

# Exploración eficiente de ambientes no estructurados con robots heterogéneos

Efficient exploration of unstructured environments

using heterogeneous robots

Presentación: 00/00/0000

Doctorando:

**Martin Nieves**

Centro de Investigación en Informática para la Ingeniería – CIII (Universidad Tecnológica Nacional, Facultad Regional Córdoba)  
martin.nieves.ar@gmail.com

Director/es:

**Gastón R. Araguás, Claudio J. Paz**

## Resumen

En robótica, la generación de un mapa es un elemento muy importante en la exploración, ya que le permite al robot desplazarse a través del entorno sin colisionar con los obstáculos. En este trabajo se presenta una introducción al concepto de exploración y los mapas comúnmente utilizados. Se exponen algunas de las dificultades encontradas en el ambiente de la robótica para la implementación de algoritmos de exploración en simuladores, y como pueden ser solucionadas mediante la utilización de contenedores Docker. Finalmente se presenta parte del trabajo actualmente en desarrollo, destinado a mejorar el tiempo de búsqueda y almacenamiento de los nodos de un mapa topológico, mediante una tabla de Hash.

Palabras claves: Exploración, Robots, Ambientes no estructurados, Mapas topológicos.

## Abstract

In robotics exploration, the map generation is a very important task, since it allows the robot to move through the environment without colliding with the obstacles. This paper presents an introduction to the concept of exploration and the representations commonly used in maps. Some of the difficulties encountered in the robotics environment for the implementation of exploration algorithms in simulators are exposed, and how they can be solved by using Docker containers. Finally, part of the work currently under development is presented, aimed at improving the search and storage time of the nodes of a topological map, using a Hash table.

Keywords: Robotic exploration, Topological maps, Unstructured environments,

## Introducción

Las horas y los días iniciales de un desastre son los más importantes en el rescate de sobrevivientes (Huder, R. C., 2013), por lo que es importante contar con un procedimiento de contingencia. Contar con una red de robots para colaborar con los rescatistas, puede mejorar la posibilidad de supervivencia para las personas atrapadas en ella. Además, con la ayuda de múltiples robots, el área puede ser explorada de forma más rápida, dando un tiempo valioso a los equipos médicos para atender a las víctimas. Por lo tanto, existe una clara necesidad de robots de búsqueda y rescate que puedan ser liberados inmediatamente después de un desastre en el que las condiciones sean demasiado peligrosas y confinadas para que personas y perros comiencen a buscar víctimas.

En comparación con los trabajadores y perros de búsqueda y rescate entrenados, los robots tienen muchas ventajas. Entre ellas se puede diferenciar que un robot de rescate, a diferencia de su contra parte humana, no sufrirá de estrés ni de fatiga (Burke, J. L., 2004). Por otro lado, los robots de rescate pueden ser desplegados en grandes cantidades, mientras que los profesionales de rescate con experiencia y los perros de rescate entrenados son recursos escasos cite (Casper J., 2003). Por

último, se puede considerar que los robots son recursos prescindibles, pero los humanos y perros de rescate no: si un robot de rescate es dañado, puede ser reparado o reemplazado fácilmente. Pero la pérdida de trabajadores de rescate podría ser muy difícil debido a su relación dentro de la sociedad (Casper J., 2003).

Debido a que los entornos de desastres urbanos son desordenados, no estructurados e impredecibles es difícil para los robots explorar y navegar de forma autónoma (Casper J., 2003). El proceso de exploración puede ser interpretado como un proceso de navegación, mediante el cual un robot en un entorno desconocido, construye un modelo del medio ambiente. Dependiendo de la aplicación, del entorno y del tipo de robot entre otras cosas, los problemas asociados a la navegación son muy diversos.

Un algoritmo de exploración puede ser definido como un procedimiento iterativo que consiste en una selección de un nuevo objetivo y una navegación hacia este objetivo. Tal algoritmo finaliza siempre que se cumpla la condición definida (objetivo de la misión). Mediante la estrategia de exploración, se determina el próximo objetivo del robot en cada paso de exploración. La misma, utiliza como información el conocimiento actual del entorno y un criterio de optimización seleccionado.

Dependiendo de la representación del espacio utilizada por el robot, existen diferentes formas de identificar las zonas no exploradas. Uno de los más populares es el uso de mapas utilizando una cuadrícula de ocupación (Moravec, H., 1985). Éstos mapas representan la probabilidad de ocupación de cada zona del entorno dentro de una cuadrícula. Según la probabilidad de ocupación, las celdas se pueden clasificar como libres, ocupadas o desconocidas. Usando esta representación, un robot puede explorar una zona desconocida, navegando hacia las celdas de frontera. Las mismas separan las celdas libres de las celdas desconocidas (Yamauchi, B., 1997).

