

Internet Electronic Journal Nanociencia et Moletrónica

Marzo 2004, Vol. 2; N°1, págs. 140-151 (2004)

Control de un Robot Caminante Aplicando Algoritmos de Búsqueda

**M. M. Bustillo Díaz¹, A. Campeche Romero¹, A. F. K. Zehe²,
E. Coraza Palafox¹, J. Italo Cortez³**

Benemérita Universidad Autónoma de Puebla, ¹Facultad de Ciencias de la Computación, Laboratorio de Arquitectura de Computadoras.

14 Sur y Avenida San Claudio,

²Facultad de Ciencias de la Electrónica

³Facultad de Estomatología

Puebla, Pue., México; Tel.: + (222) 229-55-00, ext. 7217;

e-mail: bustillo@solarium.cs.buap.mx

recibido: diciembre 12, 2003

revisado: Febrero 29, 2004

publicado: Junio 29, 2004

Citation of the article:

M. M. Bustillo Díaz¹, A. Campeche Romero¹, A. F. K. Zehe², E. C. Palafox¹, J. Italo C.³ "Control de un Robot Caminante Aplicando Algoritmos de Búsqueda", Internet Electrón. J. Nanocs. Moletrón. 2004, 2(1), 140-151

<http://www.revista-nanociencia.ece.buap.mx>

Control de un Robot Caminante Aplicando Algoritmos de Búsqueda

**M. M. Bustillo Díaz¹, A. Campeche Romero¹, A. F. K. Zehe²,
E. Coraza Palafox¹, J. Italo Cortez³**

Benemérita Universidad Autónoma de Puebla,

¹Facultad de Ciencias de la Computación, Laboratorio de Arquitectura de Computadoras,
14 Sur y Avenida San Claudio,

²Facultad de Ciencias de la Electrónica

³Facultad de Estomatología

Puebla, Pue., México; Tel.: + (222) 229-55-00, ext. 7217;

e-mail: bustillo@solarium.cs.buap.mx

recibido: diciembre 12, 2003

revisado: Febrero 22, 2004

publicado Junio 29, 2004

Internet Electron. J. Nanocs. Moletrón. 2004, 2(1), pags 140-151.

Resumen

Los robots móviles son de gran utilidad en nuestros tiempos, pues desempeñan tareas peligrosas para los humanos, como explorar o buscar en áreas peligrosas. Los robots que utilizan el sistema de locomoción mediante patas son los más adecuados para llevar a cabo este tipo de tareas por que tienen la ventaja de desplazarse muy bien en terrenos de difícil acceso. En el trabajo aquí expuesto se utiliza un robot caminante de seis patas, en el cual se implementa el algoritmo Bactracking para la búsqueda de marcas que conduzcan a una salida, en conjunto con un algoritmo de navegación basado en marcas para el desplazamiento de éste en un ambiente desconocido. La aplicación de estos algoritmos dan lugar a otras tareas tales como la locomoción, el procesamiento de imágenes, y la proyección de marcas para asegurar el camino que debe seguir el robot.

1 Introducción

Un robot móvil tiene la característica de estar montado en una plataforma movable, y tener la capacidad de desplazarse de un lugar a otro en un área de trabajo para llevar a cabo tareas asignadas. Estos requieren y aplican el conocimiento sobre su posición física, y además tienen que controlar su movimiento. En la medida, que el tamaño físico del robot se disminuye, hablamos de robots pequeños, como por ejemplo de sistemas micro-electro-mecánicos (MEMS), o de nanorobots en forma de sistemas nano-electro-mecánicos (NEMS) [1], se tienen que considerar aplicaciones, que involucran grupos (e incluso grupos grandes) de robots. La robótica cooperativa es el área de investigación multidisciplinaria actual, que incluye contribuciones de electrónica, inteligencia artificial y hasta biología; en la nanorobótica se consideran robots fabricados en escala molecular, y de un tamaño semejante a los virus conocidos.

Se utilizan técnicas de Inteligencia Artificial para la planificación de movimientos y navegación para robots, pudiendo incluir nano-mecanismos.

En la presentación de este trabajo se utiliza el algoritmo de Backtracking, que es una técnica utilizada en problemas que involucran la búsqueda de la mejor solución ó el conjunto de éstas como es el caso del juego de las ocho reinas que satisfacen ciertas condiciones, entre otros. Así también, en problemas donde se permite encontrar una solución por medio de una secuencia de decisiones [2].

