

Estimación bayesiana de características en robots móviles mediante muestreo de la densidad a posteriori

Domingo Gallardo, Francisco Escolano, Ramón Rizo,
Otto Colomina, Miguel Angel Cazorla
domingo,sco,rizo,otto,miguel@dccia.ua.es
Grupo VGIA: Visión, Gráficos e Inteligencia Artificial
Departamento de Ciencia de la Computación e Inteligencia Artificial
Universidad de Alicante

Resumen

La obtención de características estables y robustas del entorno en un robot móvil es un elemento clave para la extracción autónoma de mapas del entorno, la localización en el mismo o la navegación de una localización a otra.

Proponemos en este trabajo la estimación y el seguimiento de características topológicas de forma robusta y estable mediante la utilización de técnicas de muestreo de la función de densidad a posteriori, a partir de la formulación de un modelo probabilístico del sensor y del movimiento del robot. Estas técnicas de muestreo permiten representar una densidad arbitraria de un espacio paramétrico continuo, frente al filtro de Kalman, que necesariamente se aplica a distribuciones gaussianas, o las redes bayesianas temporales, en las que es necesario definir un conjunto discreto de estados.

Por último, comprobamos los resultados sobre lecturas de sonar obtenidas en entornos simulados, en distintas condiciones de ruido y variabilidad.

Palabras claves: estimación y seguimiento temporal, seguimiento no gaussiano, muestreo de densidad de probabilidad, características geométricas estables y robustas, robots móviles.

1 Introducción

La obtención de características estables y robustas del entorno en un robot móvil es el paso previo para una posterior extracción autónoma de mapas del entorno, localización en el mismo o navegación de una localización a otra [12]. Las características se suelen extraer del entorno a partir de sensores de ultrasonidos, elementos estándar de captación de información en la robótica móvil. Estos sensores se utilizan normalmente en anillos de 12 o 24 transductores distribuidos de forma equidistante alrededor del robot.

La popularidad del sonar se debe fundamentalmente a que se trata de un sensor de muy bajo coste, su consumo es muy pequeño y el procesamiento que se realiza a su señal es muy sencillo, haciendo posible una respuesta casi inmediata y permitiendo una reacción rápida del robot.

Sin embargo, frente a estas ventajas, el sonar tiene como principal inconveniente el elevado nivel de ruido e incertidumbre presente en sus lecturas, debido a diversos factores [7].

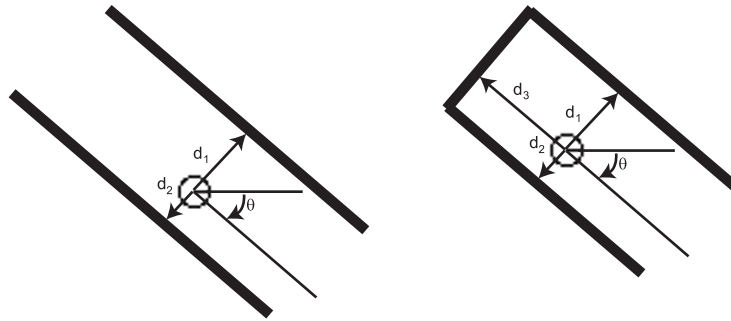


Figura 1: Características topológicas usadas en el trabajo: pasillos y finales de pasillo.

Existe una amplia colección de trabajos en los que se proponen métodos para filtrar los resultados obtenidos por estos sensores y obtener características geométricas elementales [3, 1, 10, 13, 14]. Sin embargo las características obtenidas en todos ellos son muy locales, como aristas, esquinas o segmentos, y esta propia localidad hace que sean muy sensibles al ruido, poco robustas y poco estables. Esto último se acentúa en entornos dinámicos y variables, del tipo en los que suelen evolucionar estos robots.

Frente a este planteamiento local aparece como alternativa la utilización de enfoques bayesianos en los que se va acumulando en el tiempo la evidencia asociada a la característica. Aunque este enfoque ha sido aplicado con éxito desde los primeros momentos de investigación en el campo de la robótica móvil [4], es ahora cuando se ha despertado un gran interés en su uso para aplicaciones de mayor alcance [16, 2, 5, 17, 11]. En esta línea, nuestro trabajo se basa en características topológicas de más alto nivel, como son pasillos o finales de pasillo (ver figura 1). Estas características se extraen de los datos del sonar utilizando un enfoque bayesiano que presentamos en la siguiente sección. Esta metodología es genérica y permite ser aplicada a otro tipo de características topológicas, como conexiones entre pasillos, etc.

