



UT- T00058

DOI- 7014



CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS  
DEL INSTITUTO POLITÉCNICO NACIONAL

Laboratorio de Tecnologías de Información

**Estrategia Descentralizada para la  
Exploración Multi-Robot,  
incluyendo Restricciones en Rango  
de Comunicación**

Tesis que presenta:

**Juan Carlos Elizondo Leal**

Para obtener el grado de:

**Doctor en Ciencias  
en Computación**

Director de la Tesis:

Dr. José Gabriel Ramírez Torres

Cd. Victoria, Tamaulipas, México.

Junio, 2013

**CINVESTAV  
IPN  
ADQUISICIÓN  
LIBROS**

CLASS.	UT00058
ISSUE.	UT-T00058-SS1
ISSUE DATE.	10-01-2014
ISSUE BY.	Don - 2014

10-211025-2001

© Derechos reservados por  
Juan Carlos Elizondo Leal  
2013

La tesis presentada por Juan Carlos Elizondo Leal fue aprobada por:

-----

\_\_\_\_\_  
Dr. Himer Avila George

\_\_\_\_\_  
Dr. Guadalupe Acosta Villareal

\_\_\_\_\_  
Dr. Eduardo Arturo Rodríguez Tello

\_\_\_\_\_  
Dr. Wilfrido Gómez Flores

\_\_\_\_\_  
Dr. José Gabriel Ramírez Torres, Director

Cd. Victoria, Tamaulipas, México., 28 de Junio de 2013

A mi hijo Carlos y mi esposa Esme

# Agradecimientos

- Al Dr. Gabriel por sus sabios consejos que hicieron posible la elaboración de esta tesis.
- A mis revisores por sus valiosos comentarios para mejorar la calidad de este reporte.
- Al CONACyT por haberme otorgado el apoyo económico para concluir mis estudios de Doctorado.
- Al Cinvestav-Tamaulipas institución que siempre procuró proporcionarme lo necesario para la satisfactoria realización de mi proyecto.
- A mis padres por ser los mejores guías y apoyo.
- A mis hermanos por su apoyo moral y económico, en especial a Cynthia y Beto.
- A mi hijo y mi esposa por su sacrificio y amor, por ser la mayor inspiración en mi vida.

# Índice General

<b>Índice General</b>	<b>I</b>
<b>Índice de Figuras</b>	<b>III</b>
<b>Índice de Tablas</b>	<b>V</b>
<b>Índice de Algoritmos</b>	<b>VII</b>
<b>Publicaciones</b>	<b>IX</b>
<b>Resumen</b>	<b>XI</b>
<b>Abstract</b>	<b>XIII</b>
<b>1. Introducción</b>	<b>1</b>
1.1. Problemas Inherentes a la Robótica Móvil . . . . .	3
1.2. Motivación . . . . .	7
1.3. Definición del Problema . . . . .	8
1.4. Hipótesis . . . . .	11
1.5. Objetivos de la Investigación . . . . .	11
1.6. Descripción del Enfoque Propuesto . . . . .	12
1.7. Esquema de la Tesis . . . . .	16
<b>2. Estado del Arte</b>	<b>19</b>
2.1. Exploración Clásica Multi-Robot . . . . .	20
2.1.1. Exploración Clásica Multi-Robot con Comunicación Ideal . . . . .	21
2.1.2. Exploración con Rango de Comunicación Limitado . . . . .	22
2.2. SLAM Multi-Robot . . . . .	24
2.3. Exploración Multi-Robot con Incertidumbre en Localización . . . . .	27
2.4. Comparación de las Principales Propuestas . . . . .	29
2.5. Conclusiones . . . . .	30
<b>3. EEDT: Un Nuevo Planificador de Trayectorias</b>	<b>33</b>
3.1. Introducción . . . . .	33
3.2. Transformación de Distancia . . . . .	36
3.3. Algoritmo EEDT . . . . .	42
3.4. Resultados . . . . .	50
3.4.1. Trayectorias Obtenidas . . . . .	50

3.4.2.	Evaluación de la Exactitud . . . . .	52
3.4.3.	Tiempos de Ejecución . . . . .	57
3.5.	Conclusiones . . . . .	62
<b>4.</b>	<b>Exploración Multi-Robot Usando Auto-Ofertas bajo Restricción en el Rango de la Comunicación</b>	<b>63</b>
4.1.	Introducción . . . . .	63
4.2.	Definición del Problema . . . . .	65
4.3.	Propuesta . . . . .	68
4.3.1.	Localización y Generación de Mapas . . . . .	73
4.3.2.	Cálculo de las Ofertas . . . . .	73
4.3.3.	Algoritmos Desarrollados . . . . .	77
4.4.	Resultados Numéricos . . . . .	81
4.4.1.	Motivación para el Uso de un Simulador . . . . .	81
4.4.2.	Algoritmos Usados para la Comparación . . . . .	83
4.4.3.	Pruebas Realizadas . . . . .	84
4.5.	Conclusiones . . . . .	104
<b>5.</b>	<b>Conclusiones y Trabajo Futuro</b>	<b>107</b>
5.1.	Trabajo Futuro . . . . .	109
5.1.1.	Planificación de Trayectorias . . . . .	109
5.1.2.	Exploración Multi-robot . . . . .	109
	<b>Bibliografía</b>	<b>111</b>

# Índice de Figuras

1.1. Problemas asociados con robótica móvil: (I) SLAM, (II) exploración (III) localización activa, (IV) exploración integrada [54] . . . . .	5
1.2. Modelo gráfico del problema de exploración multi-robot. Los arcos indican las relaciones causales. . . . .	10
3.1. Grafo de visibilidad para la planificación de trayectorias (las zonas verdes indican el área ocupada). . . . .	35
3.2. Ventanas utilizadas por el algoritmo de transformación de distancias de Jarvis. . . . .	39
3.3. Generación de frente de onda mediante el algoritmo de Jarvis usando la ventana chessboard. . . . .	40
3.4. Trayecto desde la celda azul marcada con 11 hasta 0 obtenido por la propuesta de Jarvis (celdas en azul). El trayecto obtenido aplicando el suavizado propuesto por Lee se muestra en rojo. . . . .	43
3.5. Ventanas de trama utilizadas por mi propuesta en el cálculo de la transformación de distancia. . . . .	45
3.6. La misma celda es asignado a los dos vecinos ( $p(v_a) = p(v_b) = p(v_i)$ ). . . . .	47
3.7. Distintas celdas son asignadas a los dos vecinos ( $p(v_a) \neq p(v_i) \vee p(v_b) \neq p(v_i)$ ). . . . .	47
3.8. Cálculo de la distancia cuando la celda vecina apunta hacia una celda visible . . . . .	48
3.9. Cálculo de la distancia cuando la celda vecina apunta hacia una celda que no es visible para la celda actual . . . . .	49
3.10. Mapas usados para las pruebas de planificación global de trayectorias . . . . .	51
3.11. Trayectorias generadas para el mapa autolab ( $405 \times 345$ celdas). . . . .	53
3.12. Mapas de visibilidad generados. Para cada región (color) las flechas apuntan hacia la próxima mejor celda visible para llegar al objetivo. . . . .	54
3.13. Imágenes resultantes para un mapa libre ( $1000 \times 1000$ celdas). . . . .	56
3.14. Frentes de onda resultantes para el mapa autolab de $405 \times 345$ celdas. . . . .	58
3.15. Frentes de onda resultantes para el mapa cave de $400 \times 400$ celdas. . . . .	59
4.1. Modelo gráfico del problema de exploración multi-robot. Los arcos indican las relaciones causales. . . . .	67
4.2. Diagrama de interacción de los estados de cada robot. . . . .	69
4.3. Dos fronteras están cercanas entre si y pueden ser exploradas por un solo robot. . . . .	70
4.4. Ejemplo de mi propuesta usando 3 robots. . . . .	72

4.5. Ejemplo de los parámetros usados en mi propuesta usando 2 robots y en la cual el robot R2 calcula las ofertas para la frontera $f_5$ . Explotación $\rho_2 + \alpha_2^5$ , exploración $\rho_1 + \alpha_1^5$ y cohesión $\varphi$ . . . . .	75
4.6. Esquema de mi propuesta en cada robot. . . . .	78
4.7. Abstracción de Hardware con Player. . . . .	83
4.8. Mapa usado para las simulaciones ( $50 \times 25$ metros) . . . . .	84
4.9. Tiempo total fuera del rango de la comunicación . . . . .	86
4.10. Tiempo máximo de aislamiento de algún robot (segundos). . . . .	87
4.11. Distancia euclidiana máxima fuera del rango de comunicación . . . . .	89
4.12. Tiempo total necesario para completar la exploración. . . . .	91
4.13. Distancia total recorrida por el equipo de robot . . . . .	92
4.14. Mapas locales usando la propuesta de Yamauchi (15 metros de rango de comunicación). . . . .	95
4.15. Mapas locales usando la propuesta de Stachniss (15 metros de rango de comunicación). . . . .	96
4.16. Mapas locales usando Auto-ofertas. . . . .	97
4.17. Exploración por un equipo de tres robots usando la propuesta de Yamauchi. . . . .	99
4.18. Exploración por un equipo de tres robots usando la propuesta de Stachniss. . . . .	100
4.19. Exploración por un equipo de tres robots usando mi propuesta. . . . .	101
4.20. Mapa generado a partir de la odometría con error gaussiano. . . . .	102
4.21. Mapa con error en odometría, construido utilizando el algoritmo MRICP . . . . .	103

# Índice de Tablas

2.1. Comparación de las propuestas más relevantes para esta tesis. . . . .	31
3.1. Error cuadrático medio (celdas). Comparación con el algoritmo de Jarvis usando distintas ventanas con el frente de onda generado desde el centro de un mapa libre de obstáculos. . . . .	55
3.2. Error cuadrático medio (celdas) relativo a mi propuesta. Comparación para distintos algoritmos de transformación de distancias para los mapas mostrados en la Figura 3.10. . . . .	57
3.3. Error mayor (celdas) utilizando las distintas técnicas de transformación de distancia. . . . .	60
3.4. Comparación de tiempo necesario para la generación del frente de onda desde distintos puntos de cada mapa (segundos). . . . .	61
4.1. Tiempo total fuera del rango de la comunicación (segundos).	86
4.2. Tiempo máximo de aislamiento (segundos). . . . .	88
4.3. Distancia euclidiana máxima fuera del rango de la comunicación (metros).	90
4.4. Tiempo total necesario para completar la exploración (segundos). . . . .	92
4.5. Distancia total recorrida por el equipo de robot (metros). . . . .	93

# Índice de Algoritmos

1.	Algoritmo de transformación de distancia de Jarvis. . . . .	38
2.	Suavizado de trayectorias propuesto por Lee. . . . .	42
3.	EEDT . . . . .	44
4.	Validación de visibilidad . . . . .	46
5.	Actualiza $d(r)$ y $p(r)$ . . . . .	49
6.	Exploración Principal . . . . .	79
7.	Cálculo de tabla de ofertas $b_k^f$ . . . . .	80
8.	Comunicación . . . . .	81

# Publicaciones

Juan Carlos Elizondo-Leal, Ezra Federico Parra-González and José Gabriel Ramírez-Torres  
*The Exact Euclidean Distance Transform: A New Algorithm for Universal Path Planning*,  
International Journal of Advanced Robotic Systems, Vol. 10, 2013, DOI: 10.5772/56581.

Juan C. Elizondo-Leal, Gabriel Ramírez-Torres, *An Exact Euclidean Distance Transform for Universal Path Planning*, cerma, pp.62-67, 2010 IEEE Electronics, Robotics and Automotive Mechanics Conference, 2010

Juan Carlos Elizondo-Leal, José Gabriel Ramírez-Torres, A New Algorithm for Exact Euclidean Distance Transform for Universal Path Planning, presentado durante el 1st International Conference on Applied Bionics and Biomechanics ICABB-2010, Venecia, Italia, 14-16 de octubre 2010.

Juan Carlos Elizondo-Leal, José Gabriel Ramírez-Torres, An Exact Euclidean Distance Transform for Universal Path Planning, proceedings 16th International Congress on Computer Science Research, Ciudad Madero, Tamaulipas, México, 20-22 de octubre 2010, pp. 152-163.

## **Estrategia Descentralizada para la Exploración Multi-Robot, incluyendo Restricciones en Rango de Comunicación**

por

**Juan Carlos Elizondo Leal**

Doctor en Ciencias del Laboratorio de Tecnologías de Información

Centro de Investigación y de Estudios Avanzados, 2013

Dr. José Gabriel Ramírez Torres, Director

A través de la historia de la humanidad, la exploración de nuevas áreas viene acompañada por la construcción de representaciones de dichas áreas (mapa). En robótica móvil la construcción y/o uso de un modelo interno del ambiente (un mapa), es uno de los principales problemas en el área. Es casi imposible para un robot operar en un ambiente si no cuenta con un modelo que guíe sus acciones. En esta tesis doctoral presento el desarrollo de una estrategia de exploración multi-robot descentralizada, donde los robots operan como agentes económicos que buscan realizar el mejor aporte posible al proceso de exploración, sin que dependan de un módulo central para la coordinación del equipo. El esquema propuesto es asíncrono, es decir, no requiere que los robots establezcan ventanas de tiempo para el intercambio de información, en las que tienen que esperar a los demás miembros del equipo para intercambiar información y establecer nuevos objetivos a explorar. En mi propuesta, cuando un robot ha terminado su misión y requiere un nuevo objetivo, analiza la situación actual de la exploración y decide, sin interferir con la misión de los demás robots, la mejor acción a tomar.

Mi propuesta contempla las características de explotación para dirigir a cada robot hacia los objetivos más cercanos a él (explotando su posición actual), mientras que los

separa de los otros miembros del equipo (evitando el trabajo redundante y las interferencias entre éstos), tratando de evitar el romper con los puentes de comunicación.

La estrategia de exploración propuesta ofrece además robustez, pues el proceso de exploración no depende de un módulo central para lograr la coordinación del equipo de robots. Es posible que un robot por una avería total, resulte excluido del proceso de exploración sin comprometer el éxito de la misión del equipo. Igualmente, es posible incluir nuevos miembros al proceso de exploración cuando este último ya se encuentra muy avanzado, y los nuevos robots se integran sin ningún problema a la tarea, pues contarán rápidamente con una versión actual del estado de la exploración gracias a la cohesión de la red de comunicación. Tenemos entonces una estrategia que es flexible y, al mismo tiempo, tolerante a fallos individuales de los robots.

Los experimentos en simulación muestran que mi esquema de exploración para medios ambientes desconocidos es una alternativa viable y eficiente a los algoritmos desarrollados hasta el día de hoy dentro de la comunidad científica en robótica.

# Abstract

## **Decentralized Strategy for Multi-Robot Exploration under Limited Communication Range**

by

**Juan Carlos Elizondo Leal**

Doctor of Science in Laboratory of Information Technology

Research Centre for Advance Study, 2013

Dr. José Gabriel Ramírez Torres, Advisor

Through the history of mankind, exploration of new areas is accompanied by the construction of representations of these areas (maps). In mobile robotics building and/or using an internal model of the environment, is one of the major issues in the field. It is almost impossible for a robot to operate in an environment if it does not have a model to guide its actions. In this thesis we present the development of a novel strategy for decentralized multi-robot exploration in which each robot works as an economic agent that searches the best possible contribution to the process of exploration, without relying on a central module for team coordination. The proposed strategy is asynchronous, that is, the robots do not need to establish time windows, in which they have to wait for the other members of the team to exchange information and select new targets to explore. In our approach, when a robot has explored their goal and requires a new goal, it analyzes the current situation of exploration and makes a decision, without interfering with the mission of the other robots.

Our approach includes the characteristics of exploitation, exploration and cohesion. In the exploitation the robot is directed at goals close to it (exploiting its current position); in the exploration the robot tries to separate itself from the other team members (avoiding redundant work and interference with others); and, in the cohesion the robot tries to avoid

breaking the communication bridges. Additionally, the approach provides robustness, since the exploration process does not depend on a central module to achieve the robots team coordination.

Moreover, it is possible for a robot, completely broken, be excluded from the exploration process without compromising success of the team's mission. It is also possible to include new robots to the exploration process when the latter is already started, and the new robots are integrated seamlessly into the task, since will count quickly with a current version of the exploration status through the cohesion of the network. Then, we have a strategy that is flexible and at the same time, fault tolerant.

The simulation experiments show that our exploration strategy for unknown environments is an algorithm viable and efficient alternative to the developed algorithms until today within the robotics research community.

# Introducción

Los robots han fascinado a la humanidad por cientos de años tratando de crear máquinas a su imagen y semejanza. Algunos ejemplos son la leyenda del Titán de Prometeo y el gigante Talos de la mitología Griega (3500 a.C.); las estatuas oráculo de los egipcios, las cuales eran controladas por sacerdotes escondidos dentro de estos (2500 a.C.); los diseños de Leonardo da Vinci (1500) y los autómatas de Jacquet-Droz (1768-1774).

El término “robot” apareció por primera vez en la obra de ciencia ficción *R.U.R. (Robots Universales de Rossum)*, escrita por el checo Karel Čapek en 1920. Dicho término deriva de la palabra checa “*robota*” que significa *trabajos forzados*.

