

# **Robótica Móvil**

## **un enfoque probabilístico**

### **Filtro de Bayes – Filtro de Partículas y localización de Monte Carlo**

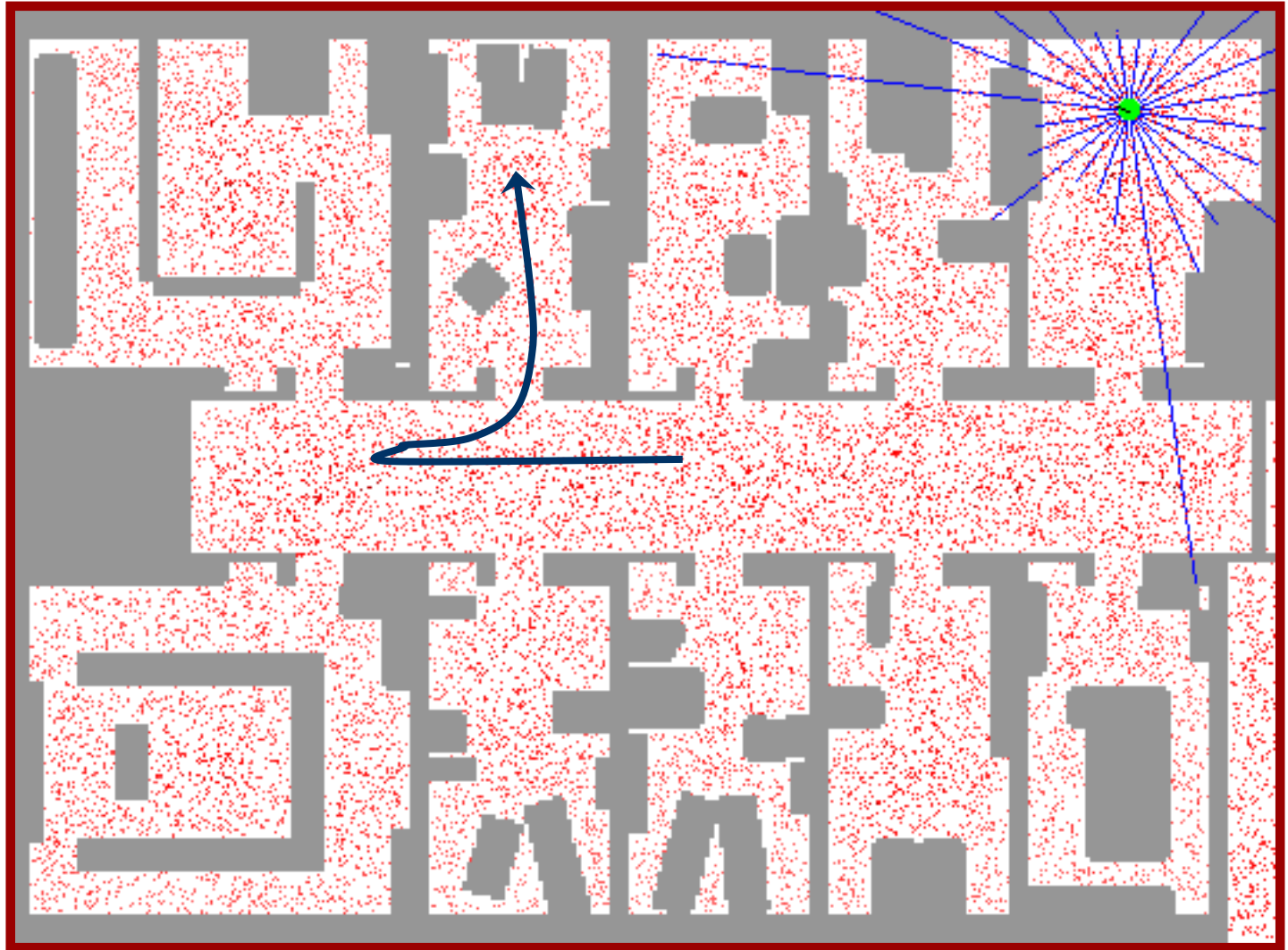
Ignacio Mas

---

# Contexto

- Ya vimos: Filtro Discreto
  - Discretiza el espacio de estados continuo
  - Alto requerimiento de memoria
  - Resolución fija (no se adapta a la estimación)
- Los filtros de partículas representan **eficientemente** distribuciones **no-Gaussianas**
- Principio básico
  - Conjunto de estados hipótesis (“partículas”)
  - Supervivencia del más apto

# Localización basada en muestras (sonar)



# Descripción Matemática

- Conjunto de muestras pesadas

$$S = \left\{ \left\langle s^{[i]}, w^{[i]} \right\rangle \mid i = 1, \dots, N \right\}$$

Estados hipótesis

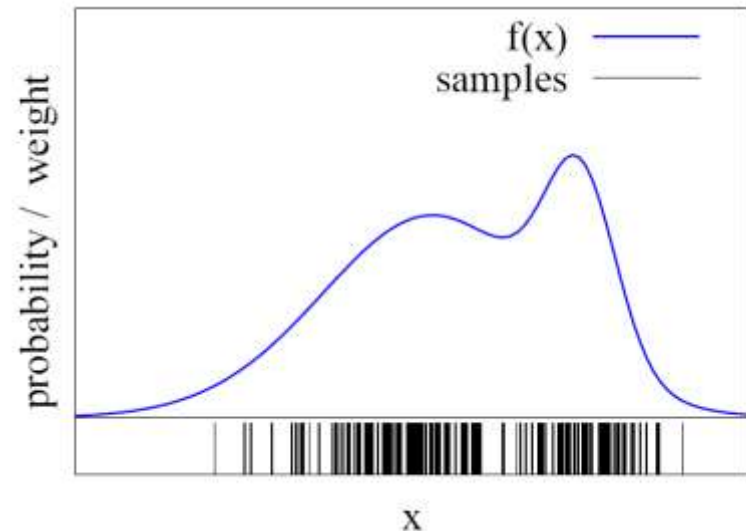
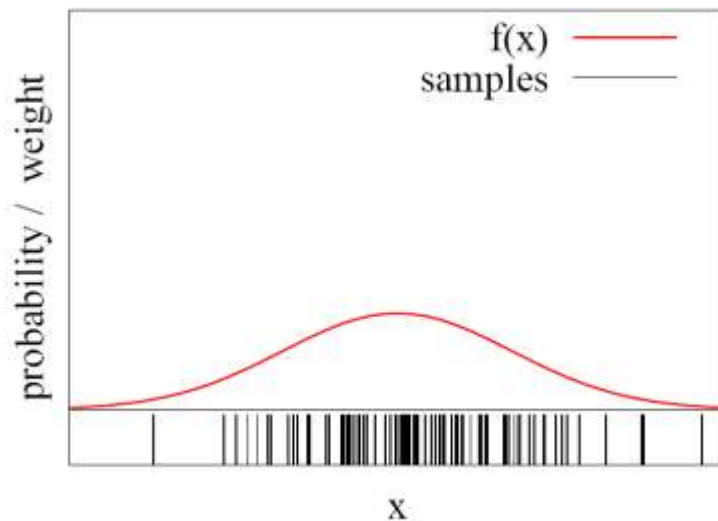
Peso de importancia

- Las muestras representan la distribución posterior

$$p(x) = \sum_{i=1}^N w_i \cdot \delta_{s^{[i]}}(x)$$

# Aproximación de funciones

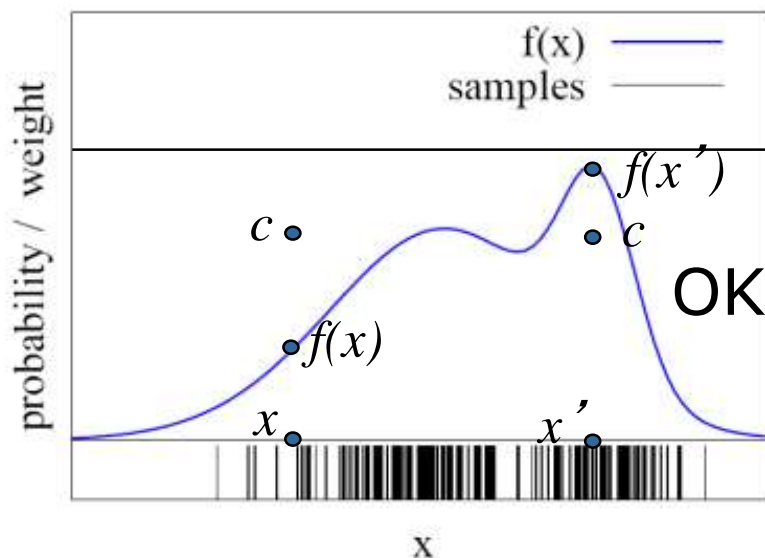
- Conjuntos de partículas pueden usarse para aproximar funciones



- Cuanto más partículas hay en un intervalo, mayor es la probabilidad de ese intervalo
- Cómo tomar muestras de una función/distribución?

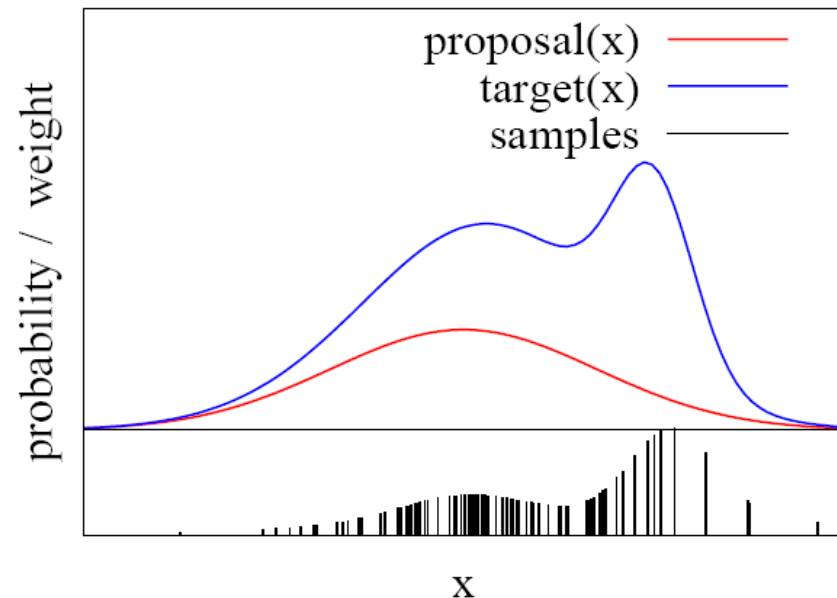
# Muestreo con rechazo

- Asumiendo que  $f(x) < 1$  para todo  $x$
- Muestrear  $x$  de una distribución uniforme
- Muestrear  $c$  de  $[0,1]$
- Si  $f(x) > c$  guardar la muestra  
sino rechazar la muestra



# Principio de Muestreo por Importancia

- Se puede usar una distribución  $g$  para generar muestras de  $f$
- Usando un peso de importancia  $w$ , se consideran las “diferencias entre  $g$  y  $f$ ”
- $w = f / g$
- $f$  es el objetivo (target)
- $g$  es la propuesta (proposal)
- Precondición:  
 $f(x) > 0 \rightarrow g(x) > 0$

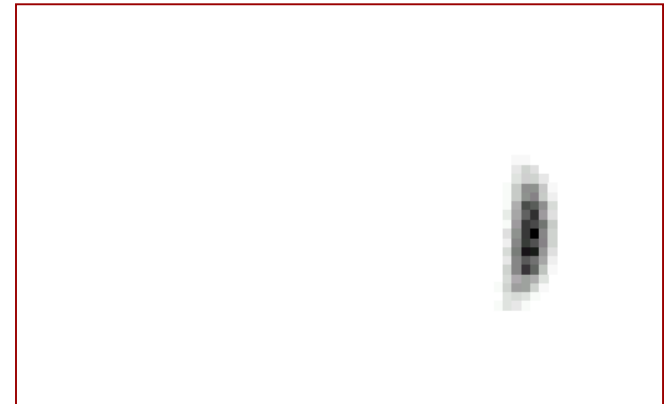
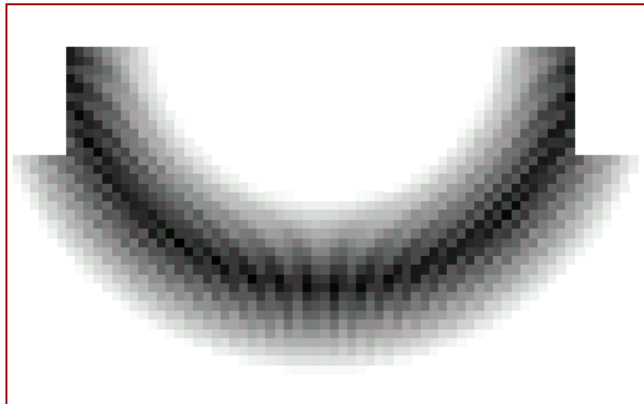
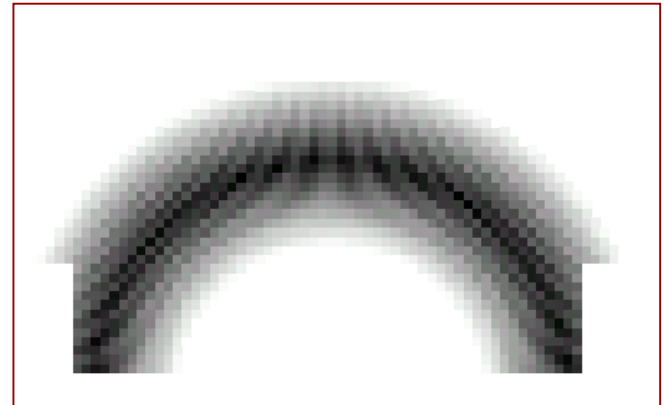
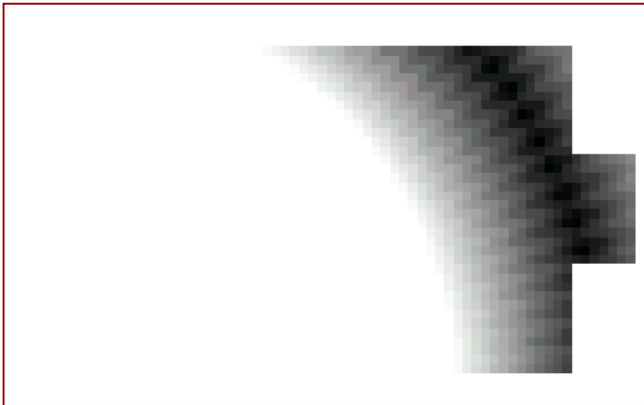
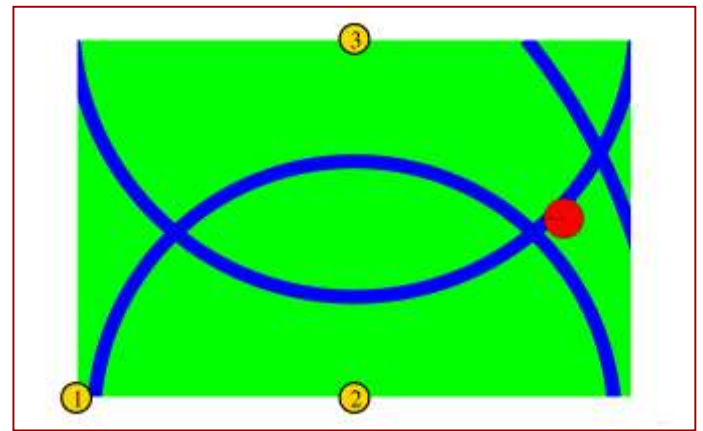


# Muestreo con Importancia con Remuestreo: Ejemplo de detección de Landmarks

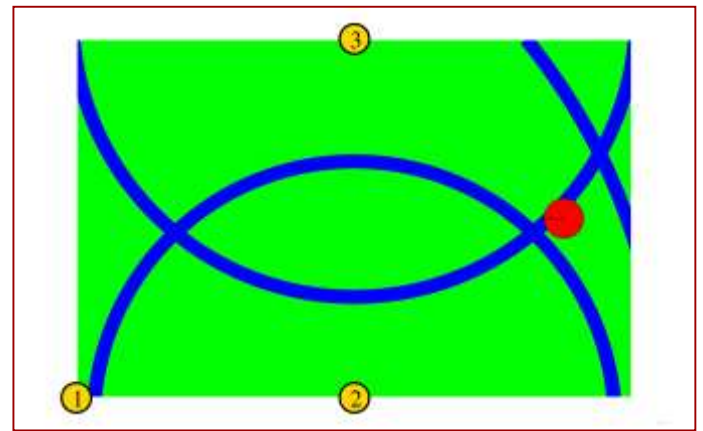




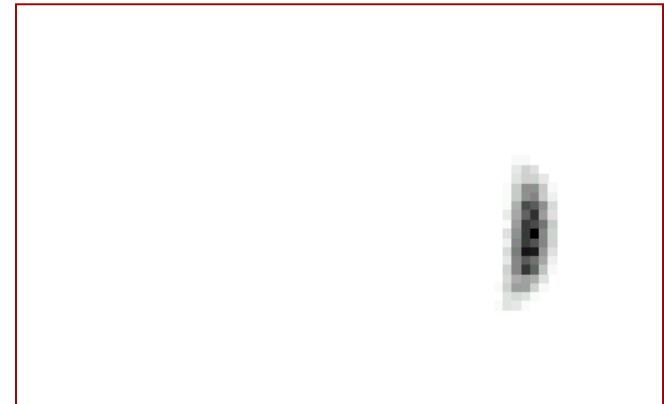
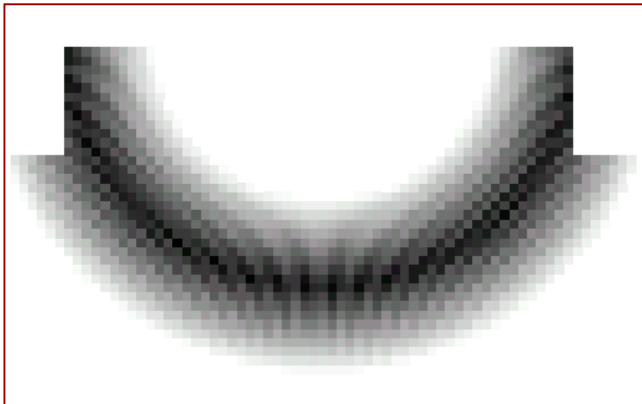
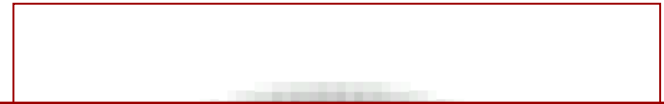
# Distribuciones



# Distribuciones

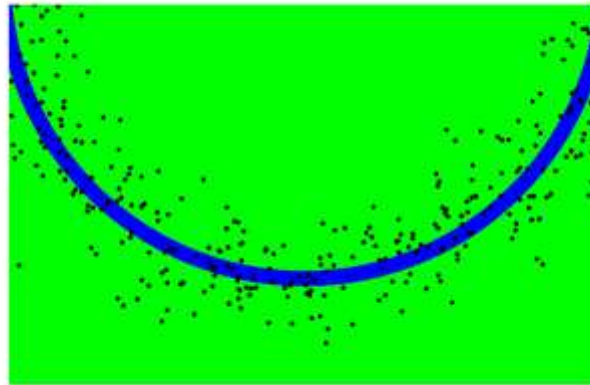
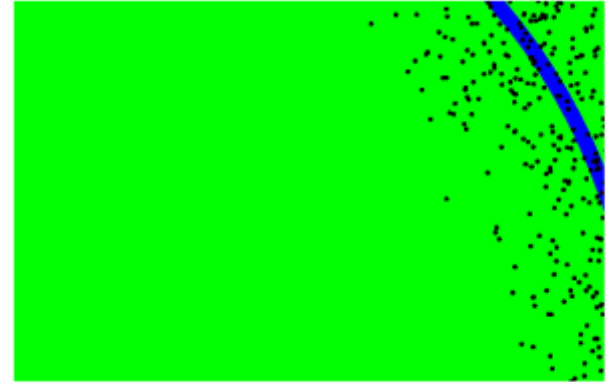
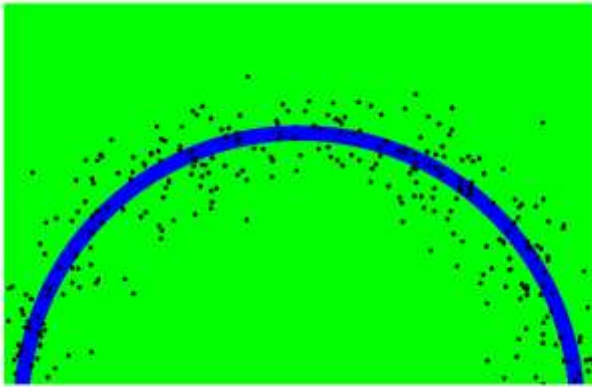


Se busca: muestras distribuidas según  
 $p(x | z_1, z_2, z_3)$



# Es simple!

Tomamos muestras de  $p(x|z_i)$  agregándole ruido a los parámetros de detección.



