

# Robótica Móvil

## un enfoque probabilístico

# SLAM – FastSLAM basado en Landmarks

Ignacio Mas

---

# SLAM

- SLAM implica estimar la pose del robot en el mapa mientras se construye el mapa del entorno.
- ¿Por qué es difícil?  
Problema del huevo y la gallina:
  - Necesito mapa par localizar y
  - Necesito una estimación de pose para hacer un mapa

# SLAM

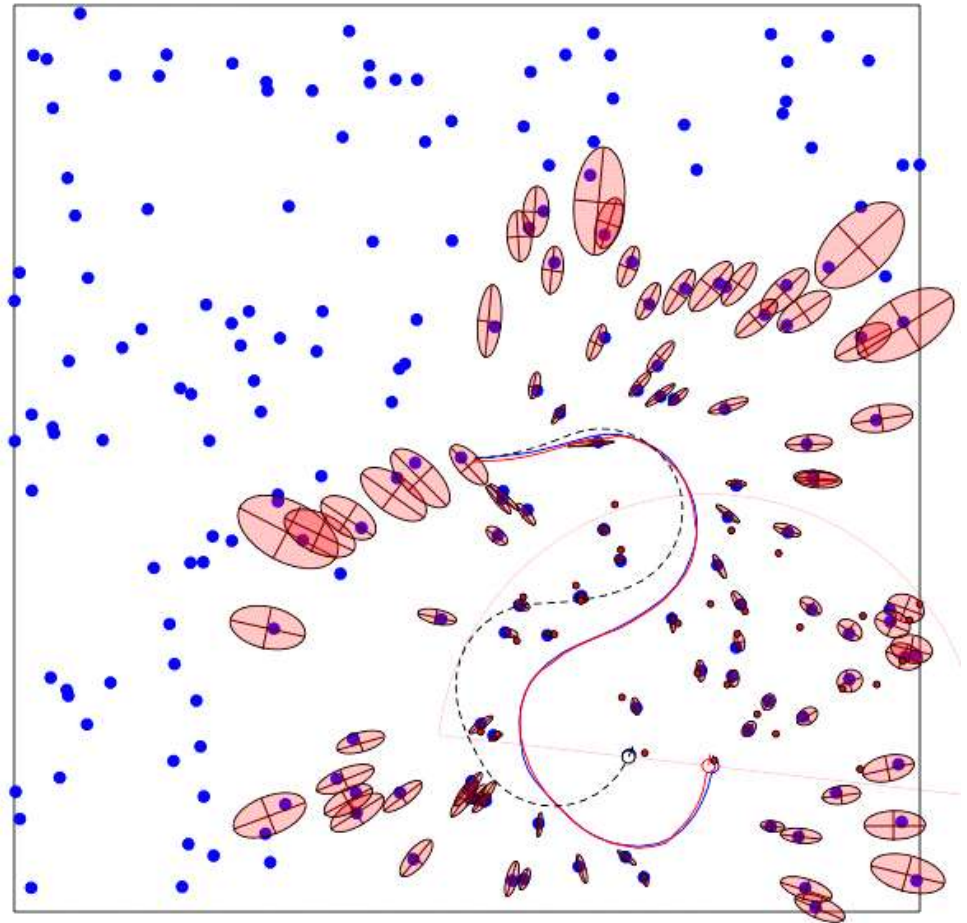
El robot se mueve en un entorno estático desconocido

## Dados:

- Las señales de control del robot
- Observaciones de features cercanas

## Estimar:

- Mapa de features
- Recorrido del robot

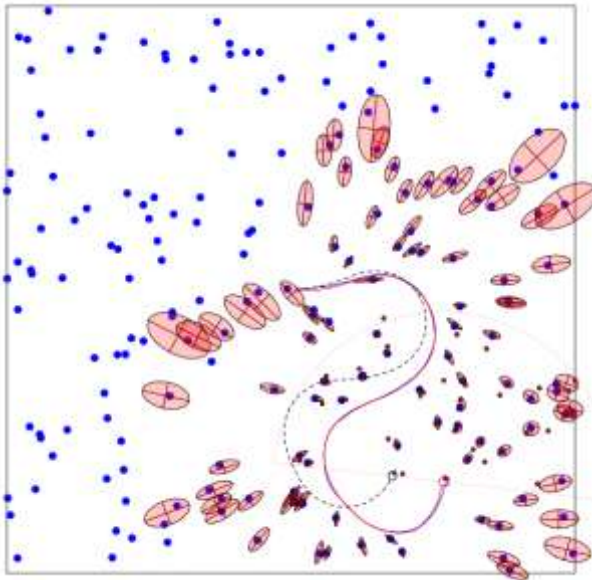


# Representaciones de Mapas

## Los modelos más comunes son:

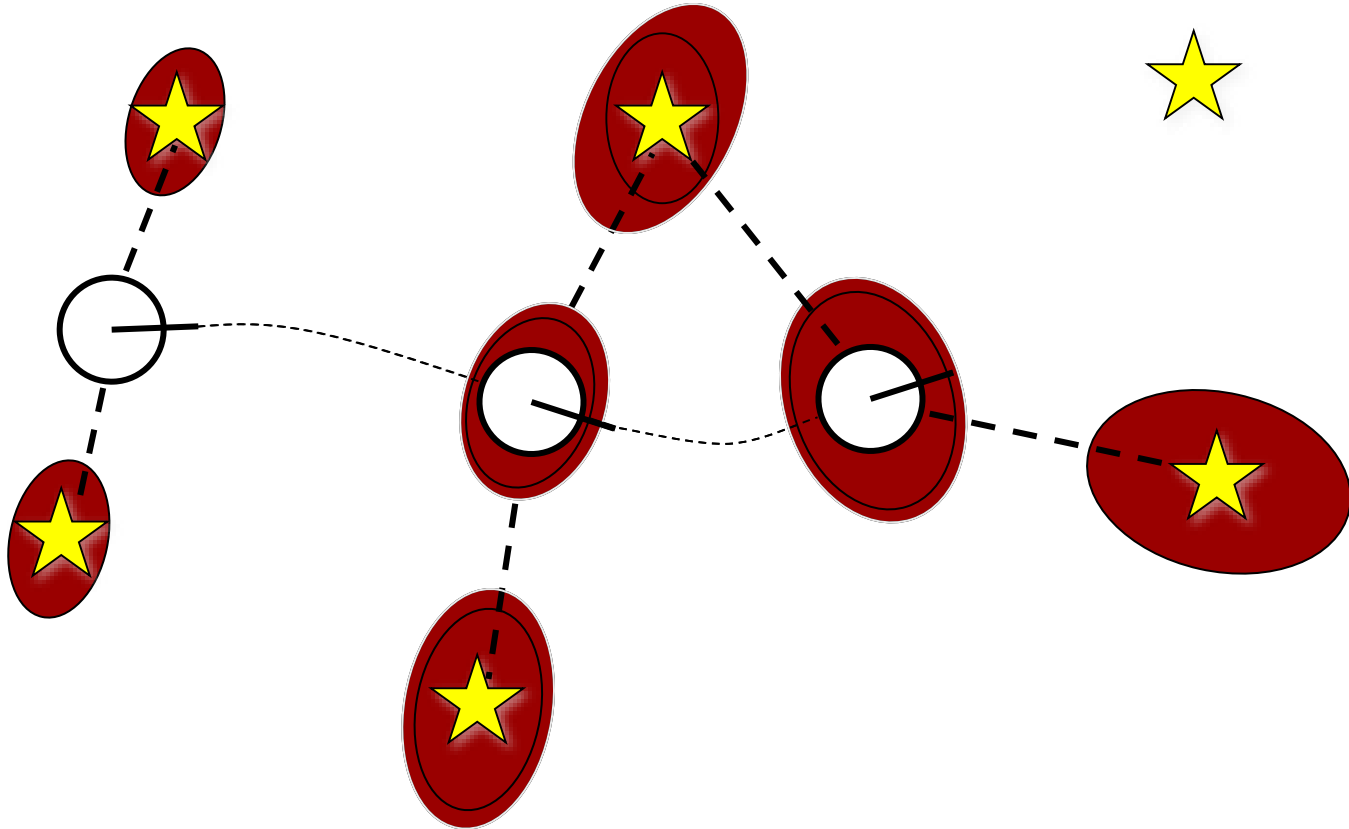
- Mapas de Features
- Mapas de grilla (mapas de ocupación o de probabilidad de reflexión)