Para la navegación se utiliza un algoritmo basado en marcas, que le permite al robot guiarse a través del espacio, utilizando como marcas flechas achatadas, para mostrar la dirección del desplazamiento del robot, indicando una probable solución: encontrar la salida. Las marcas son distribuidas de manera uniforme en el espacio de trabajo.

En resumen el resultado de este trabajo es que el robot busque una salida en espacios desconocidos, utilizando los algoritmos antes mencionados.

2 Desarrollo

A continuación se describen las etapas de desarrollo: descripción del sistema, locomoción, análisis de los algoritmos, estados de los sensores, dirección del robot.

2.1 Descripción del Sistema

Se trabaja con el robot Eyewalker, un robot móvil caminante de seis patas controlado por una tarjeta de propósito general (Eyebot) basada en el microcontrolador 68332 de Motorola. Esta tarjeta permite controlar 12 servomotores, 3 sensores infrarrojos y una cámara fija, con una resolución de 80x60 píxeles. Se cuenta con una memoria limitada para el procesamiento de datos, 1Mb en RAM y 512kb en Flash-ROM para los programas. El sistema cuenta con bibliotecas que permiten realizar procesamiento de imágenes.



Figura 1: Robot Eyewalker

2.2 Locomoción

Una característica importante de los robots caminantes es poder trasladarse de un lugar a otro con ayuda de sus piernas. A esta tarea se le conoce como locomoción, que en los últimos años ha recibido un gran interés en lo que a robots con piernas se refiere, prestando principal atención a la generación de pasos estáticamente estables.

Los robots caminantes cuentan con un número distinto de piernas. Los hay de 1, 2 (bípedos), 3 (trípodos), 4 (cuadrúpedos), 6 (hexápodos), 8 (octópodos) o más piernas. Este número de piernas es importante para poder llevar a cabo la locomoción en condiciones óptimas.

En este trabajo se utiliza el paso de trípode, que es el más rápido paso estable que un robot de seis patas puede utilizar. Donde se considera que al menos tres piernas estén siempre en tierra antes de que se ejecute un paso.

$$(L1,L4,L5),(L2,L3,L6) \quad (1)$$

2.3 Algoritmo de Búsqueda

El algoritmo Backtracking es una modificación del algoritmo en profundidad, pues sólo expande un nodo a la vez [3,9]. Este algoritmo tiene la característica de retroceder si es que no llega a las solución por una de las ramas expandidas y además es recursivo.[4].

Algoritmo de búsqueda con retroceso [8,6]:

- i) Si ABIERTA está vacía, terminar con fallo; en caso contrario, continuar.
- ii) Seleccionar el primer nodo de ABIERTA y denominarlo m .
- iii) Si la profundidad de m es igual a p o si m ya no tiene más sucesores posibles (que no hayan sido examinados anteriormente), eliminarlo de ABIERTA y regresar a (ii); en caso contrario, continuar.
- iv) Generar un nuevo sucesor m' de m e introducirlo al principio de ABIERTA, creando un puntero a m , y señalar que dicha rama ya ha sido considerada.
 - iv.i) Si m' es meta, abandonar el proceso iterativo iniciado en el paso (ii), devolviendo el camino de la solución, que se obtiene recorriendo los punteros de sus antepasados.
 - iv.ii) Si m' se encuentra en un callejón sin salida, eliminarlo de ABIERTA, (Se continúa el proceso iterativo en el paso (ii)).

Para este trabajo se toman en cuenta n flechas, distribuidas de manera uniforme en el espacio de trabajo. Estas marcas son tomadas como posibles soluciones que el robot puede tomar para llegar a la solución [2]. El algoritmo de búsqueda sólo sirve para encontrar las marcas, cuando una de ellas se ha encontrado, el algoritmo de búsqueda se deja de utilizar y sólo se utiliza el algoritmo de navegación.

El robot es puesto en un punto inicial del cual debe partir para buscar la salida o una marca, si al caminar se topa con un objeto que no es la marca seguirá buscando hasta llegar a otro.

En caso de que en el trayecto encuentre la marca inmediatamente abandona el algoritmo, de lo contrario sigue buscando hasta extender un máximo de tres nodos en una dirección, sino la encuentra regresa y prueba con la siguiente rama, en otra dirección.

