Universidad de Alcalá

Grado en Ingeniería Informática
Grado en Ingeniería de Computadores
Grado en Ingeniería en Sistemas de la Información
Grado en Sistemas de la Información

Sistemas de Control para Robots

Tema 1. Localización y mapeado

Elena López Guillén
Manuel Ocaña Miguel
Rafael Barea Navarro

Departamento de Electrónica

Índice del Tema

Universidad de Alcalá

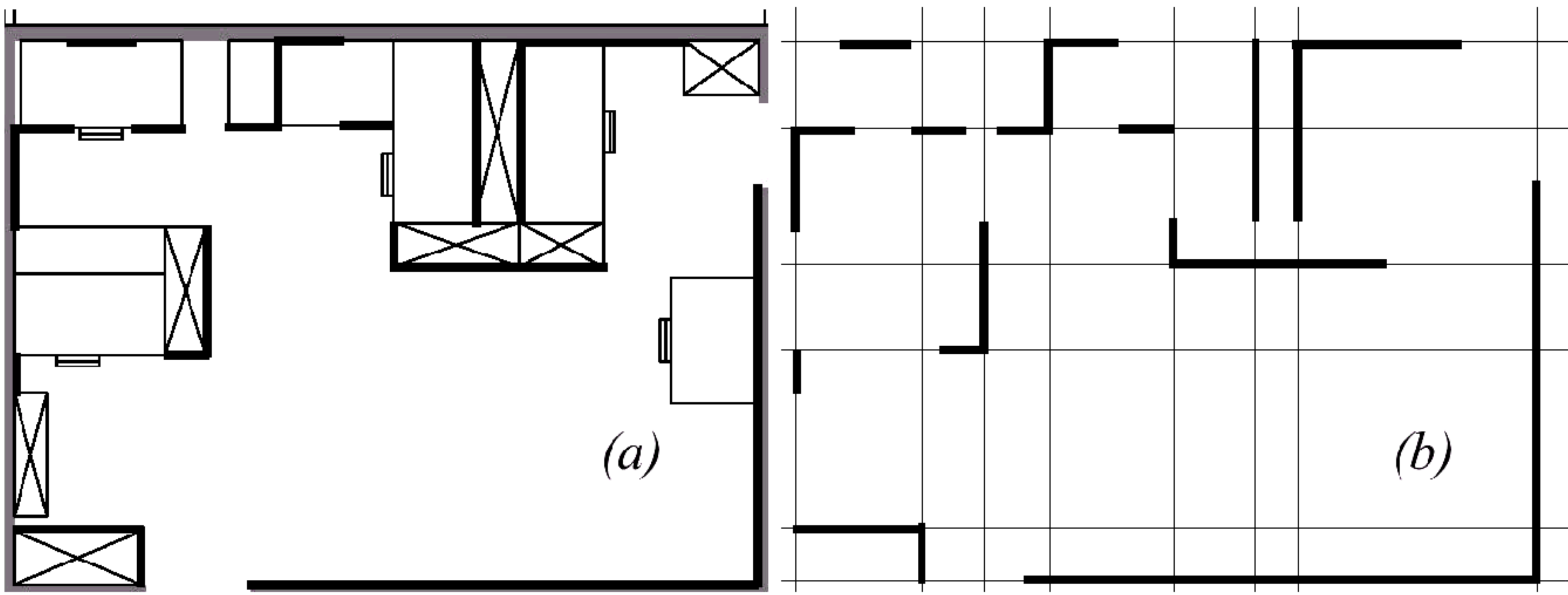
1. Mapeado: representaciones métricas y topológicas

Elección del método de representación

- Elección del método de representación, depende de:
  - La aplicación en cada caso
  - Las características observables
  - La carga computacional
- Tipos de representaciones:
  - Continuas: precisión, gran carga computacional
  - Discretas: simplificación

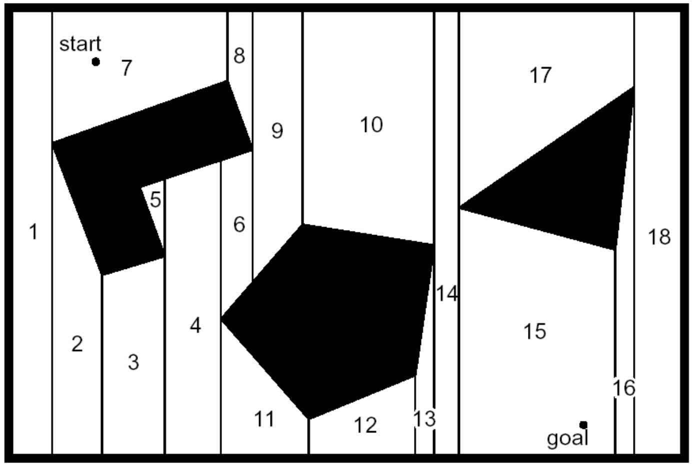
Representación continua

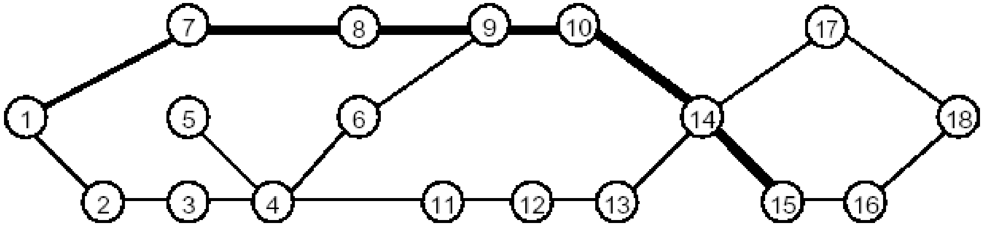
- **MAPAS GEOMÉTRICOS** basados en la arquitectura del edificio
  - Representación con un conjunto de líneas infinitas
  - Requieren una carga computacional elevada



Representaciones discretas (I)

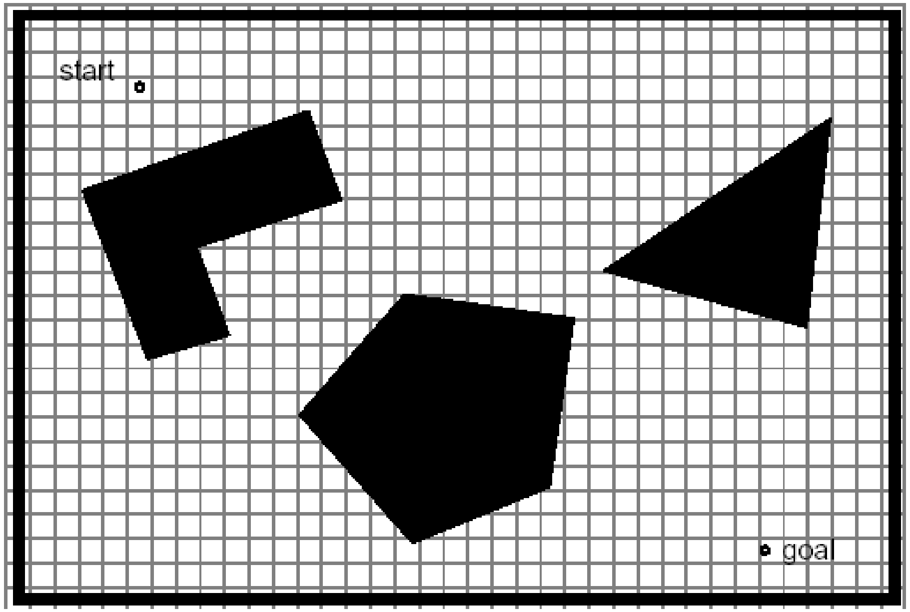
1. Descomposición en **CELDAS EXACTAS**
   - Busca cubrir el espacio no ocupado mediante teselas o polígonos
   - No es importante la posición que ocupa el robot en el área libre sino la capacidad para pasar de una a otra
   - Se almacenan las relaciones entre las diferentes áreas (diagramas de conectividad)

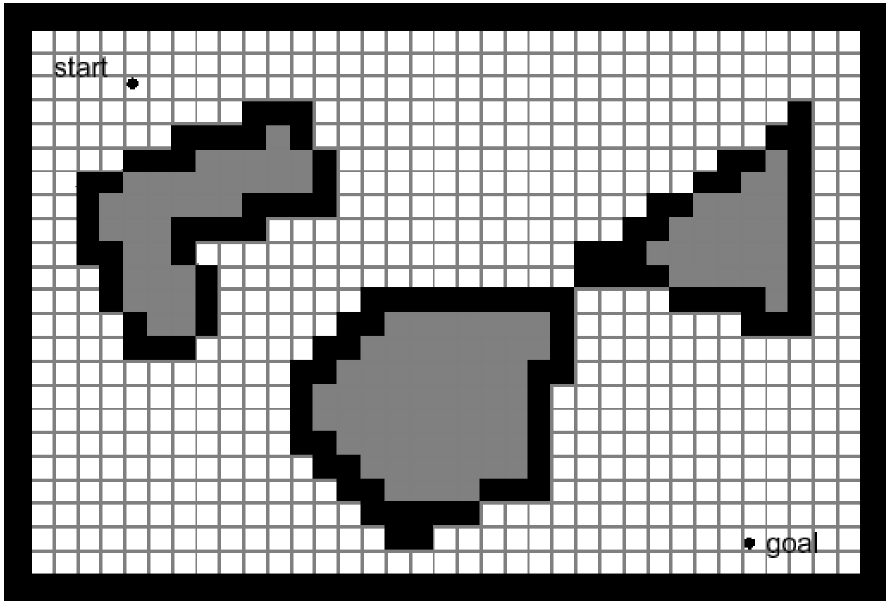




Representaciones discretas (II)

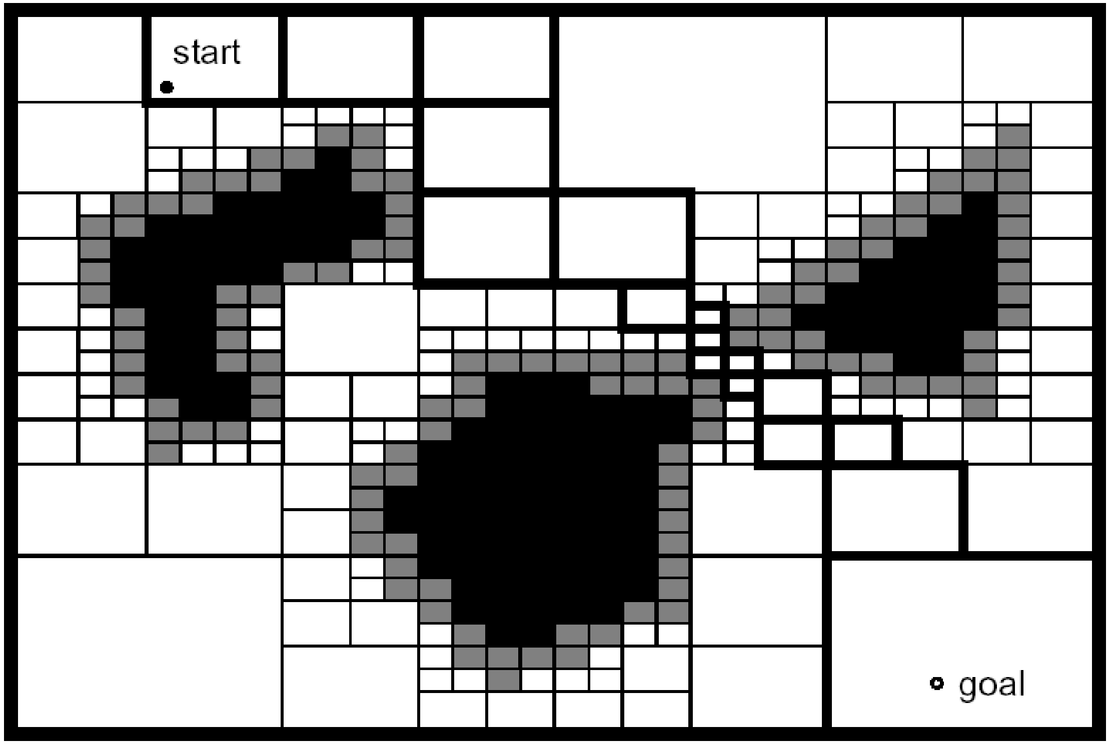
2. Descomposición en **CELDAS FIJAS**
   - Descompone en celdas fijas con dos posibles valores: ocupada (“1”) y libre (“0”)
   - También es posible añadir valor desconocido
   - Problema: desaparecen los pasos estrechos





Representaciones discretas (III)

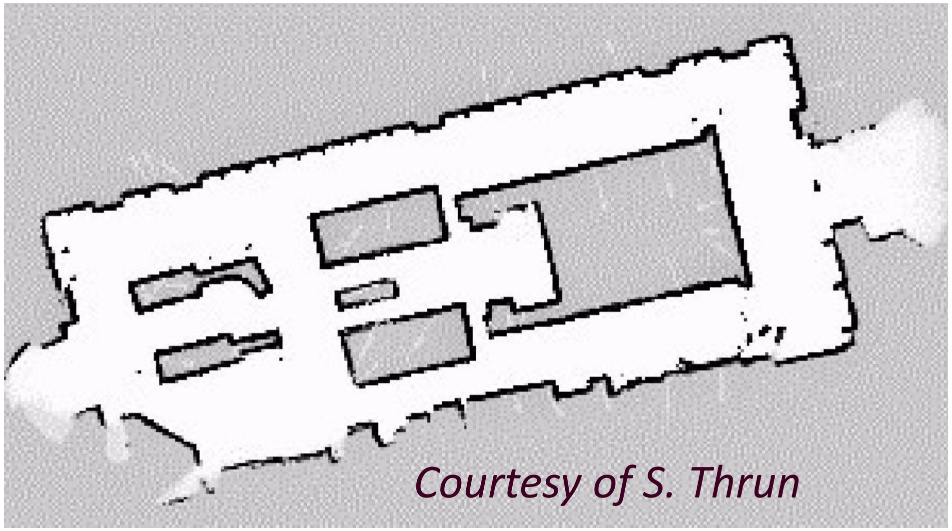
3. Descomposición **ADAPTATIVA** en celdas de tamaño variable
   - Resuelve el problema de los pasos estrechos
   - Se parte de un tamaño fijo que se mantiene en zonas libres
   - Si queda una zona ocupada en la celda se divide en cuatro y así sucesivamente hasta una resolución máxima determinada



Representaciones discretas (IV)

4. **REJILLAS DE OCUPACIÓN** utilizando celdas de un tamaño mínimo

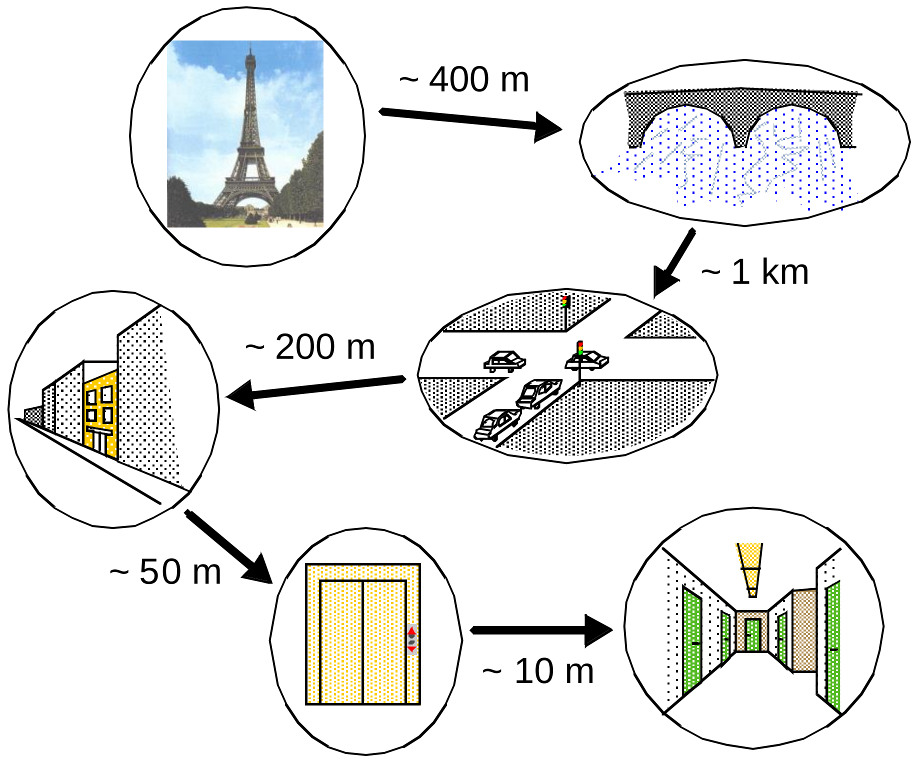
- A cada celda se le asigna un contador. El contador se incrementa con cada impacto de los sensores de distancia y se decrementa con un impacto en una celda que quedaba oculta tras ella.
- Se utilizan normalmente con sensores de distancia láser
- Problema: los mapas crecen a medida que lo hace el entorno


Courtesy of S. Thrun

Representaciones discretas (V)

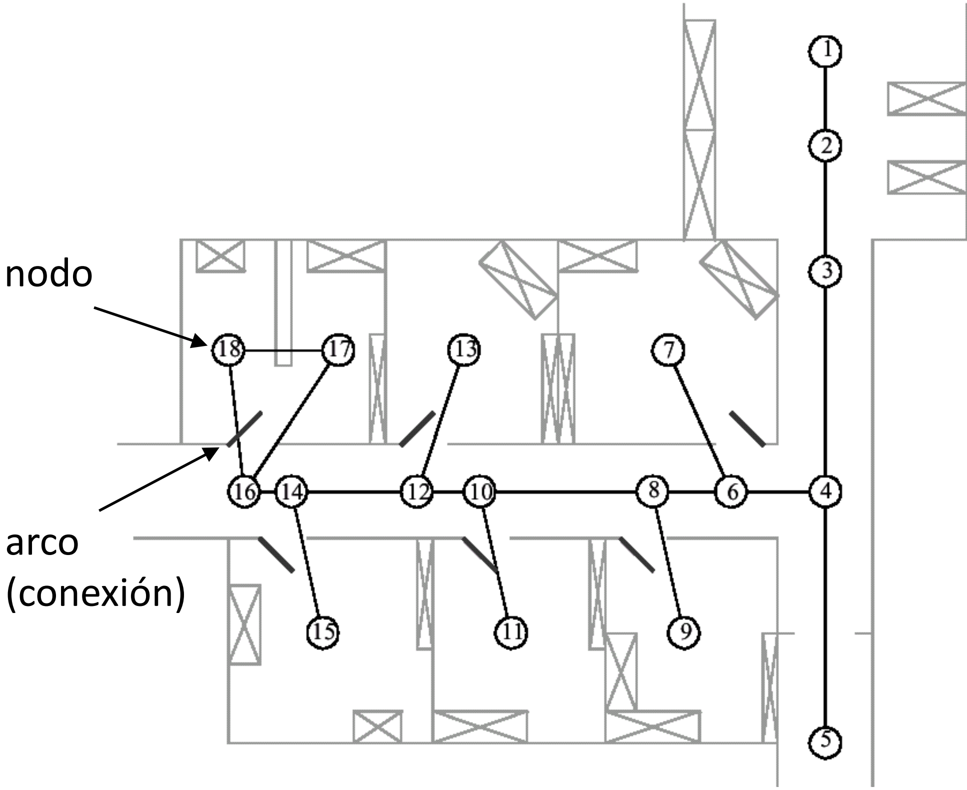
5. REPRESENTACIÓN TOPOLÓGICA

- Evitan las medidas geométricas del entorno.
- Se concentran en características relevantes del entorno que son útiles para los objetivos del robot móvil.