En concreto, el núcleo de la propuesta se basa en la utilización de técnicas iterativas de muestreo para estimar la densidad a posteriori de estas características topológicas. Estas técnicas permiten representar la densidad de probabilidad de forma explícita, mediante un conjunto (m_1, m_2, \dots, m_N) de muestras extraídas de dicha función de probabilidad. Para obtener estas muestras, utilizaremos el *algoritmo CONDENSATION* [9], una versión del *algoritmo de rechazo* [15] aplicada al muestreo de distribuciones a posteriori. Presentamos estos algoritmos en la sección 3. En la sección 4 formulamos en detalle el modelo dinámico del sistema y el modelo de verosimilitud. Por último, en el apartado 5 analizamos los resultados de aplicar este enfoque a la detección de características obtenidas por el sonar en entornos con ruidosos y con abundante variabilidad, comprobando que las características son estables y robustas.

2 Técnicas para la estimación bayesiana temporal

Un problema estándar en reconocimiento estadístico de patrones es encontrar un objeto parametrizado con el vector x con una distribución de probabilidad a priori $p(x)$ utilizando unos



Figura 2: Estimación bayesiana temporal.

datos z [6]. Aplicando este modelo genérico al problema que tratamos en el artículo, el vector x representa la parametrización de una característica a buscar en el entorno (pasillo, final de pasillo, etc.) y los datos z representan las lecturas del sonar.

Para estimar x a partir de z es necesario conocer la distribución condicional $p(z|x)$ que mide la *verosimilitud* de que unas lecturas z correspondan a un hipotético objeto x . La densidad a posteriori $p(x|z)$ representa todo el conocimiento sobre x que puede ser deducido a partir de los datos obtenidos. Esta densidad se calcula utilizando la regla de bayes

$$p(x|z) = \alpha p(z|x)p(x) \quad (1)$$

donde α es una constante de normalización independiente de x .

2.1 Estimación bayesiana temporal

La versión dinámica del planteamiento bayesiano supone que tenemos una secuencia de $t - 1$ lecturas en el tiempo $\mathbf{z}_{t-1} = (z_1, z_2, \dots, z_{t-1})$ que han sido producidas por una evolución del objeto x , cuya historia denotamos por $\mathbf{x}_{t-1} = (x_1, x_2, \dots, x_{t-1})$. La dinámica de dicho objeto ha sido producida por una secuencia de acciones también conocidas $\mathbf{a}_{t-1} = (a_1, a_2, \dots, a_{t-1})$ (en nuestro caso cada acción a será un valor de velocidad lineal v y angular ω del robot móvil). La figura 2.1 representa gráficamente el proceso.

Para realizar una estimación temporal debemos formular el modelo dinámico del sistema y el modelo de medida. En los siguientes subapartados presentamos las restricciones generales que deben cumplir ambos y en la sección 4 detallaremos su expresión para el caso del seguimiento de características del entorno a partir de datos del sonar.

2.1.1 Modelo dinámico

El modelo dinámico proporciona una expresión de la probabilidad de un determinado estado a partir de una secuencia de estados y acciones previas

$$p(x_t | \mathbf{x}_{t-1}, \mathbf{a}_{t-1}). \quad (2)$$

Asumimos que este modelo dinámico cumple la condición de Markov

$$p(x_t | \mathbf{x}_{t-1}, \mathbf{a}_{t-1}) = p(x_t | x_{t-1}, a_{t-1}), \quad (3)$$

esto es que el nuevo estado depende únicamente del estado y de la acción anterior. Asumimos también que este modelo es estacionario, esto es que no depende del instante de tiempo, por lo que deberemos formular más adelante la expresión $p(x'|x, a)$.

2.1.2 Medidas

Suponemos que las observaciones z_t son con respecto al proceso dinámico, lo que se expresa probabilísticamente como

$$p(\mathbf{z}_{t-1}, x_t | \mathbf{x}_{t-1}, \mathbf{a}_{t-1}) = p(x_t | \mathbf{x}_{t-1}, \mathbf{a}_{t-1}) p(\mathbf{z}_{t-1} | \mathbf{x}_{t-1}, \mathbf{a}_{t-1}). \quad (4)$$

También suponemos que son independientes entre si,

$$p(\mathbf{z}_{t-1} | \mathbf{x}_{t-1}, \mathbf{a}_{t-1}) = \prod_{i=1}^{t-1} p(z_i | x_i, a_i), \quad (5)$$

y que la observación únicamente depende del estado, y no de la acción realizada

$$p(z_i | x_i, a_i) = p(z_i | x_i). \quad (6)$$

Suponemos, por último, que este modelo también es estacionario y que no depende del instante de tiempo, quedando una densidad de probabilidad $p(z|x)$ que define la verosimilitud de unas lecturas dado una hipotética característica del entorno y que detallaremos más adelante.

