

Construcción del estímulo *Obstáculo* sobre un mapa de rejilla para navegación de un robot en exteriores

Lía García-Pérez, Jose María Cañas(*), María C. García-Alegre
Instituto de Automática Industrial
(*)Universidad Rey Juan Carlos

Resumen

Este trabajo presenta una metodología de generación de mapas locales dinámicos a partir de la información de la posición del robot suministrada por un GPS con corrección diferencial y por una brújula, y de las distancias a los objetos obtenidas mediante un sensor de barrido láser. El entorno local se representa mediante un mapa de rejilla asignando a cada celda un valor proporcional al estado de ocupación de la misma. Este mapa local es fundamental para la extracción de estructuras perceptuales, elaboradas a partir de la fusión sensorial sobre lo que constituye una memoria espacio temporal de alcance medio. De esta forma la extracción de la percepción *Obstáculo* permitirá a los comportamientos de navegación desarrollar estrategias más eficaces al disponer de una ventana espacio temporal más amplia que la que obtenida de la percepción instantánea del espacio circundante. Finalmente, se presenta la estrategia diseñada para la clasificación de los obstáculos detectados en móviles y estáticos.

1. Introducción

Sin lugar a dudas la capacidad de movimiento constituye una de las competencias más importantes de un robot móvil [14] y una navegación segura debe integrar siempre la habilidad de evitar obstáculos. De ahí que la mayoría de los robots móviles dispongan de algún dispositivo sensorial para evitar las colisiones en su navegación en entornos parcialmente estructurados. A partir de la información que estos sensores proporcionan se elaboran desde respuestas reflejas de parada inmediata ante obstáculos muy próximos, hasta algoritmos complejos de planificación de caminos óptimos para rodear los obstáculos imprevistos evitando la colisión, con el objetivo global de optimizar el binomio energía-tiempo [1]. En todos los casos el comportamiento generado, se apoya en algún tipo de representación del entorno del robot y en las características físicas del mismo.

En aquellas aplicaciones de la robótica móvil como son la agricultura de precisión, el medio espacial, o la intervención civil en zonas de riesgo es imprescindible desarrollar mecanismos de percepción más complejos que garanticen una toma de decisión más precisa, resultando fundamental un mayor conocimiento del entorno[4]. En exteriores los sensores más adecuados a la tarea de evitar colisión con obstáculos imprevistos son los sensores de barrido láser [3], los radares de microondas y los sistemas de visión con cámaras CCD o láser [6].

Los métodos de representación del entorno más utilizados en robótica móvil se pueden agrupar en dos clases: los basados en rejillas de ocupación y los basados en la descripción de los bordes geométricos de los objetos [10]. En este trabajo se presenta una aproximación a la construcción de mapas en exteriores basada en rejillas de ocupación, al considerar que constituye el marco más adecuado para la fusión de información proveniente de distintos sensores [5]. Otra de las ventajas de los mapas de ocupación es la versatilidad y la flexibilidad que el modelo de celdas ofrece en la definición de distintos tipos de agrupamientos de las celdillas para la construcción de los estímulos requeridos por los comportamientos del robot móvil en una aplicación concreta.

El problema de Evitar Obstáculos en Exteriores

Frente a las aplicaciones típicas de un robot de oficina, en un entorno de exteriores las distancias son mayores, los vehículos más grandes, los sensores diferentes y fundamentalmente los actuadores son más lentos. Los actuadores hidráulicos y neumáticos, utilizados en este caso, poseen una potencia mayor pero también una gran inercia frente a los motores eléctricos utilizados en robots de interiores. De ahí que estrategias de navegación adecuadas a distancias cortas en interiores sean ineficientes al trasladarlas directamente a un entorno de exteriores [14].

En este trabajo se presenta una faceta de la problemática de la navegación en exteriores de un vehículo autónomo de tamaño medio. La poca maniobrabilidad del tractor hace que sea necesario dar un rodeo muy amplio para evitar obstáculos si no se quiere chocar con ellos. Es decir, el robot debe percibir la presencia del obstáculo a cierta distancia e iniciar la maniobra con suficiente anticipación [13]. Para ello conviene conocer el carácter estático o dinámico del obstáculo. Así, en muchas ocasiones si el obstáculo es móvil no será necesario iniciar la maniobra de rodeo porque éste se habrá desplazado de la trayectoria de choque antes de que el tractor llegue a su posición. Interesaría además conocer la dirección y velocidad del obstáculo para decidir si se rodea por la izquierda o por la derecha.

