

# GENERACIÓN AUTOMÁTICA DE MAPAS 3D INTERACTIVOS PARA ROBOTS MÓVILES Y SU USO EN LA PLANIFICACIÓN DE MOVIMIENTOS

## AUTOMATIC GENERATION OF INTERACTIVE 3D MAPS FOR MOBILE ROBOTS AND THEIR USE IN PLANNING MOVEMENTS

*Diego Murillo<sup>1</sup>, Ulises Zaldívar<sup>1</sup>, Luis H. Lavin<sup>1</sup>, Juan M. Ibarra Zannatha<sup>2</sup>*

<sup>1</sup> Facultad de Informática Mazatlán, Universidad Autónoma de Sinaloa, México

<sup>2</sup> Departamento de Control Automático del CINVESTAV, México

E-mail: murillo@uas.uasnet.mx

(Enviado Septiembre 20, 2013; Aceptado Noviembre 01, 2013)

### **Resumen**

En este trabajo se presenta el desarrollo de un sistema de obtención de mapas del entorno desconocido de un robot móvil basado en la información de rango provista por un sensor láser, así como la representación automática de esta información en forma de un mundo virtual (MV) tridimensional interactivo accesible por Internet. El sistema desarrollado permite, además, la representación del espacio libre encontrado en el entorno como un grafo en el cual pueden determinarse automáticamente las rutas más adecuadas para ir de un lugar a otro del entorno previamente cartografiado. El sistema desarrollado no sólo facilita la programación de las trayectorias, sino que permite una operación segura del robot al impedir colisiones mediante su sistema de navegación reactiva.

**Palabras Clave:** Robot, Mundo Virtual, Programación, Navegación Reactiva.

### **Abstract**

This paper presents the development of a system for obtaining maps of the unknown environment of a mobile robot based on the range information provided by a laser sensor, as well as the automatic representation of this information in the form of a virtual world (MV) interactive three-dimensional Internet. The developed system allows, in addition, the representation of the free space found in the environment as a graph in which the most suitable routes can be determined automatically to go from one place to another of the previously mapped environment. The system developed not only facilitates the programming of the trajectories but also allows a safe operation of the robot by preventing collisions through its reactive navigation system.

**Keywords:** Robot, Virtual World, Programming, Reactive Navigation.

## 1 INTRODUCCIÓN

Dentro de la Robótica, una de las áreas de mayor desarrollo en la actualidad es la que involucra a los robots móviles. Algunos de los temas de más interés en Robótica Móvil son el diseño mecánico de los vehículos, el desarrollo de sistemas de alimentación eficientes y de los servomecanismos del robot, el control de movimiento a lo largo de trayectorias previamente determinadas, la planificación automática de las tareas a realizar y de las trayectorias correspondientes, el desarrollo de sistemas de cómputo y de comunicación inalámbrica, el desarrollo de sensores y de sistemas de percepción sofisticados, incluyendo temas como los de integración sensorial [1]. Entre todos estos temas, el presente trabajo se ubica en el uso de sistemas de percepción avanzados (ultrasonido, láser, visión) para la obtención de mapas del entorno del

robot y para la navegación del robot en dichos entornos, incluyendo algunos métodos reactivos de navegación.

Por otro lado, en este trabajo se plantea el uso de técnicas de simulación gráfica 3D, denominadas realidad virtual, que permiten una interactividad bidireccional con el usuario y, además, con el sistema real que simula, así como su completa accesibilidad vía Internet.

## 2 GENERACIÓN DE MAPAS

Llamamos “*mapping*” al problema de generar modelos geométricos 2D o 3D de un espacio, utilizando robots móviles basándose en información obtenida por un sensor. De acuerdo con esta información, la meta del *mapping* es determinar la localización de diversos objetos

(blancos u obstáculos) en un espacio determinado, además de las búsquedas de paredes, logrando así, la generación de mapas de un ambiente determinado. Para generar mapas de ambientes de gran tamaño, el robot tiene que resolver el problema de la localización concurrente, el cual se da debido a que la odometría del robot se encuentra lejos de estar libre de error. El *mapping* es un problema simple si la localización del robot es conocida. El problema de la localización no es una tarea difícil si existe un mapa disponible; el posicionamiento basado en mapa requiere que existan características bastante estacionarias y distinguibles en el ambiente, que los sensores sean bastantes exactos, y un poder significativo de procesamiento disponible [1].