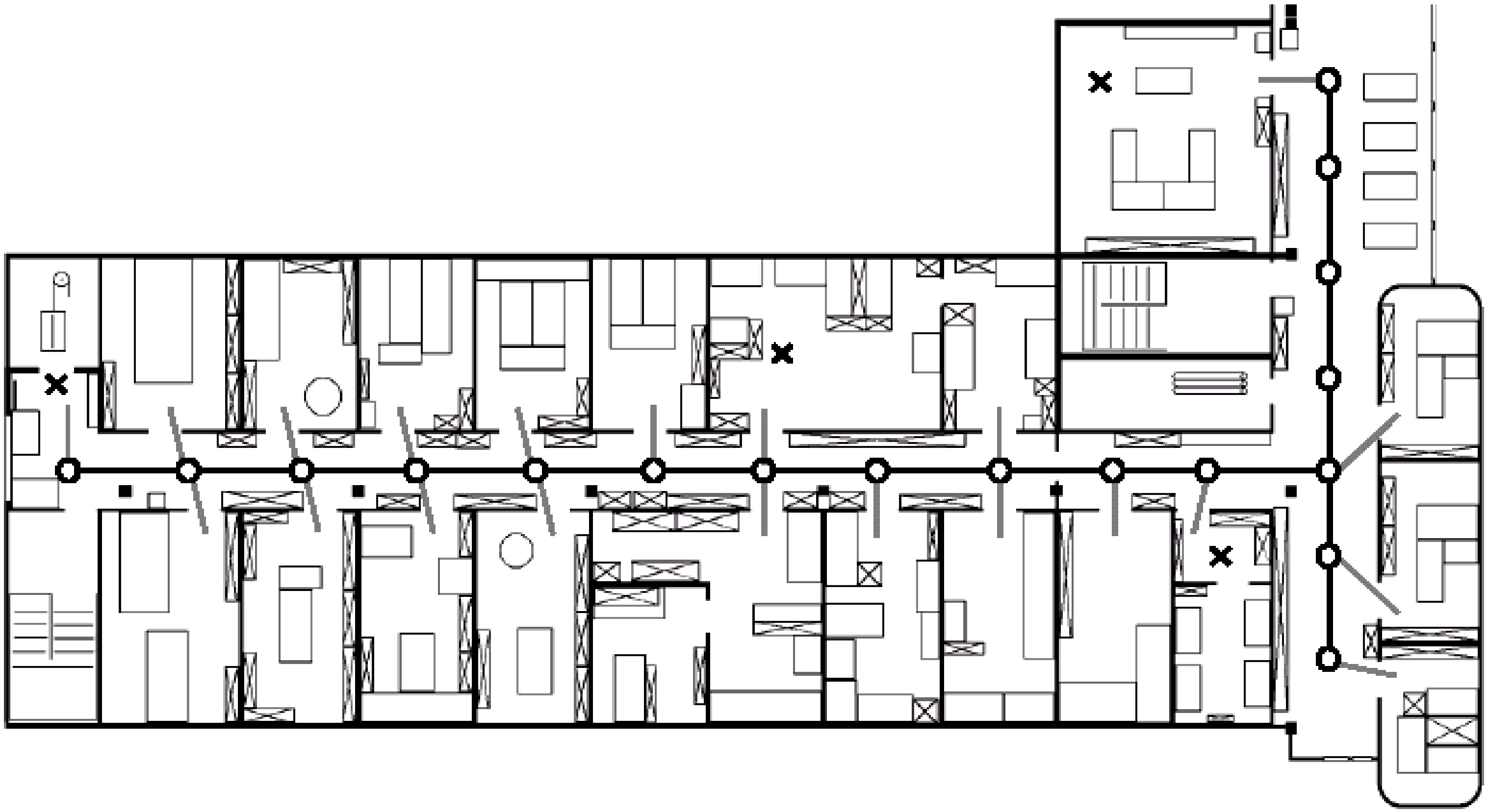
Representaciones discretas (VII)

- **Elementos** de la representación topológica:
  - Nodos y Arcos
    - Nodos: se utilizan para denotar áreas de interés
    - Arcos: se utilizan para indicar la adyacencia de dos nodos.
  - Cuando un arco conecta dos nodos, indica que se puede pasar de un nodo a otro sin necesidad de atravesar otro nodo
  - En la representación topológica los nodos no tienen por qué estar separados un tamaño fijo como en la discretización basada en celdas



Representaciones discretas (VI)

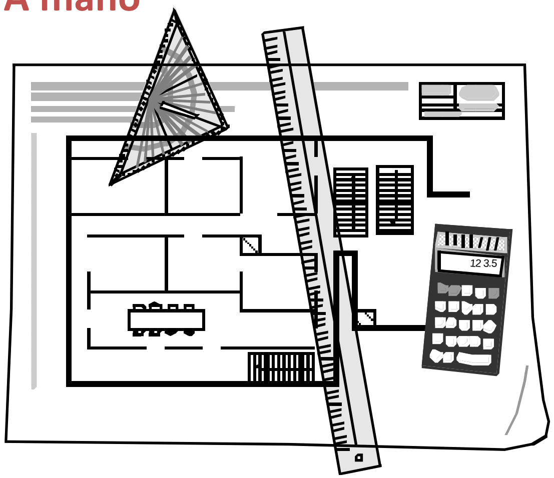
- Ejemplo de representación topológica

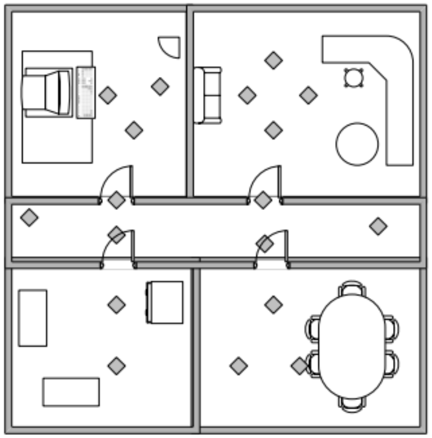


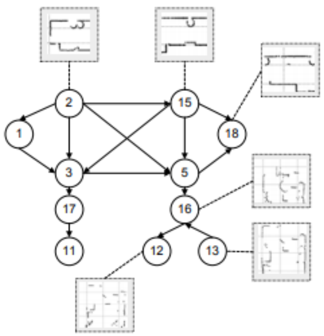
Construcción de los mapas

❑ ¿Quién construye los mapas?

1. A mano





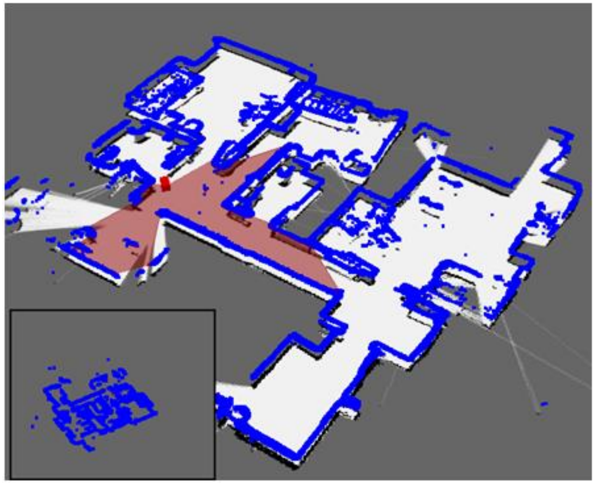


2. Automáticamente:

El robot aprende su entorno (mapeado)

Motivación:

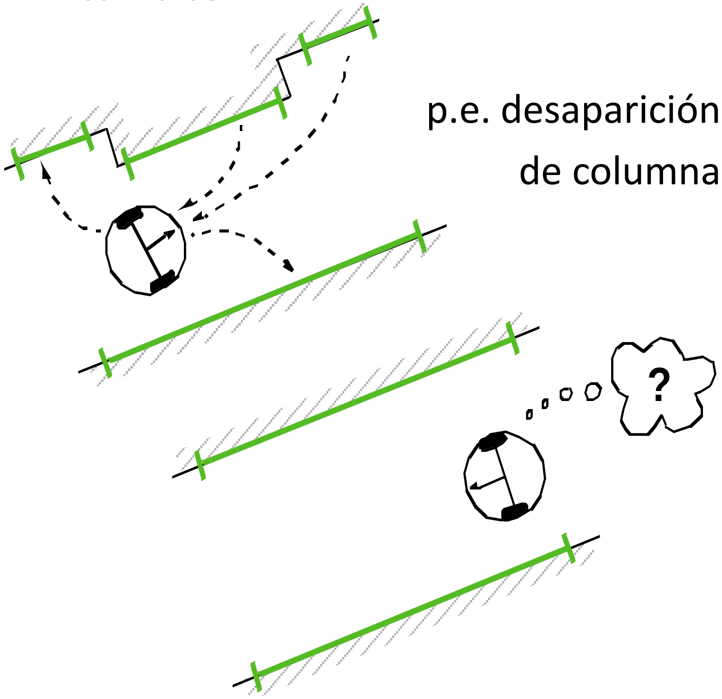
- ❑ A mano: duro y costoso
- ❑ El entorno cambia dinámicamente



Construcción de los mapas

❑ Retos

1. **Mantenimiento del Mapa**: mantener la consistencia del mapa ante cambios

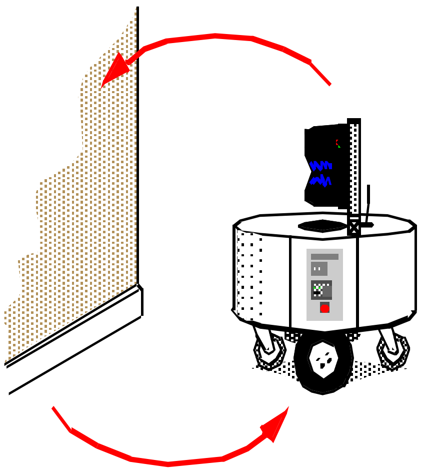


p.e. desaparición de columna

- p.e. medida de creencia de cada una de las características del entorno

2. **Representación y Reducción de la Incertidumbre**

posición del robot -> posición del muro



posición del muro -> posición del robot

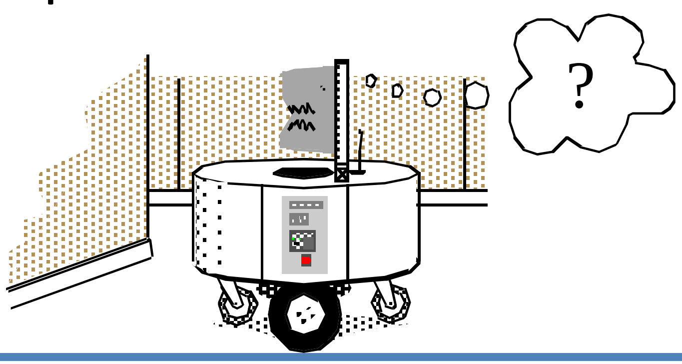
- ❑ Densidad de probabilidad sobre las posiciones de las características
- ❑ Estrategias de exploración adicionales

Universidad de Alcalá

2. Sistemas de localización: local y global

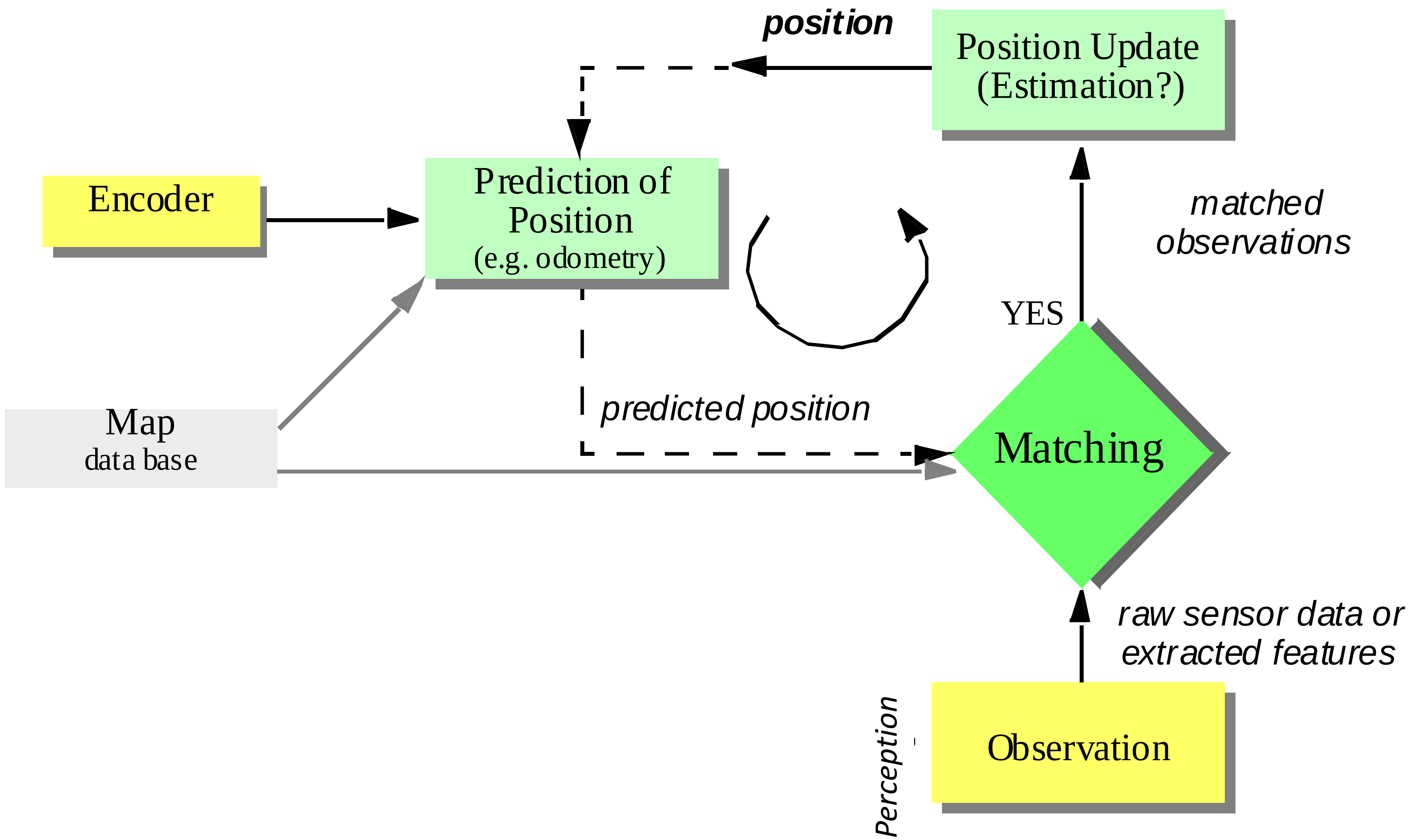
Localización local y global (I)

- **Localización**: proceso por el cual el robot obtiene su posición dentro del entorno en el que se mueve. Se clasifica como:
  - Local: se proporciona al robot la posición inicial de la que parte (p.e. odometría)
  - Global: no es necesario proporcionar información sobre su posición en el comienzo de la navegación (p.e. GPS, WiFi)
- Clasificación (¿quién realiza la localización?):
  - Entorno inteligente: es el entorno el que realiza la localización
  - Propio robot realiza la localización



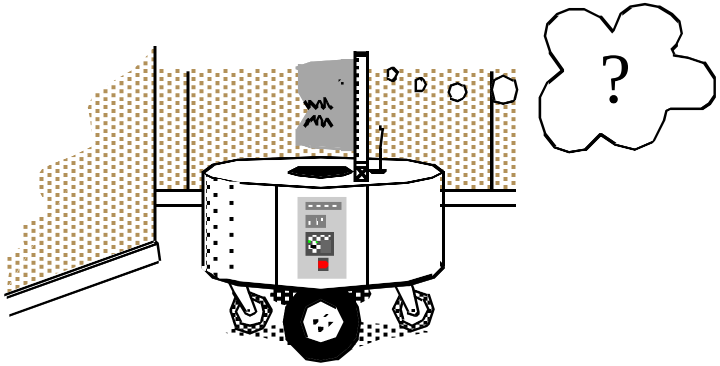
Localización local y global (II)

- El proceso de la localización es iterativo:



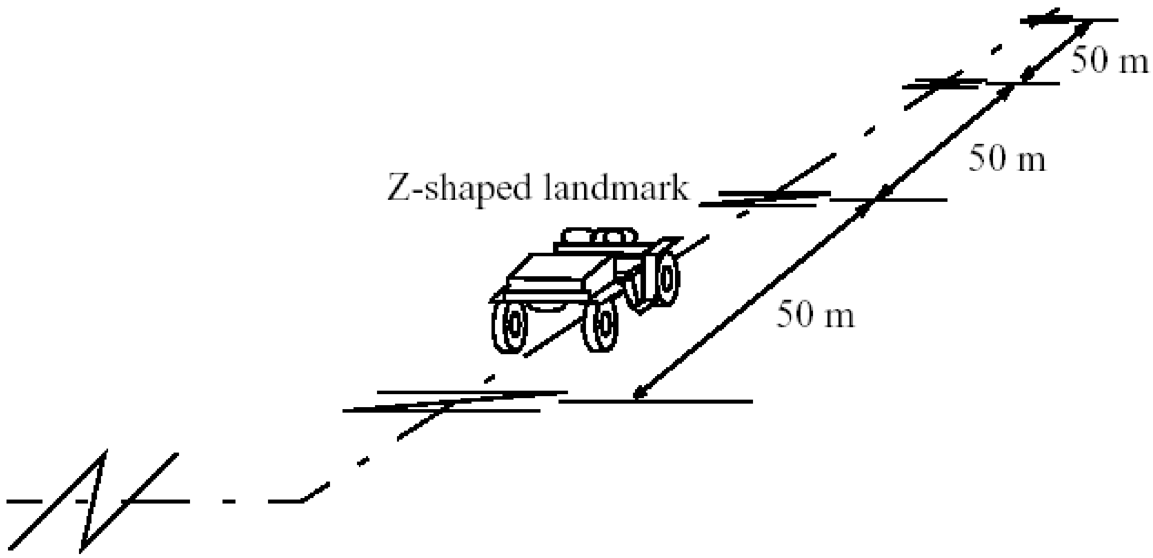
Localización local y global (III)

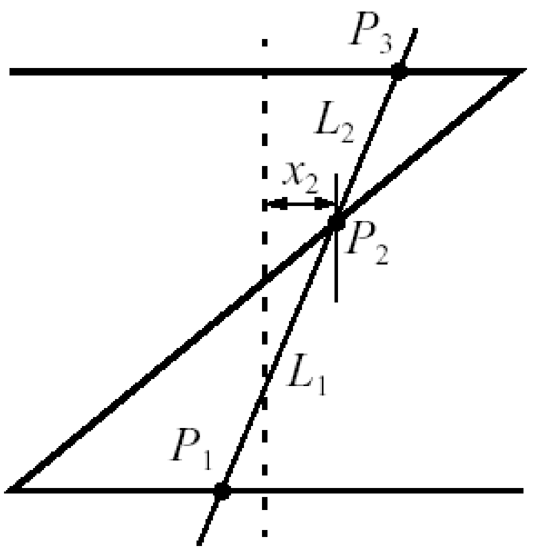
- La localización se puede llevar a cabo mediante diversas técnicas que se clasifican como:
  - Determinísticas: basadas en marcas, balizas, camino, (odometría, dead-reckoning)
  - Probabilísticas: filtros de Kalman, procesos de Markov



Localización basada en marcas (I)

- Se utilizan marcas (artificiales o naturales) en el entorno:
  - Entre marca y marca sólo se utiliza la fase de predicción
  - Cuando se detecta una marca, se utilizan sus propiedades geométricas para corregir la posición





Localización basada en marcas (II)

- Ejemplo: MDARS

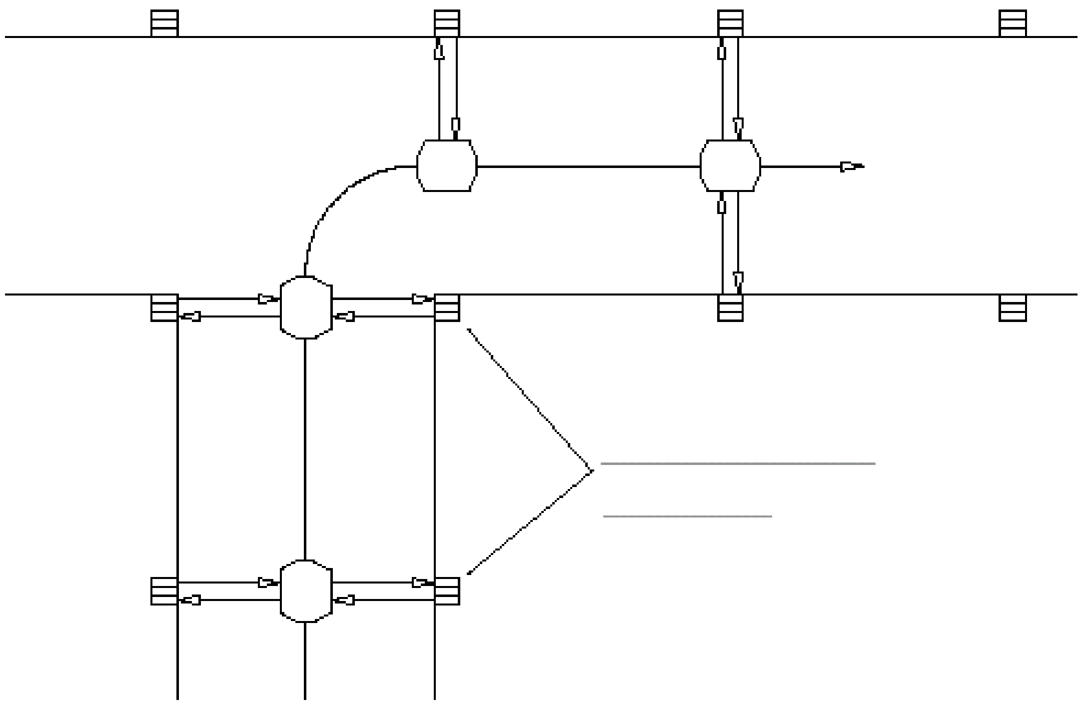


Figure 7.5: Polarized retroreflective proximity sensors are used to locate vertical strips of retroreflective tape attached to shelving support posts in the Camp Elliott warehouse installation of the MDARS security robot [Everett et al, 1994].