2.1.3 Propagación de la probabilidad a posteriori

Asumiendo las consideraciones anteriores, la propagación de la probabilidad a posteriori en los modelos de propagación bayesiana temporal viene dada por

$$p(x_t | \mathbf{z}_t, \mathbf{a}_t) = \alpha_t p(z_t | x_t) p(x_t | \mathbf{z}_{t-1}, \mathbf{a}_{t-1}), \quad (7)$$

siendo

$$p(x_t | \mathbf{z}_{t-1}, \mathbf{a}_{t-1}) = \int_{x_{t-1}} p(x_t | x_{t-1}, a_{t-1}) p(x_{t-1} | \mathbf{z}_{t-1}, \mathbf{a}_{t-1}). \quad (8)$$

Estas expresiones son el equivalente temporal a la regla de Bayes y determina de forma iterativa la distribución de probabilidad a posteriori en función de la última lectura del sensor z_{t-1} , la última acción a_{t-1} , el modelo dinámico $p(x'|x, a)$, el modelo de sensor $p(z|x)$ y la probabilidad a posteriori efectiva anterior.

El problema de estimar esta densidad a posteriori se ha venido resolviendo usualmente mediante el filtro de Kalman o mediante las redes bayesianas temporales.

2.2 Filtro de Kalman

El filtro de Kalman se ha usado con gran éxito para realizar una estimación temporal en los casos en que x es un vector en un espacio de parámetros continuo y la verosimilitud $p(z|x)$ tiene una distribución gaussiana. Este filtro estima de forma iterativa la media y la varianza de las x en cada instante de tiempo y esto sirve para caracterizar perfectamente la distribución $p(x|z)$.

La aplicación del filtro de Kalman al seguimiento de características del entorno es problemática, debido fundamentalmente al problema del *aliasing del sensor* que provoca que distintas configuraciones del entorno x produzcan parecidas lecturas del sensor z . Esto hace que

la densidad $p(z|x)$ sea mucho más complicada que una gaussiana, siendo en muchas ocasiones multimodal. Los entornos dinámicos y el ruido en las lecturas tampoco son bien soportados por esta técnica.

2.3 Redes bayesianas temporales

Las redes bayesianas temporales se utilizan como una alternativa al filtro de Kalman, ya que permiten representar distribuciones de densidad arbitrarias. Para representar la distribución $p(x|z)$ se discretiza el espacio de parámetros X en un conjunto de estados y se utiliza la regla de bayes para propagar sus probabilidades.

Si llamamos s_i a cada uno de los estados que se definen en X , y suponemos que en el instante t se ha realizado la lectura z_t después de realizar la acción a_t , las probabilidades de los estados s_i se actualizan iterando la expresión

$$p(s_i) \leftarrow p(z_t | s_i) \sum_{s_j \in X} p(s_i | s_j, a_{t-1}) p(s_j) \quad (9)$$

para todos los $s \in X$. Después de actualizar las probabilidades asociadas a cada uno de los estados, éstas se normalizan para asegurar que

$$\sum p(s_i) = 1. \quad (10)$$

El problema fundamental de las redes bayesianas es la necesidad de discretizar el espacio de parámetros, lo que añade incertidumbre e introduce una complejidad espacial (y temporal al realizar la propagación) exponencial con el número de parámetros.

3 Estimación por muestreo de la densidad a posteriori

Resumiendo el problema planteado en la sección anterior, las redes bayesianas permiten estimar probabilidades con distribuciones de densidad arbitraria, a costa de discretizar el espacio de parámetros que definen las características que estamos siguiendo, mientras que el filtro de Kalman, por otra parte, permite tratar parámetros continuos, pero únicamente es capaz de representar distribuciones gaussianas.

La solución que proponemos consiste en realizar un *muestreo* de la distribución a posteriori mediante el algoritmo CONDENSATION [9]. Este muestreo proporciona un conjunto estable de estimaciones (m_1, m_2, \dots, m_2) de la característica a detectar X , junto con las probabilidades asociadas a cada muestra. Las muestras representan de forma fiable la probabilidad a posteriori $p(x_t | \mathbf{z}_t, \mathbf{a}_t)$.

