

Influencia de los parámetros de un filtro de partículas en la solución al problema de SLAM

A. Gil, O. Reinoso, *Member, IEEE*, L. Payá, M. Ballesta

Abstract—En este artículo se realiza un análisis sobre los parámetros más significativos de un filtro de partículas en su aplicación para la solución de SLAM. Se evalúan los resultados obtenidos en la creación de un mapa visual mediante un robot móvil. Un mapa visual está formado por un conjunto de puntos significativos tridimensionales que pueden ser extraídos a partir de imágenes del entorno. En este trabajo, los puntos significativos mencionados se emplean a modo de *landmarks* que pueden ser detectadas desde diferentes distancias y puntos de vista, permitiendo la localización del robot. Se supone que el robot cuenta con un sensor de visión estéreo que le permite detectar puntos significativos y obtener medidas relativas de distancia mientras se desplaza por el entorno. En nuestro caso, el mapa está formado por la posición tridimensional de las *landmarks* referidas a un sistema global de referencia. Además, cada *landmark* cuenta con un descriptor visual que la diferencia del resto. La solución que se presenta aquí, está basada en un filtro de partículas de tipo *Rao-Blackwellised*, estimándose de forma simultánea el camino seguido por el robot así como el mapa más probable del entorno. Se presentan resultados en simulación que confirman la validez de la solución. Se simula que el robot móvil se desplaza por el entorno, obteniendo medidas de odometría ruidosas, al mismo tiempo, el agente móvil realiza medidas relativas sobre las *landmarks* del entorno, estando también estas medidas corrompidas con un ruido gaussiano. La principal contribución del artículo es el estudio realizado sobre los parámetros del algoritmo de SLAM, hallando los valores más adecuados para la resolución del problema utilizando un sensor visual. Los resultados que se muestran se han obtenido utilizando unos modelos de ruido de la odometría y del sistema estéreo obtenidos a partir de unos sensores reales comunes en robótica móvil.

Index Terms— Landmarks visuales, mapa 3D, filtro de partículas, multi-robot SLAM.

I. INTRODUCCIÓN

UN robot móvil es un ingenio mecánico capaz de desplazarse por un entorno mientras realiza una determinada tarea. Una de las principales ventajas del uso de

los robots móviles consiste en dotar a los mismos con un cierto grado de inteligencia, de manera que puedan emplearse para reemplazar a un humano en tareas peligrosas o repetitivas. Un ejemplo lo podemos ver en [1], donde se describe el funcionamiento del robot Rhino, que funciona como robot guía en el *Deutsches Museum Bonn*. Aparte de poder realizar otras tareas, la principal habilidad con la que debe contar un robot móvil es la de moverse de forma autónoma por un entorno, evitando obstáculos, objetos y personas. Es necesario indicar que en muchas ocasiones, el entorno por el que se desenvuelve el robot no estará estructurado, pudiendo existir en ocasiones objetos o personas en movimiento (lo que se conoce como entornos dinámicos).

Para poder desplazarse por el entorno, el robot necesita generalmente un modelo de este: un mapa. Se pueden distinguir principalmente dos tipos diferentes de mapas: mapas de ocupación (*occupancy grid maps*) y mapas basados en puntos característicos (*landmark-based maps*). Estos puntos característicos se denominan generalmente *landmarks* (hito o baliza en castellano. Se utilizará preferiblemente el término anglo-sajón debido a su popularidad). De esta forma, podemos

definir *landmark* como una característica del entorno que es fácilmente detectable y reconocible por el robot. Los mapas de ocupación definen qué zonas del espacio están ocupadas y cuáles están libres de obstáculos. En este caso, el entorno se discretiza en celdas de tamaño regular, almacenándose en cada celda un valor proporcional a la probabilidad de que esa zona del espacio se encuentre ocupada. Por otra parte, los mapas basados en puntos característicos definen la posición de una serie de puntos en el espacio respecto de un sistema de referencia global. Para navegar en este tipo de mapas, el robot debe ser capaz de extraer las *landmarks* a partir de los datos y obtener medidas relativas sobre ellas. Como ejemplo, podemos citar el trabajo expuesto en [2], donde se extrae la posición del tronco de un árbol a partir de las medidas proporcionadas por un sensor de distancia láser. El tronco de cada árbol puede ser utilizado como *landmark*, ya que es fácilmente detectable desde diferentes distancias y ángulos. En la figura 1(a) se muestra un mapa de ocupación. Los píxeles marcados en blanco indican zonas del espacio que están libres de obstáculos. Las zonas negras definen las zonas del espacio ocupadas. La figura 1(b) muestra un mapa basado en puntos característicos. En este caso, cada punto característico define la posición bidimensional de uno de los árboles en el parque [2].

Manuscrito recibido el 18 de Julio de 2007. Este trabajo ha sido realizado gracias a la ayuda concedida por el Ministerio de Educación y Ciencia. Ref: DPI2004-07433-C02-01. Título: Herramientas de teleoperación colaborativa: aplicación al control cooperativo de robots.

A. Gil, O. Reinoso, L. Payá y M. Ballesta trabajan en la Universidad Miguel Hernández, Departamento de Ingeniería de Sistemas Industriales, en Elche (Alicante), España. Tfno: 0034-966658620; fax: 0034-966658979; e-mail: arturo.gil@umh.es.

No obstante, contar con un mapa detallado del entorno por donde debe navegar el robot no es siempre posible. Así, por ejemplo, podemos pensar en un entorno de oficinas en el que existan armarios, mesas, cajas etc... La tarea de medir todos estos elementos y reflejarlos sobre el mapa es claramente tediosa y complicada. Por otra parte, para que el mapa tenga utilidad, será necesario rehacer el mapa cada vez que alguno de estos elementos cambia de lugar.

En consecuencia, por todo lo dicho anteriormente, para que un robot móvil pueda ser realmente autónomo deberá contar con la habilidad esencial de explorar un entorno y crear un mapa de él. Sólo así podrá ser capaz de navegar eficientemente por cualquier espacio mientras realiza la misión que se le ha encomendado. El problema de crear un mapa mientras el robot se localiza dentro de él se denomina: *Simultaneous Localization and Mapping* (SLAM) o, alternativamente, *Concurrent Map Building and Localization* (CMBL). Dicho de otra manera, el problema de SLAM trata sobre la construcción de un mapa de un entorno utilizando una secuencia de medidas obtenidas por un robot en movimiento.

Debido a que el movimiento del robot está sujeto a un error en la medida (la odometría del robot es imprecisa), la exploración de un entorno induce necesariamente un problema de localización. Este problema es de tipo recurrente, ya que, si existe un error en la localización del robot, éste, necesariamente inducirá un error en el mapa generado, que, al mismo tiempo, inducirá un error en la localización. Según se verá en el apartado IV, este problema se puede descomponer en dos partes: Una estimación del camino del robot y una estimación del mapa condicionada al camino.

En la figura 2(a) se muestra el mapa creado utilizando únicamente la medida de odometría proporcionada por el robot. Claramente se puede observar que el mapa no es consistente (que aparecen muros dobles, zonas difuminadas y elementos en el mapa que no existen en la realidad). En la figura 2(b) se muestra el mapa creado utilizando la técnica de SLAM expuesta en [3].

En este trabajo se plantea la idea de crear un mapa de *landmarks* tridimensionales. Las *landmarks* se eligen como puntos significativos de un entorno extraídos a partir de imágenes tomadas por un sistema estéreo de visión. Estos puntos los llamaremos *landmarks* visuales, ya que son puntos característicos del entorno extraídos a partir de información visual del entorno. El mapa del entorno está formado por la posición tridimensional de los puntos característicos respecto de un sistema de referencia.

