IIC2685 Robótica Móvil
I – 2022

Capítulo 4.4

Mapping con Poses Conocidas

Profesor: Gabriel Sepúlveda V.
grsepulveda@ing.puc.cl

Agenda

- Entender para qué un robot necesita un mapa
- Fundamentos de creación de mapas dadas las poses

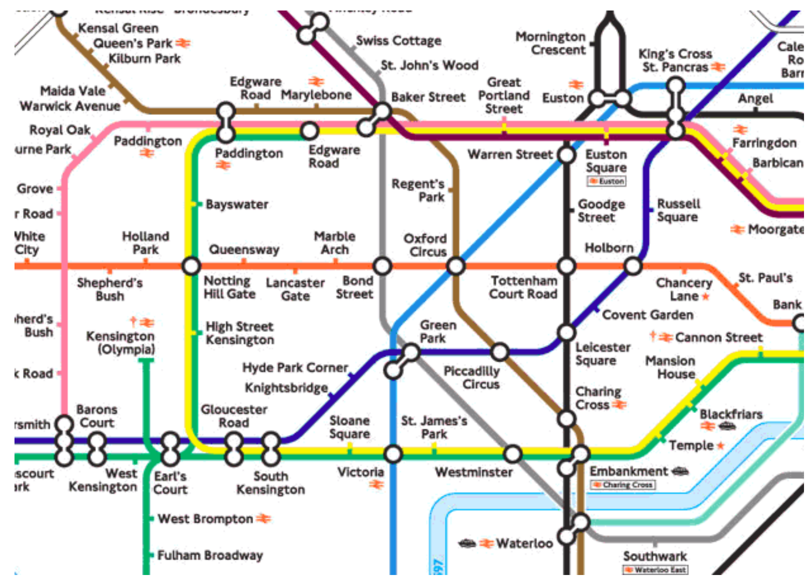
Mapping

- ¿ Es realmente necesario un mapa ?
  - Insectos: sólo comportamientos reactivos
  - Animales “superiores”: comportamientos reactivos + memoria
- Robots
  - Mapa es su memoria
  - Comportamientos reactivos siguen siendo importantes, aunque conocer el mapa permite comportamientos eficientes
  - Robots exitosos usan el mapa para localización, planear rutas, pleanear actividades, etc.

Mapping

- Dos tipos de mapas típicos
  - Métricos: representación de espacios ocupados/desocupados (grid map)
  - Topológicos: basado en landmarks (mapa de conectividad)
- Nos enfocaremos en mapas métricos.

Métrico



Topológico

Problema general de mapping

- Recordemos las Grandes Preguntas de un robot móvil

¿Dónde estoy?

¿Para dónde voy?

¿Cómo llego?

?

Problema general de mapping

- Recordemos las Grandes Preguntas de un robot móvil

¿Dónde estoy?

¿Para dónde voy?

¿Cómo llego?

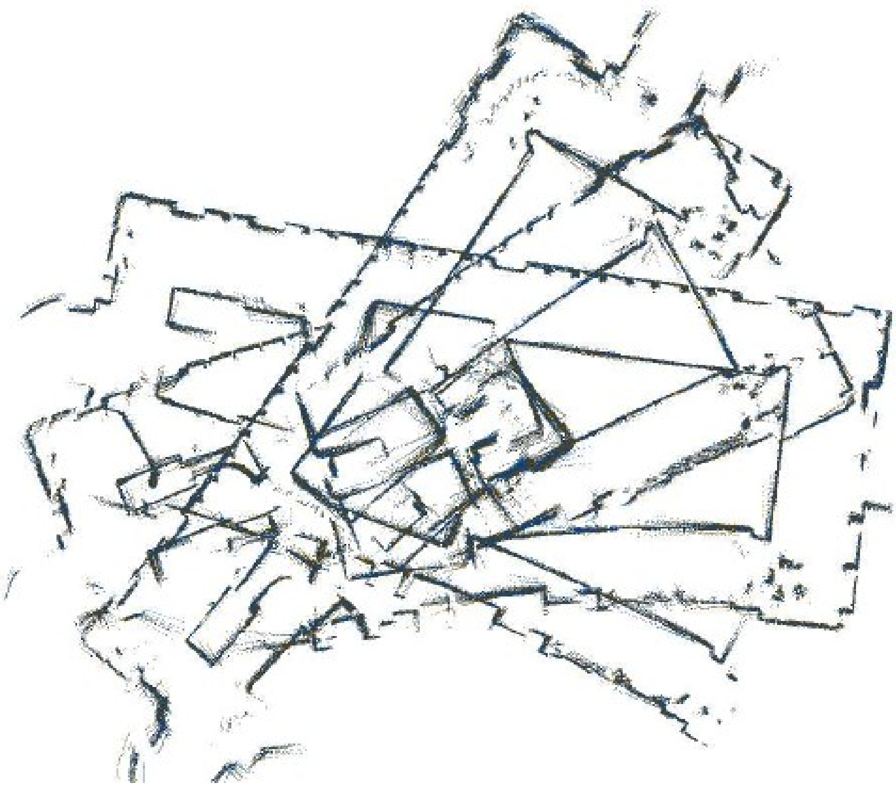
En todas las preguntas está involucrado el mapa (o memoria)!

Problema general de mapping

¿ Por qué es necesario elaborar técnicas *sofisticadas* de mapeo ?

Problema general de mapping

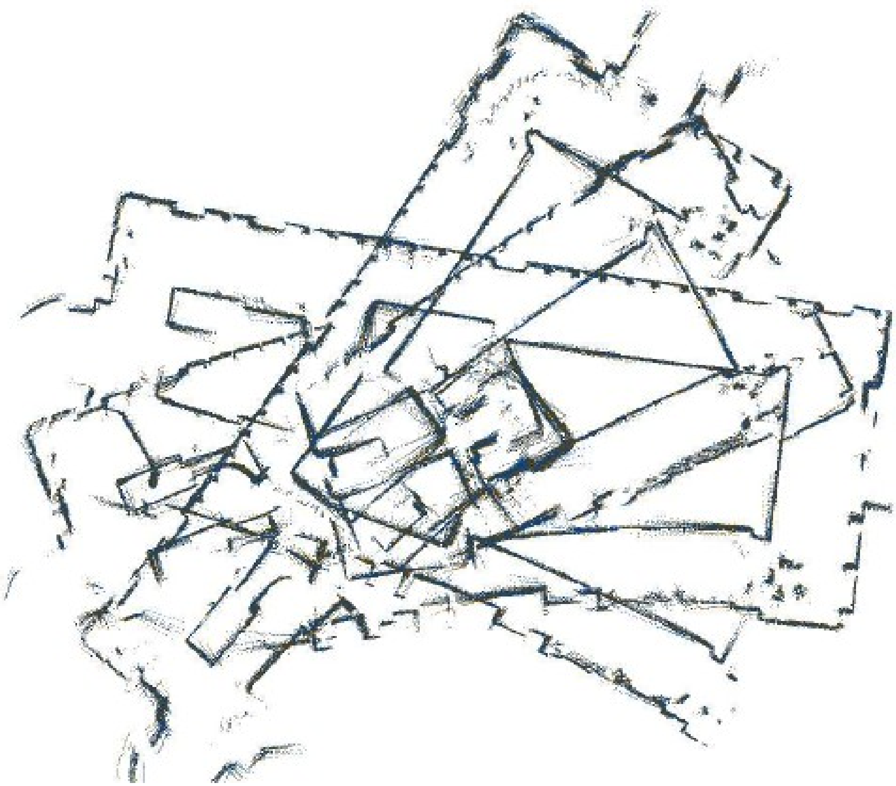
¿ Por qué es necesario elaborar técnicas *sofisticadas* de mapeo ?



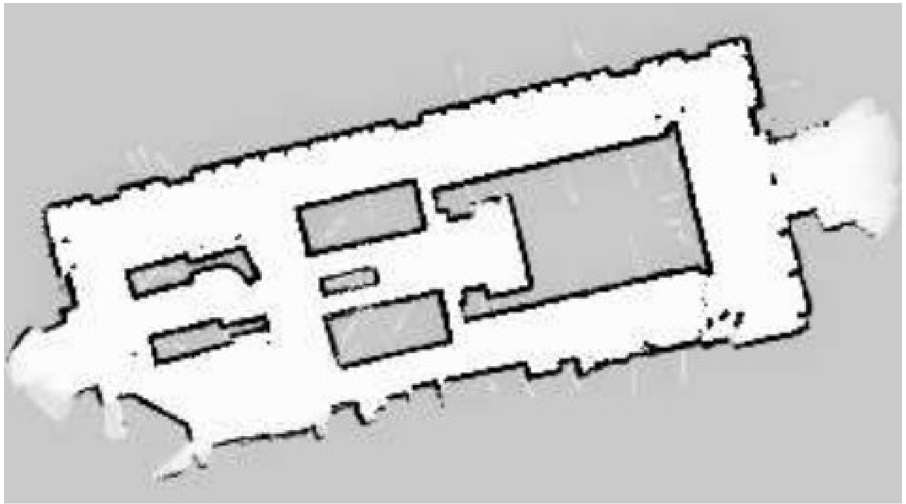
Datos de Odometría

Problema general de mapping

¿ Por qué es necesario elaborar técnicas *sofisticadas* de mapeo ?



