

## Capítulo 6

# Mapeado basado en el algoritmo EM adaptativo

Para localizar un robot es necesario un mapa del entorno en el que se mueve. Un problema interesante es la construcción automática de dicho mapa.

Esta construcción automática proporciona una mayor fiabilidad al proceso de localización, ya que los modelos bayesianos que se utilizan en este último problema son los mismos que los que se utilizan en el problema del mapeado. Un modelo de entorno construido con el mismo modelo con el que después se va a localizar el robot será más robusto que un modelo construido a mano.

En capítulos anteriores se han definido los mapas paramétricos. Veremos en este capítulo que es posible determinar los parámetros que mejor adaptan un determinado modelo a las lecturas realizadas por el robot en el entorno, *aún sin conocer las posiciones absolutas desde las que se han realizado estas lecturas*.

### 6.1 Introducción

El algoritmo EM, introducido por Dempster *et al.* (Dempster, Laird, y Rubin 1977), proporciona una técnica iterativa para realizar una estimación de máxima verosimilitud de un conjunto de parámetros en problemas en los que existen *datos ocultos* que dependen estadísticamente de los parámetros a estimar y de los datos observados.

La estimación de mapas del entorno es un ejemplo del tipo de problemas en los que se puede aplicar este algoritmo. Tal y como se detalla en el capítulo 2, el problema se puede formalizar de la siguiente manera. Un robot móvil ejecuta una secuencia acciones de movimiento ( $\mathbf{a}_1, \dots, \mathbf{a}_{N-1}$ ) por un entorno dado, tomando una serie de observaciones del mismo (lecturas de alcance, por ejemplo) ( $\mathbf{z}_1, \dots, \mathbf{z}_N$ ). Estas acciones y mediciones

constituyen los datos observados. Las posiciones del robot ( $\mathbf{x}_1, \dots, \mathbf{x}_N$ ) desde las que se han realizado las mediciones no son conocidas, y constituyen los *datos ocultos* del problema. Se trata de estimar los parámetros  $\phi = \{d_1, \dots, d_n\}$  que definen el mapa del entorno de máxima verosimilitud, dadas las acciones y observaciones y considerando los datos ocultos. Para resolver el problema se cuenta con un modelo que calcula la verosimilitud de unas lecturas dadas una posición y unos parámetros del modelo del entorno. También se dispone de un modelo de movimiento que predice la siguiente posición del robot, dada la posición actual y la acción ejecutada.

En el apéndice B se detalla el algoritmo EM. Un resumen de su funcionamiento, aplicado al problema del mapeado, es el siguiente. Sean  $X = (\mathbf{x}_1, \dots, \mathbf{x}_T)$  la secuencia de posiciones del robot (datos ocultos),  $Z = (\mathbf{z}_1, \dots, \mathbf{z}_T)$  la secuencia de observaciones del entorno y  $A = (\mathbf{a}_1, \dots, \mathbf{a}_{T-1})$  la secuencia de acciones realizadas por el robot. Los conjuntos  $Z$  y  $A$  constituyen los datos observados, y se denota por  $Y$  al conjunto total de datos  $Y = X \cup Z \cup A$ . Los parámetros a estimar son unos parámetros  $\phi$  que determinan el mapa del entorno.

El algoritmo EM busca el mapa del entorno  $\phi$  que maximiza el logaritmo de la función de verosimilitud marginal de las lecturas y acciones observadas sobre todas las posibles posiciones del entorno

$$\ln p(Z, A | \phi) = \int_X \ln p(Z, A, X | \phi).$$

Tal y como se describe en el apéndice B, se comienza por un mapa inicial  $\phi^0$  y se van obteniendo nuevas soluciones de forma iterativa. En cada iteración se parte del mapa obtenido en la iteración anterior,  $\phi^k$ , para obtener el siguiente valor de los parámetros del mapa que maximizan la *verosimilitud esperada* de los datos observados y los datos ocultos, dado la estimación anterior del mapa y los datos observados. Formalmente, se buscan los parámetros del mapa que cumplen

$$\begin{aligned} \arg \max_{\phi} E[\ln p(Z, A, X | \phi) | Z, A, \phi^k] &= \\ &= \arg \max_{\phi} \int_X \ln[p(X, Z, A | \phi)] p(X | Z, A, \phi^k). \end{aligned}$$

La última ecuación no se utiliza realmente en el algoritmo, se introduce para clarificar el término condicional de la primera expresión.

La iteración del algoritmo EM consiste en un paso de estimación (paso E), en el que se obtienen los valores esperados de las posiciones del entorno, seguido de un paso de maximización (paso M). En cada iteración la verosimilitud aumenta de forma monótona. En la tabla 6.1 se formula el algoritmo EM (ver apéndice B para mayor detalle).

<b>Algoritmo EM</b>	
1.	Sea $\phi^0$ la solución inicial
2.	Repetir hasta la convergencia
(a)	Paso E: Calcular el valor esperado de la verosimilitud de los datos completos, condicionados por los datos observados y por la solución actual $\phi^k$
	$Q(\phi   \phi^k) = E[\ln p(Z, A, X   \phi)   Z, A, \phi^k].$
(b)	Paso M:
	$\phi^{k+1} \leftarrow \arg \max_{\phi} Q(\phi   \phi^k).$

Tabla 6.1: Formulación del algoritmo *estimación-maximización*.