Hoy



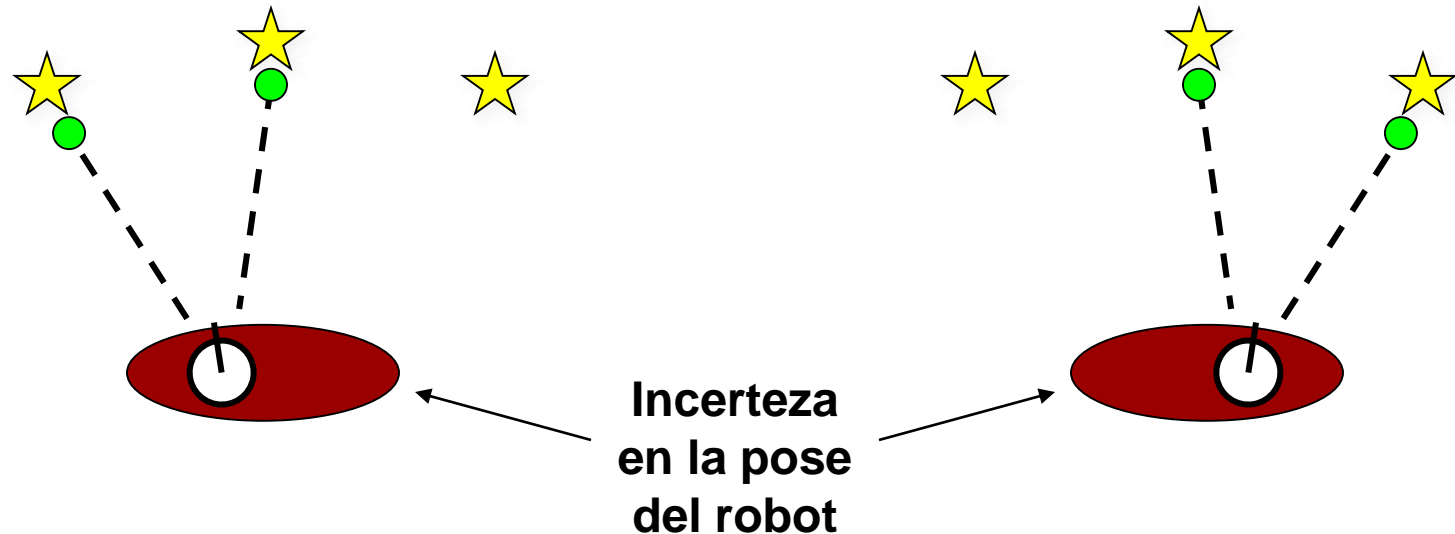
# Por qué SLAM es un problema difícil?

**SLAM:** el camino del robot y el mapa son ambos **desconocidos!**



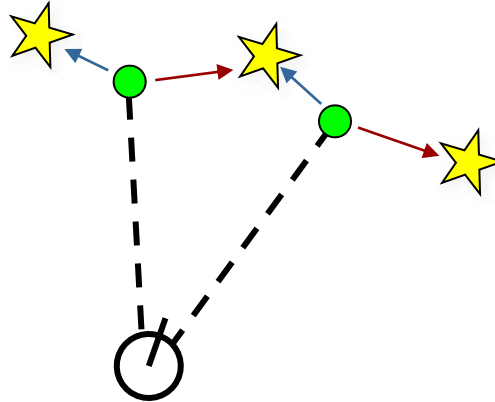
El error en la ubicación del robot se correlaciona con errores en el mapa

# Por qué SLAM es un problema difícil?



- El **mapeo entre observaciones y landmarks es desconocido**
- La asociación de datos **incorrecta** puede tener consecuencias **graves** (divergencia)
- Errores en posición dependen de asociación de datos

# Asociación de datos



- Es la asociación de observaciones con landmarks
- En general, hay más de  $\binom{n}{m}$  (n observaciones, m landmarks) asociaciones posibles
- También llamado “problema de asignación”

# Filtros de Partículas

- Representan belief con **muestras** aleatorias
- Estimación de procesos **no-lineales no-gaussianos**
- Principio de remuestreo con importancia:
  - Muestrear una nueva generación de partículas
  - Asignar un peso de importancia a cada partícula
  - Remuestrear
- Aplicaciones típicas: tracking, localización etc.



# Localización vs. SLAM

- El filtro de partículas puede usarse para resolver ambos problemas
- Localización: Espacio de estados  $\langle x, y, \theta \rangle$
- SLAM: espacio de estados  $\langle x, y, \theta, mapa \rangle$ 
  - para mapas de landmarks =  $\langle l_1, l_2, \dots, l_m \rangle$
  - para mapas de grillas =  $\langle c_{11}, c_{12}, \dots, c_{1n}, c_{21}, \dots, c_{nm} \rangle$
- **Problema:** ¡El número de partículas necesarias para representar el belief crece exponencialmente con la dimensión del espacio de estado!

# Dependencias


- ¿Existen dependencias entre ciertas dimensiones del espacio de estado?
- Si es así, ¿pueden aprovecharse para resolver el problema de forma más eficiente?

# Dependencias

- ¿Existen dependencias entre ciertas dimensiones del espacio de estado?
- Si es así, ¿pueden aprovecharse para resolver el problema de forma más eficiente?
- En el contexto de SLAM
  - El mapa depende de las poses del robot.
  - Sabemos como construir un mapa si conocemos las poses del robot (sensor).

# Distribución posterior factorizada (Landmarks)

poses    mapa    observaciones & acciones


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

# Distribución posterior factorizada (Landmarks)

poses    mapa    observaciones & acciones

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

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

The diagram includes several red arrows pointing from the text above to the equation: from 'poses' to  $x_{1:t}$ , from 'mapa' to  $l_{1:m}$ , from 'observaciones' to  $z_{1:t}$ , and from 'acciones' to  $u_{0:t-1}$ . Black arrows point upwards from the text below to the equation: from 'Posterior de SLAM' to  $x_{1:t}$ , from 'Posterior de trayecto del robot' to  $z_{1:t}$ , and from 'posiciones de landmarks' to  $l_{1:m}$ .

