

IIC2685 Robótica Móvil I – 2022

Capítulo 4.3

Localización

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

Agenda

- Entender el problema general de localización
- Fundamentos e implementación de algoritmo de *Filtro de Partículas*

Problema general de localización

Recordemos las **Grandes Preguntas** de un robot móvil

¿Dónde estoy?

¿Para dónde voy?

¿Cómo llego?



Problema general de localización

Recordemos las **Grandes Preguntas** de un robot móvil

¿Dónde estoy?

¿Para dónde voy?

¿Cómo llego?



Problema general de localización

Para responder estas preguntas contamos con:

- Mapa del ambiente
- Mediciones: modelo de sensor
- Acciones: modelo de movimiento
- Filtro Bayesiano!

$$Bel(x_t) = P(x_t | u_1, z_1, \dots, u_t, z_t)$$

$$= \eta P(z_t | x_t) \int_{x_{t-1}} P(x_t | u_t, x_{t-1}) Bel(x_{t-1}) dx_{t-1}$$

Factor de
normalización

Sensor
model

Motion
model

“Belief”
anterior

Problema general de localización

Dificultades de implementación de filtros bayesianos

- Los robots móviles deben lidiar muchas veces con incertidumbre no Gausiana
- Pdf posee forma de gran complejidad
- Modelos generales deben ajustar sus parámetros al caso particular
- ¿ Será posible encontrar una solución más simple ?

Problema general de localización

Dificultades de implementación de filtros bayesianos

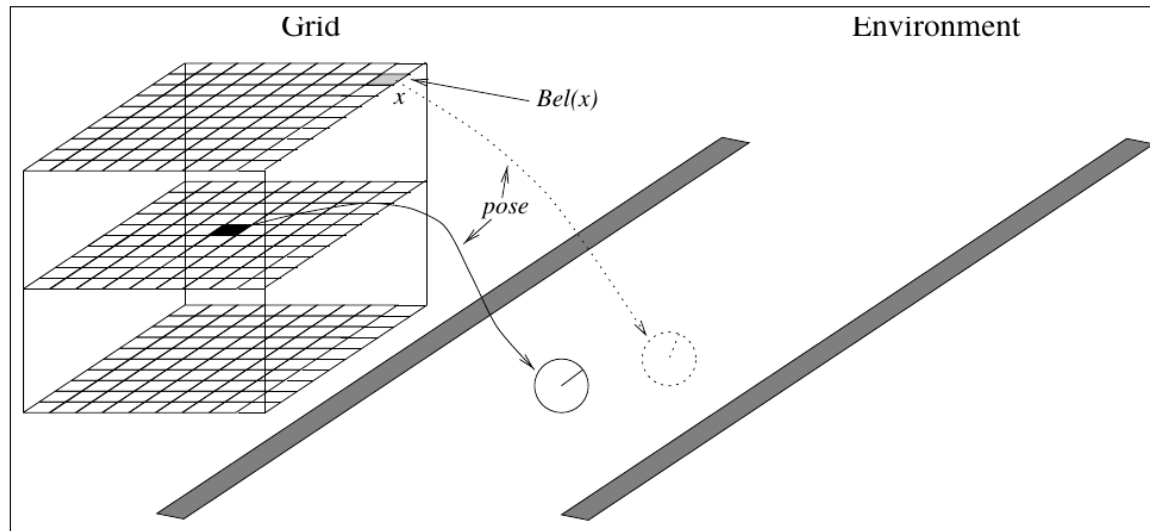
- Los robots móviles deben lidiar muchas veces con incertidumbre no Gausiana
- Pdf posee forma de gran complejidad
- Modelos generales deben ajustar sus parámetros al caso particular
- ¿ Será posible encontrar una solución más simple ?

1.- Localización basada en grilla

2.- Filtro de Partículas

Localización basada en grilla

- Aproxima la *posterior* (*belief*) utilizando un **filtro de histograma** sobre una descomposición en **grilla** del espacio de poses.
- Tamaño de grilla típico: 15 [cm] y 5°
- Espacio de poses: $\text{domain}(X_t) = x_{1,t} \cup x_{2,t} \cup \dots \cup x_{K,t}$
- Por cada celda, asigna **valores discretos** de probabilidad: $\text{Bel}(x_t) = \{p_{k,t}\}$



Localización basada en grilla

- Algoritmo de localización basada en grilla

```
1:   Algorithm Grid_localization( $\{p_{k,t-1}\}, u_t, z_t, m$ ):  
2:     for all  $k$  do  
3:        $\bar{p}_{k,t} = \sum_i p_{i,t-1} \text{ motion\_model}(\text{mean}(\mathbf{x}_k), u_t, \text{mean}(\mathbf{x}_i))$   
4:        $p_{k,t} = \eta \bar{p}_{k,t} \text{ measurement\_model}(z_t, \text{mean}(\mathbf{x}_k), m)$   
5:     endfor  
6:     return  $\{p_{k,t}\}$ 
```

Localización basada en grilla

- Ventajas
 - Utiliza información de los sensores sin necesidad de procesamiento
 - No paramétrico
 - Resuelve problema de **localización global**
 - Puede resolver el problema de *Kidnapped Robot*
 - Robot cree saber dónde está, pero de forma equivocada

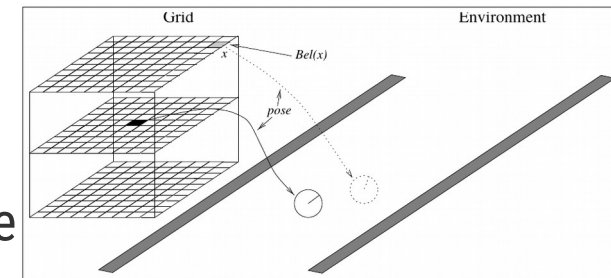
Localización basada en grilla

- Ventajas

- Utiliza información de los sensores sin necesidad de procesamiento
- No paramétrico
- Resuelve problema de **localización global**
- Puede resolver el problema de *Kidnapped Robot*
 - Robot cree saber dónde está, pero de forma equivocada

- Inconvenientes:

- Se debe escoger resolución de la grilla
 - Fina: representación **métrica** de tamaño fijo
 - Gruesa: representación **topológica** de tamaño variable
- Gran costo computacional

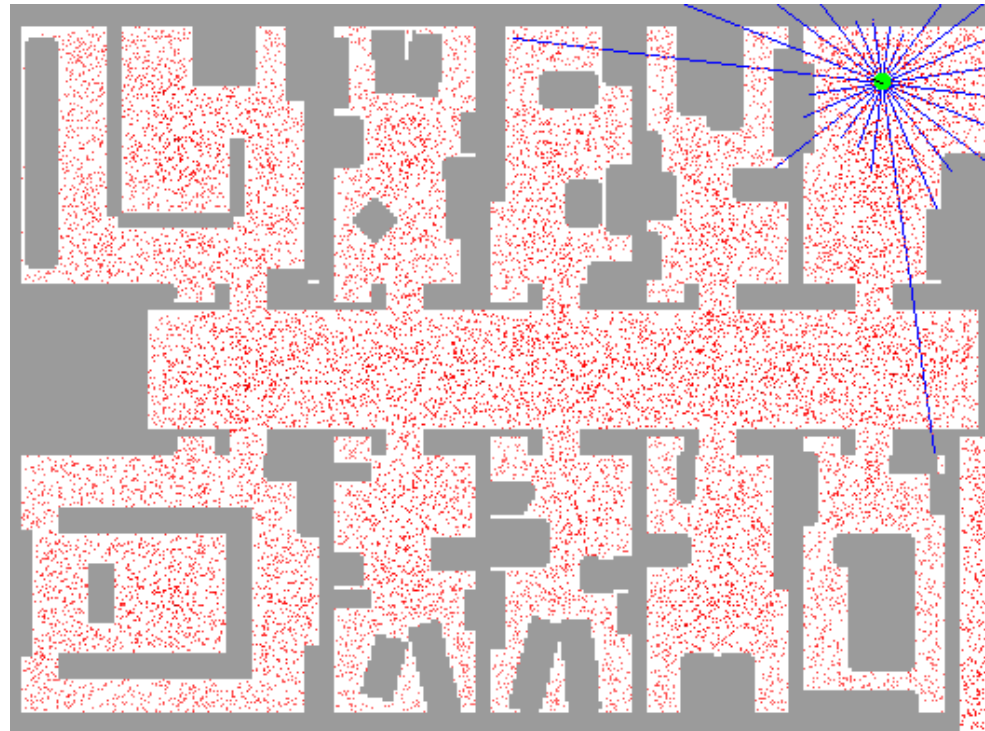


Filtro de Partículas

- Aproxima la $Bel(X_t)$ (*posterior*) a través de un conjunto de M **partículas**

$$X_t = \{x_t^{[1]}, x_t^{[2]}, \dots, x_t^{[M]}\}$$

- Donde cada partícula representará una **hipótesis** de la pose del robot

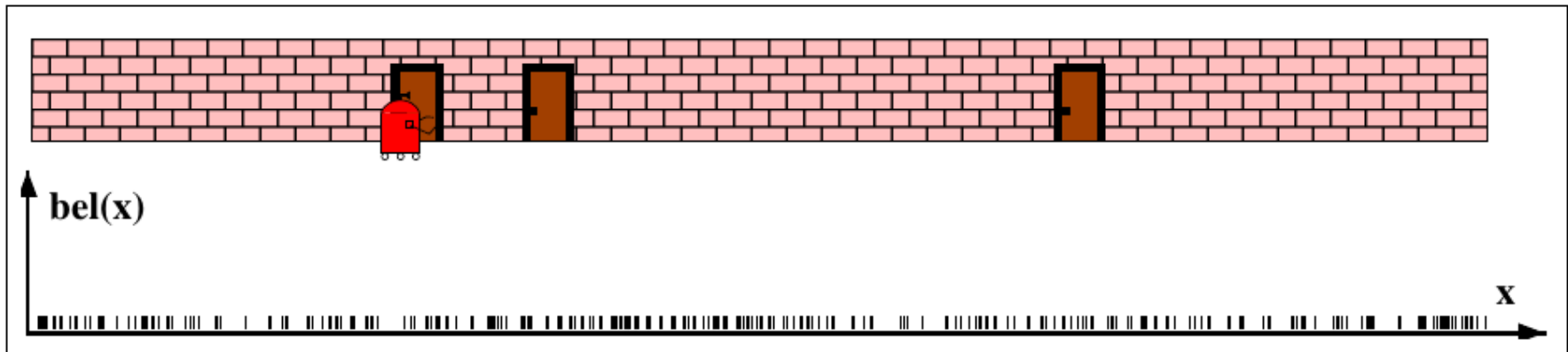


Filtro de Partículas

- Ventajas
 - Capaz de procesar información directamente de los sensores
 - No paramétrico
 - Resuelve problema de localización global
 - Puede resolver problema de *kidnapped robot*
 - Costo computacional según número de partículas: parametrizable
- Desventajas
 - Incapacidad de localizar en zonas sin partículas