## 6.2 Enfoque muestral del algoritmo EM

En un gran número de problemas no es posible calcular analíticamente los pasos de estimación y maximización, por lo que hay que utilizar otro tipo de enfoques.

Se han propuesto algunos enfoques que implementan el paso de estimación utilizando técnicas de muestreo (ver (Cappe, Doucet, Lavielle, y Moulines 1999) para un resumen de las distintas propuestas genéricas). En resumen, todos estos trabajos utilizan el valor estimado de los parámetros en el instante anterior  $\phi^k$  y los datos observados  $z_1, \dots, z_T$  para obtener un conjunto de  $N$  muestras de los datos ocultos. Los valores ocultos esperados y la función  $Q$  se pueden estimar entonces a partir de las  $N$  muestras.

En el marco de la estimación bayesiana temporal, algunos trabajos muy recientes utilizan los algoritmos de estimación muestral como estimadores del paso de estimación. Por ejemplo, North y Blake (North y Blake 1998) utilizan el algoritmo EM para estimar los parámetros que determinan el modelo de movimiento de objetos en secuencias de aprendizaje.

En nuestro caso, dado que la localización es frecuentemente multimodal, utilizar como estimador la posición esperada, calculando la media de las muestras, introduciría errores

muy importantes. Proponemos, en su lugar, utilizar un estimador de posición robusto como es la posición de máxima probabilidad.

En cuanto al paso de maximización, no conocemos propuestas genéricas en la literatura relacionada con el algoritmo EM que permitan realizar la maximización cuando no se dispone de una fórmula analítica. En el campo de la visión artificial, sin embargo, algunos trabajos proponen utilizar técnicas adaptativas para encontrar parámetros de máxima verosimilitud, dados unos datos (ver (Hornegger y Niemann 1997) para una introducción a estos métodos). Es posible aplicar estas técnicas al algoritmo EM, si se considera que en la fase de maximización se han obtenido los valores ocultos esperados.

Para aplicar el algoritmo EM al problema de estimación del mapa del entorno es necesario introducir estas adaptaciones. Un resumen de la propuesta que se plantea en este capítulo se muestra en la tabla 6.2.

**Algoritmo EM muestral**

A partir de un valor anterior de los parámetros a estimar  $\phi^k$ , de los datos observados  $Z = \{\mathbf{z}_1, \dots, \mathbf{z}_T\}$  y de las acciones del robot  $A = \{\mathbf{a}_1, \dots, \mathbf{a}_{T-1}\}$  se obtienen las posiciones de máxima verosimilitud del robot  $\hat{X} = \{\hat{\mathbf{x}}_1, \dots, \hat{\mathbf{x}}_T\}$  y se refina la estimación de los parámetros, obteniéndose  $\phi^{k+1}$ .

**1. Fase de estimación:**

Utilizar el filtro *bootstrap* suavizado con el algoritmo de Kitagawa para obtener las posiciones de máxima verosimilitud del robot  $\hat{X} = \{\hat{\mathbf{x}}_1, \dots, \hat{\mathbf{x}}_T\}$ , dada la función de verosimilitud de las observaciones  $p(\mathbf{z}_i | \mathbf{x}_i, \phi^k)$  y el modelo de movimiento  $p(\mathbf{x}_i | \mathbf{x}_{i-1}, \mathbf{a}_{i-1})$ .

**2. Fase de maximización:**

Estimar con una búsqueda aleatoria adaptativa los parámetros del mapa del entorno que maximizan la verosimilitud de las lecturas y las posiciones esperadas

$$\phi^{k+1} \leftarrow \arg \max_{\phi} \ln p(Z, A, \hat{X} | \phi)$$

Tabla 6.2: Versión muestral del algoritmo EM para estimación de mapas del entorno.

### 6.2.1 Estimación de posiciones esperadas

El filtro *bootstrap* realiza la estimación muestral de la densidad a posteriori  $p(\mathbf{x}_t | Z^t, A^{t-1})$ , representando la función por un conjunto de muestras  $\{(\mathbf{m}_t^i, \pi_t^i), i = 1 \dots N\}$ . Esta función de densidad representa toda la información conocida acerca de las posiciones del robot,  $\mathbf{x}_t$ , dada la historia de sus observaciones  $Z^t = \{\mathbf{z}_1, \dots, \mathbf{z}_t\}$  y de sus acciones  $A^{t-1} = \{\mathbf{a}_1, \dots, \mathbf{a}_{t-1}\}$ . A partir de cada conjunto de muestras es posible estimar la esperanza de la distribución a posteriori para ese instante de tiempo utilizando las muestras y sus pesos asociados

$$E[\mathbf{x}_t | Z^t, A^{t-1}] \approx \hat{\mathbf{x}}_t = \sum_{i=1}^N \pi_t^i \mathbf{m}_t^i \quad (6.1)$$

Estos valores esperados, sin embargo, no son representativos en el caso de una distribución multimodal. Por ello proponemos utilizar en su lugar la muestra de mayor verosimilitud de cada instantánea

$$\hat{\mathbf{x}}_t = \mathbf{m}_t^i | \pi_t^i \geq \pi_t^j \quad \forall j = 1 \dots N \quad (6.2)$$

como el conjunto de soluciones de la fase de estimación del algoritmo EM.