3.1 Muestreo por rechazo

Supongamos una distribución de densidad de probabilidad

$$p(x) = cg(x)h(x) \quad (11)$$

Algoritmo MUESTREO POR RECHAZO

Salida: Valores (m_1, m_2, \dots, m_N) muestreados de la densidad de probabilidad $p(x) = cg(x)h(x)$

1. $i \leftarrow 1$
2. $u \leftarrow$ valor aleatorio de la distribución $U(0, 1)$
3. $y \leftarrow$ valor aleatorio de la distribución $h(x)$
4. Si $u \leq g(y)$ hacer $m_i = y$. En otro caso ir a 2.
5. $i \leftarrow i + 1$ y saltar a 2 hasta que $i = N$

Algoritmo MUESTREO POR RECHAZO MODIFICADO

Salida: Valores (m_1, m_2, \dots, m_N) y probabilidades asociadas $(\pi_1, \pi_2, \dots, \pi_N)$ muestreados de la densidad de probabilidad $p(x) = cg(x)h(x)$

1. $i \leftarrow 1$
2. $m_i \leftarrow$ valor aleatorio de la distribución $h(x)$
3. $\pi_i \leftarrow g(m_i)$
4. $i \leftarrow i + 1$ y saltar a 2 hasta que $i = N$

Tabla 1: Algoritmo de rechazo y su versión modificada para mejorar su eficiencia.

donde $g(x)$ puede ser calculada para cualquier valor x y es posible generar muestras de la distribución $h(x)$.

Para generar una población de N muestras (m_1, \dots, m_N) de la función $p(x)$ se puede utilizar el algoritmo de aceptación y rechazo (ver tabla 3.1).

En la primera versión del algoritmo se generan muestras y de la distribución $h(x)$. El valor devuelto por $g(y)$, para cada muestra de $h(x)$ determina la probabilidad de aceptar dicha muestra. Para hacer efectiva esa probabilidad se utiliza el número u generado por la distribución uniforme, de manera que la muestra se acepta cuando este número es menor que el valor $g(y)$.

Por ejemplo, supongamos que se genera el número $y = 1.5$ y $g(1.5) = 0.75$. Esto significa que la probabilidad de aceptar $y = 1.5$ como un valor aleatorio generado por $p(x)$ es 0.75. Se utiliza el valor aleatorio u para realizar esta aceptación.

Un problema muy importante del algoritmo de rechazo es que se necesitan demasiadas muestras cuando la función $g(x)$ nos da valores de probabilidad muy pequeños. La mejora planteada por la versión modificada del algoritmo (ver tabla 3.1) consiste en ir almacenando las probabilidades obtenidas por $g(x)$ junto con las muestras. De esta forma, como salida del algoritmo se obtienen N muestras (m_1, \dots, m_N) y sus probabilidades asociadas (π_1, \dots, π_N)

Este esquema es directamente aplicable a la estimación de la probabilidad $p(x_t|z_t)$. Si observamos la ecuación 7, vemos que $p(z_t|x_t)$ y $p(x_t|z_{t-1}, \mathbf{z}_{t-1})$ hacen el papel de $g(x)$ y $h(x)$ respectivamente. Esto conduce a formular el algoritmo CONDENSATION como se detalla en el siguiente apartado.

3.2 Algoritmo CONDENSATION

El algoritmo CONDENSATION proporciona una representación explícita de la densidad a posteriori mediante un conjunto de N muestras (m_1, \dots, m_N) y sus probabilidades asociadas (π_1, \dots, π_N) . Estas muestras se generan en cada instante de tiempo a partir de las muestras del instante anterior y de las nuevas mediciones del entorno (z_t) y las nuevas acciones recién ejecutadas por el robot (a_t) .

Inicialmente, el conjunto de muestras se distribuye uniformemente en el espacio de parámetros X . Posteriormente, en cada instante de tiempo, se actualizan las N muestras de la siguiente forma. Se escoge una muestra m_i con probabilidad π_i a partir del conjunto de muestras anteriores. Esta muestra es una muestra correcta de la distribución $p(x_{t-1}|z_{t-1}, \mathbf{a}_{t-1})$. A partir

Algoritmo CONDENSATION

Inicialización: Inicializar las aleatoriamente N muestras (m_1, m_2, \dots, m_N) y normalizar sus probabilidades asociadas mediante $\pi_i \leftarrow 1/N$.

Paso del algoritmo después de realizarse la lectura z_t y la acción a_t :

1. $i \leftarrow 1$
2. $m_i \leftarrow$ valor aleatorio generado a partir de (m_1, m_2, \dots, m_N)
y de las probabilidades $(\pi_1, \pi_2, \dots, \pi_N)$
3. $m_i \leftarrow$ valor generado de $p(x_t | x_{t-1} = m_i, a_t)$
4. $\pi_i \leftarrow p(z_t | x_t = m_i)$
5. $i \leftarrow i + 1$ y saltar a 2 hasta que $i = N$

Tabla 2: Algoritmo CONDENSATION.

de ella se realiza una proyección del modelo del sistema según la acción a_t realizada por el mismo, obteniendo la nueva muestra $m'_i = p(x_t | x_{t-1} = m_i, a_t)$. Por último, como probabilidad asociada a m_i se calcula su verosimilitud con las lecturas realizadas, mediante el modelo de probabilidad del sensor, $\pi_i = p(z_t | x_t = m_i)$.

