

Robótica Móvil un enfoque probabilístico

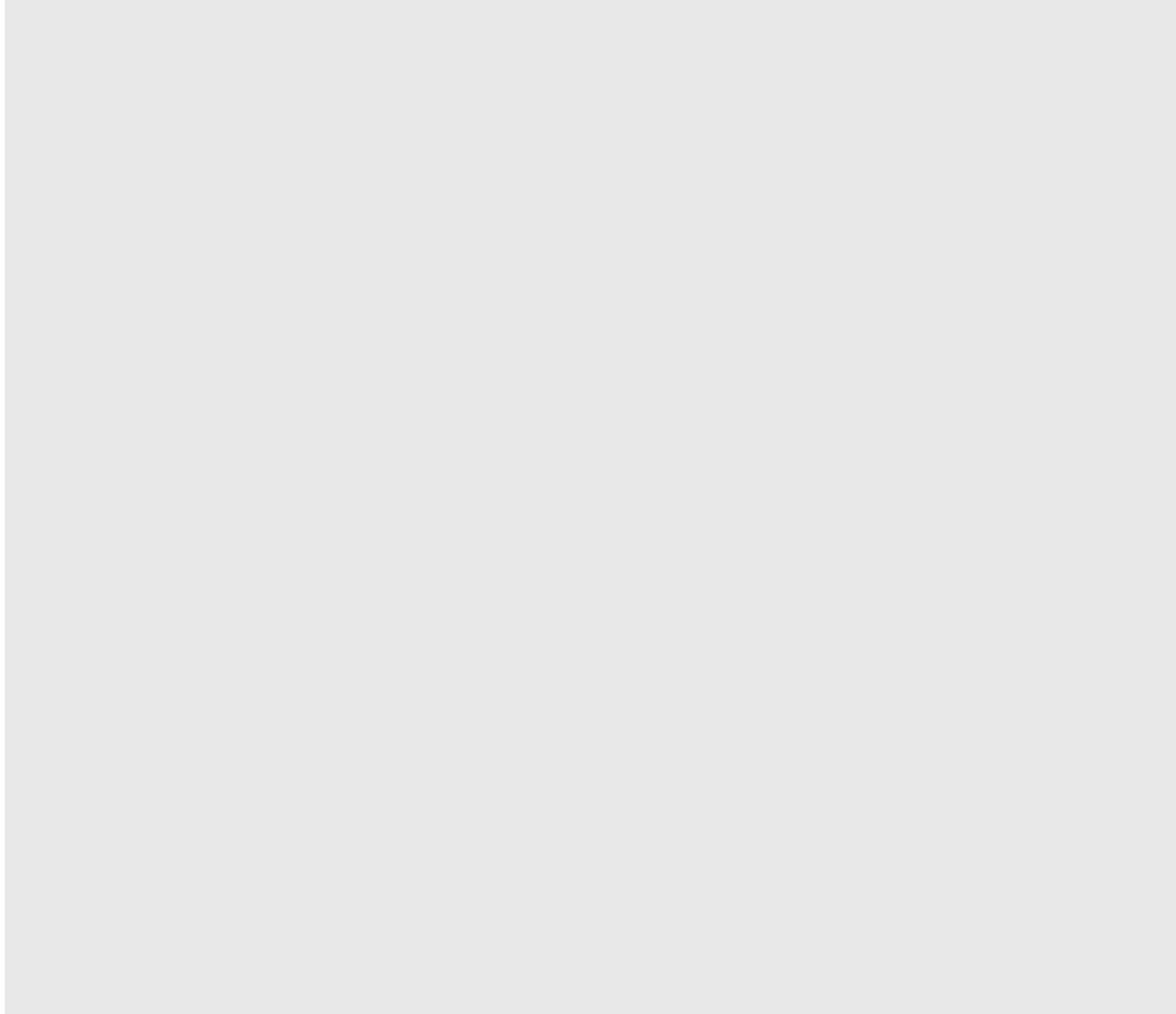
SLAM – FastSLAM basado en grillas

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

Mapeo usando odometría cruda




SLAM basado en grillas

- Se puede resolver el SLAM si **no hay landmarks** predefinidas disponibles?
- Se pueden usar los conceptos de FastSLAM para construir un **mapa de grilla**?
- Al igual que con landmarks, el mapa **depende de las poses** del robot al adquirir datos
- Si las poses son conocidas, el mapeo basado en grillas es fácil (“mapeo con poses conocidas”)

Rao-Blackwellización

poses mapa observaciones & acciones


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

Rao-Blackwellización

poses mapa observaciones & acciones

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

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

The diagram includes red arrows pointing from the text 'poses' to $x_{1:t}$, from 'mapa' to m , from 'observaciones' to $z_{1:t}$, and from 'acciones' to $u_{0:t-1}$.