Es posible, sin embargo, ajustar aun más estas estimaciones. Las estimaciones  $\hat{\mathbf{x}}_t$  recogen únicamente la información de las observaciones del robot hasta el instante  $t$ . Estas estimaciones suelen tener un alta variabilidad, ya que durante el seguimiento en línea es normal que se generen varias hipótesis de localización para un mismo instante de tiempo (recordemos otra vez que se estamos tratando con distribuciones multimodales). Pero también sucede con frecuencia que todas las hipótesis menos una desaparecen cuando se hace evidente que corresponden a distracciones o errores de estimación. Por ello, cuando se dispone de la secuencia completa de observaciones (hasta el instante  $T$ ), es posible utilizar toda esta información para mejorar estos estimadores, ya que se puede eliminar variabilidad producidas por distractores temporales. Se trata entonces de estimar la densidad de probabilidad  $p(\mathbf{x}_t | Z^T, A^{T-1})$ .

Isard (Isard y Blake 1998b), recogiendo la formulación inicial de Kitagawa (Kitawa 1996), formula un algoritmo para muestrear estas distribuciones. A continuación presentamos una adaptación de este algoritmo que considera la secuencia de acciones  $(\mathbf{a}_1, \dots, \mathbf{a}_{T-1})$  realizadas por el robot.

El algoritmo consiste en un barrido hacia adelante (equivalente al cálculo de los valores  $\alpha$  en el algoritmo Baum-Welch), en el que se generan las muestras  $\{(\mathbf{m}_t^i, \pi_t^i)\}$  para  $t = 1, \dots, T$ , utilizando el filtro *bootstrap*. Una vez generadas las muestras se reajustan los pesos  $\pi_t^i$ , para que representen la evidencia proporcionada por las observaciones posteriores  $(\mathbf{z}_t, \dots, \mathbf{z}_T, \mathbf{a}_t, \dots, \mathbf{a}_{T-1})$ .

Para hacer más compacta la notación, introducimos la siguiente modificación. Sean  $Z_j^k = (\mathbf{z}_j, \dots, \mathbf{z}_k)$  y  $A_j^k = (\mathbf{a}_j, \dots, \mathbf{a}_k)$  las secuencias de observaciones y acciones desde el instante  $j$  hasta el instante  $k$  (con  $j \geq k$ ). Con esta notación podemos expresar la probabilidad de una posición  $\mathbf{x}_t$  dado todo el conjunto de observaciones y acciones como

$$\begin{aligned} p(\mathbf{x}_t | Z_1^T, A_1^{T-1}) &\propto \\ &\propto p(\mathbf{x}_t, Z_t^T, A_t^{T-1} | Z_1^{t-1}, A_1^{t-1}) = \\ & p(Z_t^T, A_t^{T-1} | \mathbf{x}_t) p(\mathbf{x}_t | Z_1^{t-1}, A_1^{t-1}) \end{aligned} \quad (6.3)$$

Esta reordenación permite que las posiciones muestreadas  $\mathbf{m}_t^i$  permanezcan fijas después del paso de suavizado. Recordemos que el conjunto  $\{\mathbf{m}_t^i\}$  es, aproximadamente, una muestra correcta de la distribución  $p(\mathbf{x}_t | Z_1^{t-1}, A_1^{t-1})$ , por lo que, al reemplazar los pesos originales de las muestras  $\pi_t^i$  por los pesos suavizados

$$\psi_t^i = p(Z_t^T, A_t^{T-1} | \mathbf{x}_t = \mathbf{m}_t^i) \quad (6.4)$$

El conjunto  $\{(\mathbf{m}_t^i, \psi_t^i)\}$ , una vez normalizado, aproximará la distribución requerida  $p(\mathbf{x}_t | Z_1^T, A_1^{T-1})$ .

Un algoritmo recursivo para calcular las funciones de densidad  $p(Z_t^T, A_t^{T-1} | \mathbf{x}_t)$  se puede formular matemáticamente como sigue

$$p(Z_t^T, A_t^{T-1} | \mathbf{x}_t) = p(\mathbf{z}_t | \mathbf{x}_t) p(Z_{t+1}^T, A_{t+1}^{T-1} | \mathbf{x}_t) \quad (6.5)$$

$$p(Z_{t+1}^T, A_{t+1}^{T-1} | \mathbf{x}_t) = \int_{\mathbf{x}_{t+1}} p(Z_{t+1}^T, A_{t+1}^{T-1}) p(\mathbf{x}_{t+1} | \mathbf{x}_t, \mathbf{a}_t) \quad (6.6)$$

Una implementación concreta requiere la derivación de una aproximación  $\delta_t^i$  a la función  $p(Z_{t+1}^T, A_{t+1}^{T-1} | \mathbf{x}_t = \mathbf{m}_t^i)$ . La integral se puede aproximar mediante un sumatorio

$$p(Z_{t+1}^T, A_{t+1}^{T-1} | \mathbf{x}_t = \mathbf{m}_t^i) \approx \delta_t^i = \sum_{l=1}^N \psi_{t+1}^l \frac{p(\mathbf{x}_{t+1} = \mathbf{m}_{t+1}^l | \mathbf{x}_t = \mathbf{m}_t^i, \mathbf{a}_t)}{\gamma_t^i}, \quad (6.7)$$

donde

$$\psi_{t+1}^l = p(Z_{t+1}^T, A_{t+1}^{T-1} | \mathbf{x}_{t+1} = \mathbf{m}_{t+1}^l) \quad (6.8)$$

$$\gamma_t^i = \sum_{k=1}^N \pi_t^k p(\mathbf{x}_{t+1} = \mathbf{x}_{t+1}^k | \mathbf{x}_t = \mathbf{m}_t^i, \mathbf{a}_t) \quad (6.9)$$

representan, respectivamente, los pesos calculados en el paso anterior y una corrección de las probabilidades que tiene en cuenta los pesos  $\pi_t^i$ .