Los autómatas construidos antes del siglo XX no relacionaban los sensores, con las acciones sino que eran operados por humanos o eran máquinas repetitivas. A mediados del siglo XX la robótica se vio beneficiada por los avances en las tecnologías mecánica, control, computación y electrónica. Con nuevos diseños que relacionaron varias ciencias, se abrieron nuevas áreas de investigación, permitiendo la mejora de soluciones y nuevos conceptos, con lo cual surgió la robótica.

Los primeros robots, construidos en la década de 1960, surgieron de la unión de dos tecnologías: máquinas de control numérico y máquinas teleoperadas. La robótica creció principalmente en las fábricas, donde los robots industriales fueron introducidos para la automatización de procesos, como la pintura por rocío, la soldadura por puntos, el fresado y el ensamblaje de partes. Las fábricas se hicieron más seguras, flexibles y precisas, liberando a los humanos de tareas peligrosas y tediosas. Estos primeros robots industriales eran principalmente brazos mecánicos, siendo el más destacado de este tipo el robot UNIMATE, que se instaló en una cadena de montaje de General Motors en la Inland Fisher Guide Plant en Ewing (Nueva Jersey), en el año 1961 [63].

Posteriormente, surgió una línea de investigación basada principalmente en la inteligencia artificial. El primer proyecto de esta área fue el robot Shakey de la Universidad de Stanford [62], el cual fue uno de los primeros robots móviles. En este proyecto se unían la inteligencia artificial con las ciencias computacionales y las matemáticas aplicadas, para el desarrollo de máquinas inteligentes. Durante la década de 1980 surgieron nuevas líneas de desarrollo, abarcando desde robots exploradores, hasta la robótica de servicio, surgiendo así los robots móviles, capaces de resolver problemas fuera del ambiente industrial.

Los robots están clasificados en robots industriales y robots de servicio. Mientras que los robots industriales se emplean principalmente en ambientes de manufactura, los robots de servicio se utilizan en un rango mucho más amplio de ambientes, distintos a los bien definidos ambientes industriales [59]. Un robot de servicio es un robot que realiza tareas útiles para los humanos o equipos excluyendo las aplicaciones de automatización industrial. Dentro de la robótica de servicios podemos encontrar distintas aplicaciones como lo son la agricultura [10], la limpieza [42], el transporte [7], el cuidado de personas [35], seguridad y vigilancia [96], entre otras.

El creciente uso de robots conlleva de forma natural al planteamiento de tareas cooperativas, desarrollando equipos de robots que puedan desempeñarse de manera

conjunta de forma fiable, rápida y eficiente, resolviendo incluso problemas que le serían imposibles a un solo robot (e.g. el levantamiento de cargas pesadas). Este tipo de aplicaciones se conoce como *robótica cooperativa*.

El éxito de las comunidades de robots radica en el intercambio eficiente de información y en la coordinación entre los miembros del equipo, formando una red de agentes en comunicación. Sin embargo, a fin de tener una colaboración eficiente entre los robots, cada nodo (robot) de la red tiene que ser alcanzable en cada instante. Lo cual implica que además de sus tareas cooperativas, los robots tienen que asegurarse de mantener la conectividad de la red [28].

## **1.1 Problemas Inherentes a la Robótica Móvil**

A través de la historia de la humanidad, la exploración de nuevas áreas viene acompañada por la construcción de representaciones de dichas áreas (mapa). El primer mapa conocido se encontró cerca de Kirkuk (Irak) y se remonta a 2400-2200 a.C. [14]. La generación de mapas ha florecido con la exploración del mundo, especialmente en los últimos 500 años. Los mapas han sido usados para la exploración de nuevas áreas, navegación en aguas desconocidas, búsqueda de naufragios, incluso por diversión [69].

En robótica móvil la construcción y/o uso de un modelo interno del ambiente (un mapa), es uno de los principales problemas en el área. Es casi imposible para un robot operar en un ambiente si no cuenta con un modelo que guíe sus acciones. Una propuesta para la operación de robots es construir una arquitectura mínima que pueda reaccionar al ambiente y operar bajo un pequeño conjunto de hipótesis [13, 36]. Dicho enfoque no requiere de una representación interna del ambiente; sin embargo, las tareas que puede realizar están limitadas: por ejemplo, no puede aprender que cierta ruta se encuentra bloqueada, no puede calcular trayectos óptimos, entre otras. En general, los robots móviles

necesitan un mapa para operar en un ambiente particular, por lo cual el problema de la generación de mapas es fundamental en robótica móvil. Dichos mapas son necesarios para aplicaciones como son, el movimiento de objetos [66], la limpieza [38], la búsqueda y rescate [61], entre otras.

La generación de mapas con sistemas mono-robot necesita la solución de tres problemáticas distintas [54], las cuales son, *el mantenimiento del mapa, la localización y la planificación de trayectorias*.

El mantenimiento de mapas consiste en la correcta interpretación de la información obtenida por los sensores del robot y la generación de una representación del mundo (mapa). Esta tarea pretende responder a la pregunta *¿cómo se ve el mundo?*

Por otra parte, el problema de la localización trata de estimar la posición actual del robot relativa a un mapa dado, teniendo como fuentes de información los datos obtenidos por los sensores y las lecturas de odometría. Esta tarea pretende responder a la pregunta *¿dónde estoy?*

Por último, la planificación de trayectorias trata de resolver el problema de *¿cómo llegar al destino de manera eficiente?* Uno de los principales objetivos de la robótica móvil es el desplazamiento autónomo de los robots, es decir, la capacidad de moverse de un punto a otro, evitando las colisiones con los posibles obstáculos que se presenten.

Sin embargo, estas tres tareas no pueden realizarse independientemente una de otras, ya que antes de que el robot pueda responder a la pregunta de cómo se ve el mundo dado un conjunto de observaciones, es necesario conocer desde dónde se realizaron dichas observaciones. Igualmente es difícil estimar la posición de un robot sin un mapa. Por último, la planificación de trayectorias necesita del conocimiento del ambiente y de la posición del robot para poder alcanzar un objetivo dado.

La Figura 1.1 describe cómo las tareas de mantenimiento del mapa, localización y planificación de trayectorias pueden combinarse en problemas más complejos. La región

I, denominada SLAM (por las siglas en inglés de *simultaneous localization and mapping*) integra la localización y el aprendizaje de mapas como el problema de construir un mapa mientras se trata de localizar al robot dentro del mismo; en realidad, se trata de dos problemas interconectados: para construir los mapas necesitamos estar localizados y para estar localizados necesitamos tener un mapa, para lo que en general se utilizan métodos probabilísticos para su resolución [54, 80].



Figura 1.1: Problemas asociados con robótica móvil: (I) SLAM, (II) exploración (III) localización activa, (IV) exploración integrada [54]

La región II, exploración, asume que la información de posición es exacta y conocida en todo momento, y se enfoca en guiar al robot de una forma eficiente a través del ambiente con la finalidad de obtener una representación del medio ambiente [54].

La región III, localización activa, busca llevar al robot a un destino dado mientras se intenta mejorar la estimación en la posición del robot [54].

La región IV, exploración integrada, representa el problema de generar un mapa mientras se estima la localización y se guía al robot hacia objetivos definidos con la finalidad de generar el mapa de forma eficiente [54].

Además de los problemas descritos por Makarenko et al. [54] para los sistemas mono-robot, al tratar con sistemas multi-robot es necesario considerar también la restricción en el rango de comunicación y la coordinación de los robots. Existen muchos desafíos involucrados en el contexto de exploración multi-robot, siendo los más importantes la navegación, la coordinación y la comunicación.

**Navegación:** En exploración, el robot necesita seleccionar el mejor objetivo posible y planificar trayectorias en múltiples ocasiones hasta que se concluye la exploración. Por lo tanto, se requiere de una buena estrategia de navegación para que cada robot pueda desplazarse de manera segura y eficiente, evitando obstáculos durante su trayecto y resolviendo las interferencias con los otros robots [70, 20].

**Coordinación:** Múltiples robots pueden llevar a cabo tareas de una forma más rápida que un solo robot. Sin embargo, un sistema de  $n$  robots no es  $n$  veces más productivo que un único robot, debido a una variedad de factores: limitaciones de espacio, interferencia inter-robot, redundancia en la exploración, entre otros. La coordinación es necesaria para incrementar las ventajas de contar con un sistema multi-robot [92].

**Comunicación:** La colaboración entre robots se obtiene gracias a las capacidades de intercambio de información entre ellos, comúnmente asociadas a las tecnologías de radio comunicación. La comunicación debe estar garantizada en todo momento, pues de otra manera distintos robots podrían tener diferente información acerca del mundo. Ello plantea retos serios en el diseño e implementación de algoritmos para múltiples robots ya que, además de las tareas de colaboración, los robots tienen que asegurarse de mantener la conectividad de la red de comunicación [28].

## 1.2 Motivación

El diseño de algoritmos eficientes y robustos para la exploración multi-robot puede ser difícil debido a la naturaleza distribuida del problema, incluso cuando la comunicación se realiza a través de una infraestructura de red inalámbrica con comunicación perfecta. Sin embargo, en muchas aplicaciones es probable que la comunicación sea poco fiable ya que los ambientes pueden extenderse más allá del rango de comunicación del equipo o alguno de los robots puede fallar.

El considerar una red de comunicación limitada complica aún más las cosas, ya que al planear sus trayectorias el robot necesita comunicarse con los otros miembros del equipo, y al mismo tiempo su desplazamiento puede cambiar la topología de la comunicación en la red. Debido a que es necesario una conectividad entre los robots para hacer posible la coordinación, los algoritmos para sistemas multi-robot deben considerar durante la planificación de trayectorias la preservación de un grafo conexo entre los robots.

En los últimos años se han venido desarrollando propuestas eficientes para la exploración multi-robot; de éstas, muchas utilizan un esquema centralizado, donde un robot recupera toda la información actual y define los objetivos individuales de todo el equipo, con resultados muy buenos ya que la coordinación puede llegar a ser perfecta [15, 46]. La desventaja principal de estos enfoques es que pueden presentar un punto único de falla ya que si el módulo central falla la exploración se ve comprometida. Otras propuestas son las descentralizadas, donde los miembros del equipo toman decisiones para sí mismos basados en la información recibida, proporcionando resultados más pobres en la coordinación pero con la ventaja de que son más robustos ya que no cuentan con un punto único de falla [95, 6].

La mayor parte del trabajo previo desarrollado en exploración multi-robot asume que los robots son capaces de comunicarse con un módulo central o con el resto de los miembros

del equipo en cualquier instante de la exploración y desde cualquier posición, sin tomar en cuenta el rango de comunicación. En el mundo real es importante considerar esta restricción ya que los ambientes pueden extenderse más allá del rango de comunicación del equipo.

## 1.3 Definición del Problema

El problema de exploración multi-robot puede definirse de la siguiente manera. Dado un equipo de robots móviles con ruedas, que se desplazan a través de un medio ambiente plano desconocido y que parten de una posición conocida. Existe también incertidumbre en los desplazamiento de los robots, haciendo que gradualmente sea más difícil determinar sus coordenadas globales. Mientras se desplaza, cada robot es capaz de sensar su medio ambiente. Igualmente, cada robot puede comunicarse directamente con cualquier otro que se encuentre dentro de una región circular con centro en dicho robot, lo que representa el rango limitado de comunicación.

El problema de exploración multi-robot consiste en: *desarrollar una estrategia de coordinación eficiente, que permita a cada robot planificar sus trayectorias, considerando la incertidumbre de los desplazamientos, con el objetivo de obtener una descripción completa del medio ambiente desconocido.*

Formalmente, podemos describir el problema de la exploración multi-robot de la siguiente manera. Denotemos el índice del robot por  $r$ , el instante de tiempo por  $t$  y la localización del robot por  $x_t^r$ , donde  $x_t^r$  es un vector tridimensional, que consiste en sus coordenadas en el plano bidimensional más un valor rotacional para su orientación, de tal forma que la trayectoria del robot está descrita por una secuencia de posiciones como

$$X_T^r = \{x_0^r, x_1^r, x_2^r, \dots, x_T^r\}$$

donde  $T$  es un tiempo finito y la ubicación inicial de cada robot  $x_0^r$  es conocida.

La odometría provee información relativa entre dos ubicaciones consecutivas. Sea  $u_t^r$  la odometría que caracteriza el movimiento entre el instante actual,  $t$ , y uno anterior,  $t - 1$ . Entonces la secuencia

$$U_T^r = \{u_1^r, u_2^r, u_3^r, \dots, u_T^r\}$$

caracteriza el movimiento relativo del robot. Para movimientos libres de incertidumbres,  $U_T$  sería suficiente para recuperar la posición  $X_T$  desde la ubicación inicial.

Cada robot sensa los objetos en el ambiente. Sea  $m$  el mapa del medio ambiente. Las medidas obtenidas por los sensores de cada robot establecen información entre las características en  $m$  y la localización del robot  $x_t^r$ . La secuencia de mediciones está dada como

$$Z_T^r = \{z_1^r, z_2^r, z_3^r, \dots, z_T^r\}$$

Finalmente, cada robot puede comunicarse directamente con cualquier otro que se encuentre dentro de la región circular  $\beta$  con centro en el robot  $r$ , a fin de intercambiar información necesaria para la coordinación del equipo.

La Figura 1.3 ilustra la interacción de causalidad entre las variables involucradas en el problema de exploración multi-robot (en este caso se muestra para 2 robots).

El problema de exploración multi-robot consiste entonces en la recuperación de un modelo del mundo  $m$ , a partir de la secuencia de ubicaciones de cada robot  $X_T^r$ , de los datos de odometría  $U_T^r$  y de los datos obtenidos por los sensores embarcados  $Z_T^r$ , minimizando el tiempo de exploración, la distancia recorrida y el tiempo en que algún robot queda incomunicado.

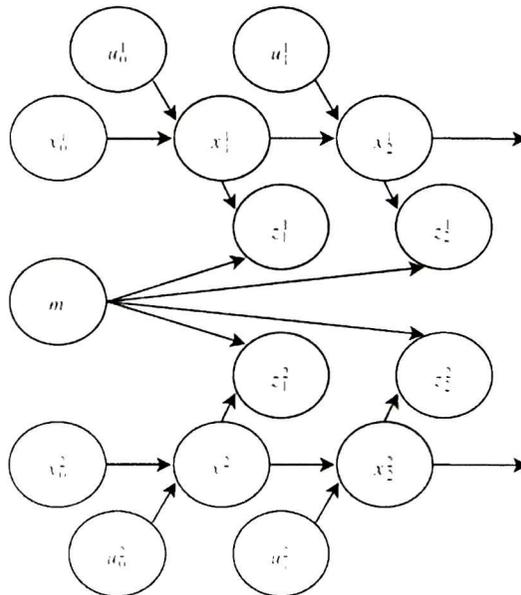


Figura 1.2: Modelo gráfico del problema de exploración multi-robot. Los arcos indican las relaciones causales.

## 1.4 Hipótesis

La hipótesis de mi trabajo doctoral la siguiente: Es posible desarrollar un esquema de coordinación eficiente para la exploración multi-robot, el cual debe de ser tolerante a fallas, asíncrono aún cuando se integren sistemas de localización y que además contemple las características de: **Explotación** con lo cual el robot se dirige a objetivos más cercanos a él (explotando su posición actual), **exploración** con lo cual el robot trate de separarse de los otros miembros del equipo (evitando el trabajo redundante y las interferencias entre éstos) y **cohesión** con lo cual los robots traten de evitar el romper con los puentes de comunicación.

## 1.5 Objetivos de la Investigación

### Objetivo general

Desarrollar una estrategia basada en economía de mercado (ofertas) que permita coordinar, de manera eficiente y descentralizada, una comunidad de robots con capacidad de comunicación en un rango limitado a la que se le ha asignado la tarea de explorar de manera completa un medio ambiente desconocido, garantizando además el resultado de la exploración aún cuando se presenten fallas en uno o varios miembros del equipo de exploración.

### Objetivos específicos

- Diseñar un nuevo algoritmo de planificación global de trayectorias, que permita a cada robot del equipo de exploración establecer ofertas realistas para cada uno de los objetivos, determinando de manera precisa la trayectoria que un robot seguiría desde su posición presente hacia el objetivo considerado, bajo el conocimiento actual del medio ambiente.

- Diseñar un esquema descentralizado de cálculo de ofertas que reflejen tres características principales:
  - Explotación, donde cada robot trate de alcanzar para sí los objetivos que le son más cercanos.
  - Exploración, donde cada robot trate de alejarse del resto del equipo
  - Cohesión, donde cada robot trate de desplazarse a una zona en la que pueda mantener comunicación con al menos otro miembro del equipo
- Integrar un método de localización basado en la odometría y en sensores láser de un robot a ruedas.

## **1.6 Descripción del Enfoque Propuesto**

Para resolver el problema de exploración multi-robot propongo un algoritmo completamente descentralizado, que además considera la restricción en el rango de la comunicación. Mi algoritmo está basado en un proceso de ofertas de mercado, en el cual cada robot calcula las ofertas de manera independiente, buscando alcanzar el mayor aporte posible al equipo en su conjunto. Cuando un robot alcanza su objetivo, el robot toma una decisión para sí mismo, involucrando a cada uno de los miembros del equipo así como el rango de comunicación, bajo un esquema descentralizado y sin la necesidad de un módulo central.