Datos de Odometría



Occupancy Grid Map

Occupancy Grid maps

- Mapa métrico, discretizado en celdas (por ejemplo, de 5 [cm] x 5 [cm])
- La estructura de grid es rígida
- Cada celda tiene tres estados: ocupada /desocupada / desconocido
- Por lo general se usa un sensor de barrido 2D (ej: LIDAR, sonar, etc.)

Mapping - definiciones

- Supondremos que nos movemos en un ambiente 2D (ej: TurtleBot), y que las poses $x_i = (x, y, \theta)_i$ son conocidas

- Formalización: dados los datos de el (los) sensor(es) ***z*** y las poses ***x***

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

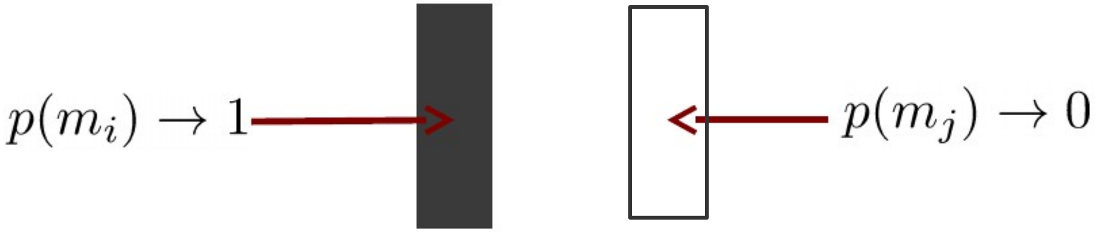
debemos calcular el mapa *m* “más probable”

$$m^* = \mathrm{argmax}_m P(m|d)$$

Occupancy Grid Maps - supuestos

- El área que corresponde a una celda puede estar *ocupada* o *desocupada*

- Cada celda es representada por una variable aleatoria binaria m_i
  - Celda ocupada : $p(m_i = ocupada) = p(m_i) = 1$
  - Celda desocupada : $p(m_i = ocupada) = p(m_i) = 0$
  - Celda desconocida : $p(m_i = ocupada) = p(m_i) = 0.5$



Occupancy Grid Maps - supuestos

- El mapa es la unión de todas las celdas. Cada una de ellas es estática (no cambia en el tiempo)

$$m = \{m_1, m_2, \ldots, m_N\}$$

- Supondremos que los estados de las celdas son condicionalmente independientes dadas las observaciones y posiciones del robot

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

$$P(m|d) = \prod_{i=1}^{N} P(m_i|d)$$

Occupancy Grid Maps – derivación

- Queremos calcular: $p(m_i \mid z_{1:t}, x_{1:t})$

Occupancy Grid Maps – derivación

- Queremos calcular: $p(m_i|z_{1:t}, x_{1:t})$

$$p(m_i|z_{1:t}, x_{1:t}) \overset{\text{Bayes}}{=} \frac{p(z_t|m_i, z_{1:t-1}, x_{1:t}) \cdot p(m_i|z_{1:t-1}, x_{1:t})}{p(z_t|z_{1:t-1}, x_{1:t})}$$

Occupancy Grid Maps – derivación

- Queremos calcular: $p(m_i | z_{1:t}, x_{1:t})$

$$p(m_i | z_{1:t}, x_{1:t}) \overset{\text{Bayes}}{=} \frac{p(z_t | m_i, z_{1:t-1}, x_{1:t}) \cdot p(m_i | z_{1:t-1}, x_{1:t})}{p(z_t | z_{1:t-1}, x_{1:t})}$$

$$\overset{\text{Markov}}{=} \frac{p(z_t | m_i, x_t) \cdot p(m_i | z_{1:t-1}, x_{1:t-1})}{p(z_t | z_{1:t-1}, x_{1:t})}$$

Occupancy Grid Maps – derivación

- Queremos calcular: $p(m_i | z_{1:t}, x_{1:t})$

$$
\begin{aligned}
p(m_i | z_{1:t}, x_{1:t}) &\overset{\text{Bayes}}{=} \frac{p(z_t | m_i, z_{1:t-1}, x_{1:t}) \cdot p(m_i | z_{1:t-1}, x_{1:t})}{p(z_t | z_{1:t-1}, x_{1:t})} \\
&\overset{\text{Markov}}{=} \frac{p(z_t | m_i, x_t) \cdot p(m_i | z_{1:t-1}, x_{1:t-1})}{p(z_t | z_{1:t-1}, x_{1:t})} \\
&\overset{\text{Bayes}}{=} \frac{p(m_i | z_t, x_t) \cdot p(z_t | x_t) \cdot p(m_i | z_{1:t-1}, x_{1:t-1})}{p(m_i | x_t) \cdot p(z_t | z_{1:t-1}, x_{1:t})}
\end{aligned}
$$

Occupancy Grid Maps – derivación

- Queremos calcular: $p(m_i | z_{1:t}, x_{1:t})$

$$p(m_i | z_{1:t}, x_{1:t}) \overset{\text{Bayes}}{=} \frac{p(z_t | m_i, z_{1:t-1}, x_{1:t}) \cdot p(m_i | z_{1:t-1}, x_{1:t})}{p(z_t | z_{1:t-1}, x_{1:t})}$$

$$\overset{\text{Markov}}{=} \frac{p(z_t | m_i, x_t) \cdot p(m_i | z_{1:t-1}, x_{1:t-1})}{p(z_t | z_{1:t-1}, x_{1:t})}$$

$$\overset{\text{Bayes}}{=} \frac{p(m_i | z_t, x_t) \cdot p(z_t | x_t) \cdot p(m_i | z_{1:t-1}, x_{1:t-1})}{p(m_i | x_t) \cdot p(z_t | z_{1:t-1}, x_{1:t})}$$

$$\overset{\text{Independencia}}{=} \frac{p(m_i | z_t, x_t) \cdot p(z_t | x_t) \cdot p(m_i | z_{1:t-1}, x_{1:t-1})}{p(m_i) \cdot p(z_t | z_{1:t-1}, x_{1:t})}$$

Occupancy Grid Maps – derivación

- Para problemas de estados binarios, es conveniente expresar las probabilidades como razón de probabilidades (*odds ratio*):

$$\frac{p(x)}{p(\neg x)} = \frac{p(x)}{1 - p(x)}$$

- O bien, su versión logarítmica (*log odds ratio*)

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

- Esto evita truncado en 0 y 1, lo cual mejora estabilidad numérica
- Además, se cuenta con función inversa:

$$p(x) = 1 - \frac{1}{1 + e^{l(x)}}$$

Occupancy Grid Maps – derivación

- Según lo anterior tenemos:

$$p(m_i | z_{1:t}, x_{1:t}) = \frac{p(m_i | z_t, x_t) \cdot p(z_t | x_t) \cdot p(m_i | z_{1:t-1}, x_{1:t-1})}{p(m_i) \cdot p(z_t | z_{1:t-1}, x_{1:t})}$$

$$p(\neg m_i | z_{1:t}, x_{1:t}) = \frac{p(\neg m_i | z_t, x_t) \cdot p(z_t | x_t) \cdot p(\neg m_i | z_{1:t-1}, x_{1:t-1})}{p(\neg m_i) \cdot p(z_t | z_{1:t-1}, x_{1:t})}$$

- De lo cual obtenemos:

$$\frac{p(m_i | z_{1:t}, x_{1:t})}{p(\neg m_i | z_{1:t}, x_{1:t})} = \frac{\dfrac{p(m_i | z_t, x_t) \cdot \cancel{p(z_t | x_t)} \cdot p(m_i | z_{1:t-1}, x_{1:t-1})}{p(m_i) \cdot \cancel{p(z_t | z_{1:t-1}, x_{1:t})}}}{\dfrac{p(\neg m_i | z_t, x_t) \cdot \cancel{p(z_t | x_t)} \cdot p(\neg m_i | z_{1:t-1}, x_{1:t-1})}{p(\neg m_i) \cdot \cancel{p(z_t | z_{1:t-1}, x_{1:t})}}}$$

Occupancy Grid Maps – derivación

- Obteniendo finalmente:

$$\frac{p(m_i|z_{1:t},x_{1:t})}{1-p(m_i|z_{1:t},x_{1:t})} = \underbrace{\frac{p(m_i|z_t,x_t)}{1-p(m_i|z_t,x_t)}}_{sensor\ inverso} \cdot \underbrace{\frac{p(m_i|z_{1:t-1},x_{1:t-1})}{1-p(m_i|z_{1:t-1},x_{1:t-1})}}_{termino\ recursivo} \cdot \underbrace{\frac{1-p(m_i)}{p(m_i)}}_{prior^{-1}}$$

- Cuya versión logarítmica toma la forma de:

$$l(m_i|z_{1:t},x_{1:t}) = \underbrace{l(m_i|z_t,x_t)}_{sensor\ inverso} + \underbrace{l(m_i|z_{1:t-1},x_{1:t-1})}_{termino\ recursivo} - \underbrace{l(m_i)}_{prior}$$