En la ecuación (2) se muestra al conjunto de marcas consideradas.

$$M = \{m_1, \dots, m_8\} \quad (2)$$

tal que $m_i \in \{0,1\}, 1 \leq m \leq 8$ de forma que al elegir un m_i , se tienen dos opciones:

- $m_i=0$ si la flecha indica derecha ó
- $m_i=1$ si la flecha indica izquierda

La m_i seleccionada será tomada como la dirección, si es el primer caso se tendrá que avanzar en dirección hacia la derecha, y hacia la izquierda en el caso dos [5,10].

Para la formación del nodo se toma en cuenta la siguiente información:

- a) El estado de los sensores, que permite encontrar las m_i
- b) La primera opción que se encuentre en m_i , será la que se siga en todo el trayecto; es decir, la dirección que se encuentre primero es la que se seguirá en todo el trayecto; por ejemplo si la flecha con dirección izquierda se encuentra primero esta será tomada en todo el trayecto, por el contrario si se encuentra primero la derecha, entonces esta será tomada en todo el trayecto.
- c) Una vez que se encuentra la marca se realiza un cálculo de distancias para seguir la dirección en forma correcta y evitar posibles problemas para encontrar la siguiente.

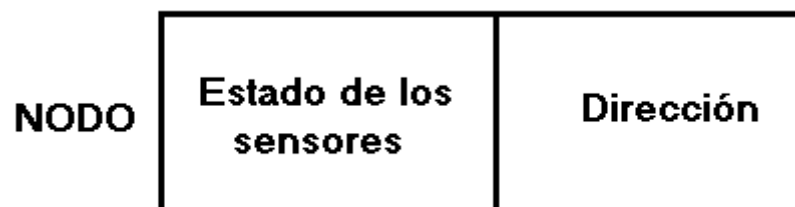


Figura 2: Elementos que forman el nodo.

2.4 Algoritmo de navegación

Se ha considerado el modelo de percepción sensorial basado en señales para la solución de este problema. En la descripción del concepto original se tiene que: “Estas señales son tratadas como si fueran puntos rodeados por un campo de influencia circular. Dentro de este campo el robot puede saber su posición con exactitud. Pero si el robot se encuentra fuera de este campo de influencia no podrá contar con información directa sobre su posición”[3,4]

Para la solución planteada de este problema con el hexápodo, se han considerado las marcas como puntos para poder determinar la dirección que el robot debe tomar.

Este algoritmo permitirá el desplazamiento del robot hasta llegar a la salida, para garantizar que el robot no se desvíe de su trayectoria, se realizará una proyección de las marcas, esto se trata con detalle más adelante.

2.5 Estado de los sensores

Los estados de los sensores dan lugar a movimientos que el robot lleva a cabo para dirigirse a cada una de las marcas. Los movimientos que el robot realiza son tres uno es caminar hacia adelante, otro girar a la izquierda y el último girar a la derecha.

En la tabla 1 se pueden observar los estados de los sensores, donde un 0 representa que no hay información en el sensor y 1 que existe información.

Sensor Frente	Sensor Derecho	Sensor Izquierdo
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

Tabla1: Estados de los ocho sensores.

Debido a que la cámara es fija y no cuenta con movimiento. El robot debe realizar giros hacia la izquierda o derecha para verificar si el obstáculo sensado es o no una marca. En la figura 3 se muestra un ejemplo de cómo el robot obtiene información del sensor izquierdo, en tanto al frente y derecha no existe ninguna información, entonces el robot gira hacia el lado izquierdo y verifica con la cámara si es una marca, en este caso si lo es, por lo que el robot sigue la dirección indicada por la flecha.

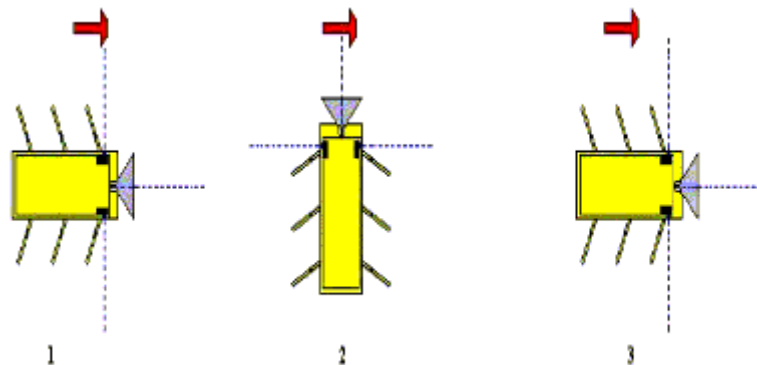


Figura 3: Movimientos para verificar una marca. 1) detección con los sensores, 2) giro hacia el lado izquierdo, 3) seguimiento de la dirección de la marca.