Sin embargo si sólo se dispone de información instantánea (últimos datos del sensor láser o valor actual de las celdas de ocupación) resulta imposible apreciar el carácter dinámico del obstáculo y las estrategias para evitarlo difícilmente serán óptimas. La estrategia más sencilla, dirigir el robot hacia la zona con más espacio libre ante la presencia de un obstáculo en su camino, fracasa si el obstáculo móvil aparece por la derecha del robot moviéndose hacia la izquierda. En este caso el comportamiento observable es la *persecución* del obstáculo.

Por todo lo anteriormente expuesto se considera fundamental el dotar al robot con la capacidad de discriminar entre objetos móviles o estáticos, a fin de generar las estrategias más adecuadas de navegación. De ahí que se proponga la existencia de una memoria espacio temporal que permita elaborar estos estímulos y caracterizar su estado de movimiento. Para ello se parte de un mapa de ocupación construido a partir de las medidas obtenidas mediante el sensor de barrido láser y los sensores de localización, inspirado en técnicas de procesado de imagen visual [7].

En la sección 2 se describe brevemente el vehículo sensorizado para navegación en exteriores, ROJO, que ha permitido la adquisición de la información sensorial requerida para la construcción del mapa local de rejilla. En el apartado 3 se describe el mapa de rejilla empleado, la extracción del estímulo *Obstáculo* y la estrategia seguida para la clasificación del mismo en estático o móvil. Los resultados experimentales, tanto en simulación como en experimentación en exteriores con el robot ROJO se presentan en la sección 4.

2. El robot ROJO

El robot ROJO es un tractor cortacésped comercial de bajo coste que ha sido adaptado en los talleres del IAI-CSIC para sustituir la conducción manual por una conducción automática a distancia. Para ello se han diseñado dos sistemas de actuación neumática que se acoplan a los ejes de control de la dirección y del freno del vehículo [12]. Para garantizar la autonomía con una única fuente de energía se ha incorporado al vehículo un compresor de aire y un tanque de almacenamiento del aire comprimido. De igual forma se han integrado en el mismo un conjunto de sensores que permiten la detección del estado interno del vehículo así como de su relación con los objetos del entorno exterior. El subsistema sensorial propioceptivo consta de dos sensores potenciómétricos que miden el giro de las ruedas y la posición del pedal de freno y un sensor de nivel de batería que indica la energía del robot. El subsistema de posicionamiento se compone de un DGPS de precisión media que proporciona la posición absoluta del vehículo, un sensor odométrico y una brújula digital. Además de los sensores de localización, se han integrado en ROJO, un láser de barrido que proporciona 361 medidas con una resolución de 0,5(*grados*) desde 0 hasta 180, dos inclinómetros y un parachoques, para la percepción del estado robot-entorno. El sistema de proceso es un ordenador PC convencional con comunicación vía Radio-Ethernet con la red inalámbrica local del IAI-CSIC

El robot ROJO dispone de un programa Servidor, denominado Server, residente en el ordenador de a bordo, dedicado a la adquisición de las señales provenientes de los distintos sensores y a la etapa de procesamiento inicial de las mismas. Server también se encarga de la ejecución del ciclo de control de los dos controladores de la dirección y el freno. Dentro del paradigma Cliente/Servidor cualquier otro programa puede conectarse a Server como cliente y solicitar el envío de las señales sensoriales o bien enviar las consignas requeridas para el control del sistema electroneumático de actuación. Este marco facilita la experimentación de distintos algoritmos de control en una misma sesión de navegación, con el único requerimiento de mantener el protocolo de comunicación de mensajes previamente definido.

La resolución de la problemática de evitar obstáculos ha llevado a plantear la generación de un mapa local con un alcance espacio temporal adecuado a la aplicación que se pretende abordar. De esta forma se puede anticipar el cálculo de una estrategia de rodeo que optimice la trayectoria global al objetivo evitando la colisión con obstáculos imprevistos. Este mapa local permitirá la detección y la clasificación de los obstáculos en función de su tamaño y de su estado de movimiento estableciéndose así un segundo nivel de optimización en la decisión sobre la estrategia de navegación a realizar. La elaboración del estímulo *Obstáculo* tiene como soporte una memoria del entorno o mapa local que se representa en este caso mediante una rejilla de ocupación. El valor de ocupación de cada una de sus celdas se calcula mediante un algoritmo de votación [2] sobre las tres últimas medidas de la distancia robot-objeto correspondiente a esa celda. El mapa de rejilla se construye a partir de la posición del robot (x_r, y_r, θ_r) obtenida mediante un receptor diferencial DGPS y una brújula digital, y de las medidas de distancia que proporciona el sensor de barrido láser.