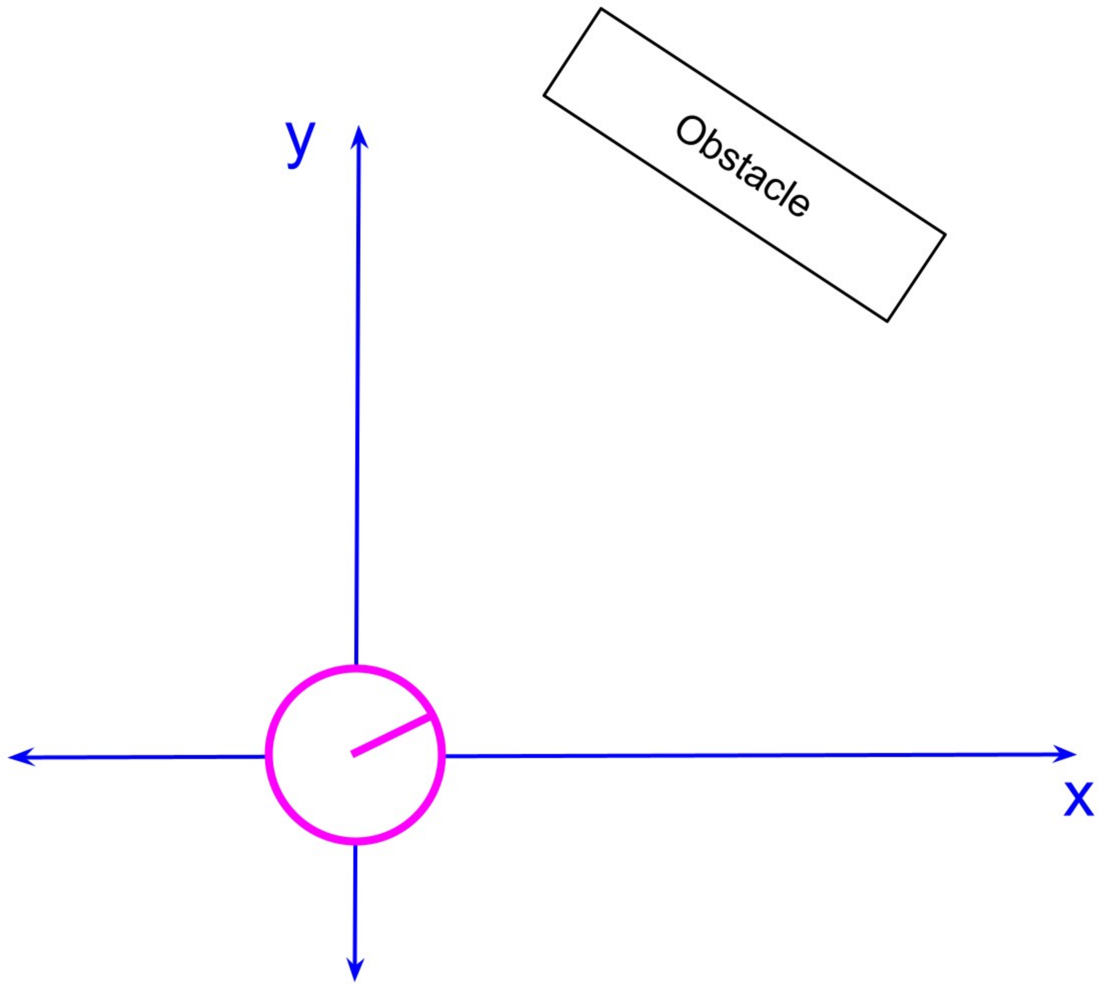
Occupancy Grid Maps - derivación

- Con esta expresión, es posible definir el algoritmo de mapeo:

Algorithm occupancy_grid_mapping($\{l_{t-1,i}\}, x_t, z_t$):
    for all cells $\mathbf{m}_i$ do
        if $\mathbf{m}_i$ in perceptual field of z_t then
            $l_{t,i} = l_{t-1,i} + \textbf{inverse\_sensor\_model}(\mathbf{m}_i, x_t, z_t) - l_0$
        else
            $l_{t,i} = l_{t-1,i}$
        endif
    endfor
    return $\{l_{t,i}\}$