Existe un gran número de propuestas enfocadas para resolver el problema del *mapping*-localización concurrente. Todos ellos pueden estar clasificados en uno de los dos paradigmas mayores de *mapping* de los robots móviles: la *geometría y topografía*. El enfoque de tipo geométrico produce una descripción métrica, con niveles de detalles finos del ambiente del robot. Mientras que el enfoque del paradigma topográfico genera un gráfico tosco del ambiente, donde la representación de nodos significa lugares (que son fáciles de distinguir) o marcas y arcos que corresponden a la conexión de lugar más cerca. Cualquier paradigma tiene desventajas significativas. Las descripciones detalladas generadas por el enfoque del paradigma geométrico son mejores para distinguir diferentes lugares; estos requieren mucha memoria y consumen mucho tiempo de cómputo, este es el método más usado pero este no es la opción más satisfactoria para representar ambientes grandes. En lado opuesto, las representaciones producidas por el enfoque del paradigma topográfico son mucho más compactas y entonces representan las mejores alternativas para trazar estos tipos de ambientes. Es más, el paradigma topográfico es mejor para resolver problemas algorítmicos (Inteligencia Artificial). Sin embargo, ellos no son aplicados satisfactoriamente para trazar ambientes grandes debido a las dificultades en distinguir adecuadamente lugares diferentes. Es decir, cuando distintos lugares dentro del ambiente parecen idénticos para los sensores de los robots.

La técnica Kalman y la técnica de la estimación de cuadro son frecuentemente usadas en el problema de *mapping*-localización [2]. También se usan las técnicas de estimación de Bayesian y los modelos de Harkov para resolver el problema de la localización-*mapping*.

Para construir un mapa el robot debe explorar el ambiente. Típicamente este asume que el robot empieza la exploración autónomamente sin tener algún conocimiento del ambiente. Las principales tareas del *mapping* son: regeneración de una ruta explorada, uno de los modelos más usados es la celda de ocupación (*occupation grid*), del cual existen versiones para 2D y 3D [3].

### 3 PROBLEMÁTICA TRATADA

En este trabajo se presenta el desarrollo de un sistema de simulación para robots móviles que genera mapas 3D,

permitiendo además la navegación dentro del mapa registrando lugares identificados en un grafo. Con esta información es posible recorrer el mapa (con ayuda del grafo) haciendo uso del ya conocido algoritmo de *Floyd-Warshall* para encontrar la ruta más corta entre todos los pares de vértices del grafo [4]. A su vez, el grafo es dinámicamente modificado si existe una variación en el entorno previamente generado por el robot, las cuales pueden ser causadas por la detección de objetos que no se encontraban anteriormente en el entorno, o bien, por el movimiento de posibles puertas si el entorno de navegación las incluye. Cuando estas variaciones suceden, el grafo se modifica y se vuelve a encontrar la nueva ruta más corta, logrando así, una navegación reactiva muy eficiente. En la Fig. 1, se pueden observar todos los módulos que conforman el sistema y la interacción entre ellos.

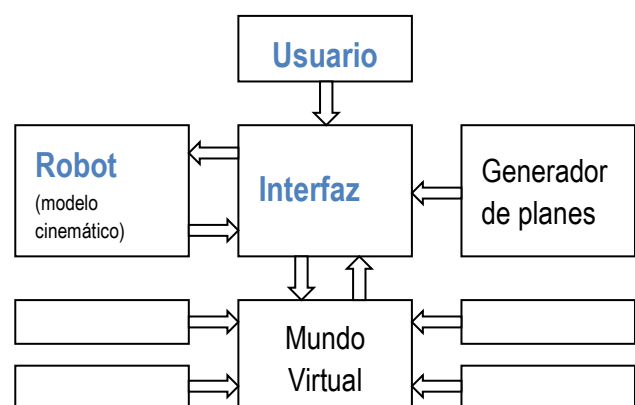


Figura 1 Diagrama del sistema desarrollado.