Localización basada en balizas (I)

- Se utilizan las posiciones conocidas de diferentes balizas en el entorno para obtener la posición mediante, por ejemplo, el algoritmo de triangulación o trilateración

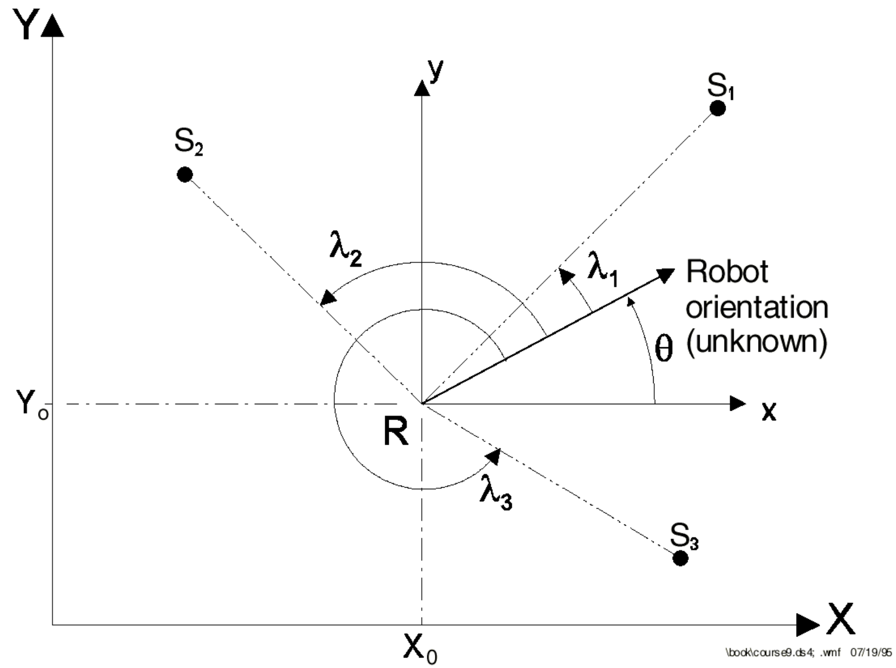


Figure 6.1: The basic triangulation problem: a rotating sensor head measures the three angles λ_1, λ_2, and λ_3 between the vehicle's longitudinal axes and the three sources S_1, S_2, and S_3.

Localización basada en el camino

- Algunos sistemas utilizan estrategias de localización basadas en el camino que debe seguir el robot.
  - En este caso la ruta a seguir por el robot está explícitamente marcada sobre el entorno
  - El robot obtiene su posición global sobre el entorno por medio de conocer su posición sobre la ruta
  - Existen diferentes estrategias:
    - Marcar el camino completo
    - Marcar las intersecciones
  - Normalmente se emplea pintura ultravioleta o marcas magnéticas sobre el entorno.

Universidad de Alcalá

3. Introducción al problema del SLAM

(localización y mapeado simultáneos)

El Problema del SLAM

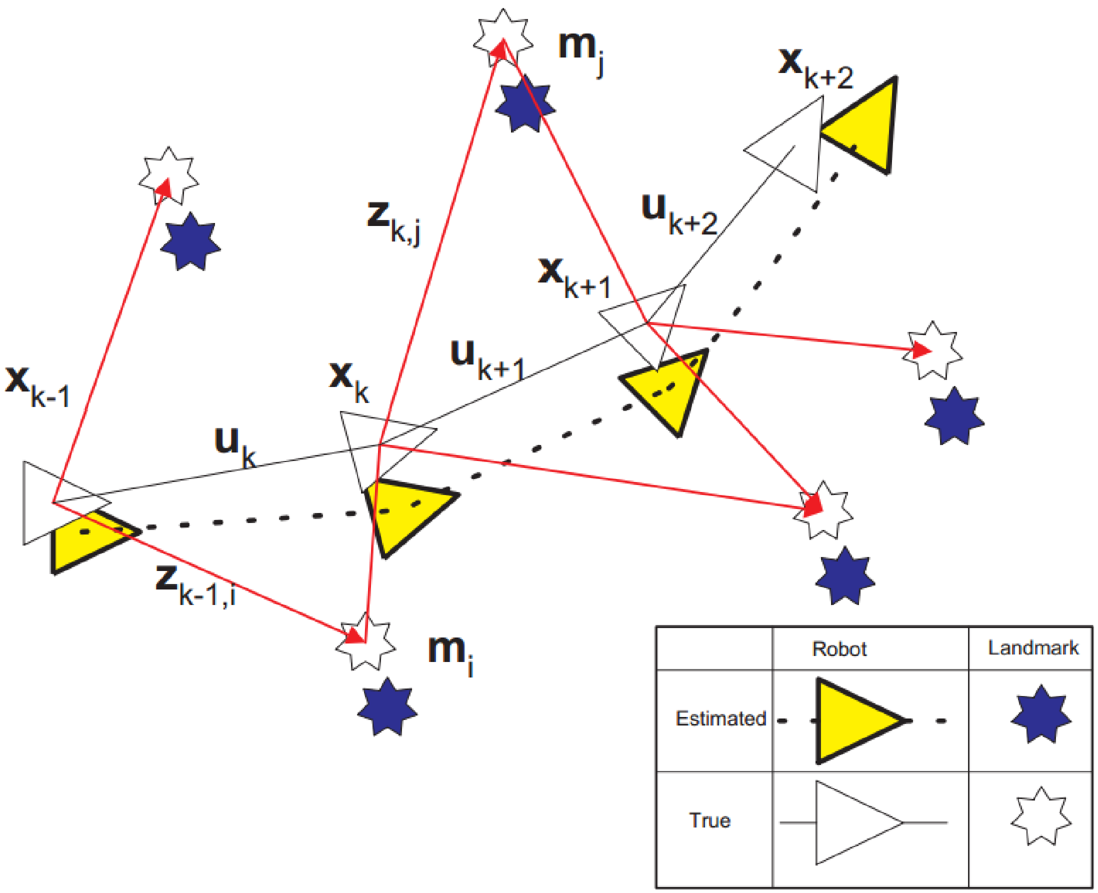
- Comenzando en un punto cualquiera del entorno el robot debería ser capaz de explorar autónomamente el entorno utilizando sus sensores y debería construir el mapa a la vez que se localiza sobre él.

SLAM

"Simultaneous Localization And Mapping"

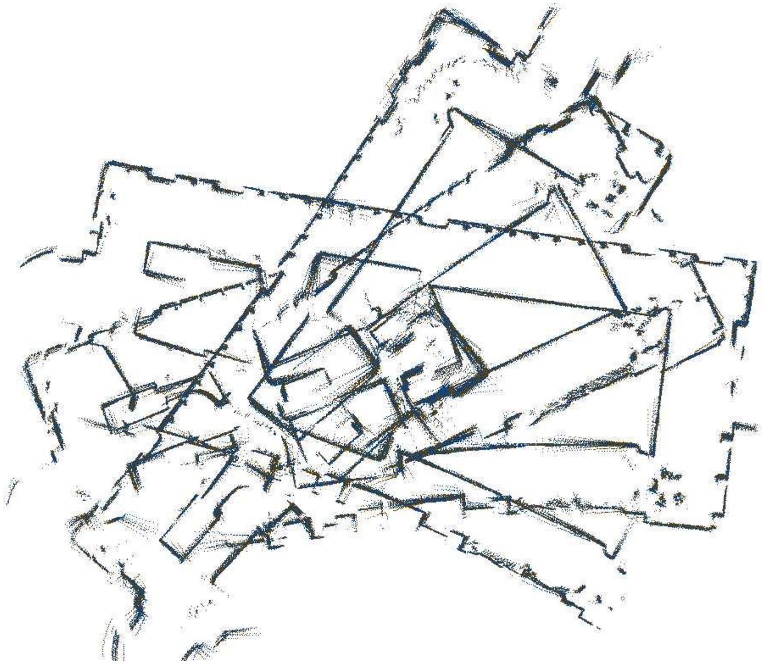
El Problema del SLAM

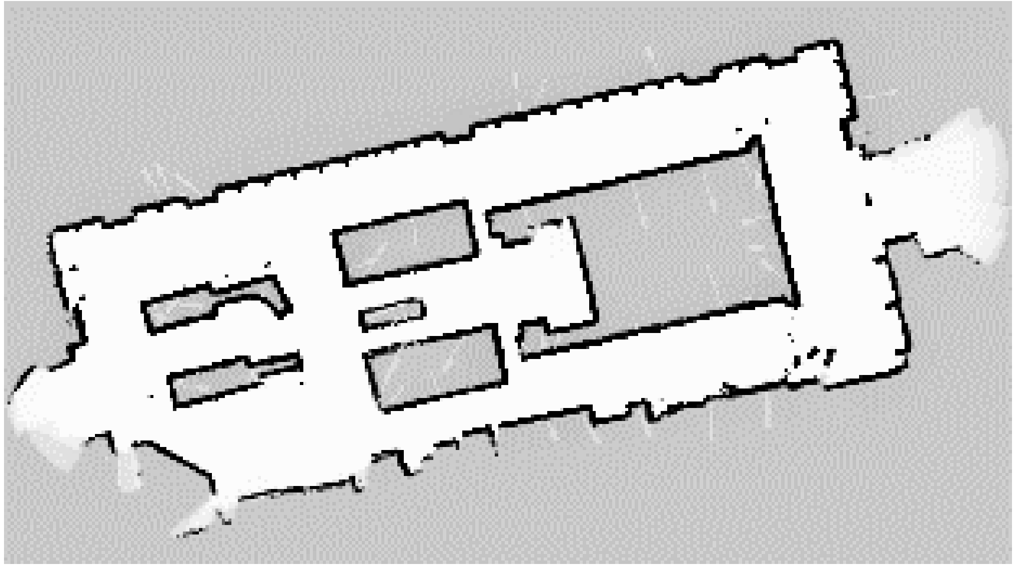
El error cometido en la localización se traslada al mapa…
Y los errores del mapa producen errores de localización…



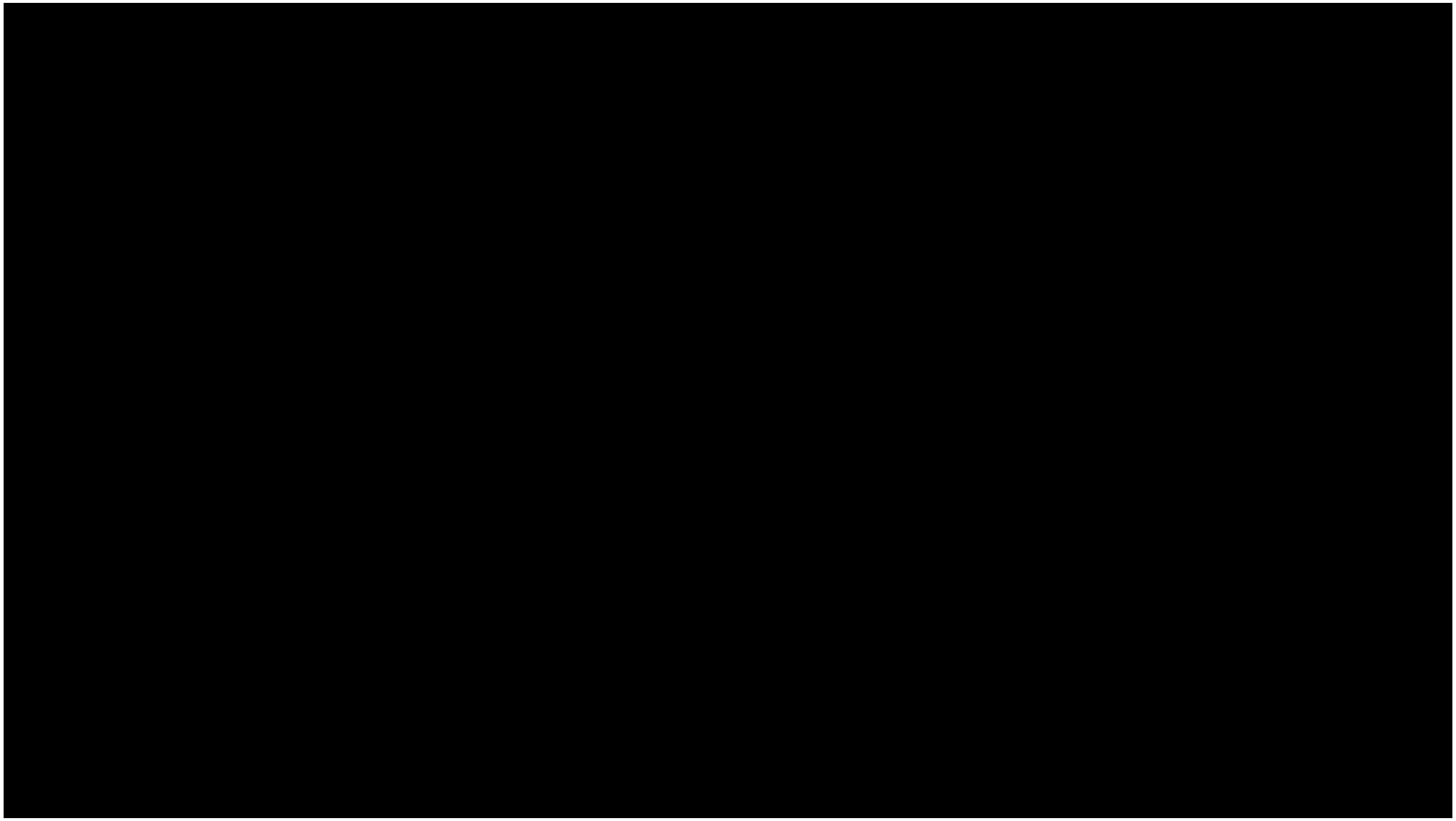
El Problema del SLAM

- **Cierre de lazos**:
  - Pequeños errores se acumulan produciendo graves errores globales, sobre todo en los cierres de lazos
  - Normalmente no suele dar problemas en la navegación local, pero si a la hora de construir el mapa

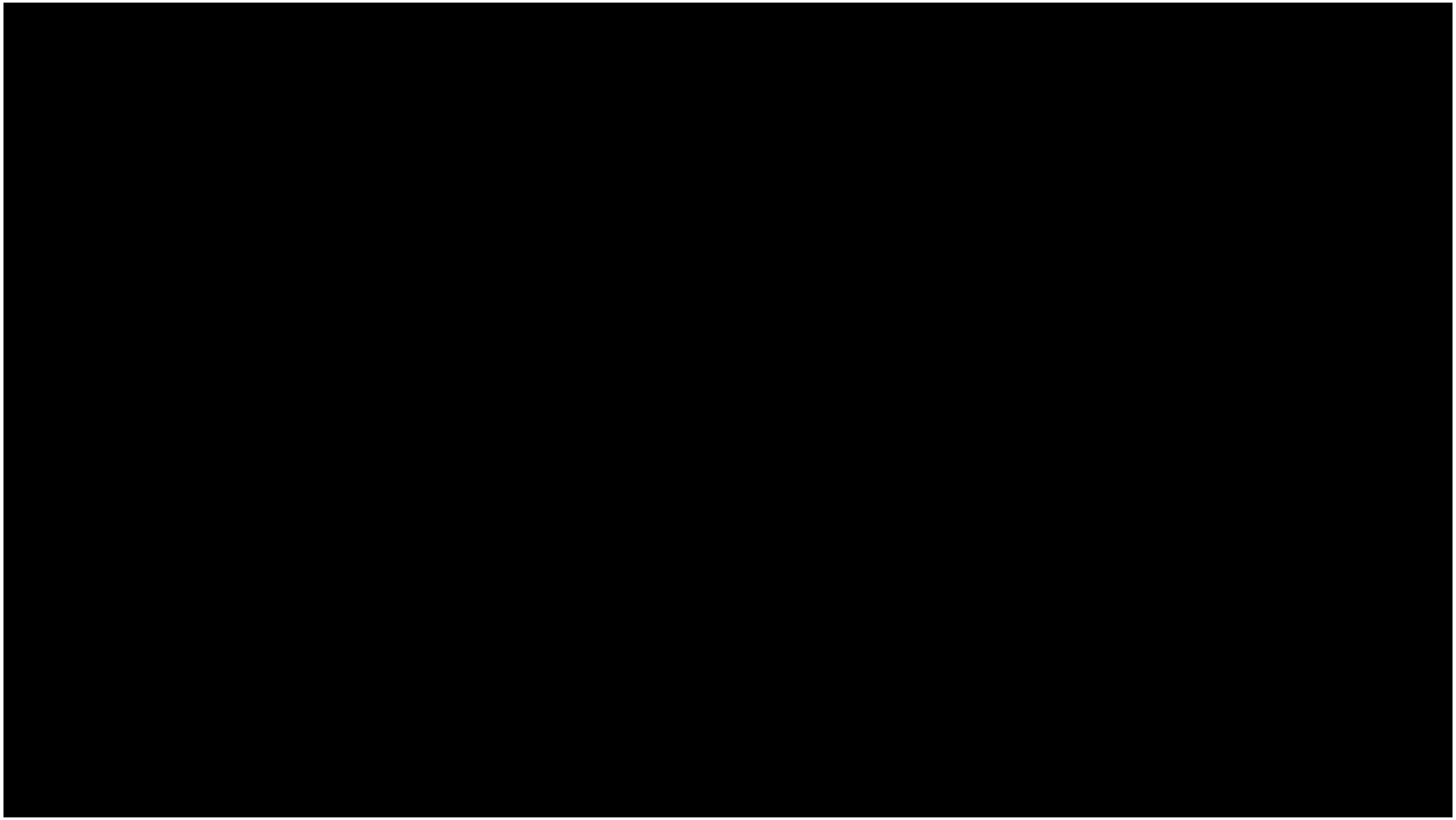




El Problema del SLAM



El Problema del SLAM



Universidad de Alcalá

4. Localización y mapeado mediante estimación bayesiana

Índice

1.1 Incertidumbres asociadas a los sistemas robóticos

Cuatro fuentes principales de incertidumbre:

- **El entorno**
  Entornos parcial o totalmente desconocidos, dinámicos e impredecibles (¿personas?)
- **El robot**
  Sistemas de actuación imprecisos
- **Los sensores**
  Sensores con información limitada y ruidosa
- **Los modelos**
  Modelos inexactos

¿Cómo conseguir sistemas robóticos robustos?

1.2 ¿Qué son los métodos probabilísticos?

MÉTODOS PROBABILÍSTICOS: Introducen información sobre las ambigüedades debidas a los modelos y a los sensores, trabajando con distribuciones de probabilidad, en lugar de con datos exactos.

Aplicación en la resolución de problemas clásicos de robótica:

1. PERCEPCIÓN - estimación de estados (LOCALIZACIÓN Y MAPEADO)
2. TOMA DE DECISIONES – optimización de las acciones (CONTROL Y PLANIFICACIÓN)

Ventajas:

- No requieren modelos exactos ni precisión en los sensores.
- Mayor robustez ante ruidos y errores de medida (aplicaciones reales).
- Posibilidad de recuperación ante fallos

Inconvenientes:

- Complejidad desde el punto de vista computacional.
- Necesidad de realizar diferentes tipos de aproximaciones
- Necesidad de discretizar el estado del robot y del entorno.