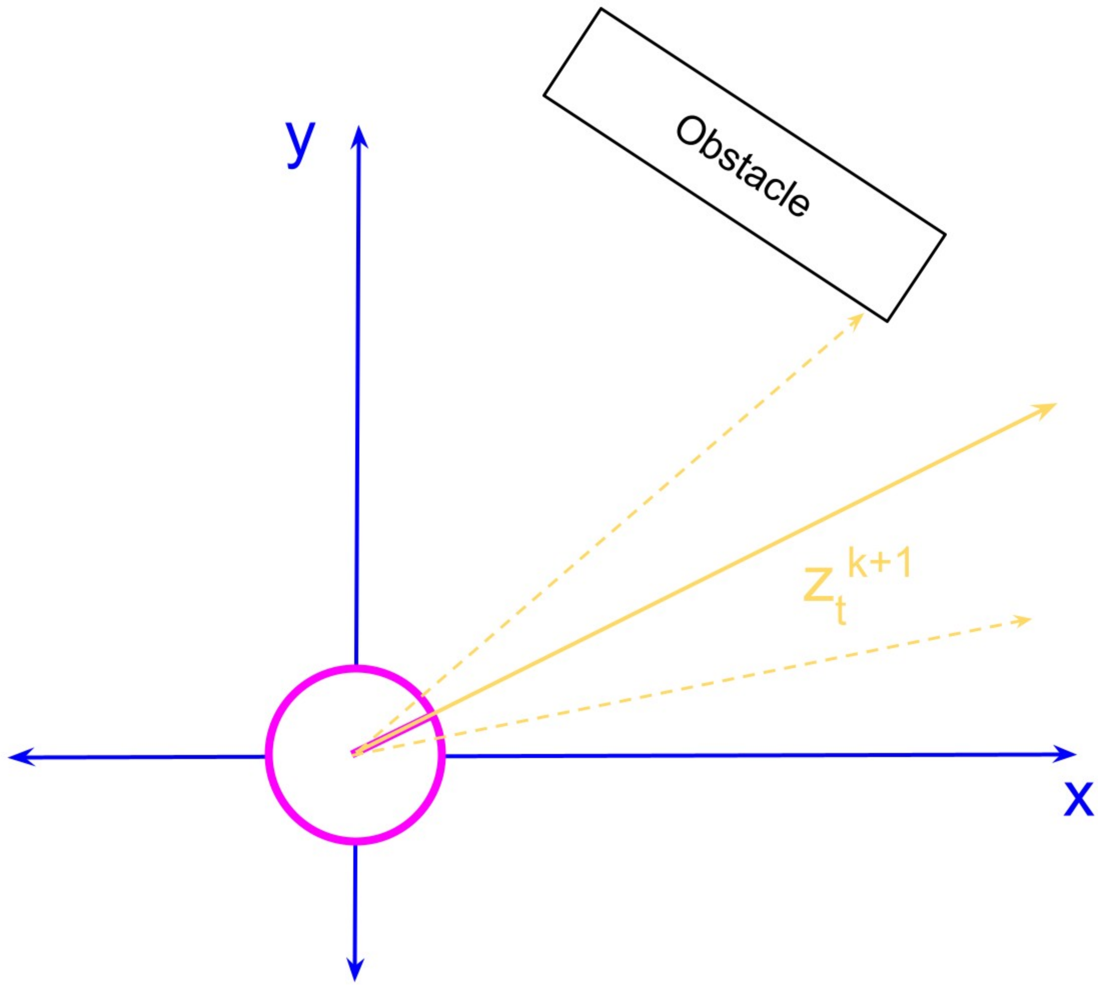
Occupancy Grid Maps - derivación

- Consideraciones geométricas para el Modelo de Sensor Inverso



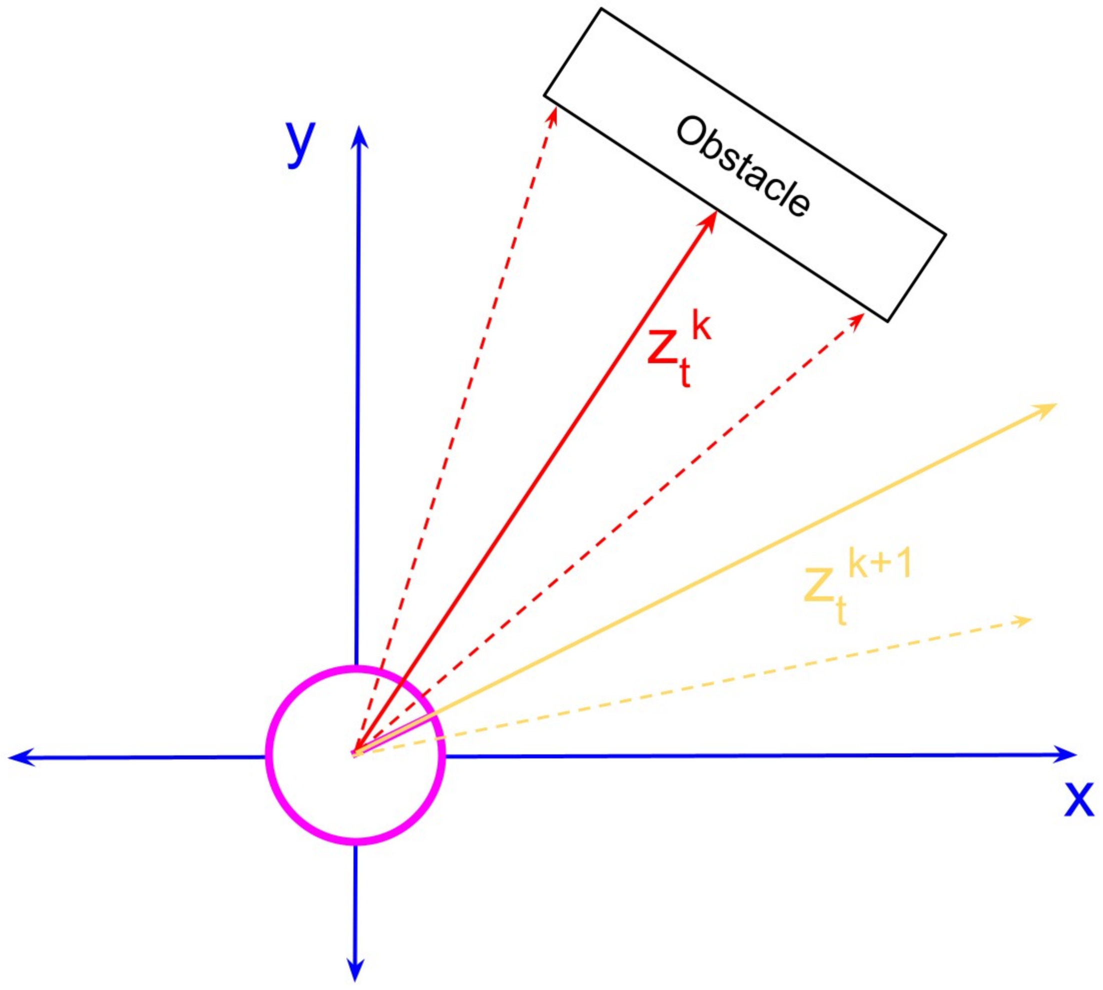
Occupancy Grid Maps - derivación

- Consideraciones geométricas para el Modelo de Sensor Inverso



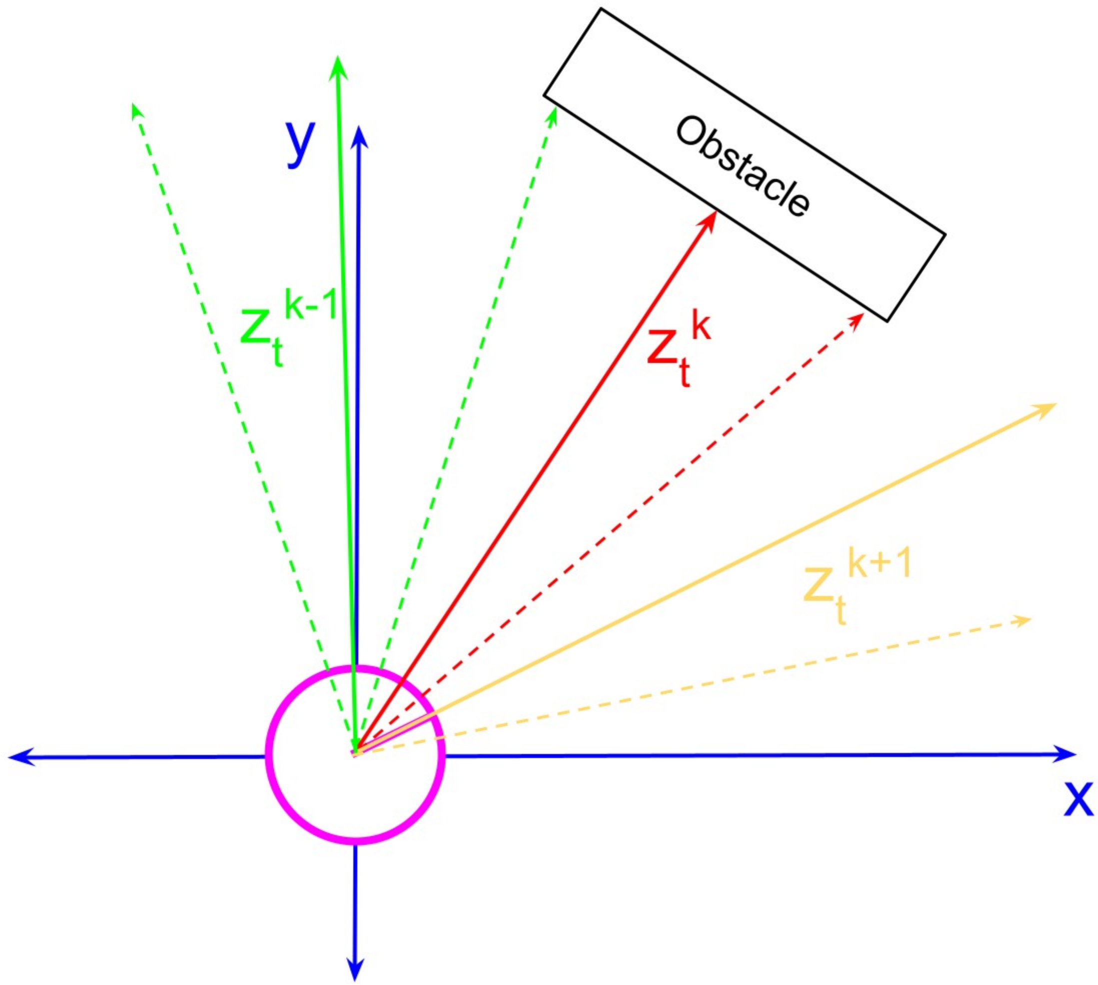
Occupancy Grid Maps - derivación

- Consideraciones geométricas para el Modelo de Sensor Inverso



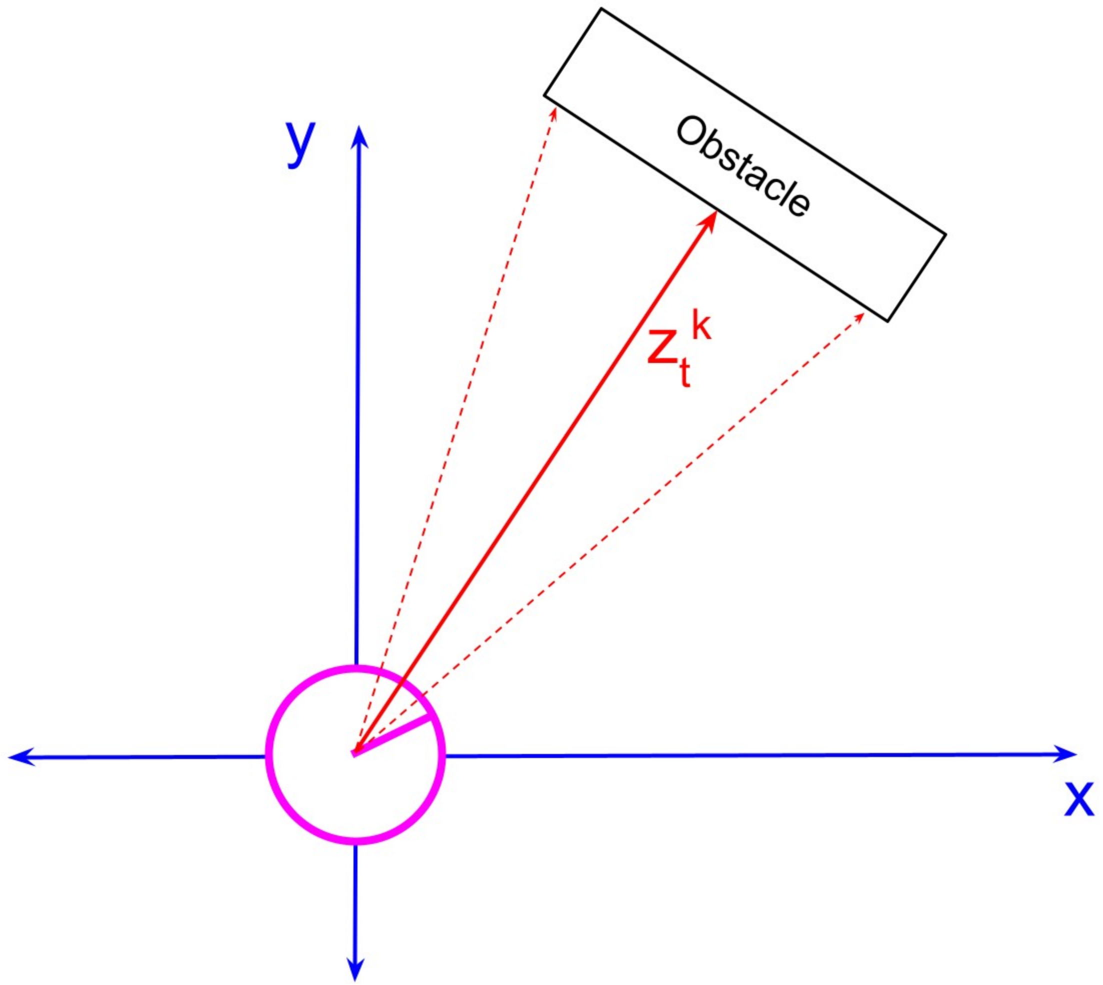
Occupancy Grid Maps - derivación

- Consideraciones geométricas para el Modelo de Sensor Inverso



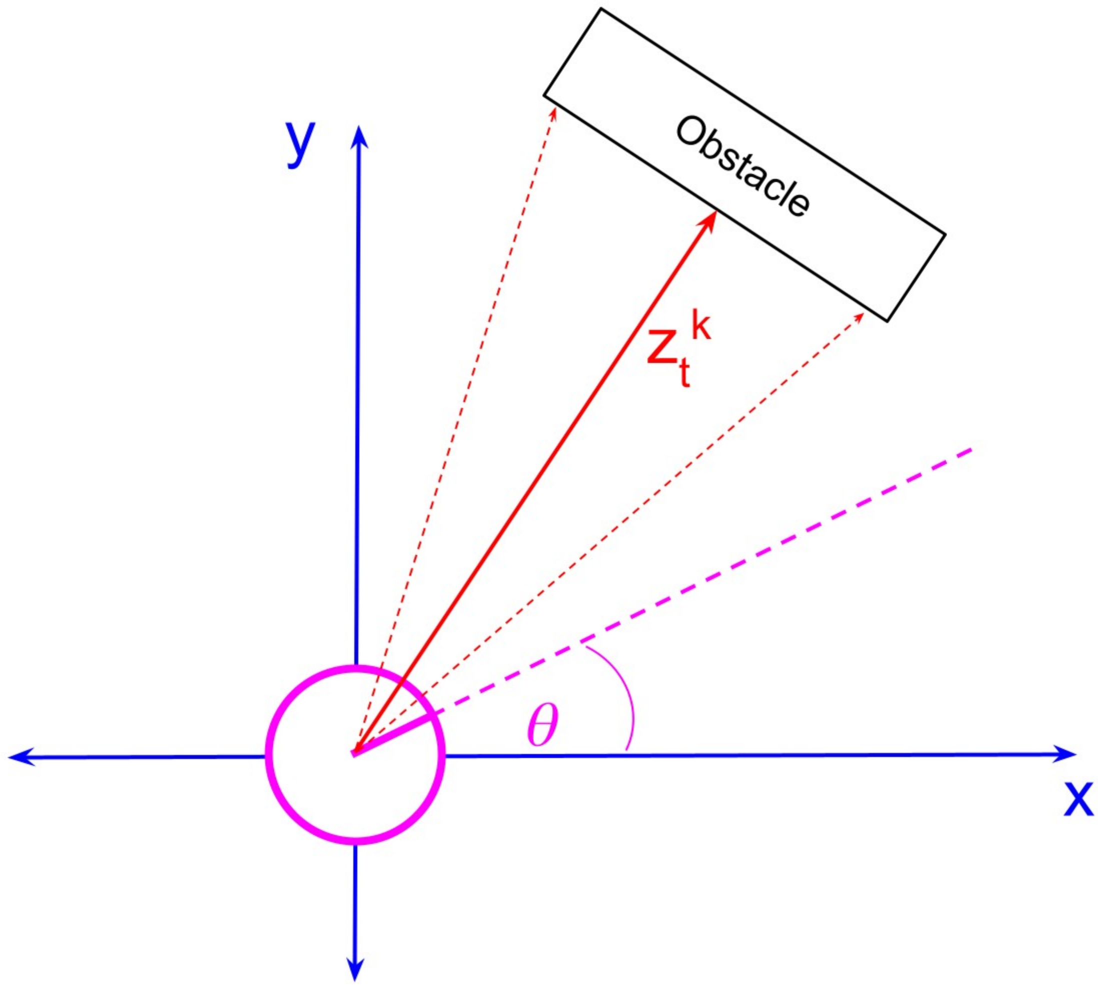
Occupancy Grid Maps - derivación

- Consideraciones geométricas para el Modelo de Sensor Inverso



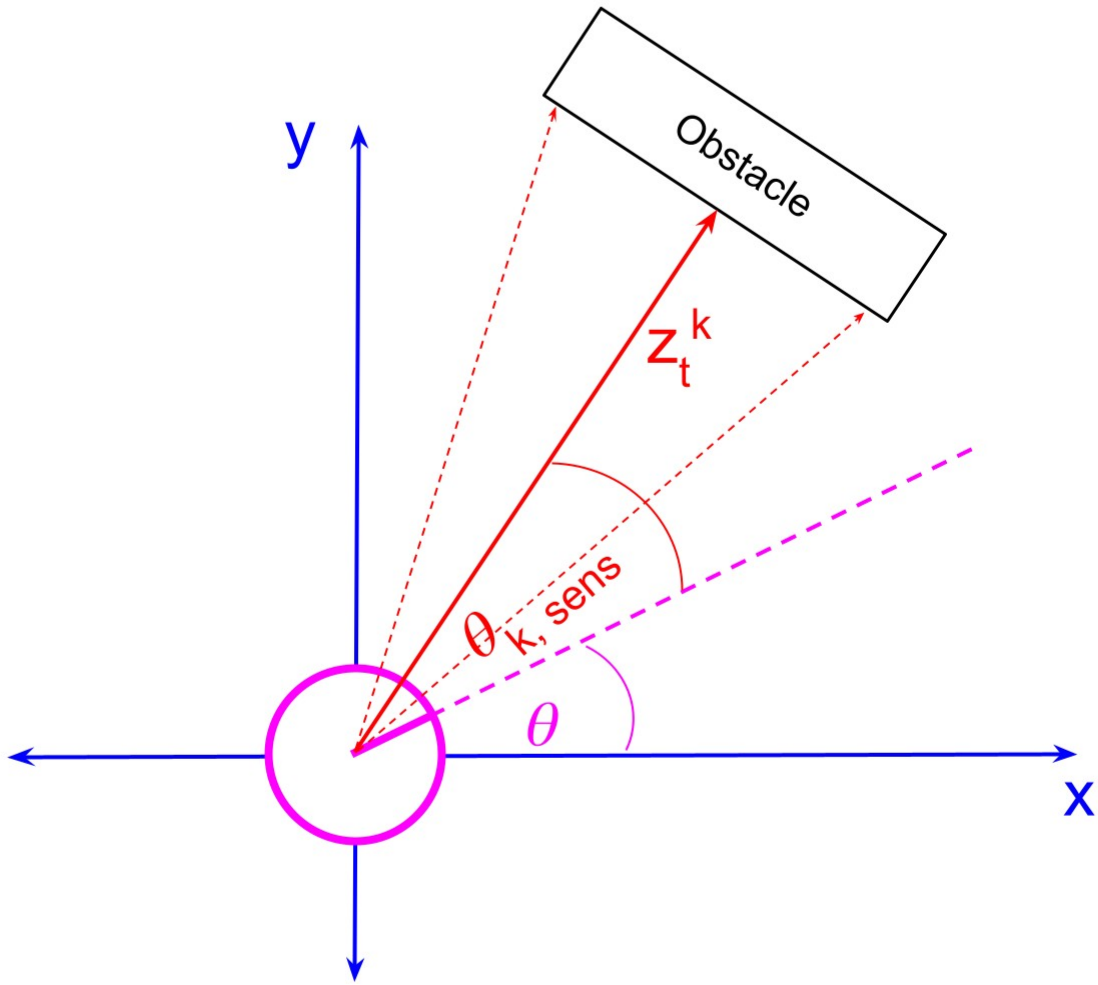
Occupancy Grid Maps - derivación

