

# Localización topológica basada en visión para robots móviles

Francisco Martín, Vicente Matellán, Pablo Barrera y Jose María Cañas  
 Grupo de Robótica, DITTE, Universidad Rey Juan Carlos  
 {fmartin,vmo,barrera,jmplaza}@gsyc.es

## Resumen

Por localización se entienden las técnicas que se encargan de solucionar el problema de determinar la posición de un robot en su entorno usando sus propios sensores. En este trabajo se presentará una técnica de localización que pretende aportar una nueva manera de localizar un robot equipado con una cámara en un entorno topológicamente estructurado mediante técnicas probabilísticas. Se mostrará su aplicación en entornos tan diferentes como espacios de oficinas o el campo de la Robocup<sup>1</sup>, y se comprobará como esta técnica es capaz de adaptarse a las tareas que el robot ha de realizar en estos entornos.

## 1. Introducción

La localización de un robot en su entorno es uno de los problemas clásicos de la robótica móvil. Es el punto de partida de otros problemas como son la interacción con el entorno, y sobre todo, la navegación. Para la mayor parte de las tareas que un robot móvil tiene que realizar depende de su posición en el mismo. En este trabajo pretendemos dar una solución de este problema para dos tareas muy distintas.

El primer caso es el de la navegación en interiores, concretamente en entorno de oficina. En estos entornos existen una gran cantidad de tareas que un robot puede realizar: llevar el correo de un despacho a otro, teleasistencia o tareas de seguridad. Un robot debe conocer su posición para trazar rutas desde su posición a su destino.

Otro escenario totalmente distinto es la competición de la Robocup. En esta competición, en particular en la categoría de las 4 patas<sup>2</sup> (fig 1), equipos de robots autónomos compiten entre sí en un juego similar al fútbol. Esta competición lo que pretende es presentar un escenario donde múltiples campos de la robótica pueden ser evaluados entre diferentes grupos de investigación a nivel mundial. En este caso, un robot debe saber su posición en el campo para saber qué hacer cuando

<sup>1</sup>[www.robocup.org](http://www.robocup.org)

<sup>2</sup><http://www.tzi.de/4legged/>



Figura 1: Categoría 4-legged. RoboCup 2006 (Osaka, Japón)



Figura 2: Sony Aibo ERS-7

le llega la pelota, irse a una posición determinada cuando no la tiene, o simplemente, enviársela a un compañero para que éste pueda tomar decisiones más “inteligentes”.

El mecanismo de localización que se propone en este artículo resuelve la localización en entornos tan diferentes como los expuestos anteriormente. Usa un enfoque probabilístico basado en Procesos de Markov Parcialmente Observables (POMDP). Divide el entorno donde el robot se ha de localizar topológicamente atendiendo a las características del entorno y/o las necesidades del robot. Se calcula una densidad de probabilidad del conjunto de estados en los que dividimos el entorno.

La plataforma en la que hemos probado esta técnica es el robot Sony Aibo ERS-7. Este robot tiene aspecto de perro (figura 2), y sus principales características son la locomoción realizada con sus 4 patas, y la sensorización basada en su cámara con resolución de 416x320 píxeles.

El problema de la localización en entornos de oficina ha sido ampliamente estudiado [20], pero la mayor parte de ellos [18] han sido diseñados pa-

ra robot equipados con ruedas, que cuentan con una información odométrica muy precisa e información sensorial en  $360^\circ$ . Estos métodos no son aplicables a robots con patas. El tipo de información recibida de los sensores de sonar o láser es más fácil de procesar que los de una cámara, que es el principal sensor en nuestro trabajo.

Otra diferencia clave con respecto a los trabajos anteriores en entornos de interior y sobre todo en el campo de la Robocup es su enfoque **topológico**. Un representación topológica del entorno, a diferencia de la representación métrica donde la posición es relativa a unos ejes de coordenadas, divide el entorno en zonas con similares características. Estas zonas no tienen por qué tener las mismas dimensiones. La división topológica es hecha dependiendo de las necesidades derivadas de las tareas de los robots, y mantiene entre cada zona una relación de vecindad. Existen varios trabajos en este sentido, como es [15], pero con robot con ruedas. La idea principal que queremos transmitir es que hay multitud de situaciones en las que una localización tan fina como la métrica no es necesaria, ya que hay multitud de posiciones de un entorno en las que el robot realizaría las mismas acciones con la información de localización. No es tan necesaria tanta precisión.