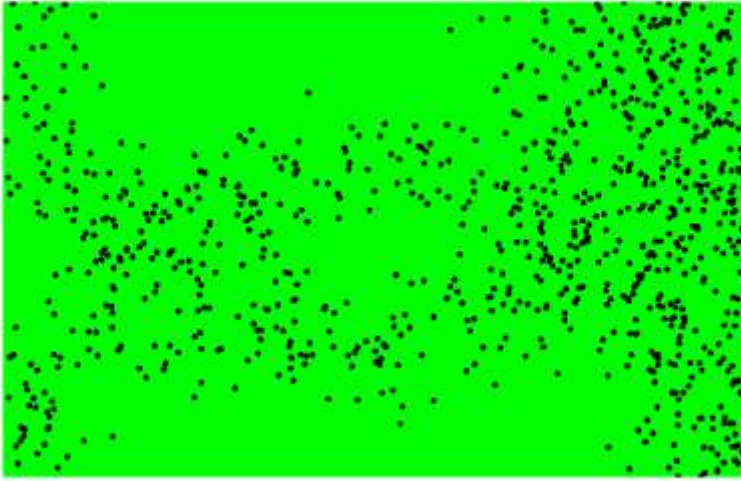
# Muestreo por Importancia

$$\text{Distribución objetivo (target) } f : p(x | z_1, z_2, \dots, z_n) = \frac{\prod_k p(z_k | x) p(x)}{p(z_1, z_2, \dots, z_n)}$$

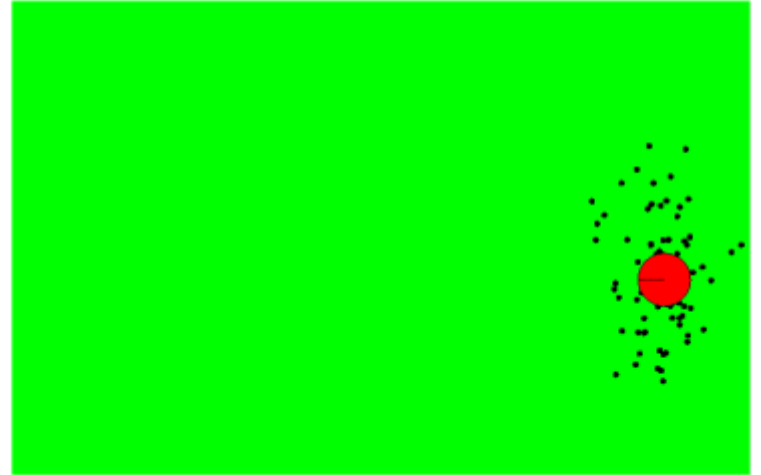
$$\text{Distribución de muestreo } g : p(x | z_l) = \frac{p(z_l | x) p(x)}{p(z_l)}$$

$$\text{Pesos de Importancia } w : \frac{f}{g} = \frac{p(x | z_1, z_2, \dots, z_n)}{p(x | z_l)} = \frac{p(z_l) \prod_{k \neq l} p(z_k | x)}{p(z_1, z_2, \dots, z_n)}$$

# Muestreo con Importancia con Remuestreo

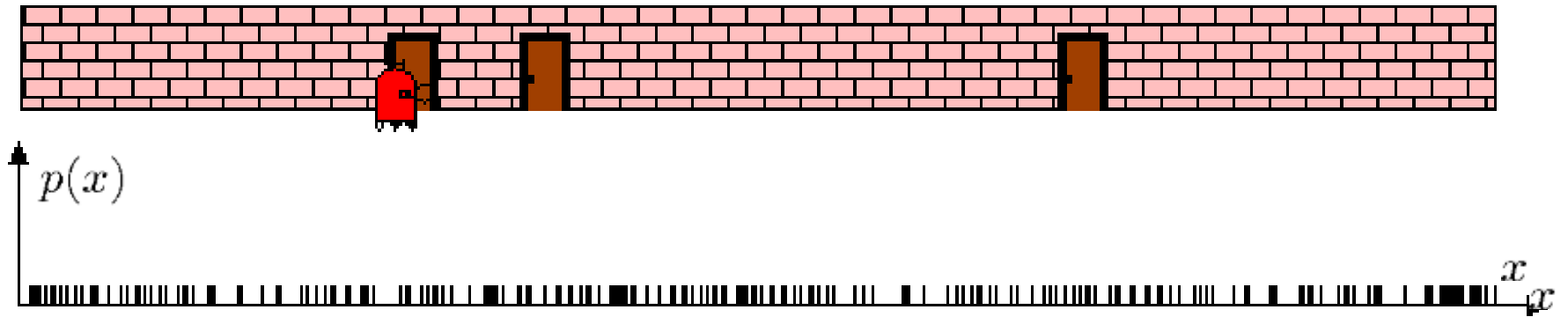


Muestras pesadas



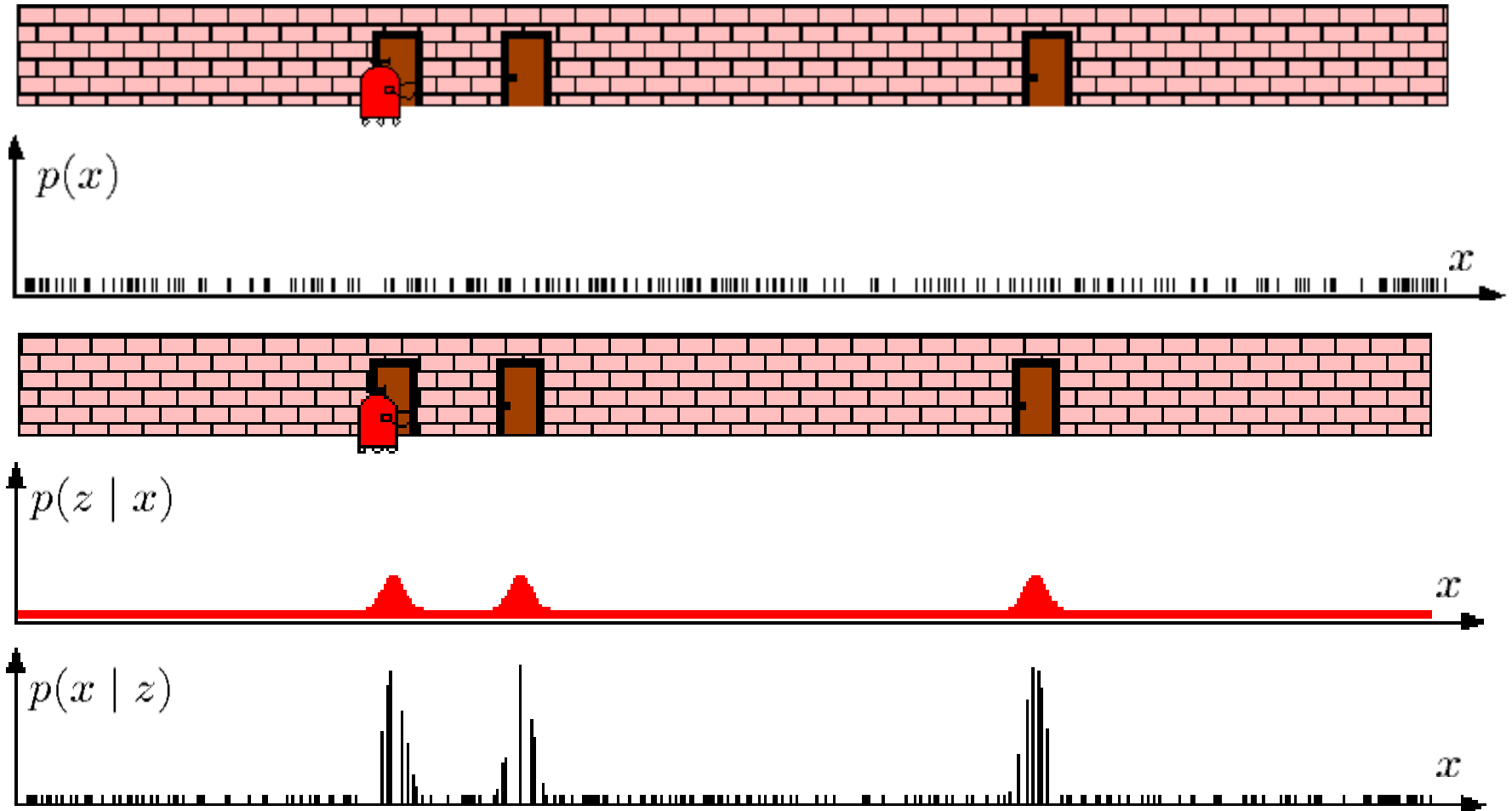
Después del remuestreo

# Filtro de Partículas



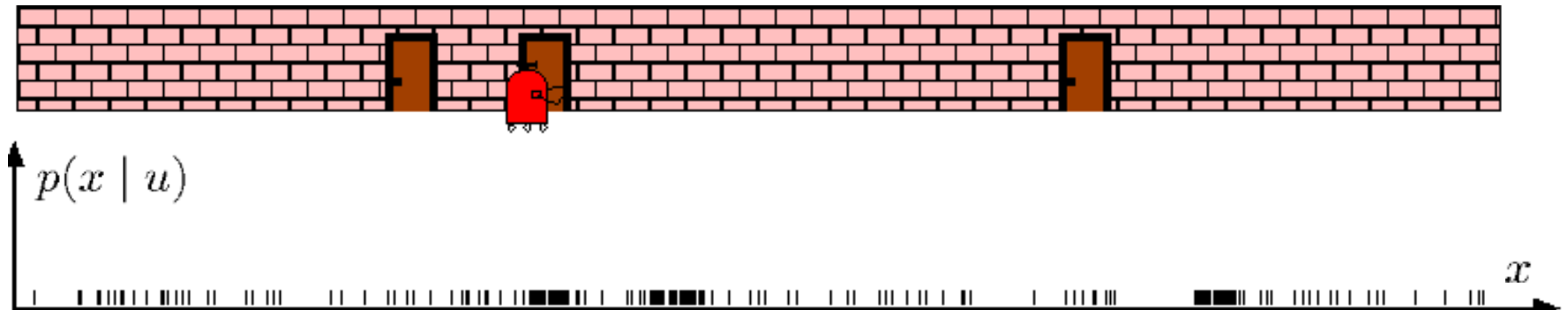
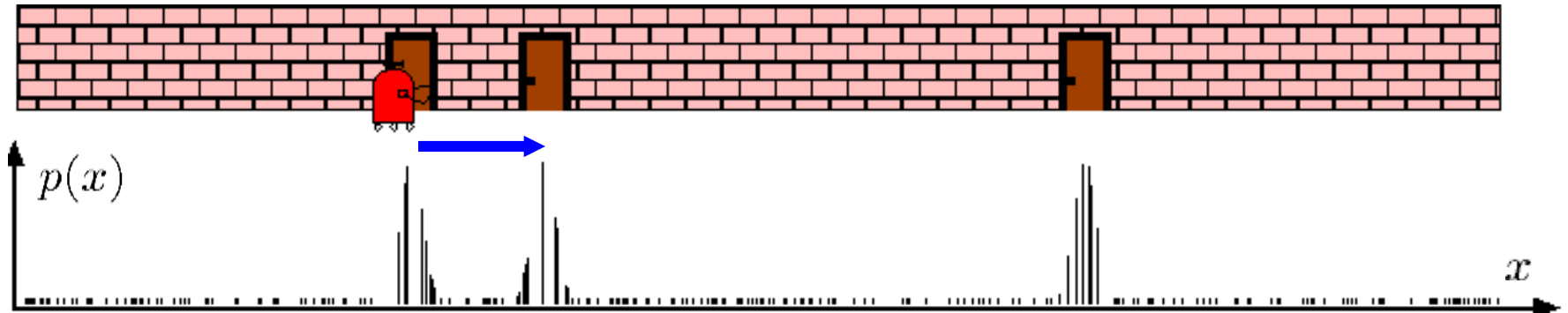
# Información de sensores: Muestreo con Importancia

$$\begin{aligned} Bel(x) &\leftarrow \alpha p(z | x) Bel^-(x) \\ w &\leftarrow \frac{\alpha p(z | x) Bel^-(x)}{Bel^-(x)} = \alpha p(z | x) \end{aligned}$$



# Movimiento del robot

$$Bel^-(x) \rightarrow \int p(x | u, x') Bel(x') dx'$$

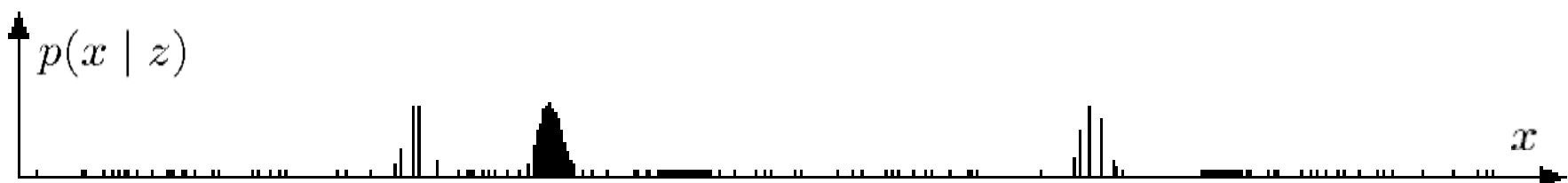
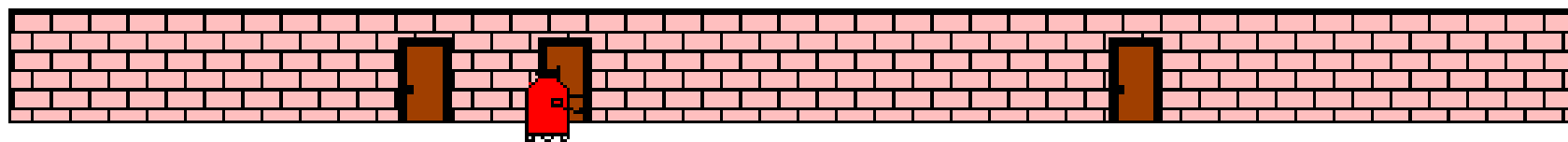
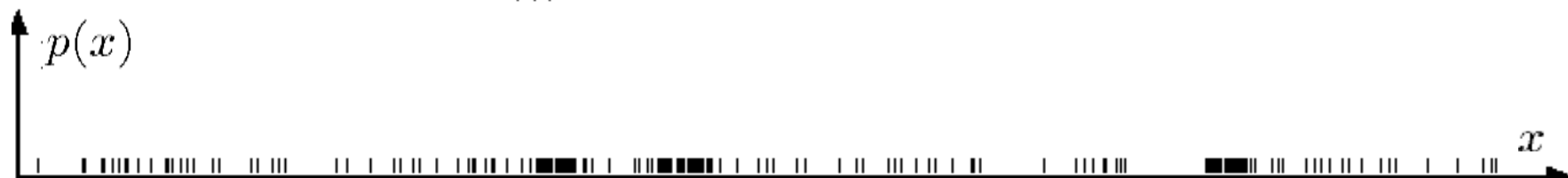
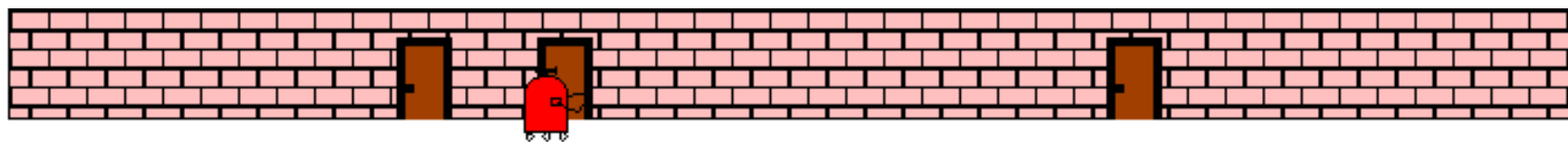