3. El estímulo *Obstáculo*

La estructura de memoria requerida para anticipar y optimizar el cálculo de trayectorias ante un *Obstáculo* y para clasificar su dinamismo se ha implementado me-

dante un mapa local de rejilla de 20×20 metros dividido en celdas de 20×20 (cm.) que se mueve solidario con el robot. Esta representación permite el cálculo de diferentes estímulos clave para la activación de comportamientos de navegación como son: *Ir_a_punto*, *Rodear_muro*, *Aproximarse_a_seto* [8]. En este caso se trata de elaborar a partir del mapa local de rejilla el estímulo *Obstáculo* desencadenante del comportamiento *Evitar_Obstáculo*. En función de las relaciones espacio temporales de los distintos obstáculos que pueden estar presentes en la escena se selecciona la estrategia de navegación más adecuada. El estímulo *Obstáculo* se define como un grupo de celdas con **estado ocupado** conectadas entre sí y pertenecientes al **Área_de_Movimiento** próxima al robot. En esta aproximación cada obstáculo se caracteriza por: posición, tamaño y velocidad.

El **Área_de_Movimiento** se define como el área semicircular centrada en la posición del robot, (x_r, y_r, θ_r) y de radio $r_m = 5,5$ (m.), figura 1, al ser el área relevante en la colisión con obstáculos, (3.1).

$$\begin{aligned} \text{Area Movimiento} = & \text{celdas } (x_{c,i}, y_{c,i}) \in \text{Mapa Local} / \\ & (x_{c,i} - x_r)^2 + (y_{c,i} - y_r)^2 < r_m^2 \text{ y} \\ & \arctan\left(\frac{y_{c,i} - y_r}{x_{c,i} - x_r}\right) \in (\theta_r - (\pi/2), \theta_r + (\pi/2)) \end{aligned} \quad (3.1)$$

3.1. Algoritmo de segmentación

El primer paso en la construcción del estímulo *Obstáculo* a partir del mapa local es la segmentación [11], es decir, el cálculo desde la rejilla de ocupación de las celdas que conforman cada obstáculo. Se trata de un algoritmo de barrido de imagen en un único paso, que procede en la siguiente forma:

1. Construcción de la **lista de celdas obstáculo**. Ésta es una lista que almacena los índices de las celdas del mapa que pertenecen al *Área_de_Movimiento* y tienen estado **ocupado**.
2. Agrupamiento de las celdas de la **lista de celdas obstáculo**. Para ello se recorre la lista anterior ordenadamente. Se asigna la celda 1 al *Obstáculo* 1 y se buscan todas las que están conectadas con ella, figura 1, asignándolas al *Obstáculo* 1. Este proceso se realiza de forma recursiva con todas y cada una de las celdas incorporadas hasta que no haya más celdas conectadas o se llegue al final de la **lista de celdas obstáculo**. Para evitar la repetición del proceso sobre celdas ya contadas, se mantiene otra lista, la **lista de celdas contadas** que incluye las celdas ya asignadas a un *Obstáculo*. Finalizado el proceso sobre el *Obstáculo* 1, se continúa con las celdas de la lista de celdas obstáculo que no pertenecen a la lista de celdas contadas.

Al final de este paso, dada la rejilla de ocupación en el instante t , $\text{mapa}(t)$, se obtiene una colección de objetos $O_i(t)$. Cada uno de ellos definido por las celdillas $c_{O_i(t)}(n)$ que lo conforman, sin caracterizar aún su tamaño y su estado de movimiento.

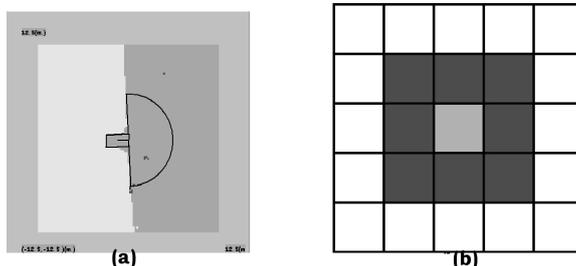


Figura 1: a)Área de Movimiento. b)Todas las celdas poseen una conectividad 8.