En la tabla 6.3 (pag. 133) se presenta un algoritmo que realiza estos cálculos.

### 6.2.2 Mapas de máxima verosimilitud

Una vez calculados los parámetros ocultos esperados (posiciones del robot)  $\hat{X} = (\hat{\mathbf{x}}_1, \dots, \hat{\mathbf{x}}_T)$  a partir de la estimación del mapa anterior  $\phi^k$  y de los datos observados, el algoritmo EM realiza un paso de maximización. En este paso se debe obtener la nueva estimación de los parámetros del mapa,  $\theta^{k+1}$ , buscando los parámetros de máxima verosimilitud. Éstos son los que maximizan el logaritmo de la verosimilitud de las posiciones esperadas, las lecturas del entorno y las acciones,

$$\phi^{k+1} \leftarrow \arg \max_{\phi} \ln p(Z, A, \hat{X} | \phi). \quad (6.10)$$

En la sección 2.6 se formulaba la función de verosimilitud  $p(Z, A, \mathbf{x}_1, \dots, \mathbf{x}_T | \phi)$  como

$$p(Z | \mathbf{x}_1, \dots, \mathbf{x}_T, \phi) p(\mathbf{x}_1, \dots, \mathbf{x}_T | Z, A),$$

y, tras desarrollar esta expresión, se llegaba a la ecuación

$$p(Z, A, \mathbf{x}_1, \dots, \mathbf{x}_T | \phi) = p(\mathbf{x}_1) \prod_{t=1}^T p(\mathbf{z}_t | \mathbf{x}_t, \phi) \prod_{t=2}^T p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{a}_{t-1}). \quad (6.11)$$

Sustituyendo esta ecuación en la ecuación 6.10 se llega a la formulación final del mapa de máxima verosimilitud

$$\begin{aligned} \phi^{k+1} &\leftarrow \arg \max_{\phi} \left[ \ln p(\mathbf{x}_1) \prod_{t=1}^T p(\mathbf{z}_t | \mathbf{x}_t, \phi) \prod_{t=2}^T p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{a}_{t-1}) \right] = \\ &= \arg \max_{\phi} \left[ \ln p(\mathbf{x}_1) + \sum_{t=1}^T \ln p(\mathbf{z}_t | \mathbf{x}_t, \phi) + \sum_{t=2}^T \ln p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{a}_{t-1}) \right]. \end{aligned}$$

Los términos  $p(\mathbf{x}_1)$  y  $p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{a}_{t-1})$  son constantes e independientes de los parámetros  $\phi$ , por lo que la expresión final queda como

$$\phi^{k+1} \leftarrow \arg \max_{\phi} \sum_{t=1}^T \ln p(\mathbf{z}_t | \mathbf{x}_t = \hat{\mathbf{x}}_t, \phi). \quad (6.12)$$

El paso de maximización del algoritmo EM depende, entonces, únicamente del modelo de observación definido (ver sección 4.2). Debido a que la densidad  $p(\mathbf{z}_t | \mathbf{x}_t, \phi)$  no tiene una fórmula cerrada no es posible derivar este máximo de forma analítica. Por ello, utilizamos un sencillo algoritmo de muestreo aleatorio adaptativo, generando un número grande de muestras de parámetros y realizando una maximización local para cada muestra.

Esta técnica de muestreo aleatorio ha sido utilizada con éxito en gran número de problemas de optimización de visión artificial (Hornegger y Niemann 1997).

**Algoritmo** de Kitagawa

Dado un conjunto de muestras de posiciones estimadas del robot  $\{(\mathbf{m}_t^i, \pi_t^i)\}$  para cada instante de tiempo  $t = 1, \dots, T$ , sustituye los pesos  $\pi_t^i$  de las muestras por los pesos  $\psi_t^i$  que representan la probabilidad de la muestra  $\mathbf{m}_t^i$  asociada dada *toda* la secuencia de observaciones  $p(\mathbf{x}_t = \mathbf{m}_t^i | Z_1^T, A_1^{t-1})$ .

1. **Inicializar** los pesos  $\psi_T^i$ :

$$\psi_T^i = \pi_T^i \quad \text{para } i = 1, \dots, N$$

2. **Iterar** hacia atrás para toda la secuencia  $t = T - 1, \dots, 1$

- (a) **Calcular** el modelo de movimiento:

$$\alpha_t^{i,j} = p(\mathbf{x}_{t+1} = \mathbf{m}_{t+1}^j | \mathbf{x}_t = \mathbf{m}_t^i, \mathbf{a}_t) \quad \text{para } i, j = 1, \dots, N.$$

- (b) **Calcular** los factores de corrección

$$\gamma_t^i = \sum_{j=1}^N \pi_t^j \alpha_t^{i,j} \quad \text{para } i = 1, \dots, N.$$

- (c) **Aproximar** las variables  $\delta$

$$\delta_t^j = \sum_{i=1}^N \psi_{t+1}^i \frac{\alpha_t^{i,j}}{\gamma_t^i} \quad \text{para } j = 1, \dots, N.$$

- (d) **Evaluar** los pesos suavizados

$$\psi_t^i = \pi_t^i \delta_t^i,$$

normalizar para que  $\sum \psi_t^i = 1$  y almacenarlos con la muestra correspondiente

$$\{(\mathbf{m}_t^i, \psi_t^i), i = 1, \dots, N\}.$$

Tabla 6.3: Algoritmo de suavizado de las probabilidades asociadas a las muestras del filtro *bootstrap*.