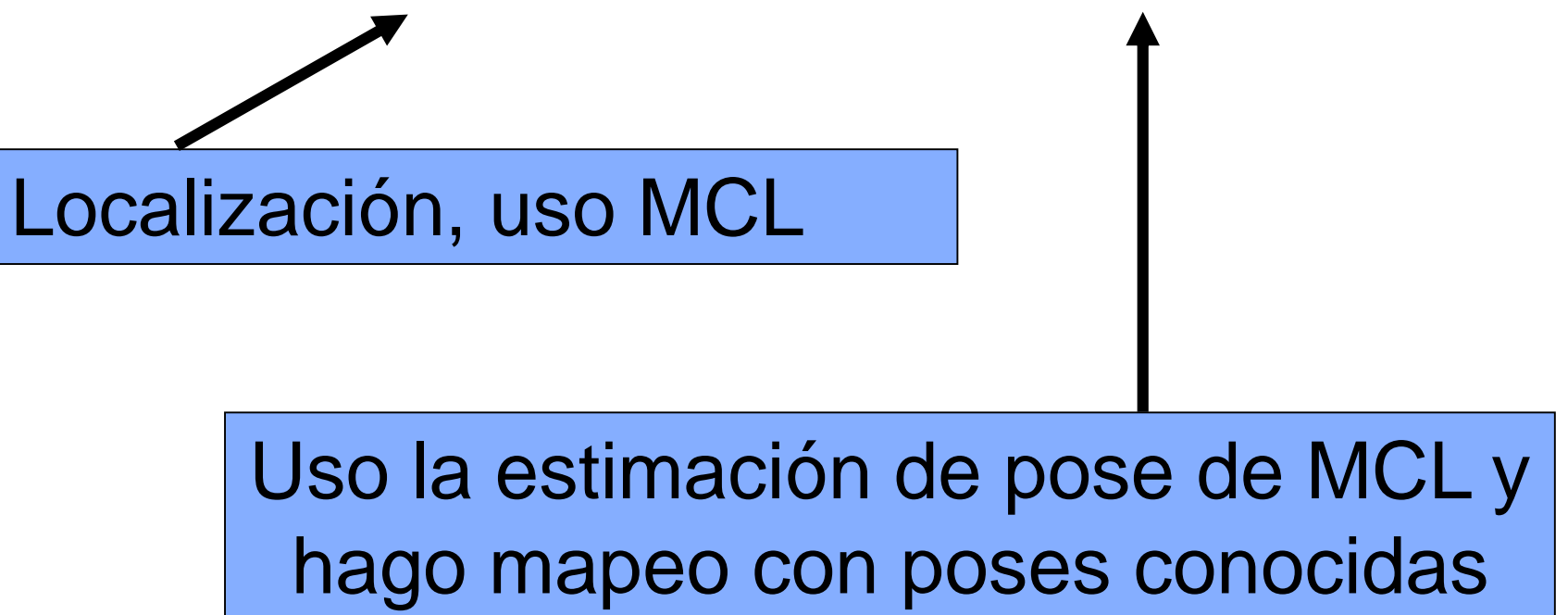
↑
Posterior de SLAM

↑
Localización del robot

↑
Mapeo con poses conocidas

Rao-Blackwellización

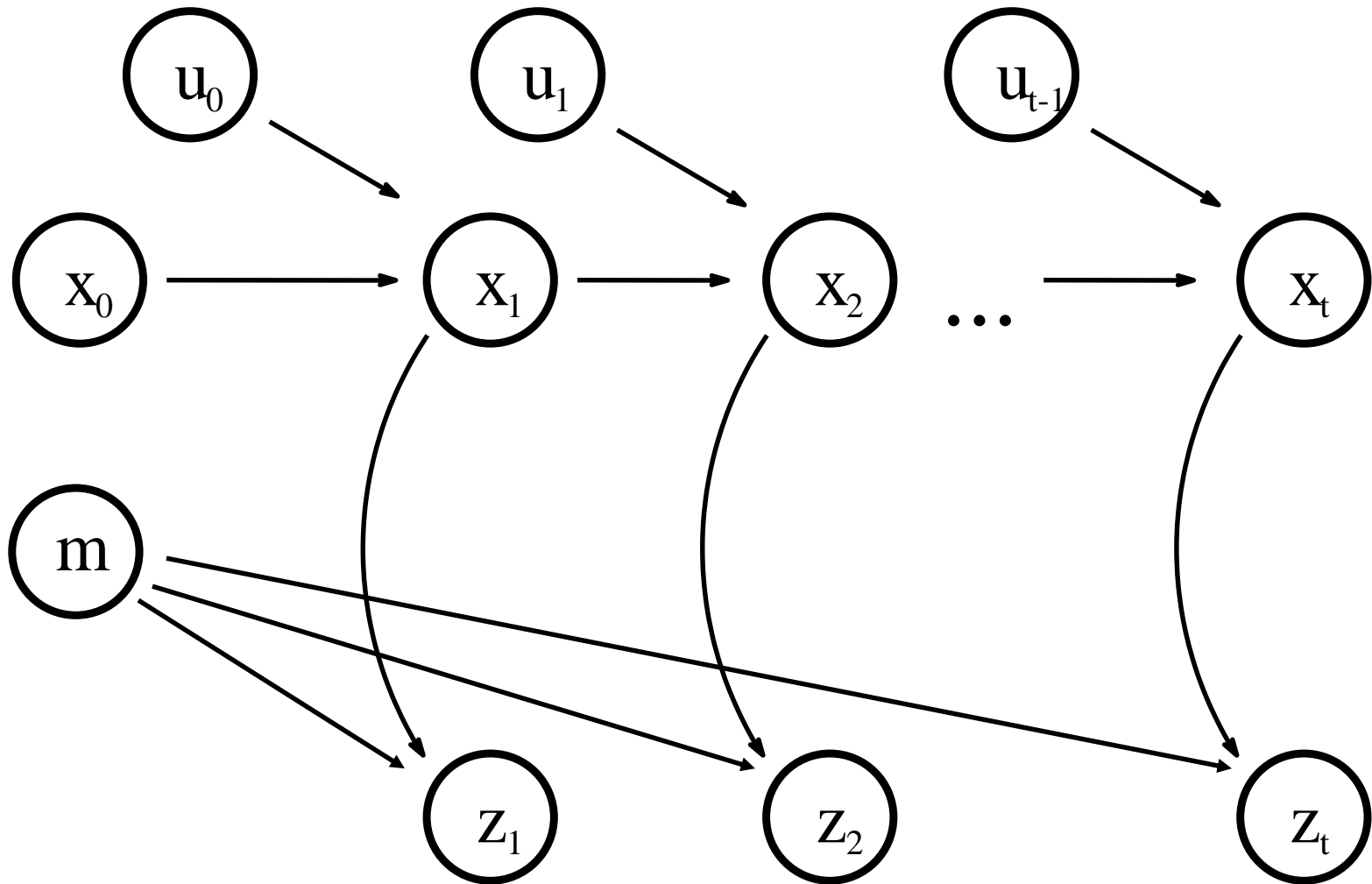
$$p(x_{1:t}, m \mid z_{1:t}, u_{0:t-1}) = p(x_{1:t} \mid z_{1:t}, u_{0:t-1}) \cdot p(m \mid x_{1:t}, z_{1:t})$$



Localización, uso MCL

Uso la estimación de pose de MCL y
hago mapeo con poses conocidas

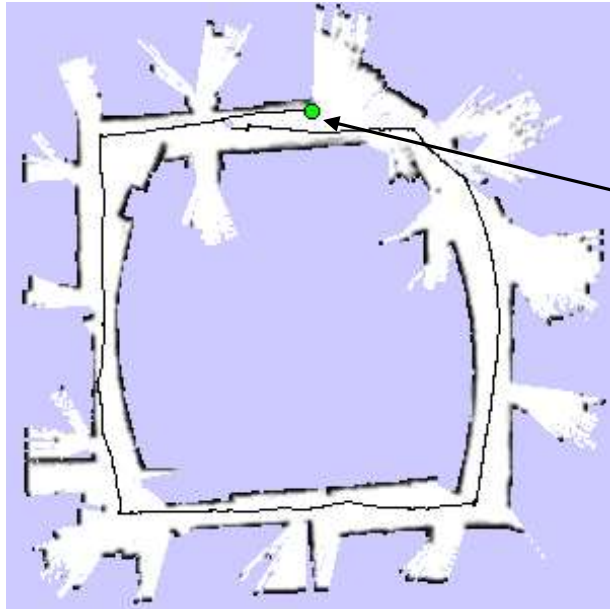
Modelo gráfico de mapeo con filtros de partículas Rao-Blackwellizadas



Mapeo con filtros de partículas Rao-Blackwellizadas

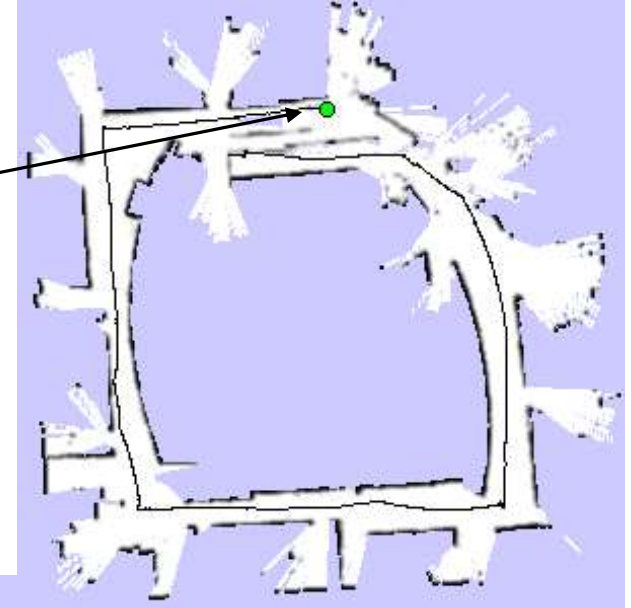
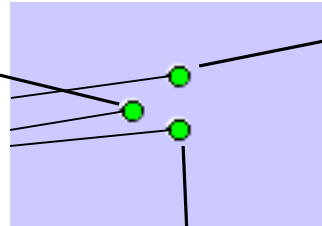
- Cada partícula representa una posible trayectoria del robot
- Cada partícula:
 - Mantiene su propio mapa y
 - Lo actualiza según un “mapeo con poses conocidas”
- Cada partícula sobrevive con una probabilidad proporcional al likelihood de las observaciones **relativas a su propio mapa**

Ejemplo de Filtro de Partículas

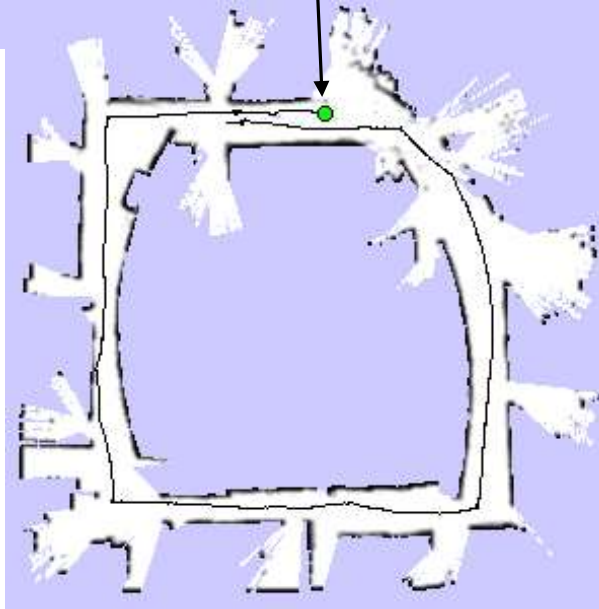


mapa de partícula 1

3 partículas



mapa de partícula 3



mapa de partícula 2

Limitaciones

- Cada **mapa es muy grande** al usar mapas de grilla
- Cada partícula mantiene su propio mapa, por lo que el **número de partículas** debe ser **bajo**
- **Solución:**
Calcular mejores distribuciones propuestas!
- **Idea:**
Mejorar la estimación de la pose **antes** de aplicar el filtro de partículas

Corrección de pose usando macheo de escaneo (Scan-Matching)

Maximizar el likelihood de la pose y mapa en el tiempo (i) relativos a la pose y mapa en el tiempo ($i-1$)

$$\hat{x}_t = \underset{x_t}{\operatorname{argmax}} \{ p(z_t \mid x_t, \hat{m}_{t-1}) \cdot p(x_t \mid u_{t-1}, \hat{x}_{t-1}) \}$$