Posterior de SLAM

Posterior de trayecto del robot

posiciones de landmarks

**¿Cómo ayuda esto a resolver el problema?**

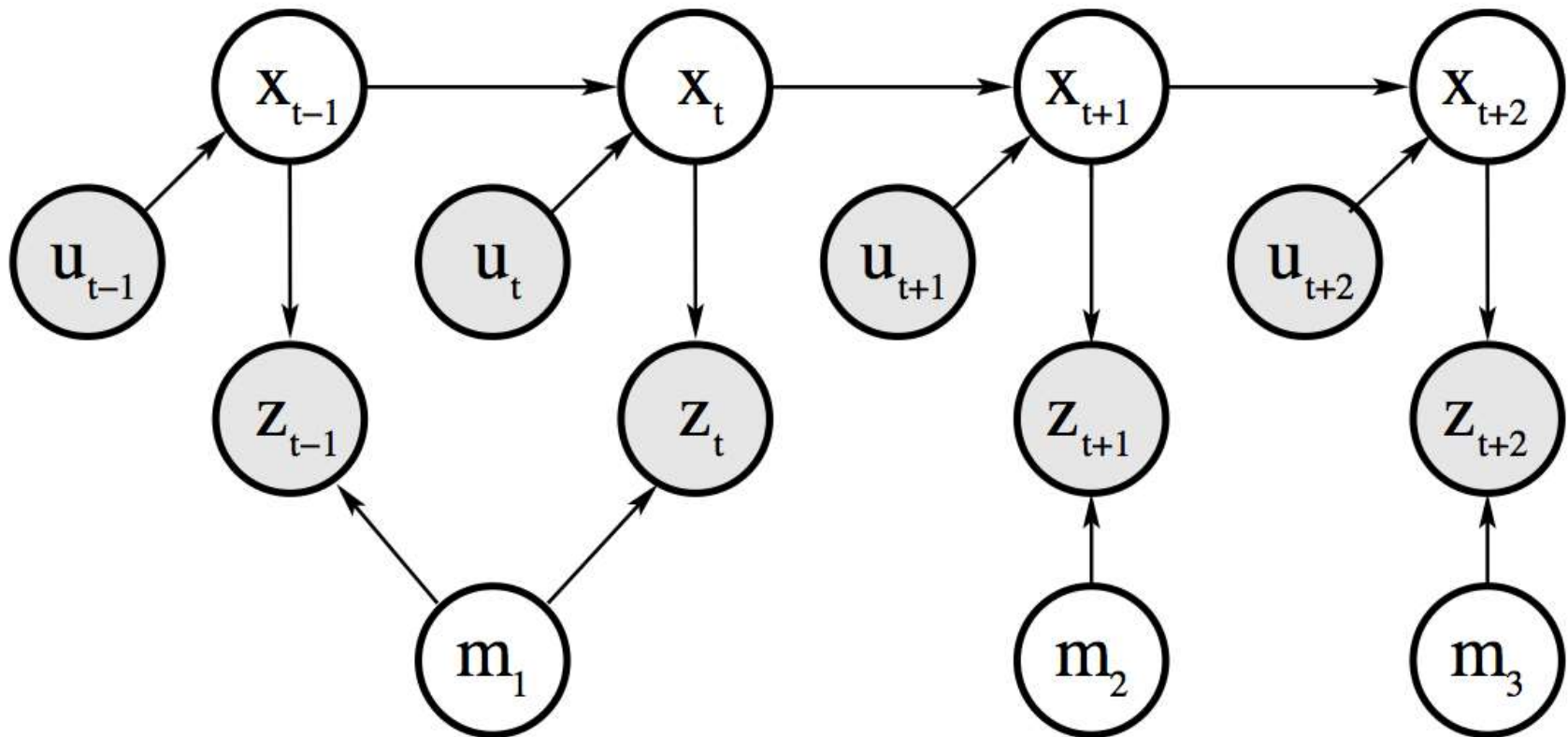
# Rao-Blackwellización

- Factorización para aprovechar la dependencia entre variables:

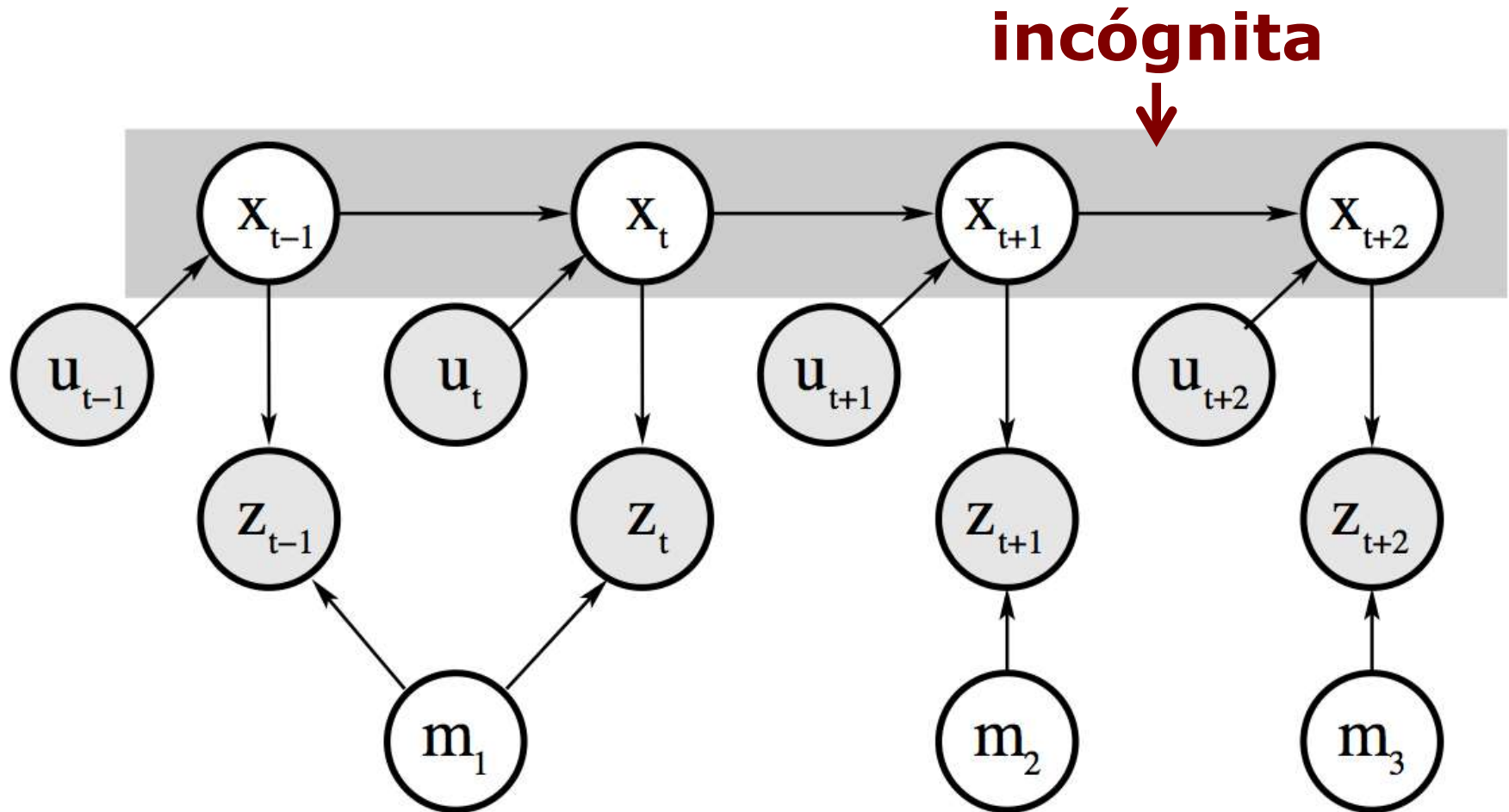
$$p(a, b) = p(a) \cdot p(b | a)$$

- Si  $p(b | a)$  puede calcularse en forma cerrada, representar sólo  $p(a)$  con muestras y calcular  $p(b | a)$  para cada muestra
- El nombre viene del teorema de Rao-Blackwell