2.6 Proyección

Cuando se ha encontrado una de las marca, se debe asegurar que el robot siga la dirección de forma correcta hasta encontrar la siguiente, en el mejor de los casos el cuadrado que indica la salida, ó bien hasta que se tope con un obstáculo.

Se ha propuesto realizar una proyección de dicha marca tomando en cuenta tres puntos de la flecha mostrados en la figura 4. Los puntos (x_1, y_1) y (x_2, y_2) son pixeles obtenidos a partir de la distancia del centro de masa a los bordes, el punto (x, y) es el centro de masa, se obtiene a partir de la ecuación (3).

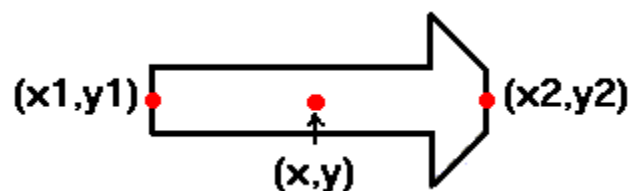


Figura 4: Localización de los tres puntos en la flecha para su proyección.

El centro de masa (\bar{x}, \bar{y}) es obtenido a partir de la media estadística:

$$\bar{x} = \frac{1}{N} \sum_{x,y \in \text{objeto}} x \quad \bar{y} = \frac{1}{N} \sum_{x,y \in \text{objeto}} y \quad (3)$$

donde: N es el número de píxeles que contiene el objeto. Los sumandos se extienden para todos aquellos píxeles pertenecientes al objeto.

La proyección de la marca se lleva a cabo en un plano cartesiano, tomando como referencia la flecha. En el eje de las x 's, se tiene que el punto (x_2, y_2) de la marca será el punto uno de la primera proyección y el punto (x'_2, y'_2) será el punto uno de la siguiente proyección, así sucesivamente hasta llegar a la siguiente marca.

La línea formada con esta proyección será formada con el punto (x_1, y_1) con $y_1 = 0$ de la flecha y la distancia d que hay entre cada una de las flechas [7]. Utilizando la forma general de la ecuación de la recta para el caso cuando $B=0$ enunciada en la ecuación (4).

$$x = -C/A \quad (4)$$

La recta obtenida servirá como referencia para llevar a cabo una medición entre el robot y la recta proyectada. En este caso se proponen tres puntos del robot y a partir de ellos, calcular la distancia entre estos puntos y los puntos de la recta como se muestra en la figura 5.

Las dimensiones del robot son de 30.6 cm de largo por 6.3 cm. de ancho, sin tomar en cuenta las bases de las patas. Por lo que el primer punto tendrá el valor de $(0, 3.15)$, el segundo de $(15.3, 3.15)$ y el tercero de $(30.6, 3.15)$.

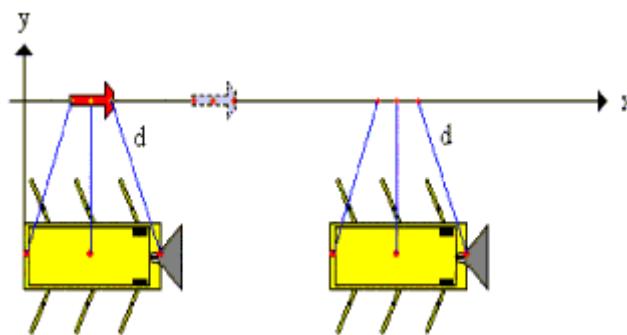


Figura 5: Proyección de la marca.

El cálculo de la distancia d entre la proyección y el robot, se lleva a cabo a partir del teorema de distancia entre dos puntos:

“La distancia d entre dos puntos $P_1(x_1, y_1)$ y $P_2(x_2, y_2)$ está dada por la fórmula”

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (5)$$

Distancia	PSDGetRaw	PSDGet
1.0 Cm	242	142
1.5 cm	161	109
2.0 cm	177	116
2.5 cm	242	142
3.0 cm	322	164
3.5 cm	300	159
4.0 cm	205	130
4.5 cm	150	101
5.0 cm	118	79
5.5 cm	99	54
6.0 cm	81	26
6.5 cm	70	15
7.0 cm	65	12
7.5 cm	72	16
8.0 cm	76	21
8.5 cm	83	28
9.0 cm	87	36

Tabla 2: Medidas obtenidas por un sensor, colocado al frente.