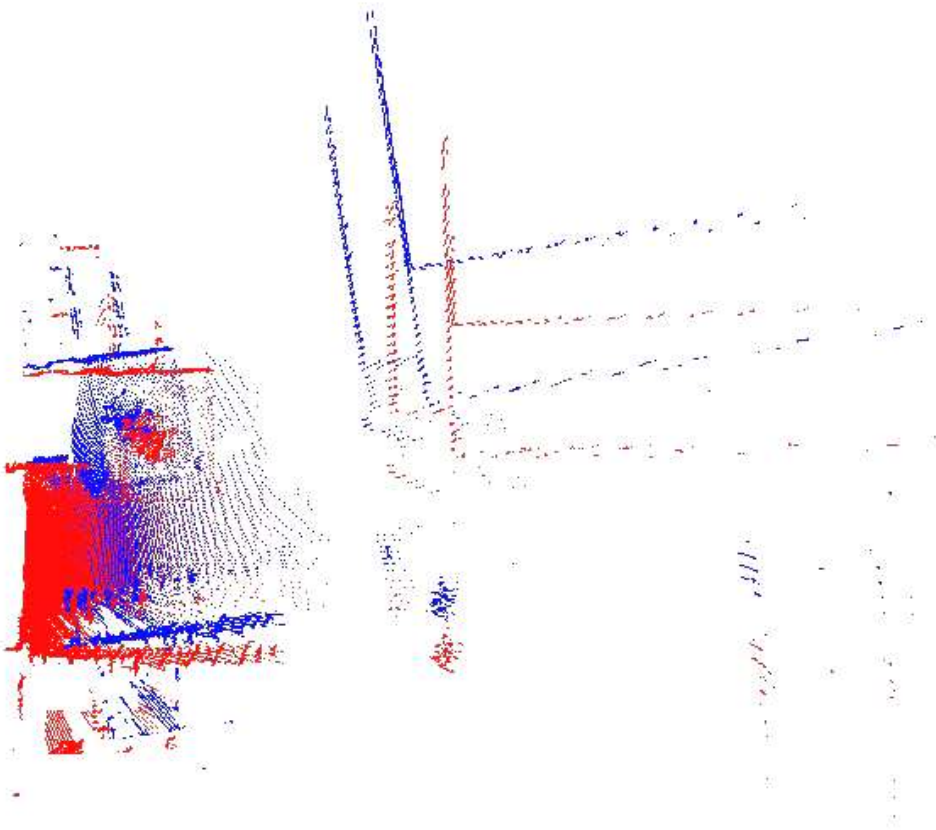
Medición actual



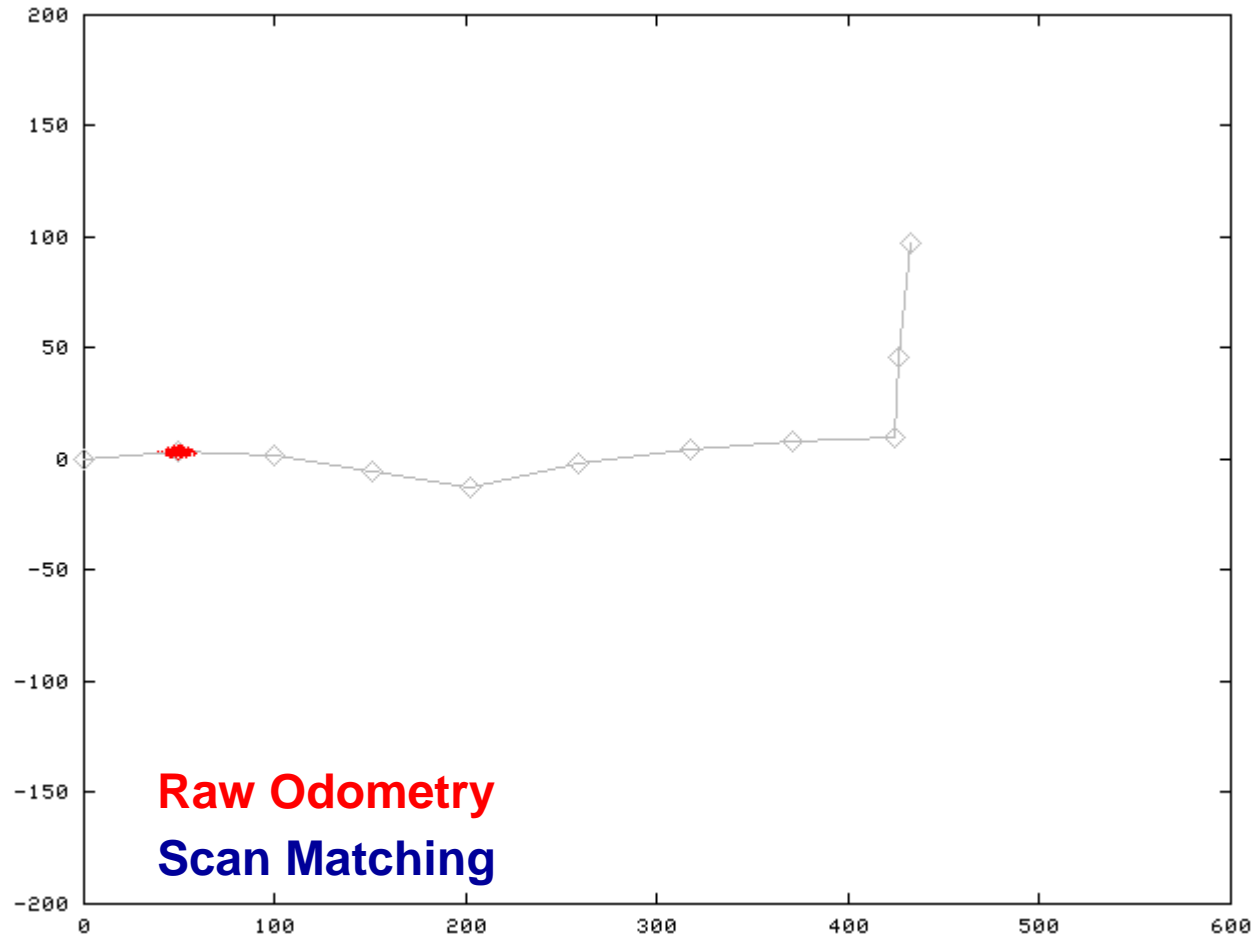
Acción de robot

mapa construido hasta ese momento

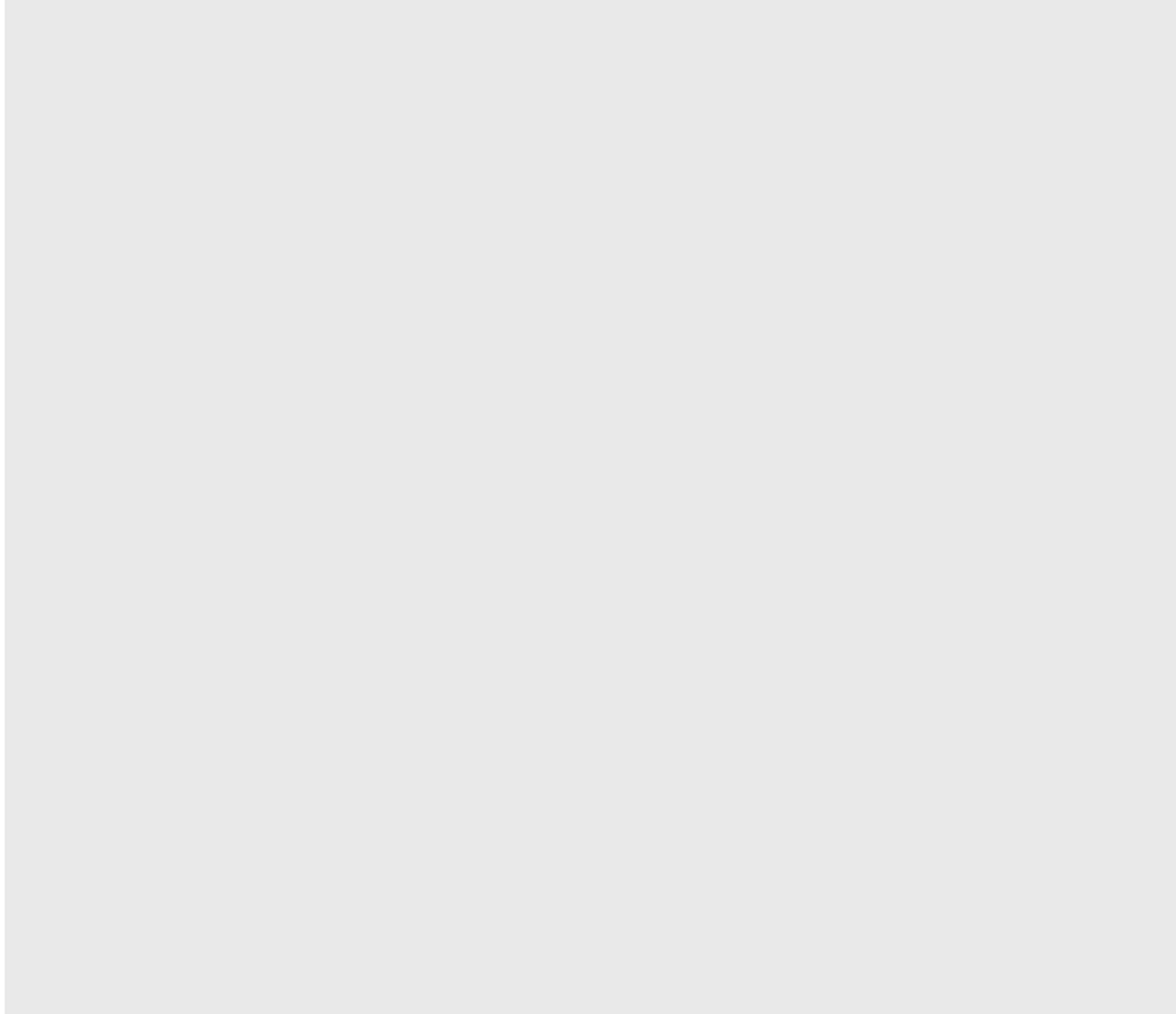
Ejemplo de Scan-Matching



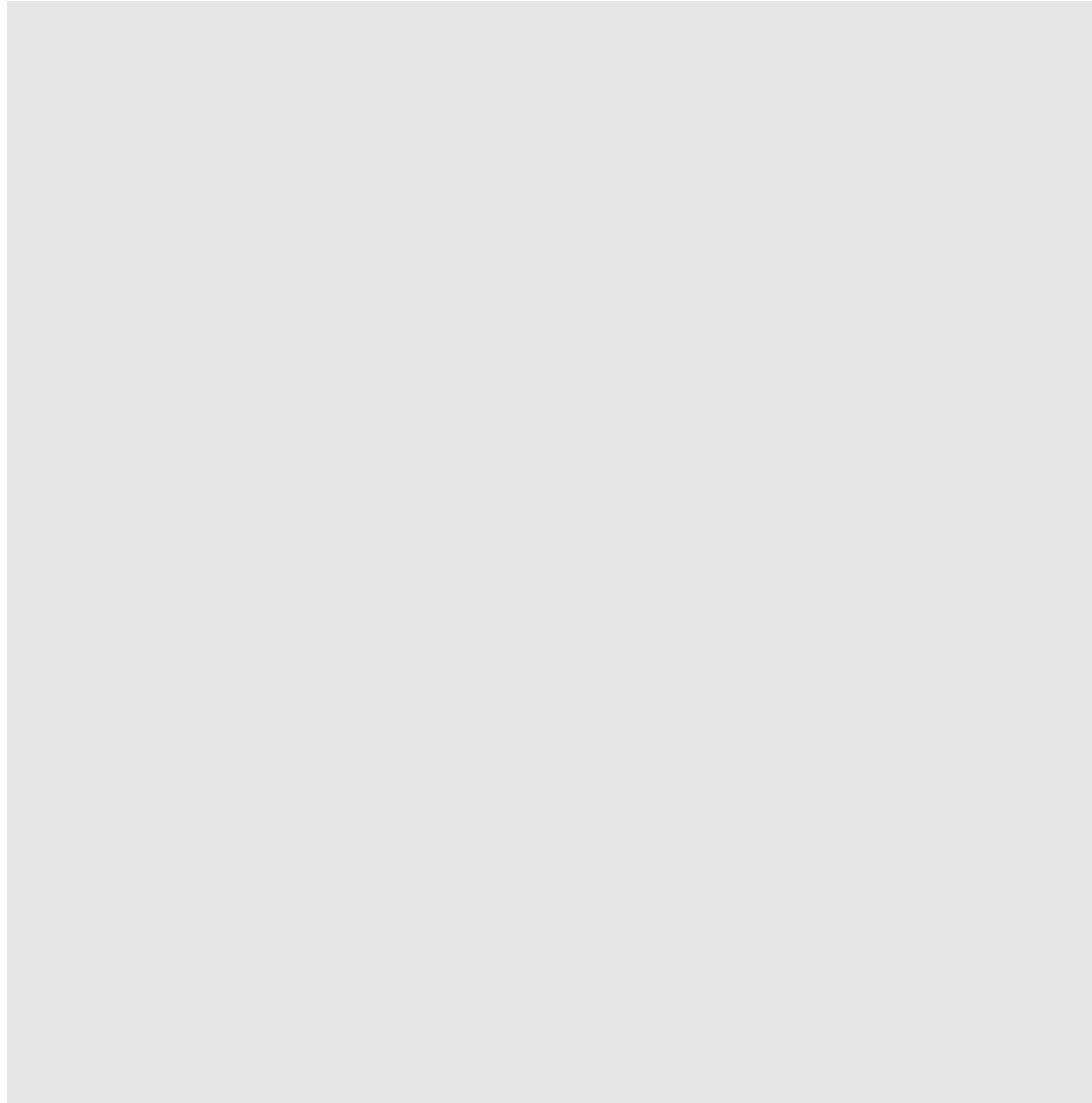
Modelo de movimiento para Scan Matching



Mapeo usando odometría cruda



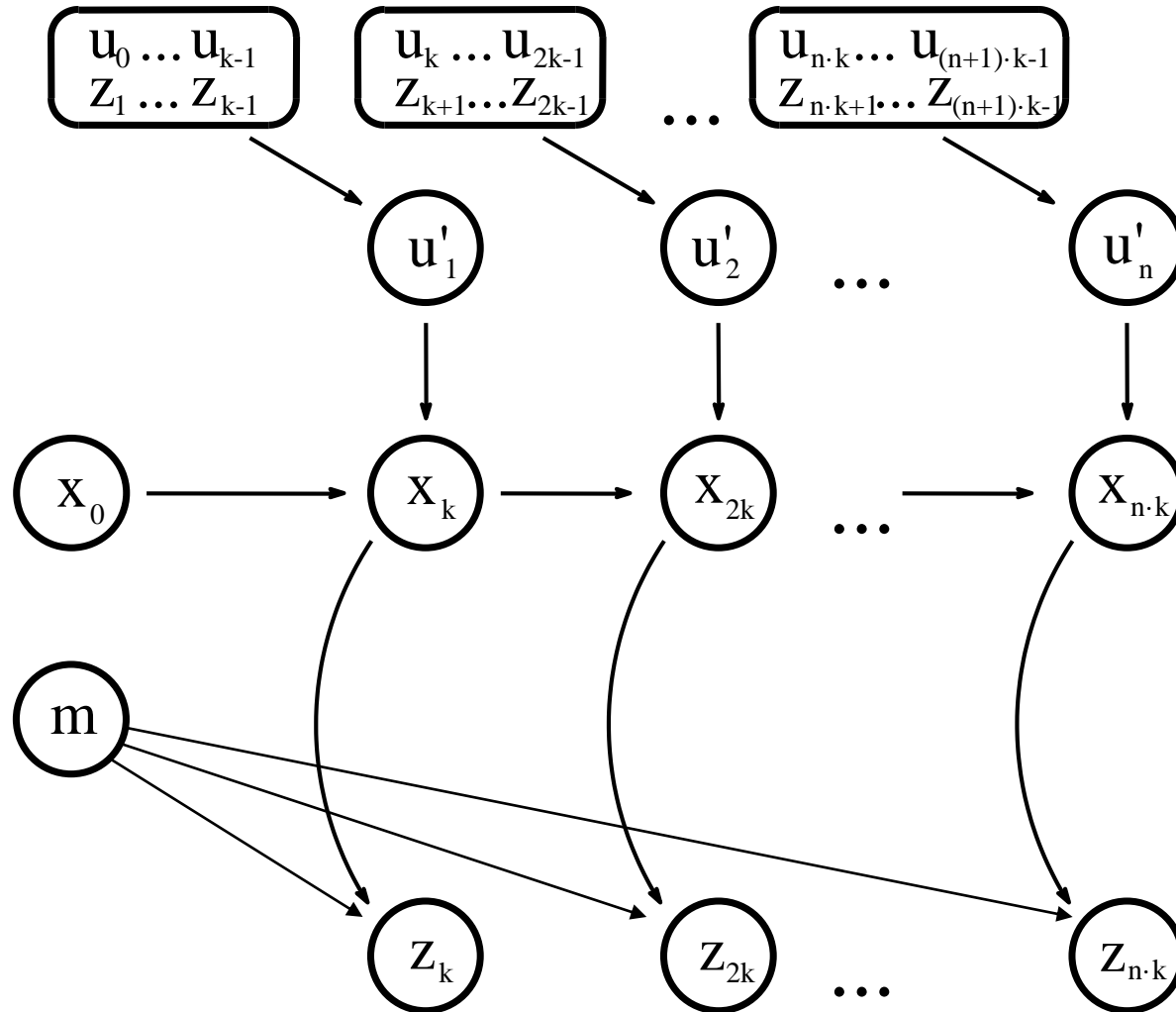
Mapeo usando Scan Matching



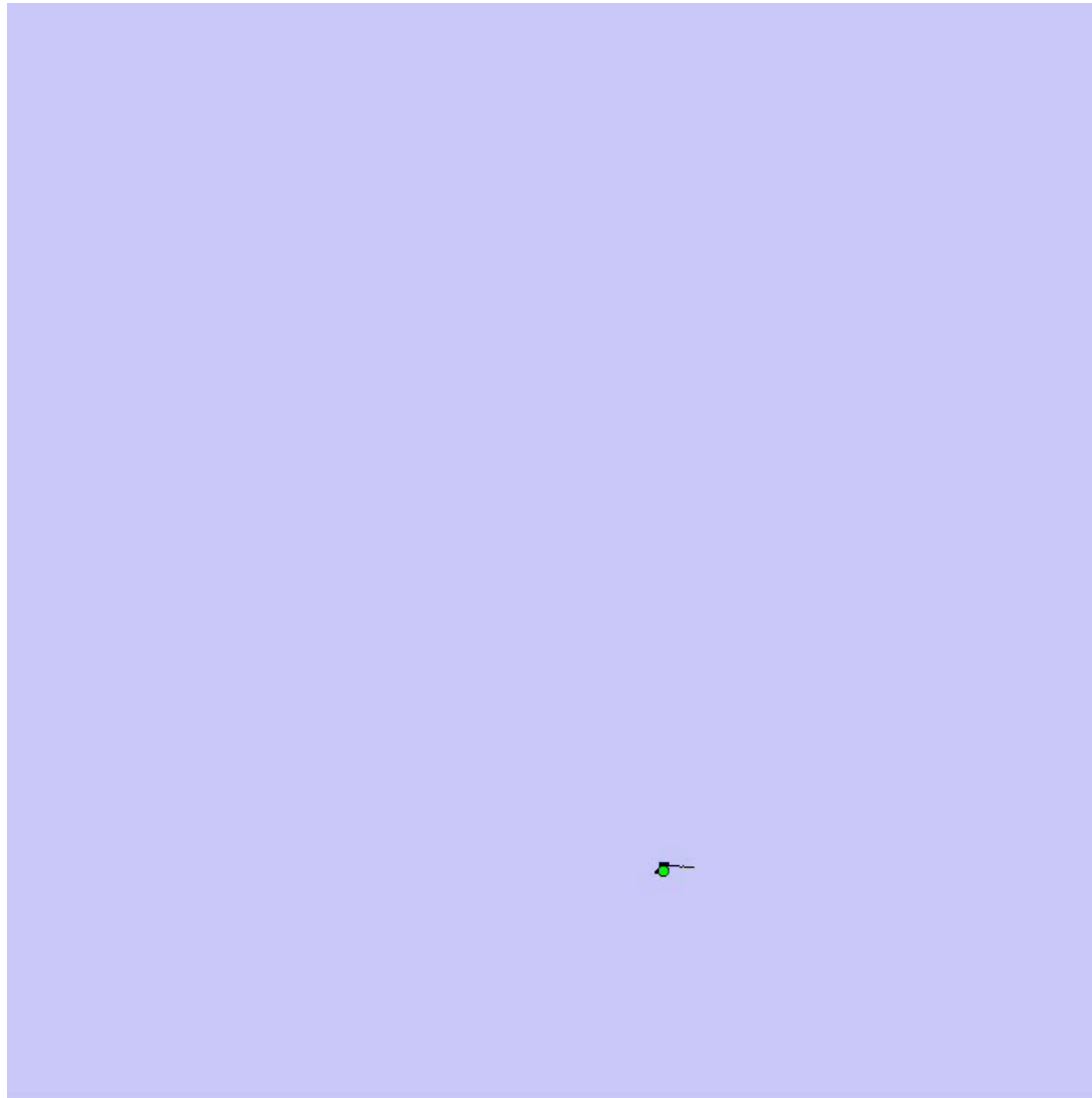
FastSLAM con odometría mejorada