Mi enfoque genera el mapa del medio ambiente desconocido de manera eficiente, a través de una coordinación entre los miembros del equipo basada en un mecanismo de mercado (ofertas y subasta). En este esquema, los robots realizan ofertas para cada uno de los objetivos (fronteras disponibles) en función de su posición con respecto al objetivo, de su posición con respecto a los demás miembros del equipo y del rango de comunicación

$\beta$ . Con este enfoque logramos modelar tres propiedades muy interesantes, inspiradas en las heurísticas de optimización multi-objetivo: la **explotación** de la posición actual, pues cada robot trata de obtener para sí los objetivos que le son más cercanos; la **exploración**, donde cada robot trata de alejarse del resto del equipo para reducir la redundancia; y la **cohesión**, ya que cada robot trata de desplazarse a una zona en la que pueda mantener comunicación con al menos otro miembro del equipo.

Para cuantificar la eficiencia de mi propuesta, la presente investigación se apoya en las siguientes métricas:

- El tiempo de exploración, es decir, el tiempo que toma al equipo de robots obtener un mapa completo del ambiente. Lo ideal es que entre más robots use el tiempo de exploración debe reducirse en un factor de  $(1/n)$ , es decir que el tiempo total de exploración para 2 robots debe ser  $1/2$  del tiempo que le toma a un solo robot, al usar 3 robots dicho tiempo debe ser  $1/3$  del tiempo que le toma a un solo robot y así sucesivamente.
- El tiempo total durante el cual al menos un robot quedó fuera del rango de comunicación, es decir tiempo total que la red de comunicación entre los robots se mantuvo rota. Este tiempo se busca minimizar para a su vez disminuir el riesgo de pérdida de información, dado que si un robot se avería durante el tiempo en que permanece aislado, toda la información nueva que haya podido recolectar sobre el ambiente se habría perdido, pues ningún otro miembro del equipo posee dicha información.
- El tiempo máximo en el que algún robot se mantuvo fuera del rango de comunicación. Este tiempo se busca minimizar, es decir que si un robot sale del rango de comunicación, este trate de reingresar a la red lo más pronto posible ya que durante dicho tiempo el mapa individual del robot aislado no integró información

recabada por los otros robots y que a su vez los otros robots no integraron la información recabada por el robot aislado.

- La separación máxima de un robot una vez que se sale del rango de comunicación. Al igual que la métrica anterior esta distancia se busca minimizar, con esta métrica se busca analizar cuando un robot se aísla y qué tanto se aleja del resto del equipo.
- La distancia recorrida por todos y cada uno de los miembros del equipo. En este caso se busca que la distancia no crezca con el número de robots, es decir que la distancia total recorrida necesaria para completar la exploración para  $n$  robots debe ser la misma que le toma realizar la exploración con un solo robot, lo cual es un indicativo importante para determinar la energía consumida y la redundancia durante la exploración.
- La consistencia de los mapas en cada momento, es decir, cuán similares son los mapas locales que contiene cada robot. Con lo cual podemos verificar la información que se perdería en caso de presentarse una falla total en alguno de los robot.

En mi propuesta, cada robot construye su propio mapa e incorpora las observaciones realizadas por los otros miembros del equipo. Los mapas se representan por medio de mallas de evidencia, en las que cada celda puede tomar uno de tres estados (ocupado, libre o desconocido). Además, aplico el concepto de frontera [94], la cual se define como el límite entre el área abierta y la zona inexplorada. Un robot considera que el mapa está terminado cuando no existen fronteras alcanzables para él, es decir que puede haber aún zonas sin explorar pero que el robot no tiene manera de llegar ahí.

Las características principales de mi propuesta son:

1. Cada robot construye y actualiza su propio mapa, independientemente de lo que esté haciendo el resto del equipo. Cuando un robot ha terminado la exploración

que tenía asignada, solicita a los miembros más cercanos sus respectivos mapas, e integra esta información a su propio mapa. Este punto agrega tolerancia a fallas ya que cada robot mantiene su mapa.

2. Los mapas son representados mediante un esquema de descomposición exacta de celdas (malla de evidencia).
3. Las posiciones de los robots y de los obstáculos están referidos hacia el origen de un sistema cartesiano global.
4. Los robots extraen información del medio ambiente a través de los sensores embarcados, la cual es utilizada para actualizar el mapa local durante la exploración.
5. Cada robot transmite la información referente a su mapa solamente cuando le es solicitada por otro miembro del equipo y está dirigida únicamente hacia ese otro robot. No hay *broadcasting* de la información.
6. La estrategia de exploración está basada en un proceso de auto-ofertas, bajo un esquema descentralizado y tolerante a fallos totales (pérdida total) de alguno de los miembros del equipo. Cada robot calcula su oferta, estima las decisiones que los otros miembros del equipo tomarían y selecciona una frontera para sí mismo.
7. La oferta para cada robot es calculada de tal manera que se optimizan tres comportamientos principales: la exploración, la explotación y la cohesión del equipo.

Cada robot trabaja bajo las siguientes condiciones:

1. Inicialmente se desconoce completamente el ambiente.
2. Se tiene un rango de comunicación limitado  $\beta$ .
3. Se conocen las posiciones iniciales de cada robot dentro de un marco de referencia.

4. Se pueden generar errores en la posición de los robots durante su desplazamiento. Estos errores son corregidos al resolver el problema de localización.
5. El rango de los sensores embarcados es limitado.

Finalmente, las principales aportaciones de este trabajo son:

1. Una estrategia de exploración multi-robot descentralizada y tolerante a fallas, la cual modela tres propiedades muy interesantes, las cuales son inspiradas en las heurísticas de optimización multi-objetivo: la **explotación** con lo cual el robot se dirige a objetivos más cercanos a él (explotando su posición actual), la **exploración** con lo cual el robot trata de separarse de los otros miembros del equipo (evitando el trabajo redundante y las interferencias entre éstos) y la **cohesión** con lo cual los robots tratan de evitar el romper con los puentes de comunicación.
2. Un nuevo algoritmo para la planificación global de trayectorias, el cual proporciona la distancia exacta para cada celda y la mejor trayectoria como una secuencia de sub-objetivos, sin necesidad de un post-procesamiento. Las trayectorias y las distancias obtenidas con mi algoritmo de planificación nos permiten construir esquemas de coordinación sobre la base de una medida de utilidad más exacta.

## 1.7 Esquema de la Tesis

Los capítulos restantes incluyen su propio marco teórico e introducción y se describen a continuación.

**Capítulo 2 Estado del Arte.** Este capítulo muestra el trabajo previo en exploración multi-robot y ha sido dividido en cuatro partes: La primera parte ofrece un estudio de las principales propuestas en exploración clásica; la segunda describe las principales

propuestas en SLAM multi-robot, la tercera describe las principales propuestas en las que se toma en cuenta la incertidumbre en la localización y por último se hace una comparativa de las propuestas más relevantes para mi trabajo de investigación.

**Capítulo 3 - EEDT: Un Nuevo Planificador de Trayectorias.** Este capítulo presenta un resultado original para la planificación global de trayectorias en mapas de mallas de ocupación, el cual fue necesario desarrollar para obtener un mecanismo de ofertas más eficaz, pues las ofertas se basan en distancias. El trabajo descrito en este capítulo dio como resultado una publicación en una revista incluida en el *ISI Thomson Science Citation Index Expanded*.

**Capítulo 4 - Exploración Multi-Robot Usando Auto-Ofertas bajo la Restricción en el Rango de la Comunicación.** Este capítulo detalla el desarrollo y el desempeño de mi estrategia de solución para el problema de exploración multi-robot bajo la restricción en el rango de la comunicación.

**Capítulo 5 Conclusiones y Trabajo Futuro** Presenta las conclusiones y recomendaciones para un trabajo futuro de los dos trabajos presentados en el capítulo 3 y 4, respectivamente.

# Estado del Arte

Este capítulo ofrece un panorama general de las principales propuestas en generación de mapas empleando múltiples robots, discutiendo sus fortalezas y sus debilidades. Me enfoco principalmente en las propuestas de exploración clásica multi-robot, SLAM multi-robot y exploración multi-robot con incertidumbre en la posición.

La exploración multi-robot se enfoca principalmente en guiar a cada robot de forma eficiente con la finalidad de obtener una representación del medio ambiente asumiendo, que la localización es perfecta y conocida en todo momento. El problema de SLAM integra la localización y el aprendizaje de mapas como el problema de construir un mapa mientras se trata de localizar al robot dentro del mismo. Finalmente, el problema de exploración multi-robot con incertidumbre en la posición representa el problema de generar un mapa mientras se estima la localización y se guía al robot hacia objetivos definidos con la finalidad de generar el mapa de forma eficiente [54]. También es necesario tomar en cuenta la restricción en el rango de la comunicación. Para algunas aplicaciones es necesario que los robots inicien con marcos de referencia distintos, siendo necesario el uso de algoritmos

para resolver la fusión de mapas [19, 9, 33, 26]. Sin embargo, en esta tesis asumo que la posición inicial es conocida, ya que en mi caso consideraré que el equipo de robots aún no se despliega en el terreno al inicio de la exploración.

A continuación se analizan las principales propuestas para la generación de mapas con múltiples robots, que comparten rasgos comunes con el enfoque desarrollado en este trabajo de tesis.

## **2.1 Exploración Clásica Multi-Robot**

Las técnicas de exploración clásicas dirigen al robot hacia objetivos que le permitan adquirir la mayor cantidad de información como sea posible en un corto tiempo. Sin embargo, en estos enfoques la incertidumbre asociada a la localización no afecta la planificación ni la generación de mapas de los robots. Muchas propuestas para la exploración de ambientes desconocidos con un solo robot han sido ampliadas al caso multi-robot [2, 29, 79, 1, 3, 5, 21, 47, 25, 64, 97]. En 1997, Yamauchi [94] sentó las bases para la exploración basada en fronteras, las cuales se definen como los límites entre el espacio explorado y el espacio inexplorado, donde el espacio explorado ha sido sentido por un sensor de rango. La base de esta idea es la de mantener al robot moviéndose hacia la frontera más cercana. Una técnica muy usada es la exploración basada en fronteras, que consiste en extraer las fronteras entre áreas conocidas y desconocidas [94, 95, 17] y visitar los lugares cercanos a las zonas inexploradas. Algunas de esas propuestas se han enfocado en la coordinación sin tomar en cuenta la restricción en el rango de la comunicación. Si la distancia entre los robots se hace muy grande el puente de comunicación podría romperse y los robots podrían explorar un área que ya ha sido explorada o en caso de la falla de algún robot aislado, su información se perdería permanentemente.

### 2.1.1 Exploración Clásica Multi-Robot con Comunicación Ideal

En [95], Yamauchi extiende el concepto de fronteras a un algoritmo de exploración multi-robot, donde los robots intercambian la información de los mapas cada vez que arriban a una nueva frontera. Esta propuesta brinda la capacidad de tolerancia a fallos, ya que si alguno de los robots falla, la exploración aún puede ser terminada. Sin embargo, no existe una coordinación explícita, ya que varios robots pueden llegar a cubrir la misma zona, con lo cual puede haber una interferencia entre varios miembros del equipo.

Simmons et al. [76, 15] desarrollaron una propuesta para la exploración multi-robot en la cual usan el concepto de fronteras y un protocolo simple de ofertas. El equipo requiere un agente central para evaluar las ofertas de todos los robots para así poder obtener la mayor información mientras se trata de reducir el costo de la distancia en los trayectos. Al ser una propuesta que requiere de un módulo central, carece de la tolerancia a fallas, ya que si el robot central falla la tarea de exploración se ve comprometida.

Zlot et al. [98] presentan una propuesta para la exploración multi-robot en la cual explotan una arquitectura de mercado. El equipo de robots continuamente negocia con los otros miembros del equipo mejorando así sus planes actuales e intercambiando información acerca de cuales regiones pueden y no pueden ser cubiertas. Su algoritmo puede mejorar la fiabilidad, robustez y eficiencia de la exploración; sin embargo, el algoritmo de generación de objetivos que usan no es tan eficiente como los basados en fronteras, y el proceso de negociación es complejo.

Baglietto et al. [6] presentan una propuesta descentralizada donde aplican el concepto de entropía y fronteras, donde la entropía cuantifica la ganancia de información asociada con cada frontera.

Poernomo y Shell Ying [68] proponen la separación de los robots en la selección de la frontera, con lo cual se reducen las posibilidades de interferencias entre los miembros del equipo. Sin embargo, asumen que el tamaño de los robots es lo suficientemente pequeño,

tal que las colisiones no pueden existir.

En [31] y [32] Elizondo-Leal et al. presentan una propuesta totalmente descentralizada para la exploración multi-robot en la que se presenta el concepto de auto-ofertas, en la cual las ofertas son calculadas por cada robot, la función de costo incluye la distancia del robot actual hacia la frontera, la distancia de los otros robots y sus objetivos. Todas estas propuestas asumen que la localización es perfecta y conocida en todo momento, además de que no consideran el impacto de tener un rango de comunicación limitado.

### **2.1.2 Exploración con Rango de Comunicación Limitado**

En la mayoría del trabajo previo se asume que la comunicación entre los robots es perfecta, lo cual hace que sus algoritmos no consideren manejar cortes en los puentes de comunicación. Burgar et al. en [16] y posteriormente Stachniss en [78], desarrollaron un algoritmo de exploración multi-robot en la cual simultáneamente considera la utilidad de explorar áreas y el costo de alcanzarlas, ellos usan mallas de evidencia para representar el mundo y el concepto de fronteras para detectar los objetivos potenciales. Además, investigan el impacto del rango de comunicación limitado. Ellos proponen el uso de un módulo central seleccionado dinámicamente, el cual recibe los mapas locales y los combina en un solo mapa que es enviado a todos los miembros del equipo. Es una buena propuesta; sin embargo, al usar un módulo central sufre de tener un punto único de falla. Stachniss et al. en [77] extienden este trabajo y consideran la información semántica del lugar, aplicando un clasificador entrenado con el algoritmo AdaBoost [34]. Esta propuesta requiere un aprendizaje previo, lo cual es una desventaja ya que se requiere de tiempo para el entrenamiento, además de que considera solamente lugares similares a los aprendidos durante dicho entrenamiento.

Sheng et al. [75] presentan un algoritmo de exploración basado en un modelo de ofertas distribuido. Ellos abordan el problema causado por un rango de comunicación limitado.

Los autores introducen una medida de cercanía, por lo que los robots tienden a mantenerse cercanos entre ellos. Para llevar a cabo la coordinación, los robots periódicamente envían su mejor oferta a los otros miembros del equipo dentro de la sub red. Al final de un tiempo constante, el robot que provee la mejor oferta es declarado como ganador y se dirige hacia el objetivo ganado. Este proceso es reiniciado para todos los otros robots.

Rooker y Birk [72] proponen una coordinación centralizada que garantiza que, durante la exploración, ningún robot perderá comunicación con el resto de los miembros del equipo. Para alcanzar este objetivo, un módulo central recolecta las posiciones actuales de cada robot y genera un conjunto de futuras configuraciones para el equipo completo (por ejemplo, las posibles futuras posiciones de los robots). Debido al gran número de combinaciones posibles, no todas las configuraciones pueden ser consideradas, pero sí una cantidad limitada de éstas. Entre este número de configuraciones generadas, el módulo central escoge la mejor de acuerdo a la función de utilidad. Esta función penaliza cuando la futura posición está ocupada por un obstáculo o cuando se saldría del rango de comunicación de algún robot. Esta propuesta tiene un punto único de falla al depender de un módulo central que se encarga de coordinar al equipo de robots.

Vazquez y Malcolm en [86] proponen un algoritmo de exploración descentralizado basado en la selección de diferentes comportamientos: evitando obstáculos, manteniendo la conectividad de la red, explorando hacia una frontera. Esta selección toma en cuenta la condición actual de la red, la cual es conocida por cada robot gracias a que periódicamente se intercambian mensajes. Para alcanzar la conectividad de la red, cada robot analiza la topología de la red y realiza una distinción entre una articulación simple y un puente.

Doniec et al. en [28] presentan una propuesta descentralizada en la que se enfocan en separar los robots y a la vez mantenerlos dentro del rango de comunicación. Dicha propuesta está basada en el problema distribuido de satisfacción de restricciones (disCSP) en el que cada robot construye una parte del disCSP mediante la adición de restricciones

con el resto del equipo. El objetivo de cada robot es proporcionado por la resolución del problema disCSP.

Pal et al. en [65] proponen un esquema híbrido para la exploración, en la cual buscan mantener la conectividad entre los robots, con la finalidad de tener un mapa preciso del ambiente en todo momento. Para mantener la conectividad, desarrollaron un protocolo de interacción líder-seguidor (LFIP). La selección del líder se realiza de manera dinámica tomando en cuenta la topología de la red.

Mukhija et al. [60], Visser et al. [87] y de Hoog et al. [24] proponen estrategias en las cuales buscan mantener la conectividad con una estación base. Mukhija et al. [60] proponen una estrategia centralizada de exploración, en la cual los robots pueden tomar el rol de encargado de la conexión o de explorador del ambiente, asegurando que en cada momento todos los robots se mantienen en contacto con la estación base de manera directa o a través de los robots de conexión. En [87, 24] los autores presentan una propuesta en la que los miembros del equipo planean explorar más allá de los límites de comunicación y periódicamente se reúnen en puntos de cita para intercambiar información.