Técnicas probabilísticas markovianas se han usado en [2], [5] y [14], pero, otra vez, siempre con robots con ruedas y sensores de radar o sonar. En [9] se usa una aproximación markoviana basada en visión, pero la información extraída de las imágenes son histogramas y características invariantes a la escala (SIFT) [13] que son calculadas a partir de un enorme conjunto de imágenes tomadas del entorno. Nuestra aproximación no necesita tomar fotos con anterioridad, solo un mapa 2D del entorno.

En el ámbito de la liga de 4 patas de la Robocup, los métodos de localización usados pueden apreciarse en la tabla 1. La más usada es el método de Monte Carlo [19][6][16][17][21][4]. Este método hace un muestreo en el espacio de estados del robot y mantiene una población de *partículas* que van evolucionando y agrupándose en las posiciones donde cree que el robot se encuentra. En [22] se compara Monte Carlo con el filtro de Kalman [8], que es un filtro adaptativo considera óptimo y que admite no linealidad y de esta manera permite abordar el problema de la localización con estas características. El filtro de Kalman Extendido (EKF) ha sido usado como único método en puntuales trabajos [11][12] en la Robocup, pero aunque el método es potente y se ha probado con éxito en otros ámbitos [3][10], en este caso no es suficientemente flexible para adaptarse a entornos

tan ruidosos. El método difuso [1][7] es una variación del markoviano, pero dividiendo el terreno de juego en un grid donde cada celda almacena una posibilidad difusa tanto de la probabilidad de la celda, como del estado más probable y su incertidumbre. Este último método intenta solucionar el problema del coste computacional de los métodos markovianos puros, pero a diferencia del nuestro, queda limitado por lo estricto de la división del terreno de juego, que en nuestro método puede no ser regular en forma de grid, y que en cada casilla no soporta la posibilidad de ser multimodal. Todos estos métodos al final son un método métrico. La alternativa propuesta es la única que pretende solucionar este problema desde el punto de vista topológico mostrando que es lo suficiente potente como realizar las funciones del robot correctamente.

Método	Nº equipos
Monte Carlo	10
EKF	2
Monte Carlo + EKF	2
Método difuso	1
Triangulación	2

Cuadro 1: Métodos de localización utilizados en la edición de la RoboCup 2004

En la sección 2 mostraremos los fundamentos de la localización markoviana que aplicaremos en la sección 3 para entornos de oficina y en 4 para la competición de la Robocup, donde profundizaremos en sus características particulares. En la sección 5 analizaremos los experimentos llevados a cabo para validar el método. Por último, en la sección 6 presentaremos las conclusiones derivadas de este trabajo.

## 2. Procesos ocultos de Markov parcialmente observables (POMDPs)

Los POMDPs se caracterizan por modelar procesos estocásticos donde se usa una densidad de probabilidad  $Bel$  sobre todos los posibles estados  $s \in S = \{s_1, s_2, \dots\}$  en los que el robot puede encontrarse.  $Bel_t(S = s)$  representa la creencia de estar en el estado  $s$  en el instante  $t$ . Al inicio del proceso de localización, dependiendo del conocimiento *a priori* de la posición del robot,  $Bel_0(S)$  puede estar uniformemente distribuido

$$Bel_0(s_i) = \frac{1}{|S|} \quad (1)$$

o centrado en el estado donde la posición inicial  $j$  se

conoce,

$$Bel_0(s_i) = 0 + \delta, \forall i \neq j, \delta \ll 0,01 \quad (2)$$

$$Bel_0(s_j) = 1 - \delta, \delta \ll 0,01 \quad (3)$$

La actualización de  $Bel$  se divide en dos pasos atómicos. El primero es la *fase de movimiento* que se aplica cuando el robot realiza un movimiento o completa una acción. Se utiliza la información de la acción para modificar su creencia sobre su estado. La segunda es la *fase de observación*, donde la creencia se actualiza en función de la información extraída de las entradas sensoriales del robot. Este proceso se realiza cuando se obtiene la información necesaria para cada paso y no tiene porque ser ejecutada de manera alternativa.