- Scan-matching brinda una corrección de pose **localmente consistente**
- Pre-corregir secuencias cortas de odometría usando scan-matching y usarlas como entrada a FastSLAM
- Se necesitan **menos partículas**, ya que el error de entrada es menor

Modelo gráfico para mapeo con odometría mejorada



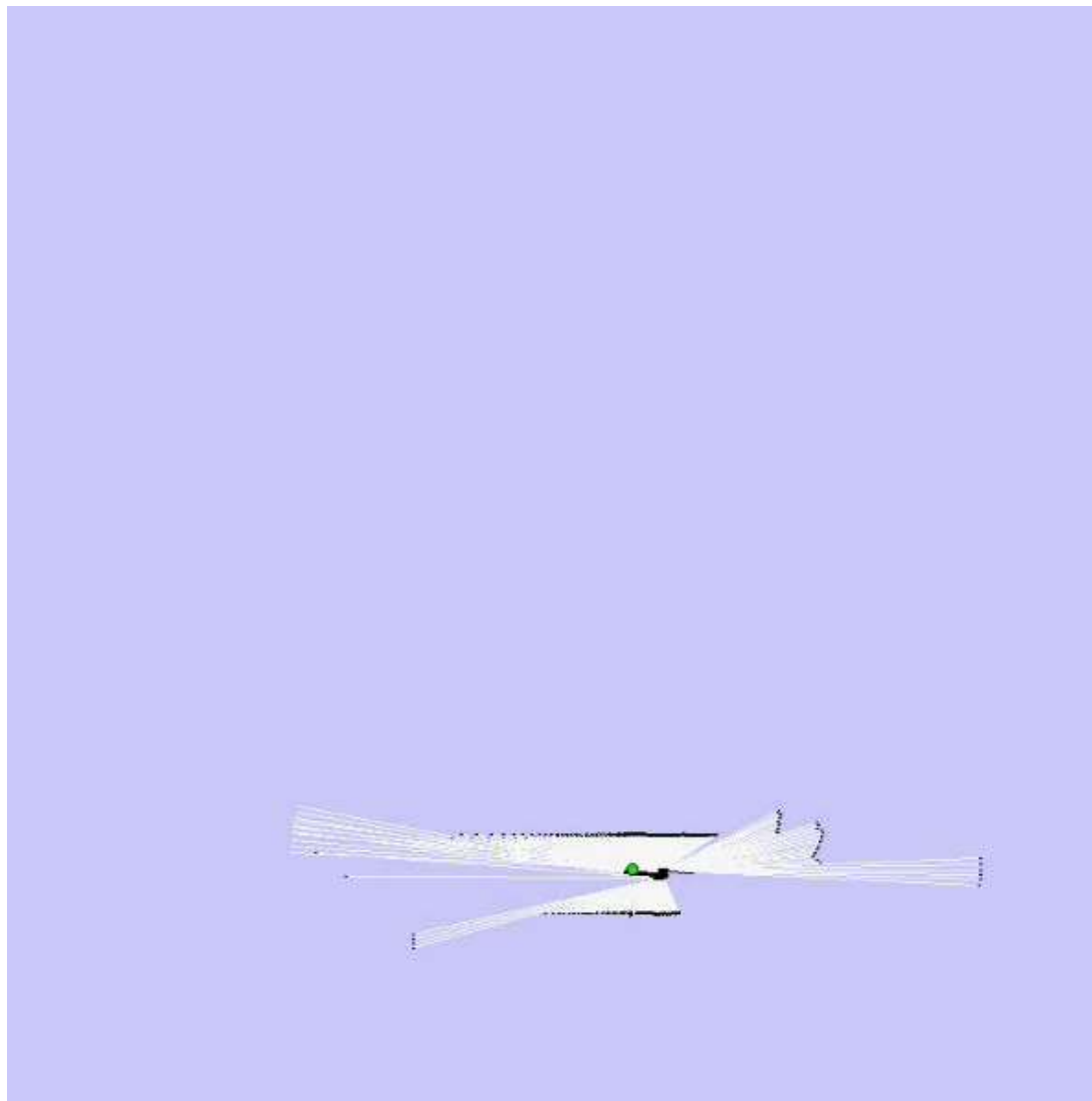
FastSLAM con Scan-Matching



FastSLAM con Scan-Matching



FastSLAM con Scan-Matching



Comparación con FastSLAM estándar

- El mismo modelo de medición
- Odometría en vez de scan matching como entrada
- Número de partículas variando entre 500 y 2000
- Resultado:



Resumen (hasta acá ...)

- Método **muy eficiente** de SLAM combinando ideas de **scan-matching** y FastSLAM
- Scan-matching se usa para transformar **secuencias de mediciones** láser en mediciones de **odometría**
- Esta variante de FastSLAM basado en grillas puede manejar **entornos más grandes** en “tiempo real”

Qué más se puede hacer?