# Recordando el modelo gráfico

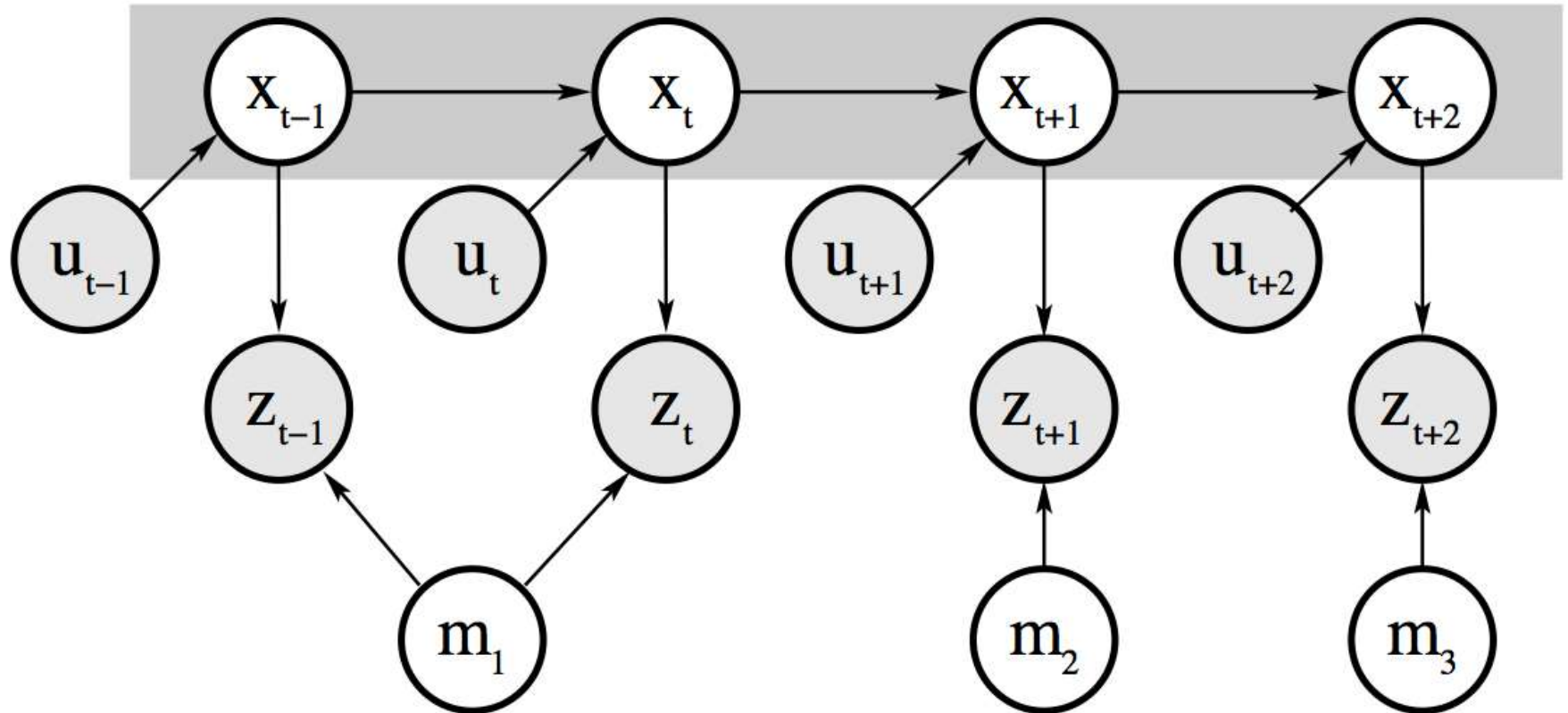


# Recordando el modelo gráfico





# Los landmarks son condicionalmente independientes dadas las poses



**Los landmarks están desconectados (son independientes) dada la tray. del robot**

# Posterior factorizada

$$\begin{aligned} & p(x_{1:t}, l_{1:m} \mid z_{1:t}, u_{0:t-1}) \\ &= p(x_{1:t} \mid z_{1:t}, u_{0:t-1}) \cdot p(l_{1:m} \mid x_{1:t}, z_{1:t}) \\ &= p(x_{1:t} \mid z_{1:t}, u_{0:t-1}) \cdot \prod_{i=1}^M p(l_i \mid x_{1:t}, z_{1:t}) \end{aligned}$$

Posterior de la  
trayectoria de robot  
(problema de  
localización)

Posiciones de  
landmarks  
condicionalmente  
independientes

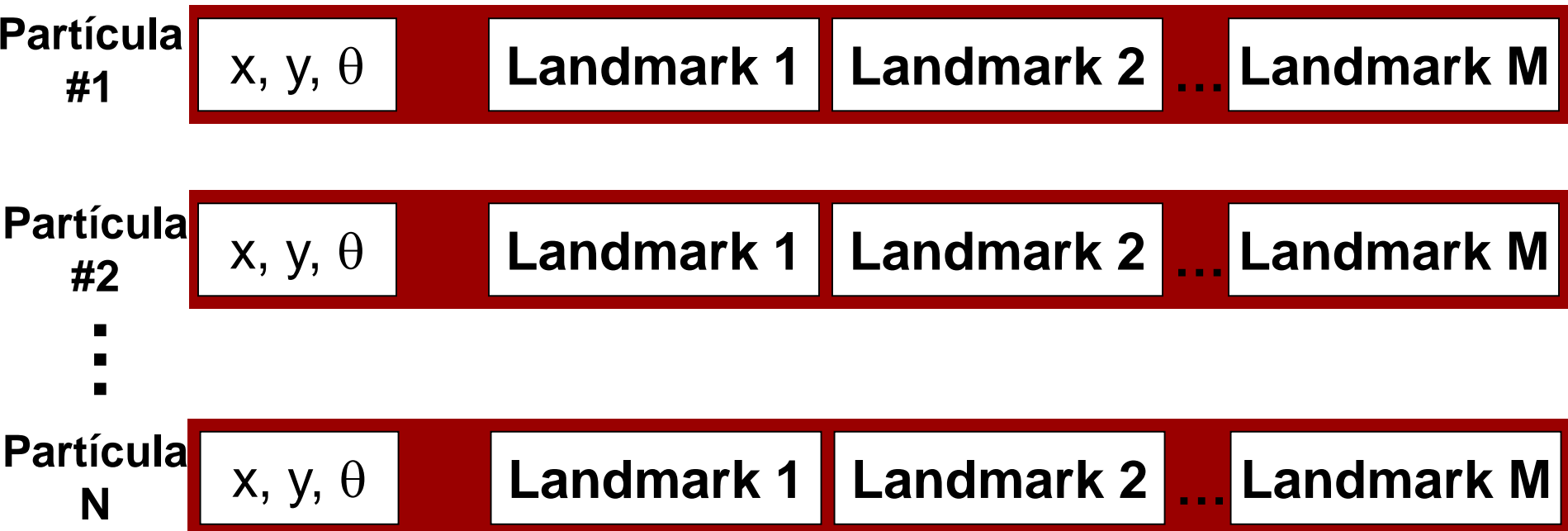
# Rao-Blackwellización para SLAM

$$p(x_{1:t}, l_{1:m} \mid z_{1:t}, u_{0:t-1}) = p(x_{1:t} \mid z_{1:t}, u_{0:t-1}) \cdot \prod_{i=1}^M p(l_i \mid x_{1:t}, z_{1:t})$$

- Dado que el segundo término puede calcularse eficientemente, es posible usar filtros de partículas!

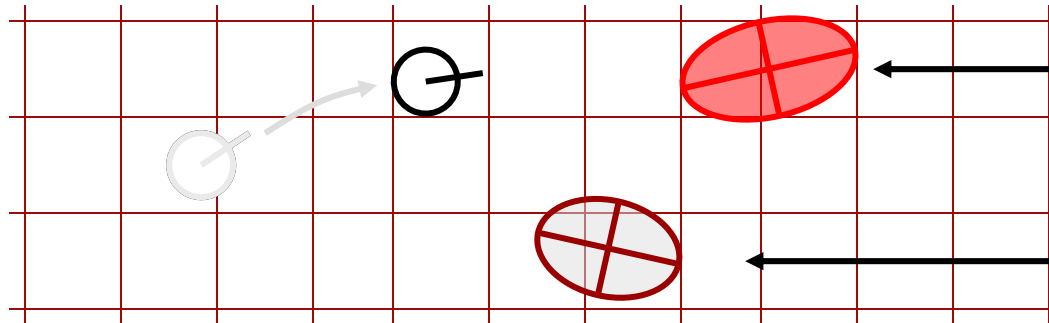
# FastSLAM

- Filtro de partículas Rao-Blackwellizado basado en landmarks [Montemerlo et al., 2002]
- Cada landmark se representa con un Filtro de Kalman Extendido (EKF) de 2x2
- Cada partícula lleva consigo  $M$  EKFs