4 Modelo de movimiento y de sensor

Para aplicar cualquier propagación temporal es necesario formular correctamente tanto el modelo de acción del sistema como el modelo del sensor.

4.1 Modelo dinámico del robot

Esta sección describe las ecuaciones de movimiento fundamentales de un robot *synchro-drive*, modelando su posición $x(t)$, $y(t)$ y su orientación $\theta(t)$ mediante las siguientes ecuaciones

$$\theta(t_n) = \theta(t_0) + \int_{t_0}^{t_n} \omega(t) dt \quad (12)$$

$$x(t_n) = x(t_0) + \int_{t_0}^{t_n} v(t) \cos \theta(t) dt \quad (13)$$

$$y(t_n) = y(t_0) + \int_{t_0}^{t_n} v(t) \sin \theta(t) dt \quad (14)$$

Las ecuaciones anteriores pueden simplificarse si se asume que el robot se mueve de forma discreta, con intervalos de control de tiempo Δt en los que la velocidad lineal es v_t y la angular ω_t . Esta suposición es asumible cuando los intervalos de muestreo de las velocidades y los datos del sonar no es demasiado grande en comparación con la velocidad del móvil. En nuestro caso, se han realizado todas las pruebas con el móvil simulado moviéndose a 25 cm/s y se han tomado lecturas cada 0.25 segundos.

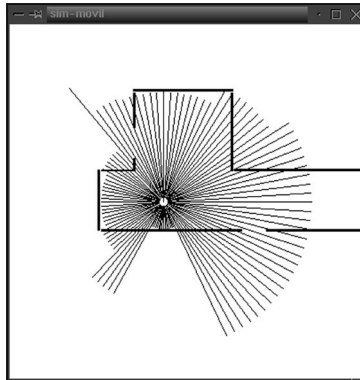


Figura 3: Simulación de las lecturas de un anillo de sensores de ultrasonidos.

La generación de muestras de $p(x_{t+1}|x_t = x, a_t = a)$ es directa a partir del modelo dinámico anterior. Sólo hay que aplicar el cambio de posición y orientación del robot sufrido tras sus movimientos v_t y ω_t a los parámetros x que definen la características. Además introducimos en la distribución un ruido gaussiano para modelar la incertidumbre en la posición real del robot. Este ruido gaussiano es fundamental a la hora de dotar al modelo de robustez.

4.2 Modelo del sensor

El modelo del sensor nos proporciona la verosimilitud de unas lecturas z obtenidas en una hipotética configuración del entorno x . Las lecturas z son distancias a los obstáculos más próximos en la dirección definida por los sensores de ultrasonidos distribuidos alrededor del robot. En nuestros experimentos hemos utilizado una configuración de 24 sensores simulados, con lo que $z = (l_1, l_2, \dots, l_{24})$.

El enfoque para el cálculo de la verosimilitud ha sido similar a las propuestas de *reconocimiento a partir de síntesis* [8]. Se ha construido un modelo realista del sensor de ultrasonidos, capaz de simular correctamente la mayor parte de características de las lecturas reales, incluyendo las que han sido despreciadas hasta el momento como fuentes de error, como los múltiples rebotes. Estos errores no pueden modelarse correctamente suponiendo un modelo gaussiano del ruido.

Para implementar el modelo se ha realizado una modificación del clásico algoritmo de trazado de rayos, adaptándolo a superficies 2D y al cálculo de las distancias de alcance [7]. Un ejemplo del resultado del algoritmo de simulación puede verse en la figura 3.

Para calcular la verosimilitud $p(z|x)$ se simulan hipotéticas lecturas obtenidas por la característica en la configuración x y se comparan con las lecturas reales, utilizando alguna medida de distancia (ver 4).

Formalmente, la función de verosimilitud se expresa como

$$p(z = (l_1, l_2, \dots, l_{24})|x) = e^{-d(\mathbf{l}_r, \mathbf{l}_s)^2/\sigma^2}, \quad (15)$$

siendo \mathbf{l}_r las lecturas reales del sonar y \mathbf{l}_s las lecturas simuladas producidas por la característica x .