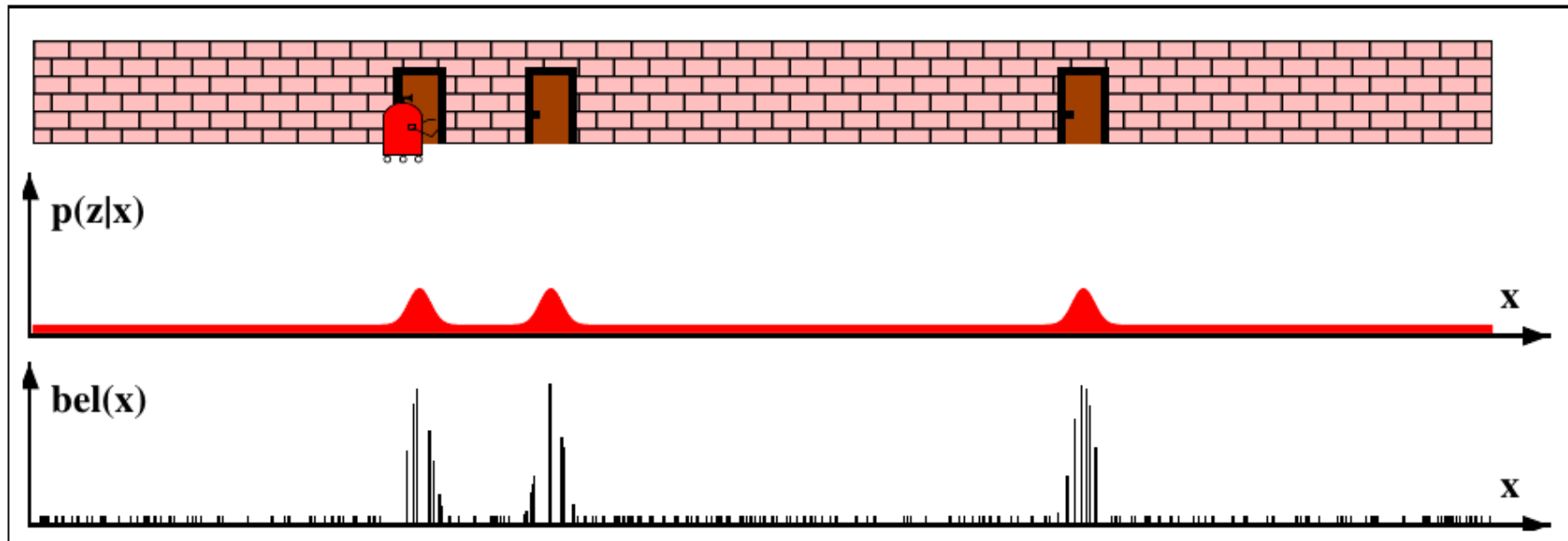
Filtro de Partículas: algoritmo

- Se lanzan M partículas sobre el espacio de poses de manera aleatoria siguiendo una distribución uniforme
- Se asigna un *factor de importancia* igual a: M^{-1}



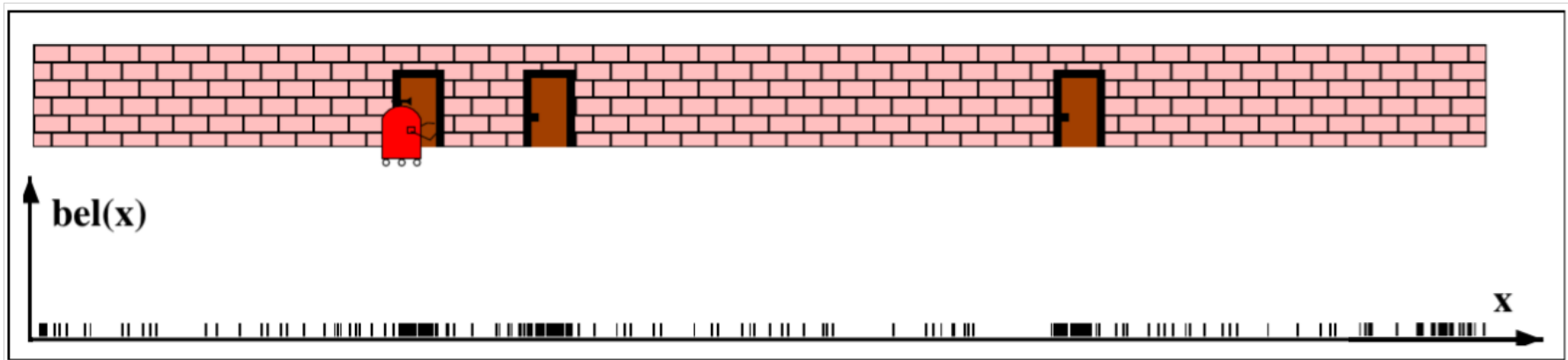
Filtro de Partículas: algoritmo

- El robot mide el ambiente y asigna un nuevo *factor de importancia* a cada partícula según el valor del likelihood $p(z|x)$
- El set de partículas es el mismo del paso anterior



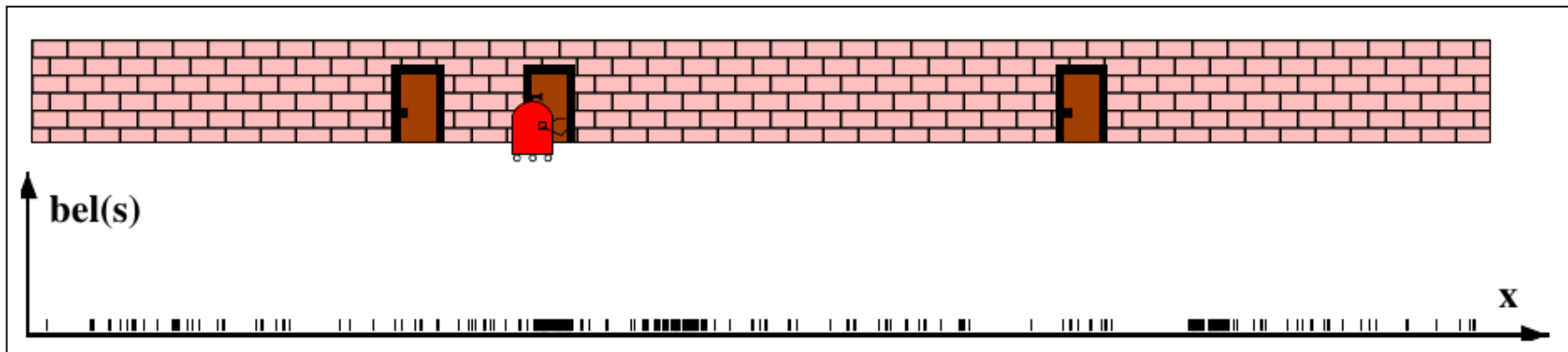
Filtro de Partículas: algoritmo

- Resampling con reemplazo de M partículas, según *factor de importancia*:
 - Variación en densidad
 - Darwin: *Sobrevivencia del más fuerte (probable)*



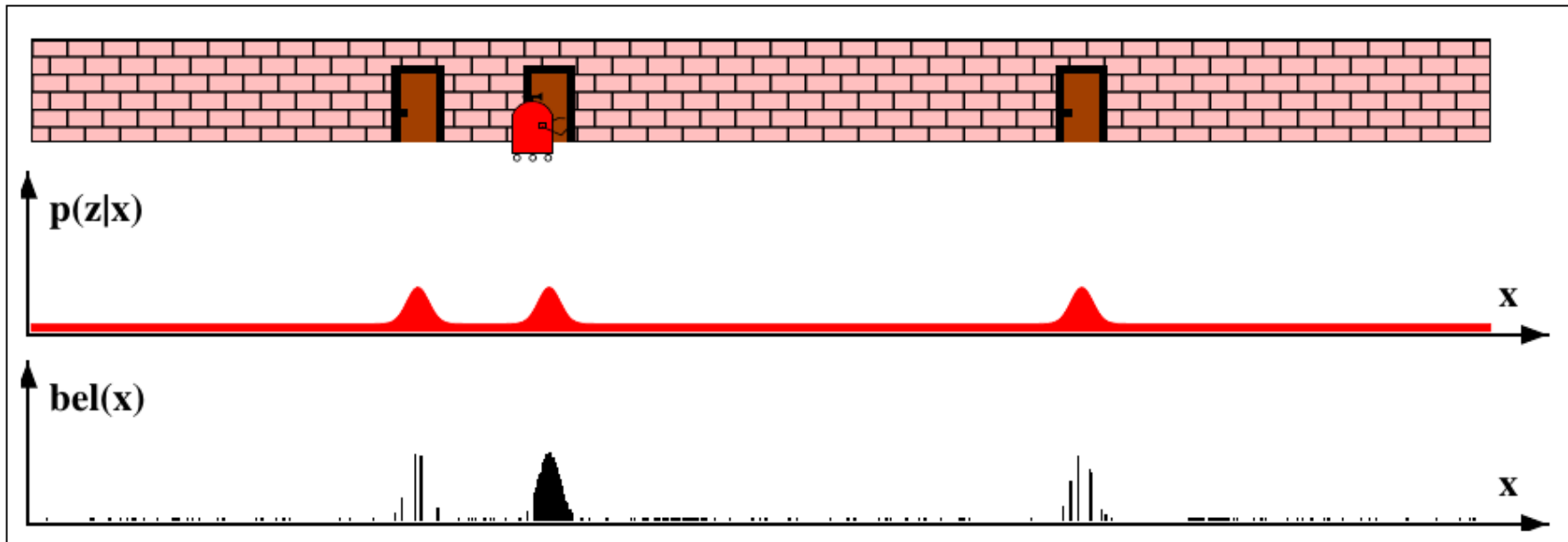
Filtro de Partículas: algoritmo

- Robot se desplaza hasta la siguiente puerta, y **partículas se mueven con él**



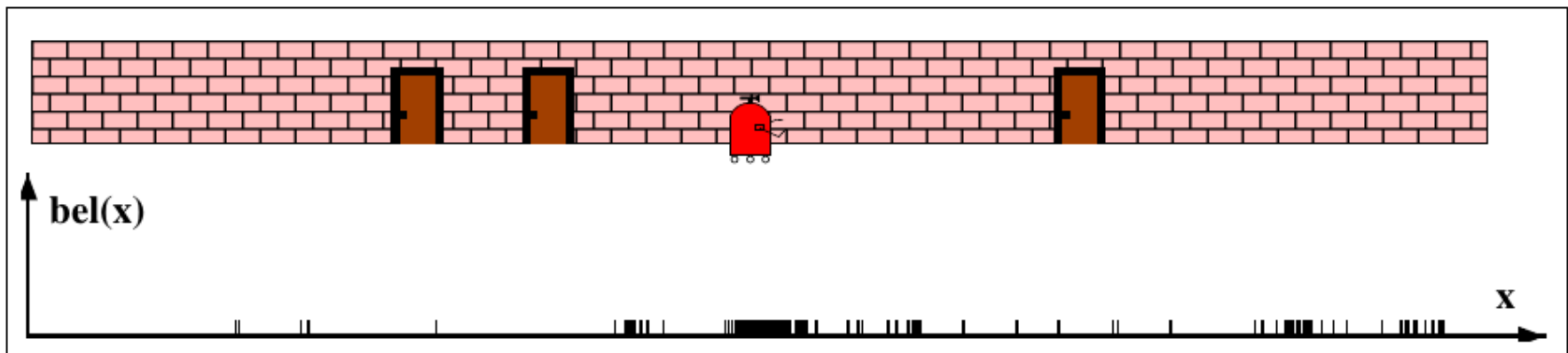
Filtro de Partículas

- Nueva medición



Filtro de Partículas

- Nuevo resampling
- Movimiento del robot y partículas



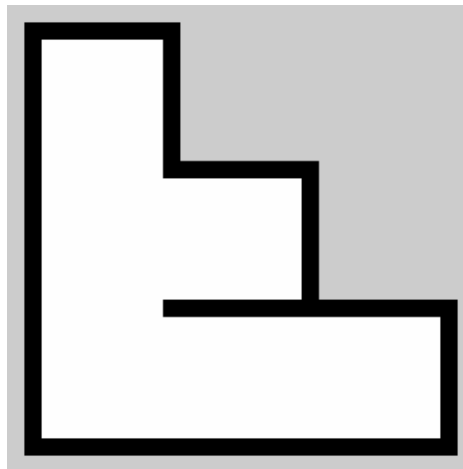
Filtro de Partículas

- Algoritmo *Monte Carlo Localization* (MCL): Filtro de Partículas

```
1:   Algorithm MCL( $\mathcal{X}_{t-1}, u_t, z_t, m$ ):  
2:      $\bar{\mathcal{X}}_t = \mathcal{X}_t = \emptyset$   
3:     for  $m = 1$  to  $M$  do  
4:        $x_t^{[m]} = \text{sample\_motion\_model}(u_t, x_{t-1}^{[m]})$   
5:        $w_t^{[m]} = \text{measurement\_model}(z_t, x_t^{[m]}, m)$   
6:        $\bar{\mathcal{X}}_t = \bar{\mathcal{X}}_t + \langle x_t^{[m]}, w_t^{[m]} \rangle$   
7:     endfor  
8:     for  $m = 1$  to  $M$  do  
9:       draw  $i$  with probability  $\propto w_t^{[i]}$   
10:      add  $x_t^{[i]}$  to  $\mathcal{X}_t$   
11:    endfor  
12:    return  $\mathcal{X}_t$ 
```