# Información de sensores: Muestreo con Importancia

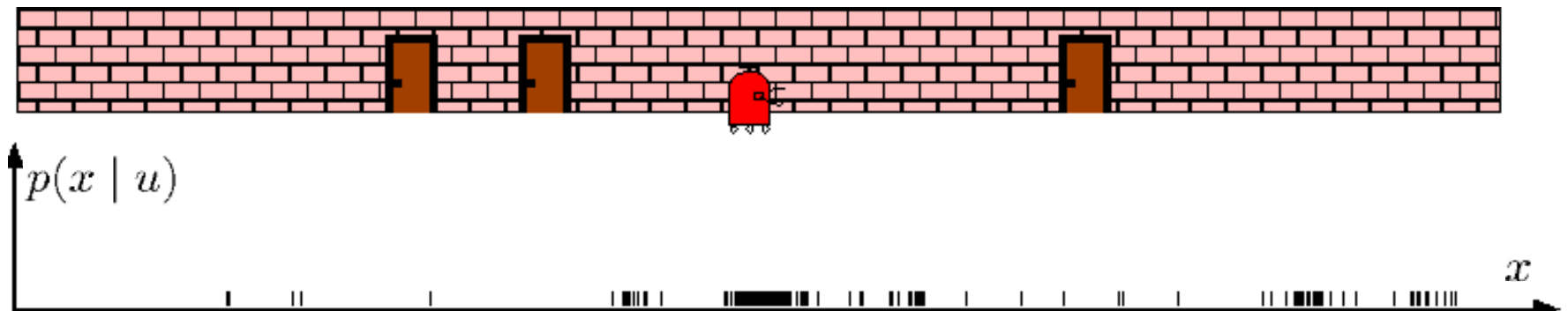
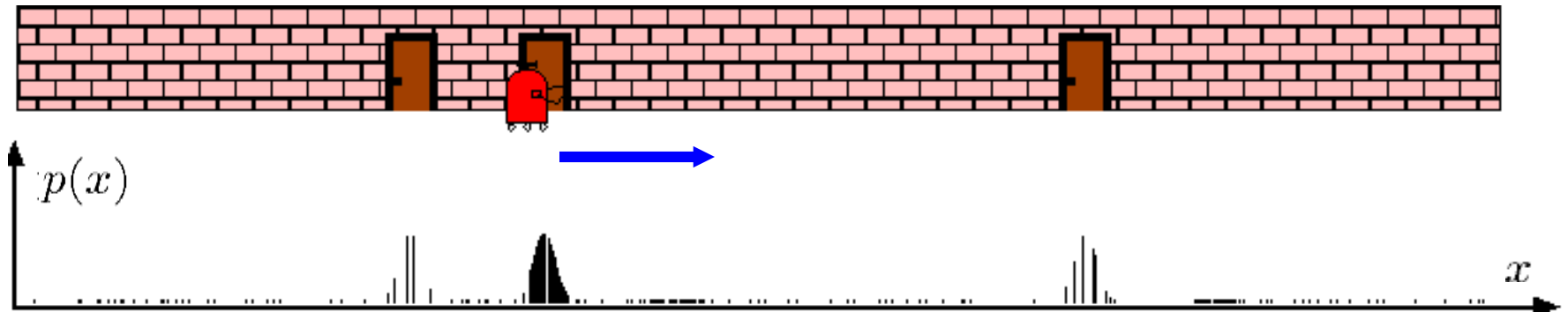
$$Bel(x) \propto a p(z | x) Bel^-(x)$$

$$w \propto \frac{a p(z | x) Bel^-(x)}{Bel^-(x)} = a p(z | x)$$



# Movimiento del robot

$$Bel^-(x) \propto \int p(x | u, x') Bel(x') dx'$$



# Algoritmo del Filtro de Partículas

- Muestrear la próxima generación de partículas usando la distribución propuesta
- Calcular los pesos de importancia :  
$$\text{peso } w = \text{distribución objetivo} / \text{distribución propuesta}$$
- Remuestreo: “Reemplazar muestras poco probables por otras más probables”

# Algoritmo del Filtro de Partículas

1. Algoritmo **particle\_filter**(  $S_{t-1}, u_t, z_t$ ):
2.  $S_t = \emptyset, \quad \eta = 0$
3. **For**  $i = 1, \dots, n$  *Generar nuevas muestras*
4.     Muestrear índice  $j(i)$  de distribución discreta dada por  $w_{t-1}$
5.     Muestrear  $x_t^i$  de  $p(x_t | x_{t-1}, u_t)$  usando  $x_{t-1}^{j(i)}$  y  $u_t$
6.      $w_t^i = p(z_t | x_t^i)$  *Calcular pesos de importancia*
7.      $h = h + w_t^i$  *Factor de normalización*
8.      $S_t = S_t \cup \{ \langle x_t^i, w_t^i \rangle \}$  *Agregar a nuevo conj. de*
9. **For**  $i = 1, \dots, n$  *partículas*
10.      $w_t^i = w_t^i / h$  *Normalizar pesos*
11. return  $S_t$

# Algoritmo del Filtro de Partículas

$$Bel(x_t) = \frac{1}{N} \sum_{i=1}^N p(z_t | x_t) p(x_t | x_{t-1}^i, u_t) Bel(x_{t-1}^i)$$

tomar muestra  $x_{t-1}^i$  de  $Bel(x_{t-1})$

propagar muestra  $x_{t-1}^i$  con  $p(x_t | x_{t-1}^i, u_t)$

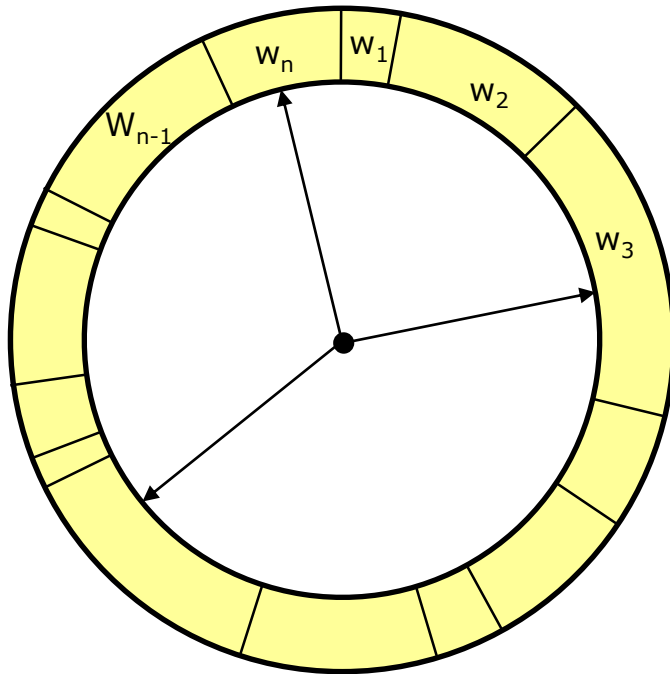
Factor de importancia para  $x_t^i$ :

$$\begin{aligned} w_t^i &= \frac{\text{distribución objetivo}}{\text{distribución propuesta}} \\ &= \frac{\eta p(z_t | x_t) p(x_t | x_{t-1}^i, u_t) Bel(x_{t-1}^i)}{p(x_t | x_{t-1}^i, u_t) Bel(x_{t-1}^i)} \\ &\propto p(z_t | x_t) \end{aligned}$$

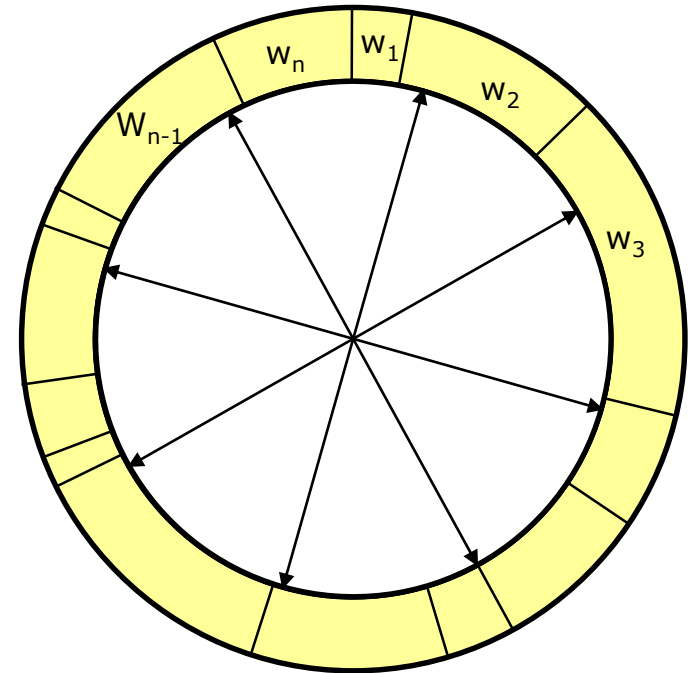
# Remuestreo

- **Dado**: un conjunto  $S$  de muestras pesadas.
- **Se desea** : una muestra aleatoria, donde la probabilidad de tomar  $x_i$  está dada por  $w_i$ .
- Haciéndolo  $n$  veces con reemplazo para generar un nuevo conjunto de muestras  $S'$ .

# Remuestreo



- Rueda de Ruleta
- Búsqueda binaria
- Complejidad:  $O(n \log n)$



- Muestreo Estocástico Universal (SUS)
- Remuestreo sistemático
- Complejidad: lineal  $O(n)$
- Fácil de implementar, baja varianza

# Algoritmo de remuestreo

1. Algoritmo **systematic\_resampling**( $S, n$ ):

2.  $S' = \emptyset, c_1 = w^1$

3. **For**  $i = 2 \dots n$

*Generar cdf*

4.  $c_i = c_{i-1} + w^i$

5.  $u_1 \sim U[0, n^{-1}], i = 1$

*Inicializar umbral*

6. **For**  $j = 1 \dots n$

*Tomar muestras ...*

7. **While** (  $u_j > c_i$  )

*Saltear hasta el próximo umbral*

8.  $i = i + 1$

9.  $S' = S' \cup \{x^i, n^{-1}\}$

*Insertar*

10.  $u_{j+1} = u_j + n^{-1}$

*Incrementar umbral*

11. **Return**  $S'$

También llamado **SUS** (stochastic universal sampling)

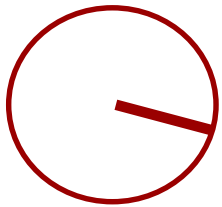


# Localización de robots móviles

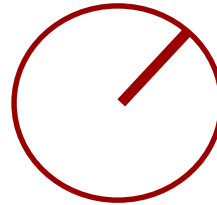
- Cada partícula es una pose potencial del robot
- La distribución propuesta (proposal) es el modelo de movimiento del robot (paso de predicción)
- El modelo de medición se usa para calcular los pesos de importancia (paso de corrección)

# Repaso de modelos de movimiento

Pose inicial

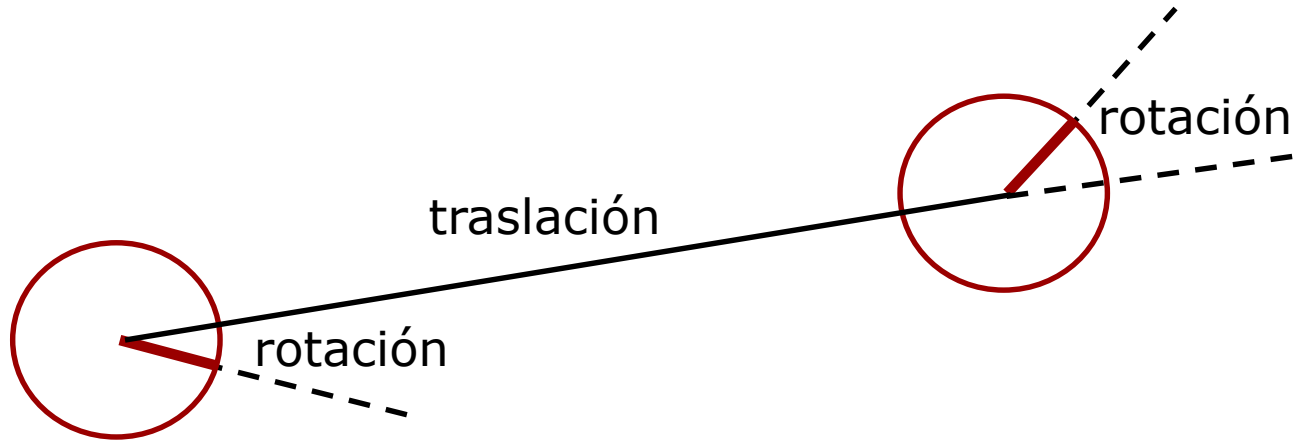


Pose final



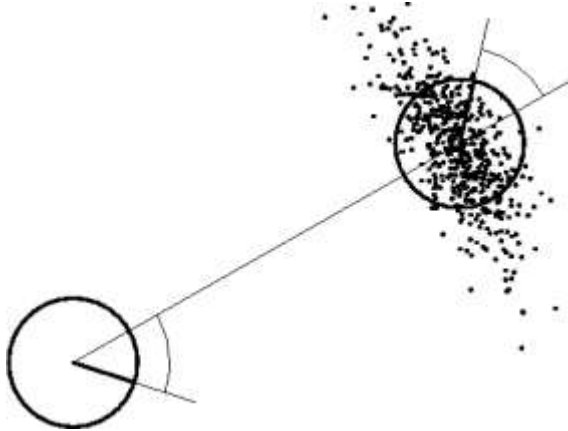
Según el movimiento estimado

# Repaso de modelos de movimiento



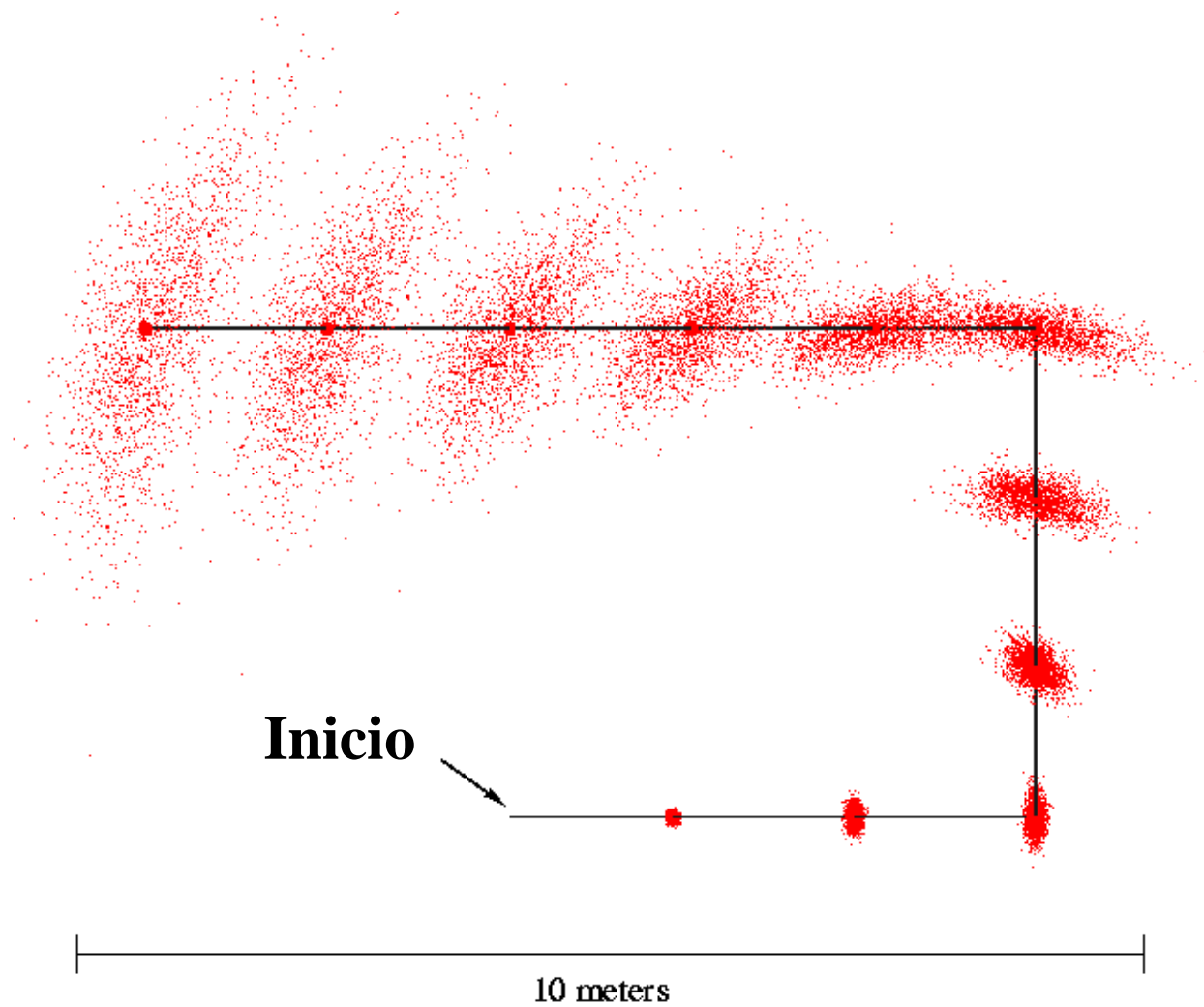
- Descomposición del movimiento en
  - Distancia recorrida
  - Rotación inicial
  - Rotación final

# Repaso de modelos de movimiento

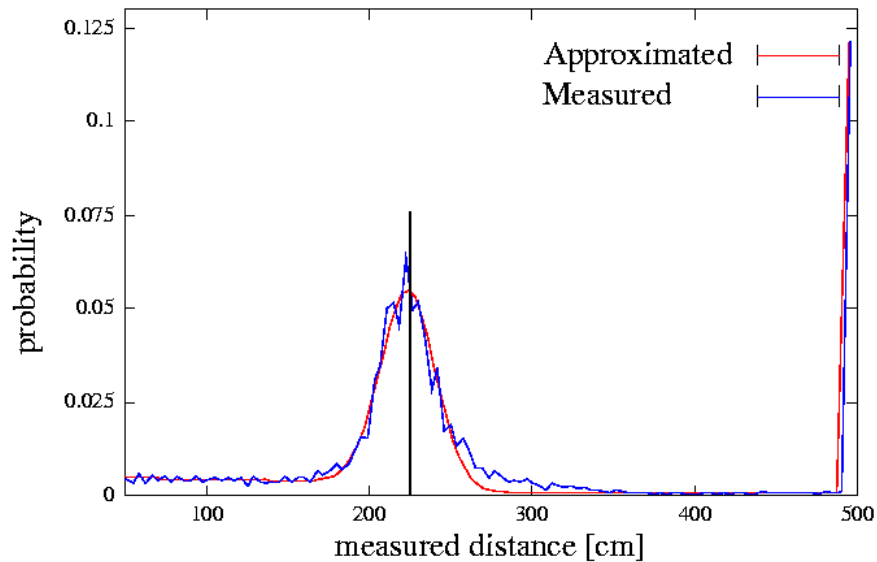


- Incerteza en el traslado del robot:  
Gaussiana sobre la distancia recorrida
- Incerteza en la rotación del robot:  
Gaussianas sobre la rotación inicial y final
- Para cada partícula, tomar una nueva pose muestreando de estas tres distribuciones normales individuales

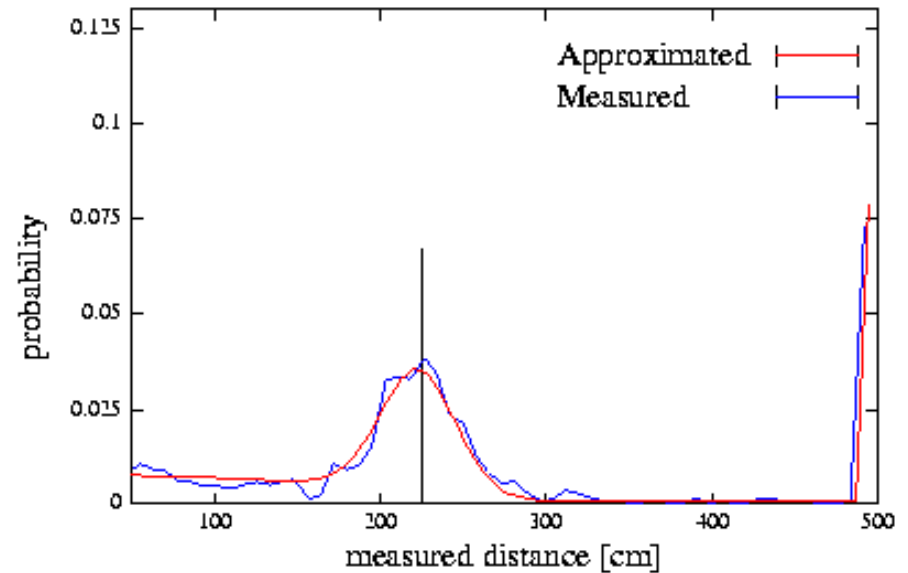
# Repaso de modelos de movimiento



# Repaso de modelo de sensor de proximidad



**Sensor Laser**



**Sensor Sonar**

# Localización de robots móviles con Filtros de Partículas (1)

- Cada partícula es una potencial pose del robot
- El conjunto de partículas pesadas aproxima la distribución a posteriori de la pose del robot (distribución objetivo)