# FastSLAM – Movimiento

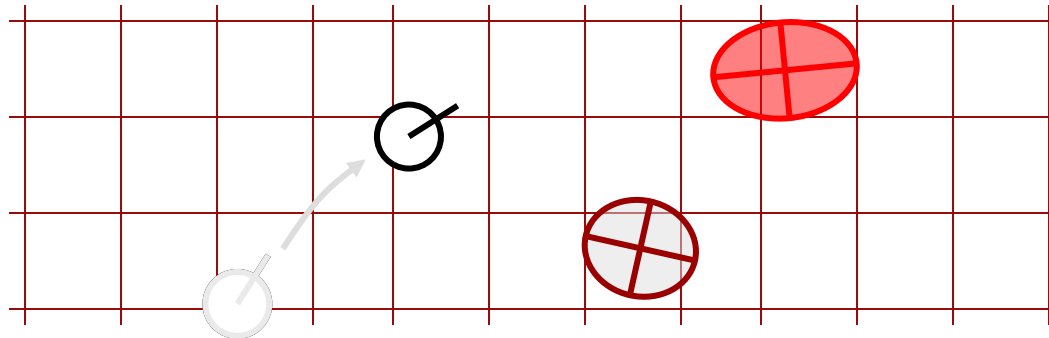
Partícula #1



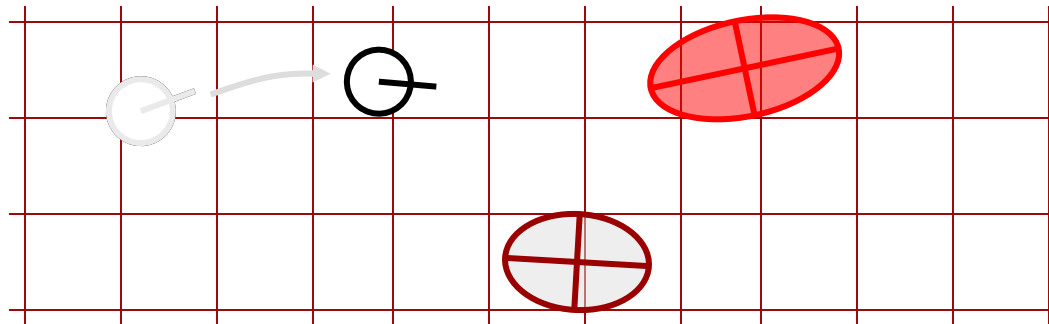
Filtro de  
Landmark #1

Filtro de  
Landmark #2

Partícula #2

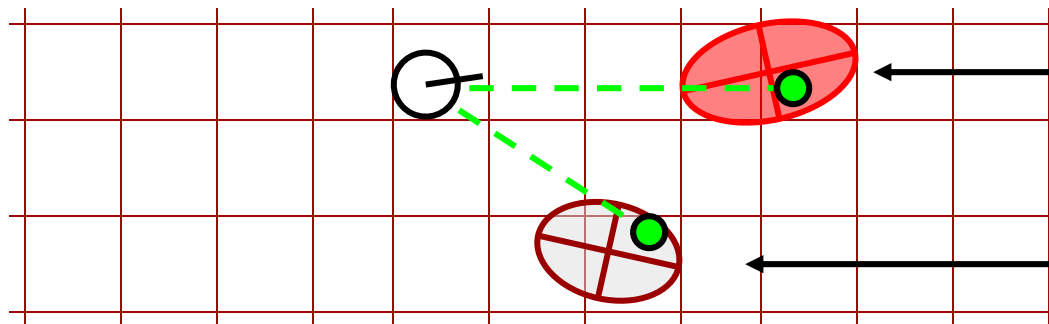


Partícula #3

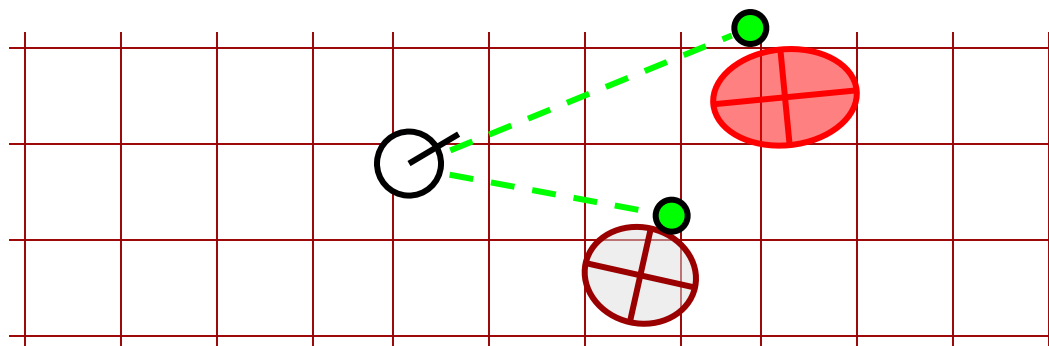


# FastSLAM – Observación

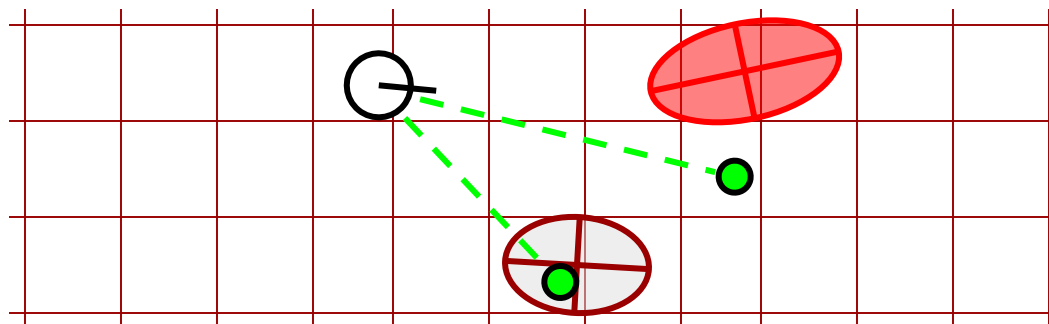
Partícula #1



Partícula #2

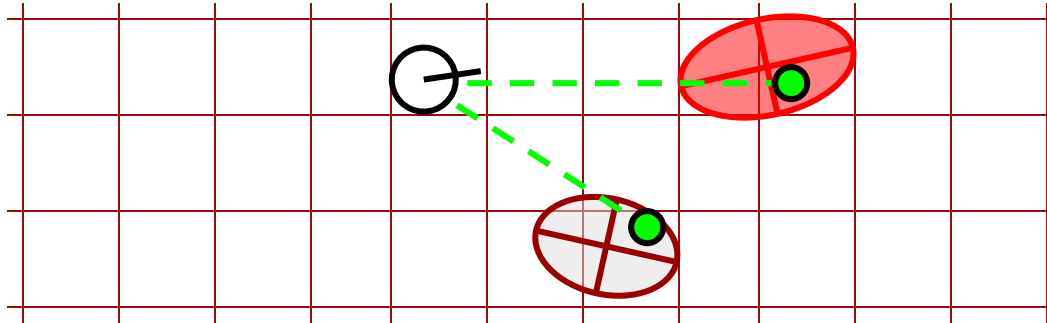


Partícula #3



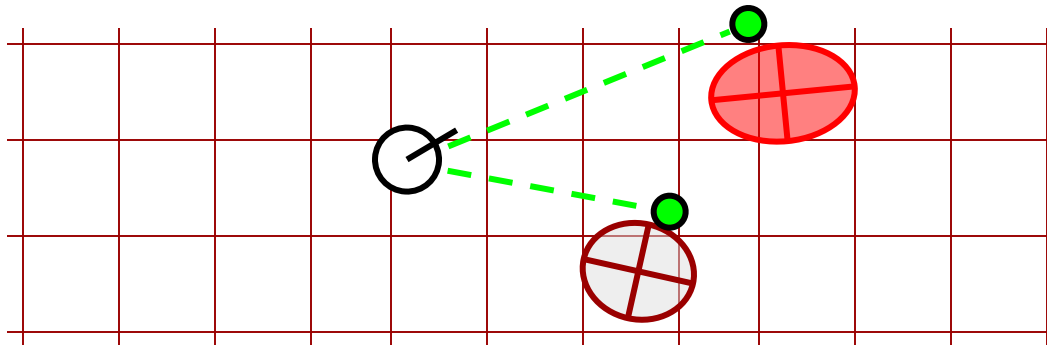
# FastSLAM – Actualiz. de sensor

Partícula #1



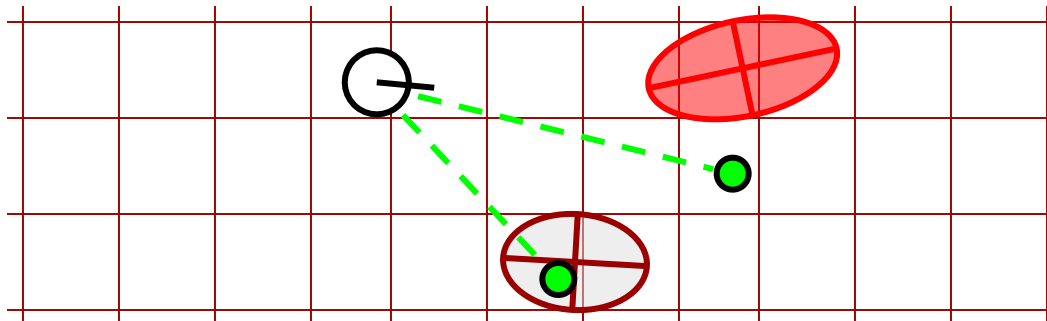
Peso = 0.8

Partícula #2



Peso = 0.4