- Consideraciones geométricas para el Modelo de Sensor Inverso



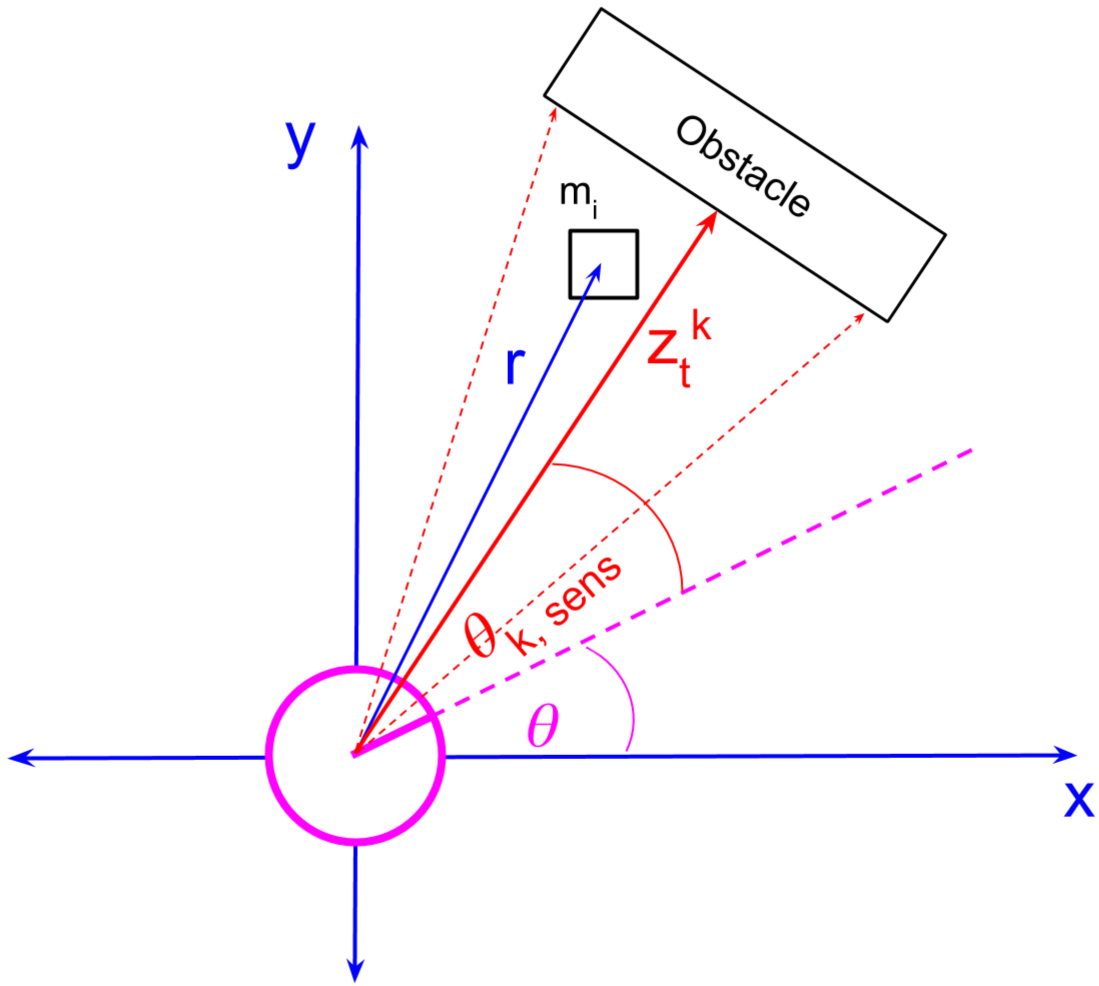
Occupancy Grid Maps - derivación

- Consideraciones geométricas para el Modelo de Sensor Inverso



Occupancy Grid Maps - derivación

- Consideraciones geométricas para el Modelo de Sensor Inverso



Occupancy Grid Maps - derivación

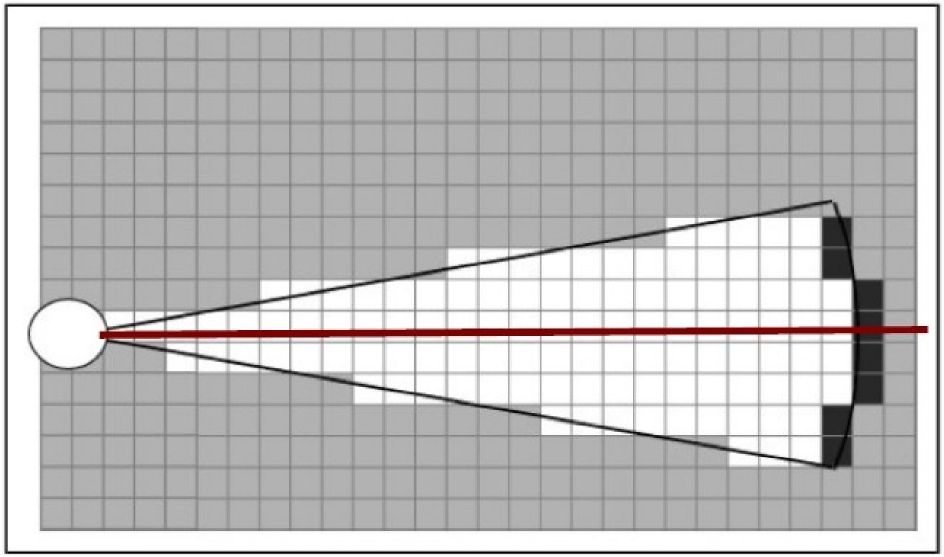
- Algoritmo *simple* para Modelo de Sensor Inverso

Algorithm inverse_range_sensor_model(i, x_t, z_t):
  Let x_i, y_i be the center-of-mass of $\mathbf{m}_i$
  $r = \sqrt{(x_i - x)^2 + (y_i - y)^2}$
  $\phi = \text{atan2}(y_i - y, x_i - x) - \theta$
  $k = \text{argmin}_j \, |\phi - \theta_{j,\text{sens}}|$
  if $r > \min(z_{\max}, z_t^k + \alpha/2)$ or $|\phi - \theta_{k,\text{sens}}| > \beta/2$ then
    return l_0