1.3 Fundamentos: teoría de la probabilidad

1.3.1. Axiomas básicos de probabilidad

p(a) representa la probabilidad de que el suceso a sea cierto:

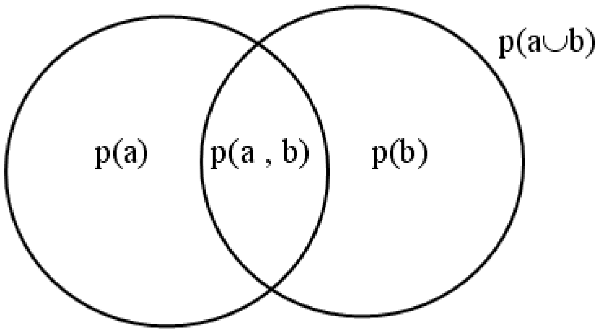
- Si p(a)=1, entonces a es "verdadero"
- Si p(a)=0, entonces a es "falso"

$$\boxed{0 \leq p(a) \leq 1}$$

UNIÓN DE SUCESOS: $p(a \cup b)$ probabilidad de a o b ciertos

INTERSECCIÓN DE SUCESOS: $p(a,b)$ probabilidad de a y b ciertos

Axioma: $p(a \cup b) = p(a) + p(b) - p(a,b)$

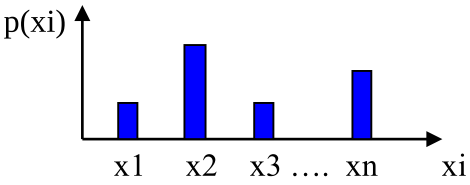


Axioma: Si a y b son independientes, $p(a,b) = p(a) \cdot p(b)$

1.3.2. Variables aleatorias continuas y discretas

VARIABLE ALEATORIA DISCRETA **X**: sólo toma conjunto finito de valores $\{x_1, x_2, \ldots, x_n\}$

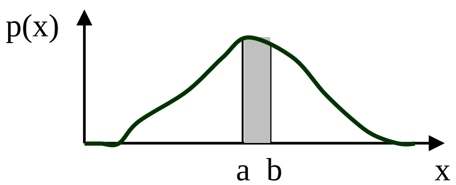
- P(**X**=x_1) o p(x_1) es la probabilidad de que **X** tome el valor x_1
- Distribución de probabilidad:



$$\sum_{xi} p(xi) = 1$$

VARIABLE ALEATORIA CONTINUA **X**: toma cualquier valor x dentro de un rango continuo

- P(**X**=x) o p(x) siempre tiende a cero
- Función densidad de probabilidad:



$$p(x \in [a,b]) = \int_a^b p(x) \cdot dx$$

$$\int_{-\infty}^{\infty} p(x) \cdot dx = \int_x p(x) \cdot dx = 1$$

1.3.3. Probabilidad condicional

- p(a|b) $\Rightarrow$ probabilidad de que suceda "a" habiendo sucedido "b"
- p(a|b,c) $\Rightarrow$ probabilidad de que suceda "a" habiendo sucedido "b" y "c"

1.3.4. Teorema de la probabilidad total

Supóngase que el suceso *a* puede ocurrir en condiciones de aparición de uno de los sucesos mutuamente excluyentes b_1, b_2,..., b_n, que forman un grupo completo:

$$p(a)=p(b_1)\cdot p(a|b_1)+p(b_2)\cdot p(a|b_2)+\ldots+p(b_n)\cdot p(a|b_n)=\sum_{i=1}^{n} p(b_i)\cdot p(a|b_i)$$

(versión discreta)

$$p(a)=\int_b p(b)\cdot p(a|b)\cdot db$$

(versión continua)

1.3.5. Teorema de Bayes (probabilidad a posteriori)

Supóngase que el suceso *a* puede ocurrir en condiciones de aparición de uno de los sucesos mutuamente excluyentes b_1, b_2,... b_n: *HIPÓTESIS*.

Teorema de Bayes: permite estimar las probabilidades de las hipótesis después de conocer el resultado de la experimentación, debido a la cual ocurrió el suceso a:

$$p(b_i \mid a) = \frac{p(a \mid b_i) \cdot p(b_i)}{p(a)}$$

- $p(bi|a) \Rightarrow$ *probabilidad a posteriori* de la hipótesis b_i, habiendo sucedido *a*.
- $p(a|bi) \Rightarrow$ *verosimilitud* (probabilidad condicional) del suceso *a* bajo la hipótesis b_i.
- $p(bi) \Rightarrow$ *probabilidad a priori* del suceso b_i.
- $p(a) \Rightarrow$ *probabilidad total* (o *evidencia*) de que suceda *a*

1.3.6. Condicionalidad en los teoremas de probabilidad total y de Bayes

Sean a, b y c tres sucesos dependientes.

- Teorema de la probabilidad total. Se desea calcular la probabilidad total de que suceda a, habiendo sucedido b, y sabiendo que ello puede ocurrir en condiciones de aparición de uno de los sucesos mutuamente excluyentes c_1, c_2, c_3, … , c_n.

Versión discreta:

$$p(a \mid b) = \sum_{i=1}^{n} p(ci \mid b) \cdot p(a \mid b, ci)$$

Versión continua:

$$p(a \mid b) = \int p(c \mid b) \cdot p(a \mid b, c) \cdot dc$$

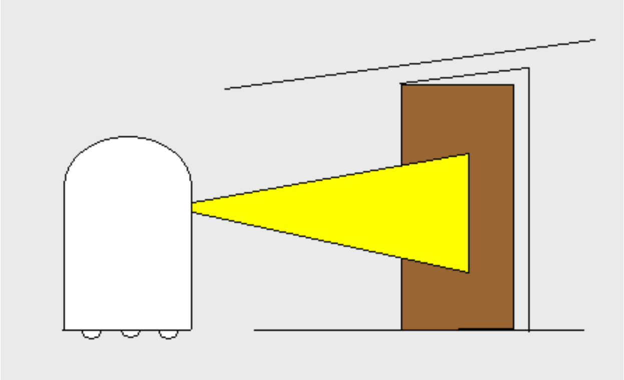
- Teorema de Bayes. Se desea calcular la probabilidad a posteriori de c, habiendo sucedido b y a.

$$p(c \mid b, a) = \frac{p(a \mid c, b) \cdot p(c \mid b)}{p(a \mid b)} \quad \text{o} \quad p(c \mid b, a) = \frac{p(b \mid c, a) \cdot p(c \mid a)}{p(b \mid a)}$$

Índice del Bloque

1. Introducción: los métodos probabilísticos en robótica
2. **Estimación de estados mediante filtros bayesianos**
   - **2.1. Planteamiento de estimación de estados: ejemplo práctico**
   - **2.2. Filtros bayesianos**
   - **2.3. Ejemplo de aplicación de un filtro de Bayes a la localización de un robot**
3. Aplicación a la localización de robots móviles
4. Aplicación a la obtención de mapas
5. Aplicación al problema del SLAM

2.1 Planteamiento de estimación de estados: ejemplo práctico



Supóngase un robot capaz de obtener una medida O que puede tomar valores de un conjunto definido $\{o_1, o_2,\ldots,o_n\}$

OBJETIVO: **estimar el estado S**={abierta, cerrada} de una puerta.

2.1.1. Planteamiento del problema de estimación del estado de la puerta

- $p(abierta|o_1)$ es el OBJETIVO DE ESTIMACIÓN (diagnóstico), difícil de obtener
- $p(o_1|abierta)$ es una verosimilitud (probabilidad causal), fácil de obtener
- $p(abierta)$ es una probabilidad a priori, que se conoce de antemano

SOLUCIÓN: **Teorema de Bayes**

$$p(abierta|o_1) = \frac{p(o_1|abierta) \cdot p(abierta)}{p(o_1)}$$

Ejemplo 1: Estimación tras la realización de una primera medida

Se realiza una primera medida de valor $O=o_1$.

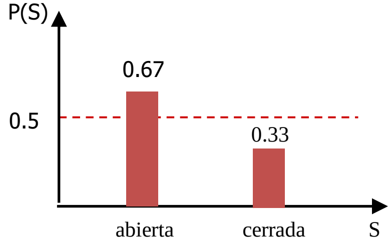
DATOS: Verosimilitudes: $p(o_1|abierta)=0.6$ $p(o_1|cerrada)=0.3$

Probabilidades a priori: $p(abierta)=p(cerrada)=0.5$

ESTIMACIÓN (PROBABILIDAD A POSTERIORI) del estado de la puerta:

$$p(abierta|o_1) = \frac{p(o_1|abierta)\,p(abierta)}{p(o_1)} = \frac{0.6\cdot 0.5}{0.6\cdot 0.5 + 0.3\cdot 0.5} = 0.67$$

$$p(cerrada|o_1) = \frac{p(o_1|cerrada)\,p(cerrada)}{p(o_1)} = \frac{0.3\cdot 0.5}{0.6\cdot 0.5 + 0.3\cdot 0.5} = 0.33$$



La primera medida ha INCREMENTADO la probabilidad de que la puerta esté abierta

Ejemplo 2: Estimación tras la realización de una segunda medida

Se realiza una segunda medida de valor $O=o_2$.

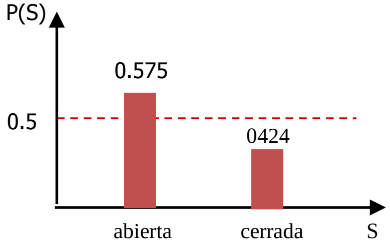
DATOS: Verosimilitudes: $p(o_2|abierta)=0.4$ $p(o_2|cerrada)=0.6$

Probabilidades a priori: $p(abierta|o_1)=0.67$ $p(cerrada|o_1)=0.33$

ESTIMACIÓN (PROBABILIDAD A POSTERIORI) del estado de la puerta:

$$p(abierta|o_1,o_2)=\frac{p(o_2|abierta)\cdot p(abierta|o_1)}{p(o_2|o_1)}=$$

$$=\frac{p(o_2|abierta)\cdot p(abierta|o_1)}{p(o_2|abierta)\cdot p(abierta|o_1)+p(o_2|cerrada)\cdot p(cerrada|o_1)}=\frac{0.4\cdot 0.67}{0.4\cdot 0.67+0.6\cdot 0.33}=0.575$$



La segunda medida ha REDUCIDO la probabilidad de que la puerta esté abierta

2.1.2. Estimación tras la obtención de la n-ésima medida

Se realizan sucesivamente medidas para mejorar la estimación. ¿Cómo se integra la n-ésima medida $O=o_n$?

ACTUALIZACIÓN RECURSIVA DE LA FÓRMULA DE BAYES:

$$p(abierta|o_1,o_2,\ldots o_n) = \frac{p(o_n\,|abierta,o_1,o_2,\ldots o_{n-1})\cdot p(abierta|o_1,o_2,\ldots o_{n-1})}{p(o_n\,|o_1,o_2,\ldots o_{n-1})}$$

PROPIEDAD DE MARKOV:

Según esta condición, o_n no depende de las medidas previas SI SE CONOCE el estado, es decir: $p(o_n|abierta,o_1,o_2,\ldots,o_{n-1})=p(o_n|abierta)$

Y la regla de Bayes queda:

$$p(abierta|o_1,o_2,\ldots o_n) = \frac{p(o_n\,|abierta)\cdot p(abierta|o_1,o_2,\ldots o_{n-1})}{p(o_n)}$$

2.1.3. Aspectos prácticos: NORMALIZACIÓN DEL ESTADO

Tras cualquier actualización del estado, las nuevas probabilidades de la distribución de estado deben sumar siempre 1 $\Rightarrow$ SE SUSTITUYE EL DENOMINADOR DE LA FÓRMULA DE BAYES POR UN FACTOR DE NORMALIZACIÓN:

$$p(abierta|o_1,...,o_n) = \eta \cdot p(o_n\,|abierta) \cdot p(abierta|o_1,....,o_{n-1})$$
$$p(cerrada|o_1,...,o_n) = \eta \cdot p(o_n\,|cerrada) \cdot p(cerrada|o_1,....,o_{n-1})$$

ALGORITMO DE NORMALIZACIÓN

Llamando s ($s \in S$) a los estados a estimar tras la realización de una medida z ($z \in Z$), la estimación de las nuevas probabilidades a posteriori de los estados se calcula del siguiente modo:

$\forall s$: $aux(s|z)=p(z|s) \cdot p(s)$

$$\eta = \frac{1}{\sum_s aux(s\,|\,z)}$$

$\forall s$: $p(s|z)=\eta \cdot aux(s|z)$

2.1.4. Conclusiones

✓ La regla de Bayes permite calcular probabilidades que de otro modo son difíciles de obtener.

✓ Bajo el supuesto de Markov, la actualización recursiva de la fórmula de Bayes permite integrar eficientemente múltiples condiciones para la estimación del estado.

2.2 Filtros bayesianos

2.2.1. Planteamiento del problema

Supóngase un sistema para el cual se definen:

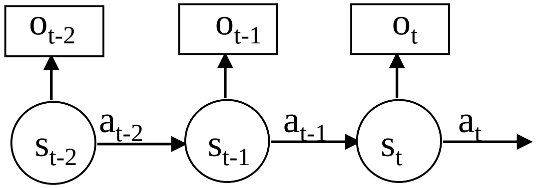
- Un conjunto de estados $s \in S$ que el sistema puede adoptar
- Un conjunto de acciones $a \in A$ que hacen pasar al sistema de unos estados a otros
- Un conjunto de observaciones $o \in O$ que es posible obtener en los diferentes estados del sistema

Se conocen además dos modelos probabilísticos del sistema:

- $p(s'|s,a)$ es el modelo de actuación que caracteriza las incertidumbres de las acciones
- $p(o|s)$ es el modelo de percepción que caracteriza las incertidumbres de las observaciones

Se supone también que el sistema es dinámico, de tal manera que a lo largo del tiempo se van sucediendo las acciones y observaciones.

$d_t=\{o_1,a_1,o_2,a_2,\ldots,o_{t-1},a_{t-1},o_t\}$



2.2.2. Objetivo de un Filtro Bayesiano. Concepto de Distribución de Creencia

OBJETIVO: estimar el estado del sistema en cada instante de tiempo t, manteniendo para ello una distribución de probabilidad sobre la variable S.

Esta distribución se conoce como DISTRIBUCIÓN DE CREENCIA (o creencia) $Bel_t(S)$ y caracteriza, para cada valor particular del estado s, su probabilidad de ser el estado real del sistema.

Inicialización de $Bel_1(S)$

- Distribución uniforme si el estado inicial es desconocido
- Distribución delta si el estado inicial es conocido

2.2.3. Propiedad de Markov

Para que sea posible aplicar un filtro de Bayes, debe cumplirse la propiedad de Markov:

$$\boxed{p(s_{t+1} | s_t, d_t) = p(s_{t+1} | s_t, a_t)}$$

Esta propiedad equivale a decir que "conocer el estado actual (presente), hace que el futuro sea independiente del pasado".

O lo que es lo mismo, "toda la historia pasada del sistema queda resumida en el estado actual que, junto con la acción actual realizada, son suficientes para estimar el siguiente estado".

2.2.4. Formulación de un filtro de Bayes

$$\boxed{Bel(s_t)} = p(s_t \mid d_t) = p(s_t \mid o_1, a_1, o_2, \ldots, a_{t-1}, o_t) =$$

R. Bayes
$$= \eta \cdot p(o_t \mid s_t, o_1, a_1, \ldots, a_{t-1}) \cdot p(s_t \mid o_1, a_1, \ldots, a_{t-1}) =$$

P. Markov
$$= \eta \cdot p(o_t \mid s_t) \cdot p(s_t \mid o_1, a_1, \ldots, a_{t-1}) =$$

T. Prob. Total.
$$= \eta \cdot p(o_t \mid s_t) \int p(s_t \mid o_1, a_1, \ldots a_{t-1}, s_{t-1}) \cdot p(s_{t-1} \mid o_1, a_1, \ldots, a_{t-1}) \cdot ds_{t-1} =$$

P. Markov
$$= \eta \cdot p(o_t \mid s_t) \int p(s_t \mid s_{t-1}, a_{t-1}) \cdot p(s_{t-1} \mid o_1, a_1, \ldots, a_{t-1}) \cdot ds_{t-1} =$$

$$\boxed{= \eta \cdot p(o_t \mid s_t) \int p(s_t \mid s_{t-1}, a_{t-1}) \cdot Bel(s_{t-1}) \cdot ds_{t-1}}$$

Modelo de percepción

p(o|s)

Modelo de actuación

p(s'|s,a)

Esta es la fórmula compacta de un filtro de Bayes

2. Estimación de estados mediante filtros bayesianos

Un filtro de Bayes también puede aplicarse en dos etapas:

1. **Etapa de predicción**, tras la ejecución de una acción a en el tiempo t-1:

$$\underbrace{Bel_t(s') = \int_S p(s' \mid s, a) \cdot Bel_{t-1}(s)}_{}$$

Modelo de actuación

2. **Etapa de estimación**, tras la obtención de una observación o en el nuevo estado

$$Bel_{posterior}(s) = \eta \cdot p(o \mid s) \cdot Bel_{anterior}(s)$$

Modelo de percepción

2.2.5. Programación de un filtro de Bayes

ALGORITMO FILTRO DE BAYES
El siguiente algoritmo permite actualizar la distribución de creencia Bel′(S) tras realizar una acción a y obtener una observación o en el nuevo estado, conocida la distribución anterior Bel(S)

$\eta = 0$

Para todo s′ hacer: $Bel'(s') = \int p(s' \mid s, a) \cdot Bel(s) \cdot ds$ Predicción

Para todo s hacer: {

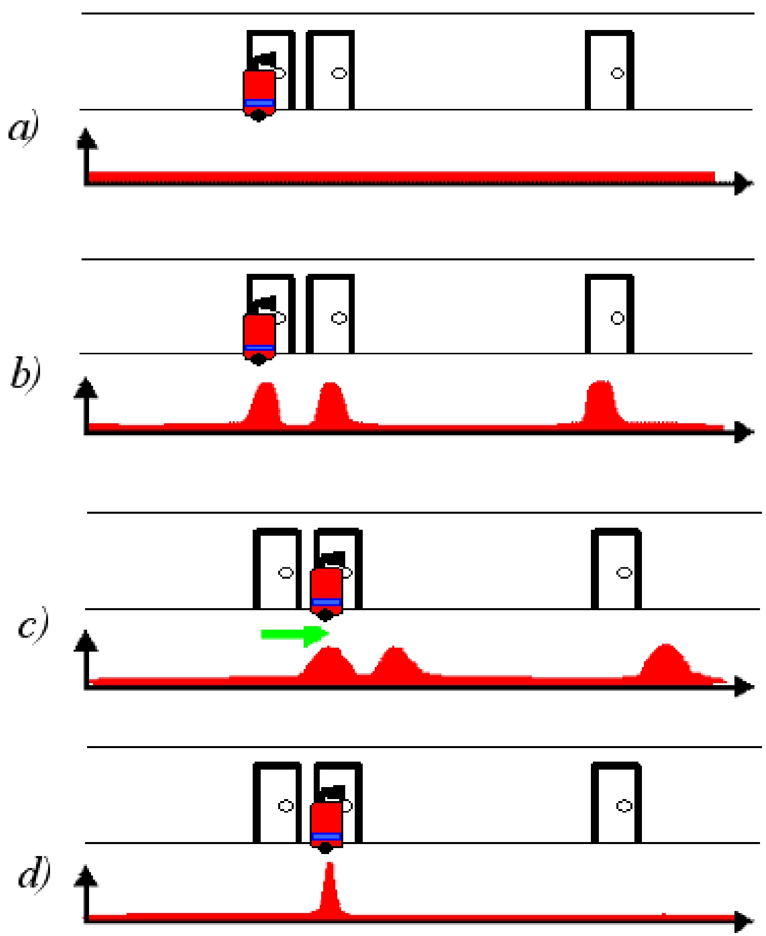
$Bel'(s) = p(o \mid s) Bel'(s)$ Estimación

$\eta = \eta + Bel'(s)$

}

Para todo s hacer: $Bel'(s) = \eta^{-1} Bel'(s)$ Normalización

2.3 Ejemplo de aplicación de un filtro de Bayes a la localización de un robot



a) Posición inicial desconocida

b) Observación de una puerta. Aplicación del modelo de observación (etapa de estimación)

c) Desplazamiento por el pasillo Aplicación del modelo de movimiento (etapa de predicción)

d) Observación de otra puerta. Aplicación del modelo de observación (etapa de estimación) FIN DE LA LOCALIZACIÓN GLOBAL

Índice del Bloque

3.1 Introducción

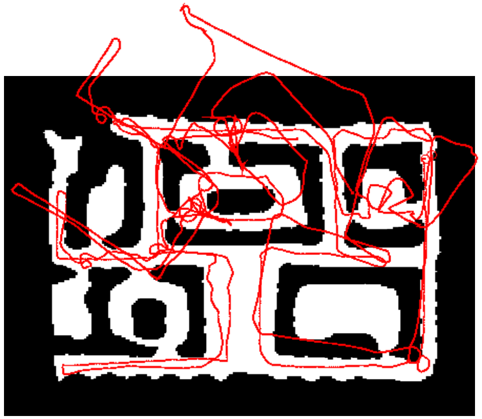
3.1.1. Localización del robot

Consiste en *estimar* la posición del robot dentro del mapa de entorno, a partir de:

* Referencias en el mapa de entorno (marcas, distancias a elementos, etc.)
* Medidas (observaciones) realizadas por los sensores
* Conocimiento de los movimientos realizados (odometría)

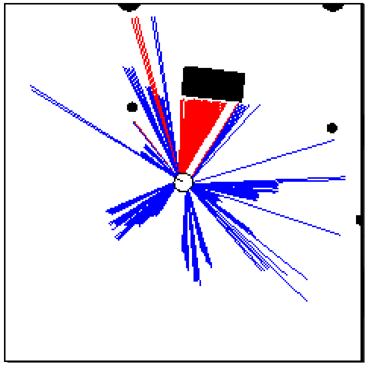
3.1.2. Problema: naturaleza de los datos sensoriales

Odometría



Sensores de distancia

(sonar, láser)



3.1.3. Problemas de localización

De menor a mayor complejidad:

- **Localización local** $\Rightarrow$ conocida la posición inicial del robot (al menos de forma aproximada), consiste en realizar un seguimiento de dicha posición que compense los errores de odometría mediante el uso de observaciones del entorno

- **Localización global** $\Rightarrow$ consiste en localizar globalmente el robot dentro del entorno, siendo su posición inicial desconocida

- **Recuperación ante fallos de localización** (problema del Kidnapping) $\Rightarrow$ consiste en localizar globalmente un robot que, conociendo su posición, es transportado instantáneamente (sin información) a otra posición alejada del entorno

3.2 Generalidades sobre localización bayesiana

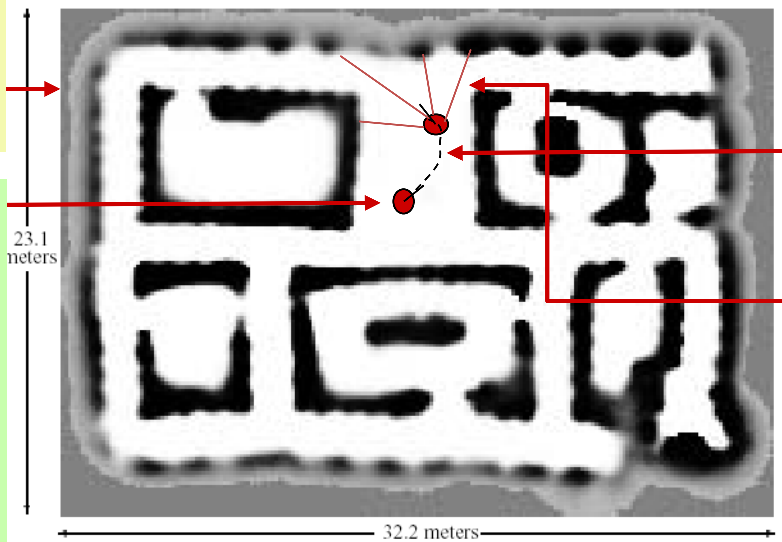
3.2.1. Los ELEMENTOS del filtro (mapa, estados, acciones y observaciones)

Mapa de entorno *m*

- Puede ser cualquier tipo de mapa (rejilla, geométrico o topológico).
- Puede haber sido obtenido experimentalmente.