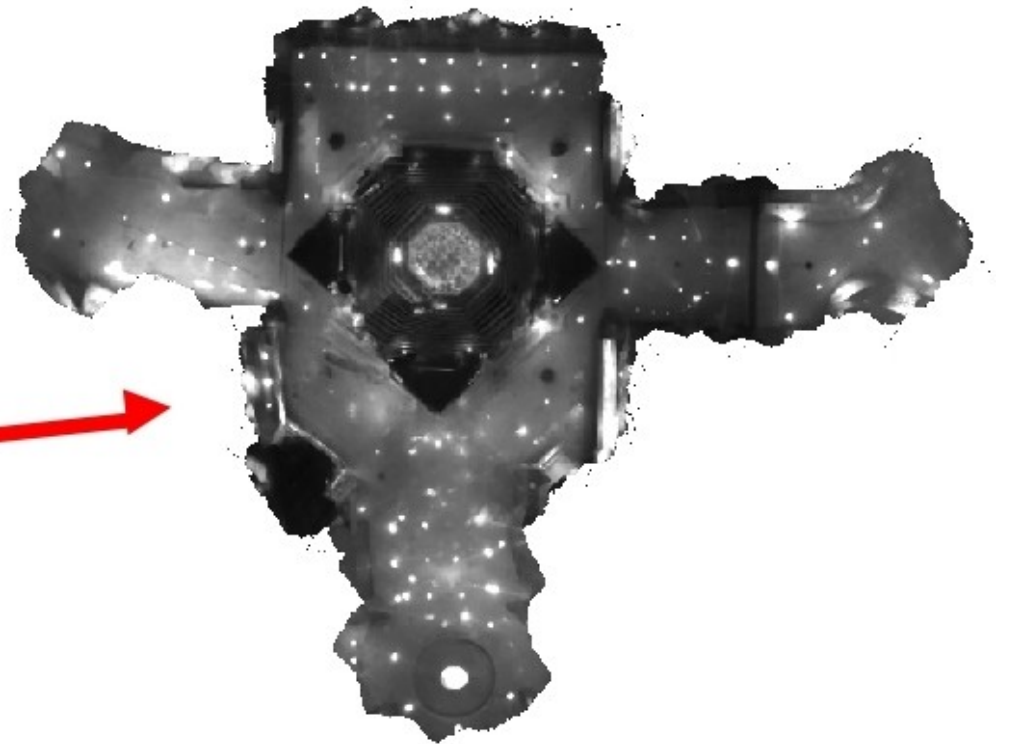
Consideraciones prácticas de implementación

- Parametrizar número de partículas (m)
 - Para ajuste de compromiso entre calidad y eficiencia
- No considerar/descartar partículas en posiciones “imposibles”
 - Tener en cuenta que movimiento puede sacar partículas del mapa
- Inyección esporádica de partículas aleatorias para evitar *Kidnapped Robot*



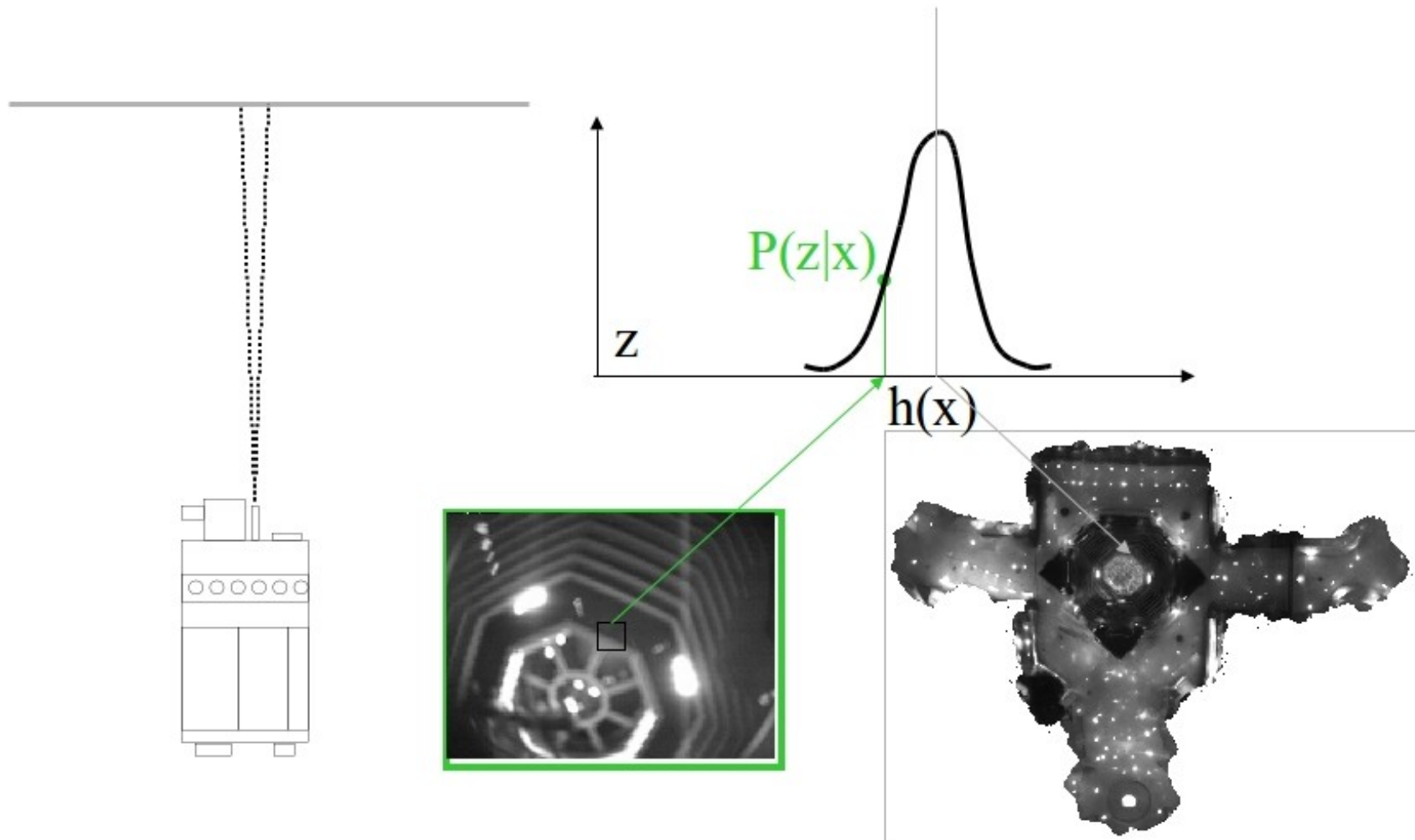
Filtro de Partículas

- **Caso de uso:** Localización en museo Smithsonian
- Usando mapas del cielo para localización