### 2.1. Fase de movimiento

Las acciones de un robot son modeladas como un modelo de acción  $p(s'|s, a)$ , que es la probabilidad de alcanzar el estado  $s'$  si la acción  $a$  es ejecutada en el estado  $s$ . Para obtener la creencia del conjunto de estados  $Bel_t(S')$ , se aplica una actualización bayesiana. Con cada acción se aplica:

$$Bel_t(s') = \sum_{s \in S} p(s'|s, a) \cdot Bel_{t-1}(s), \forall s' \in S \quad (4)$$

### 2.2. Fase de observación

Para calcular la creencia actualizada con la información sensorial  $Bel_t(S')$ , usamos el modelo de observación  $p(o|s')$ , que representa la probabilidad de estar en el estado  $s'$  habiendo obtenido la observación  $o$ .

$$Bel_t(s') = p(o|s') \cdot Bel_{t-1}(s), \forall s, s' \in S \quad (5)$$

## 3. Aplicación a entornos de interiores

Esta aplicación pretende detectar la posición de un robot móvil en un entorno de oficinas para que pueda navegar por los pasillos de un despacho a otro. Usará como principal sensor las imágenes de la cámara, y utilizará también el sensor de infrarrojos que tiene situado al lado de la cámara en el morro del robot. La información que obtendrá de la imagen de la cámara serán elementos comunes de este entorno, como son las puertas, las luces del techo y los elementos que tiene a ambos lados el robot.

Los elementos que se deben definir en cada entorno de aplicación son:

1. El mapa del entorno y como se traduce a un conjunto de estados.
2. El conjunto de acciones que el robot puede llevar a cabo y el modelo de acción asociado a los estados  $p(s'|s, a)$ .
3. El conjunto de observaciones que el robot percibe del entorno, y su modelo de observación asociado a los estados  $p(o|s)$ .

### 3.1. El conjunto de estados

A todas las posibles posiciones y orientaciones en las que el robot se puede encontrar las denominamos como “estados”. Los estados son definidos sobre un entorno de oficina (figura 3) formado por pasillos (representados como círculos) y habitaciones (representados como cuadrados). No tienen ninguna relación métrica, esto es, un nodo que presente una porción de pasillo con similares características puede tener una longitud mucho mayor que otra porción de pasillo mucho menor. Recordemos que la división es topológica y depende de las observaciones tomadas desde las posiciones del nodo y de las diferentes acciones que un robot puede tomar en una zona del entorno.

En la figura 3 se observa los pasos necesarios para dividir un entorno de oficina. Primero se identifican los nodos como partes del mapa donde el robot percibe idéntica información sensorial y las decisiones adecuadas en cada momento son también idénticas. Como el robot usa la información referentes a los elementos que tiene a los lados (pasillo, puerta o pared), como veremos en la sección 3.3, la división en nodos es realizada teniendo este criterio. De este modo, una porción de pasillo sin puertas a los lados en un mismo nodo, pero si hay una porción de pasillo que tiene una puerta en uno de los dos lados, lo consideramos un nodo diferente.

Finalmente, una vez obtenidos los nodos, como observamos en la figura 3, se obtienen los estados en los que el robot se puede encontrar. Una manera simple y homogénea es dividir cada nodo en cuatro orientaciones, ya que un robot ha de realizar diferentes acciones dentro de un nodo dependiendo si está orientado hacia la pared, una parte del pasillo o la otra.