# Localización de robots móviles con Filtros de Partículas (2)

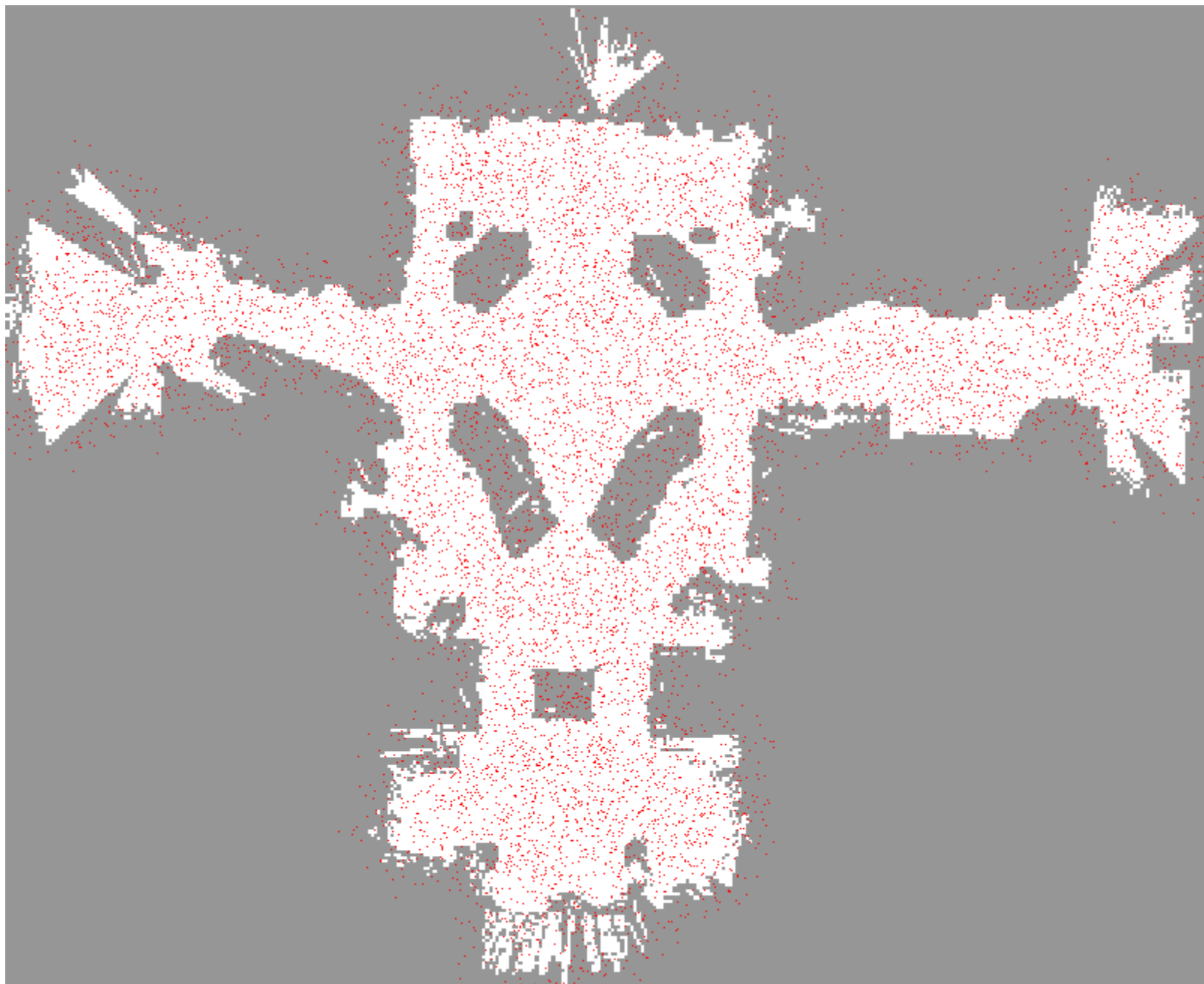
- Las partículas se toman del modelo de movimiento (distribución propuesta)
- Las partículas se pesan según el modelo de medición (modelo del sensor)
- Las partículas se remuestrean según el peso de las partículas

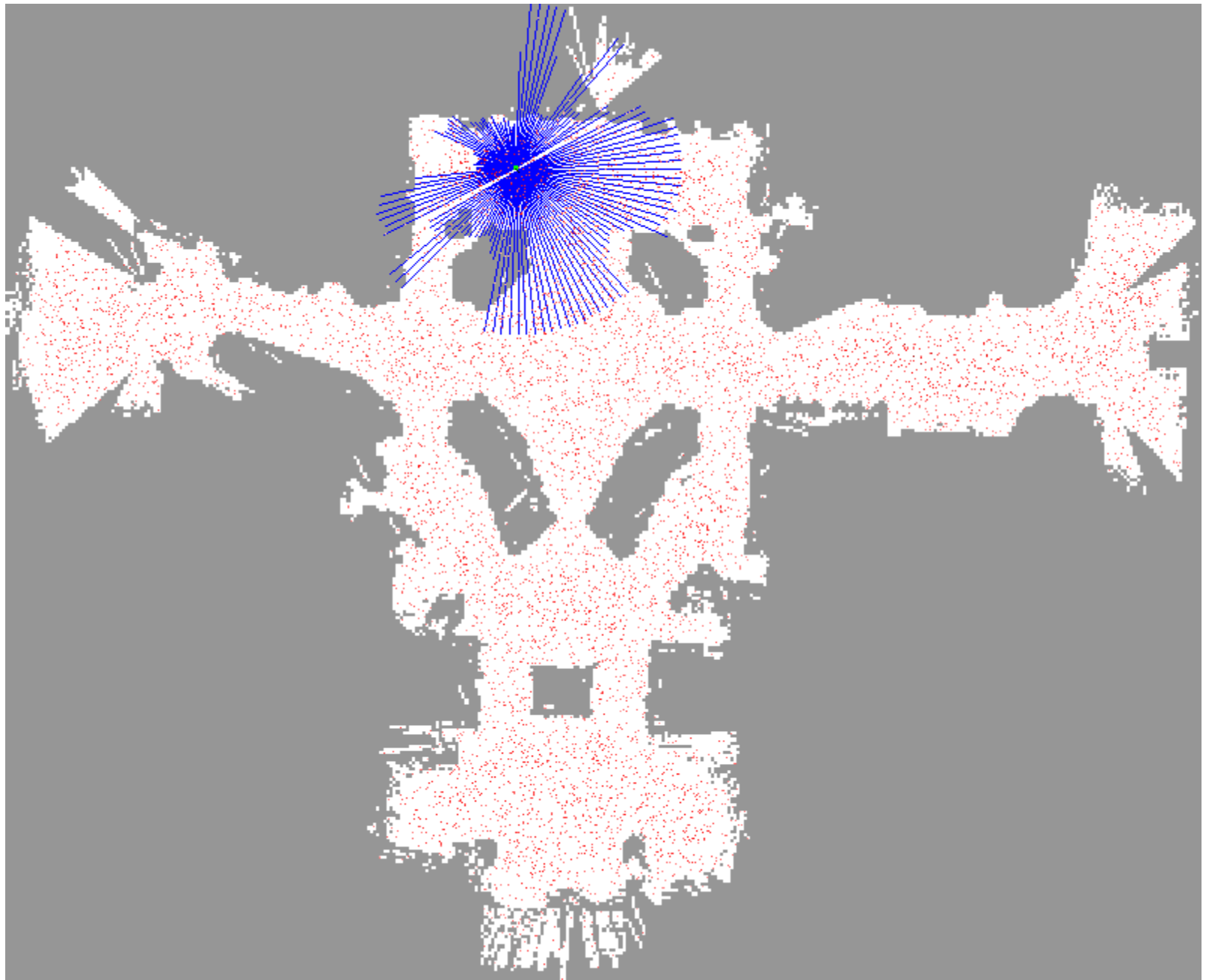


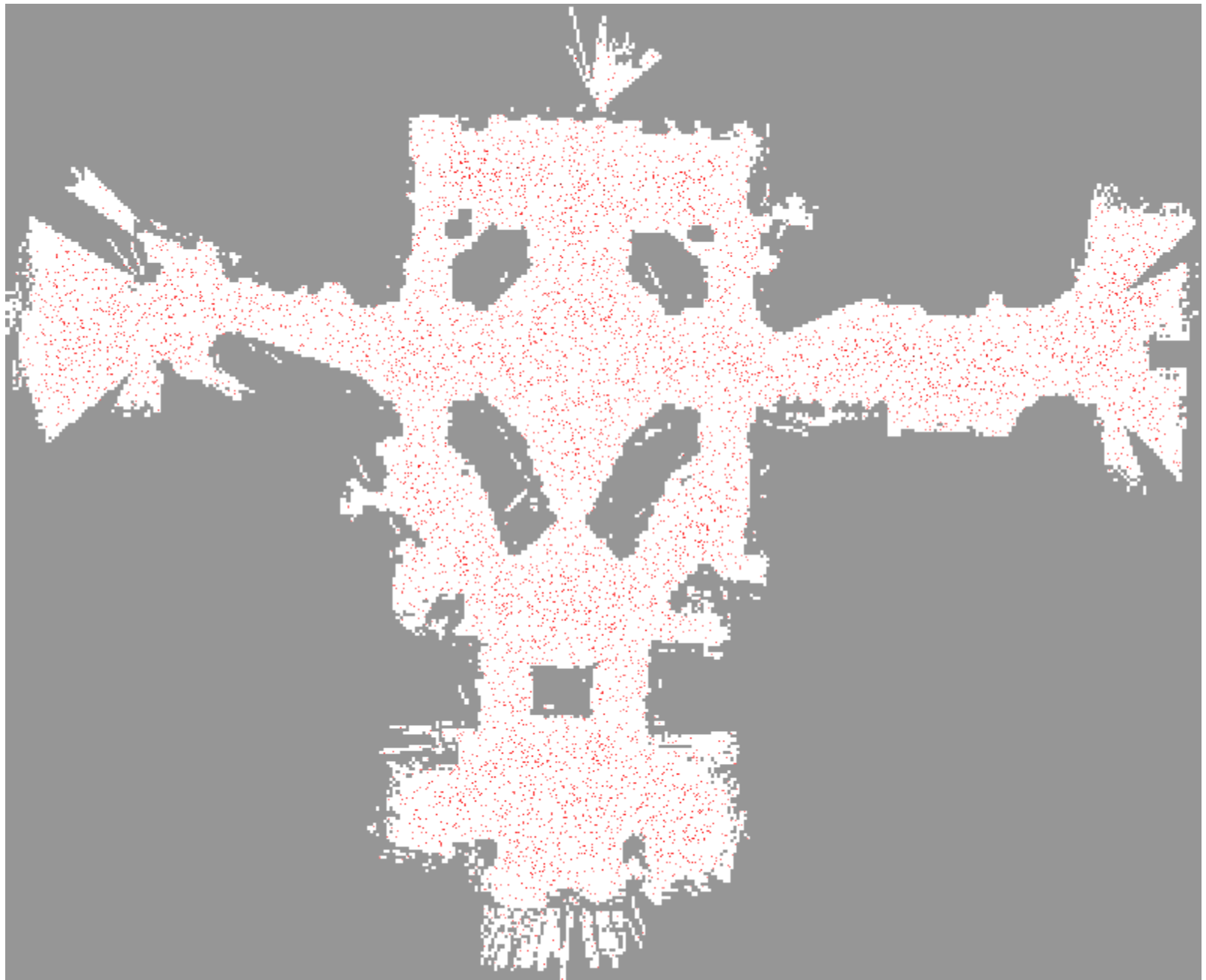
# Localización de robots móviles con Filtros de Partículas (3)

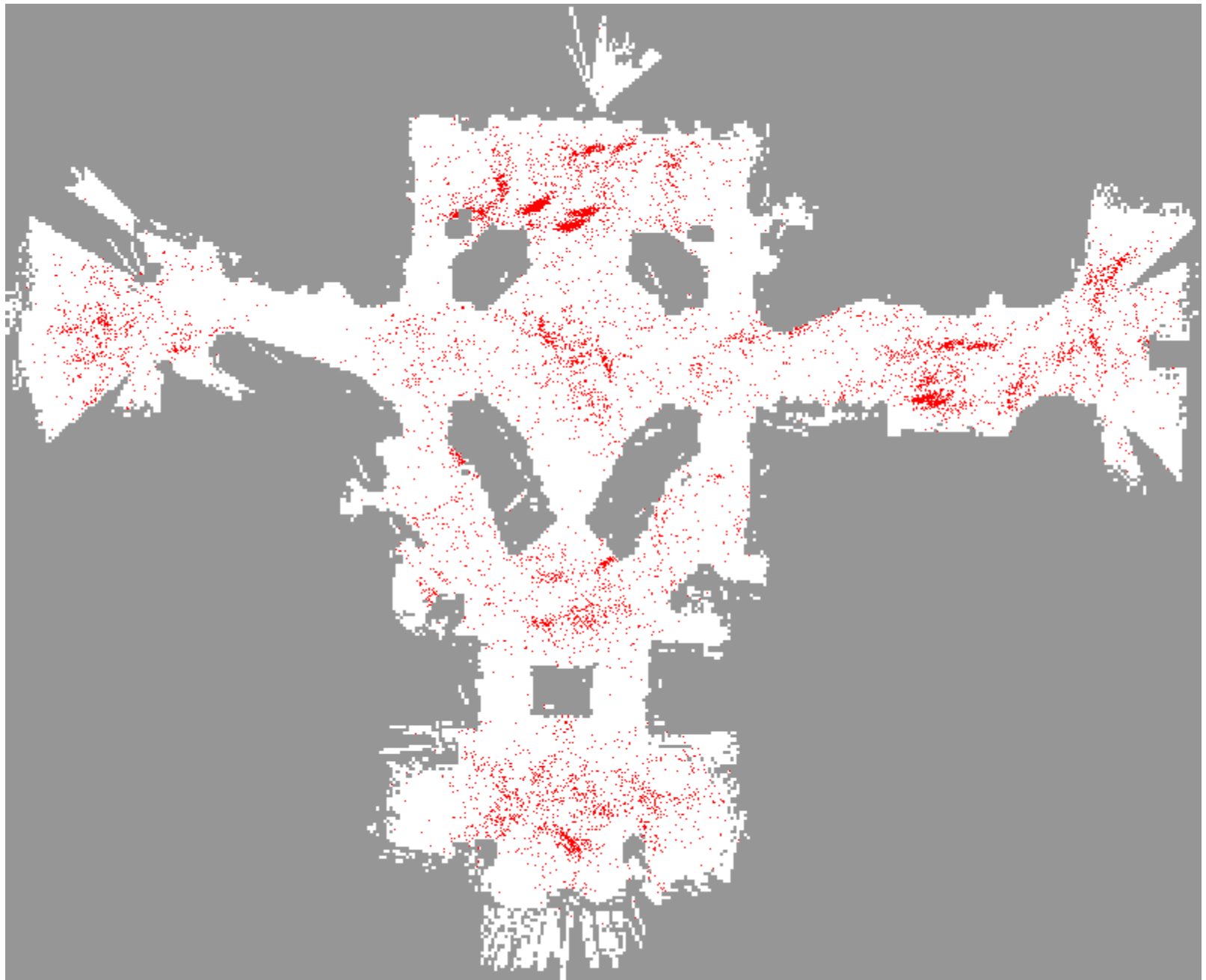
Por qué es necesario el remuestreo?

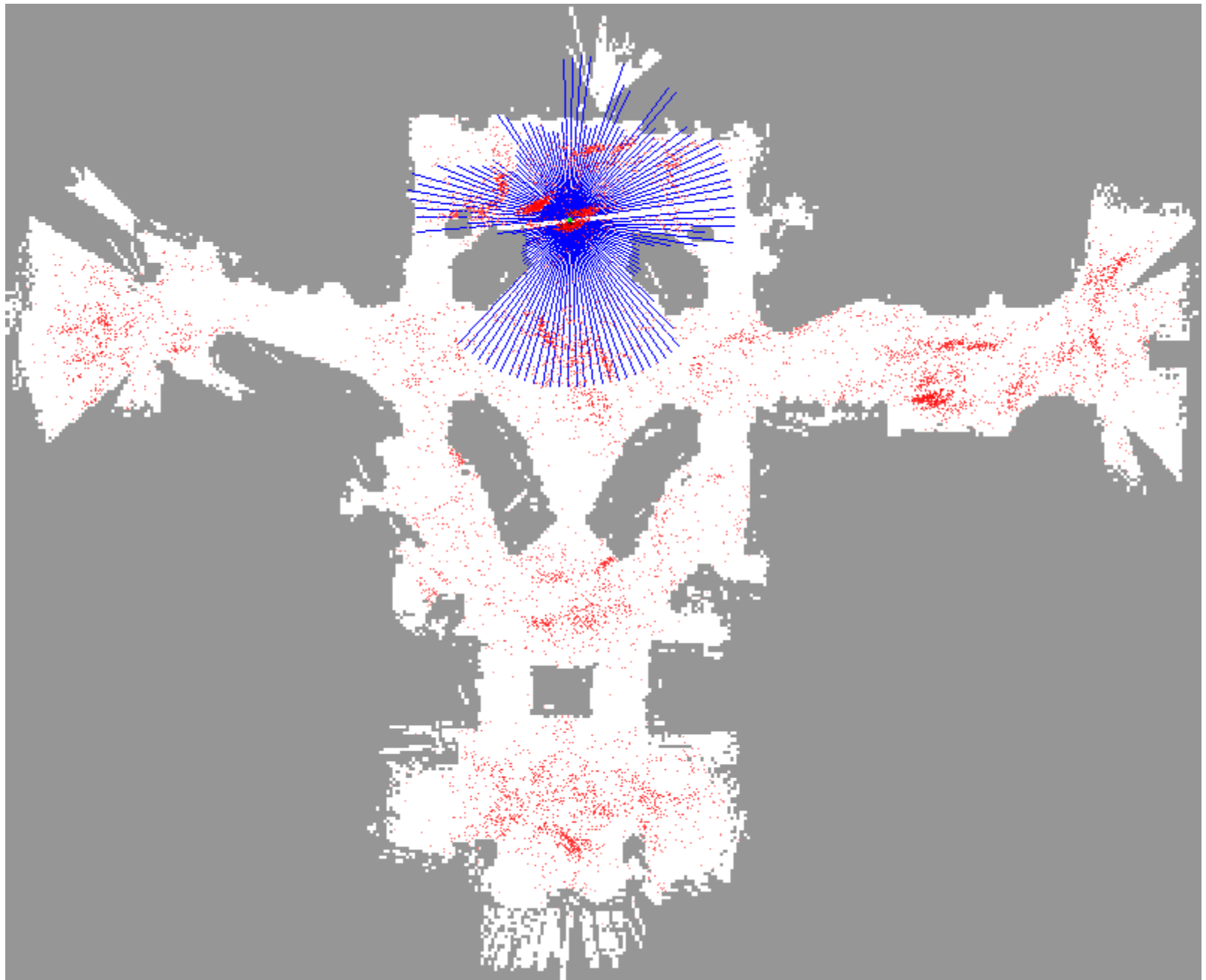
- Sólo hay un número finito de partículas
- Sin remuestreo, el filtro puede perder las “buenas hipótesis”
- El remuestreo asegura que las partículas quedan en áreas apropiadas del espacio de estados