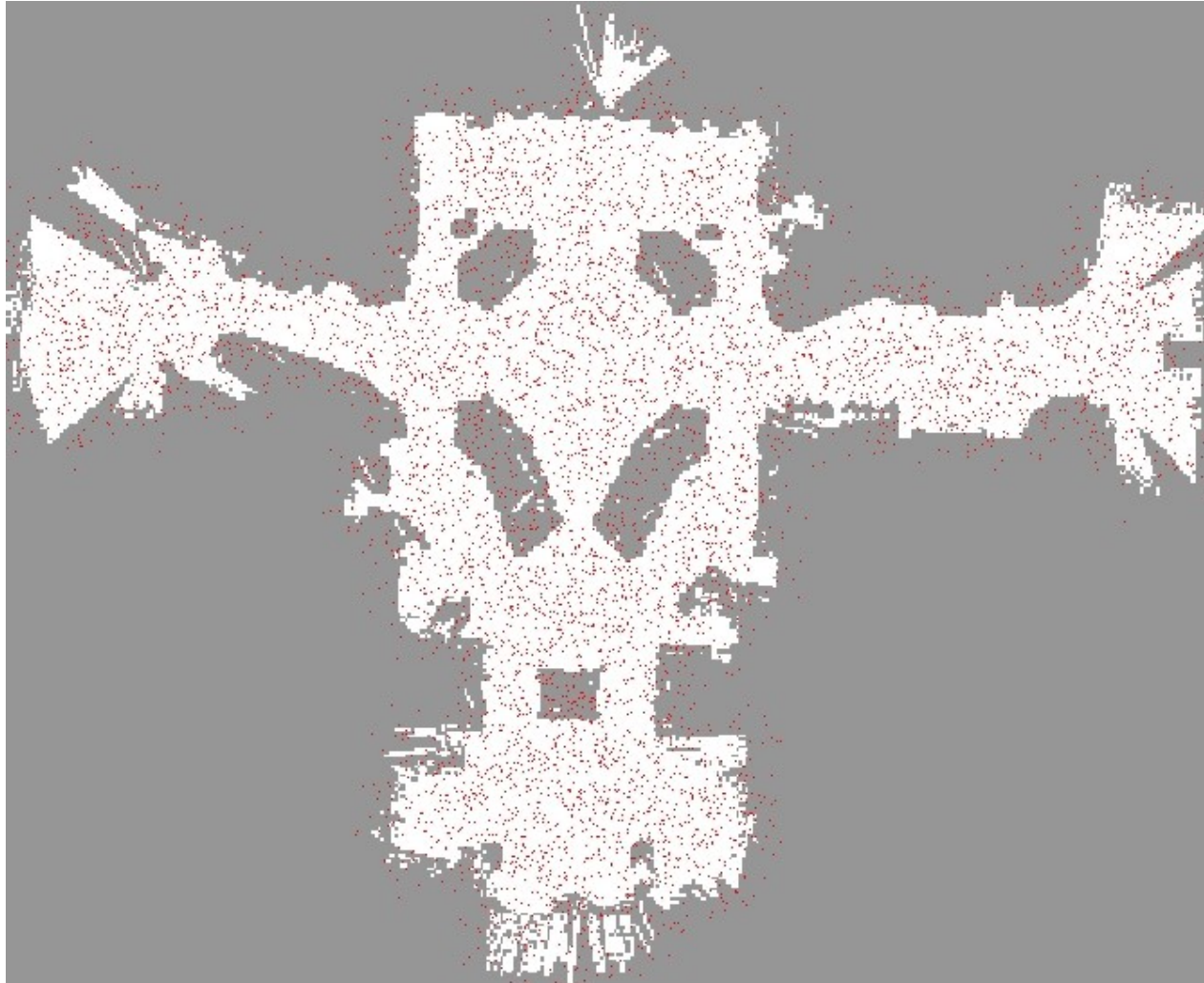
Filtro de Partículas

- Localización basada en visión



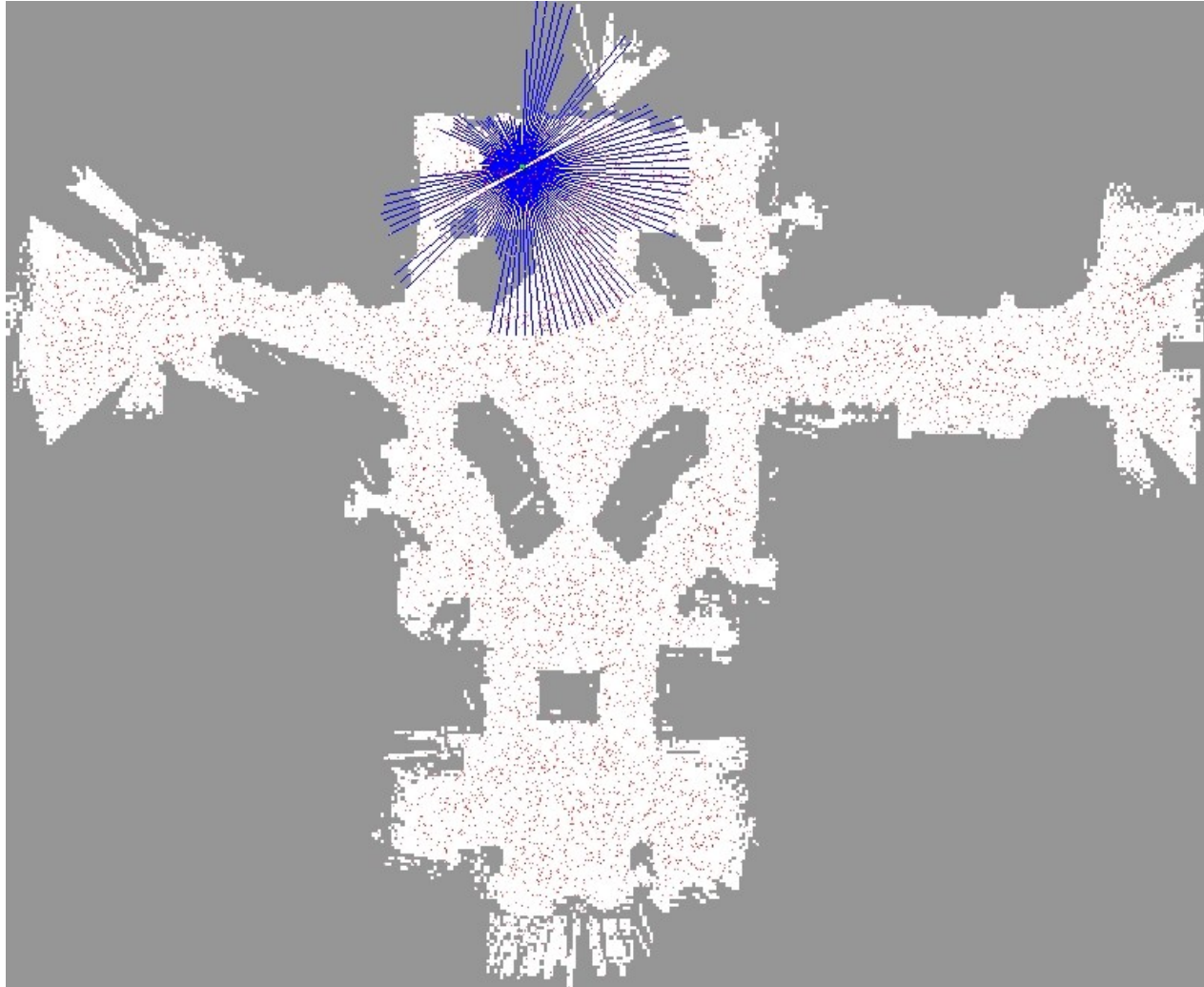
Filtro de Partículas

- Lanzamiento de partículas



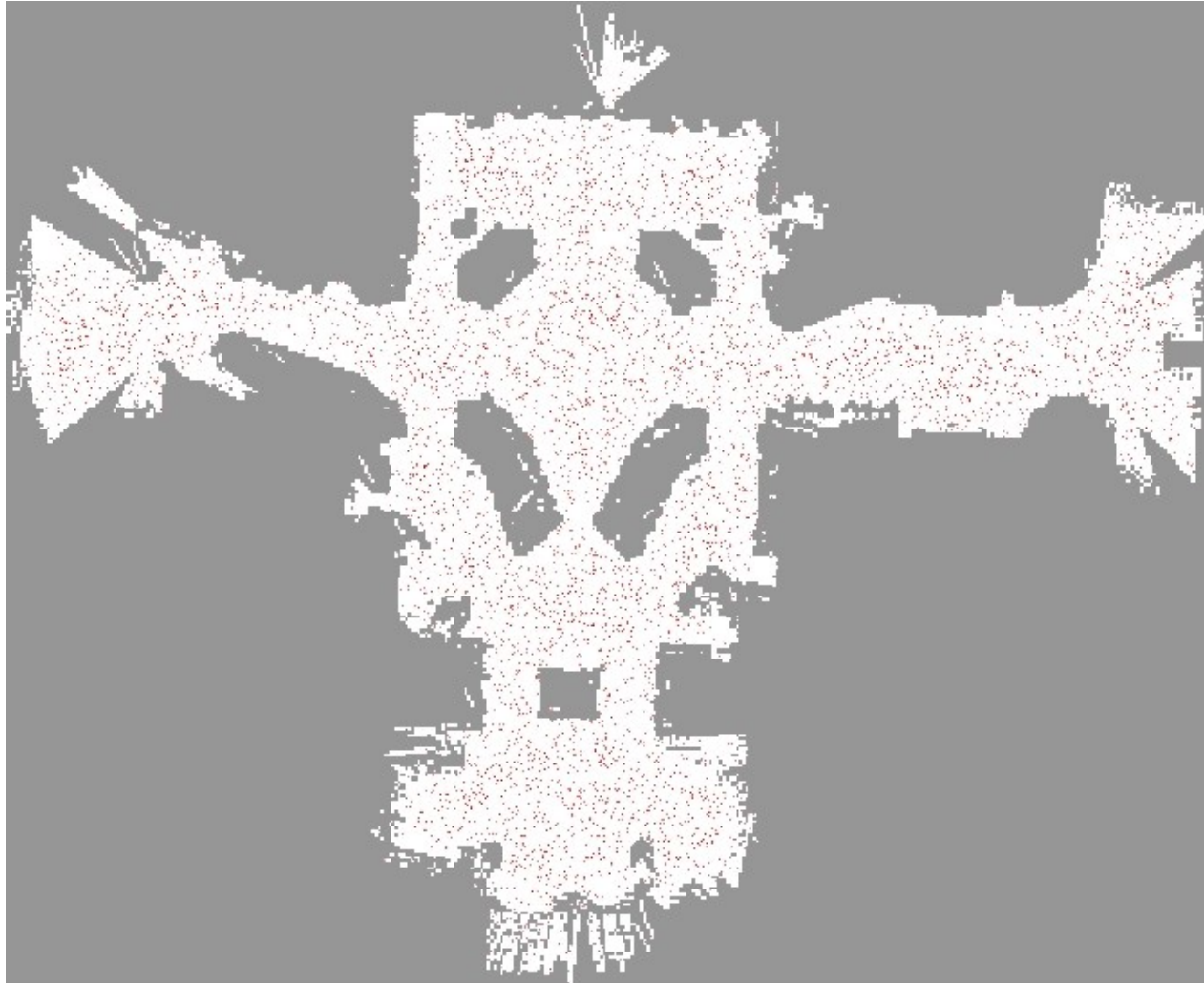
Filtro de Partículas

- Medición



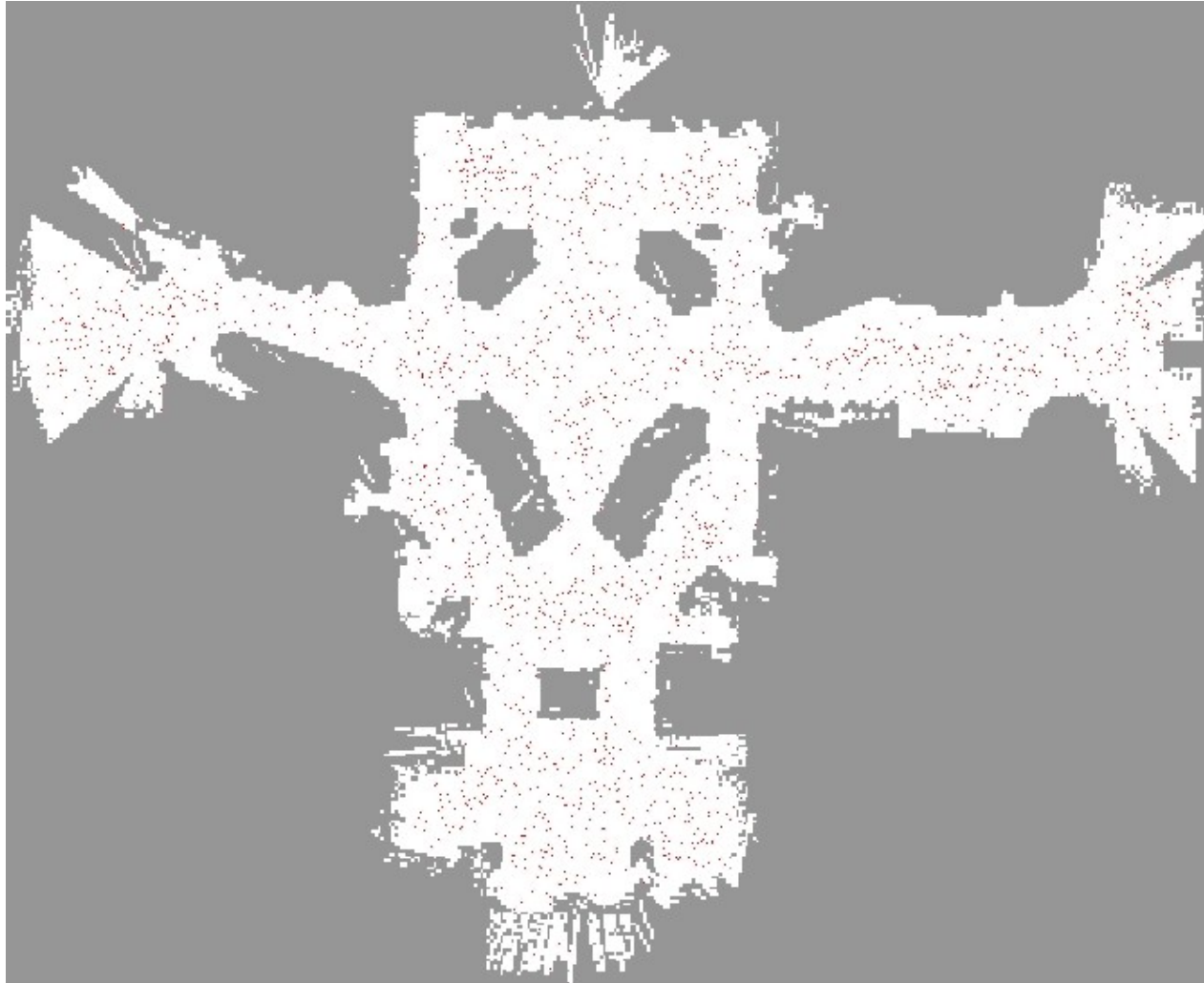
Filtro de Partículas

- Ponderación