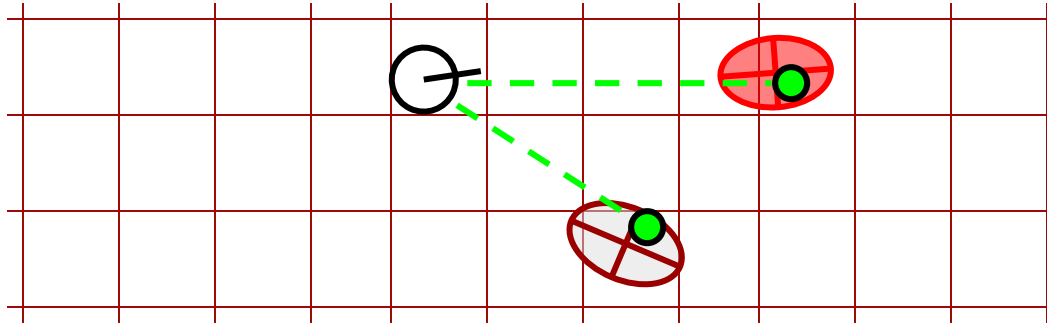
Partícula #3



Peso = 0.1

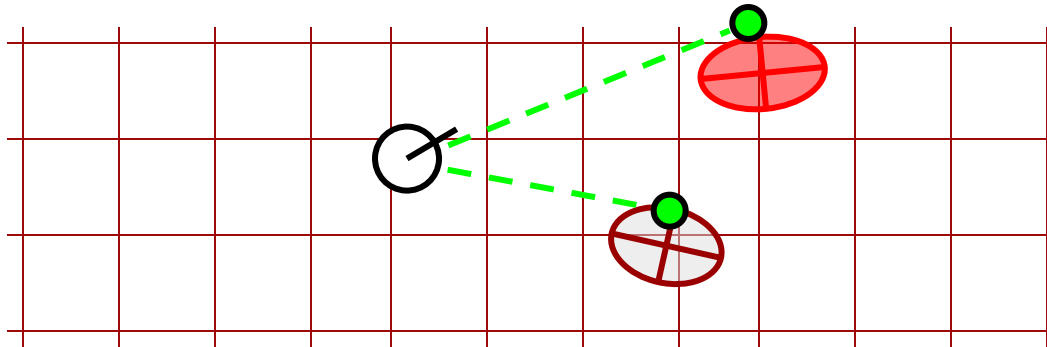
# FastSLAM – Actualiz. de sensor

Partícula #1



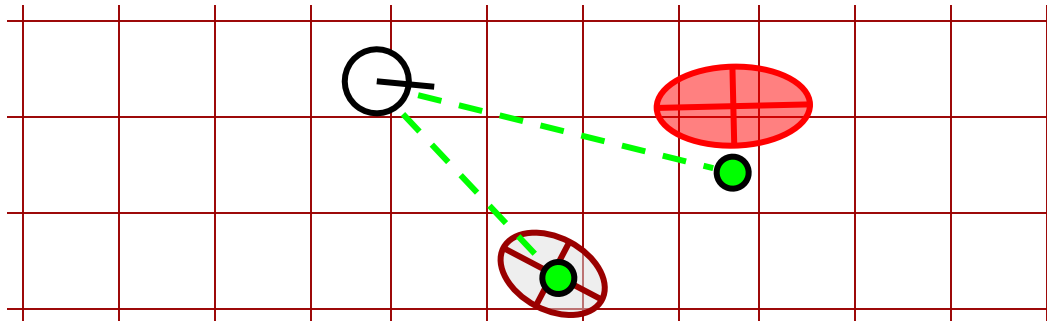
Actualización  
mapa de  
partícula #1

Partícula #2



Actualización  
mapa de  
partícula #2

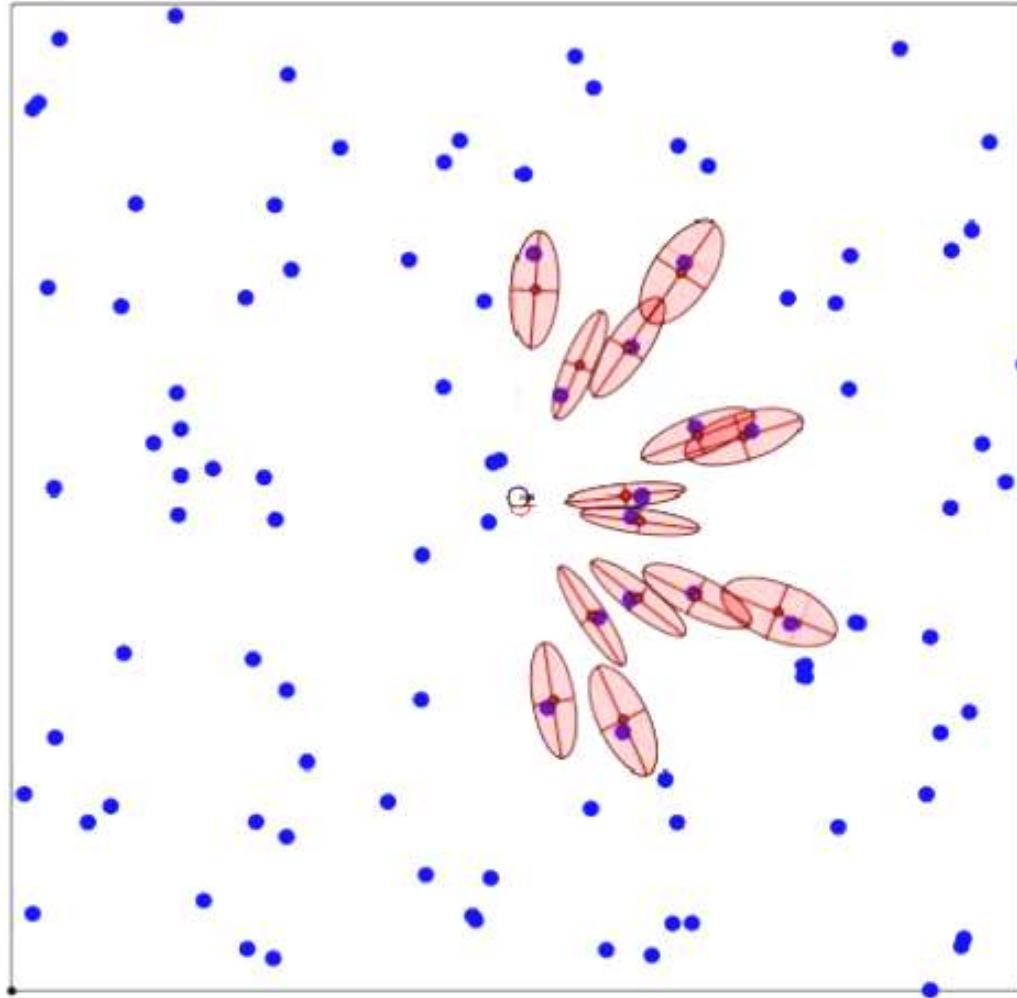
Partícula #3



Actualización  
mapa de  
partícula #3



# FastSLAM - Video



# Complejidad de FastSLAM

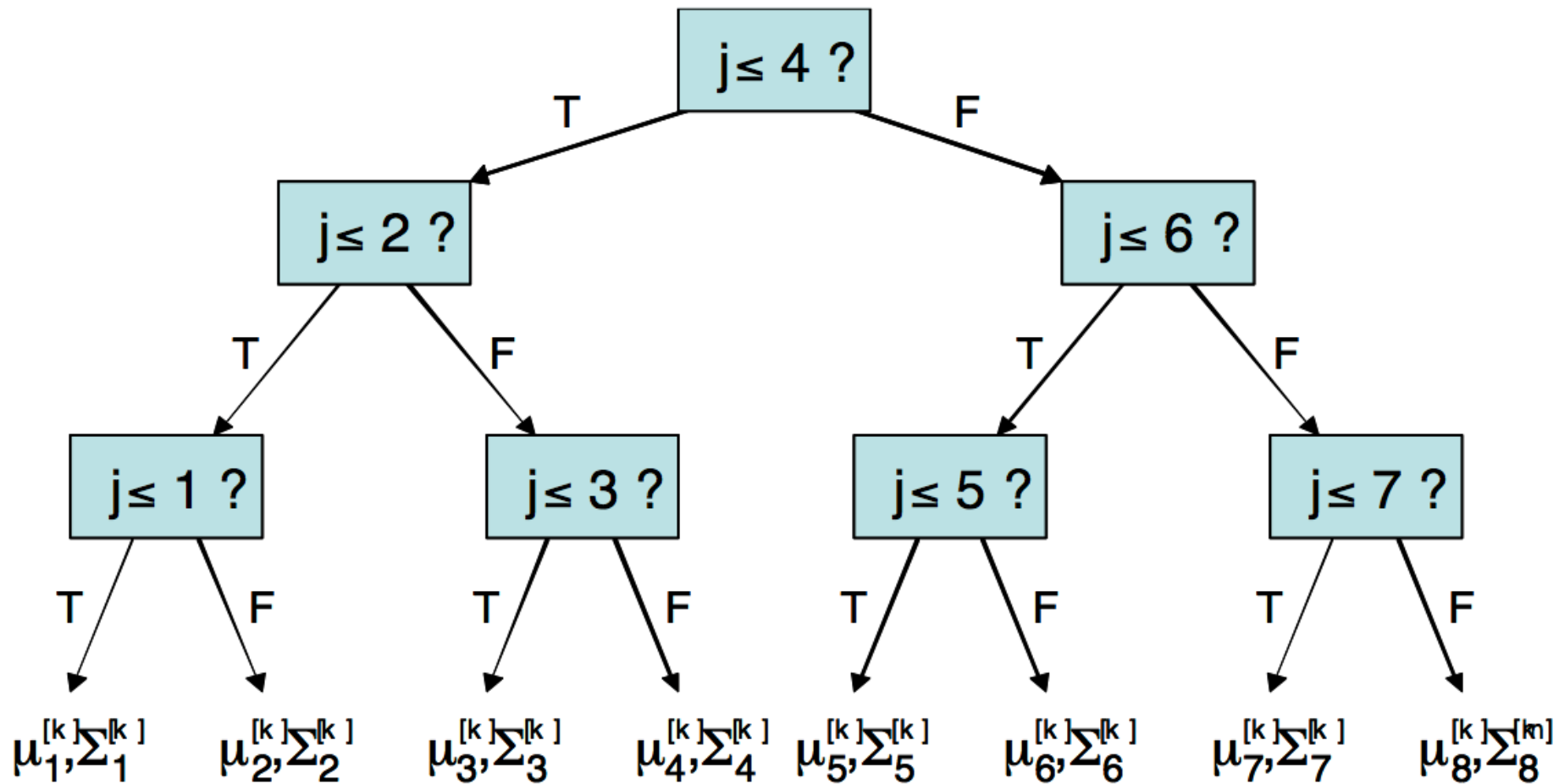
- Actualizar partículas (robot) basándose en el control  $\mathcal{O}(N)$
- Incorporar una observación en los Filtros de Kalman (dada la asociación de datos)  $\mathcal{O}(N)$