## 2.2 SLAM Multi-Robot

SLAM es el proceso en el que concurrentemente se construye un mapa del ambiente y se usa este mapa para obtener una estimación de la localización del robot. En la literatura se distinguen dos formas principales del problema SLAM. Una es conocida como el problema de SLAM completo, la cual consiste en calcular la probabilidad del trayecto completo y del mapa a partir de un conjunto de datos obtenidos de los sensores, es decir se procesan todos los datos al mismo tiempo. La segunda es conocida como el problema de SLAM en línea, en la cual se busca recuperar la localización actual del robot y actualizar el mapa en cada instante de tiempo, en lugar de recuperar el camino completo [82].

Para solucionar el SLAM multi-robot se tienen dos enfoques distintos, uno en el que se asume que la posición inicial es conocida y otro en el que la posición inicial del robot no es conocida a priori. Al utilizar múltiples robots también es necesario considerar la fusión de mapas, la cual consiste en generar un mapa consistente del ambiente con los datos recolectados por los distintos robots [67].

Williams et al. [90] extienden el algoritmo CLSF (*Constrained Local Submap Filter*), diseñado inicialmente en [91] para el problema de SLAM mono-robot, al problema de SLAM multi-robot en línea con posiciones iniciales desconocidas. En esta propuesta se mantiene un mapa global para los robots y cada robot mantiene un mapa local de características en su vecindad inmediata. Cada uno de los mapas locales tiene su propio marco de referencia, donde su posición relativa al mapa global es conocida. Los robots actualizan el mapa global con las características presentes en su mapa local. Un nuevo marco de referencia es creado utilizando la última posición del robot como estimación de la posición relativa del marco local dentro del marco global de referencia. Este método además establece un nuevo marco de referencia para los robots en los cuales la posición inicial es desconocida, mediante la construcción de una correspondencia entre conjuntos de mapas locales, el mapa global y usando pares de características para determinar y estimar la rotación y translación del marco de referencia. En esta propuesta no se consideran restricciones en la comunicación ni el problema de exploración.

Thrun y Liu en [83] extienden la técnica SEIF (*Sparse Extended Information Filters*) propuesta inicialmente para sistemas mono-robot [81], para SLAM multi-robot empleando técnicas para establecer correspondencia entre los mapas intercambiados por los distintos robots. Las incertidumbres en el mapa son manejadas por el uso de GMRFs (*Gaussian Markov random fields*). Este método construye mapas aun y cuando la posición de inicio es desconocida y las marcas son ambiguas. Todas las actualizaciones son agregadas y hechas localmente. Cada robot actualiza su mapa local y las marcas previamente detectadas en él.

La fusión de mapas entre dos robots es realizada por la búsqueda de una transformación de coordenadas (translación y rotación) relativa entre los dos mapas, y una lista de marcas correspondientes. Esto alinea los dos mapas para obtener uno solo, mediante un algoritmo basado en árboles para búsqueda de lugares similares en las configuraciones de las marcas locales. En esta propuesta no se toman en cuenta las limitantes en comunicaciones ni el problema de exploración.

Andrade-Cetto et al. [4] resuelven el problema de SLAM multi-robot usando una técnica de control no lineal llamada Linealización por Retroalimentación sobre el filtro extendido de Kalman (EKF) y un sistema de control que genera los comandos necesarios para seguir, con un alto nivel de precisión, la trayectoria planificada. La linealización por retroalimentación utiliza transformaciones algebraicas para convertir el sistema dinámico no lineal de los robots en uno lineal. Esta linealización es alcanzada por aproximaciones de estado exactas y retroalimentación del sistema dinámico. Esta propuesta no considera restricciones en comunicación ni el problema de exploración.

Howard en [37] extiende el filtro de partículas Rao-Blackwell, diseñado inicialmente para el problema de mono-robot SLAM, para solucionar el problema de multi-robot SLAM, donde la posición relativa de los robots no es conocida a priori. Para resolver el problema, este método asume que los robots pueden detectarse e identificarse entre ellos, y miden la posición relativa de los otros robots cuando cuando se encuentran. Una vez que la posición relativa es determinada, el filtro de partículas es inicializado usando las posiciones determinadas. Entonces el mapa global es actualizado usando las observaciones de ambos robots. Este método utiliza estabilización de odometría por láser, en la cual los datos del láser son usados para corregir los valores de odometría estimados. Esta propuesta no considera restricciones en comunicación ni el problema de exploración.

Wang et al. en [89] proponen una solución para el problema de SLAM multi-robot en el cual la posición inicial de cada robot es desconocida. En esta propuesta cada robot

genera un mapa local usando el algoritmo EKF SLAM; estos mapas son fusionados por intervalos de tiempos en un mapa global y se actualiza el mapa local. La fusión de mapas es llevada a cabo por medio del algoritmo D-SLAM [88]. En esta propuesta se soluciona sólo el problema de SLAM multi-robot sin tomar en cuenta la exploración ni las restricciones en comunicación.

Tong et al. en [84] extienden el SLAM basado en EKF para sistemas mono-robot a sistemas multi-robot. En esta propuesta el algoritmo EKF es aplicado para estimar las posiciones de los robots y las posiciones de las marcas. Utilizan un mapa global para representar el ambiente. Las observaciones de todos los robots son utilizadas para actualizar la localización de cada uno de estos y las marcas en el mapa. De acuerdo a esta información los robots seleccionan objetivos que beneficien al equipo completo. Esta propuesta soluciona el problema de SLAM multi-robot, sin embargo no se toman en cuenta limitantes en comunicación.

Dissanayake et al. en [27] proponen el algoritmo MRICP (*Map Referenced Iterative Closest Point*) para solucionar el problema de SLAM online para un solo robot, en el cual hace uso de un sensor láser y de la odometría del robot. Este método incorpora el algoritmo ICP [8] para el emparejamiento. Este emparejamiento se hace entre puntos que son obtenidos de un mapa de evidencia y los datos adquiridos recientemente por el sensor láser lo cual resulta en un mapa preciso del ambiente.

## 2.3 Exploración Multi-Robot con Incertidumbre en Localización

La exploración con incertidumbre en localización representa el problema de generar un mapa mientras se estima la localización y se guía al robot hacia objetivos definidos con la finalidad de generar el mapa de forma eficiente. Al incluir múltiples robots al problema de

exploración con incertidumbre en localización se debe además solucionar el problema de fusión de mapas con incertidumbre en la posición para que una vez que se ha fusionado el mapa se pueda seleccionar el mejor objetivo para cada robot. Además de que es necesario resolver el problema de la restricción en el rango de comunicación.

Konolige et al. en [48] presentan una propuesta donde los robots de manera individual generan el mapa hasta que se encuentran. Una vez que se encuentran, un robot recibe los datos de los sensores del otro robot y estima la posición relativa de estos en su propio mapa. En caso de incertidumbre, el robot construye una hipótesis de su posición y trata de reunirse en la localización construida. Si no se pueden reunir, la hipótesis es rechazada y el robot continúa con la generación de hipótesis acerca de su posición relativa. Una vez que se pueden reunir, los robots unen los mapas locales en base a la información geométrica que contienen, entonces intercambian mapas y continúan con la exploración coordinada. Además, en esta propuesta se considera un rango de comunicación limitado. Sin embargo, se enfocan en el problema de fusión de mapas cuando se desconoce la posición inicial de los robots sin tomar en cuenta errores en odometría.

Tovar et al. en [85] presentan una propuesta centralizada en la cual utilizan una función de utilidad para la selección del objetivo, en la cual se consideran la utilidad para la exploración y localización del trayecto completo hasta alcanzar el objetivo. El orden de la navegación entre objetivos es evaluado mediante un árbol de decisión considerando las distintas utilidades. Esta propuesta soluciona el problema de exploración y considera a su vez el problema de localización, sin embargo no considera la limitante en el rango de comunicación.

Juliá et al. en [44, 43] proponen un modelo híbrido (reactivo/deliberativo) descentralizado para la selección de los objetivos y un proceso centralizado para resolver el problema de localización y generación de mapa (SLAM). Al tener un proceso centralizado cuenta con un punto único de falla, además de que no considera la limitante en el rango

de la comunicación.

En [45] Jimenez-Gonzalez et al. y de Hoog en su tesis doctoral [23] extienden el trabajo presentado en [24], donde los miembros del equipo planean explorar más allá de los límites de comunicación, los robots periódicamente se reúnen en puntos de cita para intercambiar información. La localización de cada robot la solucionan usando la propuesta MRICP (*Map Referenced Iterative Closest Point*) [27] con lo cual obtienen buenos resultados en la calidad del mapa generado.

## 2.4 Comparación de las Principales Propuestas

Esta sección proporciona una comparación de las propuestas que son más relevantes para las ideas discutidas en esta tesis. Los aspectos que se consideran son los siguientes:

**Coordinación:** ¿La coordinación del equipo se lleva a cabo de una manera centralizada, descentralizada o híbrida?

- **Centralizada:** Un módulo central controla los movimientos del equipo.
- **Descentralizada:** Los miembros del equipo toman decisiones para sí mismos basados en la información recibida.
- **Híbrida:** La propuesta contiene elementos centralizados y descentralizados.

**Comunicación:** ¿La propuesta asume un rango de comunicación infinito o limitado?

- **Infinito:** la propuesta asume que todos los robots pueden comunicarse con cualquier otro robot en todo momento.
- **Limitado:** la propuesta considera el rango de la comunicación a lo largo de la exploración.

**Tolerancia a fallas:** ¿La propuesta garantiza la exploración completa del ambiente, aún cuando algún miembro del equipo falle?

**Localización:** ¿La propuesta considera el problema de la incertidumbre en la posición?

- **Centralizada:** La localización se realiza en un módulo central.
- **Descentralizada:** Cada miembro del equipo lleva a cabo su propia localización.
- **No:** No se toma en cuenta el problema de la incertidumbre en la posición.

**Métricas de evaluación:** ¿Cuáles son las variables medidas para evaluar sus resultados?

**Comparación:** ¿Con qué propuesta se comparan los resultados obtenidos?

En la Tabla 2.1 se resume la comparación entre las propuestas más relevantes para esta tesis.

## 2.5 Conclusiones

En este capítulo hemos presentado las principales propuestas relacionadas a mi trabajo de tesis, las cuales fueron divididas en los tres principales enfoques: La exploración clásica multi-robot, el SLAM multi-robot y la exploración multi-robot con incertidumbre en localización.

De la Tabla 2.1 se puede apreciar que muchas de las propuestas se comparan con la propuesta presentada por Yamauchi [95], además, ninguna de las propuestas encontradas en el estado del arte cumple con todas las características deseables descritas en la presente tesis (Coordinación y localización descentralizados trabajando bajo el rango de comunicación limitado y que sea además tolerante a fallas).

Propuesta	Coordinación	Rango de Comunicación	Tolerante a fallas	Localización	Métricas de evaluación	Comparación
Yamauchi [95]	Implícita	Infinito	Si	No	Tiempo de exploración	-
Simmons et al. [76, 15]	Centralizada	Infinito	No	No	Tiempo de exploración	Yamauchi [95]
Zlot et al. [98]	Descentralizada	Infinito	Si	No	Área cubierta /Distancia recorrida	-
Baglietto et al. [6]	Descentralizada	Infinito	Si	No	Tiempo de exploración	Yamauchi [95]
Poernomo y Shell Ying [68]	Centralizada	Limitado	No	No	Tiempo de exploración	Yamauchi [95] Burgard et al. [15]
Elizondo et al. [32, 31]	Descentralizada	Infinito	Si	No	Tiempo de exploración, Distancia recorrida	Yamauchi [95]
Burgard et al. [16, 78, 77]	Híbrida	Limitado	Si	No	Tiempo de exploración, tiempo de cómputo	Yamauchi [95]
Sheng et al. [75]	Descentralizada	Limitado	Si	No	Tiempo de exploración, Distancia recorrida, Máxima distancia entre dos robot	-
Rooker and Birk [72]	Centralizada	Limitado	No	No	Tiempo de exploración	-
Vazquez et al. [86]	Descentralizada	Limitado	Si	No	Tiempo de exploración	Yamauchi [95]
Doniéc et al. [28]	Descentralizada	Limitado	Si	No	Tiempo de exploración, Cantidad de mensajes intercambiados	-
Pal et al. [65]	Centralizada	Limitado	No	No	Distancia recorrida, Energía consumida	-
Tovar et al. [85]	Centralizada	Limitado	No	Centralizada	Distancia recorrida, Cantidad de paradas, Cantidad de objetivos sensados, Grados rotados	-
Juliá et al. [44, 43]	Descentralizada	Infinito	No	Centralizada	Tiempo de exploración, Calidad del mapa	Yamauchi [95]
Método propuesto (Capítulo 4)	Descentralizada	Limitado	Si	Descentralizada	Tiempo de exploración, Tiempo total fuera del rango de comunicación, Tiempo máximo fuera del rango de comunicación, Máxima distancia fuera del rango de comunicación, Distancia total recorrida	Yamauchi [95], Burgard et al. [16, 78]

Tabla 2.1: Comparación de las propuestas más relevantes para esta tesis.

# 3

## EEDT: Un Nuevo Planificador de Trayectorias

En la exploración multi-robot es necesario seguir trayectos para alcanzar los objetivos, por lo cual es conveniente que dichos trayectos sean generados de forma eficiente para poder así minimizar el recorrido de cada robot. En mi propuesta las distancias proporcionadas por el planificador de trayectorias sirven como medida de utilidad.

En este capítulo se aborda el problema de planificación de trayectorias en descomposición exacta de celdas, generando una transformación de distancia exacta para todas las celdas, además se identifica la próxima mejor celda visible hacia el objetivo.

### 3.1 Introducción

La planificación global de trayectorias es uno de los principales problemas en la robótica móvil, en el cual se busca encontrar una secuencia de desplazamientos dentro del medio

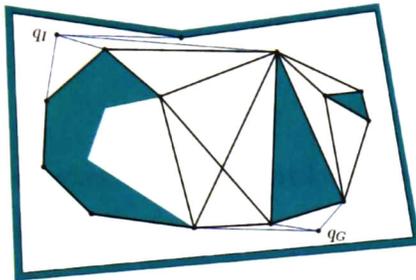
ambiente desde un punto inicial hasta un punto final, tal que el robot no colisione con algún obstáculo. Dentro de la exploración multi-robot, la planificación global de trayectorias es un tema fundamental, ya que de ella depende que se alcancen de forma eficiente los objetivos dados. Además, la distancia estimada por el planificador hacia cada uno de los objetivos se toma como una medida de utilidad en muchos algoritmos de exploración [94, 95, 76, 16, 77].

El problema de planificación global de trayectorias se puede definir como *dado un robot y una descripción del ambiente, planificar un camino libre de colisiones entre dos puntos específicos y satisfacer cierto criterio de optimización* [93]. Dicho problema es PSPACE-Completo en general, y para algunos casos es NP-Completo [18].

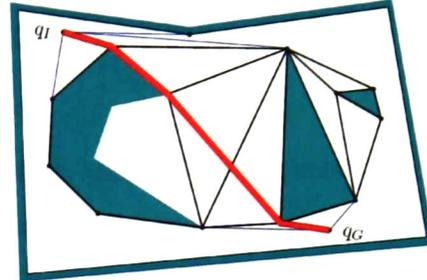
Una buena técnica para la planificación global de trayectorias en ambientes continuos es la propuesta basada en grafos de visibilidad, en la que el medio ambiente está modelado de manera poligonal. El conjunto de nodos del grafo de visibilidad está compuesto por el punto de inicio, el punto final y los vértices de todos los obstáculos [53]. Dos nodos están conectados por una arista si y sólo si existe una línea de visión entre los dos vértices DEFINIR VISIBILIDAD. El camino más corto en un grafo de visibilidad es entonces el camino más corto de todo el ambiente. Sin embargo, resolver el problema de planificación de trayectorias con grafos de visibilidad puede ser lento cuando se consideran ambientes complejos. En la Figura 3.1 se muestra un ejemplo de la planificación de trayectorias basada en grafos de visibilidad, donde  $q_I$  es el nodo de inicio,  $q_G$  es el nodo objetivo, las zonas sombreadas indican el área ocupada, las aristas indican trayectos entre vértices y la línea en rojo indica el camino óptimo.

El problema de planificación universal de trayectorias puede definirse como: *Planificación es la selección de objetivos definidos para todas las posibles situaciones* [74, 56].

Los métodos clásicos para la planificación global de trayectorias son variaciones de



(a) Para resolver la trayectoria, se conectan  $q_I$  y  $q_G$  a todos los vértices visibles, y se lleva a cabo una búsqueda en el grafo



(b) El trayecto más corto en el grafo de visibilidad es el camino más corto entre  $q_I$  y  $q_G$

Figura 3.1: Grafo de visibilidad para la planificación de trayectorias (las zonas verdes indican el área ocupada).

algunas propuestas generales: *Roadmaps*, Descomposición de Celdas, Campos Potenciales, y Programación Matemática [55].