### 3.2. Modelo de movimiento

Las acciones que el robot puede tomar en cada posición son simples, avanzar hasta el siguiente nodo, girar 90° hacia la izquierda o girar 90° a la derecha. Estos giros son en lazo cerrado de manera que el robot está obteniendo información de sus sensores, tanto de infrarrojos como, sobre todo,

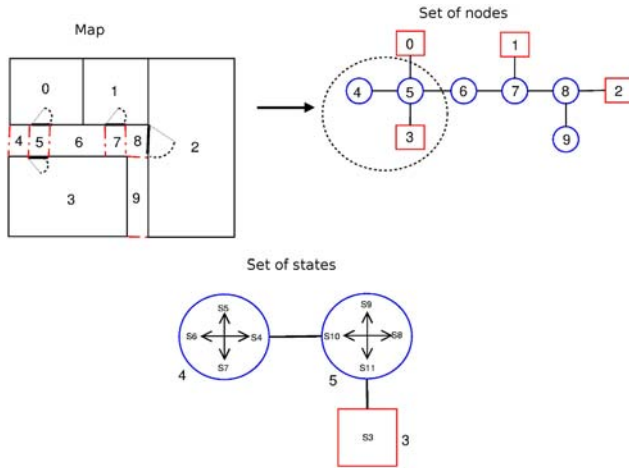


Figura 3: Formación de los estados a partir de la descomposición de los nodos del entorno

de la cámara, para detectar cuando ha cambiado de estado. Por ejemplo, si el robot está orientado hacia el fondo del pasillo y a ambos lados tiene pared, y decide realizar la acción de avanzar, el robot avanzará hacia adelante hasta que en alguno de los lados deje de encontrar pared.

El principio probabilístico de este método nos permite asociar a cada acción la probabilidad de éxito en la realización de la misma. En la tabla 2 queda definida esta probabilidad. Para el caso de los giros a ambos lados, la probabilidad de que el robot no realice el giro correcto y permanezca en el mismo estado (N) es 0.15. La probabilidad de que gire  $90^\circ$  correctamente (G) es 0.70, La probabilidad de que gire hasta el estado situado a  $180^\circ$  (GG) es 0.15, y la probabilidad de que gire hasta el estado situado a  $270^\circ$  (GGG) es cercana a 0. De igual manera, para la acción de avanzar hasta el siguiente estado, la probabilidad de quedarse en el mismo estado (N) es 0.20, de realizar correctamente la acción (A) es 0.6, y de avanzar más allá del estado al que se quiere llegar (AA y AAA) es 0.15 y 0.05 respectivamente.

Esto permite que el sistema sea robusto a colisiones, deslizamientos, errores en la detección del cambio de estado y cualquier eventualidad que pueda suceder en el mundo real donde el robot se ha de desenvolver. Los valores anteriormente mostrados son fruto de una experimentación y observación de la precisión de los movimientos y en la detección de cambios de estado.

El modelo de movimiento  $p(s'|s, a)$  queda finalmente definido con el conjunto de estados (que contiene información de vecindad) y la tabla 2. Una matriz  $n \times n$ , siendo  $n$  es número de estados, se genera *a priori* para cada acción. Estas matri-

GIzq	N: 0.15	G: 0.70	GG: 0.15	GGG: 0.0
GDer	N: 0.15	G: 0.70	GG: 0.15	GGG: 0.0
Avan	N: 0.20	A: 0.6	AA: 0.15	AAA: 0.05

Cuadro 2: Incertidumbre en la ejecución de la acción.

ces representan para cada acción  $a$ ,  $p(s'|s)$ , y son usadas cuando el robot realiza una acción para el cálculo del *Bel* actualizado.

### 3.3. Modelo de observación

Los elementos naturales con los que nos encontramos en este entorno son las paredes, las puertas y las luces del techo. Éstas últimas son útiles ya que pueden dar idea de profundidad en un pasillo. Si se evalúa un estado orientado hacia los fondos del pasillo y se detecta una elevada cantidad de luces, indica que el final del pasillo está lejos, y si se detectan pocas significa que el robot se encuentra cerca del final del pasillo.

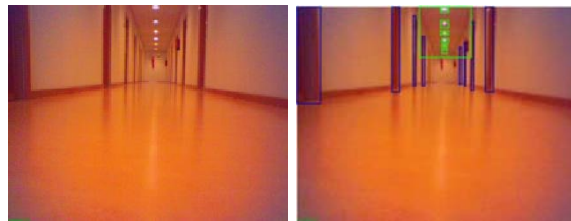


Figura 4: Detección de 6 luces y 8 puertas en un estado del pasillo alejada del final del mismo.