Además, cada punto está caracterizado por un descriptor visual, que lo diferencia del resto de puntos. Un mapa de este tipo lo llamaremos mapa visual. En lo sucesivo, suponemos que el robot está equipado con un sistema estéreo de visión que le permite medir la posición relativa de una *landmark* visual respecto del sistema de referencia de cámara. A continuación, el robot explorará el entorno, realizando una serie de movimientos y obteniendo medidas relativas de las *landmarks*. A partir de esta información, y utilizando el algoritmo de SLAM que se expone aquí, es posible estimar

simultáneamente el camino seguido por el robot y el mapa.

Como resultados del estudio, se presentan las principales conclusiones obtenidas acerca de la influencia de los múltiples parámetros considerados en la resolución del problema de SLAM a medida que un robot navega por un entorno desconocido con objeto de crear el mapa a la vez que se localiza. Una de las principales aportaciones consiste en considerar *landmarks* visuales cuentan con un determinado descriptor, de forma que el problema de la asociación de datos se puede resolver de forma más robusta. En este sentido se introduce una nueva forma de asociar las observaciones realizadas a las *landmarks* que se encuentran almacenadas en el mapa. A través de los resultados alcanzados tras multitud de experimentos en simulación se han extraído los parámetros más influyentes en la optimización del algoritmo y en la determinación del mapa más adecuado. Entre estos parámetros evaluados destacan el número de partículas del mapa, el número de observaciones introducidas por el sistema de visión, la influencia de la distancia máxima de observación, etc.

El resto del artículo se organiza de la siguiente manera: En el apartado II se comentan los antecedentes más notables relacionados con el problema de SLAM. A continuación, en el apartado III se detallan las diferentes soluciones empleadas hasta el momento para extraer y utilizar *landmarks* visuales del entorno. En el apartado IV se sintetiza una solución al problema de SLAM propuesta inicialmente por Montemerlo y Thrun en [2], [4] y [5]. En el apartado V se analiza detalladamente esta propuesta mediante un conjunto de simulaciones que permiten modificar los parámetros del filtro. Finalmente, en el apartado VI se proponen nuevos trabajos y líneas de investigación.

II. ANTECEDENTES

Hasta la actualidad, la gran mayoría de los trabajos realizados sobre SLAM se han centrado en crear mapas 2D de ocupación utilizando sensores láser y sonar. Generalmente, se asume que los mapas de ocupación se generan proyectando el espacio 3D ocupado sobre el suelo 2D del entorno. Las técnicas de SLAM utilizando datos de distancia láser son técnicas ya maduras en la actualidad, que permiten construir mapas de ocupación de manera rápida y eficiente (por ejemplo, [3], [5], [6]-[9]). En la actualidad, la principal desventaja de este tipo de mapas es que únicamente representan parcialmente el entorno. Además, estas técnicas presentan problemas cuando existen personas u objetos en movimiento durante la exploración del entorno.

Por otra parte, la utilización de sensores visuales en robótica móvil es relativamente reciente. Por ejemplo, en [10], Little y Murray presentan un sistema de navegación móvil basado en visión trinocular. En este trabajo, se extraen los puntos correspondientes en las tres imágenes utilizando un algoritmo basado en la correlación. A continuación, a partir de las correspondencias encontradas en las tres imágenes, se obtienen las coordenadas 3D de esos puntos en el espacio y se

crea un mapa local de ocupación. Con esta información, el robot es capaz de moverse por el entorno evitando obstáculos. Sin embargo, el trabajo expuesto no hace ninguna referencia a la localización del robot ni a la creación de un mapa del entorno.

En [11] y [12] aparece por primera vez la idea de crear un mapa tridimensional utilizando puntos significativos del entorno. Durante la exploración, el robot extrae imágenes del entorno utilizando un par estéreo. A continuación, el robot extrae un tipo especial de *landmarks* visuales (llamadas SIFT *features*, descritas en el apartado III). Estas *landmarks* visuales se emplazan en puntos significativos del entorno y tienen la propiedad de ser fácilmente detectables desde diferentes puntos de vista. Además, cada *landmark* visual se acompaña de un descriptor, que la caracteriza y la diferencia del resto. Dicho descriptor es invariante ante traslación, rotación, iluminación, y sólo parcialmente invariante ante cambios en el punto de vista.

En [13] se expone la manera de crear un mapa visual del entorno utilizando *landmarks* visuales, extraídas del entorno como características SIFT. En este trabajo, se utiliza un filtro de partículas *Rao-Blackwellized* para estimar de forma conjunta el camino recorrido por el robot y el mapa. En este caso, el mapa está formado por un conjunto de puntos tridimensionales, referidos a un sistema global de referencia.

III. LANDMARKS VISUALES

Durante los últimos años ha aumentado significativamente el interés por el SLAM visual, es decir, la creación de mapas utilizando únicamente información procedente de sensores visuales. Las principales razones que han suscitado este interés son:

- Los sistemas estéreo de visión son bastante más económicos que los sensores de distancia láser.
- Los sensores de distancia láser utilizados normalmente en robótica móvil ofrecen una visión 2D del entorno, mientras que sistemas de visión estéreo son capaces de obtener un modelo 3D más completo del entorno.
- Los sistemas de visión ofrecen la posibilidad de integrar en el robot móvil una gran cantidad de técnicas de Visión por Computador, como, por ejemplo: Reconocimiento de caras y objetos.

Resolver el problema de SLAM utilizando *landmarks* implica normalmente dos procesos principales: La detección, que permite al robot detectar la *landmark* desde un conjunto de poses en el entorno y la descripción de la *landmark*, que permite el reconocimiento de una *landmark* en particular en el mapa. Por ejemplo, en [4] se utilizan troncos de árboles para construir un mapa de un parque. Los troncos se extraen fácilmente a partir de los datos de distancia de un sensor láser a diferentes distancias y orientaciones. Por otra parte, la descripción de un tronco en particular podría estar basada en su diámetro. En el caso de las *landmarks* visuales, el descriptor se calcula a partir de una imagen obtenida alrededor del punto de interés en el entorno. El caso del SLAM visual, es particularmente complicado ya que:

- Los puntos significativos o *landmarks* no son siempre detectados desde todos los puntos de vista (p.e. diferentes distancias y ángulos) con lo que el robot no siempre es capaz de encontrar *landmarks* que detectó en instantes previos. Este aspecto dificulta la localización del robot cuando regresa a zonas exploradas previamente.
- La descripción de los puntos debe ser invariante ante cambios en el punto de vista de la cámara. Esta cualidad es especialmente complicada en el caso de las *landmarks* visuales. En consecuencia, la asociación de datos es un aspecto de difícil solución en este tipo de mapas.

La transformada SIFT (*Scale Invariant Feature Transform*) fue inicialmente desarrollada para aplicaciones de reconocimiento de objetos y extracción de características de imágenes [14], [15]. El proceso para extraer características SIFT se puede separar principalmente en dos partes: En una primera fase se extraen puntos significativos de la imagen. Estos puntos se extraen a partir de una diferencia de gaussianas (DoG) a diferentes escalas. Los puntos significativos se extraen como máximos y mínimos locales del resultado de la operación. A continuación, cada uno de los puntos se representa utilizando un vector de dimensión 128, calculado a partir de los gradientes de la imagen en un entorno local del punto. El descriptor así calculado proporciona invarianza ante traslaciones de la imagen, rotación, escalado y es parcialmente invariante a cambios en la iluminación y punto de vista. La transformada SIFT se ha utilizado con anterioridad en aplicaciones de robótica móvil, ofreciendo buenos resultados en la construcción de mapas [12], [13] y [16].