El principio de la descomposición exacta de celdas es, primero descomponer el espacio del ambiente  $m$  en una colección de regiones no translapables, llamadas celdas [50] en la cual cada celda de la malla se encuentra en el estado libre o en el estado ocupado, obteniéndose la descripción del espacio libre  $m_{free}$  y el espacio ocupado  $m_{obs}$ .

Los algoritmos de transformación de distancia (también llamados de frente de onda) son algoritmos de propagación de etiquetado de trayectorias en descomposición exacta de celdas, los cuales asignan un valor de distancia a cada celda a través del espacio con respecto a una o múltiples celdas iniciales, marcadas con cero. Las celdas marcadas como cero son los objetivos y la transformación de distancia va a incrementar el valor de las celdas vecinas generando así frentes de onda con las distancias cada vez mayores [39].

## 3.2 Transformación de Distancia

Los algoritmos de transformación de distancia aparecieron primero en procesamiento de imágenes como una técnica para describir la forma de las imágenes [73]. Para los propósitos de planificación de trayectorias libre de colisiones, Jarvis [39] aplica esta idea para propagar las distancias hacia los objetivos hasta rellenar todos los espacios libres. Dichas distancias fluyen alrededor de los obstáculos mediante un método de rastreo de trama, que puede requerir varios pasos para garantizar que se alcancen todos los espacios libres en los mapas complejos. El algoritmo de rastreo de trama se realiza aplicando una ventana de pesos para las operaciones entre celdas vecinas, en el cual se busca minimizar el actual valor de la distancia asignado a la celda central de dicha ventana por medio de la comparación del valor de la distancia actual con el de la celda vecina más el valor asignado a este vecino. El algoritmo básico en un espacio bidimensional es el siguiente:

Dado un mapa binario  $I$  de dimensiones  $k \times l$  definido como

$$I : \Omega \subset \mathbb{Z}^2 \rightarrow \{0, 1\}$$

, siendo  $\Omega$  la matriz de celdas que representa el espacio de trabajo del robot:

$$\Omega = \{0, 1, \dots, k\} \times \{0, 1, \dots, l\}$$

En el mapa una celda ocupada se representa como  $I(u, v) = 1$  y el espacio libre es representado como  $I(u, v) = 0$ . La posición del robot se representa por una sola celda. La posición del objetivo se representa como  $g(u_0, v_0)$ . La matriz de transformación de distancias es representada por  $d : \Omega \rightarrow \mathbb{R}$ . En la transformación de distancia de Jarvis, este primero inicializa la matriz  $d$ :

$$d(u, v) \begin{cases} \infty & \text{si } g(u_0, v_0)^c \wedge I(u, v) = 0 \\ 0 & \text{si } g(u_0, v_0) \wedge I(u, v) = 0 \\ -1 & \text{si } g(u_0, v_0)^c \wedge I(u, v) = 1 \end{cases} \quad (3.1)$$

, donde  $u$  y  $v$  representan la posición de la celda,  $g(u_0, v_0)$  representa la posición del objetivo y  $g(u_0, v_0)^c$  representa todas las celdas que no están marcadas como objetivo.

Entonces se realizan dos rastreos a través del mapa  $m$ , uno hacia adelante (desde  $u = 0$  hasta  $u = k$  y desde  $v = 0$  hasta  $v = l$ ) y otro hacia atrás (desde  $u = k$  hasta  $u = 0$  y desde  $v = l$  hasta  $v = 0$ ). Cada rastreo emplea operaciones locales entre vecinos de la celda considerada  $C$ . Distintas ventanas con la celda  $C$  y sus respectivos vecinos se muestran en la Tabla 3.2 en la cual para cada una se aplica una ventana de pesos de  $3 \times 3$  centrada en dicha celda, con  $w(0, 0) = C(u, v)$ , donde  $(u, v)$  representa las coordenadas del centro de la ventana.

El nuevo valor asignado a la casilla central se obtiene de la siguiente manera, se calcula un nuevo valor  $d'(u, v)$  para la casilla  $d(u, v)$ , si el valor  $d'(u, v)$  calculado es menor que el valor actual de  $d(u, v)$  dicho valor es actualizado con  $d'(u, v)$ . El cálculo de la casilla  $d'(u, v)$  se realiza de la siguiente manera

$$d'(u, v) := \min_{\substack{i=-1,0,1 \\ j=-1,0,1}} (d(u+i, v+j) + w(i, j)).$$

Una vez obtenido el dato  $d'(u, v)$  se actualiza la celda  $d(u, v)$  como sigue

$$d(u, v) = \min \left\{ d(u, v), d'(u, v) \right\} \quad (3.2)$$

Por ejemplo, considere el rastreo hacia adelante para la ventana *chessboard*  $w_{ChessF}$ , como se muestra en la Tabla 3.2. Dado  $w(0, 0)$  el centro de la ventana de rastreo Euclidiano la cual está centrada sobre el punto  $C(u, v)$  en  $I$  y  $d$ , donde  $I$  se encuentra

desocupada y  $d \neq -1$ . Dado  $d(u, v)$  la distancia actual hacia el objetivo, el algoritmo entonces asigna los valores de la siguiente manera:

$$d(x, y) = \min \begin{cases} d(x-1, y-1) + w(-1, -1), & d(x, y-1) + w(0, -1), \\ d(x+1, y-1) + w(1, -1), & d(x-1, y) + w(-1, 0), \\ d(x, y) \end{cases} \quad (3.3)$$

La propuesta descrita anteriormente se muestra en el Algoritmo 1.

---

**Algoritmo 1** Algoritmo de transformación de distancia de Jarvis.

---

**Input:** Un mapa  $m$

El objetivo  $g(u_0, v_0)$

**Output:** Un arreglo bidimensional con las distancias hacia el objetivo  $d$

Inicializa la matriz  $d$  (Ecuación 3.1)

**repeat**

**for**  $v = 0$  hasta  $v = l$  **do**

**for**  $u = 0$  hasta  $u = k$  **do**

      centra la ventana de rastreo hacia adelante  $C$  en  $(u, v)$

      actualiza  $d(u, v)$  (Ecuación 3.2)

**end for**

**end for**

**for**  $v = l$  hasta  $v = 0$  **do**

**for**  $u = k$  hasta  $u = 0$  **do**

      centra la ventana de rastreo hacia adelante  $C$  en  $(u, v)$

      actualiza  $d(u, v)$  (Ecuación 3.2)

**end for**

**end for**

**until** no se realizan más cambios

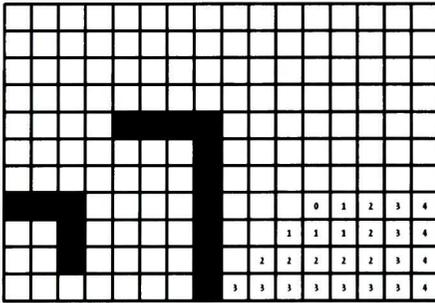
---

En la Figura 3.3 se muestra un ejemplo del resultado obtenido por el algoritmo de Jarvis, en este caso se puede observar que son necesarias dos iteraciones para completar el ambiente ya que con una no sería suficiente al quedar celdas sin valor asignado.

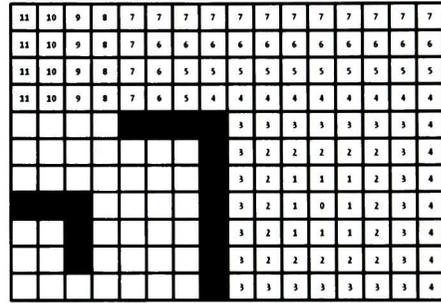
Para un mapa libre de obstáculos, la complejidad en el algoritmo de Jarvis es de  $O(kl)$  para un mapa de dimensiones  $k \times l$ . La complejidad de la transformación de distancia

	rastreo adelante	hacia	rastreo atrás	hacia
city block	- 1 - 1 <b>C</b> - - - - $w_{CityF}$	-	- - - - <b>C</b> 1 - 1 - $w_{CityB}$	
chessboard	1 1 1 1 <b>C</b> - - - - $w_{ChessF}$	-	- - - - <b>C</b> 1 1 1 1 $w_{ChessB}$	
Euclidiano	$\sqrt{2}$ 1 $\sqrt{2}$ 1 <b>C</b> - - - - $w_{EF}$	-	- - - - <b>C</b> 1 $\sqrt{2}$ 1 $\sqrt{2}$ $w_{EB}$	

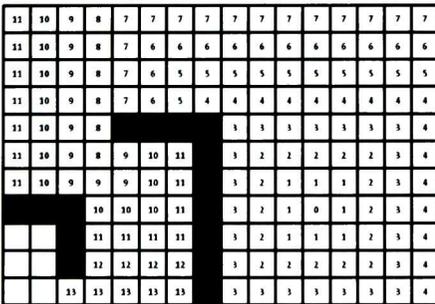
Figura 3.2: Ventanas utilizadas por el algoritmo de transformación de distancias de Jarvis.



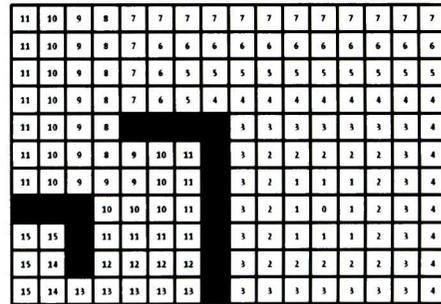
(a) Frente de onda generado en el primer ciclo de la primera iteración



(b) Frente de onda generado en el segundo ciclo de la primera iteración



(c) Frente de onda generado en el primer ciclo de la segunda iteración



(d) Frente de onda generado en el segundo ciclo de la segunda iteración

Figura 3.3: Generación de frente de onda mediante el algoritmo de Jarvis usando la ventana chessboard.

puede incrementarse dependiendo de la topología de los obstáculos (la cual determina el número de iteraciones necesarias que se tiene que repetir el ciclo de rastreo para completar el mapa) [40] y el tiempo de ejecución puede variar dependiendo del tipo de ventana usada. A partir de la transformación de distancia, podemos encontrar la trayectoria óptima para un punto de inicio dado por medio de la búsqueda de sus ocho celdas vecinas y escogiendo la celda que tenga el costo de distancia menor. Este proceso se repite hasta que no existan celdas con un valor menor al de la celda actual, es decir, cuando se ha alcanzado la celda objetivo. En [40] Jarvis compara la transformación de distancia con la propuesta A\* [53] y con los RRT (*Rapidly Exploring Random Trees*) [51] y demuestra que la transformación de distancia es un algoritmo bueno y rápido. Sin embargo, los caminos generados por la transformación de distancia típicamente consisten en un gran número de pasos pequeños entre celdas vecinas libres y con ángulos limitados a múltiplos de  $\pi/4$ .

Lee [52] describe un algoritmo para generar trayectorias suavizadas por medio del agrupamiento de los pequeños pasos entre vecinos en una menor cantidad segmentos de pasos rectos. La propuesta de suavizado inicia en la primera celda de la trayectoria y examina cada una de las celdas subsecuentes seleccionando la última celda que pueda ser alcanzada directamente desde la celda inicial en un movimiento recto dentro del espacio libre con lo cual se genera un segmento de paso recto, posteriormete la celda final del segmento se toma como celda inicial y el proceso de selección de un nuevo segmento se repite hasta que la celda objetivo es alcanzada (Algoritmo 2). Esta propuesta puede obtener trayectos suavizados, sin embargo, no garantiza la obtención del camino óptimo y es necesario aplicarla para cada una de las celdas del mapa para obtener un mapa de planificación universal.

En la Figura 3.4 se muestra el trayecto desde la celda marcada en azul con un valor de 11 hasta el objetivo marcado en verde y con un valor de 0, el trayecto generado mediante la propuesta de Jarvis usando la ventana chessboard se representa con las celdas coloreadas

---

**Algoritmo 2** Suavizado de trayectorias propuesto por Lee.

---

**Input:** Un arreglo bidimensional con las distancias hacia el objetivo  $d$ , La posición inicial  $g(u, v)$

**Output:** Un conjunto de segmentos desde la celda inicial hasta la celda objetivo  
Inicia Index en la primer celda del camino

**repeat**

    Selecciona celda vecina de menor valor

**if** Existe una línea directa entre la celda Index y la celda vecina **then**

        Agrega la celda como punto final del segmento actual

**else**

        Pon Index en el punto final del segmento

        Inicia nuevo segmento

**end if**

**until** la celda objetivo es alcanzada

---

en azul y el trayecto (línea roja) obtenido aplicando el suavizado propuesto de Lee.

### 3.3 Algoritmo EEDT

El algoritmo EEDT (*Exact Euclidean Distance Transform*) es un resultado original de este trabajo de investigación. La principal motivación consistió en obtener, para cada celda del mapa, una estimación exacta de la distancia hasta un objetivo dado. Al mismo tiempo, para cada celda obtenemos el próximo mejor punto intermedio visible hacia la meta (cercano a los vértices de los obstáculos), con la finalidad de obtener caminos suavizados sin necesidad de un post procesamiento. Para ello, empleo un método de rastreo por tramas similar al propuesto por Jarvis, debido a la eficiencia característica de este tipo de algoritmos.

En mi propuesta construyo dos matrices, la primera contiene las distancias de trayecto de cada celda hacia el objetivo (matriz  $d$ ), y la segunda contiene, para cada celda, las coordenadas de la siguiente para alcanzar el objetivo (matriz  $p$ ). El camino más corto puede ser calculado mediante la selección de un punto inicial  $(u, v)$  en el mapa

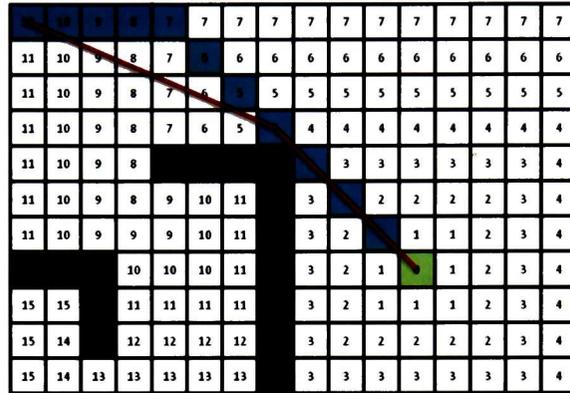


Figura 3.4: Trayecto desde la celda azul marcada con 11 hasta 0 obtenido por la propuesta de Jarvis (celdas en azul). El trayecto obtenido aplicando el suavizado propuesto por Lee se muestra en rojo.

y desde ahí seleccionar el siguiente punto mediante la matriz  $p$  donde, la próxima celda será la indicada por  $p(u, v)$ . Para un mapa libre de obstáculos de tamaño  $k \times l$  celdas, la complejidad de mi algoritmo es  $O(k^2l)$ , esta complejidad puede aumentar dependiendo de la topología del mapa, la cual determina la cantidad de iteraciones necesarias.

Las secuencias de rastreo se detienen cuando no se realizan más cambios en las matrices  $d$  y  $p$ .

El principio de operación del algoritmo propuesto es el siguiente: los valores iniciales de la matriz de transformación de distancias  $d$  son los mismos que en el algoritmo de Jarvis, es decir se inicializa de acuerdo a la Ecuación 3.1:

La matriz de puntos intermedios  $p$  es inicializada bajo el siguiente criterio:

$$p(u, v) = \begin{cases} NULL, & (u, v) \neq g(u_0, v_0) \\ \text{coordenadas de } g(u_0, v_0), & (u, v) = g(u_0, v_0) \end{cases} \quad (3.4)$$

Al igual que en la propuesta de Jarvis, se llevan a cabo dos rastreos de trama por medio de la aplicación de ventanas de rastreo sobre la malla. Las ventanas de rastreo

usadas en mi propuesta se muestran en la Figura 3.5, estas ventanas son inspiradas en las máscaras usadas en 8SED por el algoritmo de Danielsson [22] en procesamiento de imágenes. En el primer paso, la ventana  $w_{EEDTF1}$  es aplicada de izquierda a derecha en cada línea, seguido por la aplicación de la ventana  $w_{EEDTF2}$  de derecha a izquierda sobre la misma línea. Durante el segundo paso, las ventanas de rastreo 3 y 4 se aplican, línea por línea de derecha a izquierda y de izquierda a derecha respectivamente. El pseudocódigo se muestra en el Algoritmo 3.