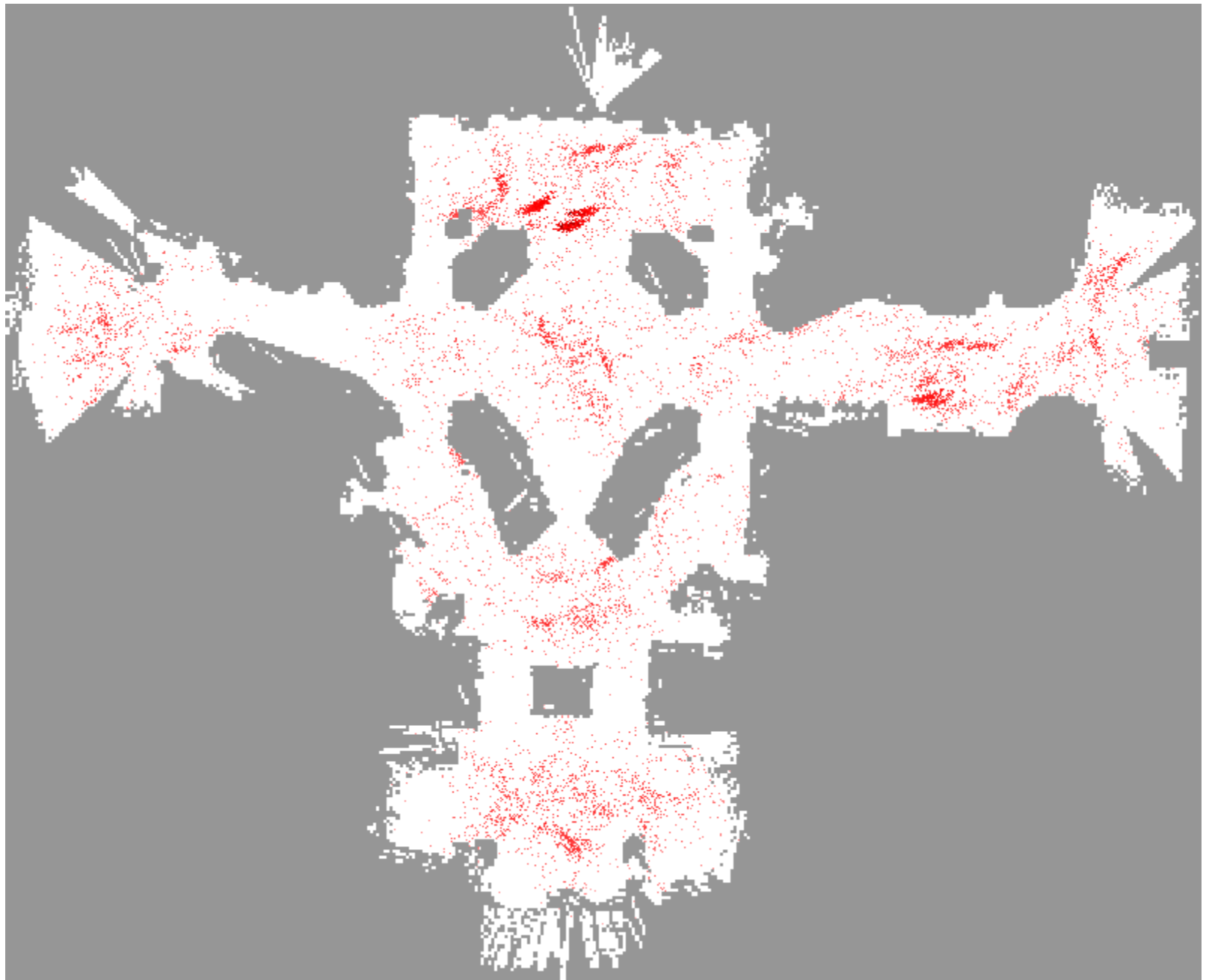


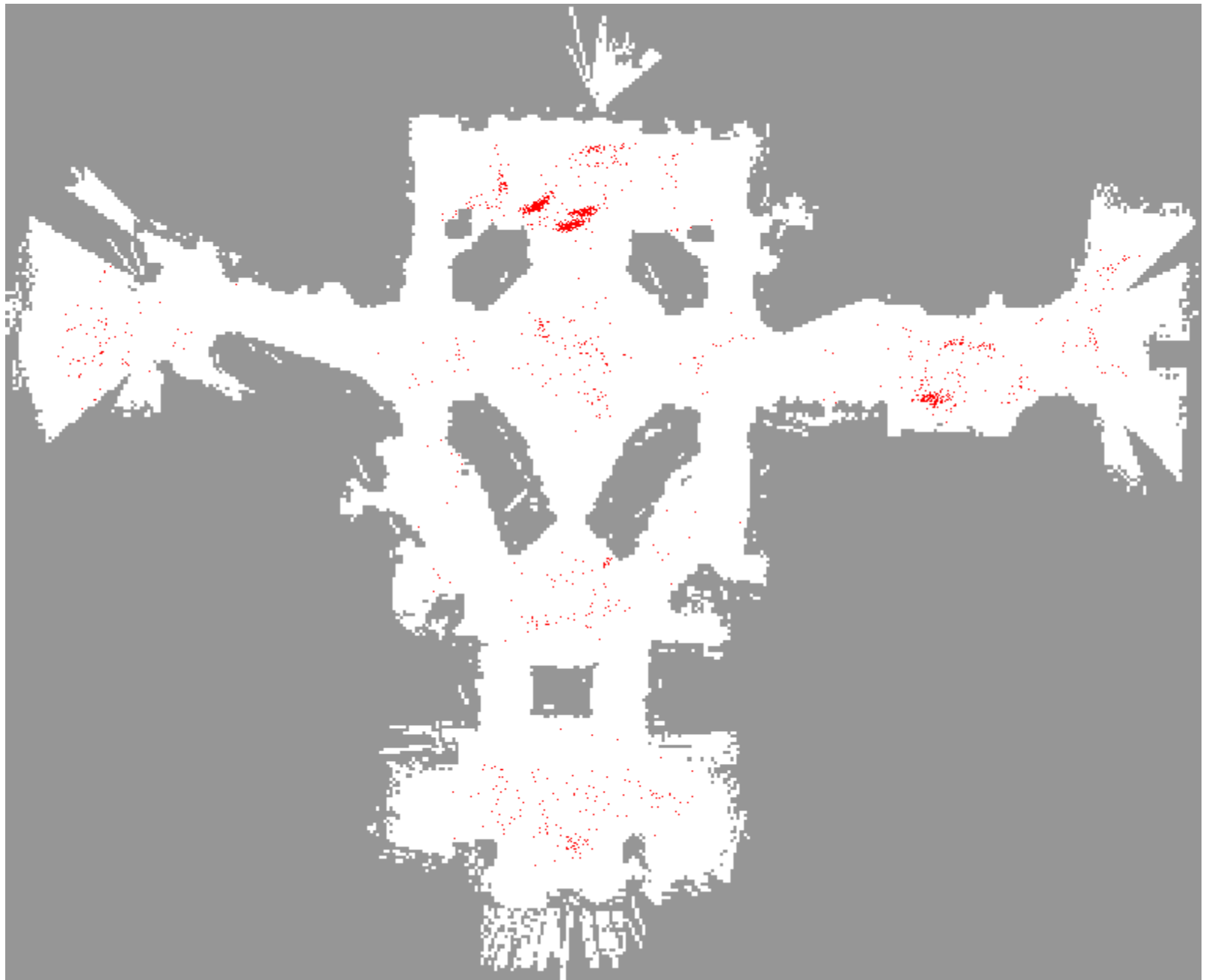






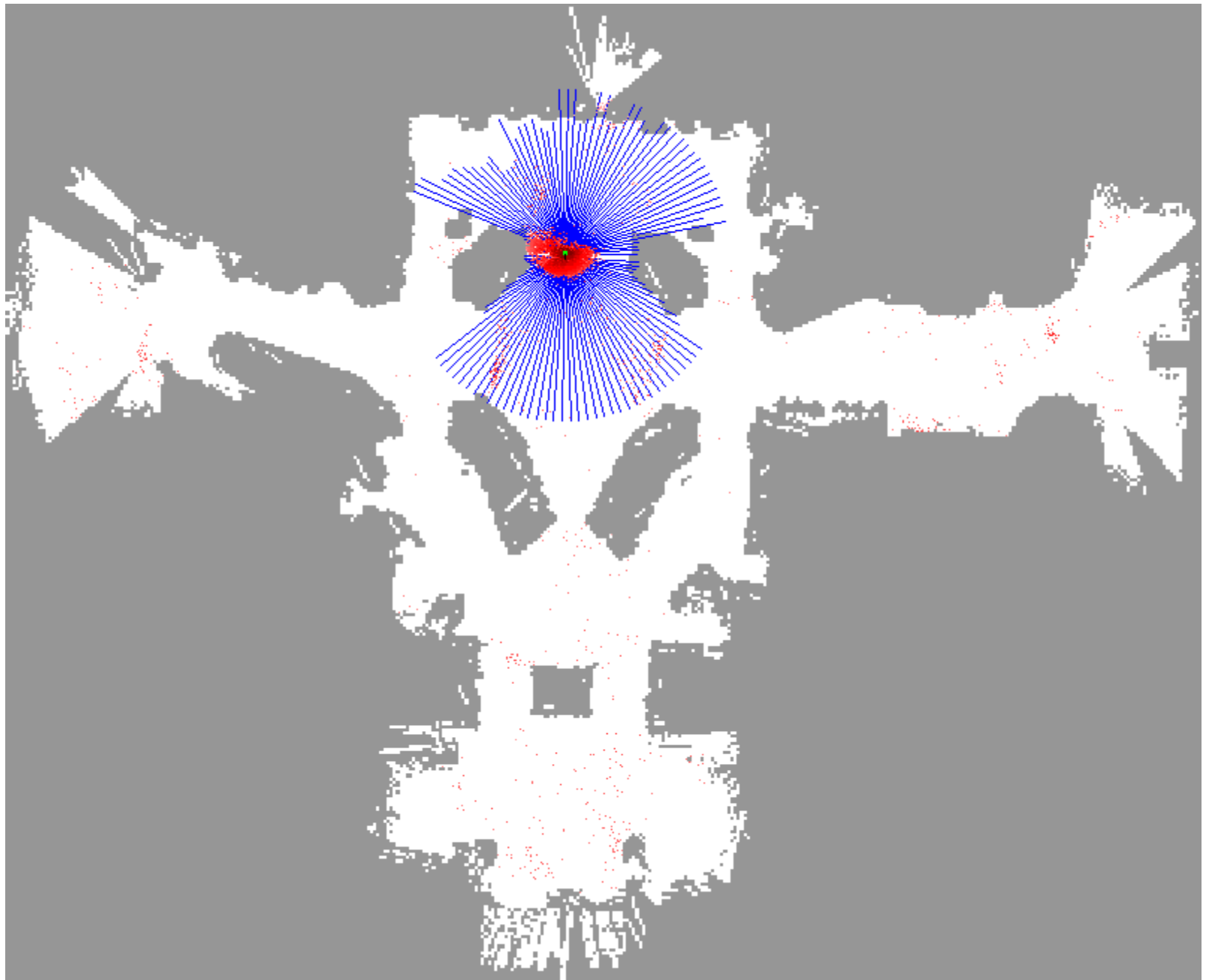




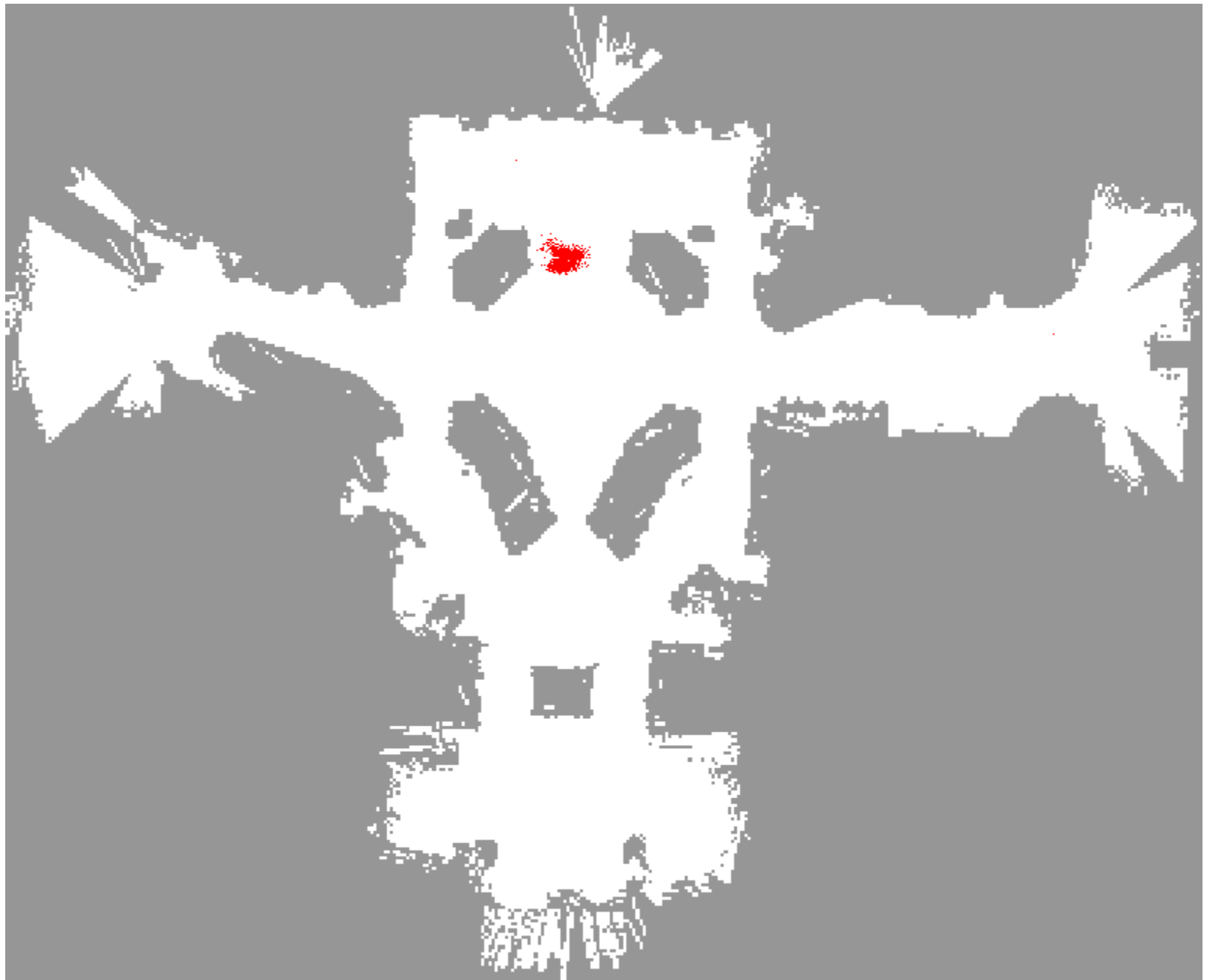


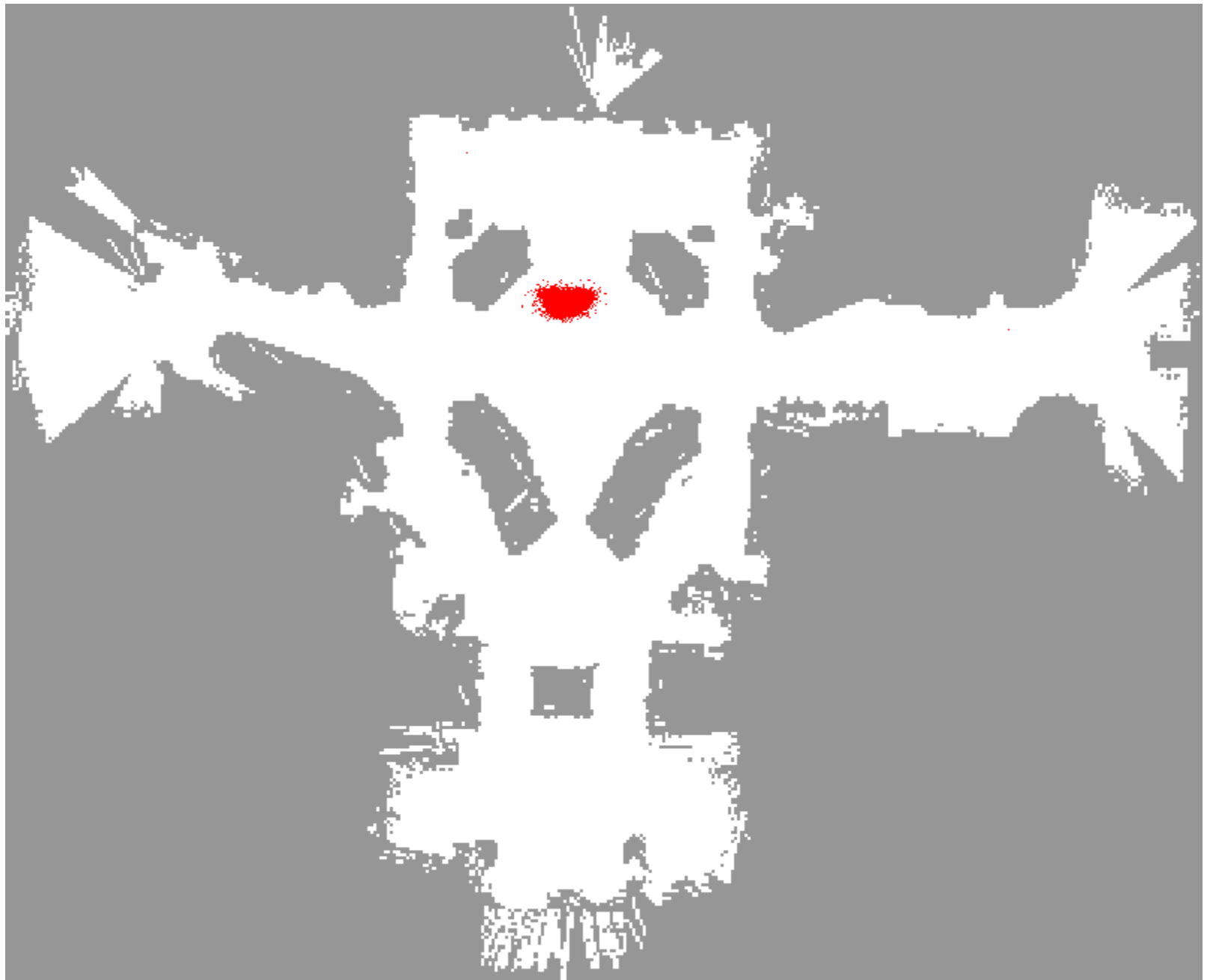


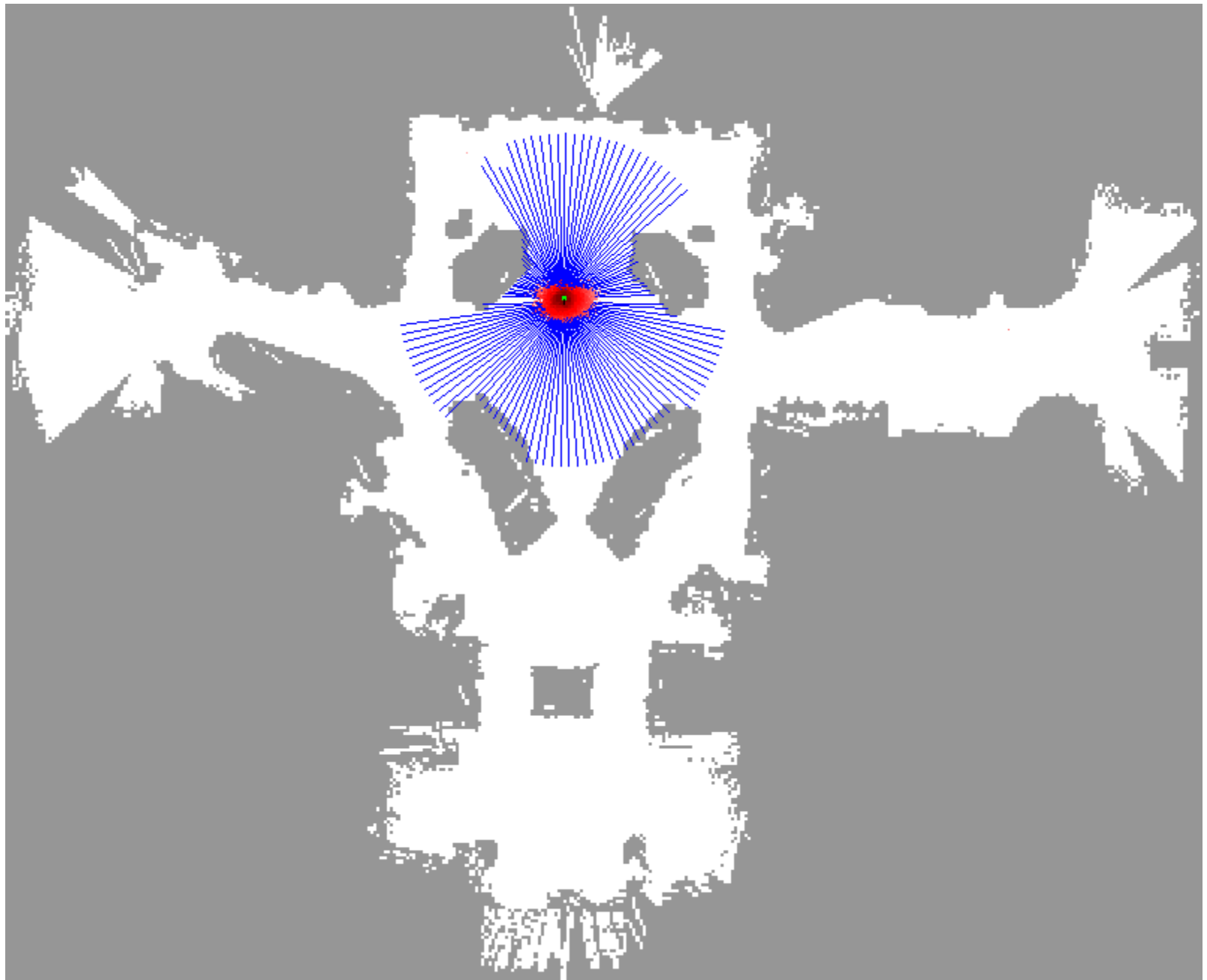