Estados del robot s∈S (posibles posiciones)

- Es la variable a *estimar* por el filtro.
- Puede definirse de muchas formas. Sobre mapas métricos, la más típica es s=(x,y,θ).
- El estado *puede ser continuo o discreto*.



Acciones a∈A

- *Cambian el estado* (posición) del robot pues generan *movimiento.*
- Pueden ser de muchos tipos. Comandos de traslación, de rotación, de velocidad de las ruedas, etc.
- Suelen medirse por *odometría.*

Observaciones o∈O

- Son *medidas* tomadas por los sensores (cualquier sensor, o varios sensores).
- Pueden ser de cualquier tipo. Distancia a marcas, scan de sonar o láser, etc.

3.2.2. Los MODELOS PROBABILÍSTICOS: modelo de actuación y modelo de percepción

a) **Modelo de actuación o movimiento** p(s'|s,a) Probabilidad de pasar al estado s' si inicialmente se encuentra en el estado s y ejecuta la acción a. ACCIONES INCIERTAS.

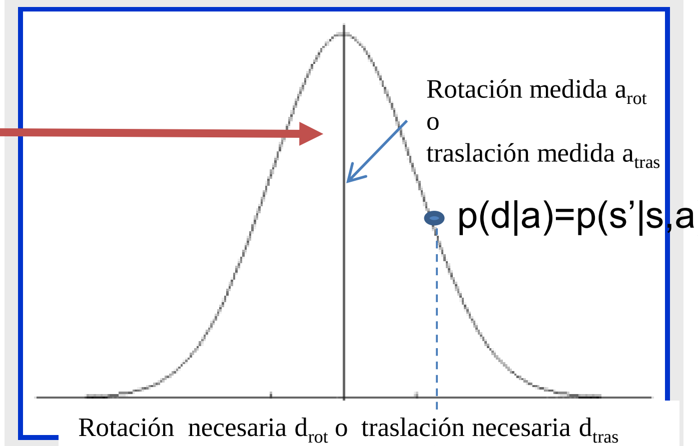
EJEMPLO: **p (s' | s , a)**

Movimiento medido **a**

$a=(a_{rot},\ a_{tras})$

ODOMETRÍA

Modelo error de odometría



ECUACIONES CINEMÁTICA

$$\begin{pmatrix} x' \\ y' \\ \theta' \end{pmatrix} = \begin{pmatrix} x + d_{tras}\cos(\theta + d_{rot}) \\ y + d_{tras}\mathrm{sen}(\theta + d_{rot}) \\ \theta + d_{rot} \end{pmatrix} \longrightarrow$$

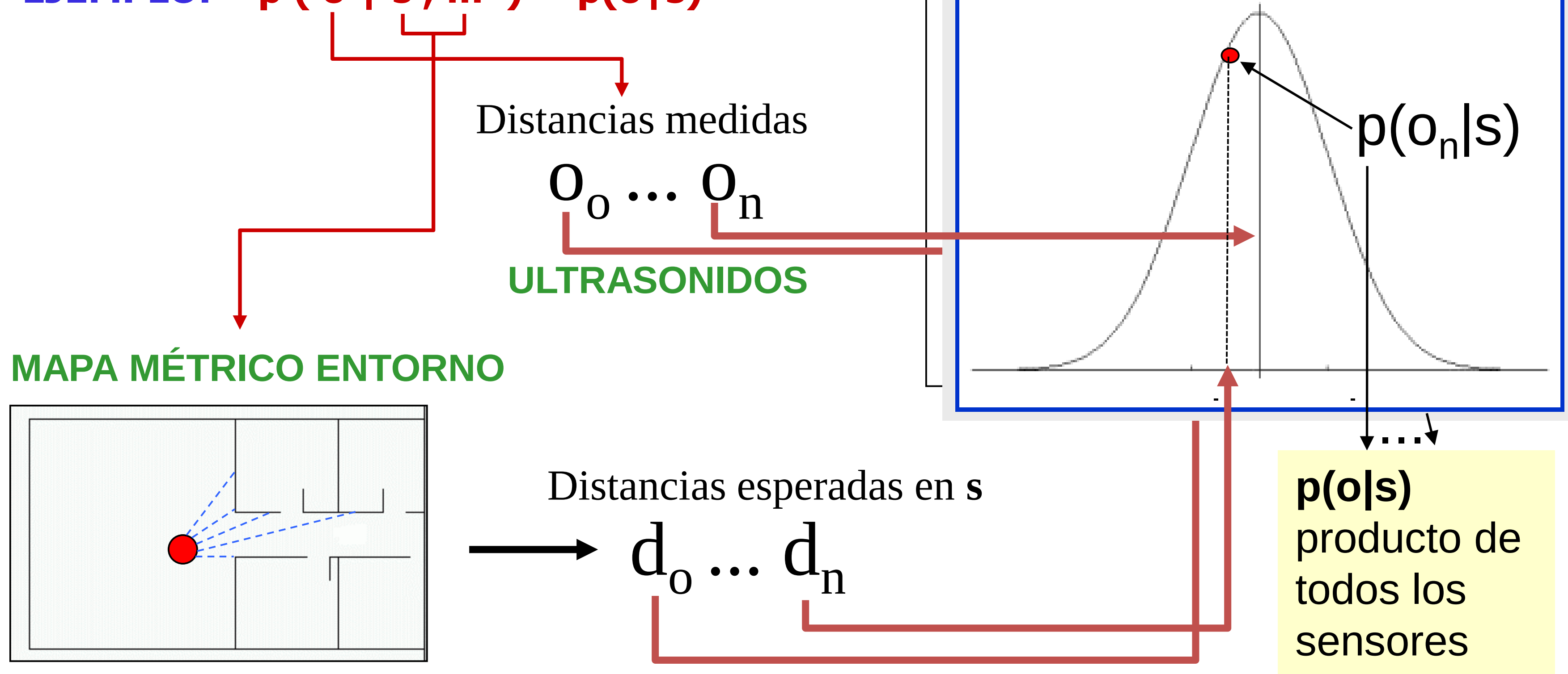
Movimiento necesario **d**

$d=(d_{rot},\ d_{tras})$

3. Aplicación a la localización de robots

b) **Modelo de observación o percepción** p(o|s) Probabilidad de realizar la observación o en el estado s. SOLAPAMIENTO PERCEPTUAL Y RUIDO EN LOS SENSORES.

EJEMPLO: $p\,(\,o\mid s\,,\,m\,) = p(o\mid s)$



3.2.3. Estimación de la localización: CREENCIA

El resultado de la estimación no es una posición exacta, sino una densidad de probabilidad sobre los estados

- Si el estado es continuo -> densidad de Creencia

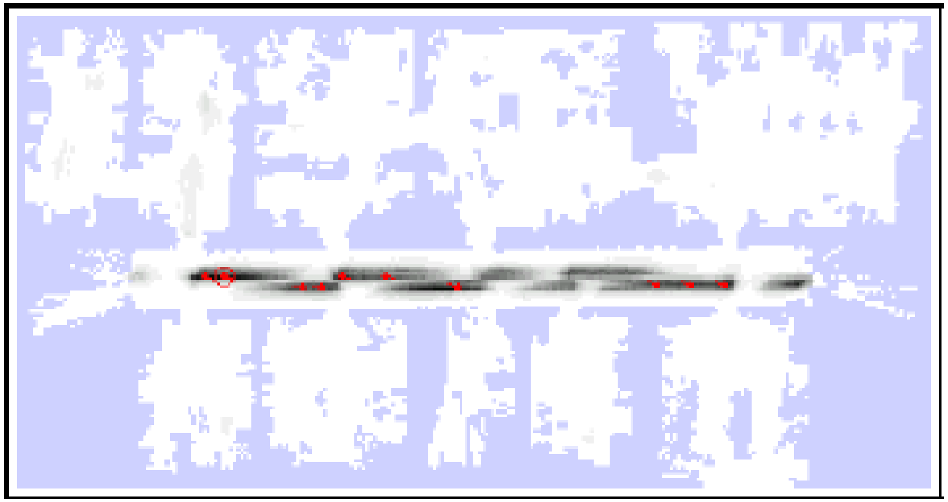
$$\int_S Bel(s) \cdot ds = 1$$

- Si el estado es discreto -> distribución de Creencia

$$\sum_{s \in S} Bel(s) = 1$$

CREENCIA Bel(S)

EJEMPLO:

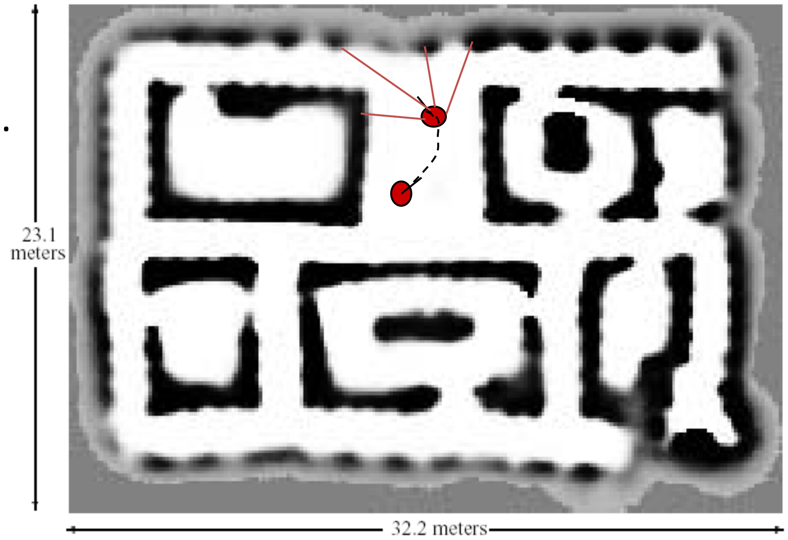


INICIALIZACIÓN DE Bel(S):

- Si posición inicial CONOCIDA: función delta en dicha posición
- Si posición inicial DESCONOCIDA: distribución uniforme
- Cualquier otra distribución según el conocimiento inicial

3.2.4. Aplicación del filtro bayesiano

- Desde su posición anterior (caracterizada por Bel(S))…
- …el robot ejecuta una acción a (desplazamiento)…
- …y obtiene una observación o (medida sensores)
- ACTUALIZACIÓN DE Bel(S)



$$\mathrm{Bel}(s') = \eta \cdot p(o \mid s') \cdot \underbrace{\int p(s' \mid s, a) \cdot \mathrm{Bel}(s) \cdot ds}_{\text{PREDICCIÓN usando el modelo de actuación}}$$

PREDICCIÓN usando el modelo de actuación

CORRECCIÓN o ESTIMACIÓN usando el modelo de observación

NORMALIZACIÓN

3.2.5. Diferentes representaciones de la localización bayesiana

$$\boxed{Bel(s') = \eta \cdot p(o \mid s') \cdot \int p(s' \mid s, a) \cdot Bel(s) \cdot ds}$$

Si el estado S es continuo y no se impone ninguna restricción adicional, un filtro de Bayes es intratable computacionalmente

SOLUCIONES PROPUESTAS:

Métodos basados en discretización (95)

- **Con representación topológica** (95)
  - Planificación con POMDPs
  - Localización global y recuperación
- **Con representación métrica (rejillas)** (96)
  - Localización global y recuperación

Filtros de Partículas (99)

- Representación basada en muestras
- Localización global y recuperación

Discretización del estado y/o creencia

Filtros de Kalman (finales 80)

- Gaussianas
- Modelos lineales
- Sólo localización local (tracking)

Multi-hipótesis (00)

- Varios filtros de Kalman
- Localización global y recuperación ante fallos

Uso de funciones continuas parametrizadas

3.3 Filtros de Kalman

3.3.1. Restricciones del filtro de Kalman respecto al filtro de Bayes

Aunque el estado S es continuo, recurre al uso de *funciones parametrizadas* :

- Modelos de actuación y percepción lineales, contaminados con ruido independiente, blanco, gaussiano y de media cero.
- La Creencia se modela también como una función de probabilidad gaussiana.

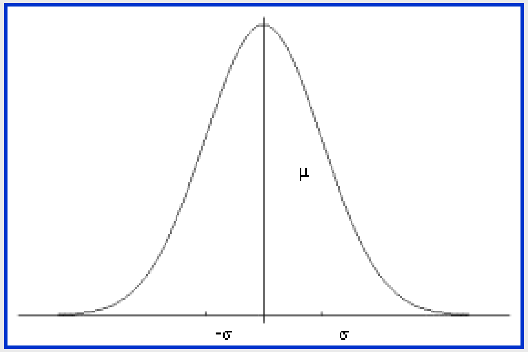
MODELOS GAUSSIANOS:

Se caracterizan por su media μ y varianza σ^2:

$$p(x)=N(\mu,\sigma^2)$$

Si **x** es multivariable… $p(\mathbf{x})=N(\boldsymbol{\mu},\boldsymbol{\Lambda})$

La Creencia se modela mediante una función de este tipo



Y los ruidos de los modelos de actuación y percepción también. Nomenclatura:

Ruido en el modelo de actuación (o del sistema): $\mathbf{w}=N(\mathbf{0},\mathbf{Q})$

Ruido en el modelo de percepción (o de medida): $\mathbf{v}=N(\mathbf{0},\mathbf{R})$

3.3.2. Planteamiento de un filtro de Kalman como estimador de estados (posición)

Los modelos que intervienen son:

1. MODELO DEL SISTEMA (lineal y con ruido gaussiano)

Equivale al MODELO DE ACTUACIÓN de un f. Bayes

$$s_t = A \cdot s_{t-1} + B \cdot a_{t-1} + w_{t-1} \longrightarrow \boxed{p(s_t \mid s_{t-1}, a_{t-1}) = N(A \cdot s_{t-1} + B \cdot a_{t-1}, Q_t)}$$

2. MODELO DE MEDIDA (lineal y con ruido gaussiano)

Equivale al MODELO DE PERCEPCIÓN de un f. Bayes

$$o_t = H \cdot s_t + v_t \longrightarrow \boxed{p(o_t \mid s_t) = N(H \cdot s_t, R_t)}$$

3. FUNCIÓN DENSIDAD DE CREENCIA GAUSSIANA

$$\boxed{Bel(s_t) = N(\hat{s}_t, P_t)}$$

- $\hat{s}_t$ es la *media* (valor estimado del estado)
- P_t es la *covarianza del error de estimación*
- El *error de estimación* es la desviación entre el estado real y el estimado $e_t = s_t - \hat{s}_t$

3.3.3. Ecuaciones del filtro de Kalman

1. INICIALIZACIÓN DE LA CREENCIA

$$\boxed{Bel(s_0) = N(\hat{s}_0, P_0)}$$

2. ETAPA DE PREDICCIÓN

$$\hat{s}_t = A \cdot \hat{s}_{t-1} + B \cdot a_{t-1}$$

$$P_t = A \cdot P_{t-1} \cdot A^T + Q_{t-1}$$

Nueva Creencia

$$\boxed{Bel(s_t) = N(A \cdot \hat{s}_{t-1} + B \cdot a_{t-1}, A \cdot P_{t-1} \cdot A^T + Q_{t-1})}$$

3. ETAPA DE CORRECCIÓN

$$\hat{s}_{t_{posterior}} = \hat{s}_{t_{prior}} + K_t(o_t - H \cdot \hat{s}_{t_{prior}})$$

con $K_t = P_t \cdot H^T \left(H \cdot P_t \cdot H^T + R_t\right)^{-1}$

Ganancia de Kalman

$$P_{t_{posterior}} = (I - K_t \cdot H) \cdot P_{t_{prior}}$$

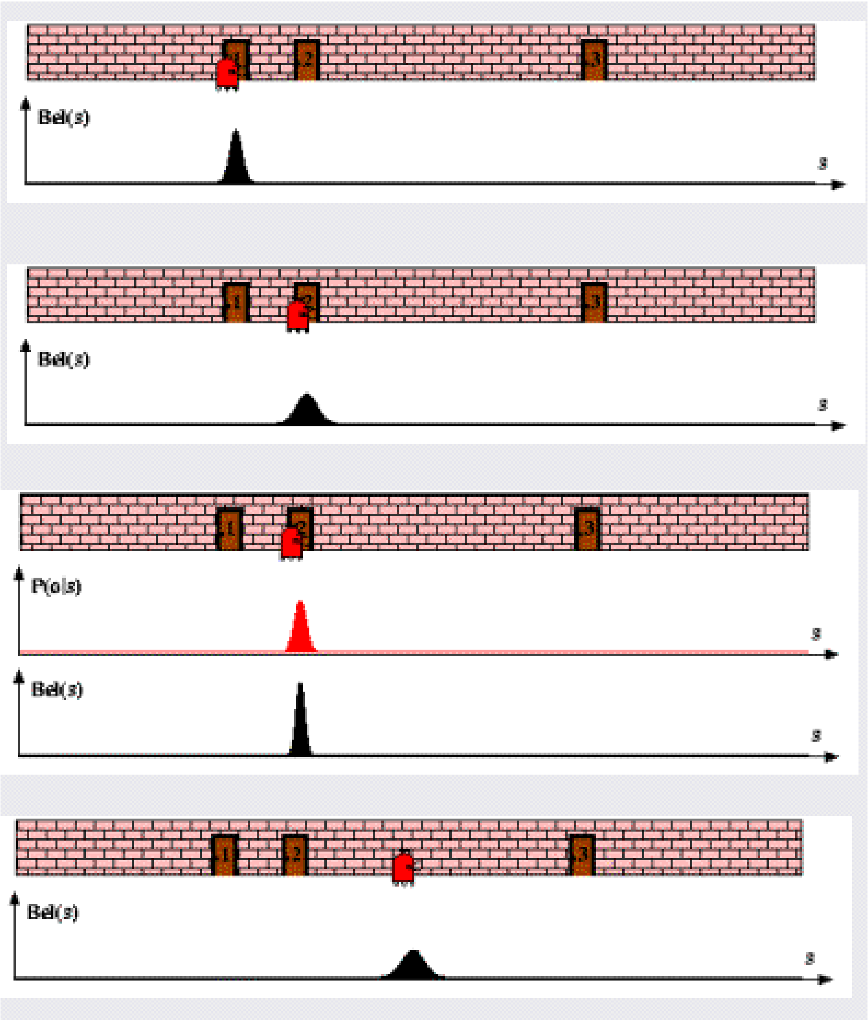
Nueva Creencia

$$\boxed{Bel(s_{t_{posterior}}) = N(\hat{s}_{t_{prior}} + K_t(o_t - H \cdot \hat{s}_{t_{prior}}), (I - K_t \cdot H) \cdot P_{t_{prior}})}$$

3.3.4. Aplicación a la localización

- La posición inicial debe ser conocida (aproximadamente), para modelarla como una gaussiana
- Sólo es capaz de seguir una hipótesis (localización local). No resuelve la localización global y el *kidnapping*.
- La dinámica del movimiento del robot debe ser lineal, y todos los ruidos gaussianos.

  Si no lineal -> **EKF**
- Las "marcas" utilizadas para el posicionamiento deben ser distinguibles para que el modelo de medida sea gaussiano



3.4 Seguimiento de Múltiples Hipótesis (MHT)

3.4.1. ¿En qué consiste?