  if $z_t^k < z_{\max}$ and $|r - z_t^k| < \alpha/2$
    return l_{occ}
  if $r \leq z_t^k$
    return l_{free}
  endif

- α: grosor mínimo de obstáculos
- β: ángulo de apertura (cono de visión) de cada rayo del sensor
- l_0, l_{ooc} y l_{free}: valores de ocupación en formato *log odds ratio*

Occupancy Grid Maps

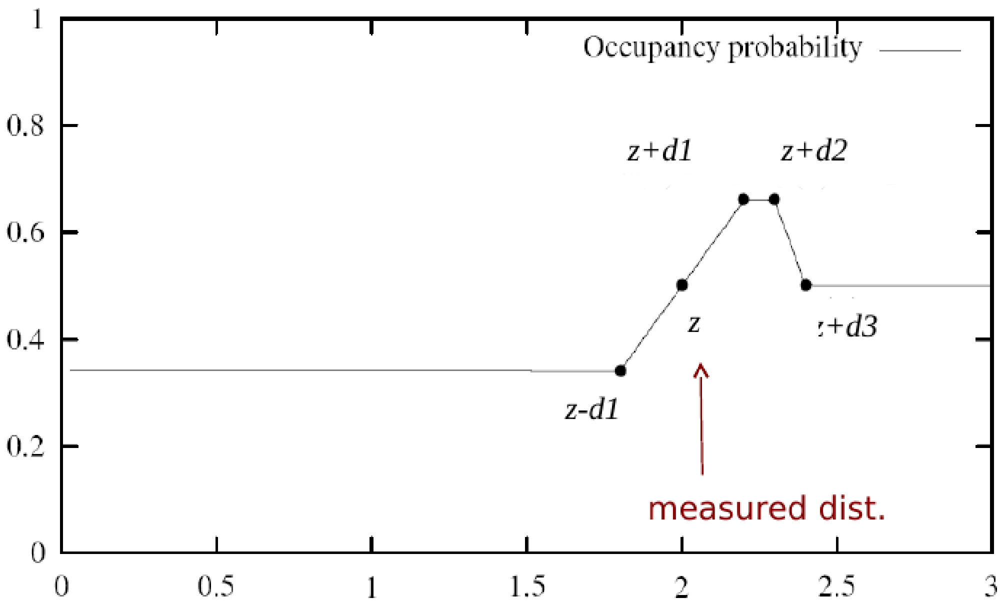
- “Perceptual field” del sensor
  - Depende de las características (ángulo de apertura de cada medición, distancia máxima)
  - Ej: Una lectura de un sonar tiene un ángulo amplio, por lo que una medición puede abarcar varias celdas



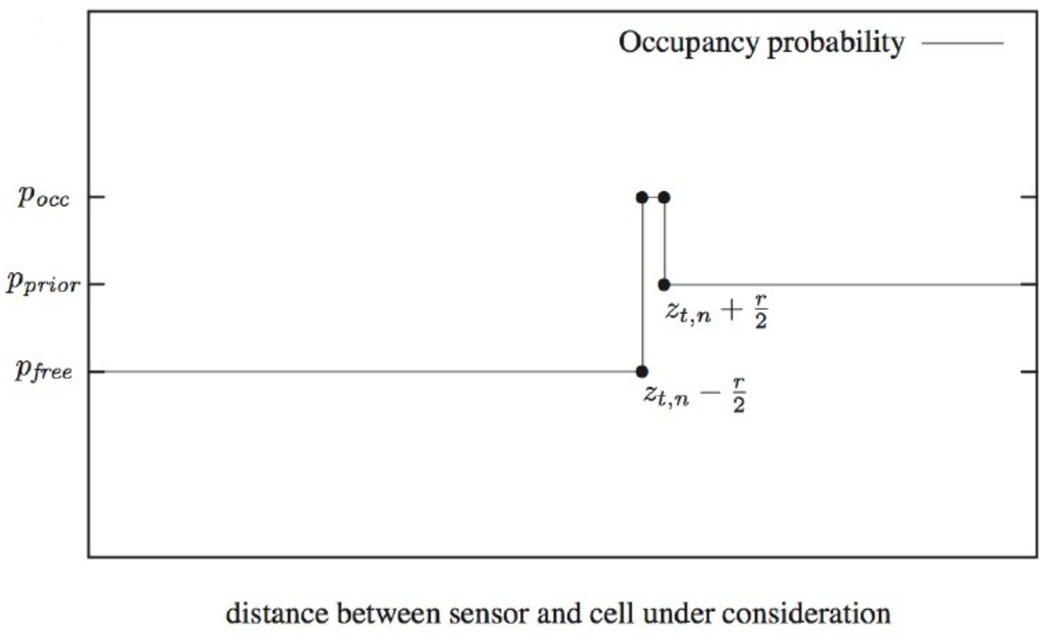
Occupancy Grid Maps

- Para una medición, el *inverse sensor model* debe considerar estas características
  - Intensidad del update lineal con relación a la distancia medida