- Remuestrear el conjunto de partículas  $\mathcal{O}(NM)$

**N = Número de partículas**  
**M = Número de landmarks**

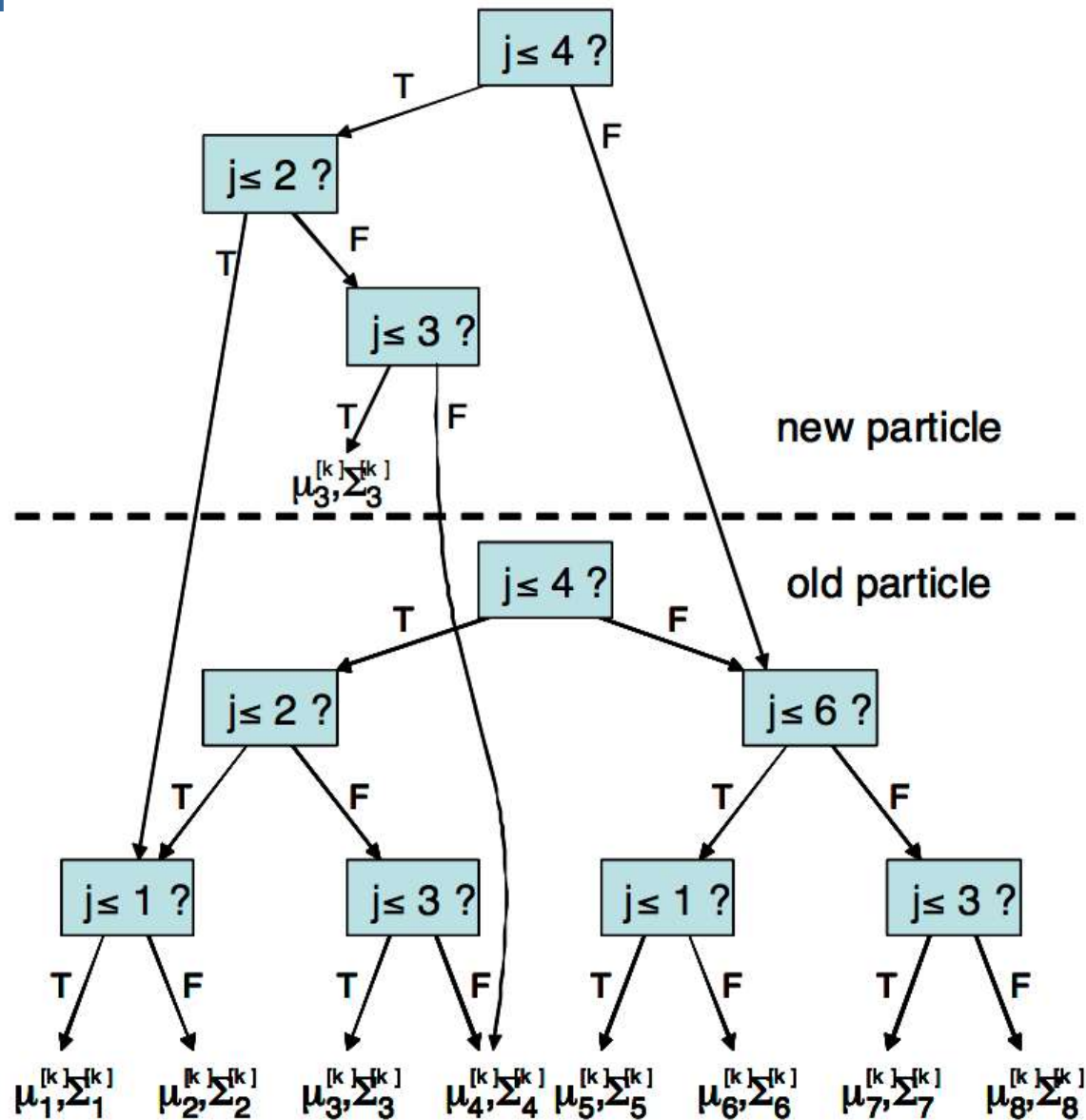
---

$$\mathcal{O}(NM)$$

# Mejorando la estructura de datos para FastSLAM



# Mejorando la estructura de datos para FastSLAM



# Complejidad de FastSLAM

- Actualizar partículas (robot) basándose en el control  $\mathcal{O}(N)$
- Incorporar una observación en los Filtros de Kalman (dada la asociación de datos)  $\mathcal{O}(N \log M)$
- Remuestrear el conjunto de partículas  $\mathcal{O}(N \log M)$