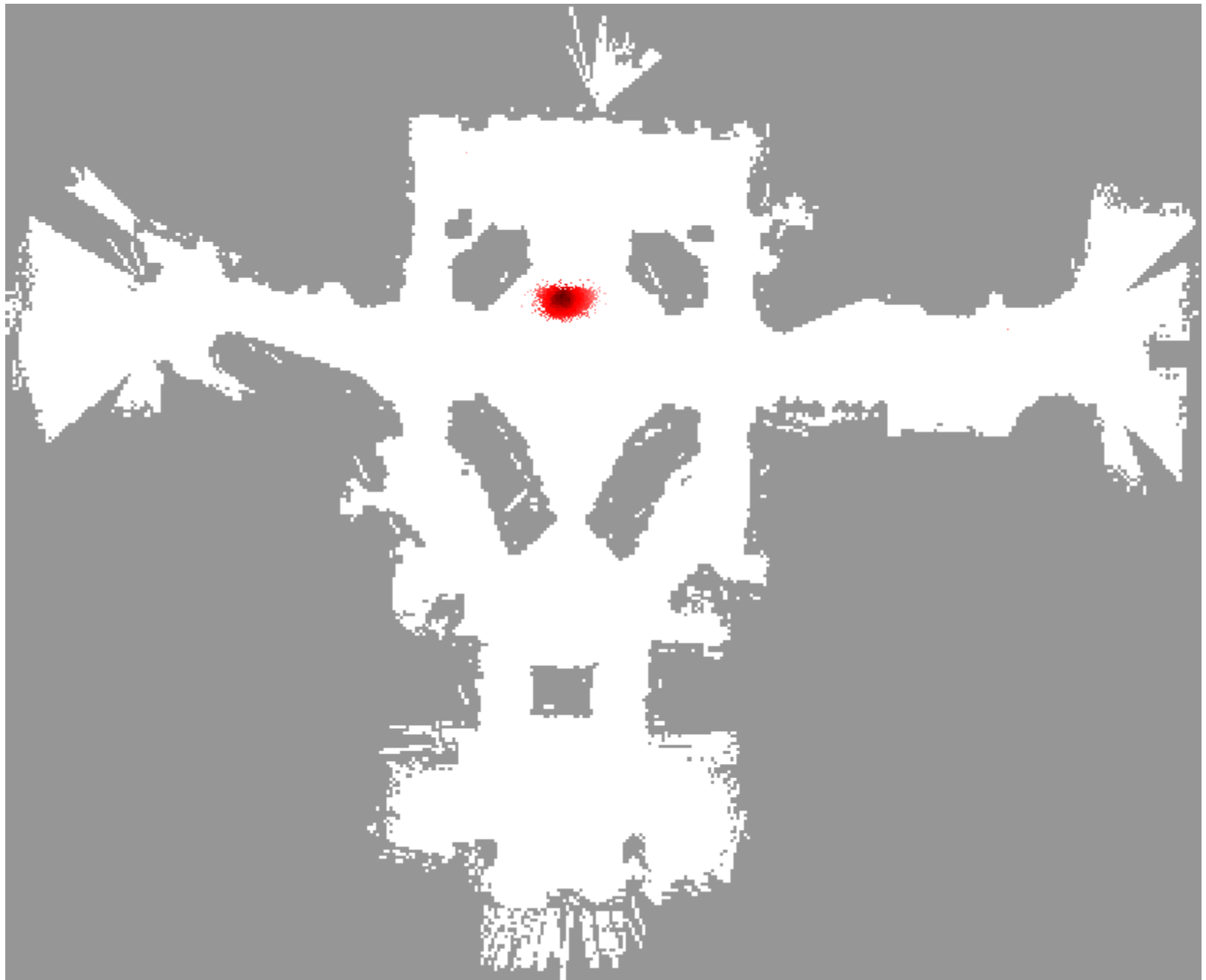


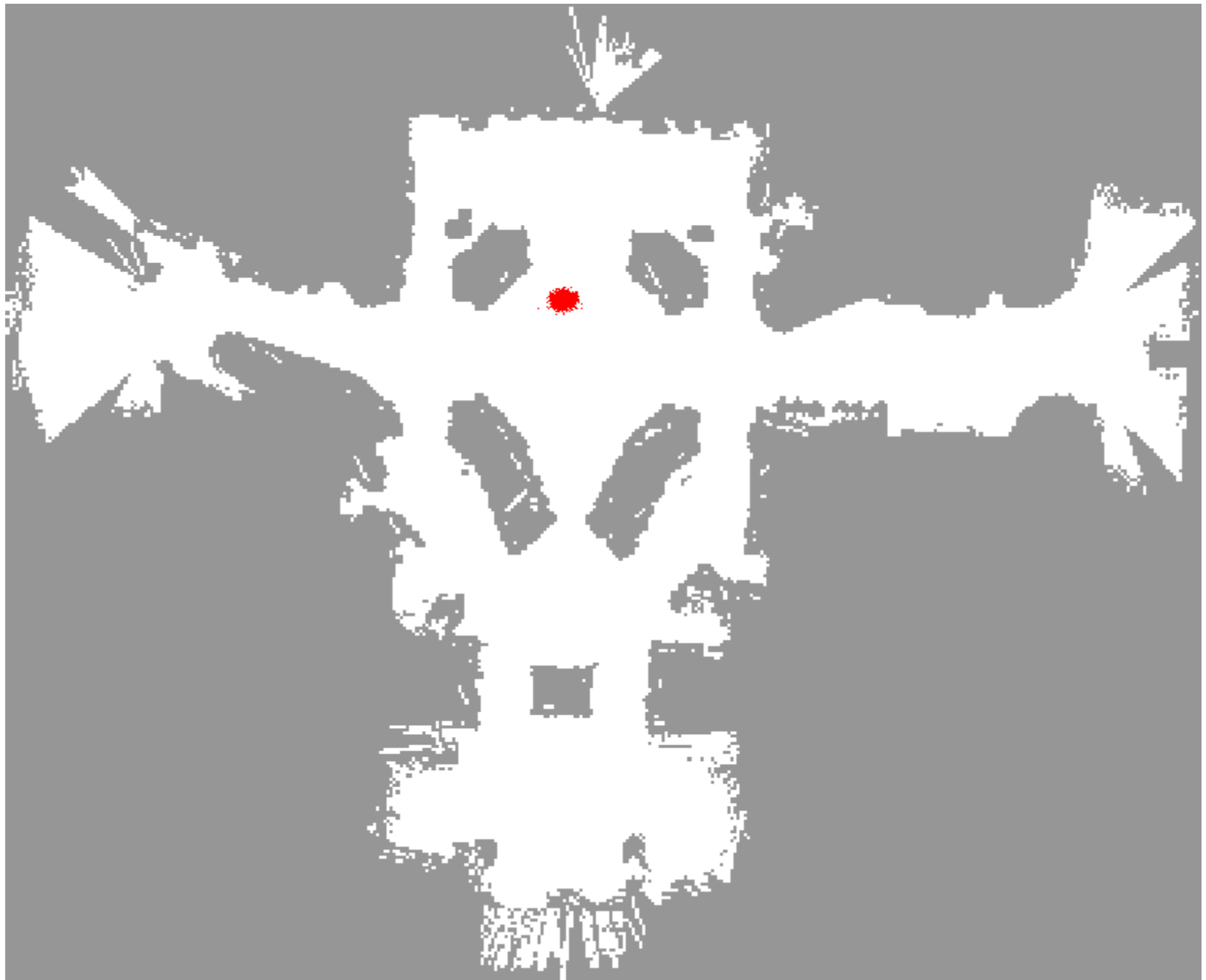




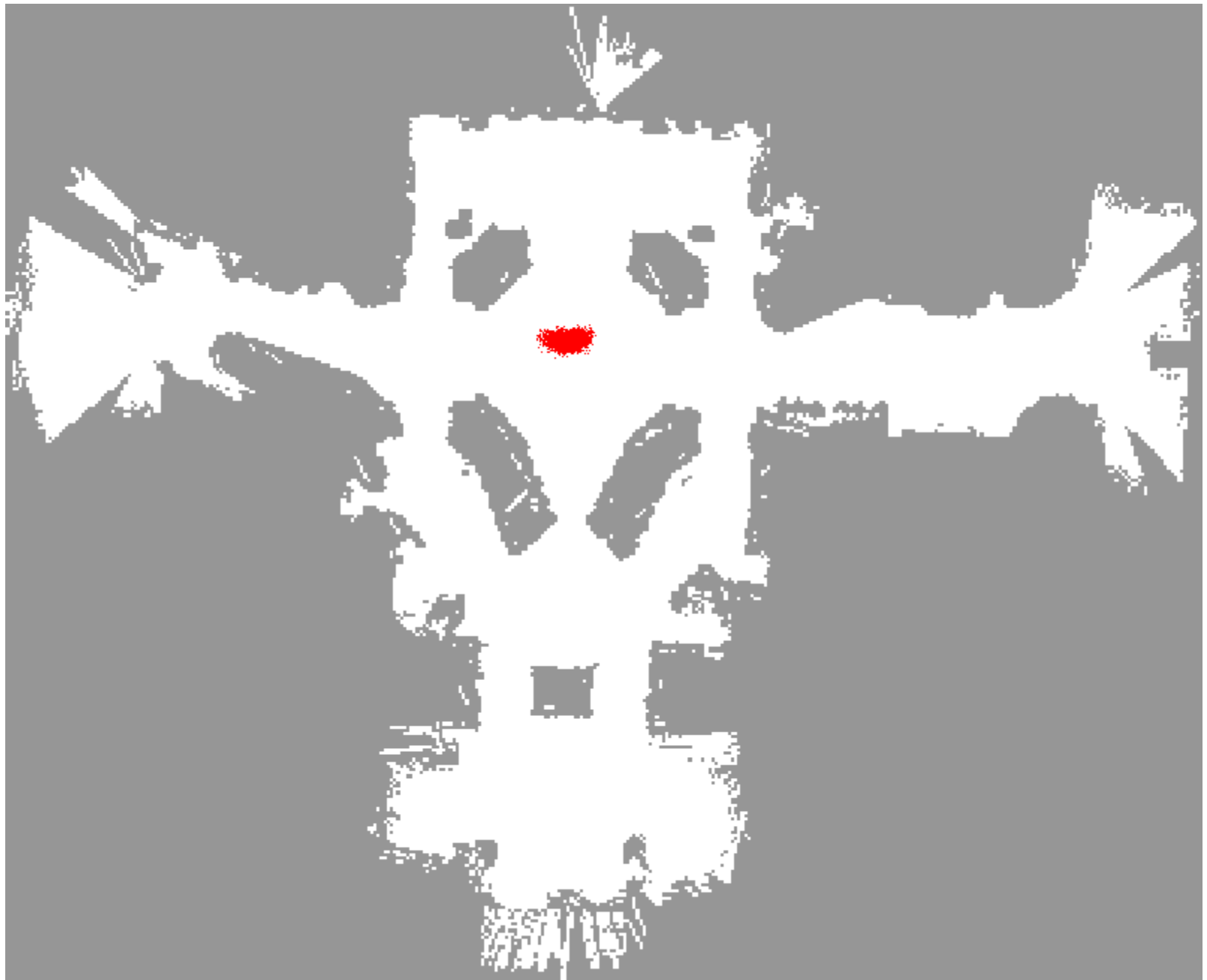


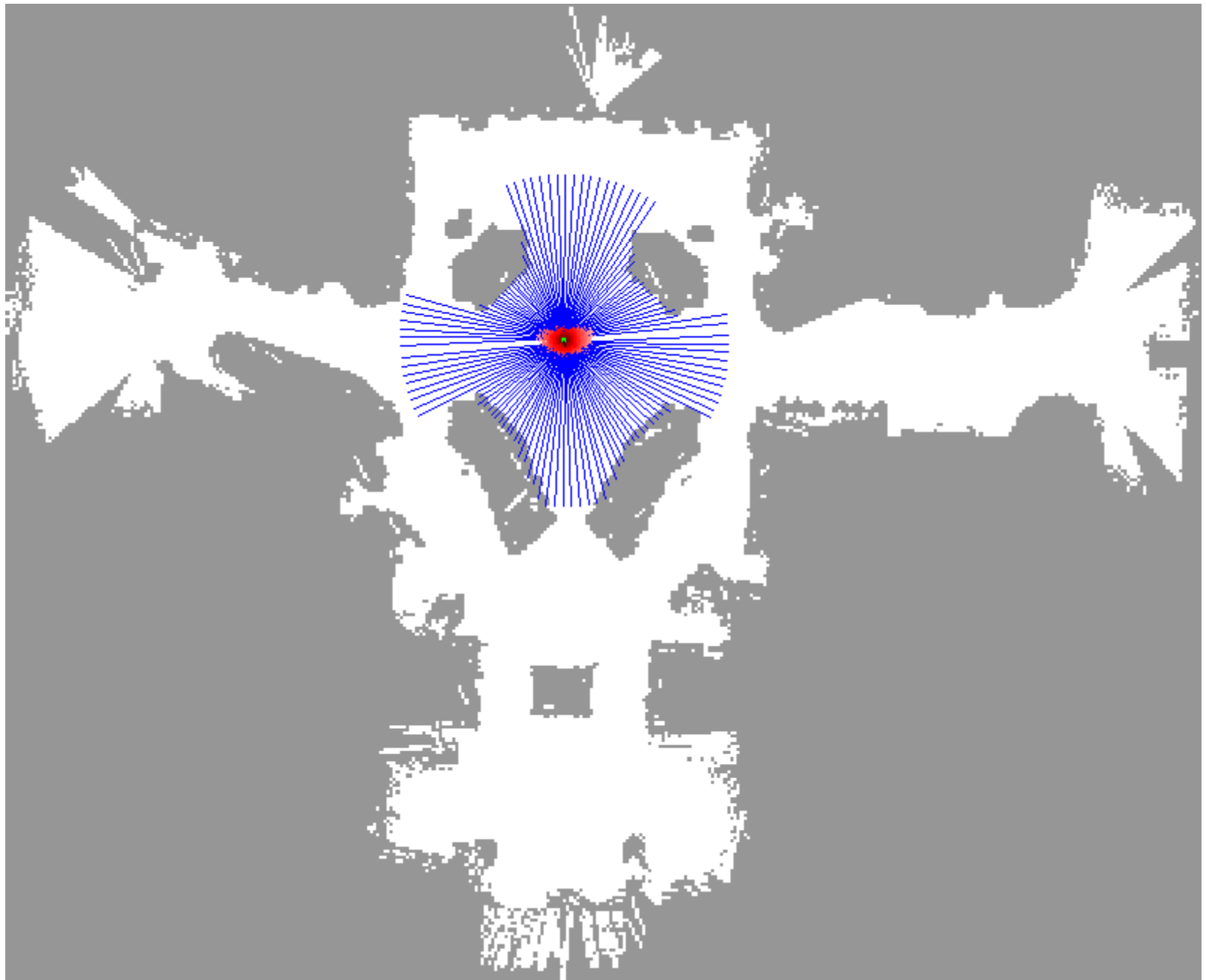




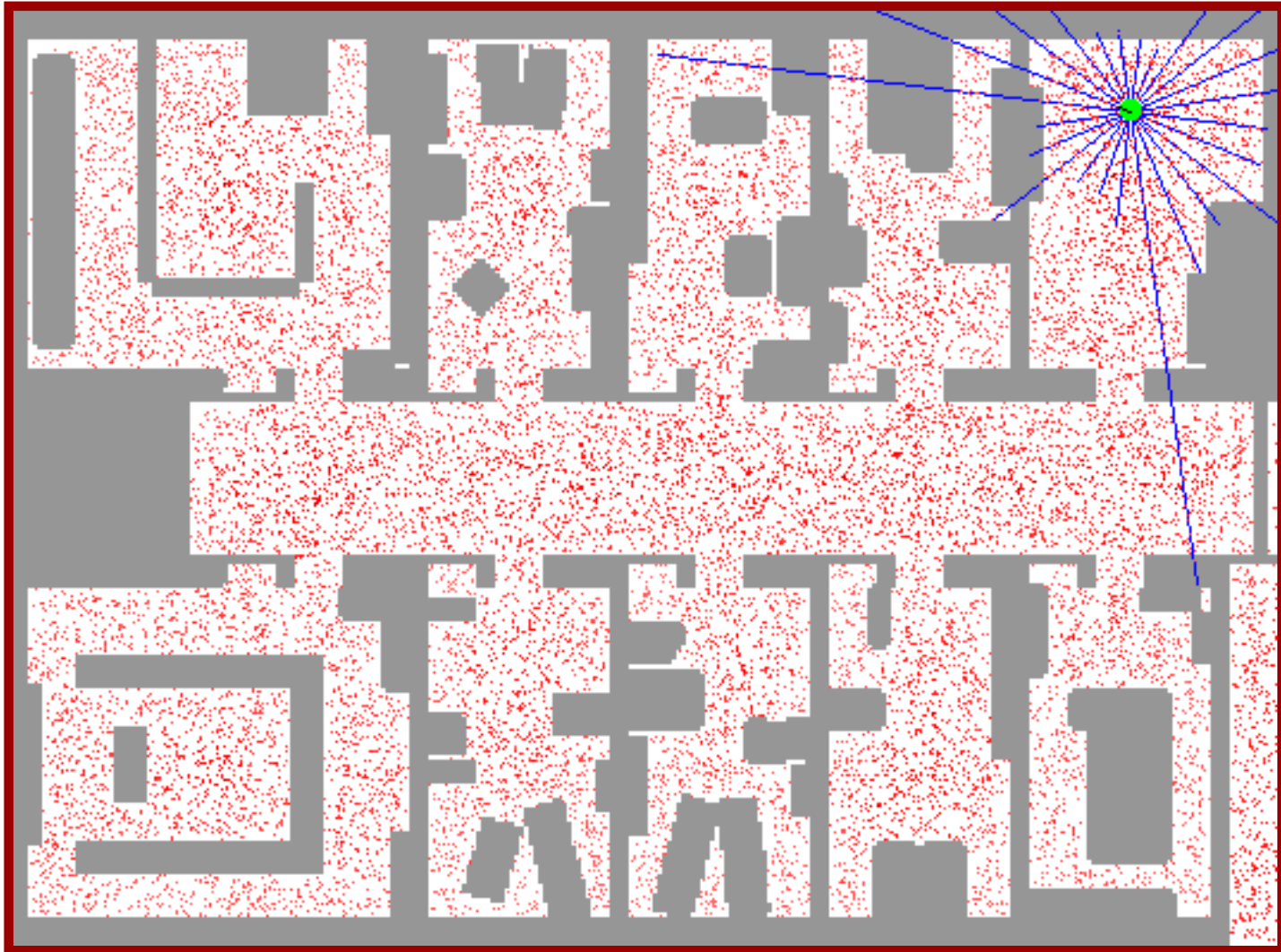




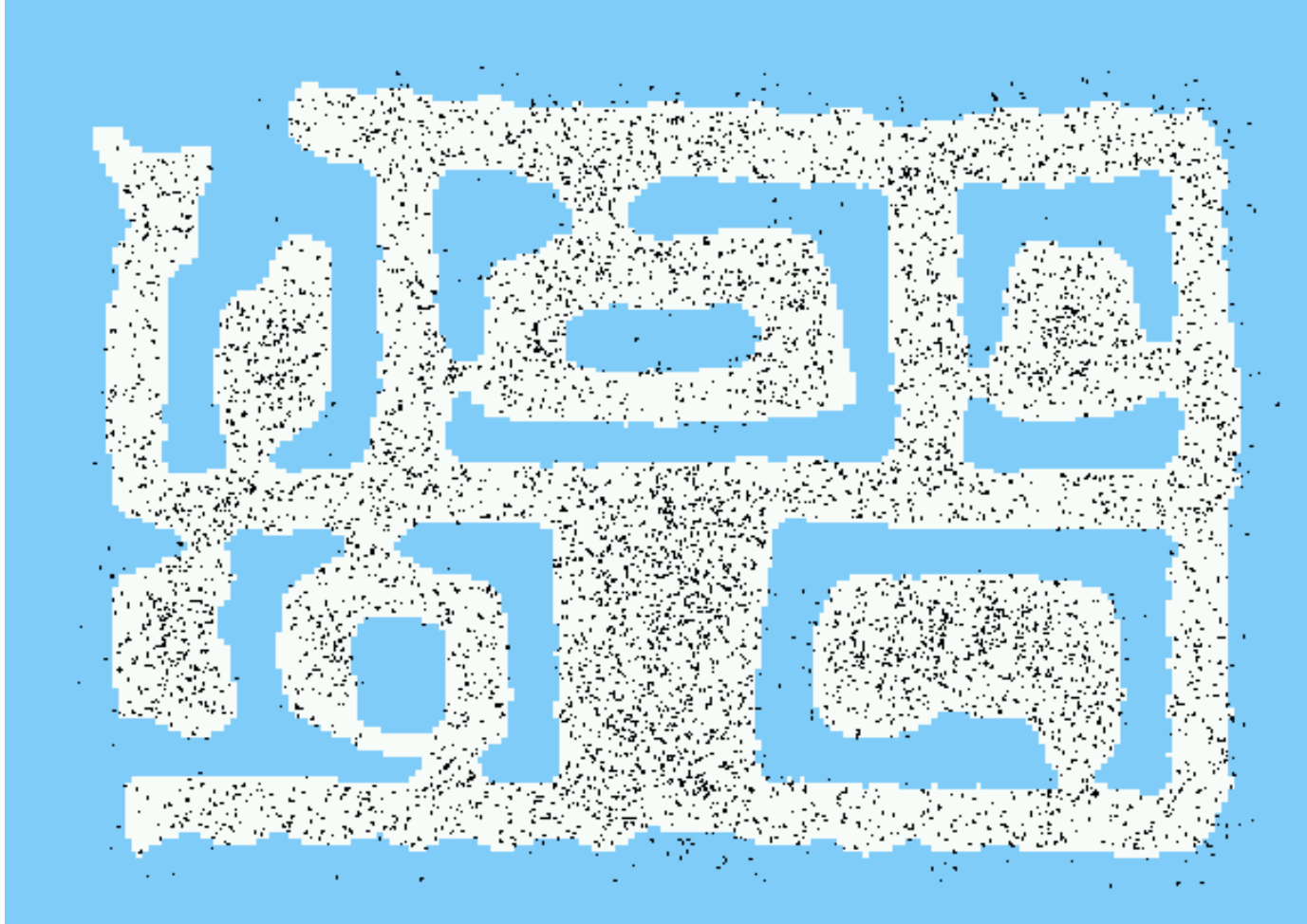




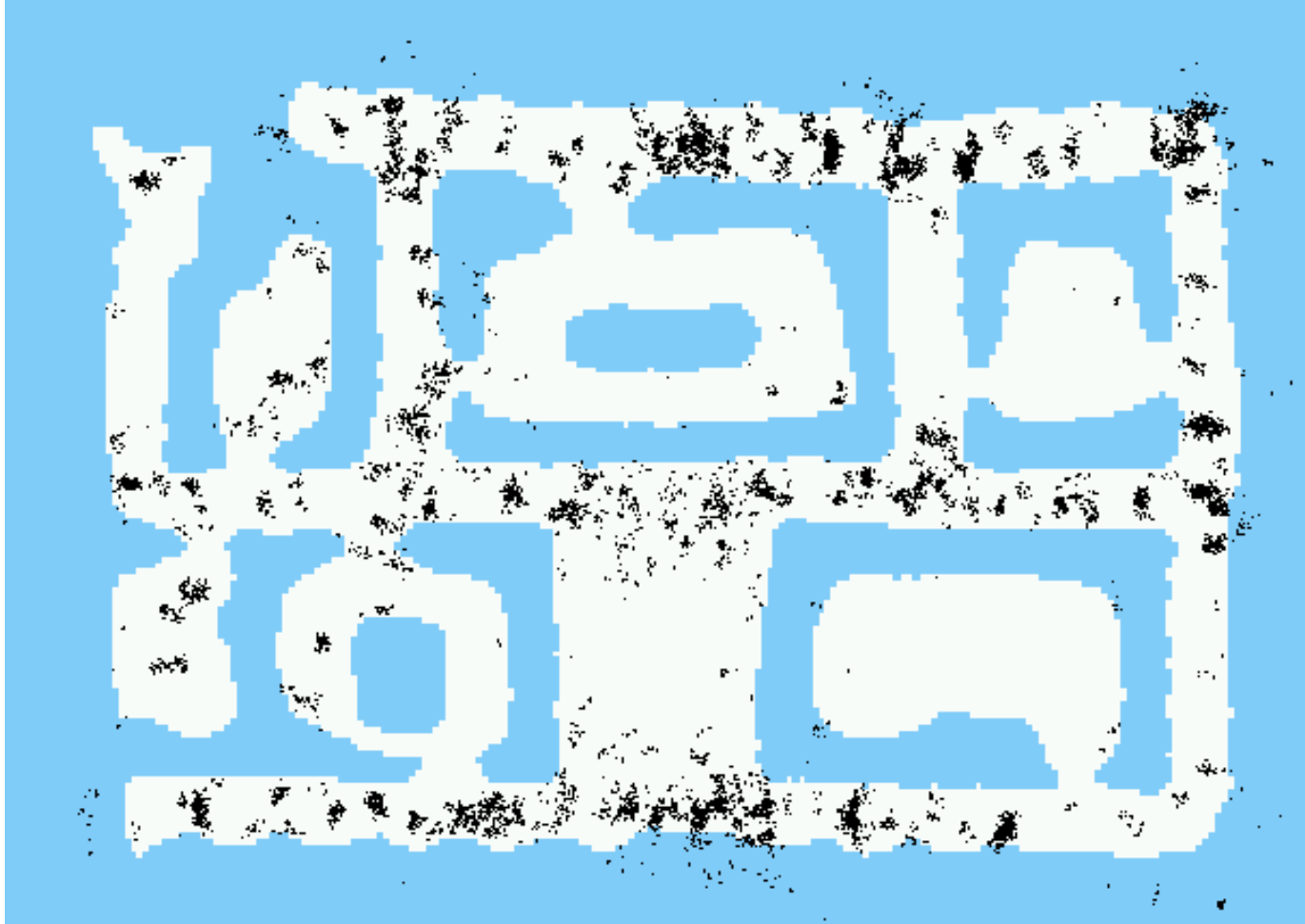