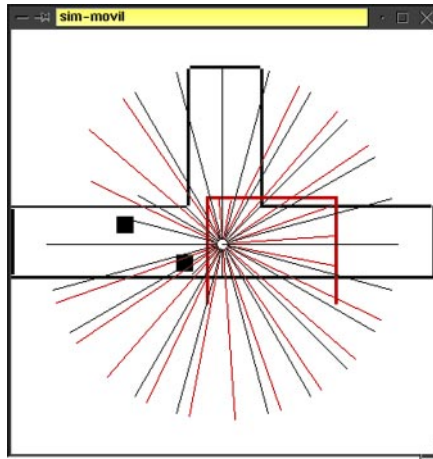


Figura 4: Lecturas reales (en negro) frente a lecturas producidas por la característica *final de pasillo* en una hipotética configuración (en rojo).

En los experimentos posteriores, la función d se ha implementado como una distancia euclídea entre las posiciones finales de las lecturas.

5 Resultados

Para comprobar la técnica propuesta se han realizado una serie de experimentos en los entornos simulados que aparecen en la figura 5. En todos los experimentos el robot se mueve evitando obstáculos a una velocidad de 25 cm/s y realiza una lectura de sensores cada 0.25 segundos. Cada vez que se realiza una lectura se ejecuta un paso del algoritmo CONDENSATION. La velocidad del algoritmo es aceptable, funcionando a 1/4 del tiempo real con $N = 300$ muestras en un procesador Pentium II. Futuras optimizaciones del código harán posible la ejecución del algoritmo en tiempo real.

En los siguientes apartados veremos ejemplos del funcionamiento del algoritmo en distintos instantes de tiempo. Dibujaremos las muestras generadas por el algoritmo sobre el entorno real, y cada muestra tendrá un tono de gris proporcional a su probabilidad.

5.1 Inicialización

En la figura 6 se puede observar el proceso de inicialización del conjunto de muestras de la característica *pasillo*. Recordemos que cuanto más oscura es la muestra mayor probabilidad asociada tiene. El número de muestras utilizadas es $N = 300$. El tiempo de una instantánea a otra es de 1 segundo.

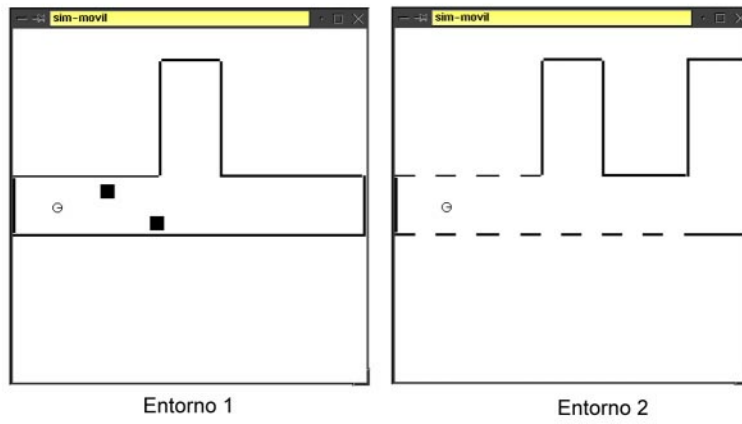


Figura 5: Algunos de los entornos de prueba en los que se han realizado los experimentos. El entorno 1 consiste en un pasillo con dos obstáculos y el 2 un pasillo con múltiples puertas.

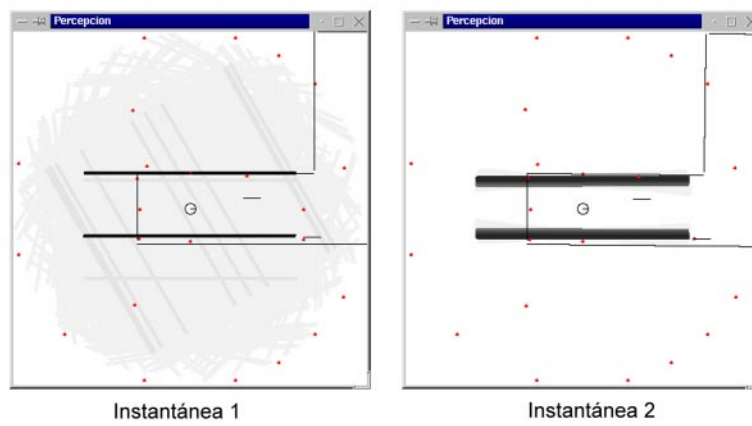


Figura 6: Inicialización de la característica *pasillo* en el entorno 1.