- Reducir más el número de partículas
- Mejorar las distribuciones propuestas para lograr mejores mapas
- Usar las características del sensor cuando se produce la próxima generación de partículas

Distribución propuesta óptima

Probabilidad de la pose
dada la info recolectada

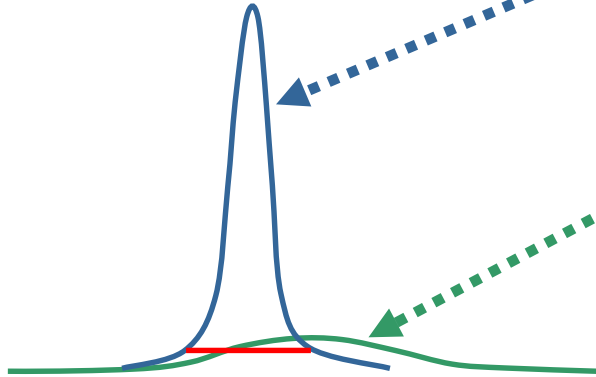
Modelo de
medición

Modelo de
movimiento

$$\underbrace{p(x_t | x_{t-1}^{(i)}, m^{(i)}, z_t, u_t)}_{\text{[Arulampalam et al., 01]}} = \frac{\overbrace{p(z_t | x_t, m^{(i)})}^{\text{Modelo de medición}} \overbrace{p(x_t | x_{t-1}^{(i)}, u_t)}^{\text{Modelo de movimiento}}}{\underbrace{\int p(z_t | x_t, m^{(i)}) p(x_t | x_{t-1}^{(i)}, u_t) dx_t}_{\text{normalización}}}$$

Distribución propuesta óptima

$$p(x_t | x_{t-1}^{(i)}, m^{(i)}, z_t, u_t) = \frac{p(z_t | x_t, m^{(i)}) p(x_t | x_{t-1}^{(i)}, u_t)}{\int p(z_t | x_t, m^{(i)}) p(x_t | x_{t-1}^{(i)}, u_t) dx_t}$$



Para láseres $p(z_t | x_t, m^{(i)})$ es muy estrecha y domina el producto.

Entonces, podemos aproximar $p(x_t | x_{t-1}^{(i)}, u_t)$ por una constante $p(x_t | x_{t-1}^{(i)}, u_t) |_{x_t: p(z_t | x_t, m^{(i)}) > \epsilon} = c$

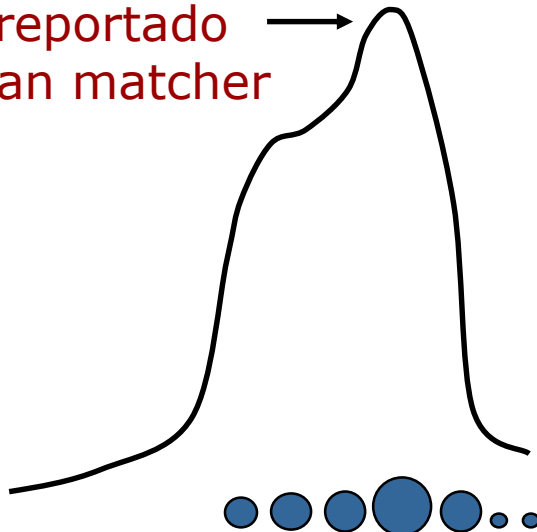
Distribución propuesta resultante

$$p(x_t|x_{t-1}^{(i)}, m^{(i)}, z_t, u_t) \simeq \frac{p(z_t|x_t, m^{(i)})}{\int_{x_t \in \{x|p(z_t|x, m^{(i)}) > \epsilon\}} p(z_t|x_t, m^{(i)}) dx_t}$$

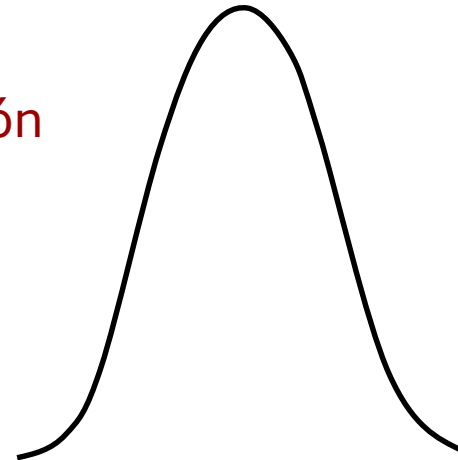
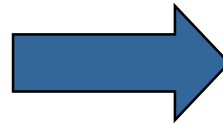
Aproximamos esta ecuación con una gaussiana:

$$p(x_t|x_{t-1}^{(i)}, m^{(i)}, z_t, u_t) \simeq \mathcal{N}(\mu^{(i)}, \Sigma^{(i)})$$

Máximo reportado
por el scan matcher



Aproximación
Gaussiana



Crear la
próxima
generación de
muestras

Puntos muestreados
alrededor del máximo

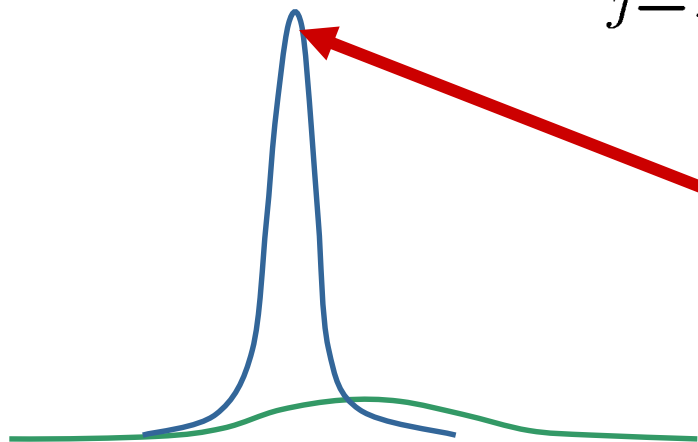
Estimando los parámetros de la gaussiana para cada partícula

$$\mu^{(i)} = \frac{1}{\eta} \sum_{j=1}^K x_j p(z_t | x_j, m^{(i)})$$
$$\Sigma^{(i)} = \frac{1}{\eta} \sum_{j=1}^K (x_j - \mu^{(i)})(x_j - \mu^{(i)})^T p(z_t | x_j, m^{(i)})$$

- x_j son un conjunto de puntos muestreados alrededor del punto x^* al que convergió el scan matching.
- η es la constante de normalización

Calculando los pesos de importancia

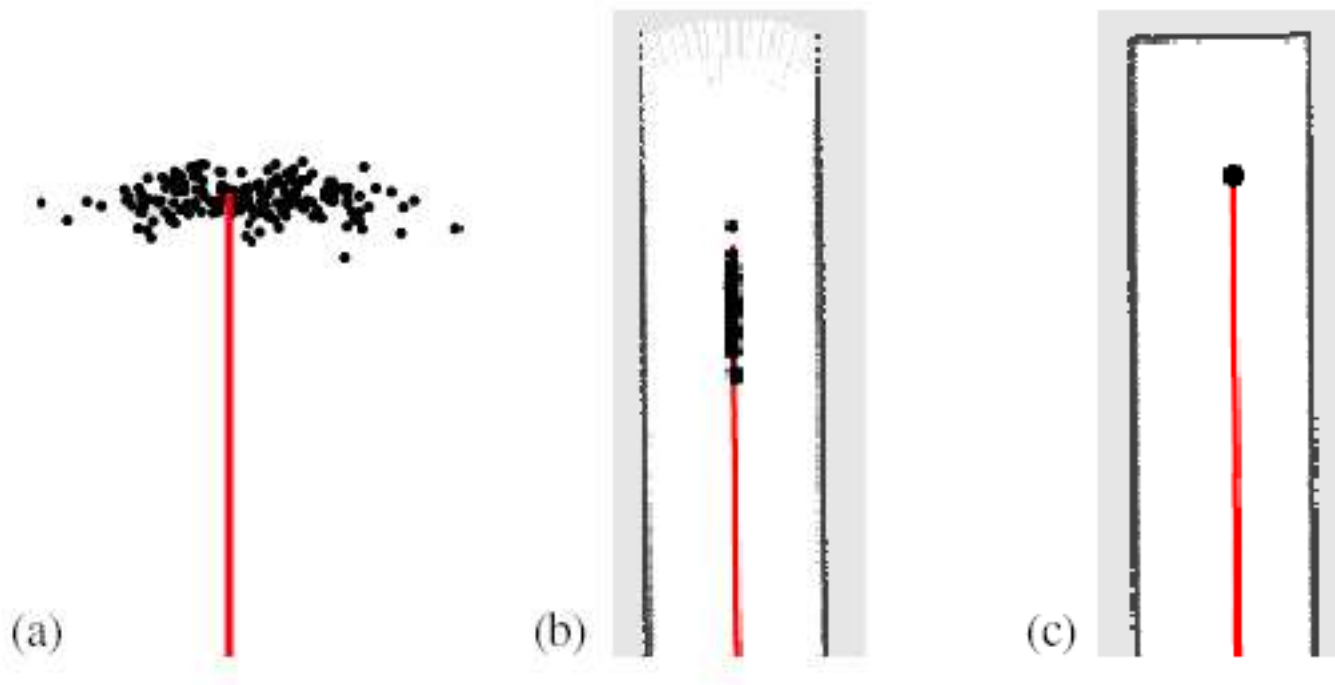
$$\begin{aligned}w_t^{(i)} &= w_{t-1}^{(i)} p(z_t | x_{t-1}^{(i)}, m^{(i)}, u_t) \\&\simeq w_{t-1}^{(i)} \int p(z_t | x_t, m^{(i)}) p(x_t | x_{t-1}^{(i)}, u_t) dx_t \\&\simeq w_{t-1}^{(i)} c \int_{x_t \in \{x | p(z_t | x, m^{(i)}) > \epsilon\}} p(z_t | x_t, m^{(i)}) dx_t \\&\simeq w_{t-1}^{(i)} c \sum_{j=1}^K p(z_t | x_j, m^{(i)})\end{aligned}$$