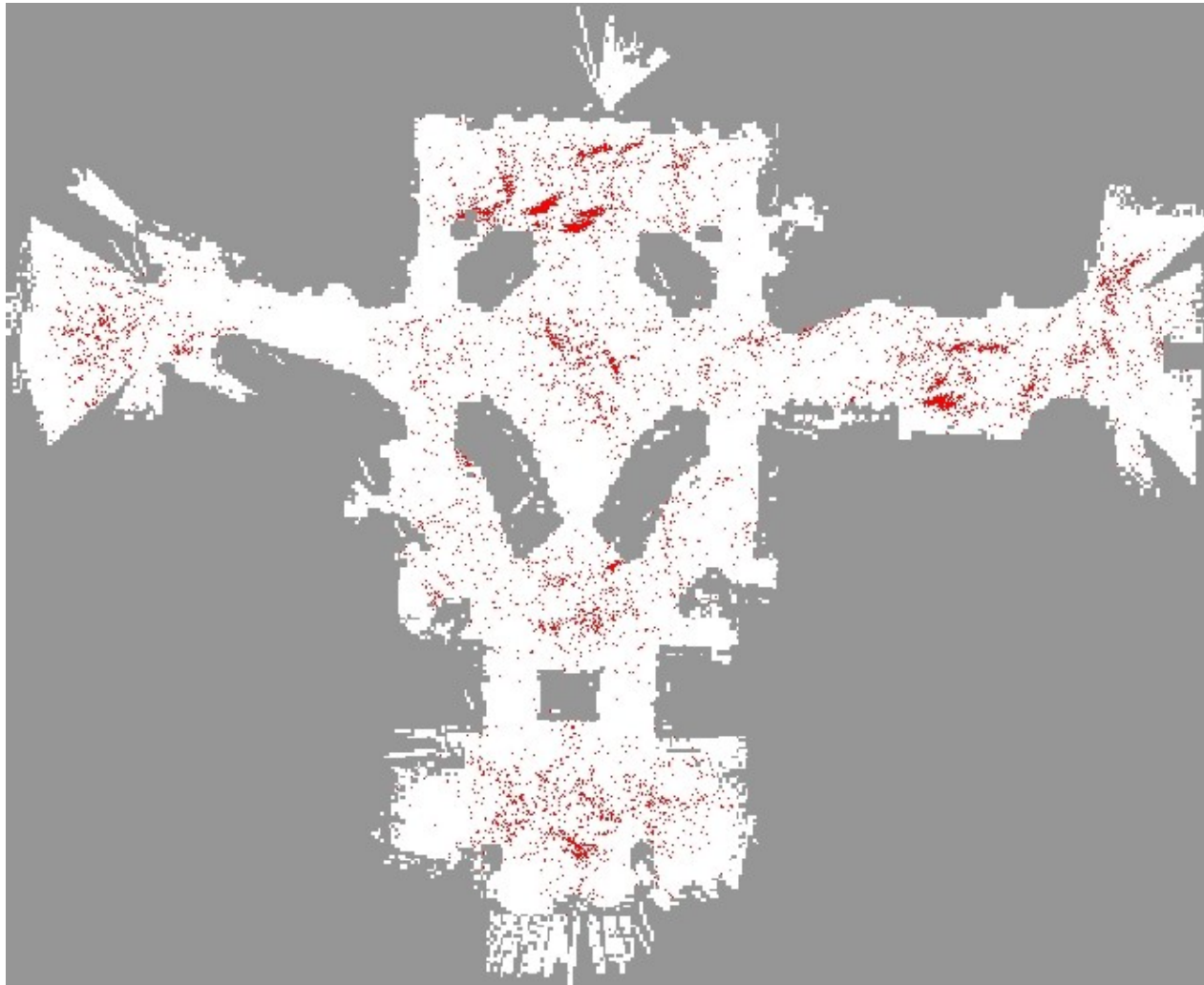
Filtro de Partículas

- Resampling



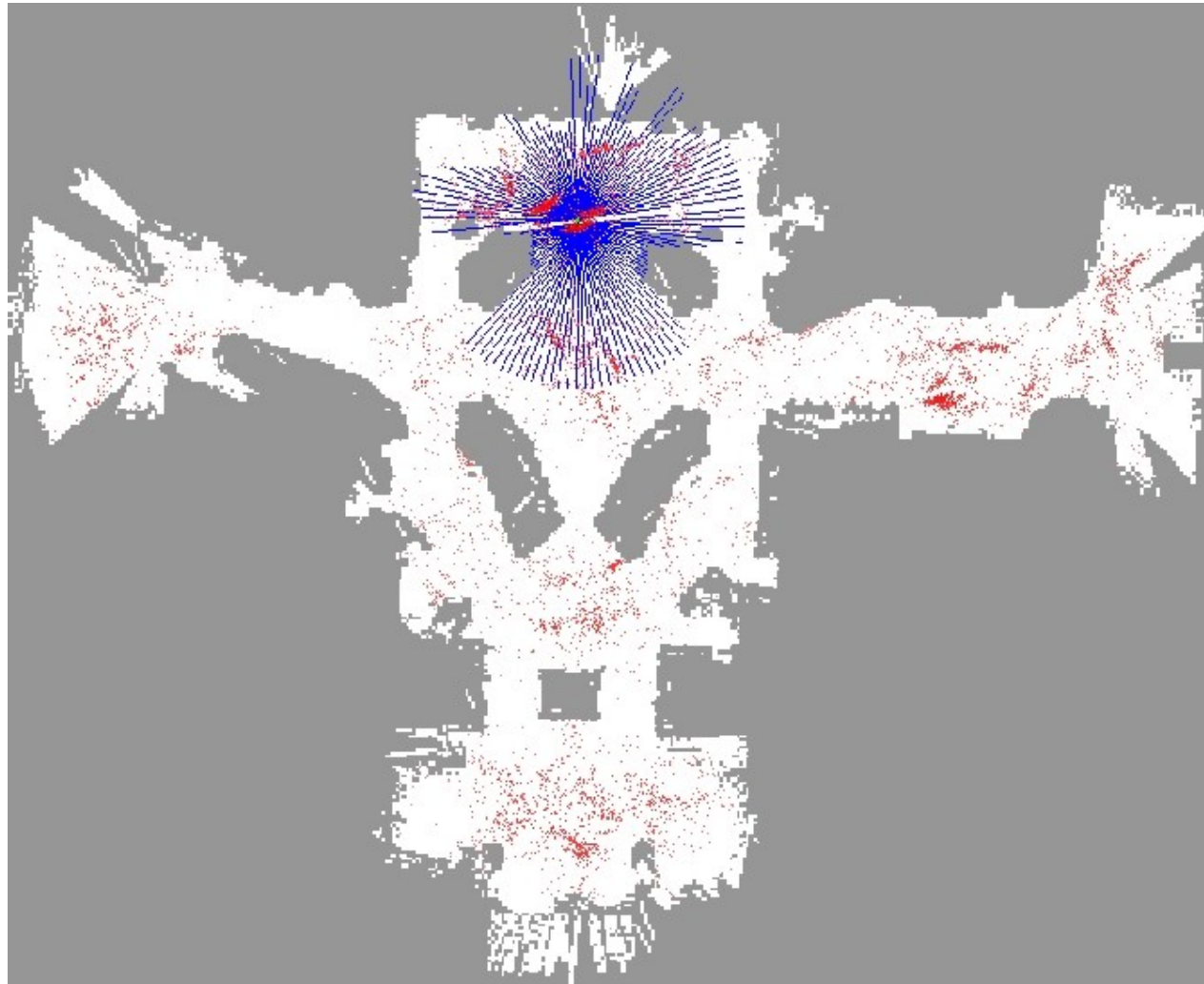
Filtro de Partículas

- Movimiento



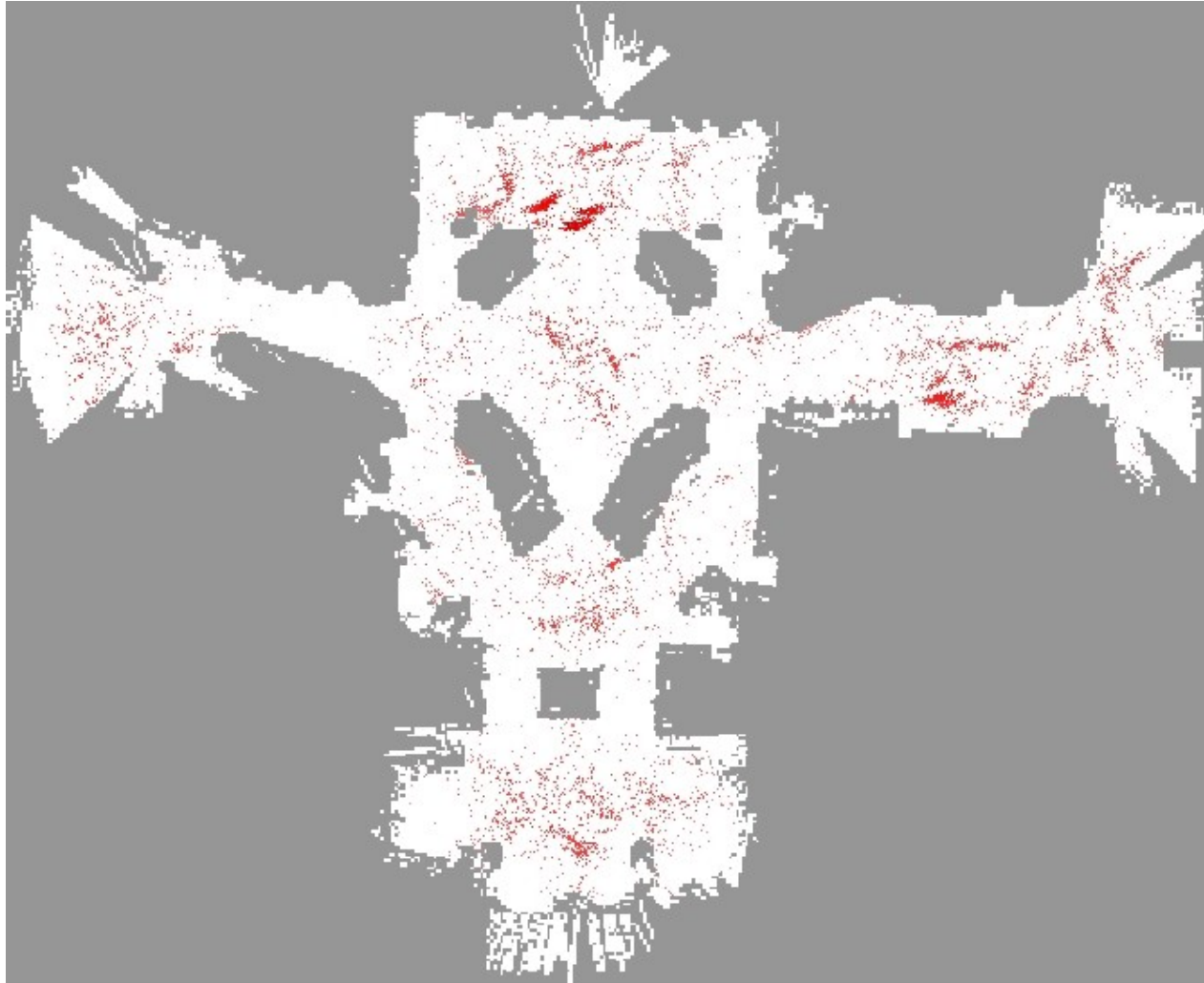
Filtro de Partículas

- Medición



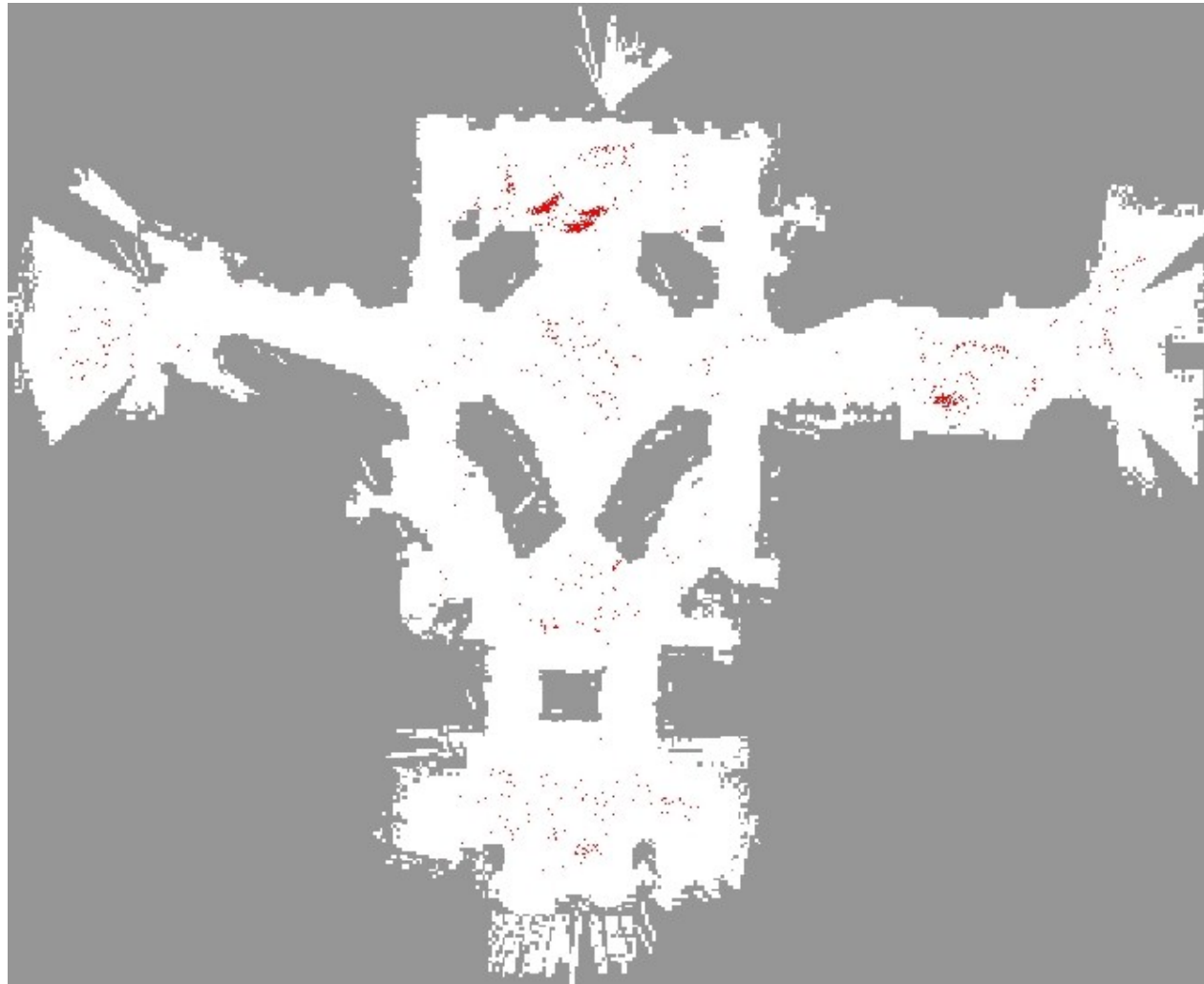
Filtro de Partículas

- Ponderación



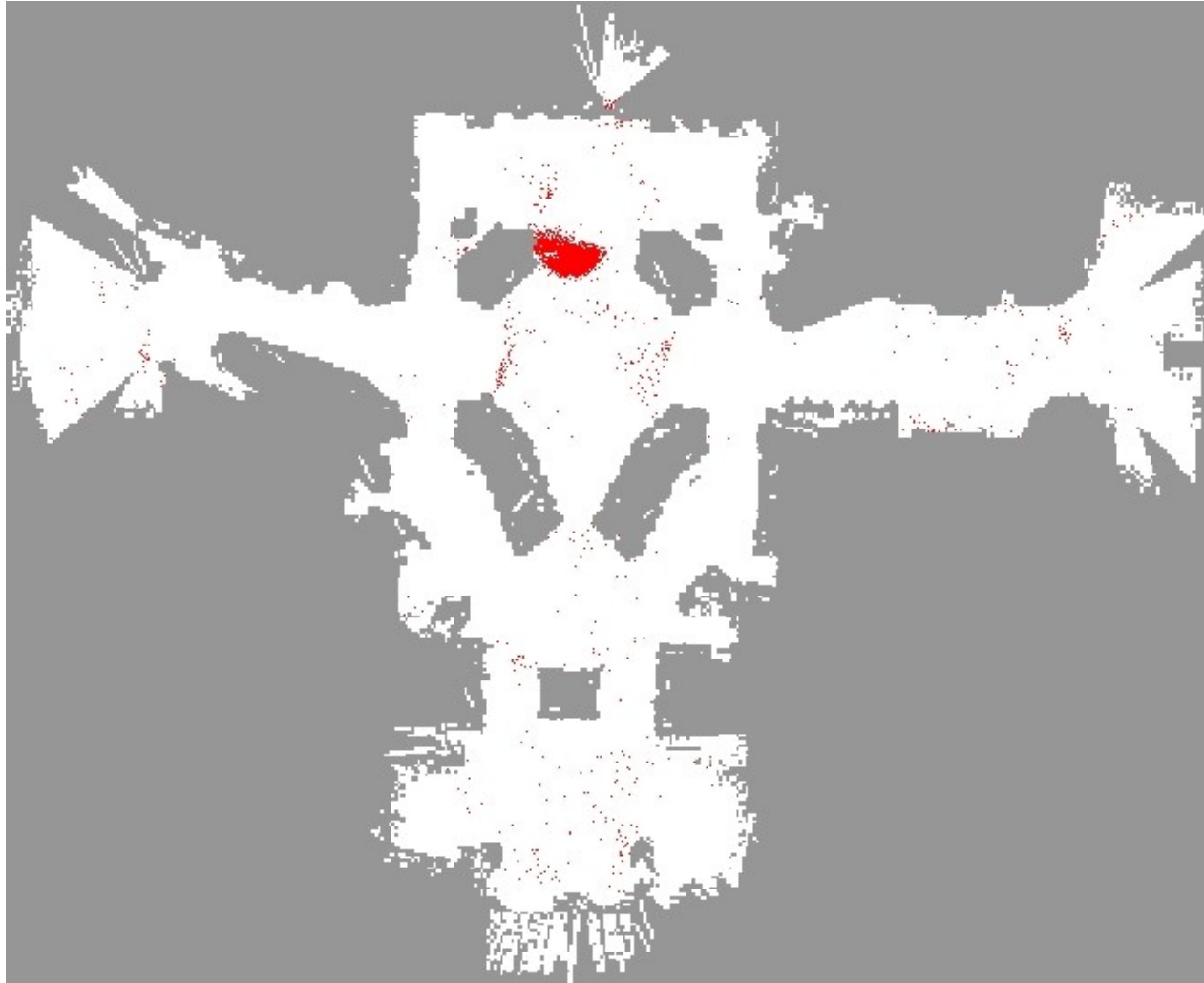