---

### Algoritmo 3 EEDT

---

**Input:** Un arreglo bidimensional (mapa)

El objetivo

**Output:** Un arreglo bidimensional con las distancias hacia el objetivo  $d$ , Un arreglo bidimensional con las celdas siguientes para alcanzar el objetivo  $p$

Inicializa las matrices  $d$  y  $p$

**repeat**

**for** Cada línea desde arriba **do**

**for** Cada celda en la línea de izquierda a derecha **do**

      aplica rastreo con la ventana  $w_{EEDTF1}$

      Actualiza  $d(r)$  y  $p(r)$  (Algoritmo 5)

**end for**

**for** Cada celda en la línea de derecha a izquierda **do**

      aplica rastreo con la ventana  $w_{EEDTF2}$

      Actualiza  $d(r)$  y  $p(r)$  (Algoritmo 5)

**end for**

**end for**

**for** Cada línea desde abajo **do**

**for** Cada celda en la línea de derecha a izquierda **do**

      aplica rastreo con la ventana  $w_{EEDTB1}$

      Actualiza  $d(r)$  y  $p(r)$  (Algoritmo 5)

**end for**

**for** Cada celda en la línea de izquierda a derecha **do**

      aplica rastreo con la ventana  $w_{EEDTB2}$

      Actualiza  $d(r)$  y  $p(r)$  (Algoritmo 5)

**end for**

**end for**

**until** no se realizan más cambios

---

	rastreo hacia adelante	rastreo hacia atrás
EEDT	$\begin{matrix} \sqrt{2} & 1 & \sqrt{2} \\ 1 & \mathbf{C} & - \\ - & - & - \end{matrix}$	$\begin{matrix} - & - & - \\ - & \mathbf{C} & 1 \\ \sqrt{2} & 1 & \sqrt{2} \end{matrix}$
	ventana $w_{EEDTF1}$	ventana $w_{EEDTB1}$
	$\begin{matrix} - & - & - \\ - & \mathbf{C} & 1 \\ - & - & - \end{matrix}$	$\begin{matrix} - & - & - \\ 1 & \mathbf{C} & - \\ - & - & - \end{matrix}$
	ventana $w_{EEDTF2}$	ventana $w_{EEDTB2}$

Figura 3.5: Ventanas de trama utilizadas por mi propuesta en el cálculo de la transformación de distancia.

Para asignar un punto intermedio válido (visible), primero se escogen 2 vecinos cercanos a la orientación de la celda que se quiere verificar, si la misma celda está asignada a las dos celdas vecinas, entonces ambos vecinos pueden "ver" la celda y por lo tanto la línea de visión entre la celda actual  $r = (u, v)$  y la celda en cuestión está libre de colisiones (Figura 3.6). Esto es  $p(v_a) = p(v_b) = p(v_i)$  con lo cual la celda intermedia puede ser asignada a la celda actual. En otras palabras cuando las celdas apuntadas por los dos vecinos son diferentes ( $p(v_a) \neq p(v_i) \vee p(v_b) \neq p(v_i)$ ), entonces es necesario verificar la línea de visión completa (Figura 3.7). La línea recta es calculada por medio del algoritmo de Bresenham [11]. El pseudocódigo para asignar una celda válida se muestra en el Algoritmo 4.

---

#### Algoritmo 4 Validación de visibilidad

---

**Input:** Un arreglo bidimensional (mapa)

matriz  $p$

**Output:** línea de visión libre de obstáculos

Identifica los dos vecinos  $v_a$  y  $v_b$  más cercanos a la línea de visión

**if** los vecinos apuntan hacia el objetivo ( $p(v_a) = p(v_b) = p(v_i)$ ) **then**

línea de visión libre de obstáculos = **true**

**else**

**if** Cada celda proporcionada por el algoritmo de Bresenham entre los dos puntos se encuentra libre de obstáculos **then**

línea de visión libre de obstáculos = **true**

**else**

línea de visión libre de obstáculos = **false**

**end if**

**end if**

---

Para cada celda  $r = (u, v)$  marcada como libre en la malla  $I$ , analizamos los vecinos  $v_i$  indicados por la ventana de rastreo, verificando la visibilidad de  $r$  hacia el punto intermedio asignado a cada vecino. Si de la celda actual  $r$ , el punto intermedio  $p(v_i)$  de la celda vecina es visible, entonces se calcula la distancia  $d_i(r)$  de  $r$  hacia el objetivo pasando por el punto intermedio  $p(v_i)$ , como se muestra en la Figura 3.8. Si el punto intermedio no es visible, entonces la distancia total  $d_i(r)$  se calcula de  $r$  hacia el objetivo pasando por el vecino

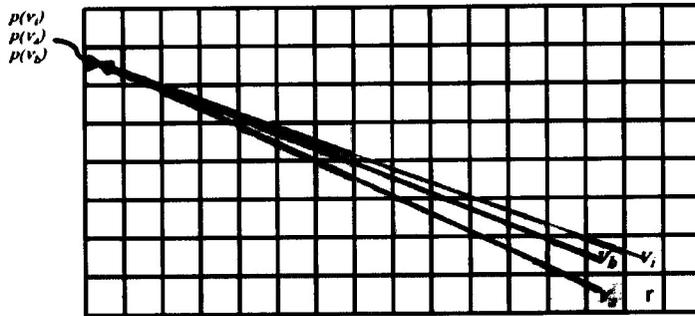


Figura 3.6: La misma celda es asignado a los dos vecinos ( $p(v_a) = p(v_b) = p(v_i)$ ).

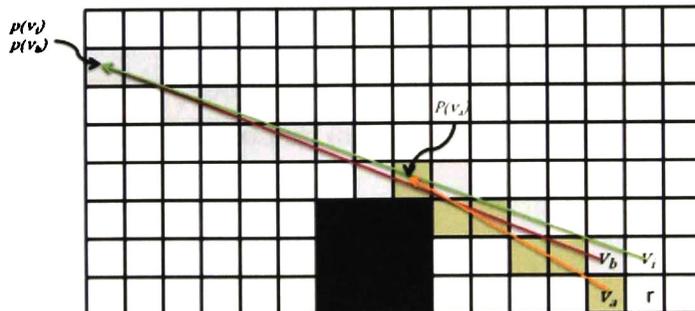


Figura 3.7: Distintas celdas son asignadas a los dos vecinos ( $p(v_a) \neq p(v_i) \vee p(v_b) \neq p(v_i)$ ).

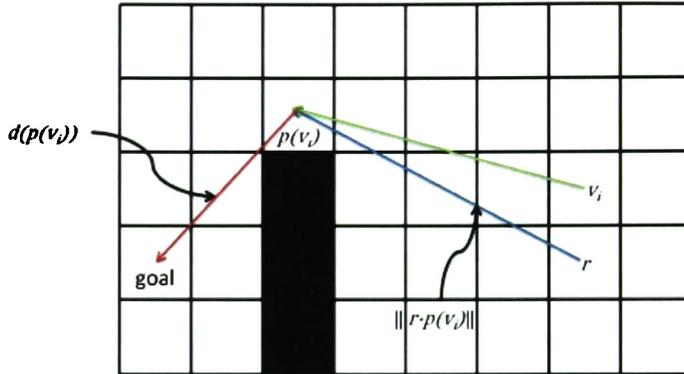


Figura 3.8: Cálculo de la distancia cuando la celda vecina apunta hacia una celda visible

$v_i$ , como se muestra en la Figura 3.9, donde  $w(i)$  es el valor correspondiente al vecino  $i$  de la ventana de rastreo  $w(\cdot)$ .

$$d_i(r) = \begin{cases} \|r - p(v_i)\| + d(p(v_i)), & p(v_i) \text{ es visible} \\ w(i) + d(v_i) & p(v_i) \text{ no es visible} \end{cases} \quad (3.5)$$

Después del análisis de las celdas vecinas (determinado por la ventana de rastreo), si  $\exists d_i(r) | d_i(r) < d(r)$ , el valor de distancia  $d(r)$  para la celda actual  $r$  es actualizada con la menor distancia calculada  $d_i(r)$ .

$$d(r) = \min \{ d(r), d_i(r) \forall i \} \quad (3.6)$$

El punto intermedio  $p(r)$  es actualizado de acuerdo a la Ecuación 3.7 (con  $p(r) = p(v_i)$  si  $v_i$  es visible o  $p(r) = v_i$  si  $v_i$  no es visible). En la Ecuación 3.6 muestro la forma en que se actualiza  $d(r)$

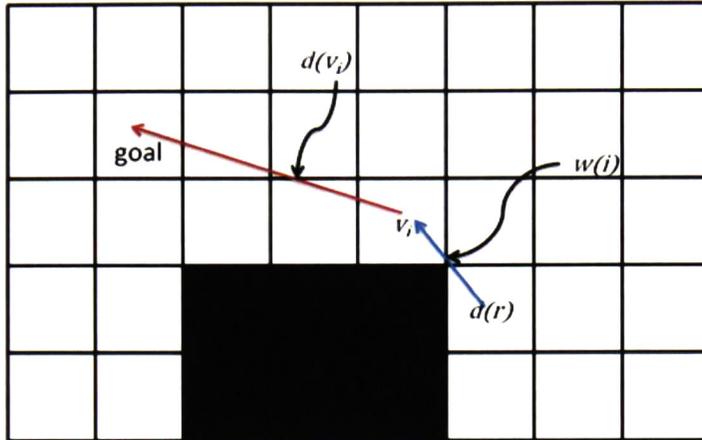


Figura 3.9: Cálculo de la distancia cuando la celda vecina apunta hacia una celda que no es visible para la celda actual

$$p(r) = \begin{cases} p(r), & \nexists d_i(r) | d_i(r) < d(r) \\ p(v_i), & \exists d_i(r) | d_i(r) < d(r) \wedge p(v_i) \text{ es visible} \\ v_i, & \exists d_i(r) | d_i(r) < d(r) \wedge p(v_i) \text{ no es visible} \end{cases} \quad (3.7)$$

El Algoritmo 5 muestra la forma en que se actualizan las matrices  $d(r)$  y  $p(r)$

---

**Algoritmo 5** Actualiza  $d(r)$  y  $p(r)$

---

**Input:** Un arreglo bidimensional (mapa), ventana, matriz  $p$ , matriz  $d$

**Output:**  $d(r)$ ,  $p(r)$

```

for Cada vecino  $v_i$  indicado por la ventana do
  if  $p(v_i)$  es visible desde  $r$  (Algoritmo 4) then
    calcula  $d_i(r)$  de  $r$  hacia el objetivo pasando por  $p(v_i)$ 
  else
    calcula  $d_i(r)$  de  $r$  hacia el objetivo pasando por  $v_i$ 
  end if
end for
    
```

Actualiza  $d(r)$  de acuerdo a la Ecuación 3.6

Actualiza  $p(r)$  de acuerdo a la Ecuación 3.7

---

## 3.4 Resultados

Comparo mi algoritmo EEDT con la transformación de distancia propuesta por Jarvis usando distintas ventanas como *city block*, *chessboard* y Euclidiano. Además, implementé el suavizado de caminos propuesto en [52] para cada uno de los resultados de transformación de distancia usando las distintas ventanas. Para llevar a cabo los experimentos, los algoritmos fueron probados en diferentes escenarios y con distintas celdas objetivo, para así analizar cómo se comportan las propuestas para distintos ambientes. En cada una de las pruebas, analicé el tiempo de ejecución, la exactitud de la distancia resultante y la cantidad de puntos intermedios necesarios para alcanzar el objetivo. Los ambientes usados en las pruebas se muestran en la Figura 3.10. El ambiente 2access fue seleccionado ya que cuenta con la característica de tener áreas abiertas amplias con pasos estrechos entre estas; los ambientes autolab, lab, cave y simple fueron seleccionados ya que vienen incluidos en el simulador player/stage [12] por lo cual están disponibles y son de uso común en robótica. El ambiente autolab es una buena representación de un ambiente típico de oficina, el ambiente lab es un ambiente más complejo con múltiples posibles trayectos para alcanzar una misma habitación, el ambiente cave es una representación donde no se tienen paredes planas y por último el ambiente simple representa un pasillo con accesos a distintas habitaciones.

Las pruebas fueron realizadas en una PC a 2.3 GHz Intel Core 2 Duo Dell con 3 GB de RAM ejecutándose bajo Ubuntu versión 9.10. Todos los algoritmos fueron implementados en C++ y compilados con g++ 4.4.1.

### 3.4.1 Trayectorias Obtenidas

La Figura 3.11 muestra un comparativo de las trayectorias obtenidas con las distintas propuestas de transformación de distancia. Las Figuras 3.11(a), 3.11(b) y 3.11(c)

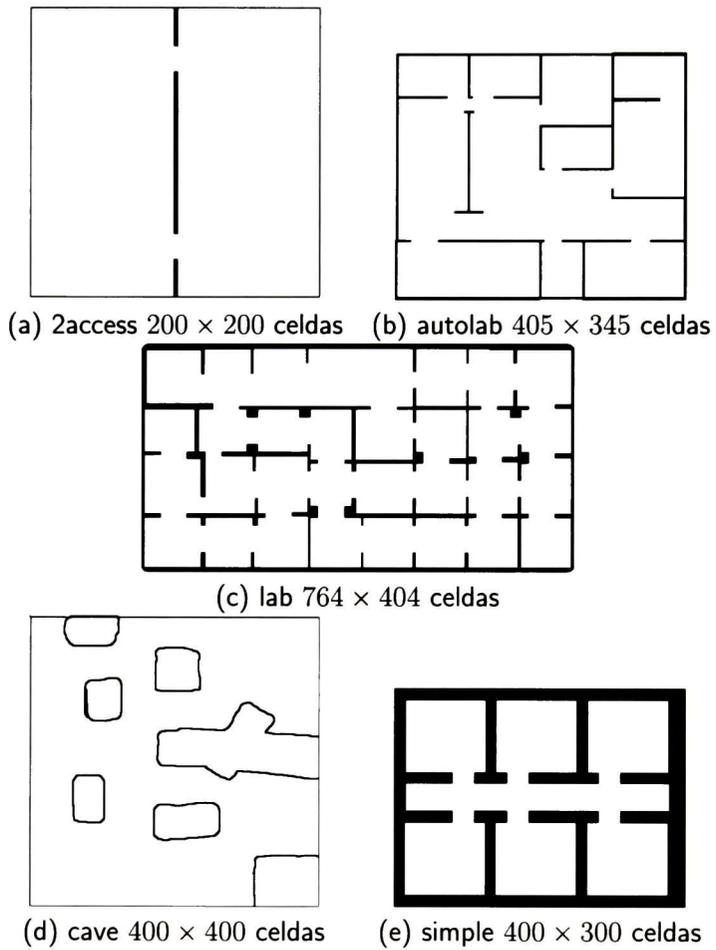


Figura 3.10: Mapas usados para las pruebas de planificación global de trayectorias

muestran los trayectos obtenidos con el algoritmo de Jarvis usando las ventanas *City block*, *Chessboard* y *Euclidiano* respectivamente, y en las cuales se aprecia que en los trayectos generados, los ángulos de desplazamiento están limitados a múltiplos de  $\pi/4$ .

Las Figuras 3.11(d) 3.11(e) y 3.11(f) muestran los resultados obtenidos al aplicar el algoritmo de suavizado ([52]) a los resultados de *City block*, *Chessboard* y *Euclidean* respectivamente. En este caso no existe limitante en el ángulo de desplazamiento durante la trayectoria. Sin embargo, no se garantiza que el trayecto generado sea el óptimo.

Finalmente mi propuesta genera trayectos suavizados con una distancia total siempre menor o igual a los resultados obtenidos por las otras propuestas, sin necesidad de realizar un post procesamiento. En la Figura 3.11(g) podemos ver algunos trayectos generados por mi propuesta.

Observando la Figura 3.11(g), podemos ver que las trayectorias generadas por mi algoritmo siempre selecciona trayectos que pasan por los vértices de los obstáculos.

En términos concretos, mi algoritmo construye un mapa de visibilidad del ambiente, como se muestra en la Figura 3.12, donde cada región tiene un objetivo visible para dicha región. En la Figura 3.12 las flechas indican la dirección de la próxima celda hacia el objetivo.

Los resultados de la Figura 3.11 muestran claramente que mi propuesta, mediante la selección consecutiva de celdas visibles dadas por la matriz  $p$ , es eficiente en la construcción de trayectorias hacia el objetivo.