3.2. Posición del obstáculo

Las coordenadas de un obstáculo corresponden al valor medio de las coordenadas de los puntos centrales de cada una de las celdas que lo forman, (3.2).

$$x_i = \frac{c_{1,x} + c_{2,x} + \dots + c_{n,x}}{n}, y_i = \frac{c_{1,y} + c_{2,y} + \dots + c_{n,y}}{n} \quad (3.2)$$

Sin embargo pensando en la función de este estímulo en los comportamientos de navegación segura, es más relevante disponer de un valor aproximado del centro del mismo, que perseguir su posición exacta mediante algoritmos geométricos más complejos. En la línea de razonamiento propuesta por Zadeh en [15], a partir de las coordenadas numéricas del punto centro del *Obstáculo* obtenemos la percepción de su localización en el *Área de Movimiento* del robot. Para ello dividimos el *Área de Movimiento* en tres zonas y asociamos a cada una de ellas una función de pertenencia, figura 2. La percepción “fuzzy” de la posición del obstáculo viene dada por el grado de pertenencia del obstáculo a cada una de las tres funciones, (3.3).

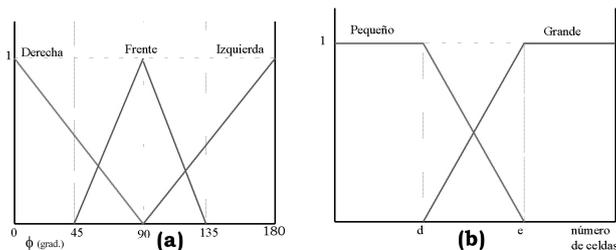


Figura 2: Funciones de pertenencia borrosas de la posición (a) y del tamaño (b).

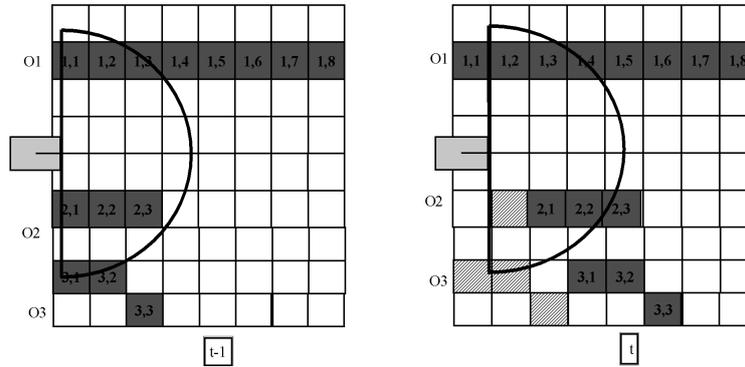


Figura 3: Se muestran las celdas ocupadas (gris) y las celdas ocupadas en t-1 pero vacías en t (rayadas). Donde O1 es un obstáculo estático, O2 móvil y O3 móvil y rápido.

$$\begin{aligned}
 \mu_{derecha}(\phi) &= \begin{cases} 1 - \frac{2 \times \phi}{\pi} & \text{si } \phi < \frac{\pi}{2}, \\ 0 & \text{si } \phi > \frac{\pi}{2} \end{cases} \\
 \mu_{frente}(\phi) &= \begin{cases} \frac{\phi \times 4}{\pi} - 1 & \text{si } \frac{\pi}{4} < \phi < \frac{\pi}{2} \\ 3 - \frac{\phi \times 4}{\pi} & \text{si } \frac{\pi}{2} < \phi < \frac{\pi \times 3}{4} \\ 0 & \text{si } \phi < \frac{\pi}{4} \text{ y } \phi > \frac{\pi \times 3}{4} \end{cases} \\
 \mu_{izquierda}(\phi) &= \begin{cases} 0 & \text{si } \phi < \frac{\pi}{2} \\ \frac{\phi \times 2}{\pi} - 1 & \text{si } \phi > \frac{\pi}{2} \end{cases} \quad (3.3)
 \end{aligned}$$

3.3. Tamaño del obstáculo

En una primera aproximación se define el tamaño en función del número de celdas que forman el obstáculo. Con base en los datos experimentales se agrupan en dos conjuntos “fuzzy”: grande y pequeño, figura 2 y ecuaciones (3.4). En ambas ecuaciones los parámetros d y e se corresponden con los límites descritos en la figura 2.