Es una GENERALIZACIÓN del filtro de Kalman que permite el seguimiento de varias hipótesis para resolver el problema de LOCALIZACIÓN GLOBAL.

- El estado S sigue siendo continuo
- La creencia admite varias hipótesis (multi-modal), todas ellas gaussianas
- Cada hipótesis es actualizada mediante un filtro de Kalman

PROBLEMÁTICA ADICIONAL:

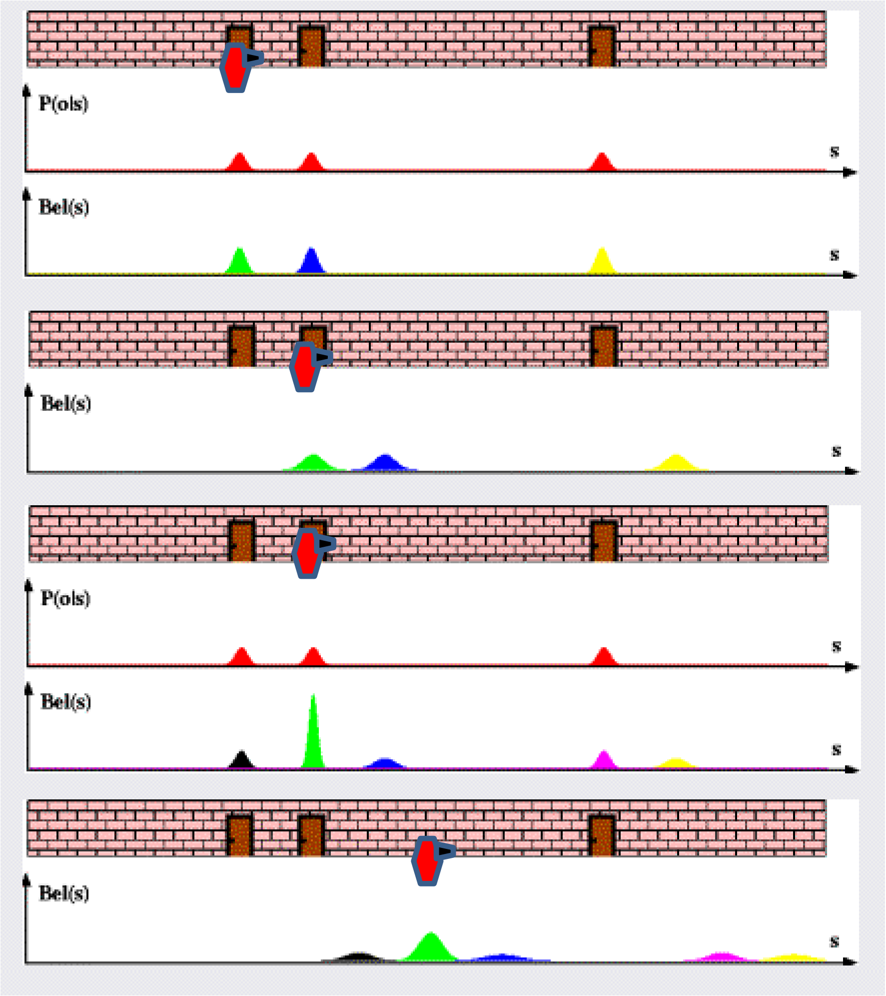
- Asociación de datos: ¿qué medida u observación corresponde a cada hipótesis?
- Gestión de las hipótesis: ¿cuándo añadir una nueva hipótesis o eliminar alguna existente?

3.4.2. Localización mediante MHT

- La posición inicial puede ser desconocida
- Es capaz de seguir varias hipótesis por lo que resuelve el problema de localización global.
- La dinámica del movimiento del robot debe ser lineal, y todos los ruidos gaussianos.

  Si no lineal -> **EKF**

- Las "marcas" utilizadas para el posicionamiento pueden ser indistinguibles (pero debe establecerse un sistema para gestionar las hipótesis)



3.5 Rejillas de probabilidad. Localización de Markov

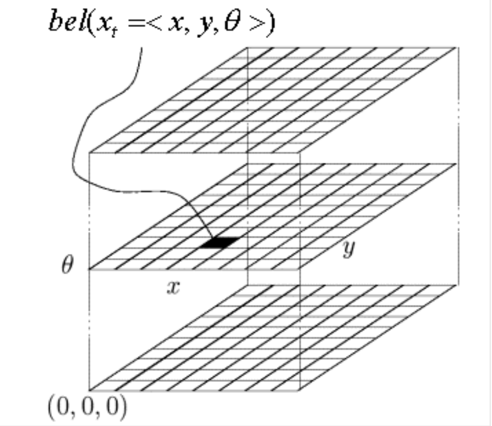
3.5.1. Discretización del estado mediante rejillas de probabilidad

La **localización de Markov** consiste en aplicar un filtro bayesiano sobre un espacio de estados discreto (número finito de estados $s \in \mathbf{S}$). Así, la integral se convierte en sumatorio:

$$\boxed{Bel(s') = \eta \cdot p(o \mid s') \cdot \sum_{\forall s \in S} p(s' \mid s, a) \cdot Bel(s) \qquad \forall s' \in S}$$

Discretización típica: REJILLAS de tamaño fijo

Por ejemplo, si el estado es la posición $s=(x,y,\theta)$



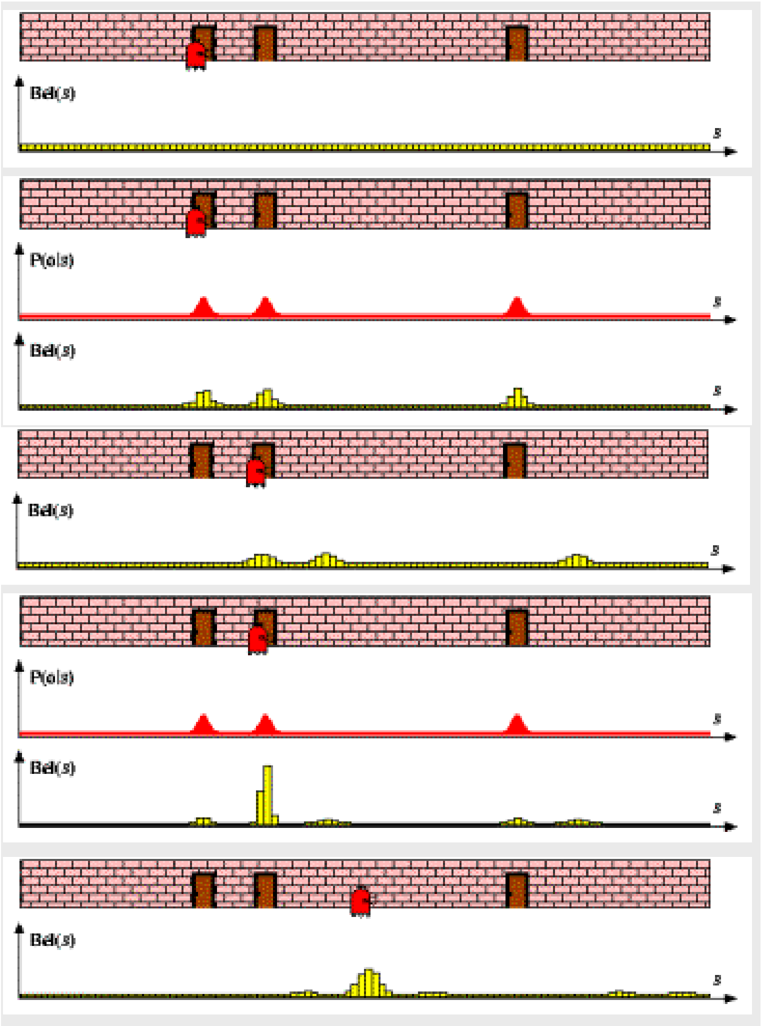
VENTAJAS respecto a filtros de Kalman y MHT:

- Los modelos no tienen que ser lineales, ni los ruidos gaussianos (no restricciones)
- La creencia no es una función parametrizada, sino una distribución de cualquier tipo

3. Aplicación a la localización de robots

3.5.2.Localización de Markov

- La posición inicial puede ser desconocida
- Es capaz de seguir varias hipótesis por lo que resuelve el problema de localización global.
- La dinámica del movimiento del robot no tiene que ser lineal, y los modelos de ruido en medida o actuación no tienen que ser gaussianos.
- Las "marcas" utilizadas para el posicionamiento pueden ser indistinguibles sin que ello conlleve complejidad adicional ninguna, como sucedía con el MHT.



3.6 Filtros de partículas. Localización de Monte Carlo

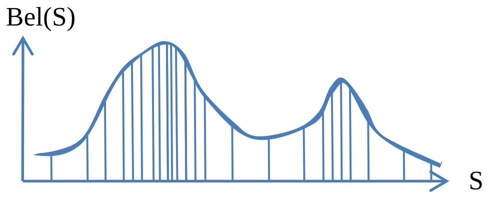
3.6.1. Generalidades

PROBLEMAS de la Localización de Markov:

- Cuando la creencia se centra en una zona, no tiene interés mantener una rejilla de tamaño fijo en todo el entorno.
- Poca eficiencia computacional si se desea tener una buena resolución en entornos amplios (aumenta el número de estados)

LOCALIZACIÓN DE MONTE CARLO (MCL):

- El estado **S** es continuo (los estados s pueden tomar cualquier valor)
- La creencia se discretiza en **partículas** (muestras de Bel(**S**) que adquirirán mayor concentración en las zonas del entorno con más probabilidad de ser la posición del robot).



3.6.2. Discretización de la creencia mediante partículas

El estado **S** es continuo

La creencia se discretiza mediante un conjunto de N partículas:

$$\boxed{\mathbf{P}=\{p^i \mid i=1,\ldots,N\}}$$

Cada **PARTÍCULA** tiene la forma $\boxed{p=\{s,w\}}$ donde:

- s $\Rightarrow$ posición de la partícula en el entorno (normalmente $s=(x,y,\theta)$)
- w $\Rightarrow$ peso de la partícula ("importancia" de la misma). Se conoce como **factor de importancia**. Cuanto mayor es el factor de importancia, mayor probabilidad tiene la partícula de ser la posición real del robot. Normalmente se normaliza el conjunto de todas las partículas de manera que la suma de sus factores de importancia sea 1.

3.6.3. Localización de Monte Carlo (MCL)

1. CONJUNTO INICIAL DE PATÍCULAS $\mathbf{P_0}$ según conocimiento inicial de la posición:

- Posición inicial desconocida $\Rightarrow$ muestras aleatorias con $w^i=N^{-1}$
- Posición inicial conocida $s_0 \Rightarrow$ todas las muestras en s_0 con $w^i=N^{-1}$

2. ETAPA DE PREDICCIÓN (movimiento del robot tras ejecutar la acción a_{t-1})

Se genera un nuevo conjunto de partículas $\mathbf{P}_t=\{p_t^i\}=\{s_t^i,w_t^i\}$, a partir del anterior $\mathbf{P}_{t-1}=\{p_{t-1}^i\}=\{s_{t-1}^i,w_{t-1}^i\}$. Cada nueva partícula p_t^i se genera en **dos pasos**:

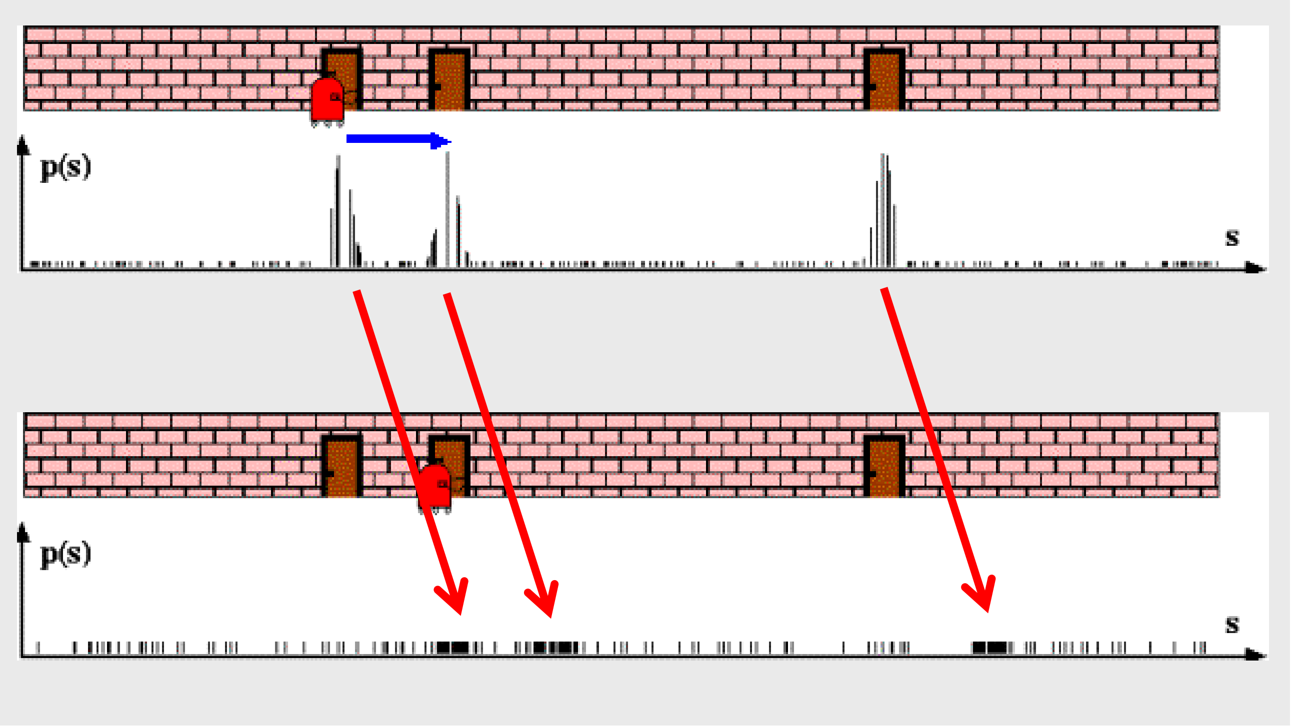
a. **Seleccionando aleatoriamente** una partícula p_{t-1}^i del conjunto anterior, con probabilidad determinada por el factor de importancia de dichas partículas (resampling).

b. **Propagando** dicha partícula a su nueva posición, utilizando el modelo de movimiento $p(s'|s,a)$. Para ello, se toma s_{t-1}^i como posición original, y se escoge aleatoriamente una muestra de la distribución $p(s_t^i|s_{t-1}^i,a_{t-1})$

El factor de importancia de la nueva partícula es $w_{t-1}^i=N^{-1}$

3. Aplicación a la localización de robots

Ejemplo de predicción:



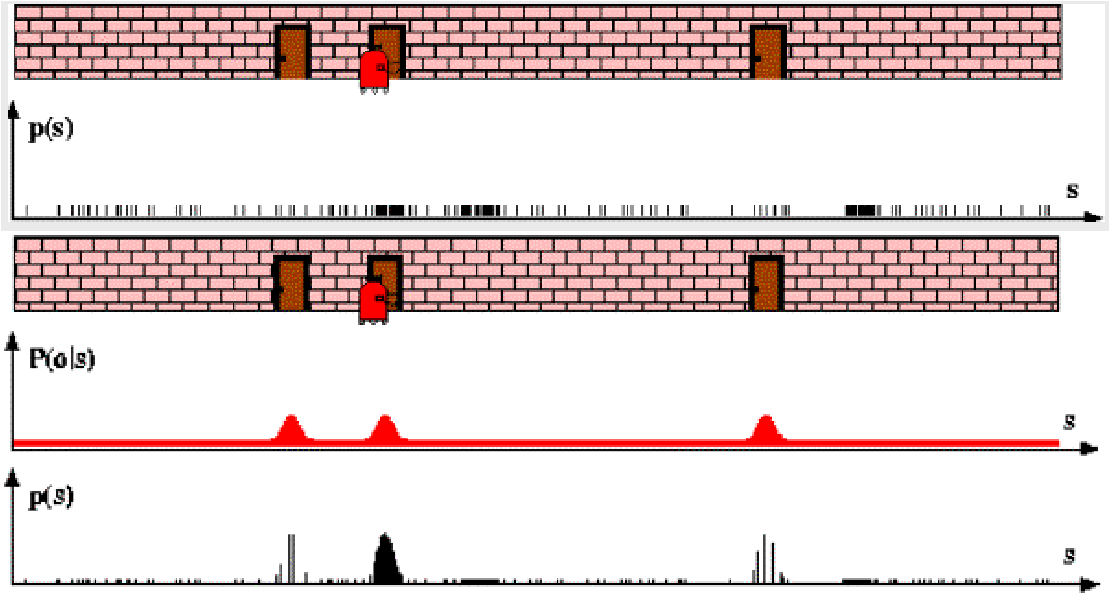
3. Aplicación a la localización de robots

3. ETAPA DE ESTIMACIÓN (obtención de una medida o_t con los sensores)

La obtención de una medida o_t se incorpora al filtro de partículas recalculando los factores de importancia de cada una de ellas del siguiente modo:

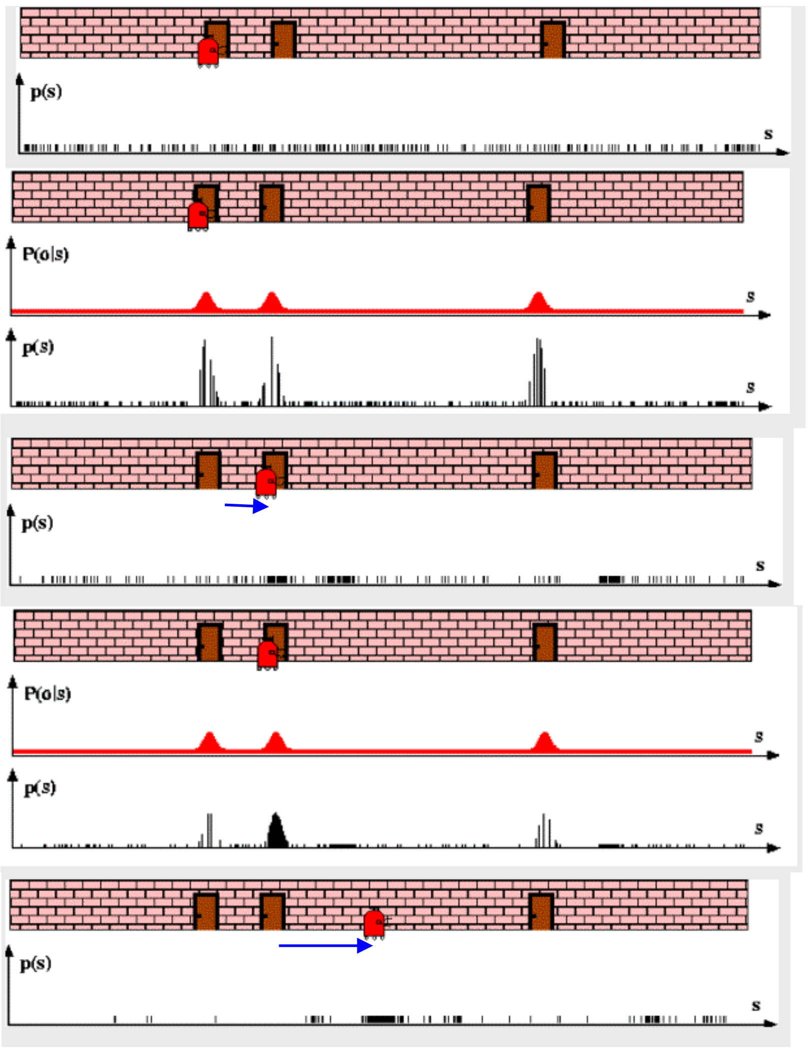
$$w_t^i = \eta \cdot p(o_t | s_t^i) \quad \text{con } \eta \text{ factor de normalización}$$

Ejemplo de estimación:



3. Aplicación a la localización de robots

Ejemplo con varias iteraciones…



a) Inicialización. Muestras aleatorias con el mismo factor de importancia

b) Observación de una puerta. Reponderación de los factores de importancia

c) Movimiento. Remuestreo del conjunto de partículas y propagación de las mismas

d) Observación de una puerta. Reponderación de los factores de importancia

e) Movimiento. Remuestreo del conjunto de partículas y propagación de las mismas

3.6.4. Algoritmo MCL

Cada iteración del MCL…

ALGORITMO MCL [obtención de **P'** a partir de (**P**,a,o)]

P'=∅
Desde i=0 hasta N hacer:
{
extraer aleatoriamente de **P** una muestra $p^i=\{s^i,w^i\}$, con probabilidad dada por los factores de importancia
generar aleatoriamente la posición final de dicha muestra s'^i según la distribución $p(s'^i \| s^i,a)$
calcular el factor de importancia de la nueva partícula $w'^i=p(o \| s'^i)$
Añadir a **P'** la nueva partícula $p'=\{s'^i,w'^i\}$
}
Normalizar los factores de importancia de **P'**