### 4 ESTADO DEL ARTE

En [5] se presenta una interfaz de navegación para robots móviles. Esta interfaz permite construir mapas de una gran escala de ambientes desconocidos, mientras que el mismo mapa es utilizado para navegar. Esta tarea se realiza con una interfaz llamada CML, en donde el robot es movido dentro del mapa que está continuamente construyendo y actualizando, porque el entorno siempre está cambiando. Por ejemplo, las puertas no son estáticas y las personas caminan en el ambiente. Cuando la navegación termina, el robot crea un mapa de gran escala de un ambiente complejo. El CML memoriza todas las rutas seguidas, las cuales usa para crear el mapa, el robot puede; sin embargo, regresar a la posición inicial automáticamente. El problema está al momento de regresar a su posición inicial, si algún objeto está bloqueando el camino, el robot no sabe qué hacer y podrá colisionar con él.

En [6] se ha logrado desarrollar trabajos de navegación reactiva muy eficientes, en donde los robots móviles utilizados pueden llegar a cualquier punto conocido, así sea el punto de partida, reduciendo considerablemente el error de posición. Uno de los principales inconvenientes que tienen, es que no tienen interfaces especializadas para el manejo y control de los

robots y no cuentan con una visualización apropiada de los mapas generados.

## 5 DESCRIPCIÓN DEL ARTÍCULO

En la sección 2 se mencionan las herramientas de software utilizadas para el desarrollo de este trabajo y una descripción global de los detalles que se deben tomar en cuenta en la construcción de un MV. En la sección 3 se hace una descripción de la plataforma utilizada como modelo, y de los aspectos cinemáticos de la misma. La sección 4, se dedica a explicar como es generada la maqueta virtual y los componentes del entorno. En la sección 5 se explica como se desarrolla la planificación de movimientos dentro del MV. Finalmente se presentan unos comentarios finales como conclusiones.

## 6 HERRAMIENTAS DE SOFTWARE UTILIZADAS

Para el desarrollo del sistema se evaluaron diferentes alternativas, y finalmente se optó por el uso de las presentadas en esta sección.

## 7 VRML

VRML es un lenguaje para construir mundos virtuales que se generan mediante código pequeño y fácil, que se transmite rápidamente vía Internet [7]. Este lenguaje no es compilado, se requiere de un *browser* (navegador) para ser interpretado. El lenguaje vrml esta compuesto por nodos (*shape*, *group*, *transform*, *box*, *sphere*, *cone*, *cylinder*, *Touchsensor*, *Planesensor*, *Spheresensor*, *Cylindersensor*, etc.), campos (*Field*, *ExposeField*, *SFBool*, *SFFloat*, *SFVec3f*, *SFTime*, *SFRotation*, etc.), y eventos (*eventOut*, *evenIn*, *ROUTE*). El primer paso para crear un objeto virtual usando VRML, es definir un nodo *shape*, el cual tiene 2 características principales: *geometry* y *appearance*. El primero puede tener las siguientes cuatro primitivas:

- a) **box** {size x y z}
- b) **sphere** {radius r}
- c) **cone** {bottomRadius r height h}
- d) **cylinder** {radius r height h}

Para formar un objeto complejo se tendrá que usar un conjunto de nodos para así formar una figura determinada, por ejemplo, un robot.

## 8 JAVA

En este proyecto se ha considerado que Java representa la mejor opción para programar la interacción y procesos existentes en el mundo virtual, debido a que ofrece una gran conectividad con VRML. Java ofrece toda la funcionalidad de un lenguaje orientado a objetos potente, pero sin las características menos usadas y más confusas de éstos [8] otra característica muy útil es el *garbage*

*collector* (reciclador de memoria dinámica) [9]. Gracias a esta característica no es necesario preocuparse de liberar memoria. Java se ha construido con extensas capacidades de interconexión TCP/IP. Existen librerías de rutinas para acceder e interactuar con protocolos como http y ftp. Esto permite a los programadores acceder a la información a través de la red con tanta facilidad como a los archivos locales. Además, una de la facilidad muy notable e importante de Java es que se pueden hacer aplicaciones que puedan ser accesibles desde la Internet. Creando proyectos más dinámicos y accesibles desde cualquier lugar del mundo. La integración de VRML con Java es una herramienta poderosa para diseñar mundos virtuales Interactivos.