$$\begin{aligned}
 \mu_{pequeño}(nceldas) &= \begin{cases} 1 & \text{si } nceldas < d \\ \frac{nceldas - e}{d - e} & \text{si } d < nceldas < e \\ 0 & \text{si } nceldas > e \end{cases} \\
 \mu_{grande}(nceldas) &= \begin{cases} 0 & \text{si } nceldas < d \\ \frac{nceldas - d}{e - d} & \text{si } d < nceldas < e \\ 1 & \text{si } nceldas > e \end{cases} \quad (3.4)
 \end{aligned}$$

3.4. Velocidad del obstáculo

En la implementación actual la caracterización de la velocidad refleja únicamente si el obstáculo es estático o se está moviendo. Para ello se analiza el cambio en el estado de las celdillas que forman los obstáculos en tiempos consecutivos, figura 3.

Sean $A(t-1), A(t), O_i(t-1)$ y $O_j(t)$ el área de movimiento y los obstáculos detectados en los instantes consecutivos $t-1$ y t . Para cada uno de los obstáculos en $t-1$ se buscan las celdas que en t siguen perteneciendo al *Área_de_Movimiento*, (3.5), y se comprueba su estado de ocupación, calculando la intersección con cada obstáculo en t , (3.5). Dado un obstáculo $O_i(t-1)$ pueden ocurrir tres casos diferentes: que todas las celdas permanezcan ocupadas, que sólo un subconjunto lo esté o bien que ninguna celda continúe en ese estado.

$$\begin{aligned} O'_i(t) &= O_i(t-1) \cap A(t) \\ I_{i,j} &= O'_i(t) \cap O_j(t) \end{aligned} \quad (3.5)$$

En el primer caso el obstáculo $O_i(t-1)$ se corresponde con algún obstáculo $O_j(t)$, es decir, $\exists j / I_{i,j} = O_i(t-1)$. Entonces las celdas que en $t-1$ estaban ocupadas lo siguen estando en t , por lo que el obstáculo se considerará estático. Si dos obstáculos en t tuviesen celdas asociadas a un único obstáculo en $t-1$, se le identifica con aquel con el que posea una intersección mayor.

En el segundo caso no todas las celdas de $O_i(t-1)$ permanecen ocupadas, por lo que el obstáculo se considerará móvil. Además se asocia al obstáculo $O_j(t)$ que contenga mayor número de sus celdas ocupadas. En desarrollos posteriores se podría estimar la velocidad teniendo en cuenta qué celdas se han vaciado y hacia dónde se ha desplazado el centro de masas del obstáculo.

En el tercer caso la intersección de $O_i(t-1)$ con todos los obstáculos en t , $O_j(t)$ es vacía, es decir, $\forall j I_{i,j} = \emptyset$. Únicamente podemos concluir que O_i era móvil pero no podemos relacionarlo con ninguno de los obstáculos presentes en el instante t .

4. Resultados experimentales

El algoritmo de extracción del estímulo *Obstáculo* se ha probado en un simulador de Mundos Virtuales desarrollado con anterioridad, [9]. Con él se han creado diferentes entornos, con distintos tipos de objetos y se ha analizado la eficacia del algoritmo en términos del número de clasificaciones correctas. Posteriormente se ha experimentado el algoritmo de percepción de *Obstáculo* con el robot real ROJO, moviéndose en el campo que rodea al IAI_CSIC sobre un terreno de grava y arena, por el que puede pasar el personal del centro, con árboles y edificios en su proximidad.

4.1. Resultados obtenidos en simulación

Las pruebas realizadas en simulación se han agrupado en dos categorías, en función del tipo de Obstáculo a detectar: estático y móvil, con el objetivo principal de evaluar la eficacia del algoritmo de clasificación.

Entornos con objetos estáticos. Los experimentos se han llevado a cabo en siete entornos diferentes, figura 4, con agrupaciones de objetos en las que varían tanto el tamaño de los mismos como su distribución espacial.

En la tabla 1, se presentan en tanto por ciento los aciertos del algoritmo en la clasificación de obstáculos para los distintos entornos. Un observador humano evalúa los aciertos del algoritmo. En la configuración espacial e y f se dispone de menos datos dado que el *Obstáculo* se encuentra dentro del *Área_de_Movimiento* del robot durante un tiempo muy corto.