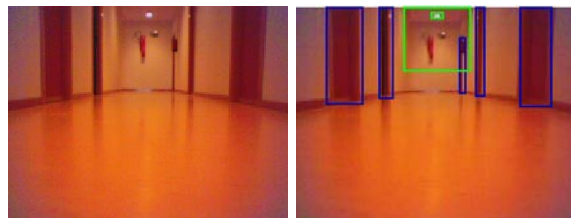


Figura 5: Detección de 1 luz y 5 puertas en un estado del pasillo cercana a la pared.

Las matrices de observación creadas para cada estado en función de la observación que debería tener  $p(o|s)$  soportan que haya errores en la extracción de información de las imágenes. Por ejemplo, si un robot en una posición debería detectar 6 luces, no debería agruparse la probabilidad sólo en 6 luces (como en la figura 6), sino que debería tenerse en cuenta error en la observación (figura 6).

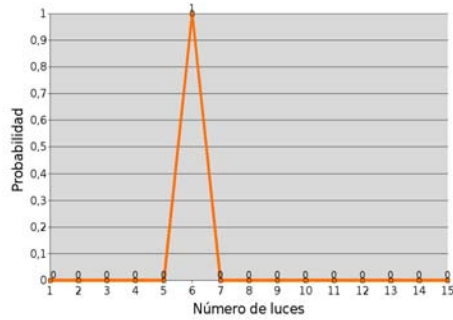


Figura 6:  $p(o|s_x)$  sin tener en cuenta errores en la observación si a priori es conocido que  $o = 6$  para  $s = x$ .

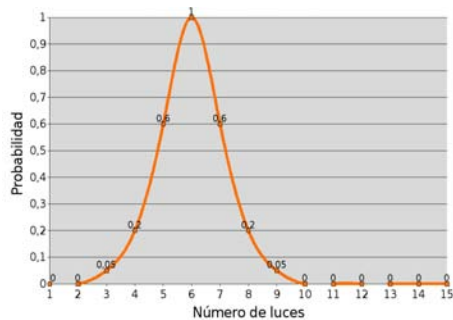


Figura 7:  $p(o|s)$  teniendo en cuenta errores en la observación si a priori es conocido que  $o = 6$  para  $s = x$ .

## 4. Aplicación a la Robocup

El entorno del campo de la Robocup es muy diferente al presentado en la sección 3, y las necesidades de localización son igualmente diferentes. El método propuesto en este artículo es capaz de adaptarse a este entorno, como mostraremos en esta sección.

El campo tiene unas dimensiones fijas de  $5400 \times 3600$  mm, y tiene una serie de marcas visuales que el robot puede usar para localizarse (figura 8). El objetivo de los robots es el de marcar goles en la portería del equipo contrario. El comportamiento de los robots es diferente dependiendo del sitio del terreno de juego donde se encuentren, y el rol que estén adoptando en cada momento. No solo han de realizar todas sus tareas, incluida la de localizarse, sino que lo han de hacer con unos tiempos de computación razonables para que el robot cumpla con su labor de una manera adecuada.

### 4.1. El conjunto de estados

Debido a la flexibilidad del método de localización mostrado aquí, la división de entorno del robot en estados se puede hacer de múltiples maneras. Podemos dividirlo en un grid regular donde



Figura 8: Campo de juego de la categoría 4-legged de la RoboCup

todos las celdas tienen las mismas dimensiones y los mismos estados para representar la orientación del robot (figura 9). Esto satisface las necesidades de poder estimar métricamente la posición del robot, que puede necesitar alguna arquitectura de comportamientos en particular. La estimación es semi directa, y el número de estados responde a un compromiso entre la precisión necesitada y los recursos computacionales disponibles.