### 6.3 Experimentos

Se presenta en esta sección un conjunto de experimentos sobre el algoritmo de Kitagawa para el suavizado de la distribución de muestras y un experimento en el que se muestra el funcionamiento del enfoque EM para el mapeado.

- **Experimento 1.**

En este experimento (figuras 6.1, 6.2, 6.3, 6.4) se aplica el algoritmo de Kitagawa a la serie temporal de muestras producidas en el experimento 1 de localización. Recordemos que en ese experimento el robot circula por un pasillo hasta que llega a una zona con un espacio abierto a su izquierda. En este momento la distribución de muestras localiza correctamente al robot y durante el resto de instantáneas de la serie la localización se realiza correctamente.

En la figura 6.1 aparecen las muestras de localización en las instantáneas 1, 30, 67 y 107, junto con una gráfica de la evolución en el tiempo de la entropía de la distribución. En esta gráfica se puede comprobar que, después de una primera fase en la que la distribución tiene un alto grado de dispersión, debido a la ambigüedad de las lecturas del pasillo, en una segunda fase la distribución termina centrándose. Esto sucede alrededor del instante 30, y continua así hasta el final de la serie.

Es de esperar que el algoritmo de Kitagawa propague hacia atrás la información precisa en la localización de la segunda fase, centrandó también la primera fase.

En la figura 6.2 vemos el resultado de aplicar el algoritmo de Kitagawa. Se muestran dos instantáneas de la serie temporal (la 16 y la 19) antes y después de aplicar Kitagawa. En la parte superior de la figura se comprueba que la distribución está muy dispersa a lo largo del pasillo, indicando que todas esas posiciones tienen igual verosimilitud. Sin embargo, después de aplicar Kitagawa (abajo) vemos cómo obtienen la mayor verosimilitud las muestras cercanas a la posición real del robot, disminuyendo drásticamente la verosimilitud del resto de muestras.

En la figura 6.3, para resaltar este efecto, se representan las 30 mejores muestras de ambas instantáneas antes (arriba) y después (abajo) de aplicar Kitagawa.

Por último, la gráfica de la figura 6.4 muestra el error absoluto medio en la posición  $x$  y la entropía de la serie temporal, también antes y después de aplicar Kitagawa. Se comprueba que el suavizado reduce drásticamente el error absoluto medio y también reduce la entropía de la distribución.

- **Experimento 2.**

En un segundo experimento (figura 6.5) se aplica el suavizado a la serie temporal resultante del experimento 5 de localización (pag. 121). Recordemos que esta serie

temporal resultaba de la localización del robot moviéndose en una habitación cerrada. La simetría del entorno resultaba en una distribución multimodal, que persistía bastante tiempo hasta que se hacía evidente el problema del colapso del filtro.

La aplicación del suavizado de Kitagawa a esta serie, como podemos ver en la figura 6.5, no elimina la multimodalidad de la distribución. Este comportamiento es correcto, ya que esta multimodalidad es producto de una ambigüedad de localización, y no de ruido que desaparece al poco tiempo.

- **Experimento 3.**

El último experimento muestra un ejemplo de aplicación del algoritmo EM para realizar un mapeado. Los datos han sido tomados del simulador con el robot evolucionando en una habitación en forma de L invertida. En la figura 6.7 se muestra la evolución del robot, junto con las lecturas de los sonares.

El modelo de habitación a estimar se puede observar en la figura 6.6. Se define mediante tres parámetros,  $d_1$ ,  $d_2$  y  $d_3$ , que miden, respectivamente, la anchura del pasillo vertical, la profundidad del pasillo horizontal y la anchura del pasillo horizontal.

El objetivo del experimento es comprobar si es posible estimar correctamente los parámetros  $d_1$ ,  $d_2$  y  $d_3$  que determinan la forma de la habitación a partir de los datos obtenidos, utilizando el algoritmo EM.

En la figura 6.8 se muestra la evolución del algoritmo EM. Recordemos que en cada iteración del algoritmo consiste en dos fases:

1. Aplicación del filtro *bootstrap* para estimar la localización del robot en toda la secuencia a partir de los datos y del mejor mapa obtenido en la iteración anterior, seguida de un suavizado de la distribución mediante el algoritmo de Kitagawa.
2. Búsqueda de los parámetros de máxima verosimilitud para la mejor posición obtenida para cada instantánea de la secuencia temporal, después de aplicada la fase anterior.

En esta figura se muestra, en la columna de la izquierda, las mejores posiciones resultantes de la aplicación del filtro *bootstrap*, seguido del algoritmo de suavizado, al mapa resultante de la iteración anterior. La columna de la derecha muestra el mapa de máxima verosimilitud obtenido con el algoritmo de búsqueda adaptativa planteado en este capítulo.

Los valores iniciales de los parámetros son ( $d_1 = 500$ ,  $d_2 = 0$ ,  $d_3 = 500$ ). Se puede comprobar cómo el algoritmo converge rápidamente al mapa correcto ( $d_1 = 250$ ,  $d_2 = 400$ ,  $d_3 = 400$ ).