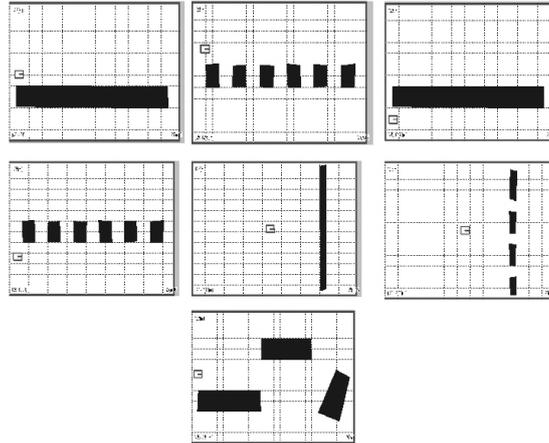


Figura 4: Los 7 entornos estáticos para las pruebas con el simulador.

Entorno	Nt	Ne	E
a	49	48	98 %
b	69	66	96 %
c	99	83	83 %
d	78	60	77 %
e	2	2	100 %
f	21	14	67 %
g	81	67	83 %

Tabla 1: Resultados para entornos estáticos. Nt, pruebas realizadas con el algoritmo de clasificación propuesto; Ne total de aciertos; E tanto por ciento de aciertos.

Entornos con objetos móviles. El simulador desarrollado permite crear Obstáculos móviles en un mundo de objetos estáticos. Los Obstáculos se definen como polígonos de n lados y su capacidad de movimiento propio viene controlada por la asignación de valores de velocidad en las direcciones del espacio 2D, v_x y v_y . Se han realizado pruebas en entornos en los que el *Obstáculo* describe diferentes trayectorias relativas al robot móvil. La situación relativa del *Obstáculo* respecto del móvil así como la tasa de aciertos en la clasificación se describen en la tabla 2.

Prueba	Trayectoria relativa obstáculo-robot				E
a	por la derecha	paralelo	1	$v_o = v_r$	96 %
b	por la derecha	paralelo	2	$v_o = v_r$	94 %
c	por la derecha	paralelo	1	$v_o > v_r$	82 %
d	por la derecha	paralelo	2	$v_o > v_r$	82 %
e	por la derecha	paralelo	1	$v_o < v_r$	73 %
f	por la derecha	paralelo	2	$v_o < v_r$	85 %
g	por la derecha	perpendicular	1		69 %
h	por la derecha	perpendicular	2		70 %
i	por la izquierda	paralelo	1	$v_o = v_r$	38 %
j	por la izquierda	paralelo	2	$v_o = v_r$	91 %
k	por la izquierda	paralelo	1	$v_o > v_r$	56 %
l	por la izquierda	paralelo	2	$v_o > v_r$	95 %
m	por la izquierda	paralelo	1	$v_o < v_r$	84 %
n	por la izquierda	paralelo	2	$v_o < v_r$	83 %
o	por la izquierda	perpendicular	1		57 %
p	por la izquierda	perpendicular	2		69 %
q	oblicuo	hacia arriba derecha	1	lento	96 %
r	oblicuo	hacia arriba derecha	1	rápido	100 %
s	oblicuo	hacia abajo izquierda	1	lento	79 %
t	oblicuo	hacia abajo izquierda	1	rápido	72 %
u	oblicuo	hacia abajo derecha	1	lento	92 %
v	oblicuo	hacia abajo derecha	1	rápido	75 %
w	oblicuo	hacia arriba izquierda	1	lento	57 %
x	oblicuo	hacia arriba izquierda	1	rápido	45 %
r	mezcla				90 %

Tabla 2: Descripción cualitativa de la trayectoria relativa del objeto móvil respecto del robot en simulación: aparición del obstáculo en el campo de visión del robot, trayectoria del obstáculo con respecto al robot, número de obstáculos y velocidad del obstáculo relativa al robot. E: tanto por cien de aciertos en la clasificación del estado de movimiento del objeto

4.2. Resultados con datos reales

En la figura 5, se muestra el mapa local dinámico obtenido en instantes consecutivos en un conjunto de pruebas en las que el robot ha permanecido estático y ante su campo de visión ha cruzado una persona; en este caso la persona se aleja del robot. Se han realizado pruebas con la persona cruzando el espacio del robot de izquierda a derecha, y de derecha a izquierda, así como con el robot moviéndose. Cabe destacar la rapidez con la que el mapa local incorpora las nuevas posiciones del *Obstáculo* móvil,

a la vez que elimina las localizaciones pasadas del móvil.