También en [17], se presenta un método de segmentación que permite identificar zonas características en una imagen, el método puede ser de utilidad si se pretende construir un mapa topológico del entorno, pero si el objetivo es obtener un mapa métrico del entorno, entonces se deben extraer puntos significativos en vez de regiones. Así, por ejemplo, Davison [18] extrae regiones significativas utilizando el operador de interés de Shi y Tomasi [19]. A continuación, a cada punto significativo se le asigna una subventana de la imagen centrada en el punto de 9x9 píxeles. Posteriormente, se utiliza una correlación normalizada para detectar el mismo punto en imágenes consecutivas y para detectar *landmarks* anteriores. Davison demuestra que esta representación es perfectamente válida para realizar SLAM monocular, siempre que los cambios en el punto de vista no sean excesivos.

IV. FASTSLAM

En esta sección se describe una solución al problema de SLAM popularmente conocida como FastSLAM. Esta solución es, en esencia, un filtro de partículas de tipo *Rao-Blackwellized* (RBPF), propuesta inicialmente por Murphy [20]. Claramente, en el problema de SLAM concurren dos aspectos diferentes:

- Estimación de la trayectoria del robot en el espacio.

- Estimación del mapa a partir de una serie de observaciones.

El problema de SLAM es de tipo recurrente, ya que si cometemos un error en la posición del robot este hecho inducirá un error en la construcción del mapa, que, a su vez, generará un error en la posición del robot. Sin embargo, los dos aspectos pueden ser separados. Es decir, si, de alguna manera, fuéramos capaces de conocer la trayectoria del robot en el entorno, entonces la reconstrucción del mapa sería un problema trivial.

Siguiendo la notación más popular en la literatura de SLAM, denominaremos s_t a la pose del robot en el instante t . Por otra parte $s^t = \{s_1, s_2, \dots, s_t\}$ se referirá al conjunto de poses del robot hasta el instante t (también llamado camino). Por otra parte, el mapa L está formado por un conjunto de N *landmarks* naturales inmóviles, situadas en un espacio tridimensional $L = \{l_1, l_2, \dots, l_N\}$. Cada *landmark* está definida como un vector tridimensional $l_k = (X_{g,k}, Y_{g,k}, Z_{g,k})$, referido respecto de un sistema de coordenadas global. El robot realiza una observación z_t sobre una *landmark* del mapa. El conjunto de observaciones realizadas por el robot hasta el instante t lo denominaremos $z^t = \{z_1, z_2, \dots, z_t\}$. El conjunto de movimientos (acciones) realizados por el robot hasta el instante t se escribirá como $u^t = \{u_1, u_2, \dots, u_t\}$. Por otra parte, la variable $c_t \in [1 \dots N]$ indicará la correspondencia de cada observación z_t con una de las N *landmarks* en el mapa. Si la observación actual no corresponde con ninguna de las *landmarks* del mapa, se indicará como $c_t = N+1$. De momento consideraremos que esta correspondencia es conocida. En la sección II-E indicaremos una solución para resolver este problema.

Si consideramos conocido el camino s^t recorrido por el robot, la construcción del mapa se reduce a la estimación de N *landmarks* independientes condicionadas al camino del robot. De esta manera, podemos descomponer el problema de SLAM en un problema de localización y en un conjunto de estimaciones de las *landmarks* condicionadas al camino del robot. Planteamos el problema de SLAM como la estimación de la siguiente función de probabilidad [4]:

$$p(s^t, L | z^t, u^t, c^t) = p(s^t | z^t, u^t, c^t) \prod_{k=1}^N p(l_k | s^t, z^t, u^t, c^t) \quad (1)$$

Es decir, la probabilidad conjunta sobre las poses del robot y el mapa L se puede expresar con el producto de dos términos: El primer término $p(s^t | z^t, u^t, c^t)$ hace referencia a la estimación del camino seguido por el robot, condicionado a las medidas de sus sensores z^t , los movimientos realizados u^t y a las correspondencias realizadas c^t . Mientras que el término de la derecha indica que el mapa puede ser representado mediante N estimadores independientes condicionados al camino del robot. De esta manera, se descompone el problema de SLAM en un problema de localización y en una serie de estimaciones individuales de *landmarks*, condicionadas al camino del robot $s^t = \{s_1, s_2, \dots, s_t\}$. El término $p(s^t | z^t, u^t, c^t)$ puede ser aproximado de forma eficiente mediante un filtro

de partículas. Cada una de las M partículas de este filtro cuenta con N estimadores independientes para cada una de las *landmarks*, implementados de forma sencilla utilizando un filtro de Kalman extendido (EKF). En consecuencia, existirán en total $N \cdot M$ filtros EKF. Los filtros de Kalman están todos condicionados al camino del robot, con lo que cada partícula contará con un conjunto de estimaciones diferentes (véase la tabla I). En el caso que nos ocupa, cada uno de estos filtros de Kalman será de dimensión 3, ya que el estado a estimar es la posición tridimensional de cada *landmark*. Representaremos cada partícula de la siguiente manera:

$$S_t^{[m]} = \{s_t^{[m]}, \mu_{1,t}^{[m]}, \Sigma_{1,t}^{[m]}, \mu_{2,t}^{[m]}, \Sigma_{2,t}^{[m]}, \dots, \mu_{N,t}^{[m]}, \Sigma_{N,t}^{[m]}\} \quad (2)$$

Donde $[m]$ se refiere al índice de la partícula. El valor de $\mu_{k,t}^{[m]}$ representa la estimación de la posición de la *landmark* l_k en el espacio, con matriz de covarianza $\Sigma_{k,t}^{[m]}$, asociada al camino de la partícula m . El conjunto de M partículas con sus mapas asociados lo denotaremos $S_t = \{S_t^1, S_t^2, \dots, S_t^M\}$. El algoritmo se puede resumir en los siguientes pasos:

- Generar un nuevo conjunto de partículas a partir del conjunto anterior.
- Actualizar las estimaciones de cada *landmark* de acuerdo con las observaciones realizadas.
- Asignar un peso a cada partícula.
- Realizar un muestreo con reposición del conjunto de partículas en función de su peso.

Según se ha expuesto, en esencia, un filtro de tipo *Rao-Blackwellised*, combina una representación de la pose utilizando partículas, con una estimación cerrada de ciertas variables. Originalmente, el algoritmo que se propone tiene una complejidad $O(MN)$, donde M es el número de partículas utilizado y N el número de *landmarks*. Montemerlo y Thrun [2], [4] y [5], desarrollan una propuesta a partir de la anterior que permite abordar el problema de SLAM con un coste computacional menor ($O(M \log N)$). Utilizando la propuesta que se resume en el presente apartado, se analiza en el apartado siguiente la influencia de los principales parámetros mediante un conjunto de simulaciones.

A. Generación de un nuevo conjunto de partículas

Como primer paso, se genera un nuevo conjunto de hipótesis S_t en base al conjunto S_{t-1} . Este conjunto de hipótesis se consigue muestreando del modelo de movimiento $p(s_t | s_{t-1}, u^t)$. Por lo tanto, calculamos una nueva pose $s_t^{[m]}$ para cada pose $s_{t-1}^{[m]}$ del conjunto S_{t-1} . La función $p(s_t | s_{t-1}, u^t)$ define el modelo de movimiento del vehículo móvil. Este modelo define la probabilidad sobre la pose s_t en el tiempo t condicionada a que en el instante anterior $t-1$ el robot se encontraba en la pose s_{t-1} y se ha ejecutado un comando de movimiento u_t .