Iteración	Parámetros	Verosimilitud
1	(200,450,400)	-71.743
2	(250,400,400)	-20.193
3	(250,400,400)	-9.408

Tabla 6.4: Evolución del algoritmo EM. Parámetros del mejor mapa obtenido en cada iteración y suma de los logaritmos de las verosimilitudes de las posiciones para ese mapa. Parámetros del mapa correcto:  $d_1 = 250$ ,  $d_2 = 400$ ,  $d_3 = 400$ .

En la tabla 6.4 se muestra el resultado del valor a maximizar (suma de los logaritmos de las verosimilitudes de las posiciones, dado el mapa) en cada iteración, junto con los parámetros que proporcionan dicho resultado.

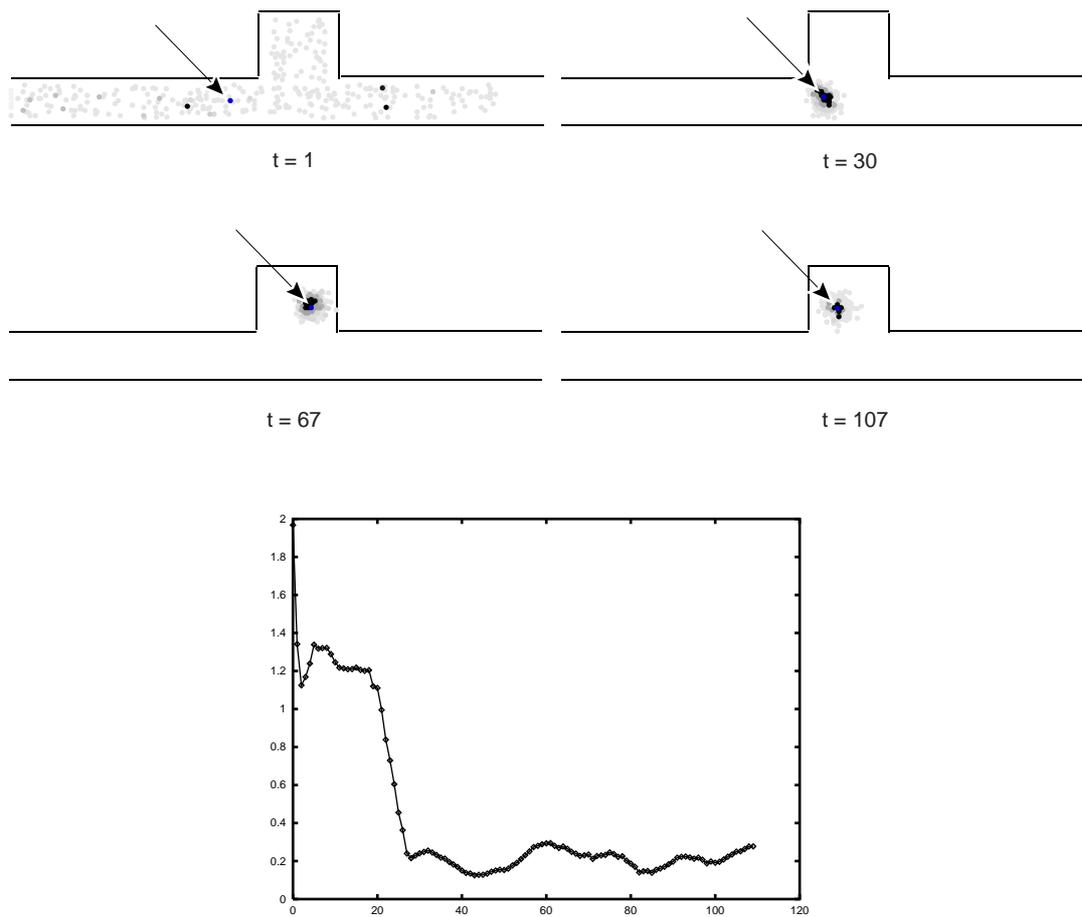


Figura 6.1: **Experimento 1.** Ejemplo de una serie temporal a la que se le aplicará el algoritmo de suavizado. Número de muestras=125. Abajo: representación de la entropía de la distribución de muestras en cada instante de tiempo (suavizada con una ventana de 3 instantes de tiempo) .

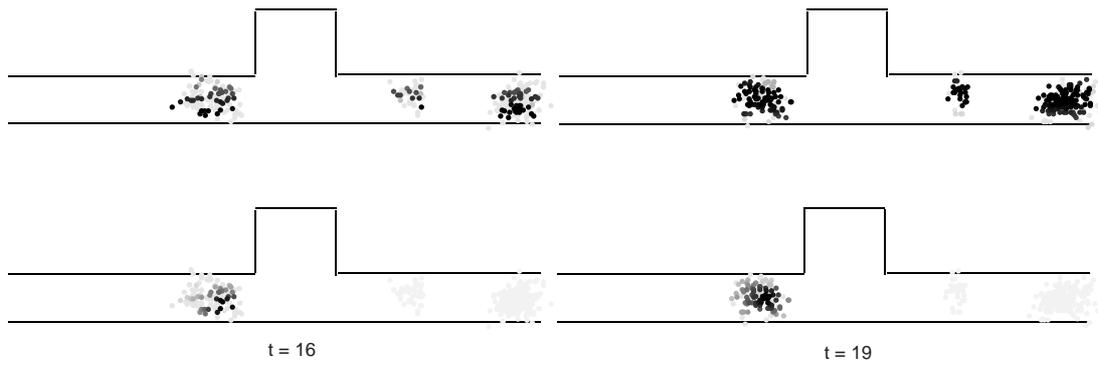


Figura 6.2: **Experimento 1.** Dos instantáneas de la serie temporal antes (arriba) y después (abajo) de aplicar el algoritmo de Kitagawa.