En (Wullschleger, F. H, 1999). utilizan una representación del entorno, donde cada segmento, corresponde a una pared diferente de un ambiente interior. Los autores de (Makarenko, A. A., 2002) utilizan una planificación de ruta basado en el costo-utilidad, donde su función de utilidad considera la ganancia de información del mapa como la capacidad de localización del robot en este punto.

Estrategias de exploración más complejas intentan determinar las mejores posiciones de observación para mejorar la eficiencia del proceso de exploración.

En particular, su objetivo es reducir el tiempo de exploración, realizando un pequeño número de pasos de exploración y limitando la ruta recorrida, mientras se construye un mapa preciso.

En general, estos métodos emplean un enfoque voraz (Tovey, C., 2003)., que consiste en mover el robot desde su ubicación actual hacia la siguiente mejor posición de observación.

La exploración con múltiples robots ha recibido una mayor atención por su notable beneficio de mayor eficiencia y robustez (Franchi A., 2009). La misma tiene muchas aplicaciones como misiones de reconocimiento, vigilancia, búsqueda y rescate en áreas peligrosas (Liu, Y., 2013), entre otras.

En los años recientes, se han desarrollado algunas propuestas para la problemática de búsqueda y rescate utilizando varios robots (Incze, M. L., 2015), (Yanguas-Rojas D., 2017).

En los últimos años la robótica, como área de investigación, ha demostrado un gran interés por la reproducción de resultados que se publican (Bonsignorio, F., 2015). Sin embargo, pese a que se avanza hacia el uso de herramientas de programación que permiten a los investigadores tener una base común de desarrollo, como ROS (Cousins, S., 2010), Gazebo (Koenig, N., 2004), los sistemas son cada vez más complejos y difíciles de reproducir. Sumado a esto, no siempre se indican todas las herramientas o dependencias utilizadas en el proceso de investigación para lograr su reproducción.

Debido a que no siempre se dispone de robots físicos, o el tiempo de preparación de los mismos es grande, en la práctica el primer paso para probar o comparar un algoritmo de exploración es mediante simulación. Dentro de los simuladores 3D, se pueden nombrar: USARSim (Carpin, S., 2004), y V-REP (Rohmer E., 2013) entre otros.

En los últimos años V-REP y Gazebo han tenido un gran uso en diferentes tareas relacionadas con la exploración utilizando robots (Mohammed S. A., 2017) y (Jia, D., 2016).

Esto se debe, en gran parte, a su integración con ROS (Robot Operating System), el cual a su vez contiene varios paquetes para tareas de exploración, algoritmos de planificación de rutas y construcción de mapas. Sin embargo, configurar el entorno no es una tarea trivial, ya sea por problemas de compatibilidad, por la inexistencia de ciertos paquetes en una determinada versión, o porque la instalación en el ambiente de trabajo personal requiere de ciertas dependencias que no pueden ser satisfechas

## Resultados

Como se mencionó anteriormente, el primer paso para la implementación y prueba de algoritmos de exploración es en un simulador. En (Nievas M., 2019). se presentó la utilización de contenedores Docker para realizar pruebas con diferentes algoritmos de exploración.

Los algoritmos fueron implementados en ROS Kinetic Kame, y evaluados mediante el simulador Gazebo V7 dentro del mismo contenedor. Se diseñaron simulaciones para utilizar dos algoritmos de SLAM y el algoritmo de exploración de fronteras más cercanas. El primer algoritmo de SLAM corresponde al paquete Hector-SLAM provisto por ROS entre sus paquetes, el mismo utiliza la odometría y la distancia de un sensor láser para generar un mapa en forma de grilla de ocupación del entorno. El mapa es generado a medida que el robot se desplaza en el entorno, pero debido a que la medición de odometría presenta incertidumbres, en las diferentes simulaciones realizadas, el algoritmo tiende a fallar (generando mal algunos sectores del mapa) cuando la velocidad de desplazamiento es muy grande, como puede verse en la Figura 1. Esto es debido a que se producen deslizamientos entre las ruedas y el piso, por lo que la medición de odometría no es correcta. El segundo algoritmo de SLAM es Gmapping, el cual está implementado como un filtro de partículas utilizando el teorema de Rao-Blackwell, para generar un mapa en forma de grilla, a partir de datos de rango láser y mediciones de odometría. Como se puede ver en la Figura 1, debido a su forma de implementación, este algoritmo demostró una mejor calidad en el mapa generado, ya que incorpora de forma explícita la incertidumbre de los sensores en el modelo.