3.6.5. Ventajas e inconvenientes del MCL

VENTAJAS:

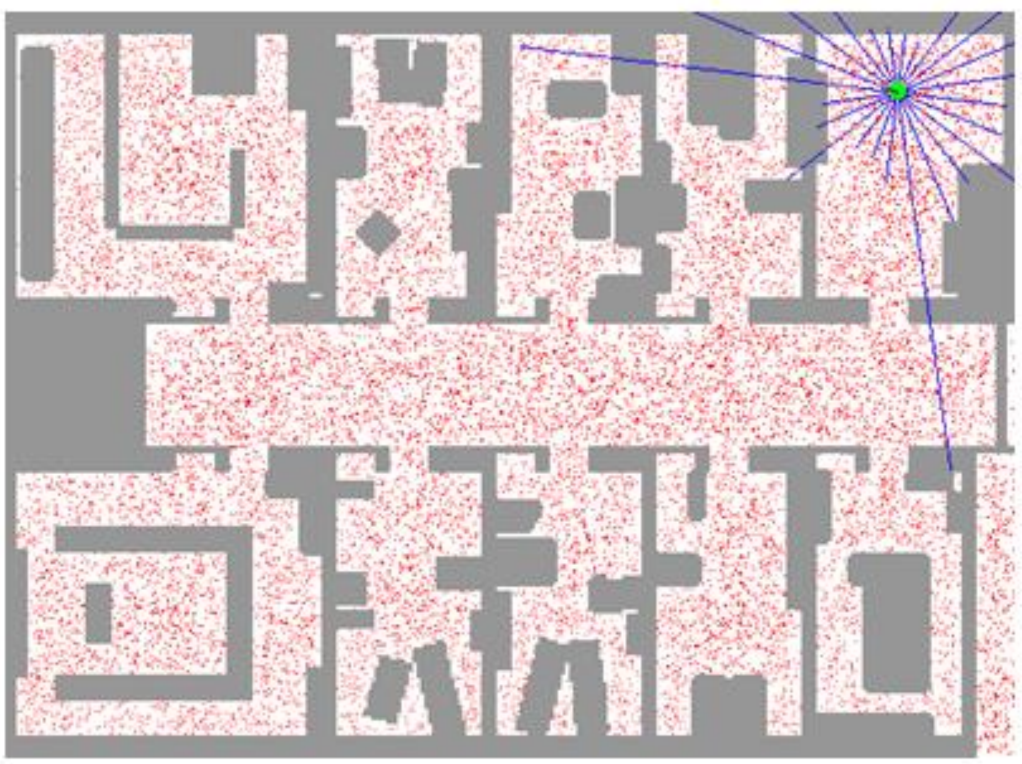
- Resuelve los problemas de localización local y localización global.
- Cualquier tipo de modelo del sensor, dinámica de movimiento y distribución de ruido.
- Centra los recursos computacionales en las áreas del entorno en que hay más probabilidad de que se encuentre el robot (mayor concentración de partículas)
- Precisión: no discretizan el estado.
- Controlando y adaptando el número de partículas, los filtros de partículas se adaptan fácilmente a los recursos computacionales disponibles
- Fáciles de implementar desde el punto de vista computacional.

INCONVENIENTES:

- Si el conjunto de partículas es pequeño, un robot bien localizado puede llegar a perderderse debido a que el proceso de "remuestreo" aleatorio no genere ninguna muestra en la localización correcta.
- Para resolver también el problema de la recuperación ante fallos (*kidnapping*) es necesario alterar el algoritmo estándar, manteniendo siempre un número reducido de partículas aleatoriamente distribuidas por el entorno.

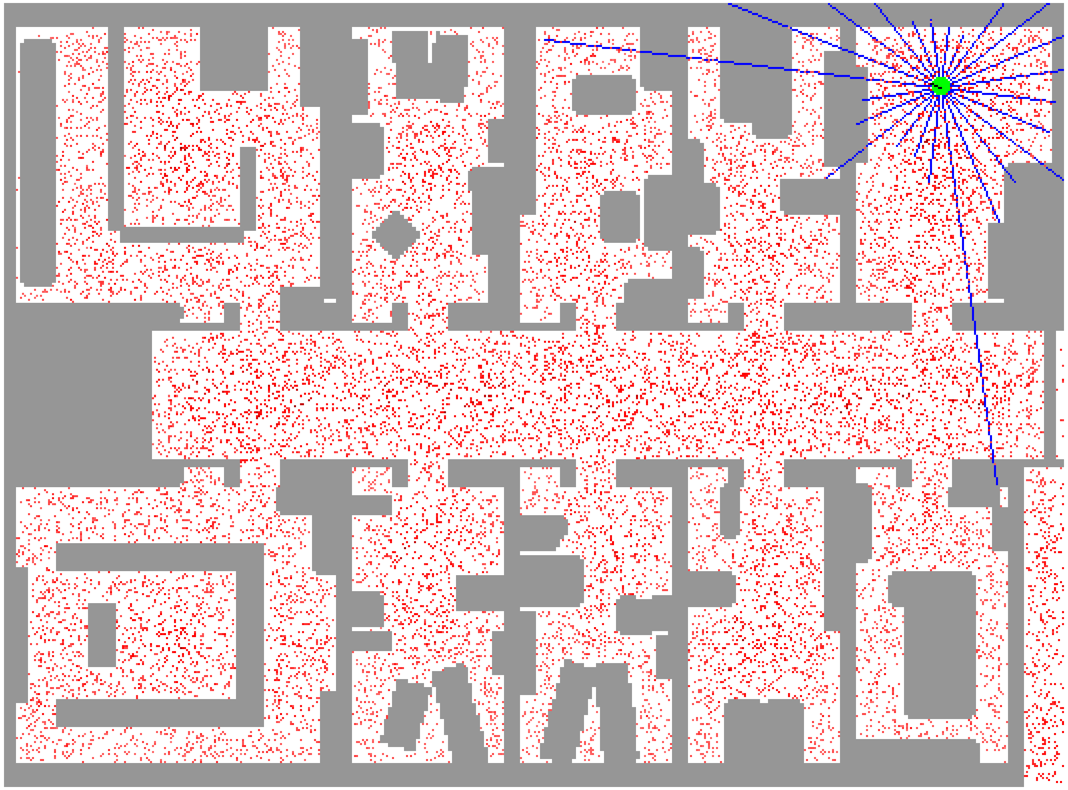
3.6.6. Ejemplos prácticos

Ejemplo de **localización global** utilizando **ultrasonidos** (anillo de 24 sensores).



3.6.6. Ejemplos prácticos

Ejemplo de **localización global** utilizando **ultrasonidos** (anillo de 24 sensores).



3. Aplicación a la localización de robots

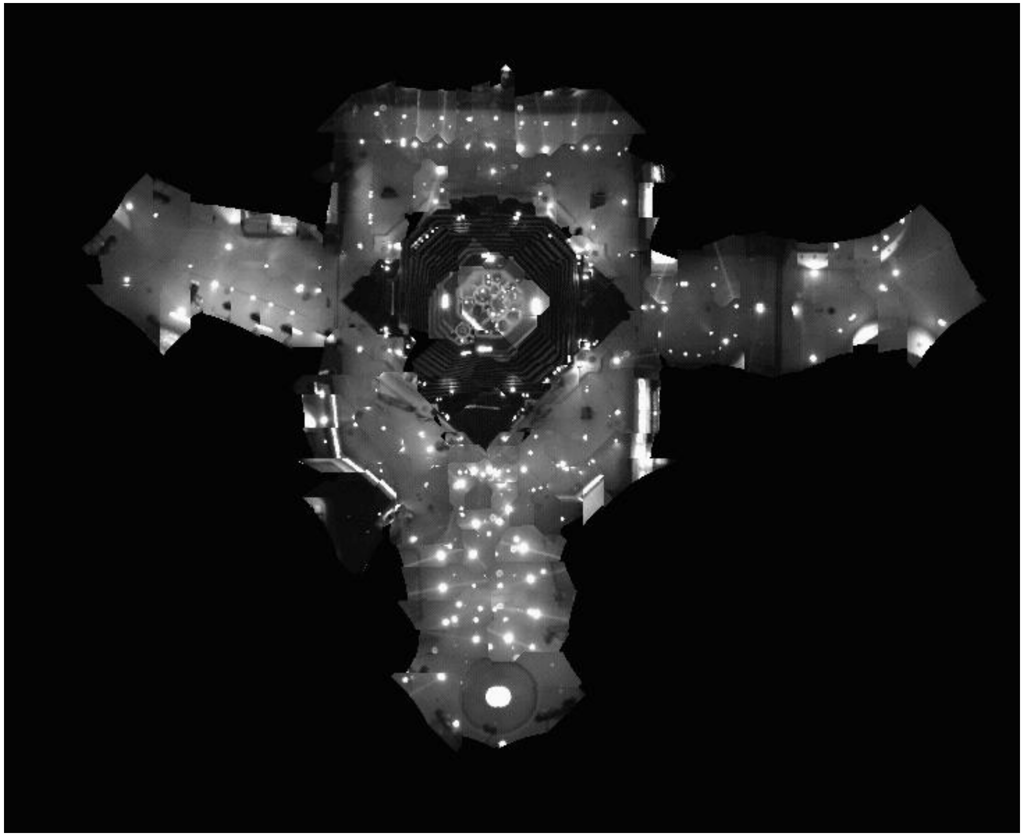
Ejemplo de **localización global** utilizando **ultrasonidos** con **número de partículas adaptativo**

3. Aplicación a la localización de robots

Ejemplo de **localización global** utilizando **láser** con **número de partículas adaptativo**

3. Aplicación a la localización de robots

Ejemplo de **localización global** utilizando **visión** (Condensation algorithm)

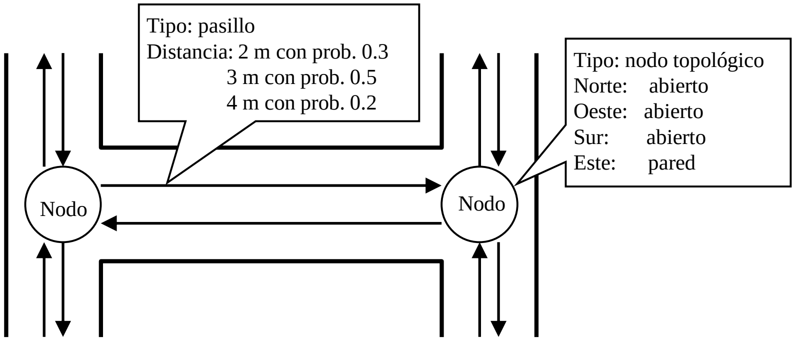


3.7 Localización topológica

3.7.1. Generalidades

- Todos los métodos anteriores utilizan como estado la posición métrica $s=(x,y,\theta)$ del robot dentro del entorno
- Cuando se trabaja con una discretización más gruesa (TOPOLÓGICA) los estados pueden ser los nodos de un grafo.

$$\boxed{Bel(s') = \eta \cdot p(o \mid s') \cdot \sum_{\forall s \in S} p(s' \mid s,a) \cdot Bel(s) \qquad \forall s' \in S}$$



- Este tipo de modelos bayesianos se formulan matemáticamente como Procesos de Decisión de Markov Parcialmente Observables (POMDPs) (se verán en el tema de planificación).
  - La localización corresponde al **estimador de estados del POMDP**
  - La planificación corresponde a las **políticas de decisión del POMDP**

3.7.2. Modelos de transición y observación

1. MODELO DE TRANSICIÓN p(s'|s,a)

- Las acciones no son movimientos métricos, sino comportamientos abstractos del tipo "girar 90º a la izquierda", "seguir pasillo", "cruzar puerta", etc.
- El modelo de transición se almacena en matrices de transición (una por cada posible acción).

EJEMPLO:

Modelo con 5 acciones…

a_o=Salir de habitación

s \ s'	0	1	2	3	4	5	6	7	8	…
0	5	0	0	0	0	0	0	0	0	
1	0	5	0	0	0	0	0	0	0	
2	0	0	5	0	0	0	0	0	0	
3	0	0	0	5	0	0	0	0	0	
4	0	0	0	0	5	0	0	0	0	…
5	0	0	0	0	0	5	0	0	0	
6	0	0	0	0	0	0	5	0	0	
7	0	0	0	0	0	0	0	5	0	
8	0	0	0	0	0	0	0	0	5	
…					…					

a_E=Entrar en habitación

s \ s'	0	1	2	3	4	5	6	7	8	…
…					…					
16	90	0	0	0	0	0	0	0	0	
17	0	0	0	0	0	0	0	0	0	
18	0	0	0	0	0	0	0	0	0	
19	0	90	0	0	0	0	0	0	0	…
20	0	0	0	0	0	0	0	0	0	
21	0	0	0	0	0	0	0	0	0	
22	0	0	0	0	0	0	0	0	0	
23	0	0	0	0	0	0	0	0	0	
…					…					

a_F=Seguir pasillo

s \ s'	…	22	23	24	25	26	27	28	29	…
…					…					
22		10	0	0	0	70	0	0	0	
23		0	0	0	0	0	0	0	0	
24		0	0	10	0	0	0	0	0	
25	…	0	0	0	0	0	0	0	0	…
26		0	0	0	0	10	0	0	0	
27		0	0	0	0	0	0	0	0	
28		0	0	70	0	0	0	10	0	
29		0	0	0	0	0	0	0	0	
…					…					

a_L=Girar a la izquierda

s \ s'	…	14	15	16	17	18	19	20	21	…
…					…					
14		5	90	5	0	0	0	0	0	
15		0	5	90	5	0	0	0	0	
16		5	0	5	90	0	0	0	0	
17	…	90	5	0	5	0	0	0	0	…
18		0	0	0	0	5	90	5	0	
19		0	0	0	0	0	5	90	5	
20		0	0	0	0	5	0	5	90	
21		0	0	0	0	90	5	0	5	
…					…					

a_R=Girar a la derecha

s \ s'	…	14	15	16	17	18	19	20	21	…
…					…					
14		5	0	5	90	0	0	0	0	
15		90	5	0	5	0	0	0	0	
16		5	90	5	0	0	0	0	0	
17	…	0	5	90	5	0	0	0	0	…
18		0	0	0	0	5	0	5	90	
19		0	0	0	0	90	5	0	5	
20		0	0	0	0	5	90	5	0	
21		0	0	0	0	0	5	90	5	
…					…					

3. Aplicación a la localización de robots

2. MODELO DE OBSERVACIÓN p(o|s)

- Las observaciones también suelen ser medidas abstractas del tipo "número de marcas del tipo A visualizadas", "detección de puerta", "detección de pasillo", etc
- El modelo de observación se almacena en matrices de observación (una por cada posible observación).

EJEMPLO:

Modelo con 3 observaciones…

$p(o_{OAU}|s)$

s \ OAU	0	1	2	3	4	5	6	7
...				...				
14	0	4	0	0	4	83	0	9
15	2	0	42	4	2	0	46	4
16	0	2	0	2	2	43	2	49
17	2	2	42	46	0	0	4	4
18	2	42	0	4	2	46	0	4
19	40	4	44	4	4	0	4	0
20	2	2	0	0	42	46	4	4
21	4	0	78	9	0	0	9	0
...				...				

$p(o_{OVM}|s)$

s \ OVM	0	1	2	3	4	5	6
...				...			
14	0	10	10	60	10	10	0
15	60	20	20	0	0	0	0
16	20	60	10	10	0	0	0
17	60	20	20	0	0	0	0
18	0	10	10	60	10	10	0
19	20	60	10	10	0	0	0
20	20	60	10	10	0	0	0
21	60	20	20	0	0	0	0
...				...			

$p(o_{OVP}|s)$

s \ OVP	0	1	2	3
...				
14	10	10	10	70
15	70	10	10	10
16	70	10	10	10
17	70	10	10	10
18	10	10	10	70
19	70	10	10	10
20	70	10	10	10
21	70	10	10	10
...				

3.7.3. Ventajas e inconvenientes de la localización topológica

VENTAJAS:

- Es posible modelar entornos mucho más amplios mediante un número más reducido de estados
- No es necesario mantener la consistencia métrica global en el entorno
- Mayor eficiencia computacional: tiempos de cálculo reducidos y necesidad de menos recursos de memoria
- Los modelos de transición y observación no deben calcularse “on-line” como en los enfoques métricos, sino que se almacenan “off-line” en tablas, a las que se accede mucho más rápido en tiempo de ejecución
- No existen restricciones en cuanto a los modelos, que pueden ser no lineales y multimodales
- Resuelve la localización local, global y el *kidnapping* (evitando la anulación completa de la probabilidad de un estado)

INCONVENIENTES:

- Menor resolución en el posicionamiento
- Acciones y observaciones más complejas

3.8 Comparativa de los métodos de localización bayesianos

	Filtro de Kalman	Seguimiento de múltiples hipótesis	Localización de Markov	Localización de Monte Carlo	Localización mediante POMDPs
Modelo Sensores	gaussiano	gaussiano	no-gaussiano	no-gaussiano	basado en caracter. "abstractas"
Modelo Creencia	gaussiano	gaussiano multimodal	rejilla fija	partículas	nodos topológicos
Eficiencia (memoria)	++	++	—	++	++
Eficiencia (tiempo)	++	++	—	++	++
Implementación	+	—	+	++	+
Exactitud	++	++	+	++	—
Robustez	—	+	++	++	+
Localización global	No	Si	Si	Si	Si

3.9 Evaluación de la incertidumbre

La localización bayesiana genera como resultado una función densidad de probabilidad o una distribución de probabilidad en lugar de un dato único (CREENCIA)

Puede ser necesario evaluar el **grado de incertidumbre** de la creencia. Para ello:

- Métodos basados en filtros de Kalman $\Rightarrow$ varianzas de las gaussianas
- Métodos que discretizan el estado o la creencia $\Rightarrow$ **entropía:**

Entropía de la creencia: $\boxed{H(Bel) = -\sum_{Bel(s)\neq 0} Bel(s) \cdot \log(Bel(s))}$ A mayor valor, más incertidumbre

Entropía normalizada: $\boxed{H_{max}(Bel) = -\sum_{s\in S} \frac{1}{m} \cdot \log\left(\frac{1}{m}\right) = -\log\left(\frac{1}{m}\right) = \log(m)}$ (0= delta; 1= uniforme)

CLASIFICACIÓN DEL RESULTADO:

- **Etapa de localización global** $\Rightarrow$ la varianza o entropía (incertidumbre) es elevada
- **Etapa de localización local** $\Rightarrow$ la varianza o entropía (incertidumbre) es reducida. El robot está *globalmente localizado*

Índice del Bloque

4.1 Introducción

- En el apartado anterior, el robot se localiza a partir de un mapa conocido a priori (problema de estimación de bajas dimensiones (x,y,θ)).

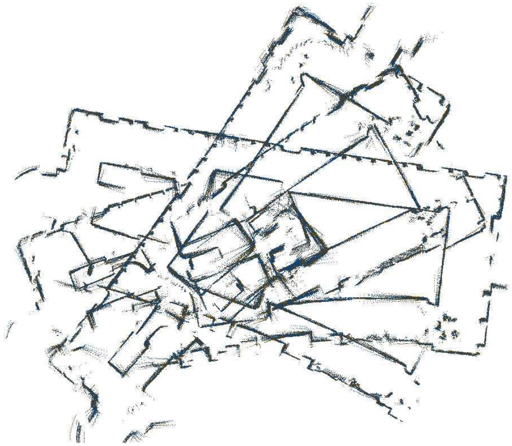
- Sin embargo, en muchas aplicaciones el mapa no está disponible, y el robot debe obtenerlo a partir de sus sensores (problema de estimación de dimensión mucho más elevada).

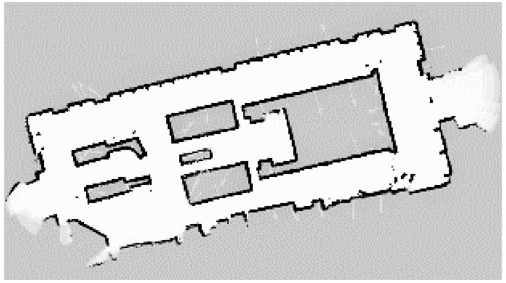
- Se aborda en este apartado el problema de **mapeado suponiendo que la posición del robot es conocida** sin errores.

4. Aplicación a la obtención de mapas

Factores que influyen en la **dificultad del mapeado**:

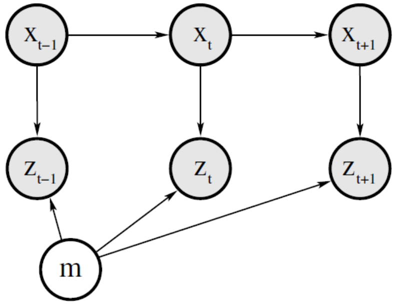
- **Tamaño del mapa** respecto a el rango de percepción sensorial del robot
- **Ruido en la percepción y la actuación**
- **Solapamiento perceptual** (lugares distintos que se "ven" igual por los sensores del robot – dificultad para establecer correspondencias)
- **Lazos** (regresar al mismo punto por un camino diferente – dificultad en su detección)





4.2 Rejillas de ocupación

- Método introducido por Moravec y Elfes in 1985
- Representa el entorno mediante una rejilla.
- Estima la probabilidad de que cada celda esté ocupada por un obstáculo.
- **Suposiciones principales:**
  - La posición del robot es conocida.