Puntos muestreados alrededor
del máximo del likelihood
de observación

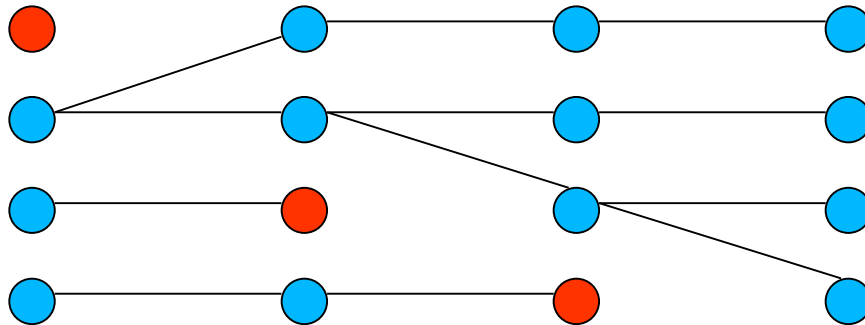
Dist. Propuesta mejorada

- La distribución propuesta se adapta a la estructura del entorno



Remuestreo

- Muestrear de la distribución propuesta mejorada reduce los efectos del remuestreo
- De todas formas, remuestrear en cada paso limita la “memoria” del filtro
- Supongamos que perdemos el 25% de las partículas en cada paso, en el peor caso tenemos una memoria de solo 4 pasos.



Objetivo: reducir el número de acciones de remuestreo

Remuestreo selectivo

- Remuestrear es peligroso, ya que se pueden perder muestras importantes (problema de agotamiento de partículas)
- En caso de distribuciones propuestas subóptimas, el remuestreo es necesario para lograr convergencia.
- Pregunta importante ¿Cuándo remuestrear?

Número de partículas efectivas

$$n_{eff} = \frac{1}{\sum_i \left(w_t^{(i)}\right)^2}$$

- Asumiendo pesos de partículas normalizadas que suman 1: $\sum_{i=1}^n w_t^{(i)} = 1 \Rightarrow n_{eff} \in [1, n]$
- Medida empírica de que tan bien se aproxima la distribución objetivo con muestras tomadas de la distribución propuesta
- Describe la “varianza en los pesos de las partículas”
- Es máxima para pesos iguales. En este caso, la distribución es parecida a la propuesta

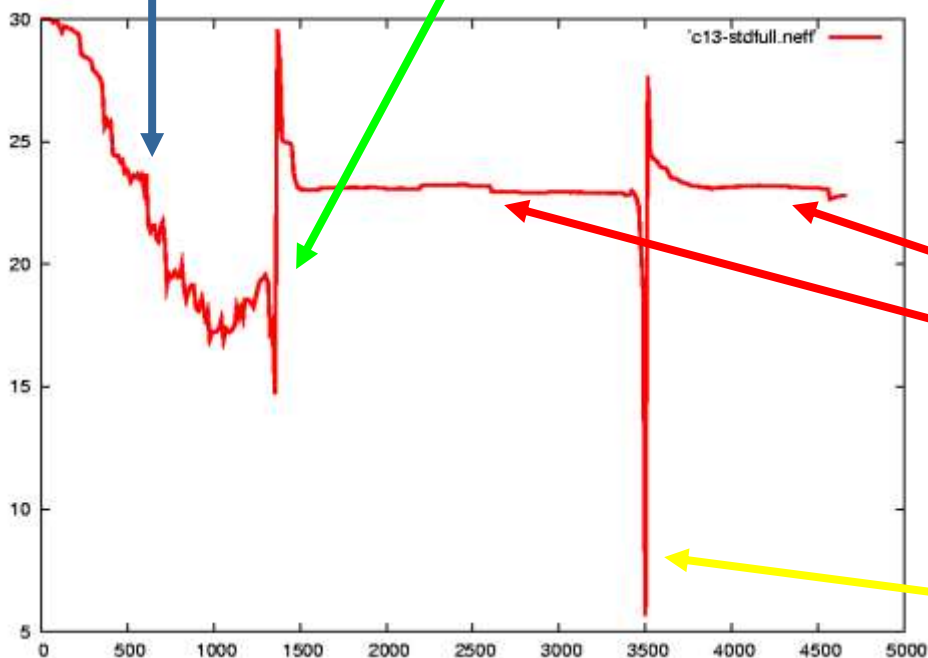
Remuestreo con n_{eff}

- Si nuestra aproximación se parece a la propuesta, no se necesita remuestreo
- Sólo se remuestrea cuando n_{eff} cae por debajo de un umbral, generalmente $\frac{n}{2}$
- Desarrollado por [Doucet, '98; Arulampalam, '01]

Evolución de n_{eff}

Visitando
áreas nuevas

Cerrando
primer lazo



Visitando áreas
conocidas

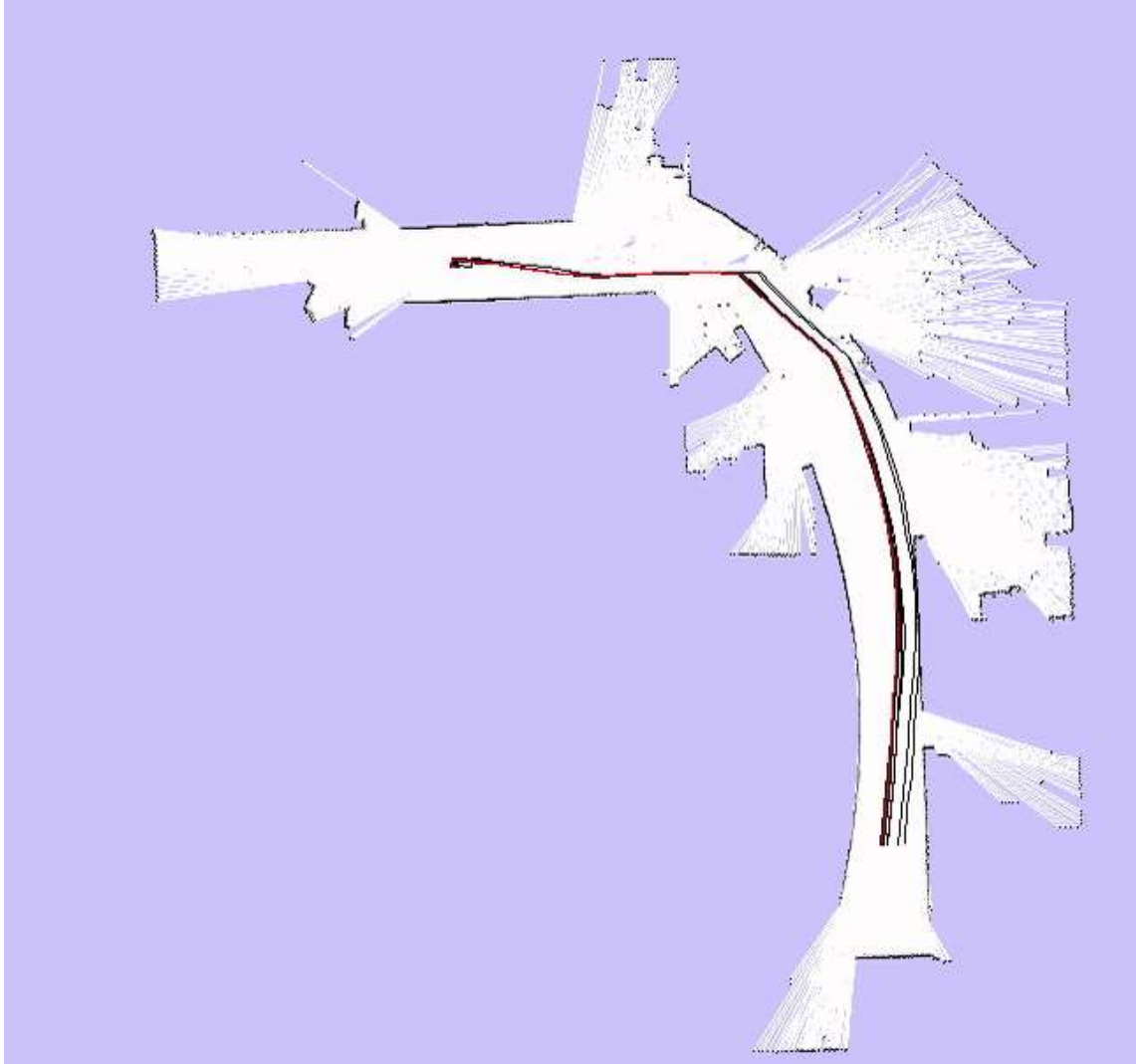
Otro cierre de lazo

Intel Lab



- **15 partículas**
- 4 veces más rápido que tiempo real
P4, 2.8GHz
- 5cm de resolución
haciendo scan matching
- 1cm de resolución en el
mapa final