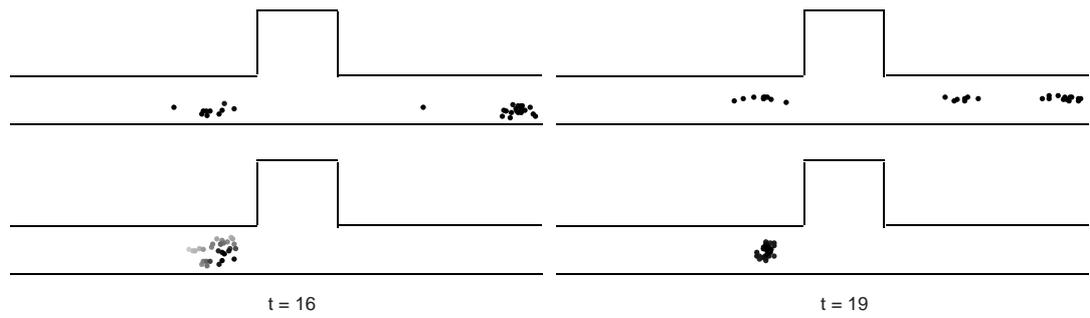


Figura 6.3: **Experimento 1.** Selección de las 30 mejores muestras de cada instantánea de la figura anterior.

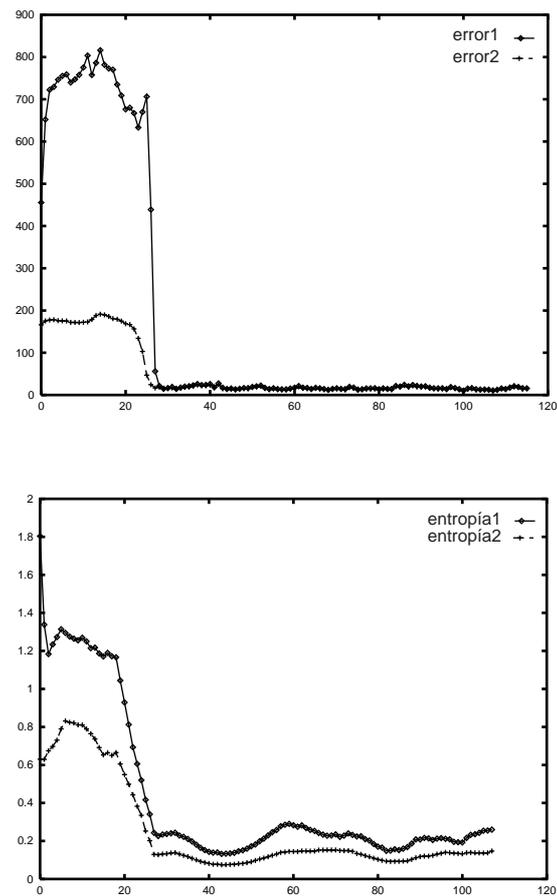


Figura 6.4: **Experimento 1.** Arriba: Error absoluto medio en la posición  $x$  del robot de la serie de la figura 6.1 antes (error1) y después (error2) de aplicar el algoritmo de Kitagawa. Abajo: entropía de la misma serie temporal antes (entropía1) y después (entropía2) de aplicar el algoritmo de Kitagawa. Número de muestras=125.

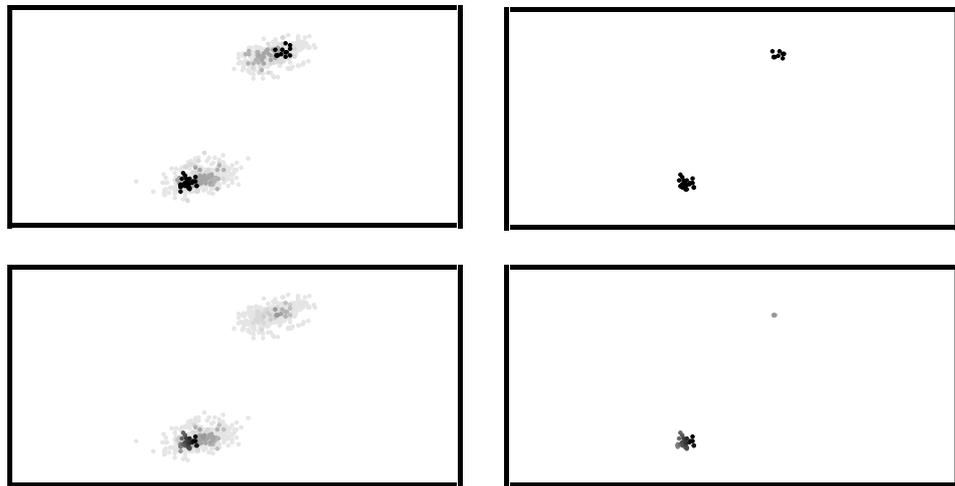


Figura 6.5: **Experimento 2.** Instantánea 29 de la serie temporal mostrada en la figura 5.17 antes (arriba) y después (abajo) de aplicar el algoritmo de Kitagawa. A la derecha de cada una las 30 muestras con mayor verosimilitud. El algoritmo de Kitagawa no elimina la multimodalidad de una distribución ambigua.