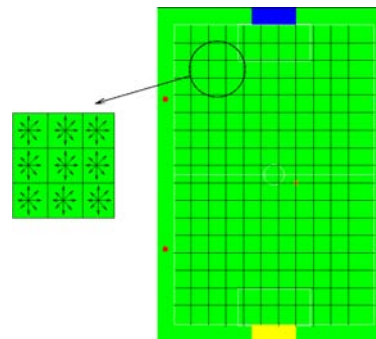


Figura 9: Campo dividido en nodos de la misma dimensión. Cada nodo tiene 8 estados separados de forma uniforme

Otra posibilidad es configurar el espacio de estados dependiendo de las acciones que el robot tenga que hacer en cada zona del campo (figura 10). Por ejemplo, un robot en tareas defensivas en la mitad de su propio campo no necesita saber nada más que si está en la derecha o izquierda del campo, y una orientación aproximada para lanzar la pelota hacia el campo contrario. Esta decisión necesita menos precisión de localización que posiciones cerca de la portería contraria por parte de los delanteros. La localización se orienta hacia el uso que de ella se va a hacer.

### 4.2. Modelo de movimiento

Al usar una aproximación tipo grid, el modelo de movimiento usado está basado en la odometría. Se realiza un *difuminado* de la probabilidad de cada estado a los cercanos topológicamente, atendiendo a la odometría recibida.

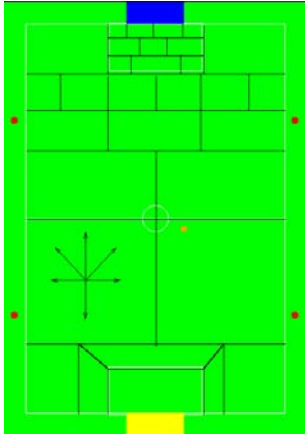


Figura 10: Campo dividido en nodos con distintas dimensiones. Cada uno tiene un número diferente de estados, que corresponden a orientaciones arbitrariamente diseñadas para cada nodo

En la figura 11 se puede ver la evolución del grid de probabilidad atendiendo a la información odométrica. Se observa como se difumina a los estados cercanos, para tener en cuenta errores entre el movimiento real y el percibido por el robot.

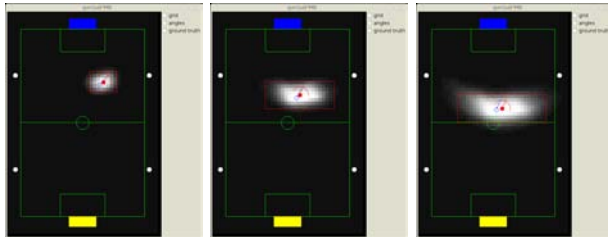


Figura 11: Modelo de movimiento basado en odometría. Se observa la influencia de los estados más probables en sus vecinos.

#### 4.3. Modelo de observación

La información que el robot usa para localizarse es la posición relativa al robot de las marcas visuales que existen en el campo. Estas marcas son las dos porterías y las cuatro balizas. El sistema ha de ser robusto a errores en estas observaciones, ya que el robot está realizando un movimiento continuo, y las medidas pueden no ser exactas.

Internamente el robot almacena tres valores para cada marca visual  $\langle \rho, \Theta, anchor \rangle$ , donde  $\rho$  es la distancia a la marca,  $\Theta$  es el ángulo respecto al eje del robot, y  $anchor$  es un valor que nos indica la confianza de esta información. Si el  $anchor$  es elevado, la información se considera correcta y reciente; si es bajo, la información no es válida. Este valor de  $anchor$  disminuye con el tiempo desde que la información de la marca visual es refrescada, y con la odometría.

Para hacer el modelo de observación tolerante a errores dentro de nuestro algoritmo, hemos asignado *a priori* la probabilidad de, estando en el estado  $s$ , encontrarse con la observación  $o$ . En la figura 12 se muestra una observación de una baliza en cierto estado. En ese estado, las probabilidades de encontrar esa baliza están dibujadas en azul. La  $p(o|s)$  depende del ángulo con que se percibe la baliza. Si está en la zona oscura es 1, si está en la clara, es 0,6, ya que puede haber habido un ligero error en la observación. Si está en el azul más claro es 0,3, por haberse producido un error considerable, que hay que tener en cuenta. Si está en otra región se considera improbable obtener esa observación en ese estado.