Sonar

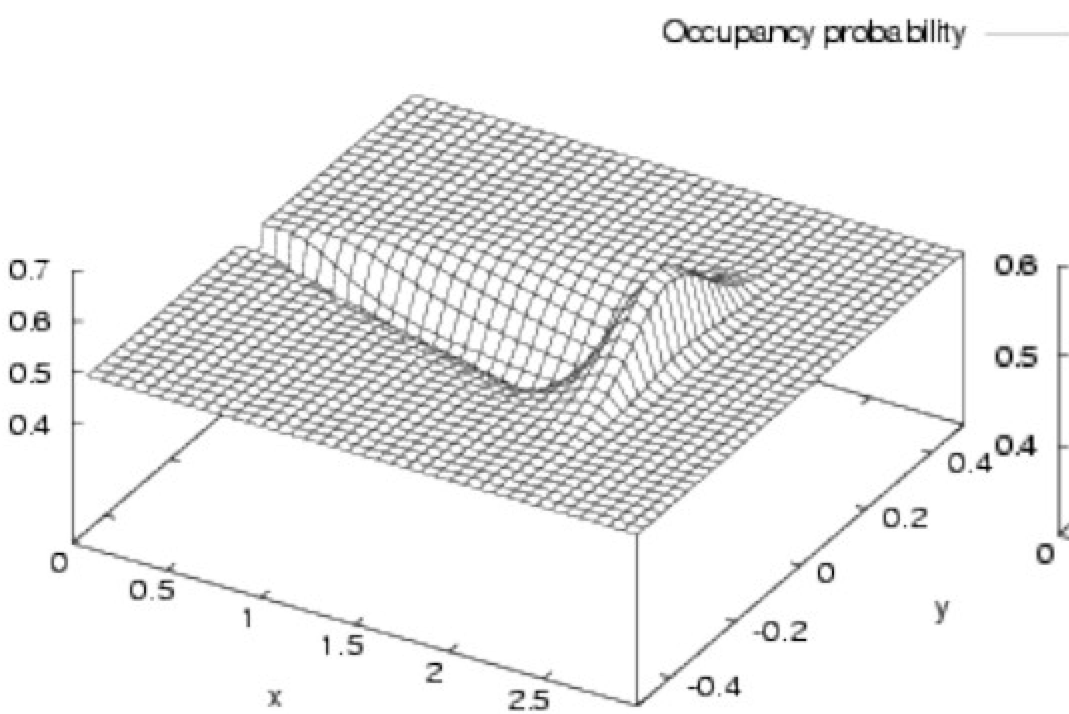


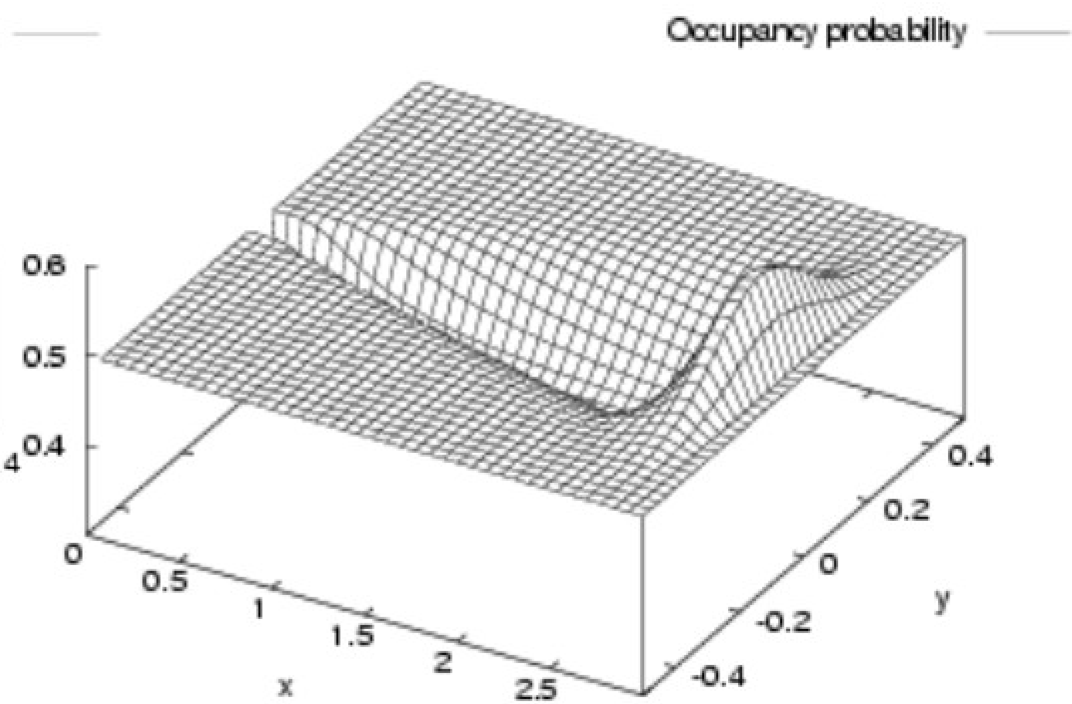
Laser



Occupancy Grid Maps

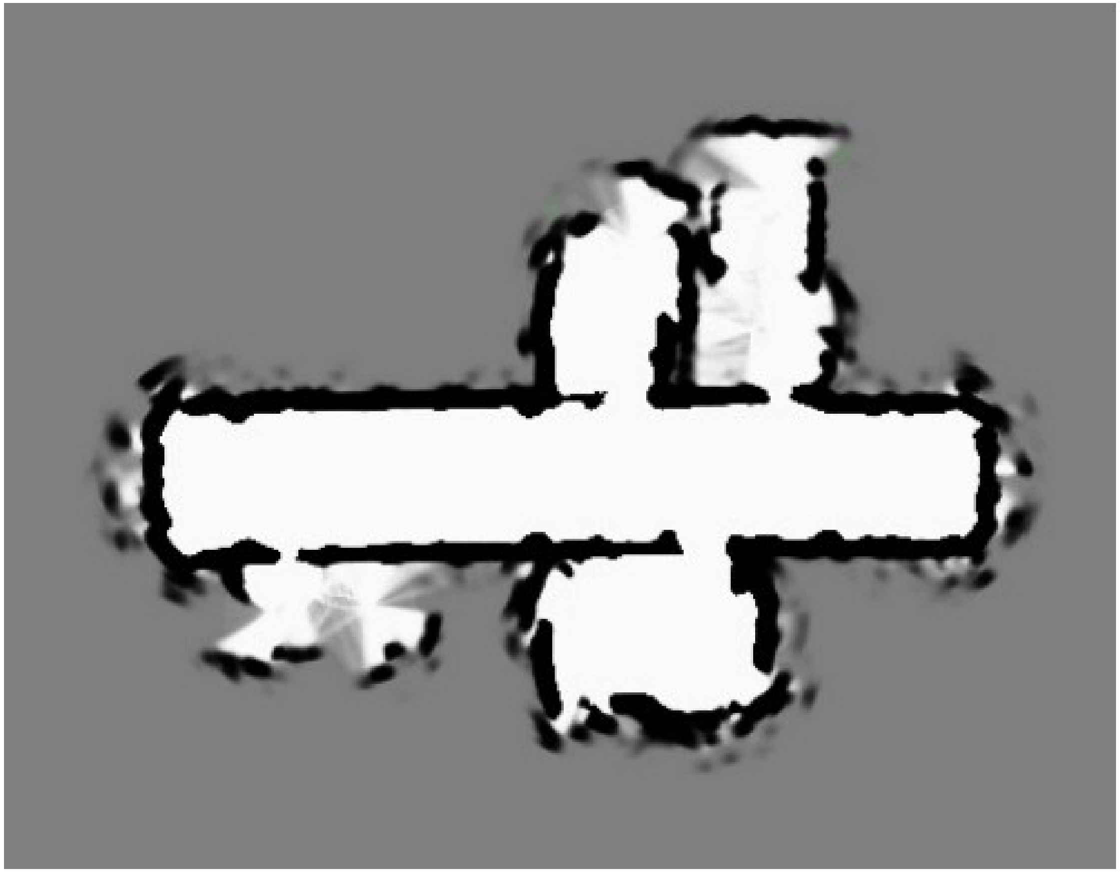
- Ángulo se ajusta a una distribución normal