## 9 CONSTRUCCIÓN DEL MV

Para crear el MV, los elementos existentes en el mundo real deberán ser estudiados, y así, obtener sus propiedades físicas y geométricas, de esta manera es posible pasarlo a la computadora siendo representado por el modelo gráfico apropiado. La representación de la parte mecánica que usa VRML no es complicada, siguiendo la regla del lenguaje, esta tarea es bastante fácil. La interacción entre el operador y los objetos virtuales no tienen grandes problemas. Proporcionar algunas propiedades físicas al mundo virtual, como la conducta dinámica y cinemática a los objetos en movimiento, es una tarea más compleja, especialmente para implementar algunas interacciones entre el MV y la interfaz creada en Java [10]. El propósito de la interfaz del sistema es permitir la comunicación entre Java y VRML para dar una interactividad bidireccional en el mundo virtual.

## 10 DESCRIPCIÓN DE LA PLATAFORMA

Para la realización de este proyecto, se tomó como modelo el robot móvil Labmate, el cual cuenta con sensores propioceptivos y exteroceptivos, en donde los propioceptivos nos permiten medir información odométrica, la cual se utiliza para determinar su posición en el ambiente.

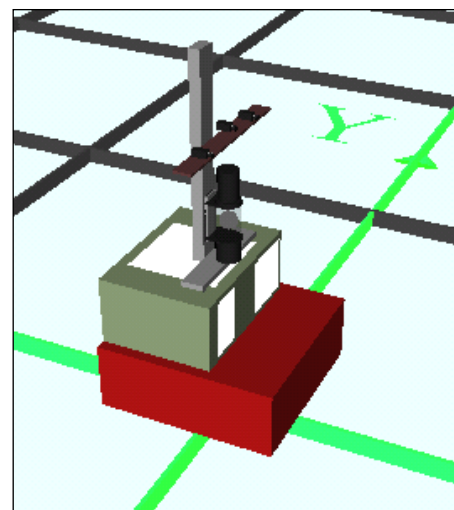


Figura 2 Robot virtual Labmate.

Los sensores exteroceptivos permiten obtener información del ambiente en el que se encuentra el robot. Estos están compuestos por sensores infrarrojos, cámaras y un sensor láser, siendo este último el utilizado en este proyecto para la simulación en la generación de mapas. En la Fig. 2 se muestra el modelo virtual del Labmate.

## 10 NAVEGACIÓN DEL ROBOT

En la interfaz de usuario existe una caja de controles, en la cual hay una serie de botones que permiten mover al robot hacia los siguientes sentidos: adelante, atrás, izquierda y derecha, modificando con estos las variables de velocidad lineal  $v$  y velocidad angular  $w$ . Al presionar uno de estos botones, se envía desde java al mundo virtual estos valores. Ver Fig. 3.

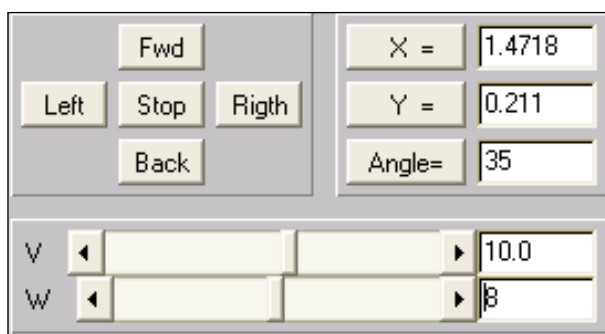


Figura 3 Caja de controles.

## 11 CINEMÁTICA DEL ROBOT

La cinemática directa básicamente nos ayuda a encontrar la posición en el espacio cartesiano y la orientación del robot. Las variables utilizadas para el modelado del robot son  $x$ ,  $y$  y  $\theta$ . Estos valores se calculan partiendo de las variables  $v$ ,  $w$  y  $t$ .