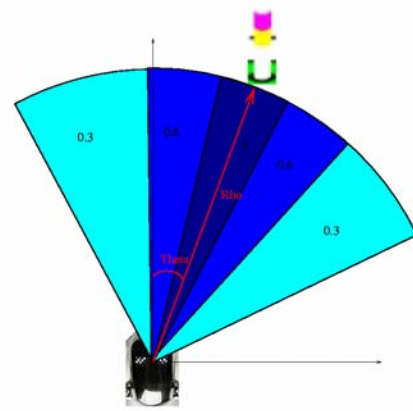


Figura 12: modelo de observación  $p(o|s)$ .

## 5. Experimentos

Para medir el método en entorno de interiores, se comparó el estado real del robot con la estimación más probable del robot. En el experimento mostrado en la figura 13 el robot recorre un entorno real de oficinas del punto 1 al 2. Se ha medido la distancia en transiciones de estados (pasos) desde la estimación del robot hasta su posición real, y los resultados se muestran en 14.

Los experimentos sobre el campo de la Robocup consistieron, al no disponer de groundtruth, en colocar al robot en posiciones aleatorias del terreno de juego y hacer que se dirigiera al centro haciendo uso únicamente de la información de localización, y que permaneciese allí. Los resultados después de varias pruebas son que la media del error en  $X$  es 297mm, y en  $Y$  es 207mm. La desviación estándar de cada uno es 45mm y 130mm respectivamente.

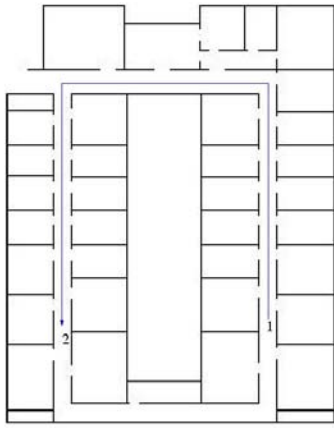


Figura 13: Recorrido del robot en el experimento.

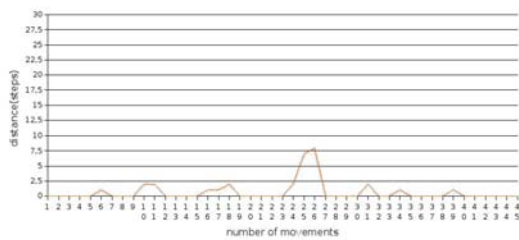


Figura 14: Error en la estimación de la posición.

## 6. Conclusiones

El enfoque topológico ha resultado ser adecuado en las dos aplicaciones que hemos mostrado en este trabajo. Estas aplicaciones han sido abordadas, sobre todo la segunda, con un enfoque métrico. El enfoque topológico ha permitido almacenar la vecindad entre los estados que en la realidad son adyacentes. Asimismo, ha permitido poder configurar la granularidad de la división del entorno en estos estados, y no siempre esta división puede ser regular. La división puede atender a las acciones que un robot puede tomar en cada parte del juego. Hemos mostrado cómo dividir el entorno en zonas de distinta dimensión donde el robot realizaría acciones idénticas.

Las pruebas realizadas muestran que el enfoque es correcto y funcional en las aplicaciones tomadas en cuenta. Futuras pruebas con groundtruth y un estudio detallado del comportamiento de los robots, en el caso de la Robocup, aportará unos resultados sólidos con los que dar los siguientes pasos. Estos pasos irían encaminados a una posible configuración del espacio de estados dinámicamente en función de los roles de los jugadores, precisión necesitada en cada momento y recursos computacionales del robot.

## Agradecimientos

Este trabajo ha sido parcialmente financiado por el Ministerio de Ciencia y Tecnología, en el proyecto ACRAE: DPI2004-07993-C03-01 y la Comunidad de Madrid en el proyecto RoboCity 2030: S-0505/DPI/0176.