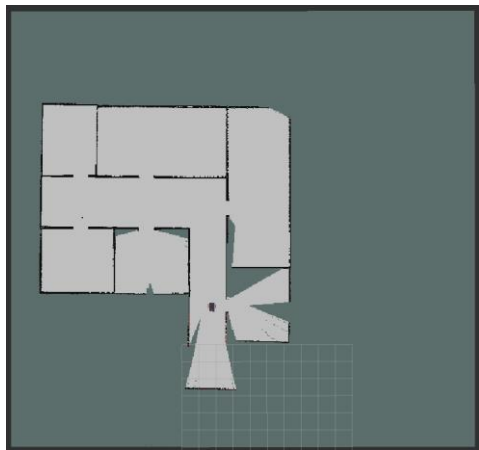


Figura 1: Mapa generado con el paquete Gmapping

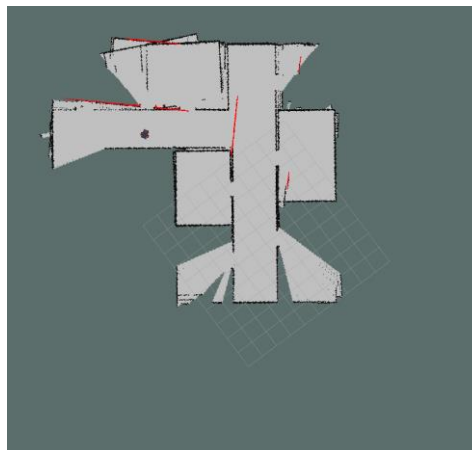


Figura 2: Mapa generado con el paquete Hector-SLAM

También presentaron las ventajas de utilizar contenedores a la hora de realizar las pruebas, entre las cuales se pueden mencionar la aislación del entorno de trabajo del entorno local y, debido al uso de contenedores Docker en Linux, el flujo de trabajo prácticamente no se modifica, por lo que el usuario tiene una rápida adaptación a su uso. En el mismo trabajo también se presenta un repositorio en el cual puede encontrarse el archivo Dockerfile y los algoritmos utilizados en las pruebas, a los fines de poder reproducir los resultados obtenidos. Además, se incluyen diferentes escenarios de pruebas, a los fines de realizar una evaluación más exhaustiva de los algoritmos de exploración.



Figura 3: Escenario de oficina no estructurado

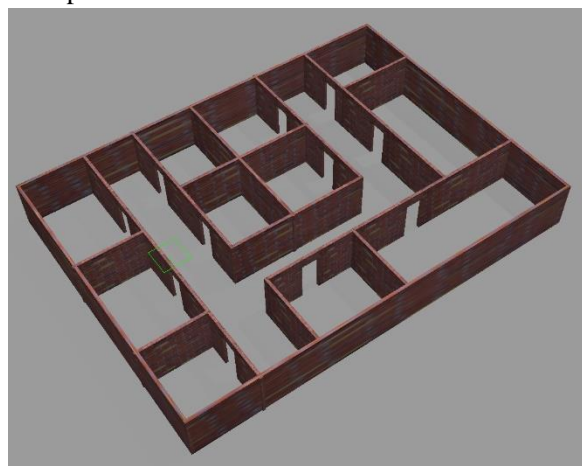


Figura 4: Escenario de oficina

Entre los escenarios disponibles, se incluye dos ambientes de oficinas compuestos por habitaciones conectadas por pasillos, con la diferencia que uno presenta el amueblamiento y el otro no, como puede verse en las Figuras 3 y 4 respectivamente. Esta diferencia entre los ambientes es a los fines de representar un ambiente no estructurado, en el cual no se puede asumir que las paredes y habitaciones van a estar libres de obstáculos.

Junto con el entorno de simulación se proveen dos robots diferenciales (Figura 5), completamente configurados con el paquete "navigation stack" provisto por ROS. El mismo permite la implementación del algoritmo de exploración de fronteras más cercanas (Yamauchi, B. 1997), en el cual el robot se desplaza hacia la zona no visitada más próxima, dentro de una región delimitada, como puede verse en la Figura 6.