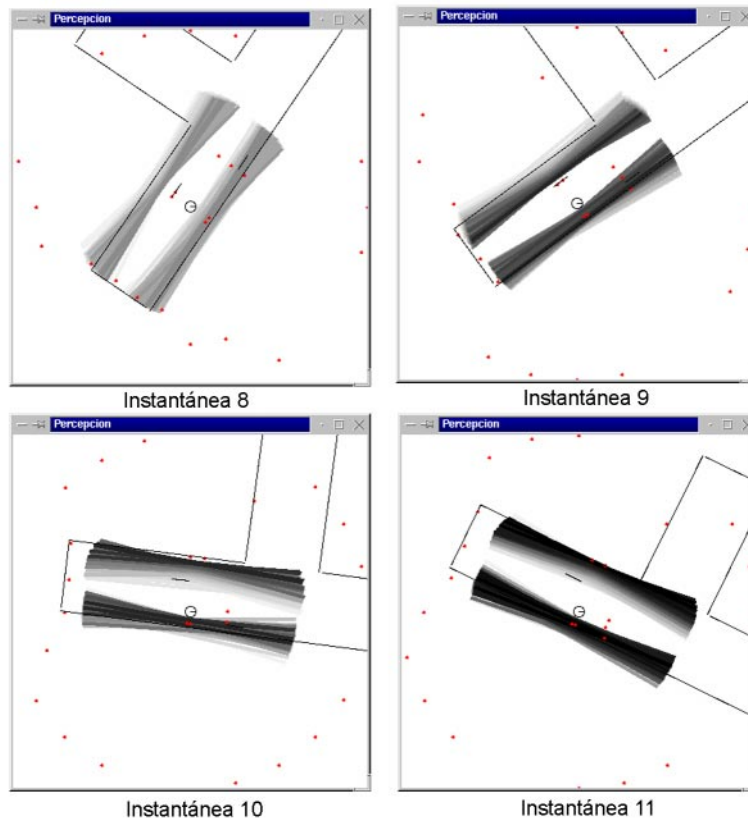


Figura 7: Seguimiento del pasillo moviéndose el robot en el entorno 1. El obstáculo puede verse como un segmento recto paralelo al pasillo.

5.2 Seguimiento de pasillos con obstáculos

En la figura 7 se puede ver la continuación de la situación anterior. Una vez centrado el conjunto de muestras alrededor del pasillo real todas las muestras bajan en verosimilitud al pasar el robot frente a un obstáculo (instantánea 8). En la instantánea 9, el ruido gaussiano del modelo de movimiento genera algunas muestras de pasillos más cercanos al obstáculo, pero la media de la distribución no cambia de forma sensible. En las instantáneas 10 y 11 el robot ha superado el obstáculo, vuelven a realizarse lecturas del pasillo real y la distribución se mueve otra vez hacia el pasillo real.

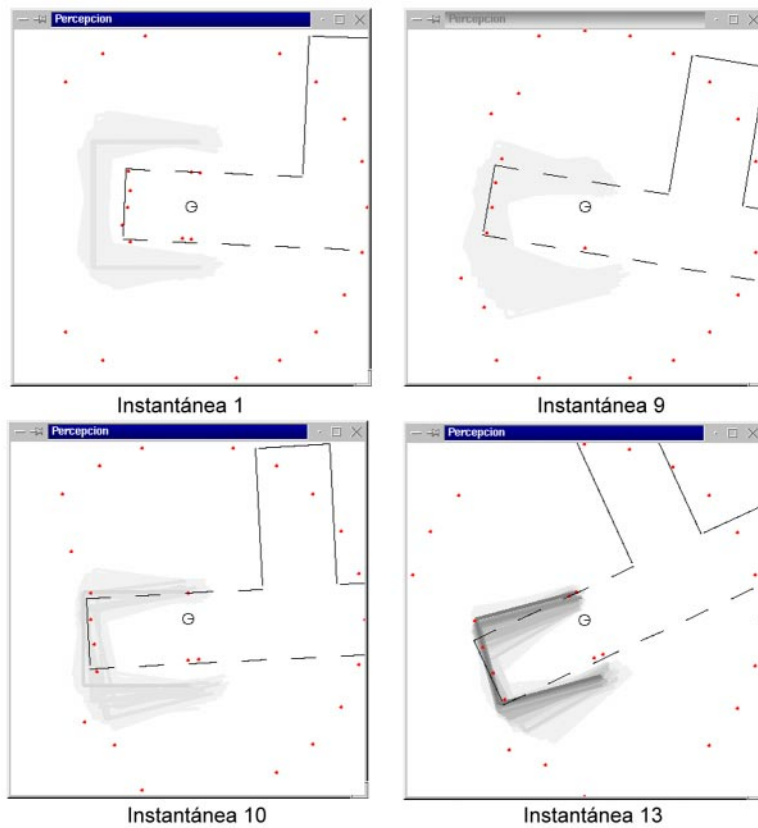


Figura 8: Seguimiento de finales de pasillo moviéndose el robot en el entorno 2.