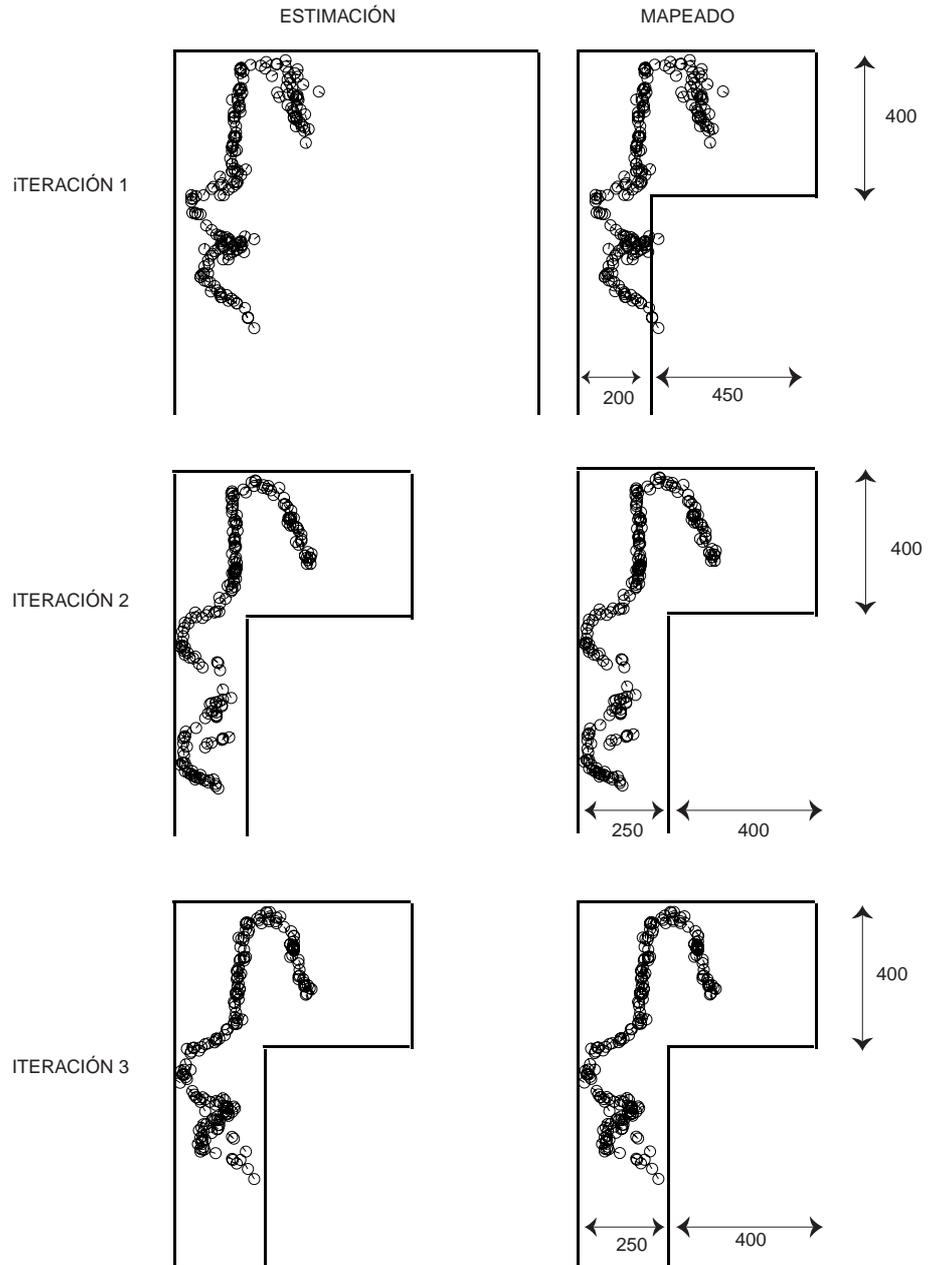


Figura 6.8: **Experimento 3.** Evolución del algoritmo EM para estimar los mejores parámetros  $d_1$ ,  $d_2$  y  $d_3$  que definen el mapa de la habitación.

## 6.4 Discusión

Se ha presentado en este capítulo un algoritmo de *estimación-maximización* (EM) que resuelve el problema del mapeado. En este problema, el robot evoluciona por el entorno recogiendo lecturas de los sensores e incrementos de posición obtenidos por odometría. Una vez realizada la exploración, se busca *off-line* el mapa que mejor explica las mediciones obtenidas, aún desconociéndose las posiciones absolutas por las que el robot ha pasado.

Para ello, el algoritmo EM itera una primera fase de estimación y una segunda de mapeado.

La fase de estimación se basa en la utilización del filtro *bootstrap* para obtener la serie completa de posiciones del robot en todos los instantes de tiempo. A continuación se corrigen las probabilidades de las muestras de esta serie temporal utilizando el algoritmo de Kitagawa, y se escogen las muestras de más probabilidad en cada instante de tiempo.

La fase de maximización busca los parámetros del mapa que maximizan la verosimilitud de las lecturas en las posiciones obtenidas en la fase anterior.

Se ha comprobado el buen funcionamiento del algoritmo de Kitagawa para realizar la corrección de las probabilidades de las muestras y el correcto funcionamiento del algoritmo EM para ajustar correctamente los parámetros, obteniendo el mapa del entorno.