La generación del nuevo conjunto de partículas se puede realizar a partir de las ecuaciones cinemáticas del robot, suponiendo que existe cierta cantidad de ruido en las acciones de control [21]. En la figura 3 se muestra un ejemplo de la generación de nuevas partículas. La línea sólida representa la acción de control realizada por el robot, correspondiente con la lectura de odometría realizada durante el movimiento. En el

comienzo del movimiento, todas las partículas se sitúan concentradas en el inicio de la línea sólida.

B. Estimación de las landmarks

La función de probabilidad $p(l_k | s^t, z^t, u^t, c^t)$ de cada una de las N landmarks, condicionada al camino s^t se estima utilizando filtros EKF de reducidas dimensiones. Debido a que las estimaciones de las landmarks están condicionadas al camino del robot, N filtros EKF están asociados a cada partícula. Por el momento, consideraremos que la observación z_t corresponde a la landmark l_{ct} . En el apartado IV-E introduciremos una solución a dicho problema. La actualización de la landmark l_k se realiza de forma independiente utilizando las ecuaciones clásicas de un filtro EKF que se indican a continuación:

$$\hat{z}_t = g(s_t^{[m]}, \mu_{c_t, t-1}^{[m]}) \quad (3)$$

$$G_{l_{ct}} = \nabla_{l_{ct}} g(s_t, l_{ct})_{s_t = s_t^{[m]}, l_{ct} = \mu_{c_t, t-1}^{[m]}} \quad (4)$$

$$Z_{c_t, t} = G_{l_{ct}} \Sigma_{c_t, t-1}^{[m]} G_{l_{ct}}^T + R_t \quad (5)$$

$$K_t = \Sigma_{c_t, t-1}^{[m]} G_{l_{ct}}^T Z_{c_t, t}^{-1} \quad (6)$$

$$\mu_{c_t, t}^{[m]} = \mu_{c_t, t-1}^{[m]} + K_t (z_t - \hat{z}_t) \quad (7)$$

$$\Sigma_{c_t, t}^{[m]} = (I - K_t G_{l_{ct}}^T) \Sigma_{c_t, t-1}^{[m]} \quad (8)$$

La variable \hat{z}_t se denomina la predicción de la observación para la landmark c_t . El modelo de observación $g(s_t, l_{c_t})$ se aproxima linealmente por la matriz Jacobiana $G_{l_{ct}}$. Se asume que el ruido en la observación es gaussiano con matriz de covarianzas R_t . En la ecuación (7) se actualiza la estimación anterior sobre la landmark $\mu_{c_t, t-1}^{[m]}$ en términos de la innovación $v = (z_t - \hat{z}_t)$. Finalmente, la ecuación (8) actualiza la matriz de covarianzas $\Sigma_{c_t, t-1}^{[m]}$ de la partícula m asociada a la landmark c_t . La actualización de cada filtro de Kalman requiere un tiempo constante para cada partícula, ya que las dimensiones de cada EKF son fijas.

A continuación se obtendrá la matriz R_t asociada al ruido en la observación. En el caso que nos ocupa, se están realizando medidas relativas sobre landmarks puntuales utilizando un sistema de visión estéreo. Si suponemos que el par estéreo es estándar (cámaras pin-hole y ejes perfectamente alineados), entonces las coordenadas 3D de un punto referidas al sistema de coordenadas de la cámara izquierda se pueden calcular como:

$$X_r = \frac{B(c - C_x)}{d}, Y_r = \frac{B(r - C_y)}{d}, Z_r = \frac{fB}{d} \quad (9)$$

Donde, (c, r) es la proyección del punto 3D en la imagen izquierda y d es la disparidad asociada al punto. El parámetro B recibe el nombre de línea base, y corresponde a la separación horizontal entre las cámaras del par estéreo. Los parámetros C_x y C_y se refieren al punto de intersección del eje óptico con el plano imagen en ambas cámaras. Podemos

calcular las covarianzas asociadas a la medida $z_t = (X_r, Y_r, Z_r)$ suponiendo una propagación del error lineal [21]:

$$\sigma_{X_r}^2 = \frac{B^2 \sigma_c^2}{d^2} + \frac{B^2 (c - C_x)^2 \sigma_d^2}{d^4} \quad (10)$$

$$\sigma_{Y_r}^2 = \frac{B^2 \sigma_r^2}{d^2} + \frac{B^2 (r - C_y)^2 \sigma_d^2}{d^4} \quad (11)$$

$$\sigma_{Z_r}^2 = \frac{f^2 B^2 \sigma_d^2}{d^4} \quad (12)$$

Durante los experimentos hemos utilizado $\sigma_d = 0.5$ pixels y $\sigma_r = \sigma_c = 10$ pixels. Estos valores se han extraído tras un análisis experimental de los sistemas de visión disponibles en nuestro laboratorio. La matriz de ruido R_t se calcula como $R_t = \text{diag}(\sigma_{X_r}^2, \sigma_{Y_r}^2, \sigma_{Z_r}^2)$.

C. Asignación de un peso a cada partícula

Las partículas generadas con el modelo de movimiento están distribuidas según $p(s^t | z^{t-1}, u^t, c^{t-1})$, ya que no se ha tenido en cuenta la última observación z_t realizada por el robot. En consecuencia, no coincide con la distribución que deseamos estimar: $p(s^t | z^t, u^t, c^t)$, en la cual se incluye toda la información de odometría y sensores hasta el momento t . Esta diferencia se corrige a través de un proceso llamado muestreo con reposición en función de la importancia (*sample importance resampling* o SIR). En esencia, se asigna un peso a cada partícula en función de la calidad con la que la pose y el mapa concuerdan con la observación actual z_t . En nuestro caso, escogeremos el peso de la partícula m según se expone en [2]:

$$\omega_t^{[m]} = \frac{1}{\sqrt{|2\pi Z_{c_t, t}|}} \exp \left\{ -\frac{1}{2} (z_t - \hat{z}_{c_t, t})^T [Z_{c_t, t}]^{-1} (z_t - \hat{z}_{c_t, t}) \right\} \quad (13)$$

Con motivo de mantener una notación sencilla, se ha considerado que el robot realiza únicamente una sola observación z_t en cada instante de tiempo. En el caso en el que se realicen un conjunto de B observaciones $z_t = \{z_{t,1}, z_{t,2}, \dots, z_{t,B}\}$, cada una con peso asociado $\omega_{t,1}^{[m]}, \omega_{t,2}^{[m]}, \dots, \omega_{t,B}^{[m]}$ entonces, se calcula el peso total $\omega_t^{[m]}$ como:

$$\omega_t^{[m]} = \prod_{i=1}^B \omega_{t,i}^{[m]} \quad (14)$$

D. Muestreo con reposición

Una vez se ha asignado un peso a cada partícula de acuerdo con la ecuación (13), se normalizan todos los pesos, de manera que $\sum_{i=1}^M \omega_t^{[i]} = 1$. A continuación, se genera un nuevo conjunto de partículas S_t al muestrear con reemplazamiento del conjunto S_{t-1} . Cada partícula $S_t^{[m]}$ formará parte del nuevo conjunto o no, según la probabilidad asociada a su peso. De esta manera, partículas con pesos mayores serán muestreadas con mayor probabilidad que aquellas que tengan un menor peso.