5.3 Seguimiento de finales de pasillos

En la figura 8 se comprueba el funcionamiento del algoritmo siguiendo finales de pasillo en un entorno complicado como el número 2. La dificultad de este entorno se debe a que el robot no obtiene ninguna lectura del pasillo cuando pasa frente a las puertas. El número de muestras de este experimento es el mismo que el anterior ($N = 300$). En la instantánea 1 se ven las muestras ya inicializadas. De la instantánea 1 a la 9 el robot pasa frente a diversas puertas, con lo que las muestras han evolucionado según el modelo dinámico del robot, sin ser reforzadas por lecturas del sonar. Sin embargo, cuando el robot vuelve a detectar el pasillo vemos como en las instantáneas 10 y 13 el algoritmo vuelve a reforzar las muestras correctas.

6 Conclusiones

Se ha presentado la aplicación del algoritmo CONDENSATION al problema del seguimiento de características topológicas a partir de datos de sensores de ultrasonidos. Este algoritmo permite estimar una distribución a posteriori arbitraria en un espacio de parámetros continuo, resolviendo las deficiencias de los métodos empleados usualmente como son el filtro de Kalman o las redes bayesianas temporales.

Se han mostrado resultados experimentales realizados sobre entornos simulados en los que se comprueba la robustez y estabilidad de las características extraídas.

Referencias

- [1] Billur Barshan and Roman Kuc. Differentiating sonar reflections from corners and planes by employing an intelligent sensor. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-12(6):560–569, June 1990.
- [2] Hugh Durrant-Whyte Daniel Pagac, Eduardo M.Ñebot. An evidential approach to probabilistic map-building. In *Proceedings of the International Conference on Robotics and Automation*, Los Alamitos, CA, USA, May 1996. IEEE Computer Society Press.
- [3] M. Drumheller. Mobile robot localization using sonar. *IEEE Trans. Pattern Analysis and Machine Intelligence*, PAMI-9(2):325–332, 1987. See also: S. B. Thesis, Dept. of Mechanical Engineering, MIT, 1984 and MIT AI Lab Memo BZ6, Mobile Robot Localization Using Sonar.
- [4] Alberto Elfes. Sonar-based real-world mapping and navigation. *IEEE Journal of Robotics and Automation*, RA-3(3):249–265, June 1987.
- [5] D. Fox, W. Burgard, S. Thrun, and A.B. Cremers. Position estimation for mobile robots in dynamic environments. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI'98)*. AAAI Press, 1998.
- [6] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, 1990.
- [7] Domingo Gallardo, Otto Colomina, and Ramón Rizo. Simulación realista de sensores de ultrasonidos. Technical report, Departamento de Ciencia de Computación e Inteligencia Artificial. Universidad de Alicante, 1998.
- [8] Ulf Grenander. *General Pattern Theory: A Mathematical Study of Regular Structures*. Oxford University Press, 1993.
- [9] Michael Isard and Andrew Blake. Contour tracking by stochastic propagation of conditional density. In *Proc. Europea Conf. Computer Vision 1996 (AAAI'98)*, pages 343–356. Springer Verlag, 1996.
- [10] J.H. Ko, W.J. Kim, and M.J. Chung. A method of acoustic landmark extraction for mobile robot navigation. *IEEE Transactions of Robotics and Automation*, 12(3):478–485, 1996.

- [11] S. Koenig and R. Simmons. Xavier: a robot navigation architecture based on partially observable markov decision process models. In D. Kortenkamp, P. Bonasso, and R. Murphi, editors, *Artificial Intelligence and Mobile Robots*. AAAI Press, 1998.
- [12] D. Kortenkamp, P. Bonasso, and R. Murphi, editors. *Artificial Intelligence and Mobile Robots*. AAAI Press, 1998.
- [13] Ben J.A. Krose, Kai M. Compagner, and Franciscus C.A. Groen. Accurate estimation of environment parameters from ultrasonic data. *Robotics and Autonomous Systems*, 11:221–230, 1993.
- [14] Philip John McKerrow. Echolocation – from range to outline segments. *Robotics and Autonomous Systems*, 11:205–211, 1993.
- [15] W.H. Press, S.A. Teulosky, W.T. Vetterling, and B.P. Flannery. *Numerical Recipes in C*. Cambridge University Press, 1988.
- [16] A. M. Sabatini and O. Di Benedetto. Towards a robust methodology for mobile robot localization using sonar. In Edna Straub and Regina Spencer Sipple, editors, *Proceedings of the International Conference on Robotics and Automation. Volume 4*, pages 3142–3148, Los Alamitos, CA, USA, May 1994. IEEE Computer Society Press.
- [17] S. Thrun. Learnig metric-topological maps for indoor mobile robot navigation. *Artificial Intelligence*, 99(1):21–72, 1998.