3 Pruebas y resultados

Se realizaron pruebas con cada uno de los dispositivos conectados al robot, para verificar y calibrar el funcionamiento de estos.

Una de las pruebas se realizó con los servomotores, consistió en utilizar combinaciones para el movimiento de las piernas, se probaron diferentes velocidades, así como diferentes número de pasos.

En cuanto a las pruebas de los sensores, se realizaron varias mediciones a diferentes distancias utilizando la función PSDGet(sensor), incluida en el sistema operativo del robot.

Otra de las pruebas realizadas con los sensores consistió en que el robot caminara hacia el frente, y utilizando el sensor ubicado en la parte frontal del robot pudiera detenerse cada vez que se encontrara la presencia de un objeto con el sensor. También se realizaron pruebas con los otros dos sensores, el izquierdo y el derecho. Pero a diferencia de la prueba anterior sólo se envió un letrero al LCD que indicara que existía un objeto en ese lado. Más adelante se probaron las funciones de girar izquierda y girar derecha, cada una con el sensor respectivo.

Con la cámara lo primero que se probó fue que las imágenes pudieran ser obtenidas y se pudieran ver en el LCD. Una vez realizado lo anterior se prosiguió a calcular la distancia a la que deberían estar las flechas para que la cámara pudiese obtenerlas completas. La distancia ideal fue a 45cm, pero se tuvo que cambiar un poco el tamaño de las marcas para que la cámara pudiese sensar a una distancia mayor.

Se verificó que la función IPSobel (imagen, imagen1), obtuviera correctamente el borde de la imagen origen, para después trabajar con la imagen obtenida. Una vez aplicado el filtro se escribieron en el LCD cada uno de los píxeles del borde. La comparación de las flechas y el cuadrado se hicieron utilizando la variable imagen1.

Una vez probados por separado los dispositivos, y calculadas las distancias que se requerían para la implementación de los algoritmos. Se prosiguió a la programación por separado de las funciones que se utilizarían, comenzando con pruebas pequeñas hasta llegar a lo que la función debía realizar.

La programación de los algoritmos requirió de varias pruebas y de varios intentos sobretodo, cuando los sensores ó la cámara no detectaban presencia alguna. Esto ocurrió algunas veces debido a que no se percató que la pila estaba baja.

4 Conclusiones

Para este trabajo se propusieron e implementaron dos algoritmos, uno de búsqueda y uno de navegación en combinación, con la finalidad de encontrar una salida en un espacio desconocido, utilizando un robot caminante.

En la implementación de los dos algoritmos, se obtuvieron buenos resultados, tomando en cuenta que el robot es de recursos limitados. Si los recursos crecieran se abre la posibilidad de sustituir estos algoritmos por otros que utilicen redes neuronales o algoritmos genéticos.

En cuanto a la utilización de los servos, sensores y cámara, no existieron tantos inconvenientes pues las funciones que se incluyen en el sistema operativo del robot, pudieron satisfacer las tareas implementadas.

Este trabajo sirve como punto de partida para estudiar a fondo la programación de robots caminantes y poder hacer aportaciones interesantes en el mundo de la robótica

5 Referencias.

- [1] A.Zehe, www.moletrónica.buap.mx
- [2] <http://www.lsi.upc.es/~iea/transpas/>
- [3] Stuart Russell, Peter Norving.: *Inteligencia Artificial Un Enfoque Moderno.* PrenticeHall, México (1996)
- [4] <http://www.aisa.uvigo.es/DOCENCIA/InteligenciaArtificial/Transparencias%20IA-tema%208-parte%204.pdf>
- [5] <http://www.inf.udec.cl/~intartif/mnvoa/map4.htm>
- [6] <http://www.lcc.uma.es/~iaic/retroceso.pdf>
- [7] Lehmann. *Geometría analítica.* uteha, México (1959)
- [8] <http://webesaii.upc.es/usr/cecilio/articles/IAP02Q2Sesion01Algoritmosdebusqueda.pdf>
- [9] Winston. *Inteligencia Artificial tercera edición.* Addison-Wesley Iberoamericana, U.S.A. (1994)
- [10] K. S. Fu, R. C. González, C. S. G. Lee. *Robótica control, detección, visión e inteligencia.* McGrawHill. México (1988)