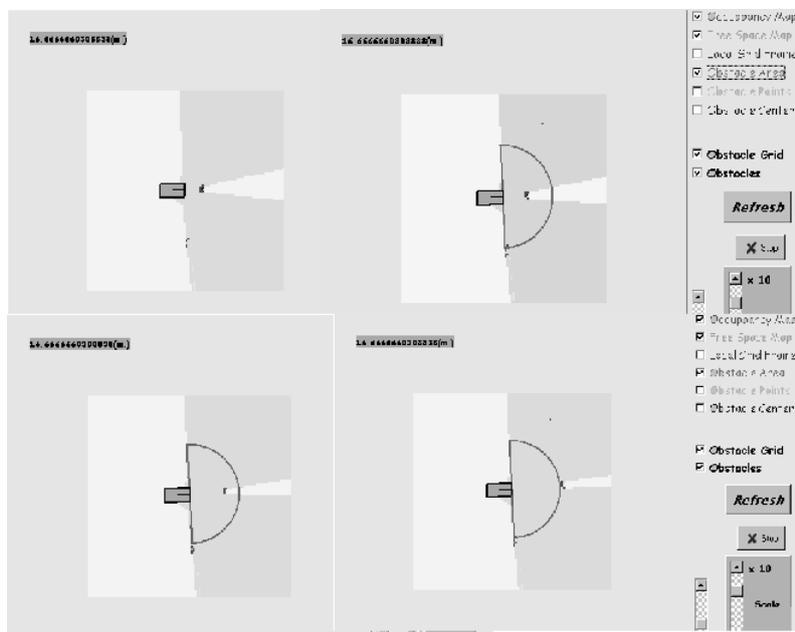


Figura 5: Mapa del entorno local del robot con persona alejándose de él.

4.3. Comentarios a la aproximación propuesta

La estrategia seguida en el cálculo y descripción de los obstáculos es sencilla, y ésta ha sido una característica perseguida ante los requerimientos de una navegación en tiempo real. Sin embargo, esto implica una excesiva simplificación del problema tratado y por ello da lugar a comportamientos inconsistentes frente a determinadas topologías de obstáculos.

Obstáculos muy rápidos o muy lentos. Según la dinámica de la detección del movimiento, los Obstáculos que se mueven muy rápido en relación al movimiento del robot darán lugar a patrones clasificados al describir el algoritmo como caso 3 y por lo tanto O_i es un *Obstáculo* móvil pero desconocemos a que *Obstáculo* corresponde en t . Esto es muy importante si el *Obstáculo* se está acercando al robot rápidamente o si presenta un comportamiento errante dentro del *Área de Movimiento* del robot. Esta razón es la que explica el bajo tanto por ciento de aciertos en los casos k, t y x de la tabla 2. Si por el contrario la velocidad del *Obstáculo* es muy pequeña el robot lo percibirá como un *Obstáculo* estático. Este fallo perceptual también se encuentra en los humanos como una incapacidad de distinguir entre objetos en reposo y objetos moviéndose muy lentamente.

Simplificación en la medida del tamaño. Al definir el tamaño como el número de celdas que forman el *Obstáculo* se obvia el hecho de que con el sensor de barrido láser se detectan contornos y no superficies, por lo tanto en algún caso la medida del tamaño

estimado puede dar lugar a una percepción falsa, por ejemplo en el caso de objetos perpendiculares al movimiento del tractor.

Lecturas erróneas. Aunque el sensor láser es de alta precisión (50(mm.) en 30(m.)), se pueden obtener esporádicamente lecturas erróneas que darán lugar a la detección de Obstáculos en áreas de *Espacio Libre*. Al no existir ninguna restricción en el mínimo número de celdas que dan lugar a la aparición de un *Obstáculo*, el algoritmo de construcción del estímulo es incapaz de filtrar este ruido. Ésta es la fuente más importante de errores en la clasificación de los obstáculos. Sin embargo, al considerar el tamaño como una propiedad inherente al *Obstáculo*, los agentes de comportamiento que utilicen esta percepción sí pueden descartar aquellos Obstáculos cuyo tamaño no supere un determinado umbral.

5. Conclusiones

La capacidad de distinguir entre obstáculos estáticos y móviles en un punto clave para un robot no holonómico que navega en exteriores parcialmente estructurados, ya que permite anticipar el cálculo de la trayectoria en tiempo real y desarrollar estrategias de navegación eficaces en la consecución de un objetivo. En este trabajo se ha presentado la elaboración del estímulo *Obstáculo* que activará en el comportamiento *Ir_a_Punto* la planificación de una trayectoria acorde con la nueva configuración de obstáculos en la escena.

