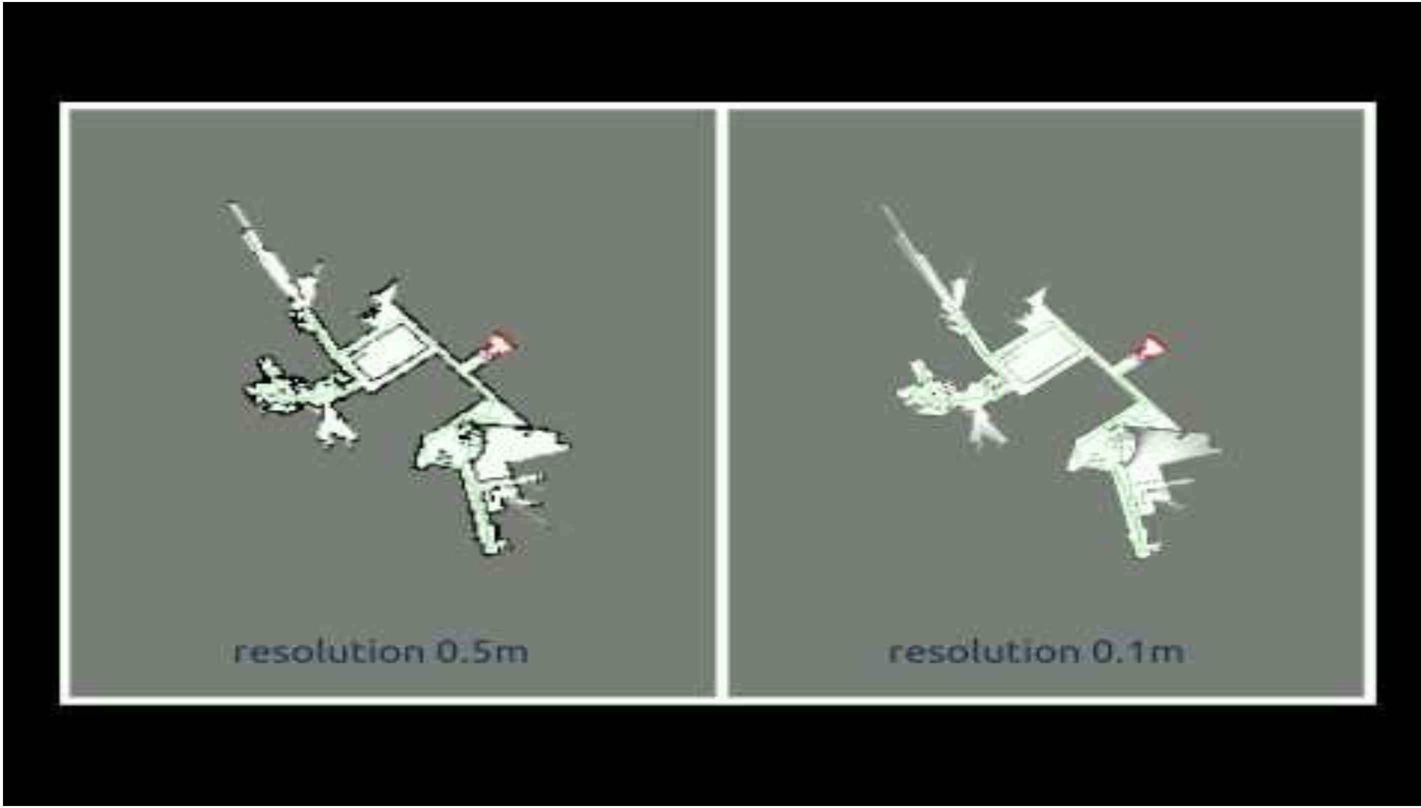


Mapa generado por el algoritmo. Al irse acumulando la evidencia, se filtra el ruido

Ejemplo: MIT CSAIL 3rd Floor



Video



Resumen

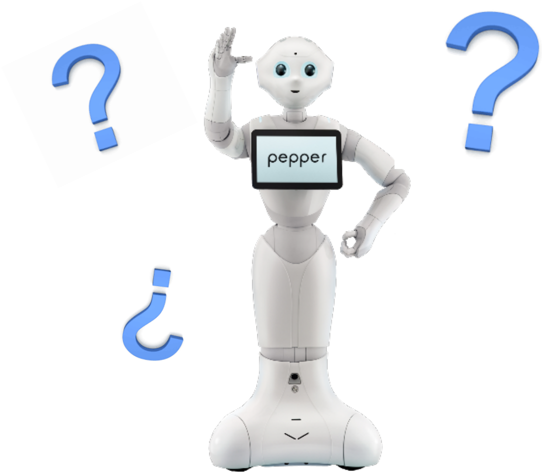
- Los mapas basados en grid de ocupación son un modelo popular para representar el entorno.
- Los mapas basados en grid de ocupación discretizan el espacio en celdas independientes.
- Cada celda es una variable aleatoria binaria que estima si la celda está ocupada
- Estimamos el estado de cada celda utilizando un filtro de Bayes binario
- Esto conduce a un algoritmo eficiente para mapear asumiendo que las poses del robot son conocidas.

Bibliografía

- Thrun, Burgard, Fox. Probabilistic Robotics. MIT Press, 2005. Cap. 4 (Filtros bayesianos binaries) y 9 (Occupancy grid mapping)
- Algunas transparencias tomadas de Wolfram Burgard

Robots Móviles

Mapping



Otto Colomina Pardo
Sergio Orts Escolano