E. Asociación de datos

Hasta el momento se ha obviado un problema de gran importancia en SLAM, es decir, hemos supuesto que la medida z_t corresponde a la *landmark* c_t del mapa. Este problema es generalmente conocido como el *Data Association Problem*. En un trabajo previo se analiza el problema de la asociación de datos teniendo en cuenta medidas de distancia diferentes y teniendo en cuenta la apariencia visual de las características detectadas ([13], [23]). Según se ilustra en la figura 4 en algunos casos este problema puede ser de difícil solución. En la figura 4 se representan dos *landmarks* asociadas con dos elipses, representando la incertidumbre sobre su posición. Además, con línea discontinua se representa una medida z_t junto con su elipse de error. En este caso, no podemos distinguir con claridad a cual de las dos *landmarks* se debe asociar la medida z_t .

En la práctica, este es un problema de difícil solución, ya que el robot debe decidir si una observación fue generada por una de las *landmarks* del mapa, o se debe crear una *landmark* nueva. En nuestro caso, asociaremos la observación z_t a la *landmark* que minimice la distancia de Mahalanobis sobre todas las *landmarks* del mapa creado hasta el momento:

$$d = (z_t - \hat{z}_{c_t,t})^T [Z_{c_t,t}]^{-1} (z_t - \hat{z}_{c_t,t}) \quad (15)$$

Donde $Z_{c_t,t}$ es la matriz de covarianzas asociada a la innovación $v = (z_t - \hat{z}_t)$. Si el valor mínimo de la distancia de Mahalanobis supera cierto umbral límite d_0 , entonces se considera que la medida observada no corresponde con ninguna de las *landmarks* del mapa, y se creará una nueva *landmark* (apartado IV-F).

En los algoritmos de SLAM clásicos basados en el filtro de Kalman [8], [24]-[26], una única asociación de datos se escoge para todo el filtro. Es un hecho ampliamente constatado que, en presencia de errores en la asociación de datos todo el filtro puede diverger [27]. En cambio, el algoritmo expuesto aquí es capaz de funcionar aun cuando existan errores en la asociación de datos. Esto es debido a que cada partícula representa una hipótesis diferente sobre el camino del robot y la asociación de datos se hace para cada partícula independientemente, de manera que el mapa asociado a una partícula puede diferir del mapa asociado a otra. Por tanto, se mantienen múltiples hipótesis sobre la pose del robot y sobre el mapa. Aquellas partículas que hayan escogido las asociaciones correctas recibirán pesos mayores, ya que explican correctamente las observaciones realizadas. Por contra, aquellas partículas que hayan confundido su asociación de datos recibirán pesos menores, con lo que con gran probabilidad serán eliminadas en el proceso de muestreo.

F. Adición de nuevas landmarks

Según se dijo, si la función de distancia (15) tiene un valor alto para todas las *landmarks* del mapa deberemos crear una *landmark* nueva, ya que la observación actual z_t no se explica adecuadamente con ninguna de las *landmarks* existentes en el mapa hasta el momento. En este caso, el mapa asociado a la

partícula se debe ampliar, inicializándose un nuevo filtro EKF para dicha *landmark* con los siguientes valores:

$$c_t = N + 1 \quad (16)$$

$$\mu_{c_t,t}^{[m]} = g^{-1}(s_t^{[m]}, z_t) \quad (17)$$

$$\Sigma_{c_t,t}^{[m]} = G_{c_t}^T R_t^{-1} G_{c_t} \quad (18)$$

$$\omega_t^{[m]} = p_0 \quad (19)$$

Donde la función $g^{-1}(s_t^{[m]}, z_t)$ calcula la posición global de una marca a partir de la medida relativa z_t realizada por el robot. El valor p_0 es un peso asignado de forma experimental que determina la probabilidad con la que sobreviven las partículas que deciden crear una *landmark* nueva en su mapa.

G. Estimación de la mejor pose y el mejor mapa

Elegiremos el camino y el mapa asociados a la partícula con mayor peso acumulado durante toda la exploración. Alternativamente, elegiremos el índice de la partícula que maximice:

$$\hat{m} = \arg \max_m \sum_{t=1}^A \log(\omega_t^{[m]}) \quad (20)$$

Donde A es el número de movimientos realizados por el robot.

V. RESULTADOS DE SIMULACIÓN

A continuación, se describirá el entorno que se ha utilizado para realizar las simulaciones. Consideramos que existe un agente móvil en el entorno y que este puede realizar observaciones sobre una serie de *landmarks* tridimensionales en el entorno. En la figura 5 se puede observar una vista tridimensional del entorno que se simula y del robot real. Se representa un entorno interior, donde existen una serie de *landmarks* situadas sobre las paredes u objetos del entorno. La visibilidad de las *landmarks* está restringida por los objetos del entorno (p.e. el robot no puede detectar una *landmark* a través de una pared). Las *landmarks* observadas se indican con elipsoides, de tamaño proporcional a la incertidumbre en su posición. La principal ventaja de realizar una serie de experimentos en simulación es que se conoce el camino real seguido por el robot, así como el mapa. Esto nos permite evaluar con exactitud los resultados obtenidos al variar algunos de los parámetros del algoritmo. En concreto, evaluaremos la bondad de los resultados cuando se varía el número M de partículas empleado, la distancia máxima de observación d_{max} y el número B de medidas a integrar en cada instante.

El robot realiza una trayectoria predefinida en el entorno simulado, según se puede observar en la figura 6 con trazo continuo. Según se describe en la sección IV-A se simula una odometría ruidosa, marcada con línea discontinua. La cantidad de ruido que hemos introducido en la odometría es de la misma magnitud que el ruido que aparece típicamente en los experimentos con robots reales. Sobre la misma figura se señala con cuadrados la mejor estimación del camino seguido por el robot hasta el instante t . Las medidas relativas realizadas por el robot sobre las *landmarks* del mapa llevan asociadas un ruido distribuido según una $N(0, R_t)$ según el

ruido calculado en las ecuaciones (10)-(12). Se supone que, durante la simulación, la única información disponible sobre el movimiento del robot es su odometría y las medidas realizadas. Terminada la simulación, se compara el camino estimado con el camino real, para evaluar la calidad de los resultados. Por otra parte, se ha supuesto la asociación de datos conocida durante las simulaciones. Como se puede observar la estimación del camino realizado por el robot sigue con bastante fidelidad el camino real.

En la figura 7 se compara la posición y orientación reales del robot, la odometría y la estimación proporcionada por el algoritmo durante los diferentes movimientos del robot. Se observa claramente como el error en la odometría crece sin control, mientras que el error en la estimación se mantiene dentro de un intervalo.

Se han realizado un conjunto de simulaciones variando el número M de partículas del filtro. El mismo experimento se ha repetido 20 veces para cada valor de M , manteniéndose durante todos los experimentos las mismas lecturas de odometría y el mismo mapa. En la figura 8(a) se muestran los resultados, donde se ha evaluado el error RMS entre el camino real del robot. Con barras de error se indica un intervalo de 2σ calculado sobre el resultado de cada iteración para cada valor de M . El error RMS de posición calculado con la odometría tiene un valor de 3 m. En la figura, se puede observar claramente como decrece el error en la estimación según aumenta el número de partículas empleadas. Esto es debido a que la función de probabilidad $p(s^t | z^t, u^t, c^t)$ se aproxima con más exactitud conforme aumenta el número de partículas utilizadas. El límite inferior en el error de localización viene marcado por el modelo de error de las observaciones (10)-(12). Observando la gráfica se puede ver que a partir de 200 partículas la mejora en la localización no es significativa y demuestra que se pueden obtener unos buenos resultados utilizando 200-300 partículas. En la figura 8(b) se muestra el error RMS evaluado entre el mapa estimado por el algoritmo y el mapa real. El mapa estimado se ha elegido como aquél perteneciente a la partícula con mayor peso acumulado durante todo el camino.