El mapa local desarrollado, esto es la memoria espacio temporal de que se dispone, y el tipo de obstáculo detectado posibilita el replanteamiento “on-line” de la navegación en curso.

Los resultados de detección obtenidos con el algoritmo propuesto en este trabajo así como la extrema rapidez en la actualización del mapa local constituyen un paso fundamental en la generación de estrategias de navegación de carácter deliberativo/reactivo en tiempo real.

El algoritmo de elaboración del estímulo, es susceptible de mejoras tanto en la estimación del tamaño del obstáculo como en el cálculo del módulo y dirección de la velocidad. El cálculo de la velocidad de los obstáculos permitirá el empleo de algún método de predicción de sus posiciones futuras.

Agradecimientos

La investigación se encuentra financiada en su totalidad por el proyecto CICYT-TAP98-0781: Arquitectura Multiagente: generación de comportamiento complejo para un robot de pulverización en exteriores y el Ministerio de Ciencia y Tecnología mediante una beca predoctoral. Los autores quieren agradecer a los Drs. Domingo Guinea y Angela Ribeiro las múltiples reflexiones y el continuo soporte en el trabajo realizado, así como al Prof. Arne Halme y su equipo por la supervisión de parte de este trabajo durante la estancia de uno de los autores en Helsinki University of Technology.

Referencias

- [1] J. Borenstein and Y. Koren, *The vector field histogram- fast obstacle avoidance for mobile robots*, IEEE Journal of Robotics and Automation **7** (1991), no. 3, 278–288.
- [2] J. M. Cañas and L. García-Pérez, *Construcción de mapas dinámicos: comparativa*, WAF02 (Murcia, Spain), 2002.
- [3] Dwayne C. Carver and Lauren M. Peterson, *Laser radar in robotics*, Proceedings of the IEEE **84** (1996), no. 2, 299–320.
- [4] Michel Devy, Raja Chatila, Philippe Fillatreau, Simon Lacroix, and Fawzi Nashashibi, *On autonomous navigation in a natural environment*, Robotics and Autonomous systems (1995), no. 16, 5–16.
- [5] Alberto Elfes, *Autonomous mobile robots: Perception, mapping and navigation*, vol. 1, pp. 60–71, IEEE Computer Society Press, 1991.
- [6] Everett, *Sensors for mobile robots*, John Wiley and Sons, 1995.
- [7] E. Fabrizi and A. Saffiotti, *Extracting topology-based maps from gridmaps*, Proc. of the International Conference on Robotics and Automation, ICRA 2600 (San Francisco, CA, USA), 2000, pp. 2972–1978.
- [8] M.C. García-Alegre, A Ribeiro, L. García-Pérez, R. Martínez, D. Guinea, and A. Pozo-Ruz, *Autonomous robot in agriculture tasks*, 3ECPA-3 European Conf. On Precision Agriculture (Montpellier, France), june 2001.
- [9] L. García-Pérez and M.C. García-Alegre, *A simulation environment to test fuzzy navigation strategies*, 10th IEEE International Conference on Fuzzy Systems (Melbourne, Australia), 2001.
- [10] Jorge Gasós and Alejandro Martín, *A fuzzy approach to build sonar maps for mobile robots*, Computers in Industry **32** (1996), no. 2, 351–167.
- [11] Kwun Han and Manuela Veloso, *Reactive visual control of multiple non-holonomic robotic agents*, Int. Conf. on Robotics and Automation ICRA'98 (Leuven Belgium), 1998.
- [12] Rubén Martínez, Eugenio Villanueva, Domingo Guinea, and María C. García-Alegre, *Automatización robot rojo: descripción diseño mecánico*, Tech. Report TR-07/00b, Instituto de Automática Industrial, 2000.
- [13] Jun Miura, Hiroshi Uozumi, and Yoshiaki Shirai, *Mobile robot motion planning considering the motion uncertainty of moving obstacles*, Proc. 1999 IEEE Int. Conf. on Systems, Man and Cybernetics, vol. 4, october 1999.
- [14] Ulrich Nehmzow and Carl Owen, *Robot navigation in the real world: Experiments with manchester's fortytwo in unmodified, large environments*, Robotics and Autonomous systems **33** (2000), 223–242.
- [15] Lotfi Zadeh, *From computing with numbers to computing with words -from manipulation of measurements to manipulation of perceptions*, IEEE Transactions on Circuits and Systems-I: Fundamental theory and applications **45** (1999), no. 1, 105–119.