# Localización basada en muestras (sonar)



# Distribución inicial



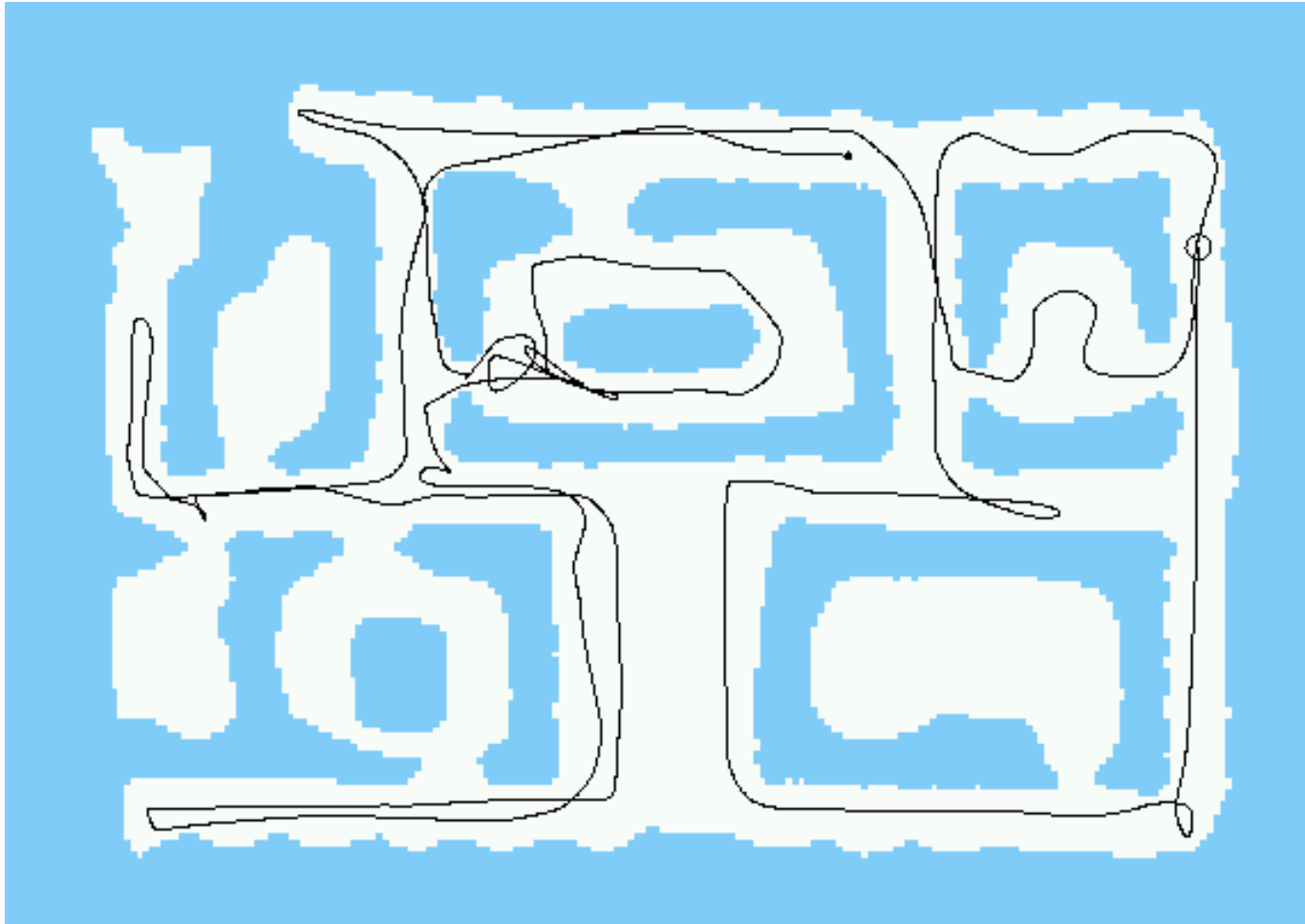
# Después de incorporar 10 mediciones de ultrasonido



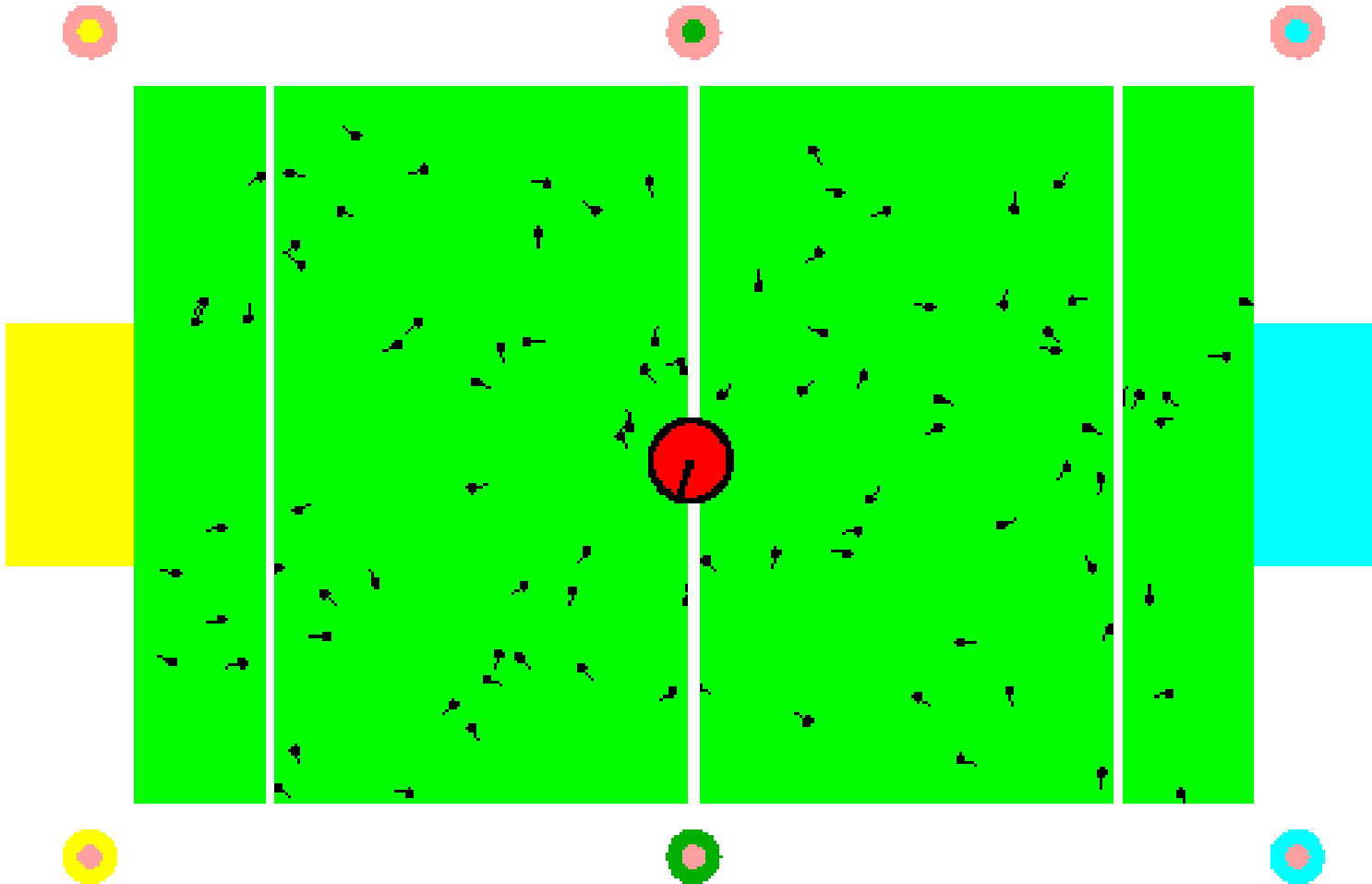
# Después de incorporar 65 mediciones de ultrasonido