  - La ocupación de cada celda (m[xy]) es independiente de las demás.

$$Bel(m_t) = P(m_t \mid x_1, z_2 \ldots, x_{t-1}, z_t)$$
$$= \prod_{x,y} Bel(m_t^{[xy]})$$

4. Aplicación a la obtención de mapas

- **Idea principal:** actualizar cada celda utilizando un filtro de Bayes binario

$$\boxed{Bel(m_t^{[xy]}) = \eta\ p(z_t \mid m_t^{[xy]}) \int p(m_t^{[xy]} \mid m_{t-1}^{[xy]}, \mathrm{x}_{t-1}) Bel(m_{t-1}^{[xy]})\, dm_{t-1}^{[xy]}}$$

- **Suposición adicional:** el mapa es estático

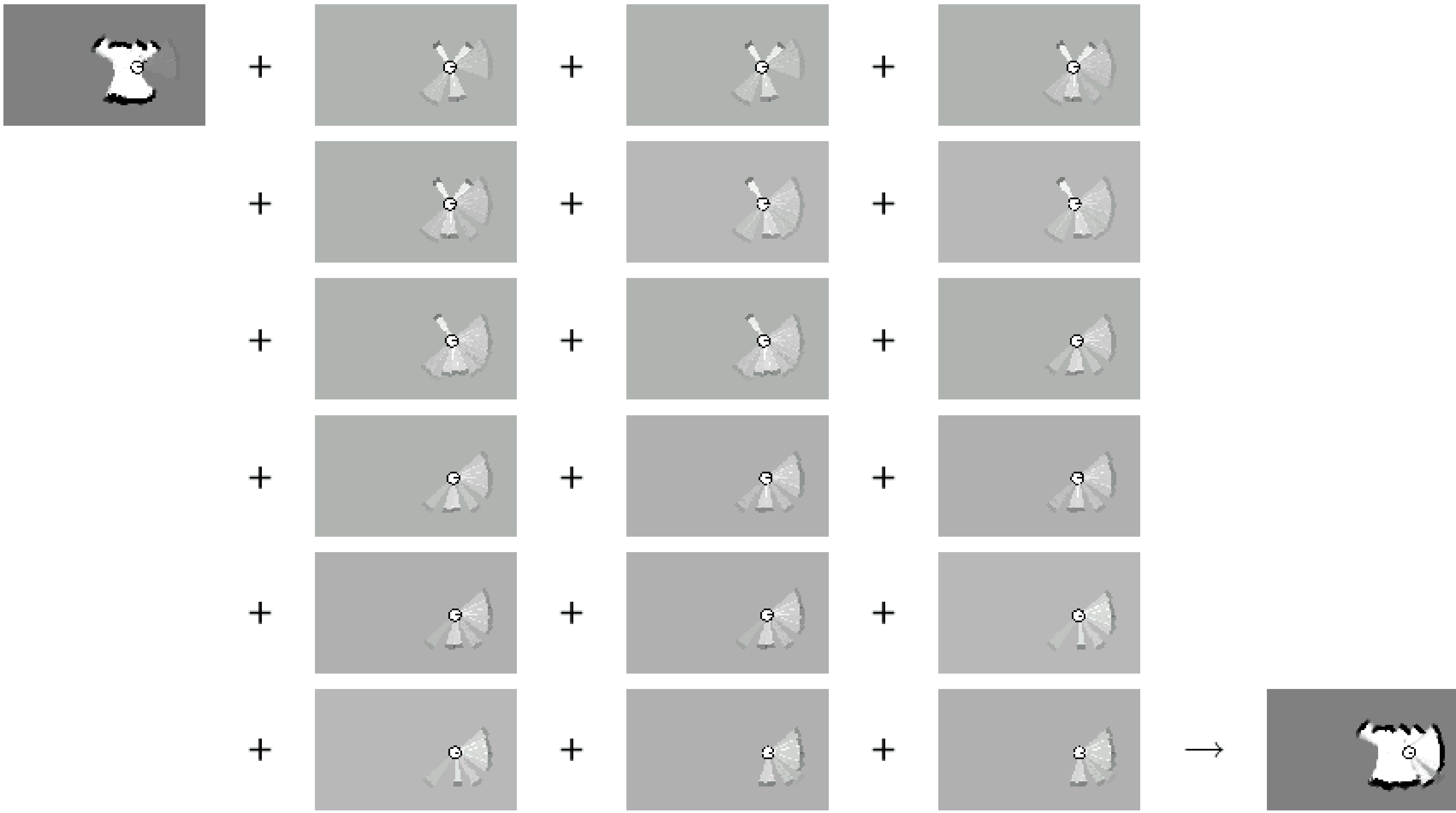
$$\boxed{Bel(m_t^{[xy]}) = \eta\ p(z_t \mid m_t^{[xy]}) Bel(m_{t-1}^{[xy]})}$$

- El mapa se actualiza utilizando un **modelo inverso del sensor**.

$$\boxed{Bel\left(m_t^{[xy]}\right) = 1 - \left(1 + \frac{P\left(m_t^{[xy]} \mid z_t, u_{t-1}\right)}{1 - P\left(m_t^{[xy]} \mid z_t, u_{t-1}\right)} \cdot \frac{1 - P\left(m_t^{[xy]}\right)}{P\left(m_t^{[xy]}\right)} \cdot \frac{Bel\left(m_{t-1}^{[xy]}\right)}{1 - Bel\left(m_{t-1}^{[xy]}\right)}\right)^{-1}}$$

4. Aplicación a la obtención de mapas

Ejemplo de obtención incremental de un mapa de rejilla de ocupación:



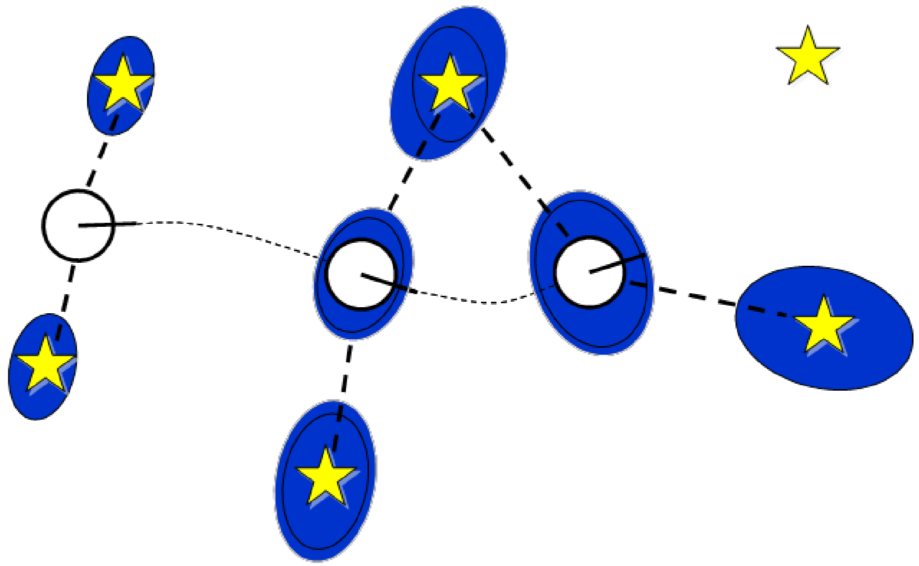
Índice del Bloque

5.1 Introducción

- En la práctica, el robot no conoce su posición exacta mientras va obteniendo el mapa.
- Problema de tipo “chicken and egg”: para obtener el mapa se requiere la posición, y para obtener la posición se requiere el mapa.
- **SLAM: Simultaneous Localization and Mapping**
  - Dadas:
    - ❑ Las acciones realizadas por el robot
    - ❑ Las observaciones realizadas del entorno (marcas cercanas)
  - Estimar:
    - ❑ La posición de las marcas del mapa
    - ❑ El camino seguido por el robot dentro del mapa

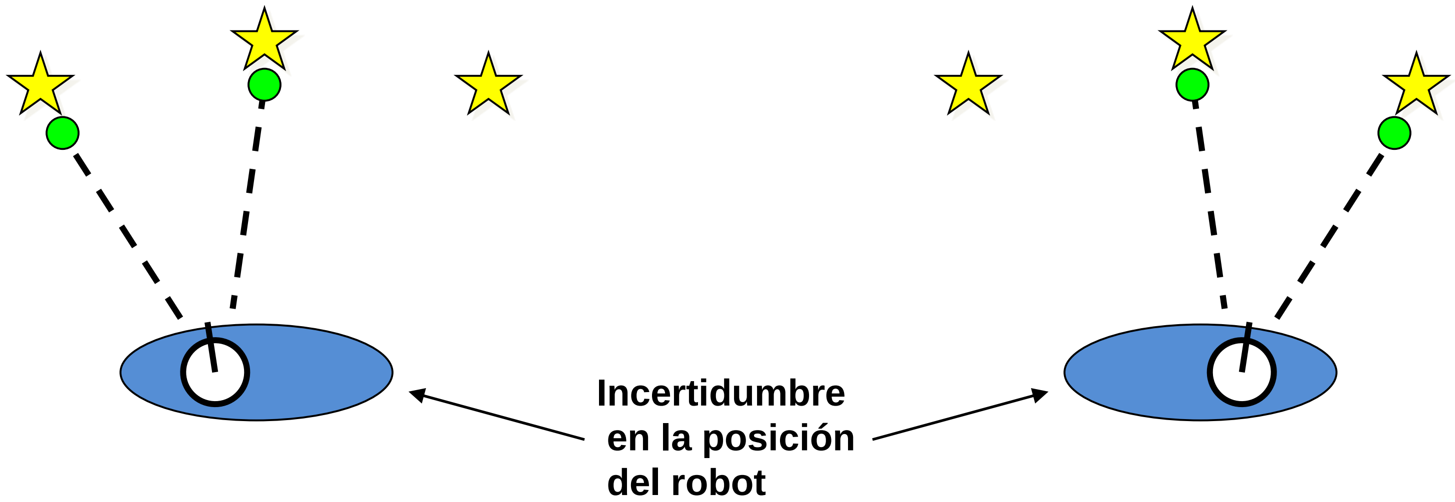
5.2 Estructura del problema del SLAM

SLAM: el camino seguido por el robot y el mapa son ambos desconocidos



El error cometido en la localización está correlado con el error del mapa

5. Aplicación al problema del SLAM

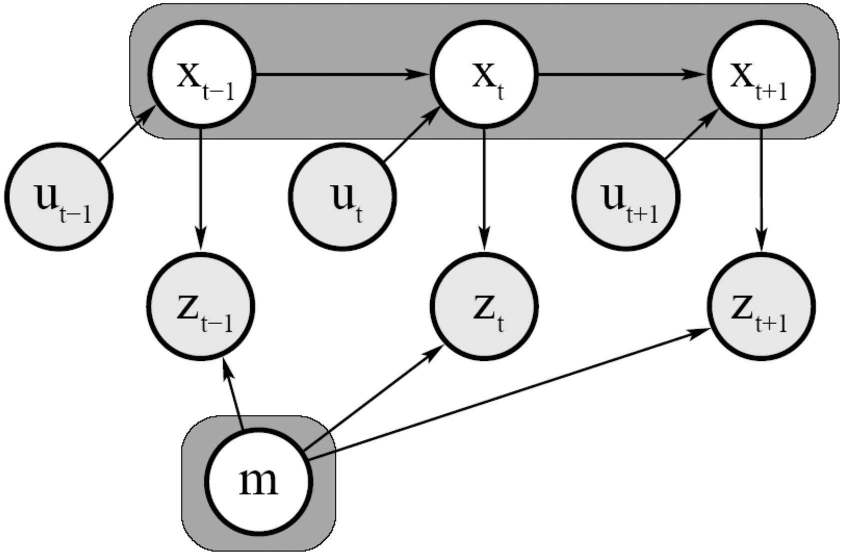


- En una aplicación real, la correspondencia entre las marcas y las observaciones no se conoce
- Hacer asociaciones incorrectas produce grandes errores en el mapa
- El error de posición está correlado con las asociaciones de datos

5.3 Tipos de problemas de SLAM

- Full SLAM: estima el mapa y la trayectoria completa del robot

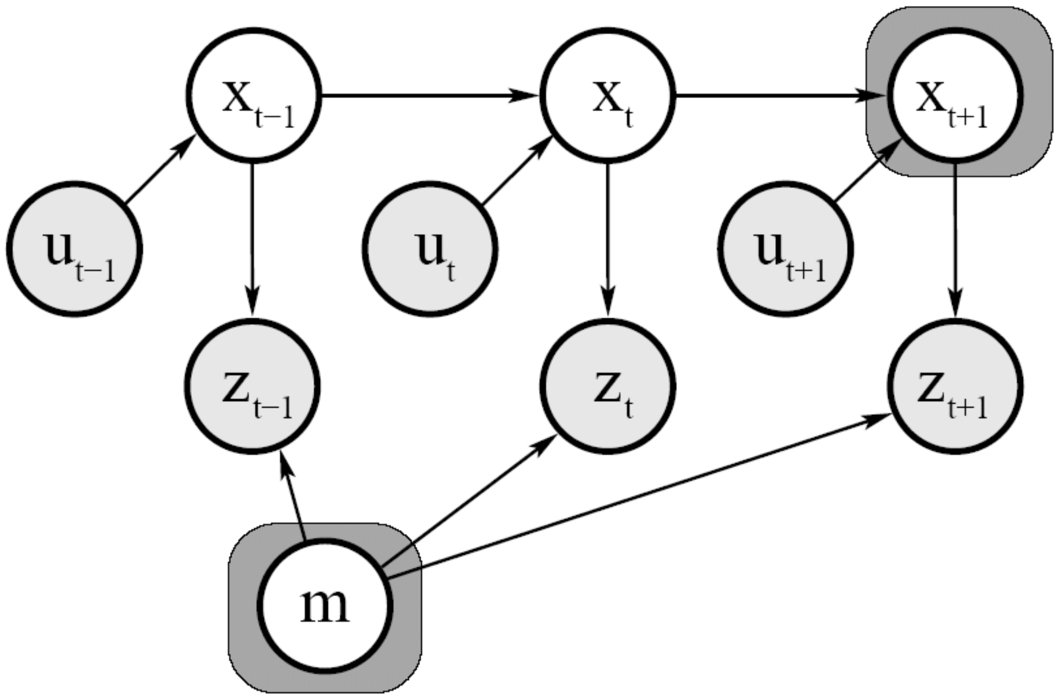
$$p(x_{1:t}, m \mid z_{1:t}, u_{1:t})$$



5. Aplicación al problema del SLAM

- Online SLAM: estima el mapa y la posición actual del robot

$$p(x_t, m \mid z_{1:t}, u_{1:t}) = \int\int \ldots \int p(x_{1:t}, m \mid z_{1:t}, u_{1:t})\, dx_1\, dx_2 \ldots dx_{t-1}$$



5.4 Técnicas de SLAM

- Scan matching
- EKF SLAM
- Fast-SLAM
- Graph-SLAM
- SEIFs
- iSAM, etc.

EKF SLAM

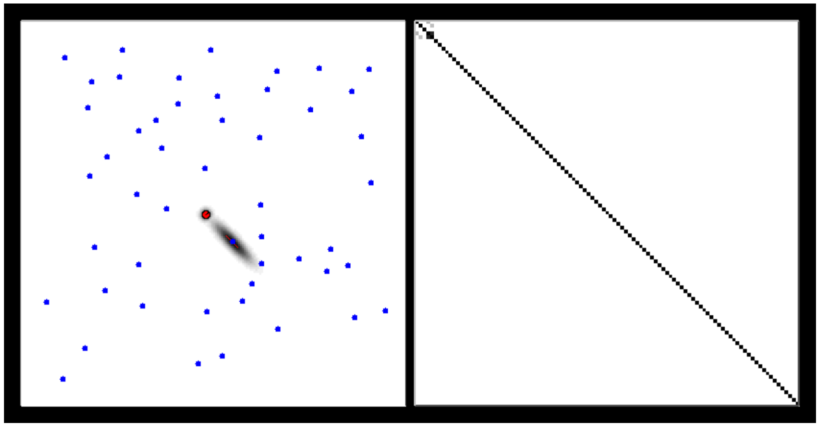
- Se añaden las marcas al vector de estados:

$$Bel(x_t, m_t) = \left\langle \begin{pmatrix} x \\ y \\ \theta \\ l_1 \\ l_2 \\ \vdots \\ l_N \end{pmatrix}, \begin{pmatrix} \sigma_x^2 & \sigma_{xy} & \sigma_{x\theta} & \sigma_{xl_1} & \sigma_{xl_2} & \cdots & \sigma_{xl_N} \\ \sigma_{xy} & \sigma_y^2 & \sigma_{y\theta} & \sigma_{yl_1} & \sigma_{yl_2} & \cdots & \sigma_{yl_N} \\ \sigma_{x\theta} & \sigma_{y\theta} & \sigma_\theta^2 & \sigma_{\theta l_1} & \sigma_{\theta l_2} & \cdots & \sigma_{\theta l_N} \\ \sigma_{xl_1} & \sigma_{yl_1} & \sigma_{\theta l_1} & \sigma_{l_1}^2 & \sigma_{l_1 l_2} & \cdots & \sigma_{l_1 l_N} \\ \sigma_{xl_2} & \sigma_{yl_2} & \sigma_{\theta l_2} & \sigma_{l_1 l_2} & \sigma_{l_2}^2 & \cdots & \sigma_{l_2 l_N} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \sigma_{xl_N} & \sigma_{yl_N} & \sigma_{\theta l_N} & \sigma_{l_1 l_N} & \sigma_{l_2 l_N} & \cdots & \sigma_{l_N}^2 \end{pmatrix} \right\rangle$$

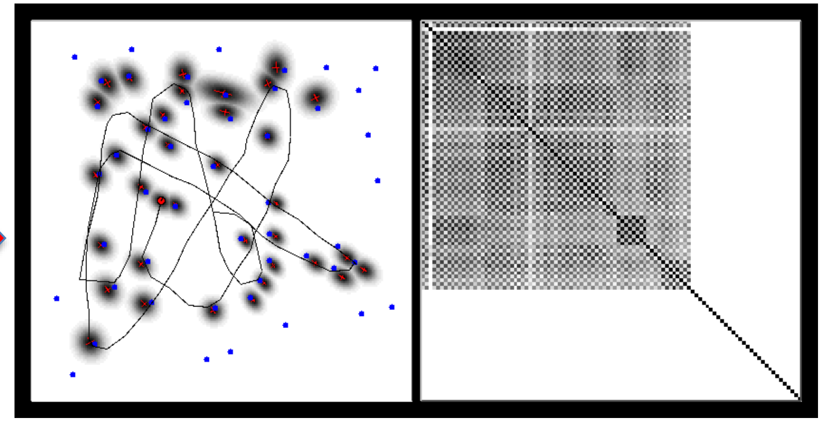
- Sólo puede manejar el orden de pocos cientos de marcas

5. Aplicación al problema del SLAM

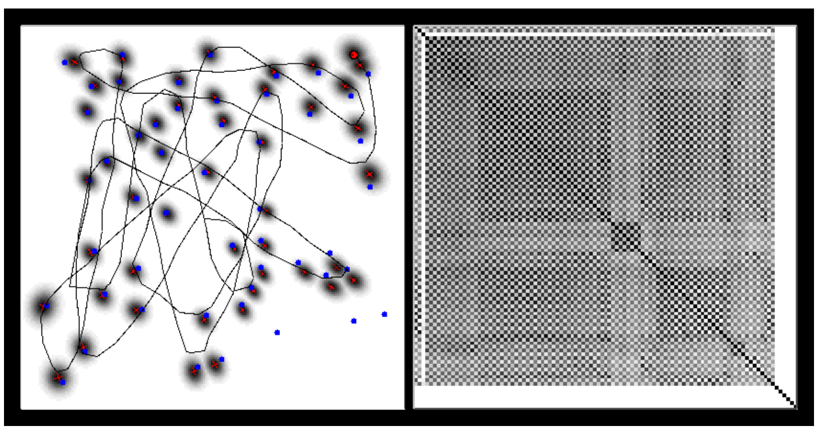
Ejemplo de evolución del EKF SLAM:



Mapa Matriz de covarianza



Mapa Matriz de covarianza



Mapa Matriz de covarianza

RESUMEN:

- Cuadrático en función del número de marcas: $O(n^2)$
- Dificultad para converger en caso de que haya fuertes no linealidades.
- Diferentes aproximaciones permiten implementarlo en tiempo real.