Durante las simulaciones, hemos supuesto que un robot es capaz de detectar una *landmark* siempre que se encuentre a una distancia inferior a d_{max} y dentro de su campo de visión. Se han realizado diferentes simulaciones variando la distancia máxima de detección, agrupando los resultados en la figura 9. Según se puede observar, el error en la localización disminuye al aumentar la distancia máxima de observación. Cuando el robot observa *landmarks* a una distancia corta, continuamente está explorando nuevo espacio, con lo que su incertidumbre aumenta rápidamente. Al contrario, cuando la distancia máxima de observación es grande, el robot puede construir un mapa local y mantenerse localizado dentro de él, con lo que su incertidumbre crece más despacio. Según se observa en la figura, a partir de 10 metros de distancia máxima de observación, la mejora en la localización no es significativa. Este hecho se debe a que el modelo de ruido expuesto introduce grandes errores al observar *landmarks* lejanas, hecho que no mejora excesivamente la localización. En la figura 9(b) se puede observar la misma tendencia con el error RMS sobre las *landmarks* del mapa.

Cuando se obtienen medidas de distancia utilizando un sistema estéreo de visión, es una práctica común limitar la distancia máxima que se va a medir, lo que equivale generalmente a establecer una disparidad mínima en la medida. En la práctica, la mayoría de los sistemas de visión estéreo para interiores pueden proporcionar medidas fiables para una distancia máxima de unos 10 m. Según se puede observar en la figura 9 esta distancia es suficiente para construir un mapa con buena precisión.

VI. CONCLUSIONES

En este trabajo se ha expuesto un algoritmo que permite crear un mapa tridimensional mientras el robot móvil navega por un entorno desconocido. El mapa creado está basado en *landmarks* puntuales que son extraídas de imágenes adquiridas por un sistema estéreo de visión. La utilización de este tipo de mapas constituido por marcas tridimensionales del entorno resulta de una elevada utilidad en tareas de navegación autónoma por parte del robot móvil.

A partir del algoritmo que resuelve el problema de SLAM, se ha propuesto una nueva forma de realizar de manera más robusta el problema de la asociación de datos. Con este planteamiento se han realizado un conjunto de experimentos que muestran la evolución de los parámetros más influyentes en dicho algoritmo. Los resultados obtenidos permiten evaluar la calidad en la localización del robot y en la determinación del mapa tridimensional en función de la variación de cada uno de estos parámetros. Como resultado de los mismos se observa que no es necesario un número de partículas muy elevado con objeto de alcanzar una buena precisión en la localización tanto del robot como de las *landmarks* visuales que configuran el mapa. De igual forma se observa cómo a medida que aumenta la distancia de observación por encima de un determinado valor no se obtienen resultados más precisos.

Como trabajo futuro se plantea la extensión de estos trabajos a un entorno formado por múltiples robots que cooperan en la doble tarea tanto de localización como de construcción de un mapa tridimensional. Asimismo, se plantea tener en cuenta las localizaciones o detecciones de un robot por parte de otro robot con objeto de disminuir la incertidumbre de los resultados alcanzados. Al introducir esta nueva información se espera que los resultados de localización mejoren. Otra línea de investigación abierta se centra en las estrategias de exploración necesarias en un mapa de este tipo. Consideramos que el problema es de gran interés, ya que implica la coordinación de todo el equipo de robots para explorar el entorno para obtener el mejor resultado posible.

VII. REFERENCIAS

- [1] W. Burgard, A. Cremers, D. Fox, D. Hähnel, G. Lakemeyer, D. Schulz, W. Steiner, and S. Thrun, "Experiences with an interactive museum tourguide robot," *Artificial Intelligence*, vol. 114, no. 1-2, 2000.
- [2] S. Thrun, M. Montemerlo, D. Koller, B. Wegbreit, J. Nieto, and E. Nebot, "Fastslam: An efficient solution to the simultaneous localization and mapping problem with unknown data association," *Journal of Machine Learning Research*, 2004.

- [3] C. Stachniss, G. Grisetti, D. Hähnel, and W. Burgard, "Improved rao-blackwellized mapping by adaptive sampling and active loop-closure," in *SOAVE*, Ilmenau, Germany, 2004, pp. 1–15.
- [4] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "Fastslam: a factored solution to the simultaneous localization and mapping problem" in *Eighteenth national conference on Artificial intelligence*. Menlo Park, CA, USA: American Association for Artificial Intelligence, 2002, pp. 593–598.
- [5] M. Montemerlo, "Fastslam: A factored solution to the simultaneous localization and mapping problem with unknown data association," Ph.D. dissertation, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, July 2003.
- [6] S. Thrun, S. Thayer, W. Whittaker, C. Baker, W. Burgard, D. Ferguson, D. Hähnel, M. Montemerlo, A. Morris, Z. Omohundro, C. Reverte, and W. Whittaker, "Autonomous exploration and mapping of abandoned mines," *IEEE Robotics and Automation Magazine*, 2005.
- [7] S. Thrun, "A probabilistic online mapping algorithm for teams of mobile robots," *International Journal of Robotics Research*, vol. 20, no. 5, pp. 335–363, 2001.
- [8] J. E. Guivant, F. R. Masson, and E. M. Nebot, "Simultaneous localization and map building using natural features and absolute information," *Robotics and Autonomous Systems*, vol. 40, pp. 79–90, 2002.
- [9] W. Burgard, M. Moors, and F. Schneider, "Collaborative exploration of unknown environments with teams of mobile robots," *Proc. of the Dagstuhl Seminar on Plan-Based Control of Robotic Agents*. Springer-Verlag, 2002.
- [10] D. Murray and J. J. Little, "Using real-time stereo vision for mobile robot navigation," *Autonomous Robots*, vol. 2, no. 8, pp. 161–171, 2000.
- [11] S. Se, D. Lowe, and J. Little, "Vision-based mobile robot localization and mapping using scale-invariant features," *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2051–2058, 2001.
- [12] —, "Global localization using distinctive visual features," *Proceedings of the 2002 IEEE/RSJ Intl. Conference on Intelligent Robots and Systems EPFL*, 2002.
- [13] A. Gil, O. Reinoso, O. M. Mozos, and C. S.W. Burgard, "Improving data association in vision-based slam," in *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Beijing, China, 2006.
- [14] D. Lowe, "Object recognition from local scale-invariant features," *International Conference on Computer Vision*, vol. 2, 1999.
- [15] —, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 2, no. 60, pp. 91–110, 2004.
- [16] S. Se, D. Lowe, and J. Little, "Vision-based global localization and mapping for mobile robots," *IEEE Transactions on Robotics*, vol. 21, no. 3, pp. 364–375, 2005.
- [17] E. O. Nunes e A. Conci, "Segmentação por Textura e Localização do Contorno de Regiões em Imagens Multibandas", *IEEE LATIN AMERICA TRANSACTIONS*, VOL. 5, NO. 3, JUNE 2007, pp. 185-192.
- [18] A. J. Davison, "Real-time simultaneous localisation and mapping with a single camera," *ICCV*, 2003.
- [19] J. Shi and C. Tomasi, "Good features to track," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1994, pp. 593–600.
- [20] K. Murphy, "Bayesian map learning in dynamic environments," *Neural Information Processing Systems (NIPS)*, 1999.
- [21] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. The MIT Press, 2005.
- [22] P. R. Bevington and D. K. Robinson, *Data Reduction and Error Analysis for the Physical Sciences*. McGraw-Hill, 1992.
- [23] A. Gil, O. Reinoso, C. Fernandez, A. Rottmann, and O. M. Mozos, "Simultaneous localization and mapping in unmodified environments using stereo vision," in *Proceedings of the 3rd International conference on Informatics in control and automation in robotics*, Portugal, 2006, pp. 482–488.
- [24] M. W. M. G. Dissanayake, P. Newman, S. Clark, H. Durrant-Whyte, and M. Csorba, "A solution to the simultaneous localization and map building (slam) problem," *IEEE Transactions on Robotics and Automation*, vol. 17, 2001.
- [25] J. E. Guivant and E. Nebot, "Optimization of the simultaneous localization and map-building algorithm for real-time implementation,"

IEEE Transactions on Robotics and Automation, vol. 17, no. 3, pp. 242–257, 2001.