### 3.4.2 Evaluación de la Exactitud

Con la finalidad de comparar las distancias de trayecto obtenidas por mi algoritmo contra varias propuestas de transformación de distancia, calculamos el error relativo entre ellos. Para esto usé un mapa libre de obstáculos donde el objetivo es la celda central de dicho mapa y calculé el error cuadrático medio (RMSE). Además, evalué la exactitud

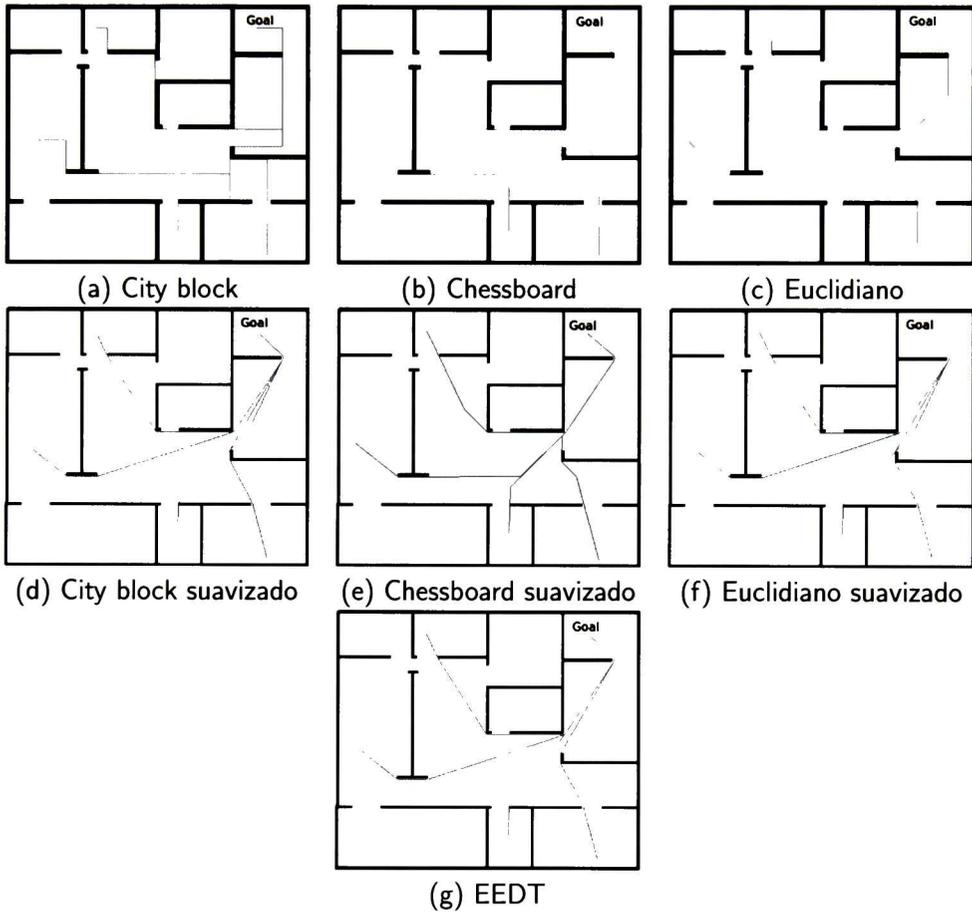
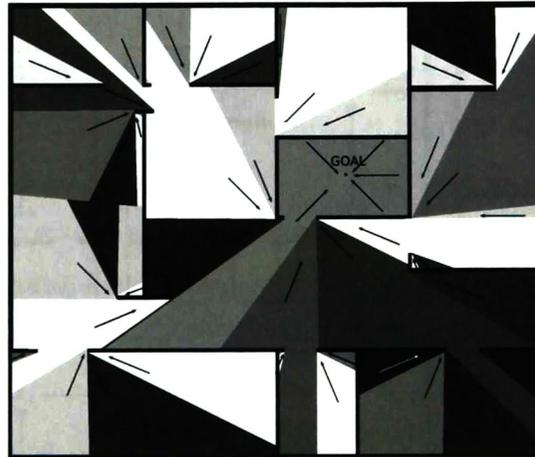
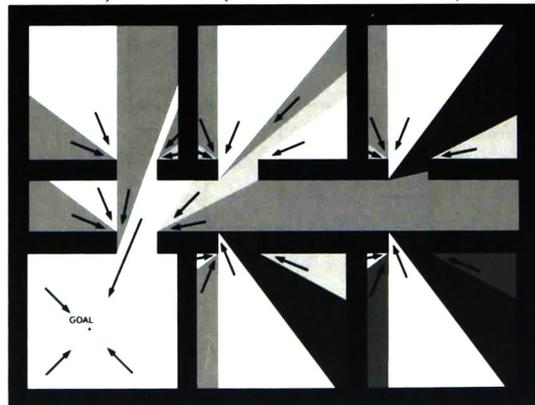


Figura 3.11: Trayectorias generadas para el mapa autolab (405 × 345 celdas).



a) autolab (4045 × 3445 celdas)



b) simple map (4000 × 3000 celdas)

Figura 3.12: Mapas de visibilidad generados. Para cada región (color) las flechas apuntan hacia la próxima mejor celda visible para llegar al objetivo.

Algoritmo	Tamaño del mapa			
	256 × 256	512 × 512	1000 × 1000	5000 × 5000
city block	35.3	70.60	137.89	689.49
city block suavizado	0	0	0	0
chessboard	17.58	35.16	68.69	343.45
chessboard suavizado	0	0	0	0
Euclidiano	5.89	11.79	23.03	115.17
Euclidiano suavizado	0	0	0	0
EEDT	0	0	0	0

Tabla 3.1: Error cuadrático medio (celdas). Comparación con el algoritmo de Jarvis usando distintas ventanas con el frente de onda generado desde el centro de un mapa libre de obstáculos.

en los ambientes mostrados en la Figura 3.10, con la celda objetivo ubicada en distintas posiciones y calculé el RMSE y la peor medida.

En la Tabla 3.1 mostramos el RMSE para un ambiente libre de obstáculos con distintas resoluciones de mapa. El resultado de una transformación de distancia debe de ser circular. La Figura 3.13 muestra que mi propuesta y la propuesta de Jarvis aplicando el suavizado propuesto por Lee [52] generan ondas circulares y donde la escala de gris de cada celda representa la distancia hacia el objetivo. Esto también puede apreciarse en la Tabla 3.1.

Sin embargo, en mapas con obstáculos (Figuras 3.14 y 3.15), mi algoritmo genera frentes de onda suavizados y redondeados, mientras que la propuesta de Jarvis obtiene resultados sin bordes redondeados. Aplicando a la propuesta de Jarvis el suavizado de Lee proporciona un frente de onda con bordes redondeados pero con algunos errores en el cálculo de las distancias, como se puede apreciar en los patrones rotos del frente de onda (Figuras 3.14(d), 3.14(e), 3.14(f), y en las Figuras 3.15(d), 3.15(e) y 3.15(f)).

Los resultados se resumen en las Tablas 3.2 y 3.3, donde se presenta el error relativo del algoritmo de Jarvis y del algoritmo de Jarvis con suavizado comparados con mi propuesta. Cabe resaltar que mi propuesta obtiene siempre una distancia más corta hacia el objetivo. En la Tabla 3.2 podemos ver el RMSE para cada celda en cada uno de los mapas y en la

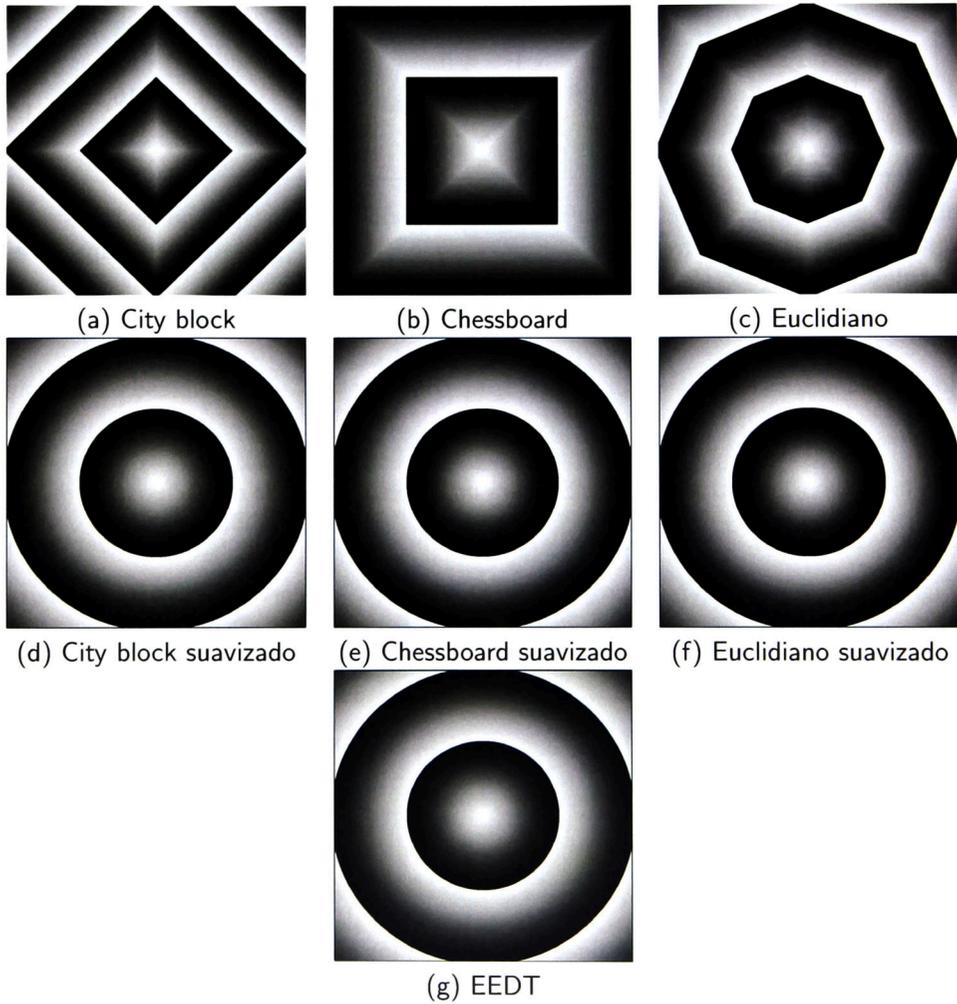


Figura 3.13: Imágenes resultantes para un mapa libre ( $1000 \times 1000$  celdas).

Algoritmo	Mapa					
	2access 100×150	2access 190×150	autolab 30×70	autolab 130×150	lab 157×337	lab 343×267
city block	36.134	38.829	110.936	72.500	90.103	72.100
city block suavizado	0.120	1.564	0.958	0.709	5.376	3.676
chessboard	17.607	14.660	49.412	36.267	31.704	28.874
chessboard suavizado	2.614	0.952	4.430	3.602	5.318	2.687
Euclidiano	5.233	8.480	18.293	9.914	19.494	14.035
Euclidiano suavizado	0.120	1.205	1.156	0.405	5.118	2.442
EEDT	0.000	0.000	0.000	0.000	0.000	0.000

Algoritmo	Mapa			
	cave 70×70	cave 320×130	simple 150×200	simple 240×340
city block	73.816	95.492	35.388	60.516
city block suavizado	5.029	8.733	1.922	2.588
chessboard	29.093	36.288	13.234	24.847
chessboard suavizado	6.686	5.204	0.458	0.000
Euclidiano	15.712	20.399	7.920	11.318
Euclidiano suavizado	3.888	6.700	1.922	2.588
EEDT	0	0	0	0

Tabla 3.2: Error cuadrático medio (celdas) relativo a mi propuesta. Comparación para distintos algoritmos de transformación de distancias para los mapas mostrados en la Figura 3.10.

Tabla 3.3 mostramos el mayor error medido. Los peores resultados fueron generados con la propuesta de Jarvis, mientras que con la propuesta de Jarvis con post procesamiento (propuesta de Lee) en algunos casos el error es cercano a cero. Sin embargo, el aplicar dicho suavizado a la propuesta de Jarvis incrementa el tiempo de cómputo comparado con mi propuesta, como podemos ver en la Tabla 3.4.

### 3.4.3 Tiempos de Ejecución

Para determinar los tiempos de ejecución, he realizado pruebas sobre los mapas mostrados en la Figura 3.10, con los frentes de onda generados a partir de distintos puntos.

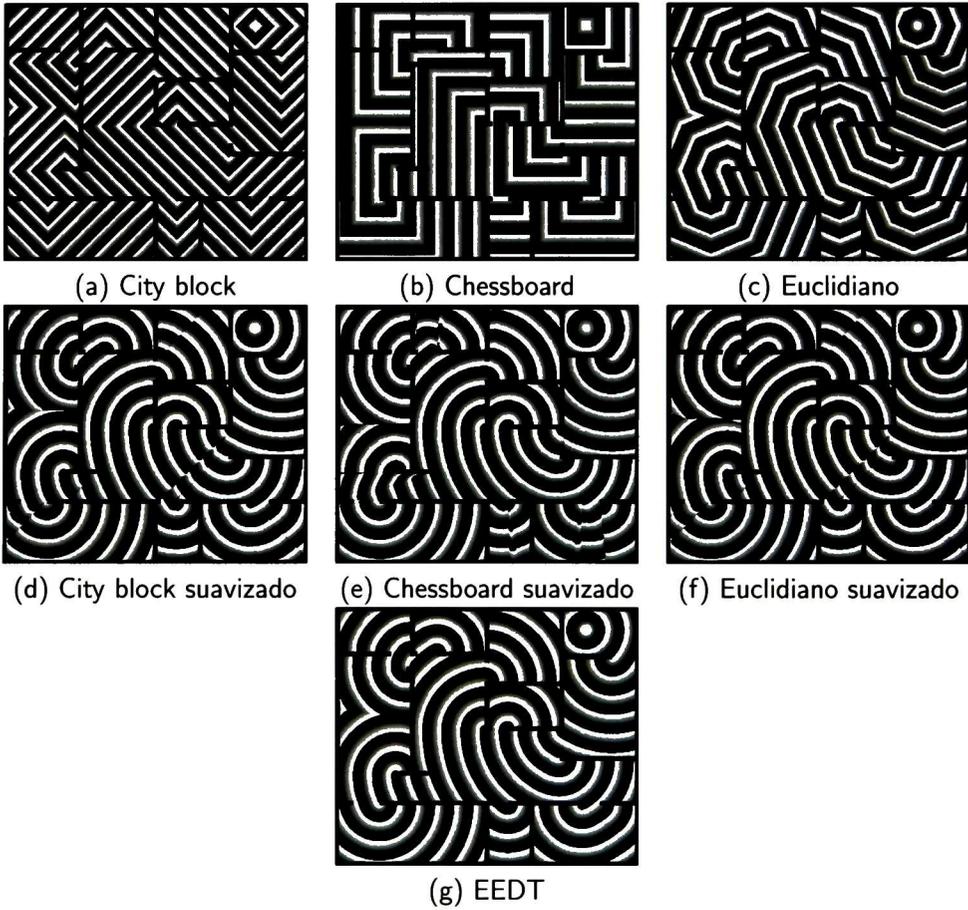


Figura 3.14: Frentes de onda resultantes para el mapa autolab de  $405 \times 345$  celdas.

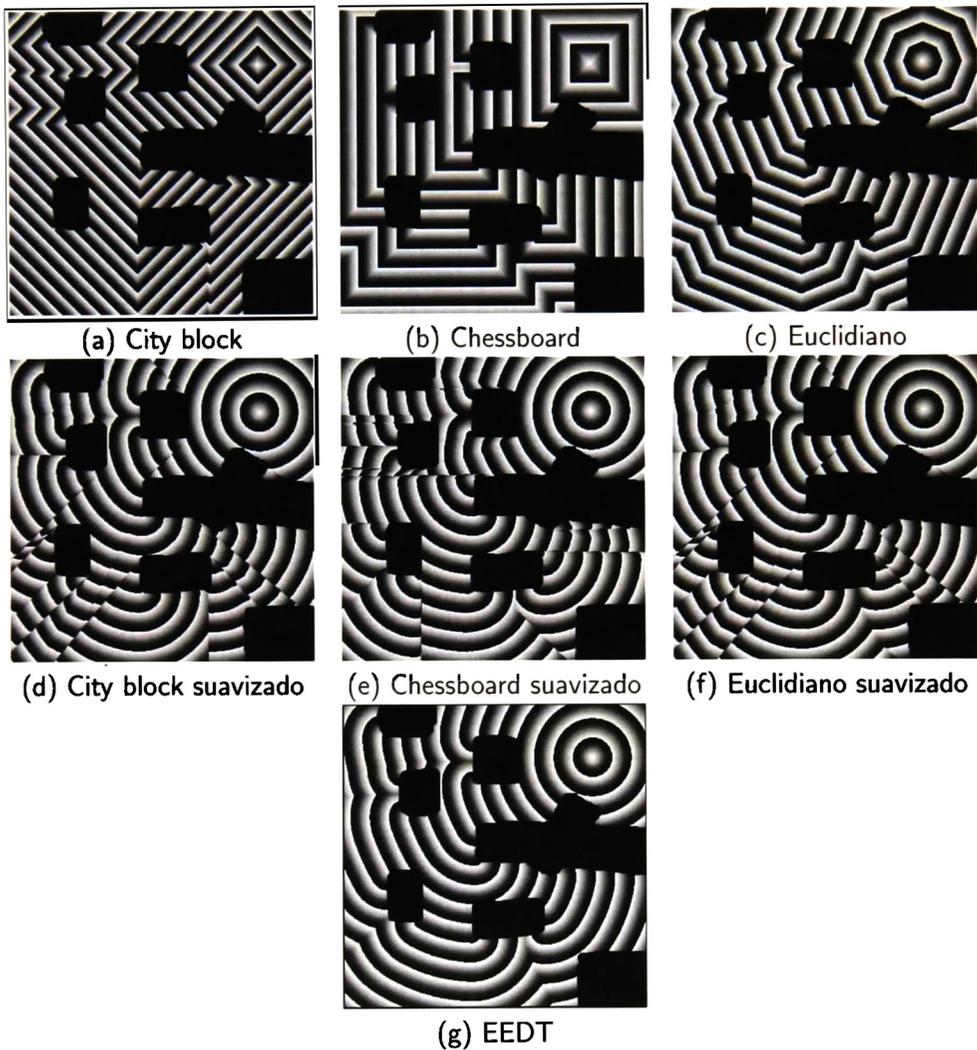


Figura 3.15: Frentes de onda resultantes para el mapa cave de  $400 \times 400$  celdas.