**N = Número de partículas**

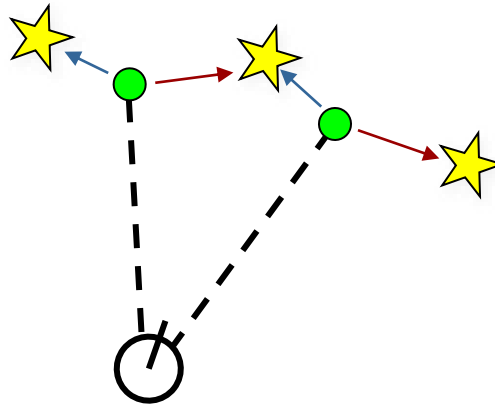
**M = Número de landmarks**

---

$$\mathcal{O}(N \log M)$$

# Asociación de datos

- ¿Qué observación pertenece a qué landmark?



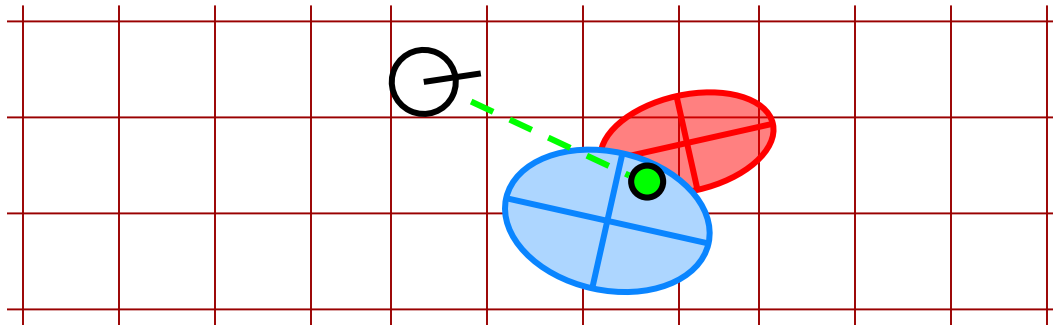
- Una solución robusta de SLAM debe considerar las posibles asociaciones de datos
- Las potenciales asociaciones también dependen de la pose del robot

# Asociación de datos Multi-Hipótesis

- La asociación de datos se hace para cada partícula
- El error de la pose del robot se elimina de las decisiones de asociación de datos



# Asociación de datos por partícula



¿La observación fue generada por el landmark rojo o azul?

$$P(\text{observación}|\text{rojo}) = 0.3 \quad P(\text{observación}|\text{azul}) = 0.7$$

- Dos opciones para la asociación por partícula
  - Elegir el más probable
  - Elegir una asociación aleatoria pesada por el likelihood de observaciones
- Si la probabilidad es muy baja, generar una nueva landmark



# Resultados – Victoria Park

- 4 km de recorrido
- $< 5$  m RMS error de posición
- 100 partículas

Azul = GPS

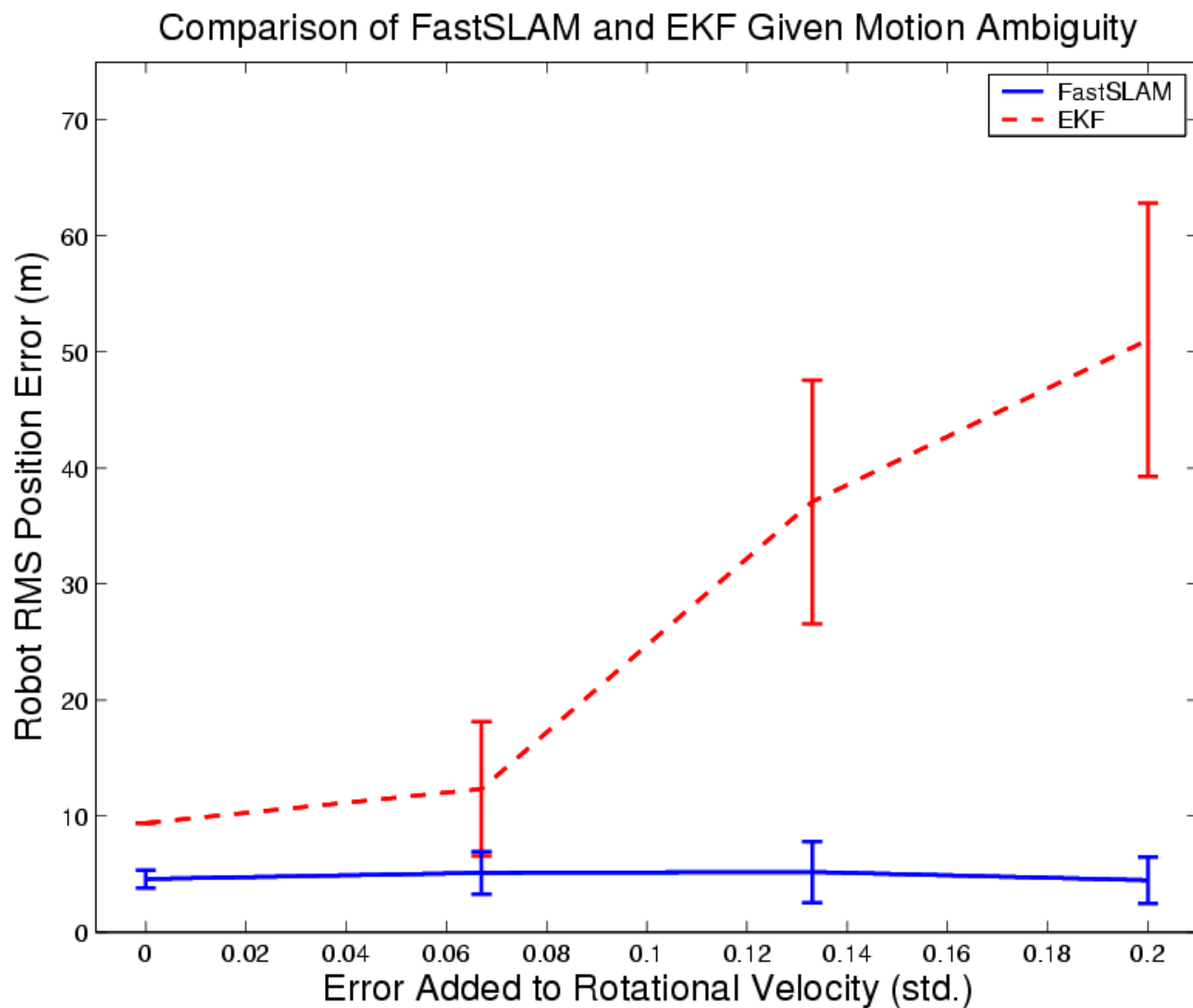
Amarillo = FastSLAM



# Resultados – Victoria Park (Video)



# Resultados – Errores



# Resumen FastSLAM

- FastSLAM permite calcular la posterior del SLAM en problemas de estimación de baja dimensión
  - Escala a casos con más de 1 millón de features
- FastSLAM permite sacar la incerteza de la pose del robot del problema de asociación de datos
  - Robusto a ambigüedades en asociación de datos
  - Permite que la decisión de asociación se retrase hasta que se colecte evidencia que no es ambigua
- Ventajas sobre método EKF clásico (especialmente con no-linealidades)
- Complejidad de  $O(N \log M)$