## 12 MODELOS CINEMÁTICOS

A continuación se presenta el modelo cinemática del robot para calcular la posición del robot ( $x, y$ ) y su orientación ( $\theta$ ).

para  $x$ :

$$x_t = x_{t-1} + \frac{v}{w} [\sin(\theta_t + \theta_{t-1}t) - \sin \theta_t]$$

para  $y$ :

$$y_t = y_{t-1} + \frac{v}{w} [\cos(\theta_t + \theta_{t-1}t) - \cos \theta_t]$$

para  $\theta$ :

$$\theta_t = \theta_{t-1} + wt$$

En la Fig. 4 se puede observar al robot desplazado de su origen, lo cual se logra modificando las variables anteriormente mencionadas.

## 13 MODELADO DEL ENTORNO

Los modelos virtuales que se utilizan en este proyecto son el robot y el mapa. El mapa está descrito en una base de datos que contiene un conjunto de vértices en un espacio bidimensional, a través de los cuales se genera el mapa en el MV. En el modelado también se utilizan unas marcas, las cuales sirven para crear un grafo sobre el mapa, el cual se utiliza para la planificación de movimientos. Además, se utilizan otros objetos que sirven para representar obstáculos en el mapa, los cuales, al aparecer durante la navegación automática del robot, obligan a realizar una navegación reactiva durante esta.

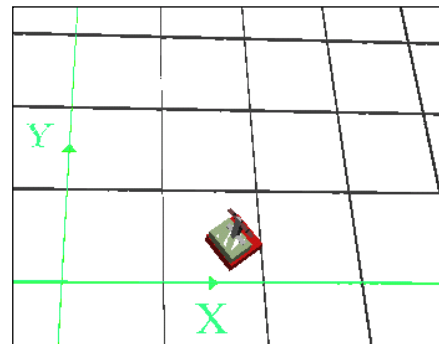


Figura 4 Robot en movimiento.

## 14 GENERACIÓN DE MAPAS

En el mundo virtual, cuando el robot está realizando una navegación en el ambiente, va detectando paredes por medio del sensor láser simulado, el cual detecta todos los obstáculos que se encuentren en un radio de 4 metros próximos al robot, generando así, mapas 3D de manera automática. Para representar el mapa en la base de datos, se utiliza una matriz  $V$  que contiene las coordenadas de cada vértice del mapa, y otra matriz  $A$ , donde se tiene la información que indica cuáles vértices están unidos, especificando también, si se trata de una pared o una puerta. Por ejemplo:

$$V_{i,1}=\text{coord } x \quad V_{i,2}=\text{coord } y$$

En donde cada  $V_i$  es un vértice del mapa.

Para los vértices que están unidos, y que son paredes, se representan de la siguiente forma:

$$A_{ij}=1$$

y esto quiere decir que el vértice  $i$  se une al vértice  $j$ , formando una pared entre ellos.

Los vértices que están unidos, y que son puertas, se representan de la siguiente forma:

$$A_{ij}=2$$

aquí se está indicando que los vértices  $i$  y  $j$ , se unen para representar una puerta, mientras que, para indicar que los vértices no están unidos, simplemente se asigna el valor de 0.

$$A_{ij}=0$$

Esta representación se puede observar en el mapa generado de la Fig. 5.

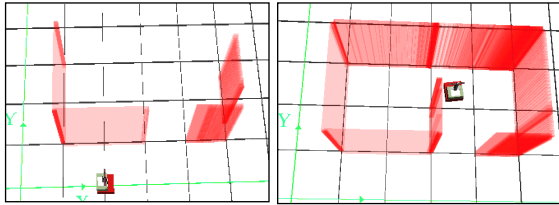


Figura 5 Modelado de un mapa o edificio.

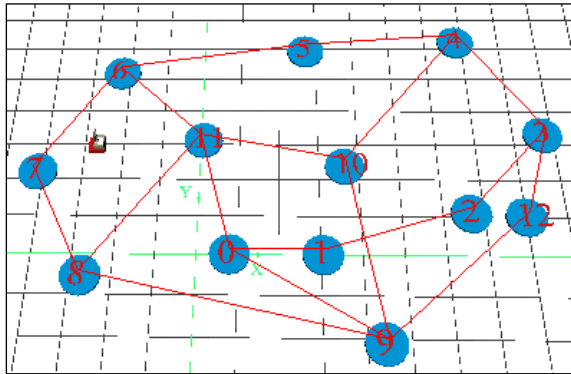


Figura 6 Marcas en el MV.