# Trayectoria estimada

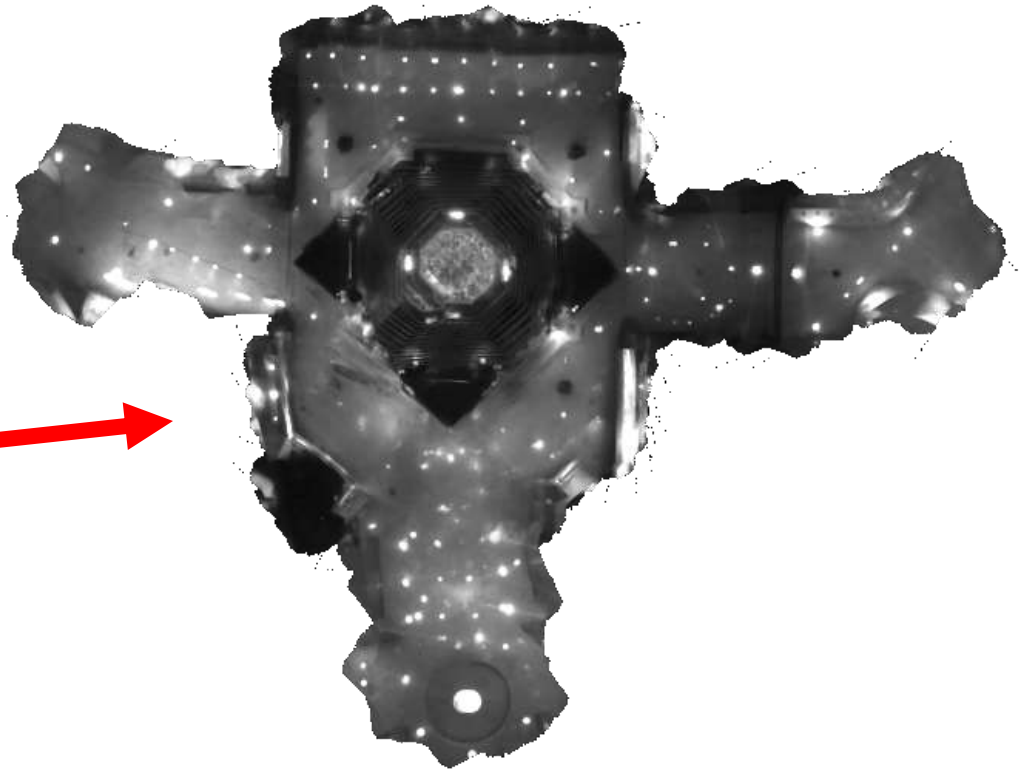


# Localización de robots AIBO (robocup)

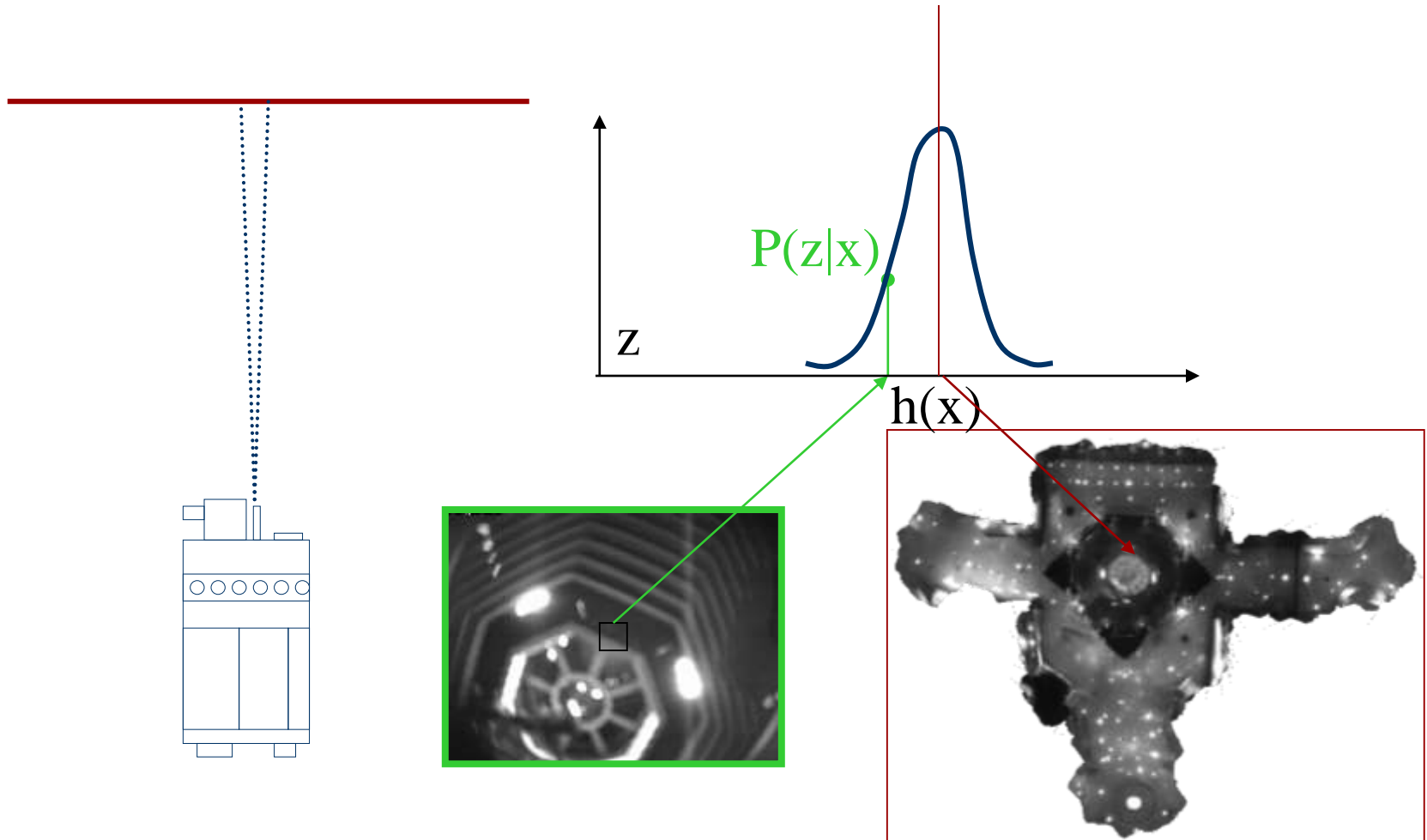




# Mapas del techo para Localización

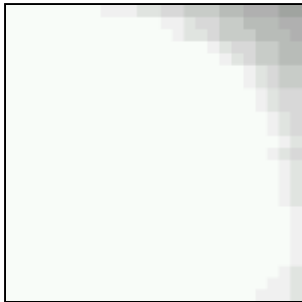


# Localización basada en visión

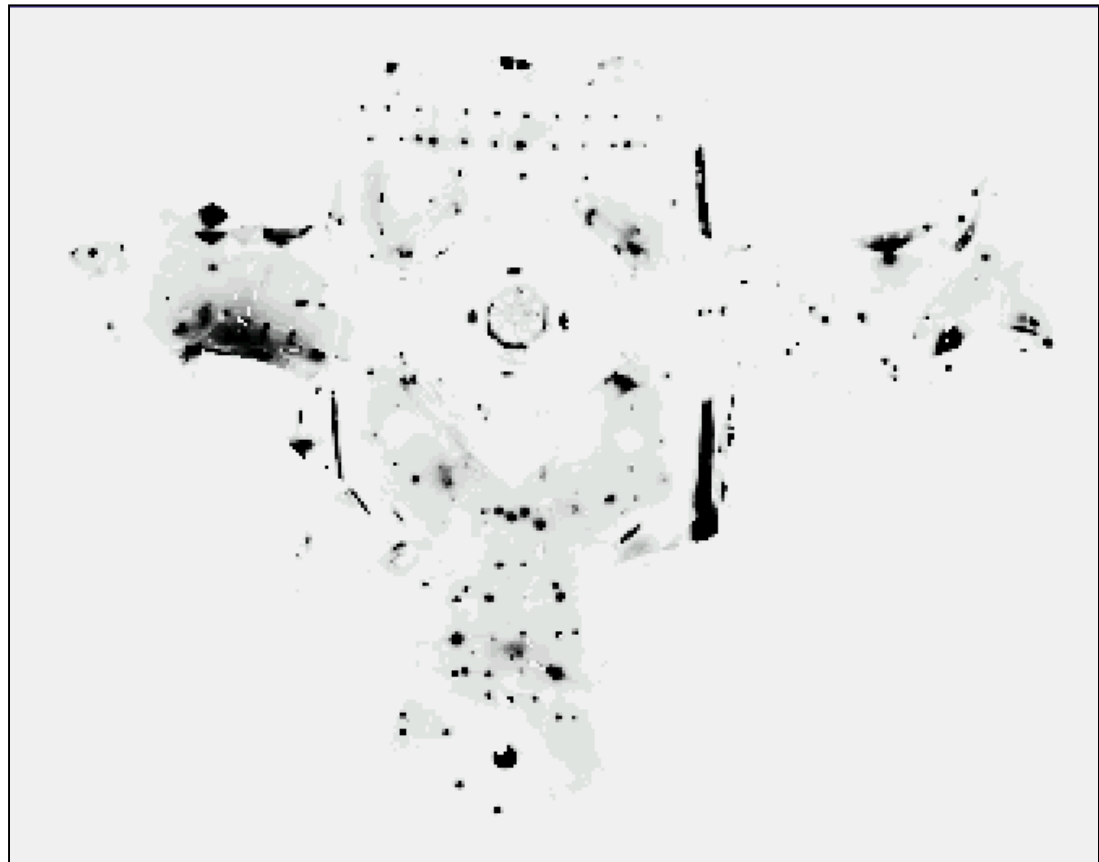


# Debajo de una luz

Medición  $z$ :

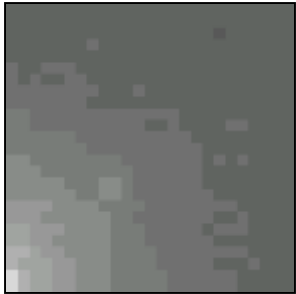


$P(z/x)$ :

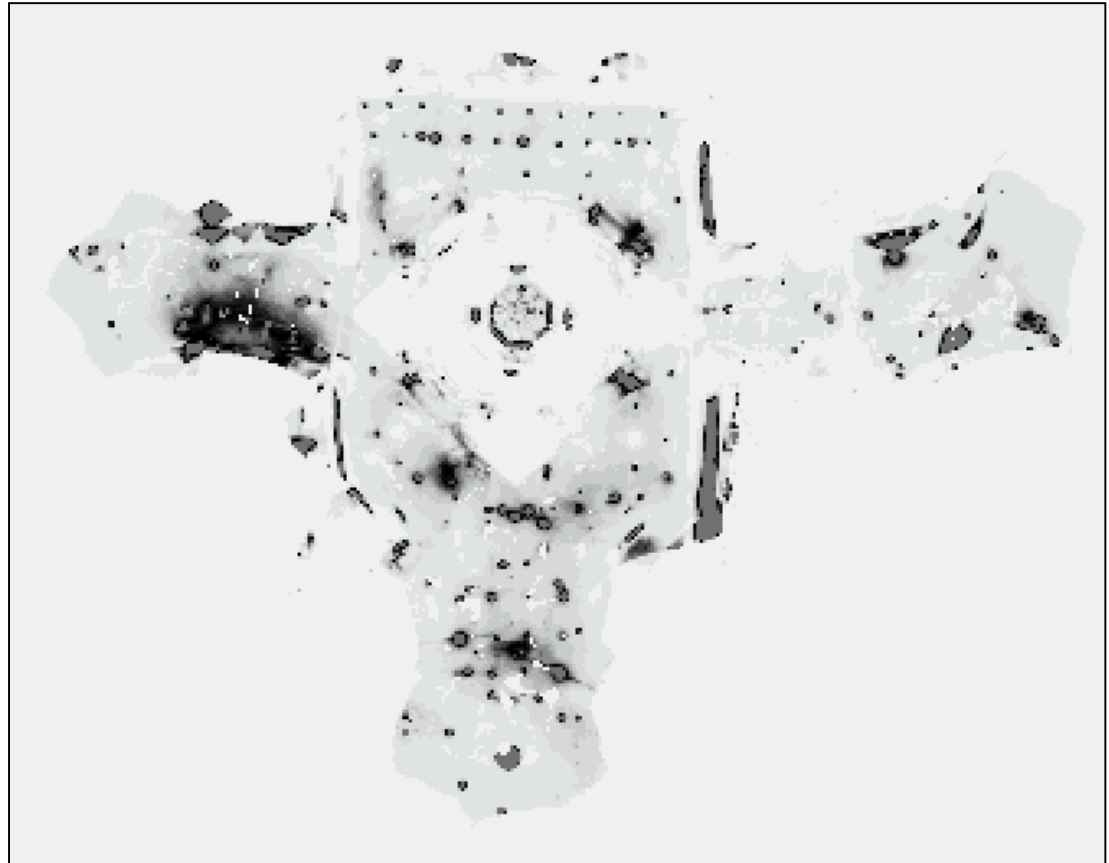


# Próximo a una luz

Medición  $z$ :



$P(z/x)$ :

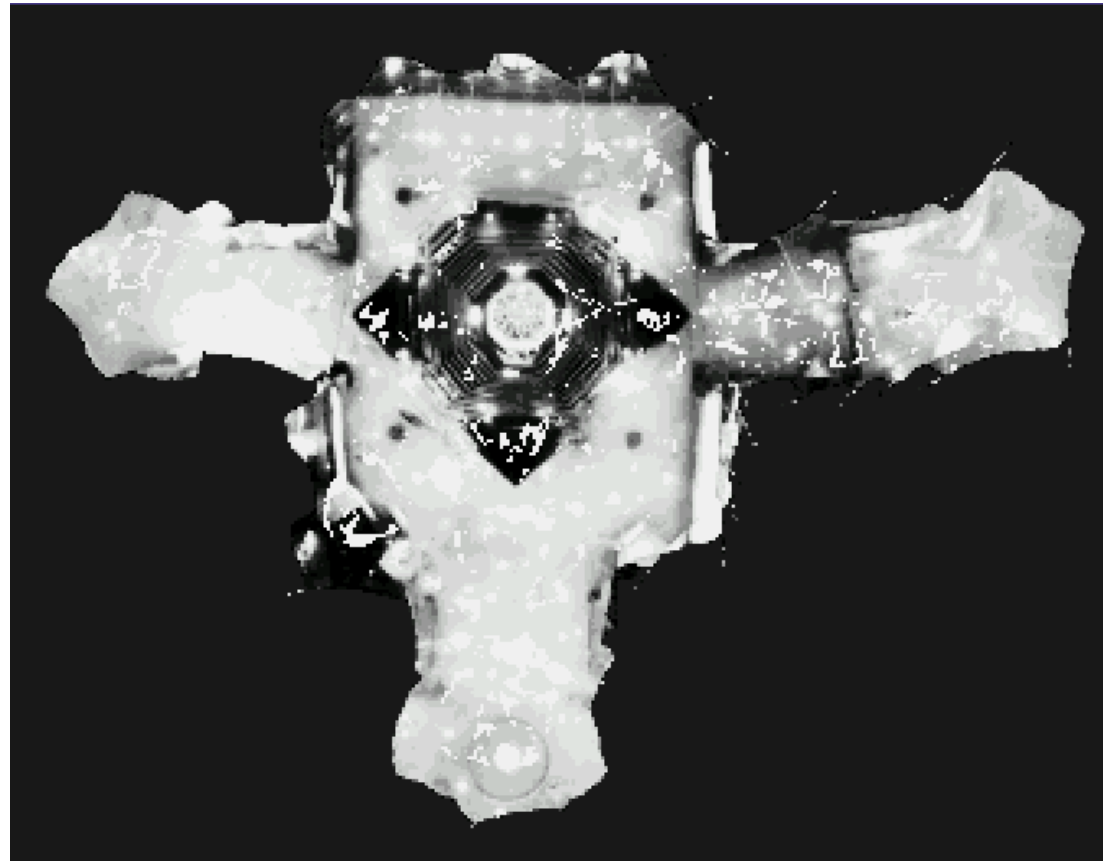


# En lugares sin luz

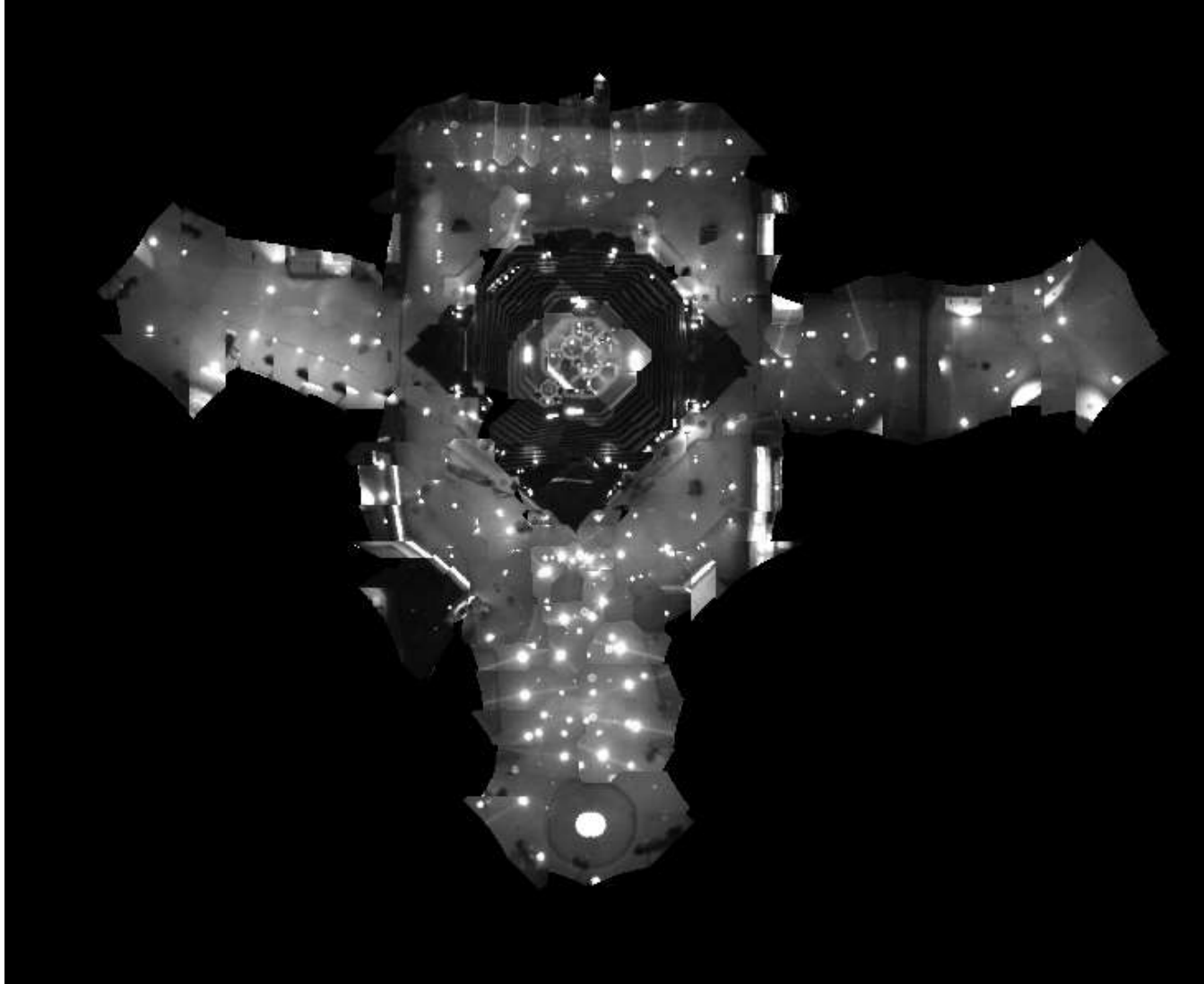
Medición  $z$ :



$P(z/x)$ :



# Localización global usando visión



# Limitaciones

- El método propuesto puede
  - Seguir la pose de un robot móvil y
  - Localizar globalmente el robot
- Cómo podemos lidiar con errores de localización (por ejemplo, el problema del robot secuestrado)?

# Enfoques

- Agregar **aleatoriamente** un número finito de muestras
- Esto asume que el robot puede ser teletransportado en cualquier momento
- Otra opción es insertar muestras aleatorias de **manera proporcional** al promedio del **likelihood** de las partículas



# Resumen – Filtro de Partículas

- Los filtros de partículas son una **implementación** de filtrado recursivo Bayesiano
- Representan la densidad de probabilidad posterior a través de un **conjunto de muestras pesadas**
- Pueden modelar distribuciones **no Gaussianas**
- Se muestrea la **distribución propuesta** y los pesos consideran las diferencias con la distribución objetivo

# Resumen – Localización de FP

- Las partículas se propagan según el **modelo de movimiento**.
- Se pesan según el likelihood de las **observaciones**.
- En el **remuestreo**, se toman nuevas partículas con una **probabilidad proporcional al likelihood** de las observaciones.