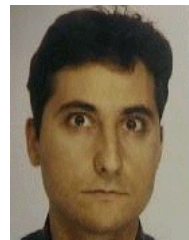
- [26] R. Smith, M. Self, and P. Cheeseman, "Estimating uncertain spatial relationships in robotics," *I.J. Cox and G.T. Wilfong, editors, Autonomous Robot Vehicles*, vol. 17, no. 3, pp. pp.167–193, 1990.

- [27] J. Neira and J. Tardos, "Data association in stochastic mapping using the joint compatibility test," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 6, pp. 890–897, 2001.

VIII. BIOGRAFÍAS



Arturo Gil recibió el grado de Ingeniero Industrial en la Universidad Miguel Hernández de Elche (UMH) en el año 2002, recibiendo el Premio Extraordinario de Ingeniería de la UMH ese mismo año. De 2000 a 2002 participa activamente en diversos proyectos en el Área de Ingeniería de Sistemas y Automática de la UMH. Desde 2003 trabaja como Profesor Colaborador en la Universidad Miguel Hernández, impartiendo materias relacionadas con control, informática y robótica. Sus líneas de investigación se centran en la robótica por computador. En la actualidad investiga sobre técnicas de construcción de mapas visuales mediante equipos de robots móviles. Es autor de numerosas comunicaciones y artículos que tratan sobre los temas citados. Es miembro de CEA-IFAC.



Óscar Reinoso recibió el grado de Ingeniero Industrial y Doctor Ingeniero Industrial en la Universidad Politécnica de Madrid (UPM) en 1991 y 1996 respectivamente. Desde 1994 a 1997 trabaja en la empresa Protos Desarrollo S.A. en el departamento de Investigación y Desarrollo en sistemas de Visión Artificial. Desde 1997 es Profesor Titular en la Universidad Miguel Hernández de Elche, donde imparte docencia en materias relacionadas con el control, robótica y visión por computador. Las líneas de investigación en las que participa se centran en robótica, robots teleoperados, robots trepadores, control visual y sistemas de inspección visual. Es autor de varios libros, artículos y comunicaciones en los temas anteriormente citados. Es miembro de CEA-IFAC y de IEEE.



Luis Payá obtuvo el grado de Ingeniero Industrial en la Universidad Miguel Hernández de Elche (UMH) en el año 2003, obteniendo el Premio Nacional de Ingeniería, por el Ministerio de Educación y Ciencia. Desde Octubre 2002 a Octubre de 2003 trabaja como Ingeniero en el Departamentode I+D de Boix Maquinaria S.A. Desde 2003 trabaja como Profesor Colaborador en la UMH, impartiendo su docencia en el área de Ingeniería de Sistemas y Automática. Sus temas de investigación se centran en la Robótica Móvil, Robótica Cooperativa y en la Visión por Computador. Actualmente se encuentra investigando en técnicas de navegación de robots móviles basadas en apariencia utilizando las imágenes omnidireccionales. Además, estoy trabajando en técnicas para construcción de mapas de forma cooperativa mediante un equipo de robots.

Mónica Ballesta recibió el grado de Ingeniero Industrial en la Universidad Miguel Hernández de Elche (UMH) en 2005, siendo Premio Extraordinario de Fin de Carrera del curso 2005/2006. Durante el año 2006, ha trabajado como investigadora con el grupo ARVC del departamento de Ingeniería de Sistemas Industriales de la UMH en el proyecto "Robots cooperativos para la vigilancia e inspección de edificios e instalaciones industriales". Desde abril de 2007 es becaria FPI de la Generalitat Valenciana y desarrolla su investigación dentro del mismo grupo ARVC, en temas de robótica y tratamiento de imágenes. Es estudiante de doctorado dentro del programa "Automática y Robótica" en la UMH.

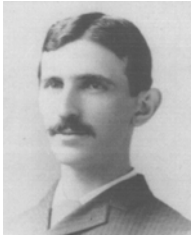
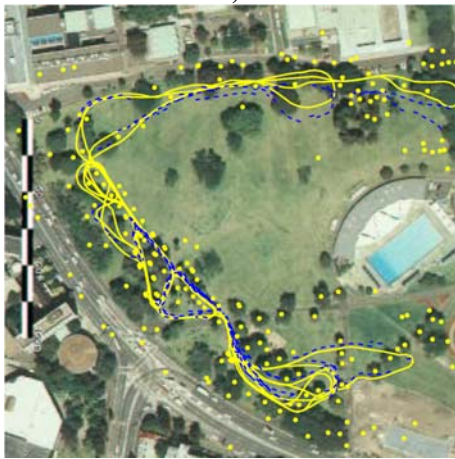


Fig. 2. (a) Mapa con odometría del robot (b) Trayectoria corregida



a)



b)

Fig. 1. Mapas de ocupación y de landmarks

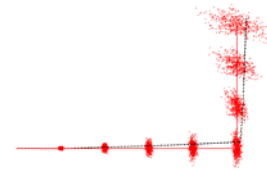


Fig. 3. Generación de partículas según el modelo de movimiento

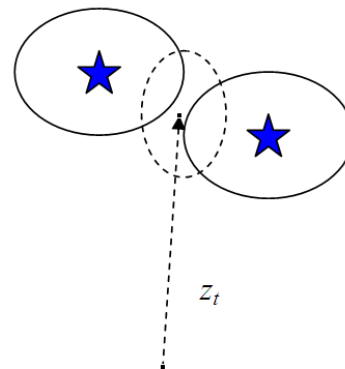
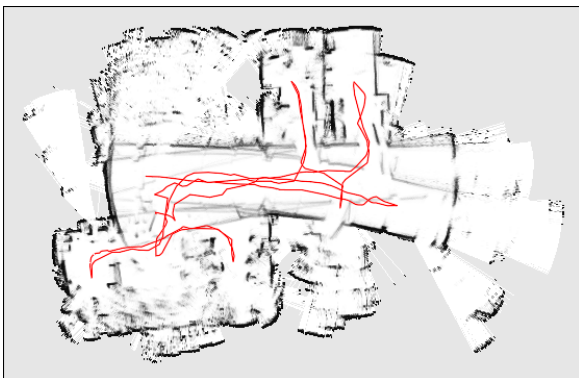


Fig. 4. El problema de la asociación de datos



a)