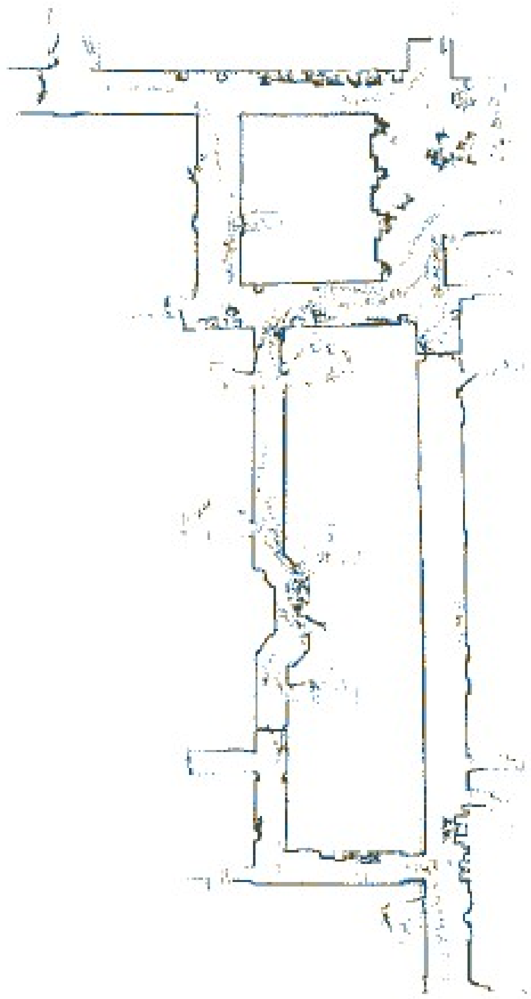
Occupancy Grid Maps

- Ejemplo de mapa con Sonar

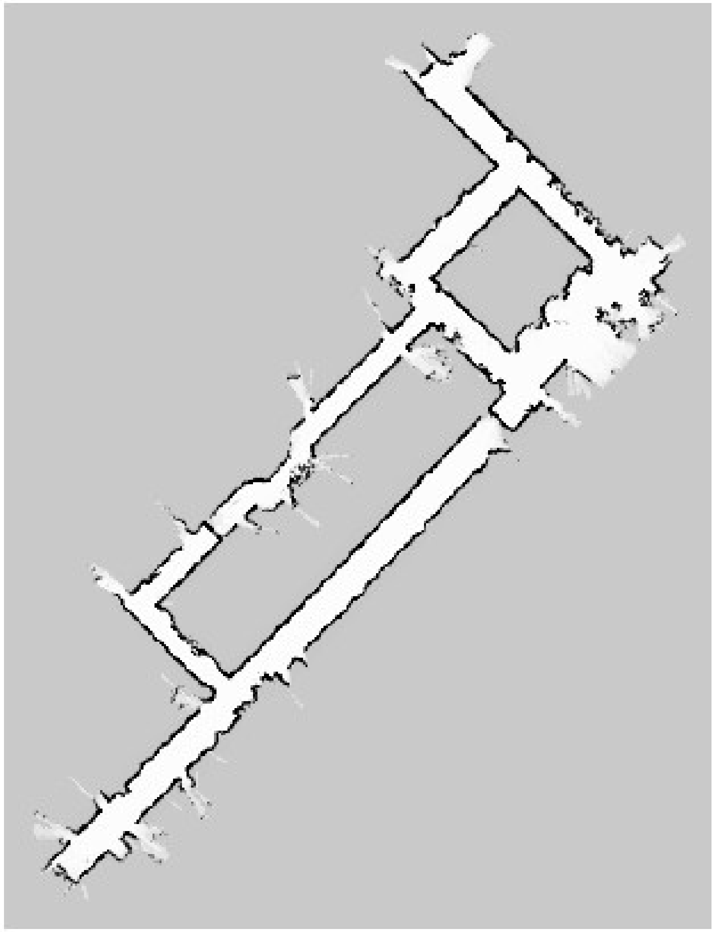


Occupancy Grid Maps

- Ejemplo de mapa con Lidar

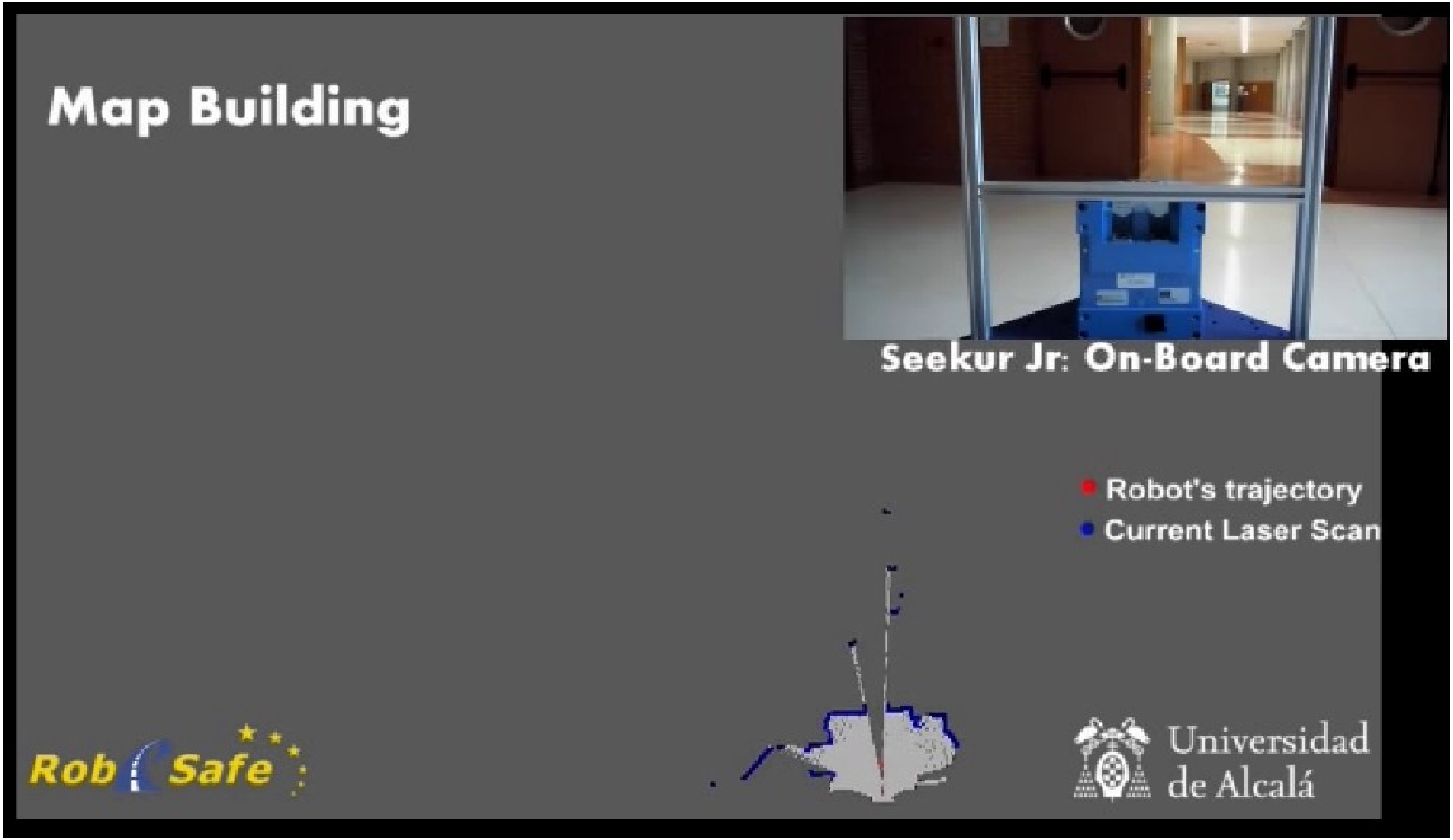


Medición cruda



Occupancy Grid Map

Grid maps



Map Building
Seekur Jr: On-Board Camera
Robot's trajectory
Current Laser Scan
Rob Safe
Universidad de Alcalá

Resumen

- Luego de este largo recorrido por los fundamentos de Robótica Probabilística tenemos
  - Localización con mapa conocido
  - Mapeo con pose conocida

Resumen

- Luego de este largo recorrido por los fundamentos de Robótica Probabilística tenemos
  - Localización con mapa conocido
  - Mapeo con pose conocida
- ¿ Qué fue primero ?, ¿ el huevo o la gallina ?

Resumen

- Luego de este largo recorrido por los fundamentos de Robótica Probabilística tenemos
  - Localización con mapa conocido
  - Mapeo con pose conocida
- ¿ Qué fue primero ?, ¿ el huevo o la gallina ?
- ¿ Cómo calculamos el mapa sin pose conocida ?, ¿ como obtenermos la pose sin contar con un mapa ?

Resumen

- Luego de este largo recorrido por los fundamentos de Robótica Probabilística tenemos
  - Localización con mapa conocido
  - Mapeo con pose conocida
- ¿ Qué fue primero ?, ¿ el huevo o la gallina ?
- ¿ Cómo calculamos el mapa sin pose conocida ?, ¿ como obtenermos la pose sin contar con un mapa ?
- Para poder resolver este problema de forma práctica necesitamos obtener el mapa y la pose de forma simultánea

Resumen

- Luego de este largo recorrido por los fundamentos de Robótica Probabilística tenemos
  - Localización con mapa conocido
  - Mapeo con pose conocida
- ¿ Qué fue primero ?, ¿ el huevo o la gallina ?
- ¿ Cómo calculamos el mapa sin pose conocida ?, ¿ como obtenermos la pose sin contar con un mapa ?
- Para poder resolver este problema de forma práctica necesitamos obtener el mapa y la pose de forma simultánea
- En el año 1986, en la IEEE Robotics and Automation Conference, empieza a cobrar interés la idea de resolver ambos problemas de forma simultánea.

Resumen

- Luego de este largo recorrido por los fundamentos de Robótica Probabilística tenemos
  - Localización con mapa conocido
  - Mapeo con pose conocida
- ¿ Qué fue primero ?, ¿ el huevo o la gallina ?
- ¿ Cómo calculamos el mapa sin pose conocida ?, ¿ como obtenermos la pose sin contar con un mapa ?
- Para poder resolver este problema de forma práctica necesitamos obtener el mapa y la pose de forma simultánea
- En el año 1986, en la IEEE Robotics and Automation Conference, empieza a cobrar interés la idea de resolver ambos problemas de forma simultánea.
- En el año 1995, en el International Symposium on Robotics Research, se define la estructura general del problema y se acuña el término:

SLAM: Simultaneous Localization and Mapping

Resumen

- Vimos como generar un mapa métrico suponiendo poses conocidas.
- Siempre las celdas son tratadas en forma independiente, para generar un problema tratable.
- Mapas se suponen estáticos, aunque en la realidad no lo son.
- Cuando, además del mapa, las poses del robot son desconocidas, se agrega incertidumbre en x_t. Esto da lugar a un nuevo problema que requiere de localización y mapeo de forma simultánea.
- En el año 1986, en la IEEE Robotics and Automation Conference, empieza a cobrar interés la idea de resolver ambos problemas de forma simultánea.
- En el año 1995, en el International Symposium on Robotics Research, se define la estructura general del problema y se acuña el término:

SLAM: Simultaneous Localization and Mapping

Bibliografía

- ***Probabilistic Robotics***, Thrun, S., Burgard, W., Fox, D.