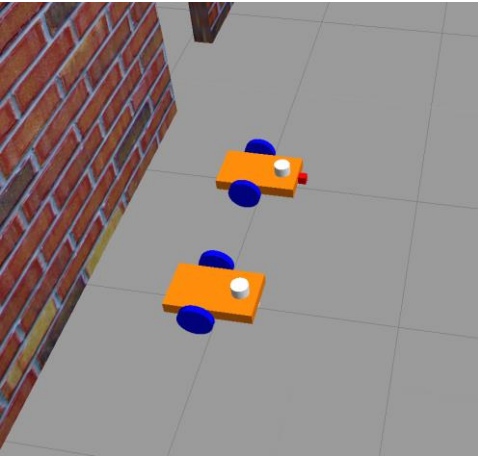


Figura 5: Robots utilizados en la simulación

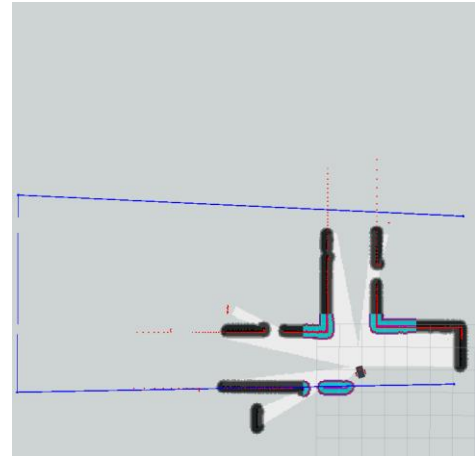


Figura 6: Región delimitada para la exploración autónoma

Actualmente se está trabajando sobre un algoritmo para mejorar el tiempo de búsqueda y almacenamiento de los nodos de un mapa topológico, mediante una tabla de Hash. A diferencia de los mapas métricos anteriormente mencionados, los mapas topológicos se representan mediante un grafo (Figura 7), donde los vértices corresponden a lugares accesibles por el robot, y las aristas representan conexiones entre esos lugares.

Es importante destacar que, a diferencia del mapa en forma de grilla de ocupación, un mapa topológico no preserva las propiedades métricas del entorno (Figura 8). Esto, sin embargo, se complementa con el bajo requerimiento de memoria para almacenar el grafo y, con un aumento en la velocidad de búsqueda, ya que la cantidad de elementos almacenados es significativamente menor. El algoritmo se enfoca en el almacenamiento del mapa en forma de grafo, y la búsqueda de los nodos ya visitados, mediante una tabla de Hash, la cual provee un tiempo de acceso constante a los elementos almacenados en la misma.

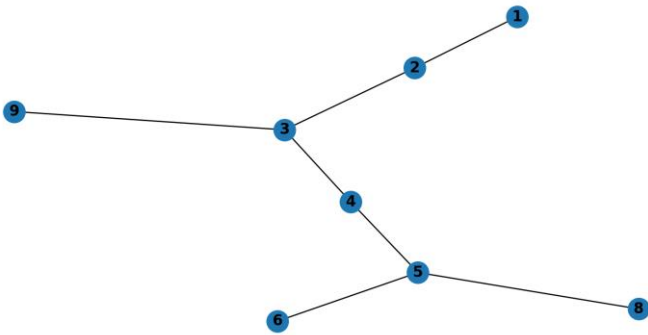


Figura 7: Grafo generado por el algoritmo

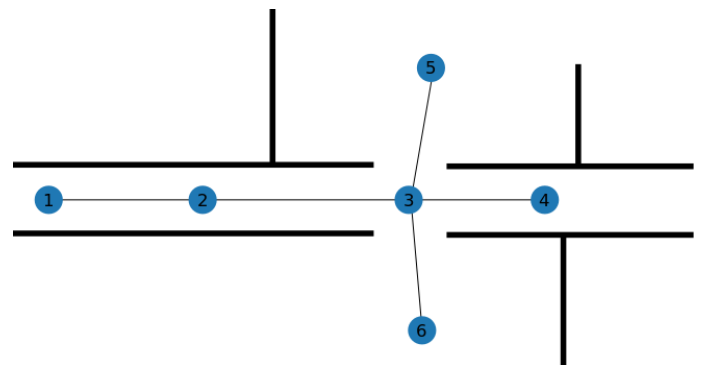


Figura 8: Grafo superpuesto sobre el escenario de pruebas

Particularmente se están realizando pruebas en árboles, los cuales son una clase especial de grafos en la que no se presentan ciclos y aseguran que existe solo un camino entre alguna de sus hojas, al nodo inicial o raíz del árbol. Mediante la combinación de la búsqueda en la tabla de Hash y el almacenamiento en árboles, se espera lograr una mejora en la velocidad de búsqueda de los lugares ya visitados, manteniendo un tamaño reducido en memoria.