Filtro de Partículas

- Resampling



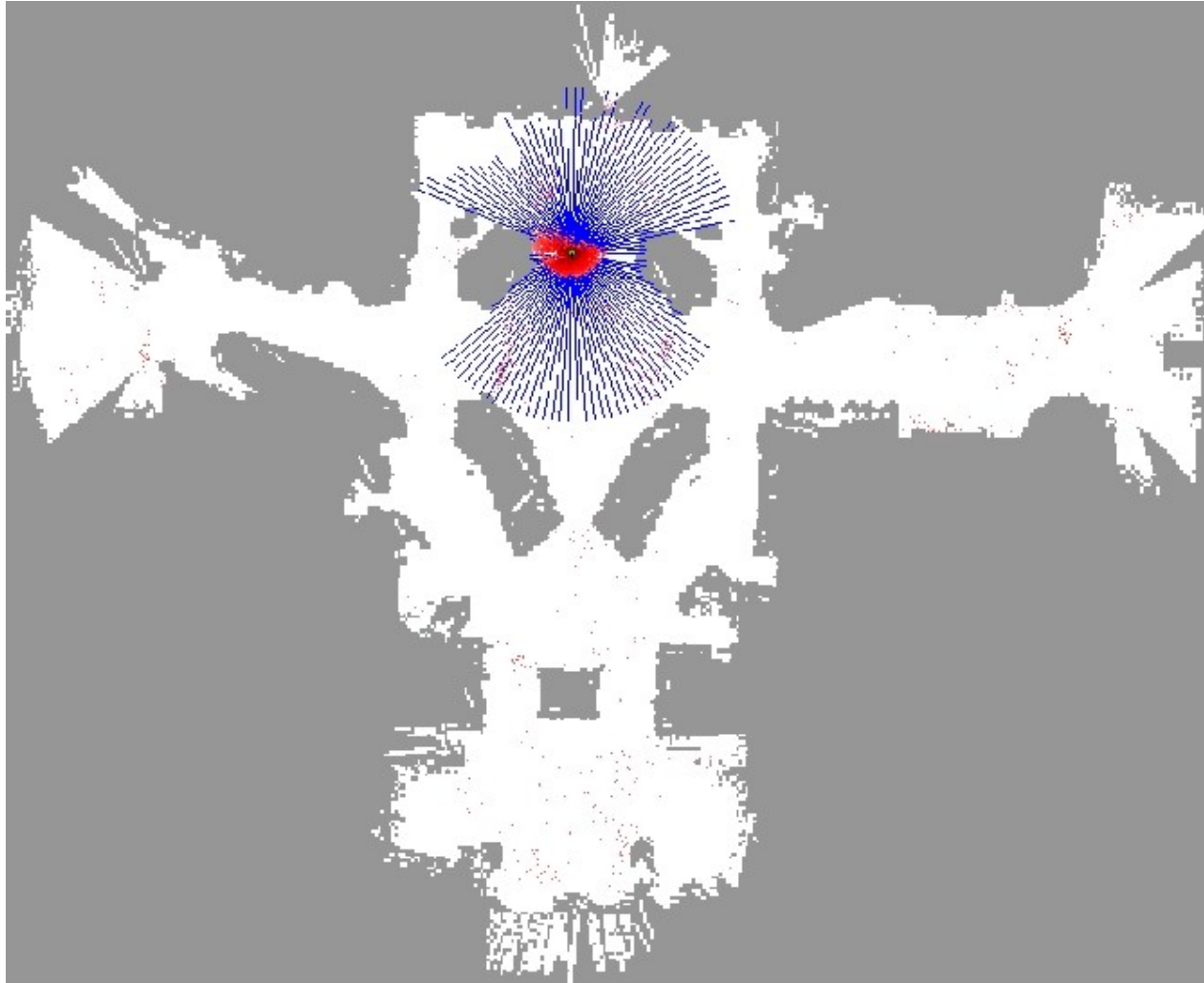
Filtro de Partículas

- Movimiento



Filtro de Partículas

- Medición



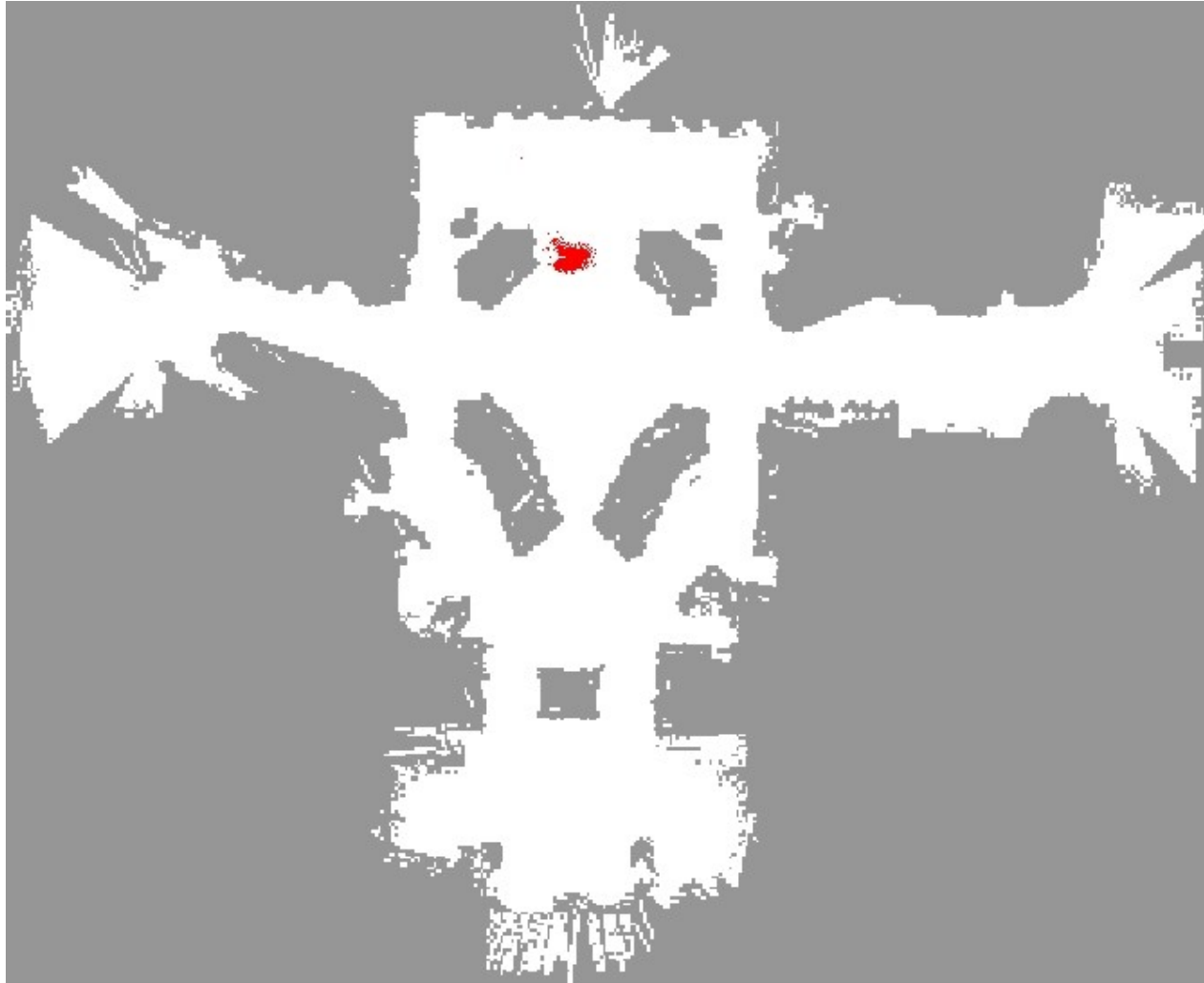
Filtro de Partículas

- Ponderación



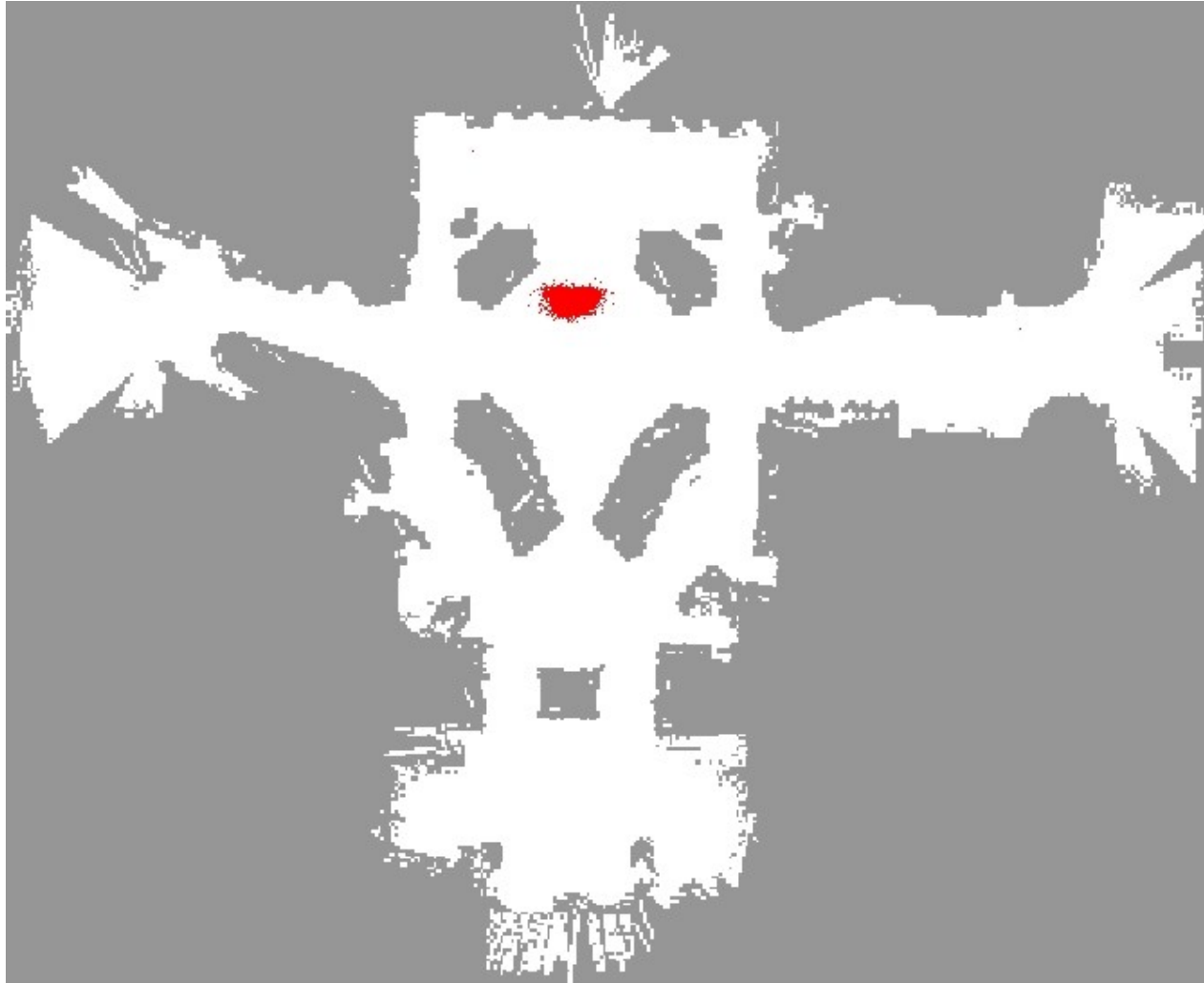
Filtro de Partículas

- Resampling



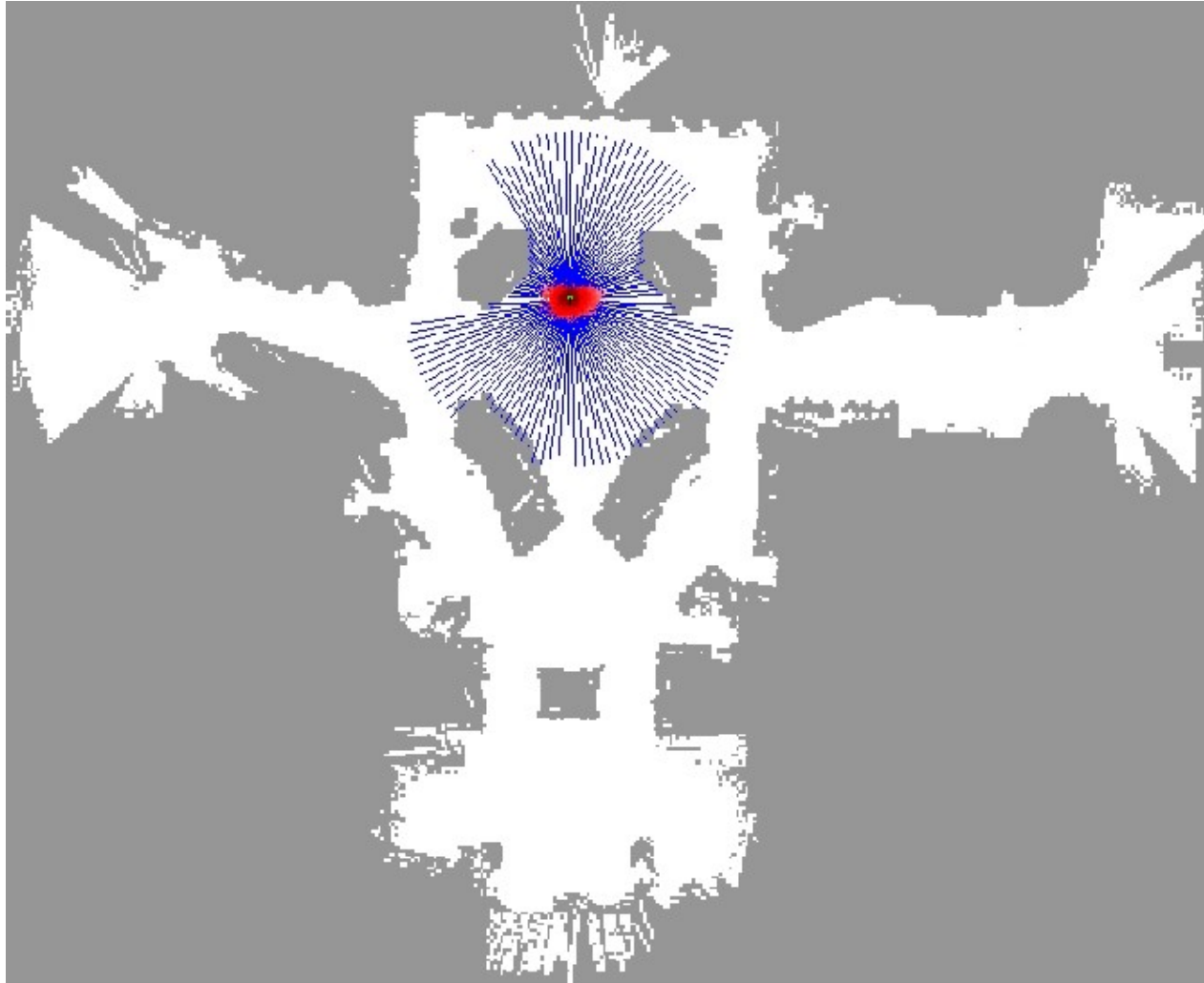