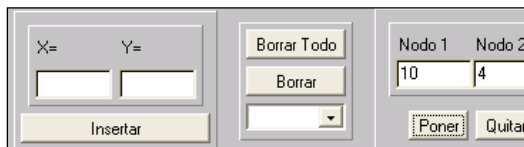


Figura 7 Interfaz que permite manipular el grafo.

## 15 OBSTÁCULOS

En el mapa modelado dentro del MV, es posible crear objetos arbitrariamente en alguna parte del mapa, esto con el fin de crear un ambiente dinámico en el cual el robot, pueda cambiar sus rutas tomando nuevas decisiones.

Estos objetos se representan como *cubos* en el mundo virtual, y pueden ser cambiados de lugar por medio de la interfaz de usuario, pues se deben simular objetos que pueden aparecer en el mapa de manera repentina, incluso objetos móviles y/o personas.

## 16 PLANIFICACIÓN DE MOVIMIENTOS

Una vez que el robot hizo un reconocimiento del entorno y ha modelado el mapa, es posible definir marcas en cualquier parte del MV, esto con el fin de etiquetar espacios o lugares específicos del mapa. El robot puede ir de un lugar a otro automáticamente recorriendo el mapa, pero pasando por estas marcas, las cuales están representadas como un grafo en el MV.

## 17 CREACIÓN DEL GRAFO

Mientras el robot navega en el mundo virtual, se pueden ir plasmando las marcas anteriormente mencionadas, usando estas marcas, el robot puede ir de un punto a otro o regresar a su origen, lo cual es relativamente sencillo para el usuario, pues simplemente se le indica al robot en alto nivel (ve de la marca 3 a la marca 9), o bien, simplemente se selecciona con el *mouse* la marca destino.

Además para completar el grafo y las uniones de las marcas, se puede utilizar la interfaz de usuario para hacerlo manualmente, o bien, el robot puede ir uniéndolas según sea el orden en el que se estén creando. Si se crean marcas que tengan mucha proximidad entre ellas, el sistema las fusiona suponiendo que se trata de la misma marca, o bien, del mismo lugar geográfico. En las Figs. 6 y 7 se muestra el MV con las marcas y la interfaz de usuario que permite hacer modificaciones del grafo.

## 18 PLANIFICACIÓN: RUTA MÁS CORTA

Para decidir cual es la ruta más corta dentro del grafo, se utiliza el conocido método de *Floyd – Warshall* [Cor 90], el cual encuentra la ruta más corta entre todos los pares de vértices de un grafo, De esta manera, el robot puede recorrer todo el mapa o MV de manera automática (ver Fig. 8) y de una forma óptima.

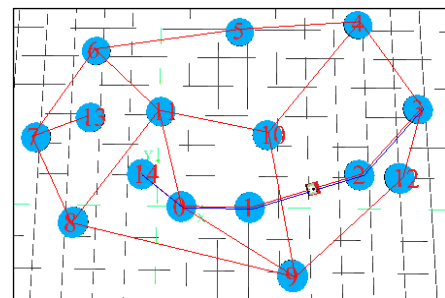


Figura 8 Ejemplo de la ruta más corta entre 14 y 3.

Cada vez que el usuario desee hacer un recorrido por el mundo virtual, usando las marcas para ir de un sitio a otro, el sistema siempre encontrará la ruta mas corta para llegar al lugar destino.

## 19 NAVEGACIÓN REACTIVA

Cuando el robot esta haciendo un recorrido por el MV de manera automática, utilizando las marcas, es posible crear obstáculos (silla, mesa, etc.) en el entorno de manera arbitraria, que impidan el libre traslado del robot de un punto a otro. Cuando esto sucede, el sistema elimina de manera automática la unión de las marcas afectadas en el grafo y hace una nueva búsqueda de la ruta mas corta. Cuando esto sucede, el robot tomo otro camino para llegar al destino (si es posible), logrando de esta manera una eficiente navegación reactiva en el mapa del MV. La interfaz de usuario permite la creación y eliminación de objetos según se desee. En la Fig. 9 se pueden observar

obstáculos impidiendo el paso del robot durante la navegación.