Intel Lab



- **15 partículas**
- Comparado con FastSLAM con Scan-Matching, las partículas están más cerca de la distribución verdadera

Campus de la Univ. de Freiburg



- **30 partículas**
- 250x250m²
- 1750 m (odometría)
- 20cm de resolución haciendo scan matching
- 30cm de resolución en el mapa final

Campus de la Univ. de Freiburg

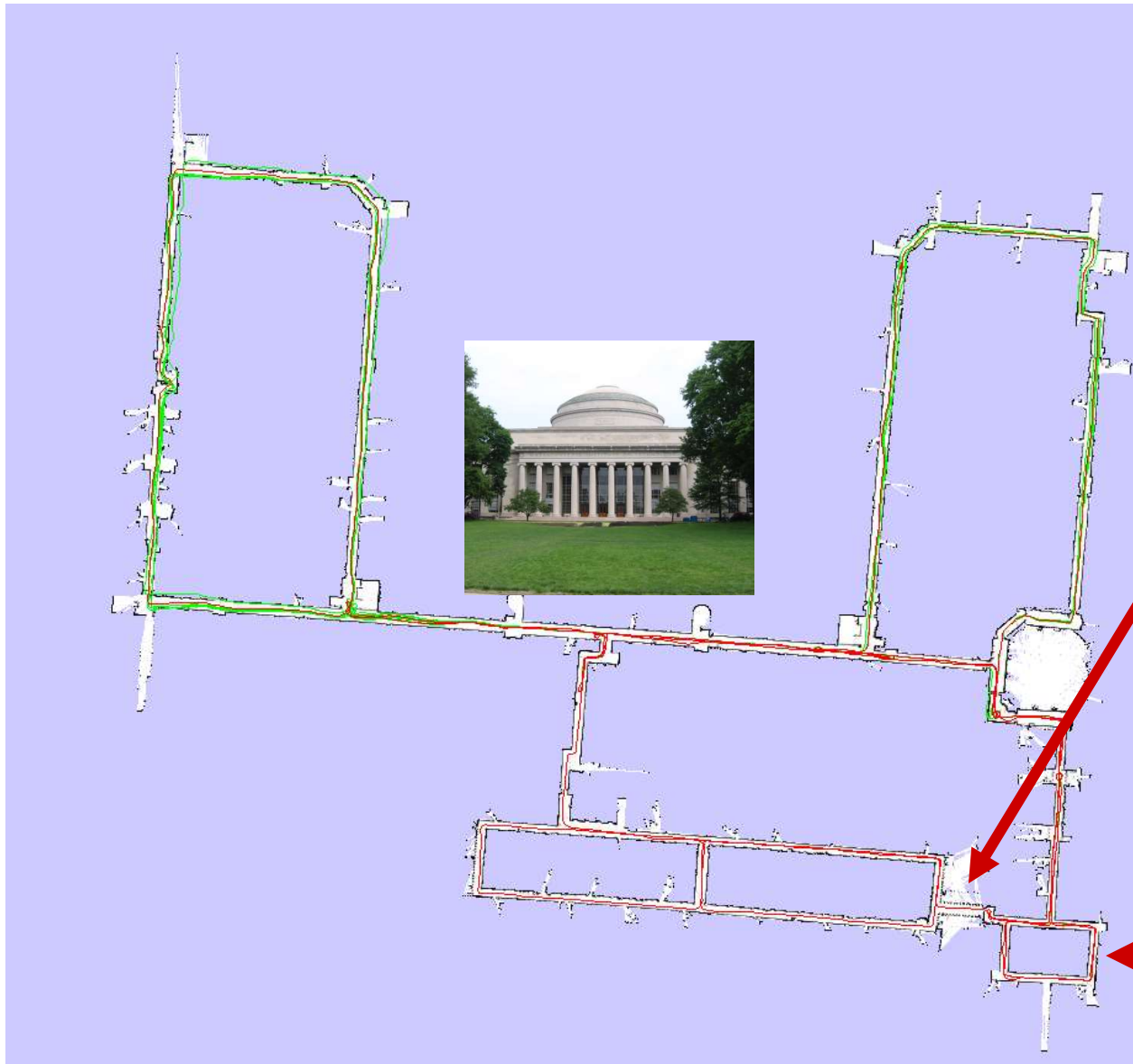


MIT Killian Court

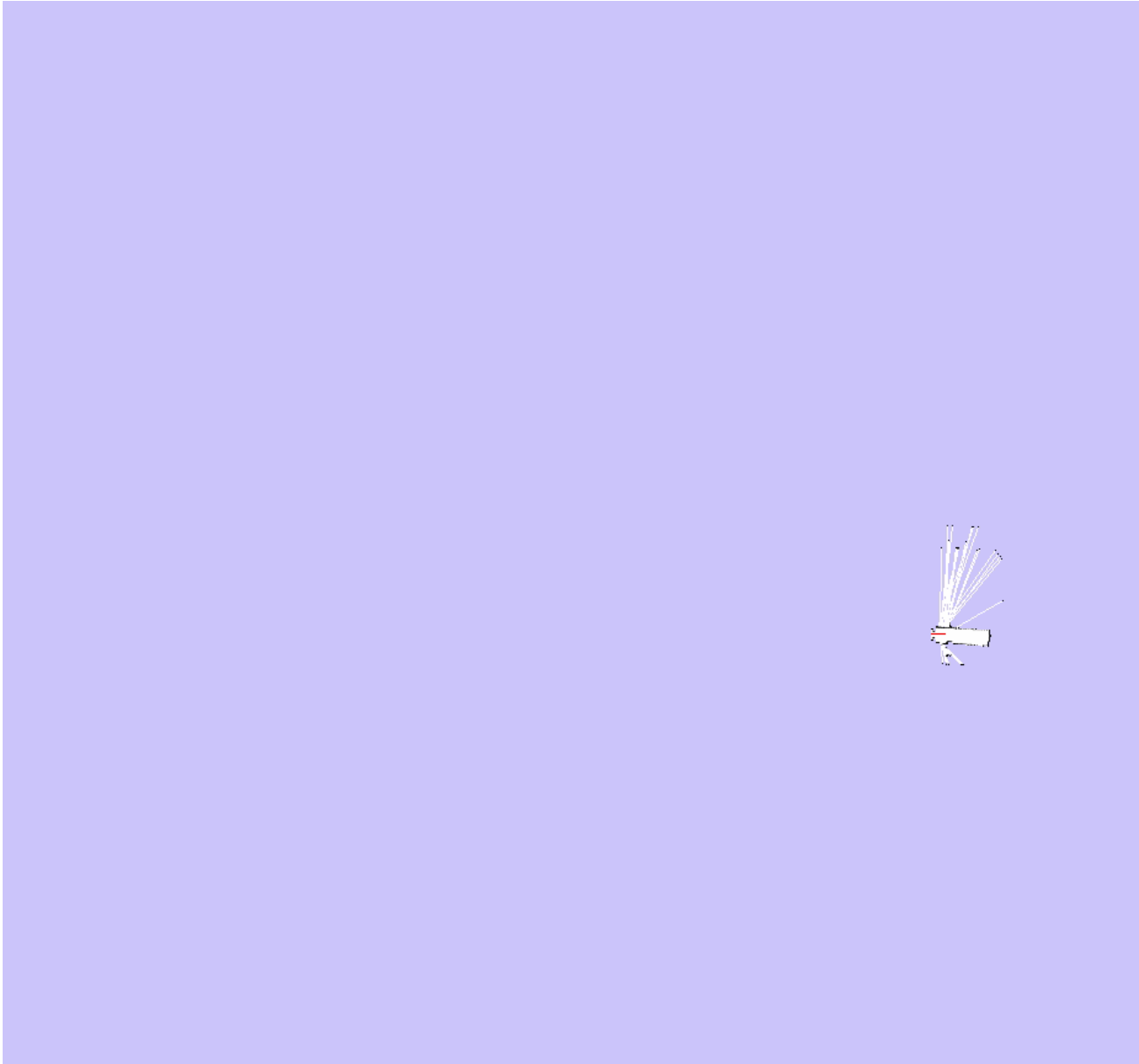


- El **“infinite-corridor-dataset”** de MIT

MIT Killian Court



MIT Killian Court - Video



Conclusión

- El método de FastSLAM también se puede aplicar a **mapas de grilla**
- Con mediciones precisas del sensor se obtienen **buenas dist. propuestas** y filtros eficientes
- Es similar a hacer scan-matching aplicado a cada partícula
- La cantidad de **partículas** y pasos de **remuestreo** pueden ser **reducidas** significativamente
- Versiones mejoradas de FastSLAM basado en grillas pueden manejar **entornos más grandes** que la implementación original en “tiempo real” ya que necesitan un orden de magnitud menos de muestras