## Referencias

- [1] P. Buschka, A. Saffiotti, and Z. Wasik. Fuzzy landmark-based localization for a legged robot. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 1205–1210, Takamatsu, Japan, 2000.
- [2] Anthony R. Cassandra, Leslie Pack Kaelbling, and James A. Kurien. Acting under uncertainty: Discrete bayesian models for mobile robot navigation. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1996.
- [3] C. Estrada, J.Ñeira, and J.D. Tardós. Hierarchical slam: real-time accurate mapping of large environments. *IEEE Transactions on Robotics*, 21(4):588–596, 2005.
- [4] Dieter Fox, W. Burgard, H.Kruppa, and Sebastian Thrun. A monte carlo algorithm for multi-robot localization. Technical Report CMU-CS-99-120, Computer Science Department, Carnegie Mellon University, Pittsburgh, PA, 1999.
- [5] Franck Gechter, Vincent Thomas, and François Charpillet. Robot localization by stochastic vision based device. In *The 5th World Multi-Conference on Systemics, Cybernetics and Informatics - SCI 2001 ? The 7th International Conference on Information Systems Analysis and Synthesis - ISAS 2001, Orlando, FL, USA*, Jul 2001.
- [6] Pablo Guerrero and Javier Ruiz del Solar. Auto-localización de un robot móvil aibo mediante el método de monte carlo. *Anales del Instituto de Ingenieros de Chile*, 115(3):91–102, 2003.
- [7] D. Herrero-Pérez, H. Martínez-Barberá, and A. Saffiotti. Fuzzy self-localization using natural features in the four-legged league. In D.Ñardi, M. Riedmiller, and C. Sammut, editors, *RoboCup 2004: Robot Soccer World Cup VIII*, LNAI. Springer-Verlag, Berlin, DE, 2004.
- [8] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME—Journal of Basic Engineering*, 82(Series D):35–45, 1960.

- [9] Jana Kosecká and Fayin li. Vision based topological markov localization. In *Proceedings of the 2004 IEEE International Conference on Robotics and Automation*, Barcelona (Spain), April 2004.
- [10] Evgeni Kyriy and Martin Buehler. Three-state extended kalman filter for mobile robot localization. Technical report, Carnegie Mellon, 2002.
- [11] R. Lastra, P. Vallejos, , and J Ruiz-del Solar. Integrated self-localization and ball tracking in the four-legged robot soccer league. In *Proceeding of the 1st IEEE Latin American Robotics Symposium LARS 2004*, México City (México), Oct 2004.
- [12] R. Lastra, P. Vallejos, , and J Ruiz-del Solar. Self-localization and ball tracking for the robocup 4-legged league. In *Proceeding of the 2nd IEEE Latin American Robotics Symposium LARS 2005*, Sao Luis (Brazil), Sept 2005.
- [13] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, 2004.
- [14] María E. López, Luis Miguel Bergasa, and M.S.Escudero. Visually augmented POMDP for indoor robot navigation. *Applied Informatics*, pages 183–187, 2003.
- [15] Dandapani Radhakrishnan and Illah Nourbakhsh. Topological localization by training a vision-based transition detector. In *Proceedings of IROS 1999*, volume 1, pages 468 – 473, October 1999.
- [16] Thomas Röfer and Matthias Jüngel. Fast and robust edge-based localization in the sony four-legged robot league. In *RoboCup*, pages 262–273, 2003.
- [17] Thomas Röfer and Matthias Jüngel. Vision-based fast and reactive monte-carlo localization. In *ICRA*, pages 856–861, 2003.
- [18] Reid Simmons and Sven Koenig. Probabilistic navigation in partially observable environments. In *Proceedings of the 1995 International Joint Conference on Artificial Intelligence*, pages 1080–1087, Montreal (Canada), July 1995.
- [19] Mohan Sridharan, Gregory Kuhlmann, and Peter Stone. Practical vision-based monte carlo localization on a legged robot. In *IEEE International Conference on Robotics and Automation*, April 2005.
- [20] S.Thrun. Robotic mapping: A survey. *Technical Report CMU-CS-02-111*, 2002.
- [21] J. Wolf, W. Burgard, and H. Burkhardt. Robust vision-based localization by combining an image retrieval system with monte carlo localization. *IEEE Transactions on Robotics*, 21(2):208–216, 2005.
- [22] David C.K. Yuen and Bruce A. MacDonald. A comparison between extended kalman filtering and sequential monte carlo technique for simultaneous localisation and map-building. In *Proc. Australian Conference on Robotics and Automation*, Auckland, New Zealand, November 2002.