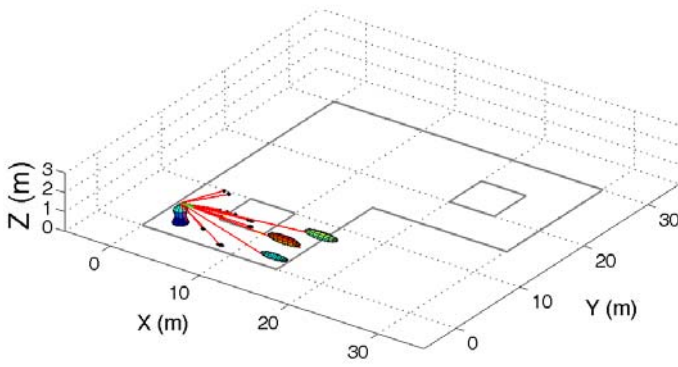


Fig. 5. Entorno de simulación utilizado

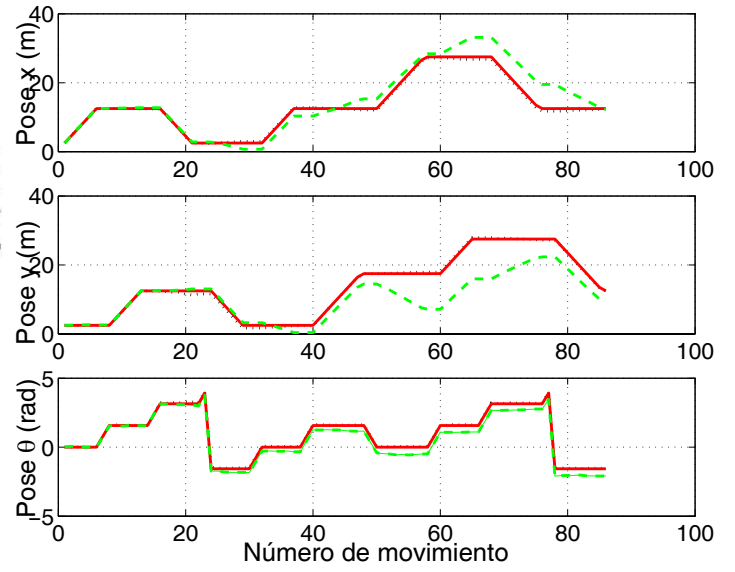


Fig. 7. Pose del robot durante el movimiento

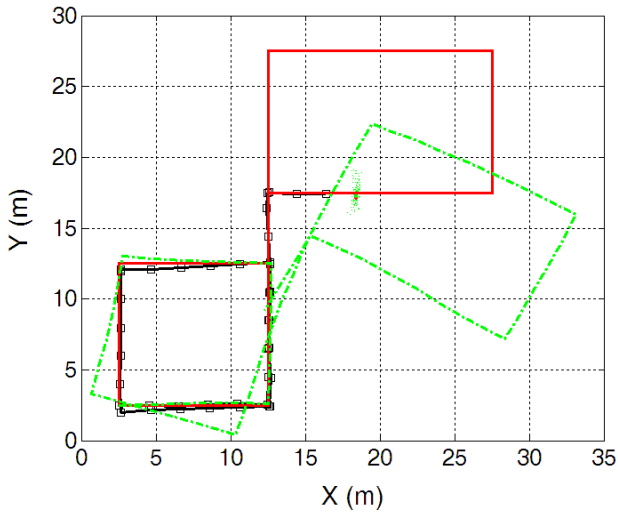
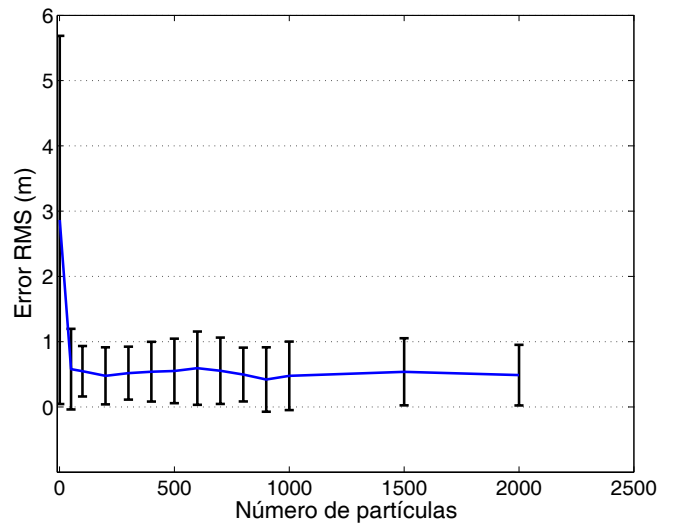


Fig. 6. Camino real (línea continua) y odometría (discontinua)



a)

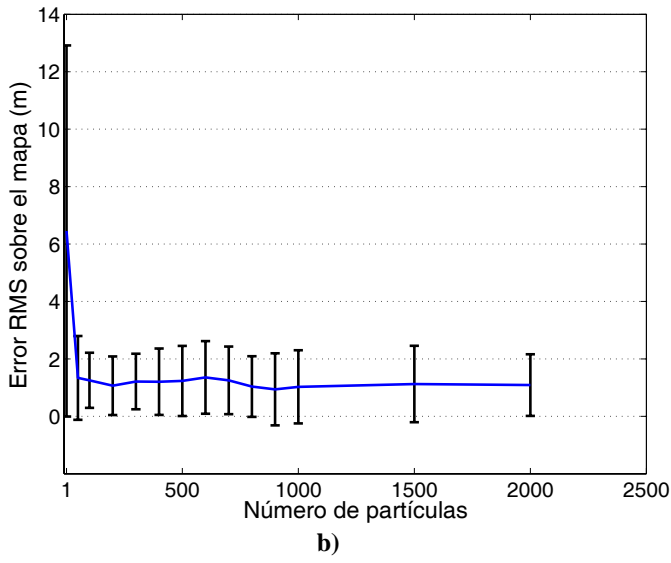


Fig. 8. Fig. 8(a) error RMS posición. Fig. 8(b) error RMS mapa

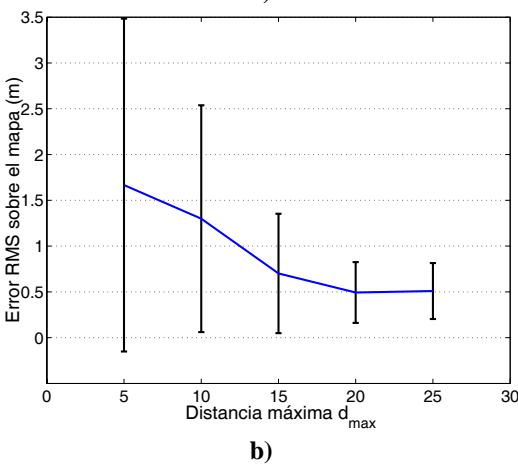
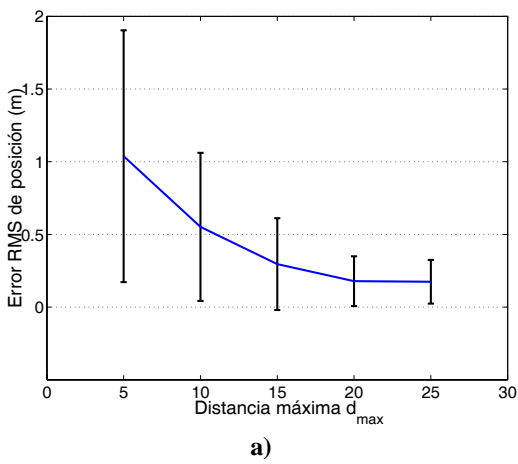


Fig. 9. (a)error RMS posición y (b) Error RMS mapa al variar la distancia d_{max}