## Referencias

Huder, R. C. (2013). *Disaster operations and decision making*. John Wiley & Sons.

Burke, J. L., Murphy, R. R., Covert, M. D., & Riddle, D. L. (2004). Moonlight in Miami: Field study of human-robot interaction in the context of an urban search and rescue disaster response training exercise. *Human-Computer Interaction*, 19(1-2), 85-116.

Casper, J., & Murphy, R. R. (2003). Human-robot interactions during the robot-assisted urban search and rescue response at the world trade center. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 33(3), 367-385.

Moravec, H., & Elfes, A. (1985, March). High resolution maps from wide angle sonar. In *Proceedings. 1985 IEEE International Conference on Robotics and Automation* (Vol. 2, pp. 116-121). IEEE.

Yamauchi, B. (1997, July). A frontier-based approach for autonomous exploration. In *cira* (Vol. 97, p. 146).

- Wullschleger, F. H., Arras, K. O., & Vestli, S. J. (1999, September). A flexible exploration framework for map building. In *1999 Third European Workshop on Advanced Mobile Robots (Eurobot'99). Proceedings (Cat. No. 99EX355)* (pp. 49-56). IEEE.
- Makarenko, A. A., Williams, S. B., Bourgault, F., & Durrant-Whyte, H. F. (2002, October). An experiment in integrated exploration. In *IEEE/RSJ International Conference on Intelligent Robots and Systems* (Vol. 1, pp. 534-539). IEEE.
- Tovey, C., & Koenig, S. (2003, October). Improved analysis of greedy mapping. In *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453)* (Vol. 4, pp. 3251-3257). IEEE.
- Franchi, A., Freda, L., Oriolo, G., & Vendittelli, M. (2009). The sensor-based random graph method for cooperative robot exploration. *IEEE/ASME Transactions on Mechatronics*, *14*(2), 163-175.
- Liu, Y., & Nejat, G. (2013). Robotic urban search and rescue: A survey from the control perspective. *Journal of Intelligent & Robotic Systems*, *72*(2), 147-165.
- Incze, M. L., Sideleau, S. R., Gagner, C., & Pippin, C. A. (2015, May). Communication and collaboration of heterogeneous unmanned systems using the joint architecture for Unmanned Systems (JAUS) standards. In *OCEANS 2015-Genova* (pp. 1-6). IEEE.
- Yanguas-Rojas, D., Cardona, G. A., Ramirez-Rugeles, J., & Mojica-Nava, E. (2017, October). Victims search, identification, and evacuation with heterogeneous robot networks for search and rescue. In *2017 IEEE 3rd Colombian Conference on Automatic Control (CCAC)* (pp. 1-6). IEEE.
- Bonsignorio, F., & Del Pobil, A. P. (2015). Toward replicable and measurable robotics research [from the guest editors]. *IEEE Robotics & Automation Magazine*, *22*(3), 32-35.
- Cousins, S., Gerkey, B., Conley, K., & Garage, W. (2010). Sharing software with ros [ros topics]. *IEEE Robotics & Automation Magazine*, *17*(2), 12-14.
- Koenig, N., & Howard, A. (2004, September). Design and use paradigms for gazebo, an open-source multi-robot simulator. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)* (Vol. 3, pp. 2149-2154). IEEE.
- Carpin, S., Lewis, M., Wang, J., Balakirsky, S., & Scrapper, C. (2007, April). USARSim: a robot simulator for research and education. In *Proceedings 2007 IEEE International Conference on Robotics and Automation* (pp. 1400-1405). IEEE.
- Rohmer, E., Singh, S. P., & Freese, M. (2013, November). V-REP: A versatile and scalable robot simulation framework. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 1321-1326). IEEE.
- Mohammed, S. A., & Gomaa, W. (2017). Exploration of Unknown Map for Safety Purposes using Wheeled Mobile Robots. In *ICINCO (2)* (pp. 359-367).
- Jia, D., Wermelinger, M., Diethelm, R., Krüsi, P., & Hutter, M. (2016, October). Coverage path planning for legged robots in unknown environments. In *2016 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)* (pp. 68-73). IEEE.
- Nievas, M., Paz, C., Araguás, G. (2019). Dockerización de ROS para despliegue ágil de algoritmos de exploración. In *48 Jornadas Argentinas de Informática JAIIO*.