Filtro de Partículas

- Movimiento



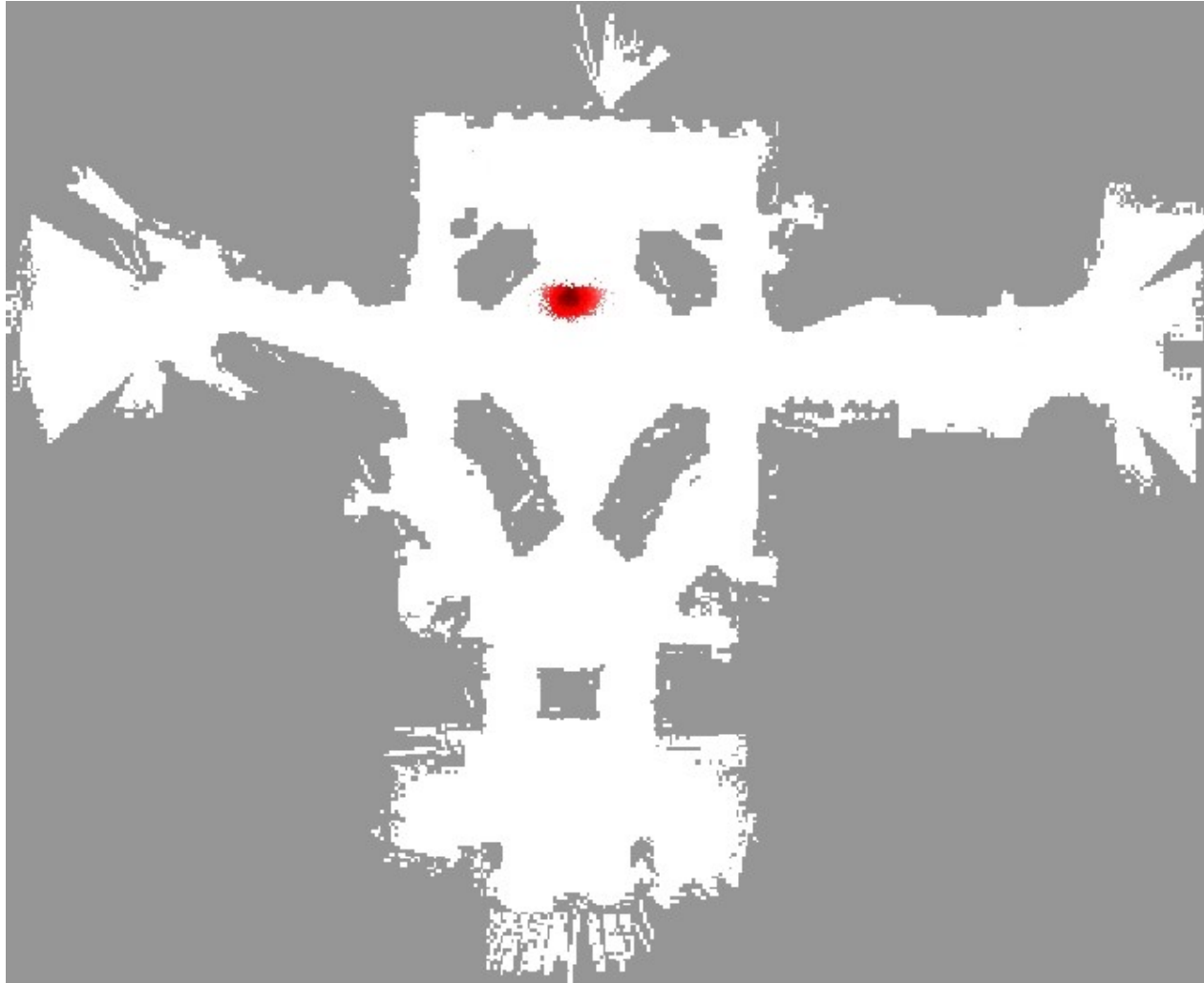
Filtro de Partículas

- Medición



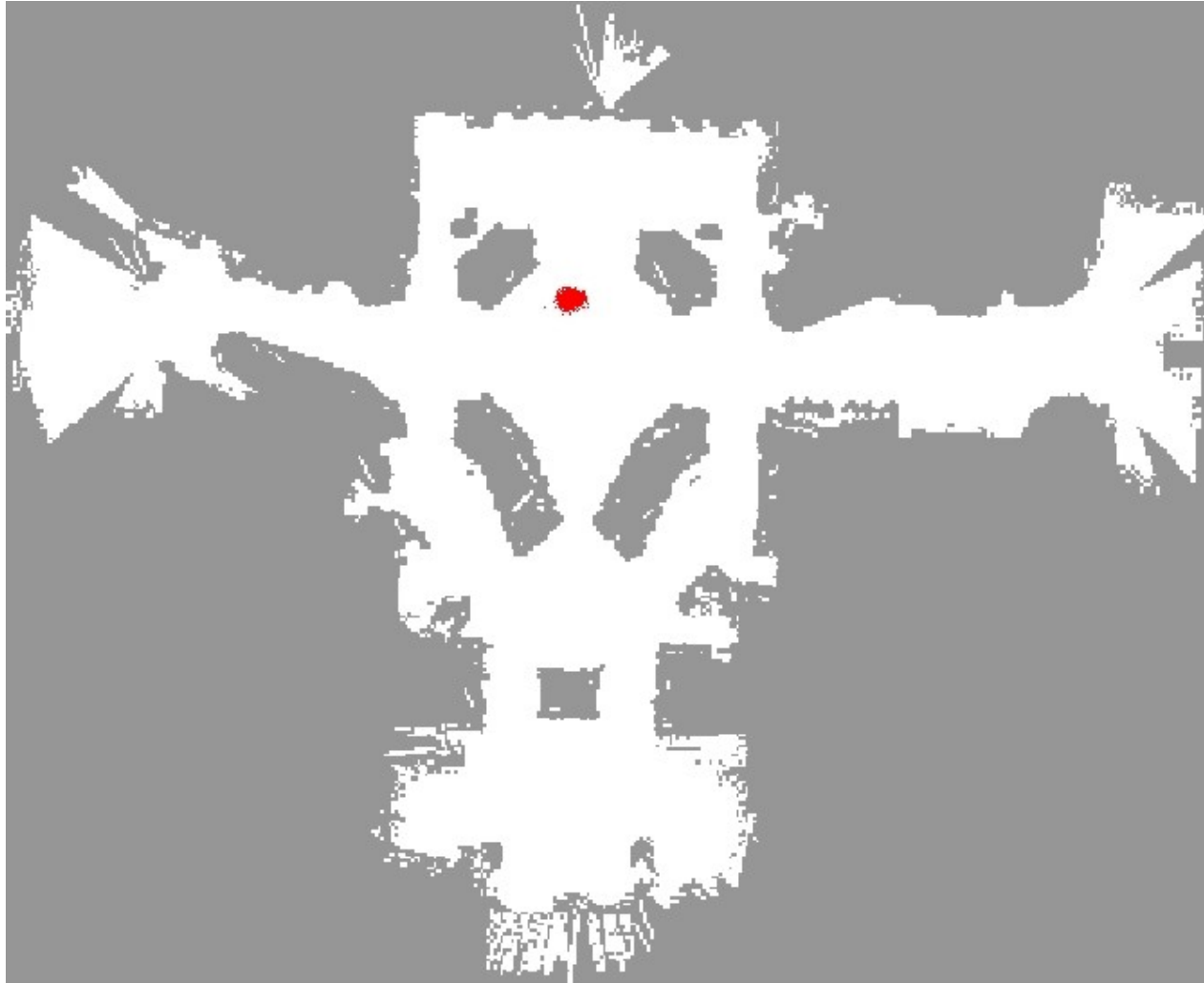
Filtro de Partículas

- Ponderación



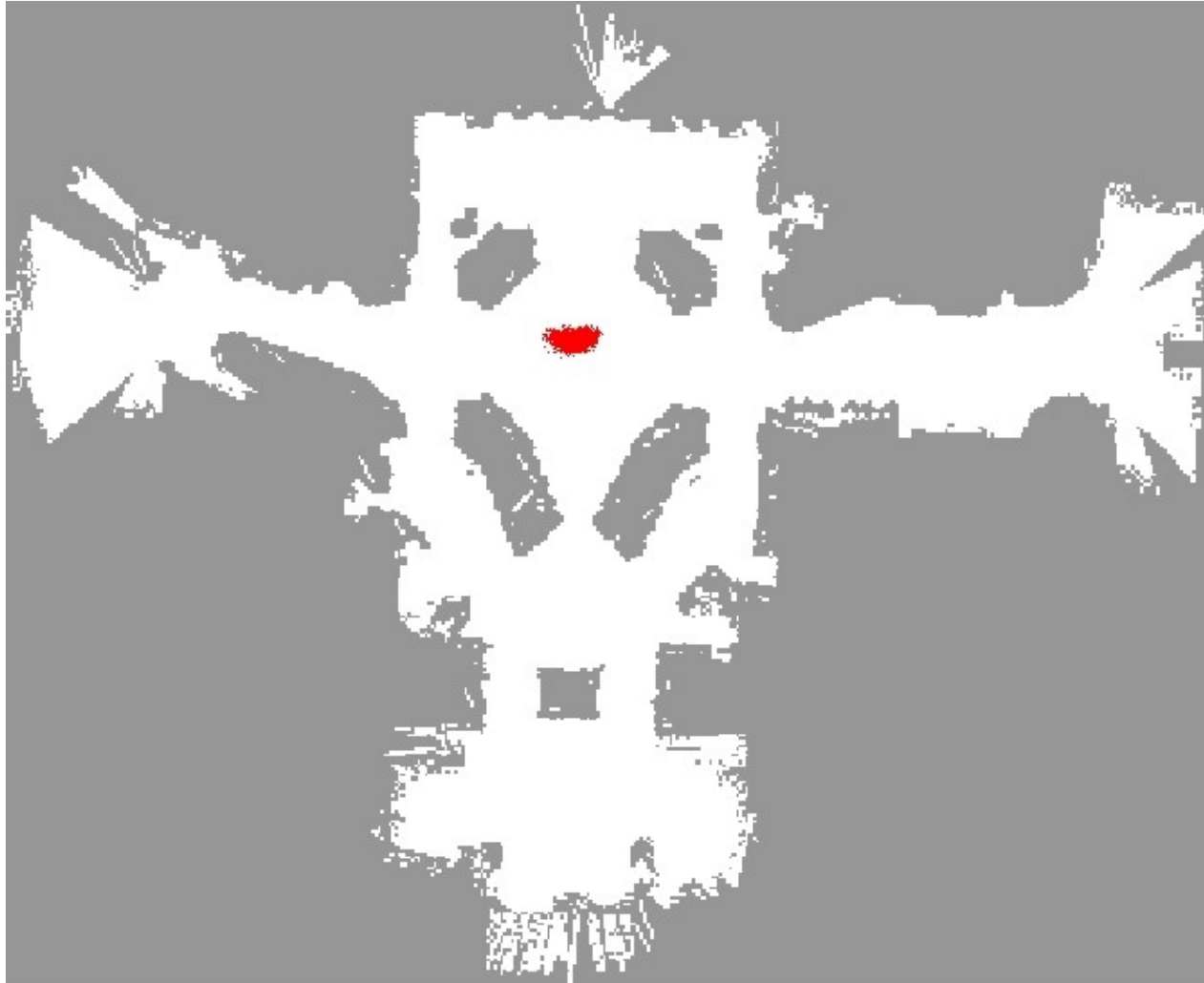
Filtro de Partículas

- Resampling



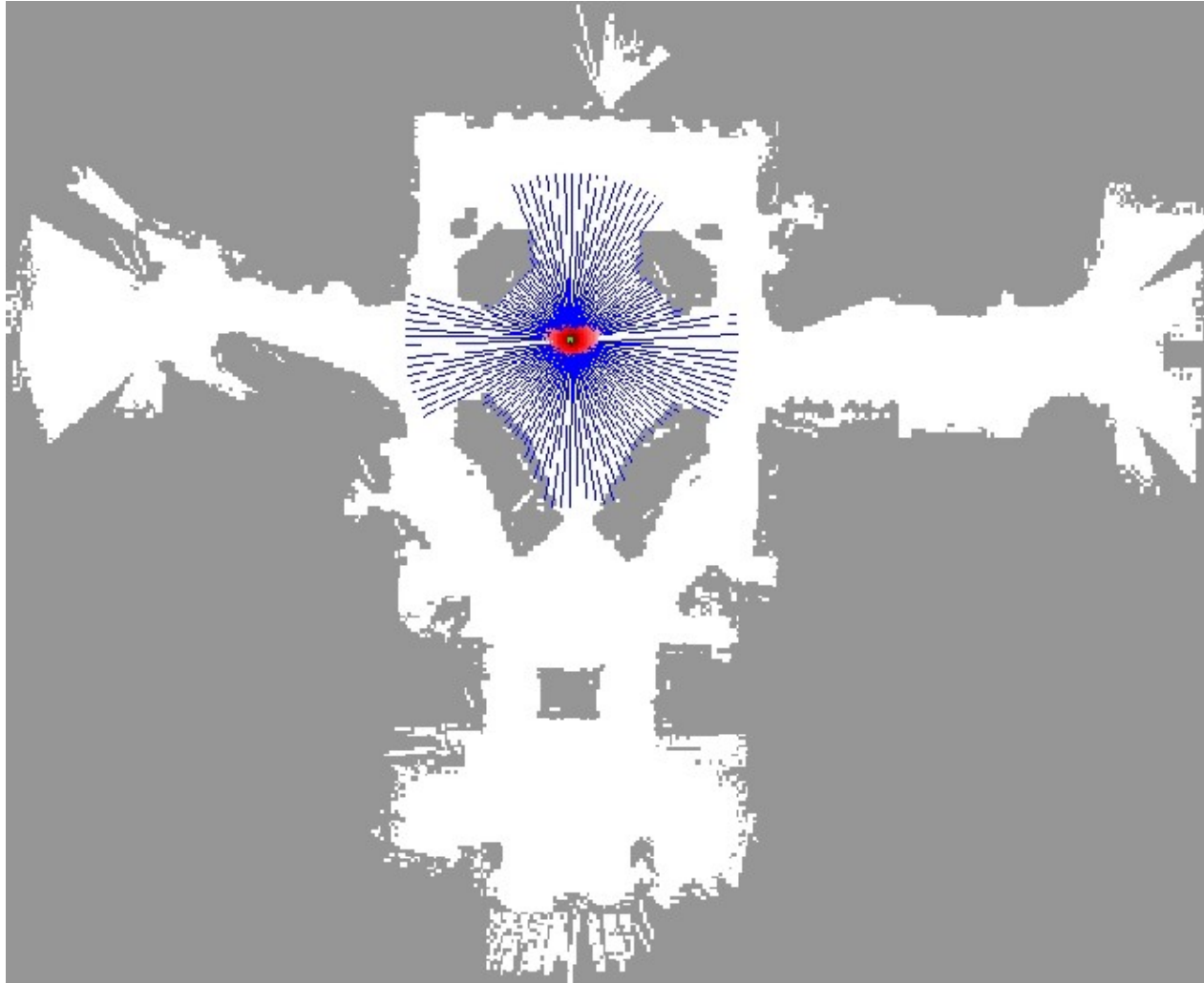
Filtro de Partículas

- Movimiento



Filtro de Partículas

- Medición



Resumen

- Se presenta problema general de localización
- Dos enfoques para resolver el problema de forma aproximada:
 - Basada en grilla
 - Filtro de Partículas
- Se presentan detalles de funcionamiento e implementación de filtro de partículas

Bibliografía

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