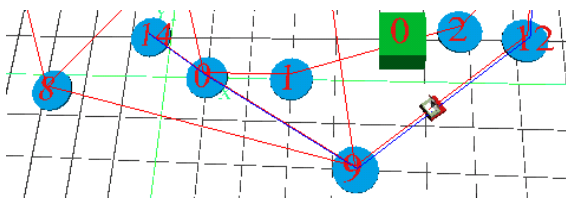


Figura 9 Representación de obstáculos en el MV.

## 20 CONCLUSIONES

Con el desarrollo de este sistema, se probó de manera simulada, la implementación de un método computacionalmente conocido (*Floyd-Warshall*), en el marco de la robótica móvil, particularmente para apoyar a la navegación reactiva de robots móviles. Se hizo la simulación de un sensor láser para detectar objetos en el ambiente de trabajo del robot móvil, el cual, cabe mencionar que proporciona información ideal, puesto que la simulación realizada no tiene tolerancia a errores, siendo que en un sistema real se deben utilizar métodos de localización para tratar de minimizar los errores de posición que van surgiendo. El modelado 3D del mapa, partiendo de información que se encuentra en un plano 2D, permite tener una visión más realista del entorno, y VRML es una herramienta muy fácil y práctica para el desarrollo del sistema. El robot móvil utilizado como modelo es el LabMate, el cual se encuentra en el Laboratorio de Robótica del Departamento de Ingeniería en Sistemas del Centro Politécnico Superior de la Universidad de Zaragoza en Zaragoza, España.

## 21 REFERENCIAS

- [1] Ibarra, J.M; Zaldívar, U; Iturbe, E; López, T; Montano, L; Minguez, J. "3D Mapping for Mobile Robots Using Interactive Virtual Worlds and Internet Teleoperation", 15th Triennial World Congress of the International Federation of Automatic Control, IFAC'02. Barcelona, 2002.
- [2] Borenstein, J.; Everett, H. R.; Feng, L. "Where am I?" Sensors and Methods for Mobile Robot Positioning. U. of Michigan, USA. April, 1996.
- [3] Thrun, S.; Beetz, M.; Bennewitz, M.; Burgard, W.; Cremers, A. B.; Dellaert, F; Fox, D.; Hähnel, D.; Rosenberg, C.; Roy, N.; Schulte, J.; Schulz, D. "Probabilistic Algorithms and the Interactive Museum Tour-Guide Robot Minerva". International Journal of Robotics Research, 2000.
- [4] Cormen, T. H.; Leiserson, C.; Rivest, R. "Introduction to Algorithms", section 26.2, pp. 558-562, 1990.
- [5] Newman. P.; Leonard, J. "Real-Time CML. A Geometric Hierarchy". MIT Dept. Ocean Engineering. 2001.
- [6] Minguez, J.; Montanom L. "Nearness Diagram (ND) Navitatio : Collision Avoidance in Troublesome Scenarios", IEEE transaction on robotics and automation Vol. 20. February 2004.
- [7] The Virtual Reality Modeling Language. International. Copyright © 1997 The VRML Consortium Inc. - URL: <http://www.vrml.org/technicalinfo/specifications/vrml97/index.htm>.
- [8] Deitel, H. M.; Deitel, P. J. "Java: How to Program", Prentice-Hall, Upper Saddle River New Jersey, 1998.
- [9] The Java Tutorial: Third Edition. A practical guide for programmers. URL: <http://java.sun.com/docs/books/tutorial/>
- [10] Ibarra Zannatha, J. M.; Zaldívar Colado, U. Wiederhold G. "An Approach to Internet Robotics: Generation of Interactive Virtual Worlds and Internet Teleoperation". 1st International Conference on Information Technology in Mechatronics, ITM'01, Estambul, Turquía, 2001.