Algoritmo	Mapa					
	2access 100×150	2access 190×150	autolab 30×70	autolab 130×150	lab 157×337	lab 343×267
city block	72.816	92.711	199.183	147.033	180.936	157.261
city block suavizado	2.088	16.916	11.564	20.915	42.484	32.233
chessboard	42.191	49.733	99.143	78.076	86.089	82.521
chessboard suavizado	38.652	8.440	31.814	24.685	27.548	29.435
Euclidiano	11.273	18.037	42.088	26.842	34.255	29.713
Euclidiano suavizado	2.088	11.954	17.547	8.139	18.788	15.489
EEDT	0	0	0	0	0	0

Algoritmo	Mapa			
	cave 70×70	cave 320×130	simple 150×200	simple 240×340
city block	151.932	200.427	77.298	126.618
city block suavizado	51.094	81.920	8.279	13.431
chessboard	75.919	112.933	41.109	66.060
chessboard suavizado	79.653	57.742	2.324	0.001
Euclidiano	38.316	46.832	16.126	26.753
Euclidiano suavizado	21.708	31.188	8.279	13.431
EEDT	0	0	0	0

Tabla 3.3: Error mayor (celdas) utilizando las distintas técnicas de transformación de distancia.

Algoritmo	Mapa					
	2access 100×150	2access 190×150	autolab 30×70	autolab 130×150	lab 157×337	lab 343×267
city block	0.13	0.01	0.06	0.048	0.138	0.158
city block suavizado	1.64	2.3	25.25	11.768	57.56	53.69
chessboard	0.018	0.018	0.081	0.061	0.205	0.203
chessboard suavizado	1.579	2.828	25.195	12.211	57.731	51.203
Euclidiano	0.018	0.016	0.084	0.062	0.25	0.213
Euclidiano suavizado	1.632	3.004	25.28	11.75	57.1	54.39
EEDT	0.035	0.042	0.23	0.205	0.5	0.49

Algoritmo	Mapa			
	cave 70×70	cave 320×130	simple 150×200	simple 240×340
city block	0.048	0.057	0.028	0.039
city block suavizado	17.468	17.177	5.378	10.629
chessboard	0.075	0.102	0.044	0.045
chessboard suavizado	20.315	15.4602	5.59	10.835
Euclidiano	0.075	0.15	0.045	0.045
Euclidiano suavizado	17.44	19.02	5.41	10.665
EEDT	0.215	0.228	0.134	0.043

Tabla 3.4: Comparación de tiempo necesario para la generación del frente de onda desde distintos puntos de cada mapa (segundos).

Para cada punto de origen del frente de onda, el tiempo de ejecución fue promediado de 31 ejecuciones distintas. Los tiempos promedios se muestran en la Tabla 3.4, donde podemos observar que mi propuesta requiere más tiempo que la propuesta de Jarvis sin post procesamiento. Sin embargo, para obtener resultados comparables, es necesario aplicar el algoritmo de suavizado a la propuesta de Jarvis. Al hacerlo, se incrementa considerablemente el tiempo de ejecución en comparación con mi propuesta.

## 3.5 Conclusiones

En este capítulo he presentado un algoritmo para la obtención de transformación de distancia para la planificación de trayectorias, donde además de obtener la distancia exacta para cada una de las celdas del mapa, obtenemos la mejor trayectoria al objetivo como una secuencia de sub-objetivos, sin necesidad de un post-procesamiento, como en la mayoría de las propuestas de la literatura. De los resultados obtenidos puedo demostrar que mi propuesta es muy eficiente en la selección del próximo punto ya que siempre selecciona el próximo mejor punto visible. Además, mi propuesta no tiene restricciones en el ángulo en la construcción del camino, restricción que tiene el algoritmo propuesto por Jarvis. Por último podemos destacar que mi propuesta no necesita realizar una búsqueda exhaustiva de los puntos visibles, es decir que es ampliamente explotada la información de visibilidad de las celdas vecinas con la finalidad de agregar eficiencia a mi algoritmo. Las mejoras obtenidas en el algoritmo de planificación benefician en gran medida el área de la robótica, ya que además de obtener distancias exactas, estas medidas permiten construir esquemas de coordinación sobre la base de una medida de utilidad más exacta.

# 4

## Exploración Multi-Robot Usando Auto-Ofertas bajo Restricción en el Rango de la Comunicación

### 4.1 Introducción

La exploración de medios ambientes y la respectiva generación de mapas son temas de gran relevancia en la robótica móvil. Para ser capaces de navegar de manera efectiva, los robots móviles autónomos deben contar con la capacidad de construir representaciones de los ambientes. La construcción del mapa implica el uso de una buena estrategia de exploración y, en el caso de sistemas multi-robot, es necesario considerar la restricción impuesta por el rango limitado de comunicación.

Existen muchas aplicaciones en las que la exploración es una parte integral de la misión del robot, como son búsqueda y rescate, manejo de materiales peligrosos, exploración

planetaria, entre otros. Al utilizar un equipo de robots para la exploración, se obtienen múltiples ventajas, comparado con el uso de un único robot, como son la tolerancia a fallos individuales, la concurrencia y su correspondiente disminución del tiempo necesario para la exploración y la fusión de los mapas de modo que los errores pueden ser disminuidos.

En esta tesis, propongo el uso de un mecanismo de coordinación de los robots basado en ofertas, que permite a cada robot decidir individualmente los mejores objetivos de exploración, en un proceso que incluye a cada uno de los miembros del equipo sin necesidad de tiempos muertos, logrando una estrategia de exploración totalmente descentralizada.

Los objetivos de exploración de mi propuesta retoman el concepto de fronteras propuesto por Yamauchi [95], como la división entre la zona explorada y la zona desconocida. Estas regiones son agrupadas, en función del rango de alcance de los sensores embarcados, para formar los posibles objetivos de los robots.

En la función de utilidad de mi propuesta, he logrado representar tres métricas principales que permiten modelar comportamientos basados en las heurísticas de optimización multiobjetivo, los cuales son la explotación, la exploración y la cohesión:

- **Explotación:** busca que cada robot se dirija hacia las fronteras más cercanas tratando de minimizar las distancias recorridas
- **Exploración:** busca la separación de los robots con la finalidad de minimizar el trabajo redundante y la interferencia entre ellos
- **Cohesión:** busca mantener a los robots en comunicación con otros miembros del equipo con la finalidad de mantener los mapas de cada equipo actualizados y en caso de la falla de algún robot evitar que la información recabada por éste se pierda.

El funcionamiento de mi estrategia de coordinación es la siguiente: cuando un robot necesita un nuevo objetivo de exploración, solicita individualmente a los demás miembros del equipo información sobre sus mapas y su posición actual; con la nueva información, el

robot realiza un proceso que denominamos *auto-oferta*, en el cual realiza un estimado de cuál sería la mejor distribución de objetivos entre los miembros del equipo y anuncia a los demás robots únicamente cuál es el objetivo que toma para sí; a partir de este momento, el robot se dirige a su objetivo sin interrupción, solamente respondiendo a aquel robot que le solicite información del mapa. La única interrupción posible se presenta cuando algún otro miembro del equipo, durante su propia tarea de exploración, explora el objetivo seleccionado, por lo que envía al primero una señal que denominamos *de parada*, para detener la misión y obligarlo a auto-ofertar un nuevo objetivo.

La función de utilidad utilizada para calcular las auto-ofertas busca asignar al robot una frontera inexplorada cercana a su posición actual (con lo que se modela la propiedad de explotación de la posición), mientras intenta separarlo de los potenciales objetivos de los otros robots (evitando la redundancia y modelando la exploración del ambiente) pero sin romper los puentes de comunicación (manteniendo la cohesión del equipo).

Individualmente, los mapas son modelados utilizando una malla de ocupación, donde cada celda puede estar en uno de tres estados: ocupado, libre o desconocido. El estado de cada celda es actualizado utilizando los datos proporcionados por los sensores de distancia, basados en láser, embarcados en cada robot. Los resultados numéricos demuestran que mi propuesta es eficiente y tolerante a fallos.

## 4.2 Definición del Problema

Recordemos la definición de exploración multi-robot dada anteriormente: Dado un conjunto de  $r$  robots cuyas trayectorias están definidas por las secuencias de posiciones como

$$X_T^r \{x_0^r, x_1^r, x_2^r, \dots, x_T^r\}$$

donde  $x_t^r$  es un vector tridimensional, que consiste en sus coordenadas en el plano bidimensional más un valor rotacional para la orientación del robot  $r$  en el instante  $t$ , y donde la ubicación inicial de cada robot  $x_0^r$  es conocida.

Sea  $u_t^r$  la odometría que caracteriza el movimiento entre el instante  $t$  y el instante  $t - 1$ . Entonces la secuencia

$$U_T^r = \{u_1^r, u_2^r, u_3^r, \dots, u_T^r\}$$

caracteriza el movimiento relativo del robot, obtenido por integración del movimiento de los motores de las ruedas.

La secuencia de mediciones tomadas del medio ambiente con los sensores láser embarcados en los robots está dada como

$$Z_T^r = \{z_1^r, z_2^r, z_3^r, \dots, z_T^r\}$$

Finalmente, cada robot puede comunicarse directamente con cualquier otro que se encuentre dentro de la región circular  $\beta$  con centro en el robot  $r$ , a fin de intercambiar información necesaria para la coordinación del equipo.

La Figura 4.2 ilustra la interacción de causalidad entre las variables involucradas en el problema de exploración multi-robot (en este caso se muestra para 2 robots).

El problema de exploración multi-robot consiste entonces en la recuperación de un modelo del mundo  $m$ , a partir de la secuencia de ubicaciones de cada robot  $X_T^r$ , de los datos de odometría  $U_T^r$  y de los datos obtenidos por los sensores embarcados  $Z_T^r$ , minimizando el tiempo de exploración, la distancia recorrida y el tiempo en que algún robot queda incomunicado.

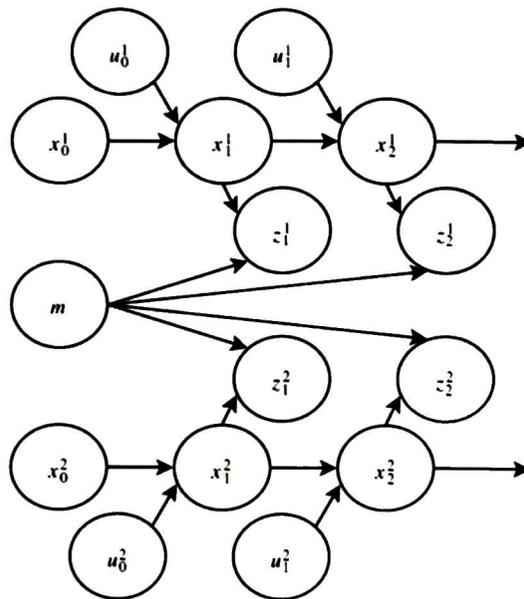


Figura 4.1: Modelo gráfico del problema de exploración multi-robot. Los arcos indican las relaciones causales.

## 4.3 Propuesta

Mi propuesta usa un mecanismo de ofertas basado en la asignación de objetivos y la restricción en el rango de la comunicación para coordinar las acciones de los robots. Mi algoritmo es un proceso totalmente descentralizado y asíncrono, ya que cada robot estima la oferta de los otros miembros del equipo y su propia oferta, información que le sirve para tomar una decisión sin interferir en las decisiones de los otros miembros del equipo y sin requerir sincronizarse en el tiempo con ellos. Además, si el objetivo seleccionado es descubierto por otro robot, este envía una señal de parada con la finalidad de indicar al primer robot que su objetivo ha sido explorado y obligarlo a reiniciar un nuevo proceso de ofertas para seleccionar un nuevo objetivo.

Así, cada robot se encuentra en uno de tres estados posibles:

1. Construyendo objetivos: El robot se encuentra en el proceso de análisis de información y de auto-oferta para seleccionar el objetivo que más beneficie a la exploración. Una vez que se cuenta con un objetivo, se procede a navegar hacia él.
2. Navegando hacia el objetivo: El robot se desplaza hacia el objetivo definido. Una vez que ese objetivo ha sido explorado (por este robot o por algún otro miembro del equipo, notificado por una *señal de parada*) se inicia un nuevo proceso de auto-oferta.
3. Avería total: El robot deja de funcionar totalmente y se excluye del proceso de exploración.

En paralelo con lo anterior, un módulo de comunicación y recepción de mensajes maneja la información desde y hacia los demás miembros del equipo.

La Figura 4.3 muestra el funcionamiento, en forma de diagrama de estados, del comportamiento de cada robot.

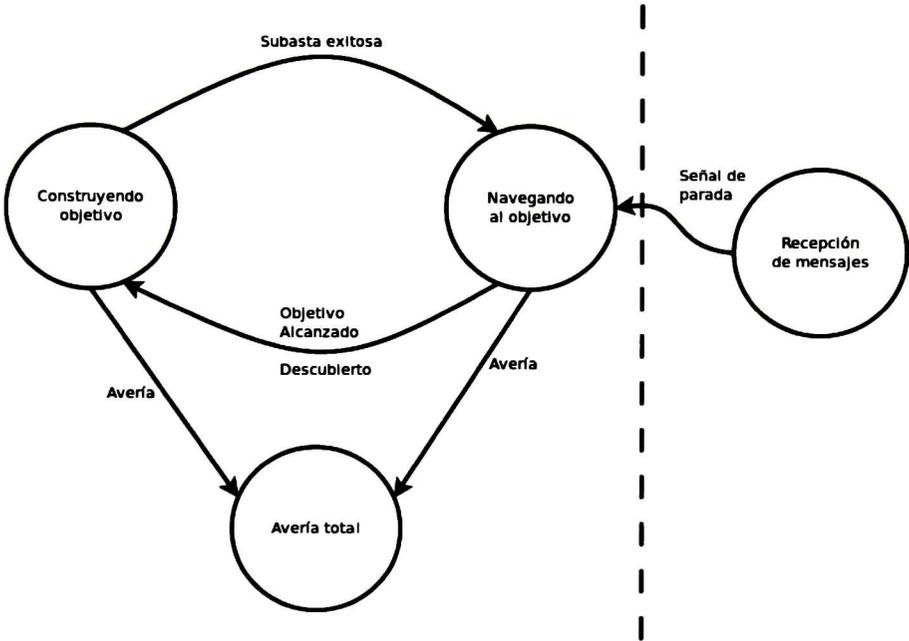
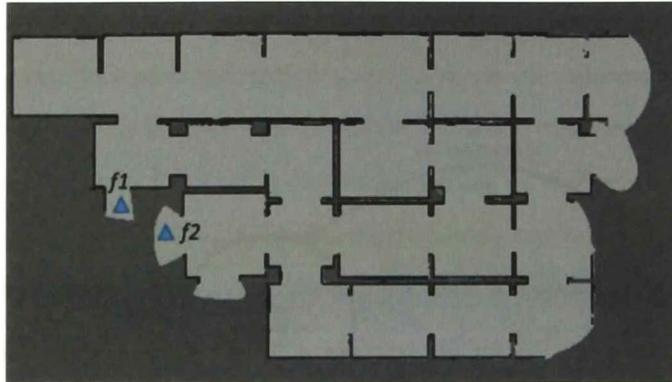
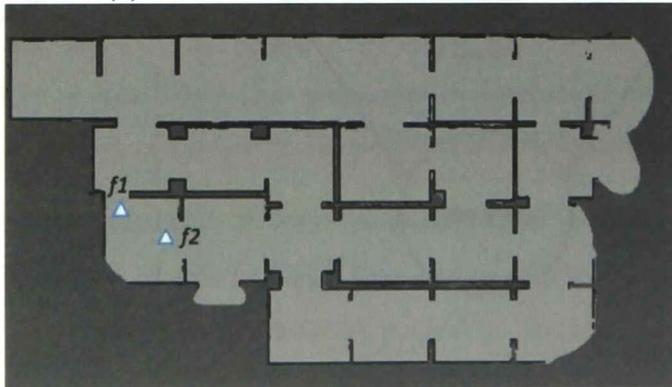


Figura 4.2: Diagrama de interacción de los estados de cada robot.



(a) Dos fronteras están cercanas entre sí



(d) Un robot explora la frontera  $f_2$  y consecuentemente la  $f_1$

Figura 4.3: Dos fronteras están cercanas entre si y pueden ser exploradas por un solo robot.

Para ilustrar el funcionamiento del uso de la *señal de parada*, podemos observar en la Figura 4.3 dos fronteras ( $f1$  y  $f2$ ) muy cercanas entre sí, pero que en el proceso de auto-ofertas serían asignadas a dos robots distintos. Si un robot alcanza la frontera  $f2$  puede también explorar la frontera  $f1$  por lo cual ya no es necesario explorar nuevamente la frontera  $f1$ . Si un robot ya se dirigía hacia  $f1$ , se le notifica que debe elegir un nuevo objetivo.

En mi enfoque, el sensado y generación del mapa se llevan a cabo continuamente durante el proceso de navegación y no únicamente al alcanzar el punto objetivo (como es el caso de otras propuestas presentadas en el Estado del Arte). Por otra parte, la frontera objetivo se considera como alcanzada cuando el robot se encuentra a una distancia corta de éste (la mitad del rango de sensado) y la frontera está dentro de la línea de visión del robot, con lo cual, el robot puede iniciar un nuevo proceso de oferta.

En la Figura 4.3 podemos ver un ejemplo de implementación de la estrategia propuesta usando tres robots, cada uno de los cuales ejecuta tres procesos en paralelo:

1. Principal: Es el encargado de administrar cada uno de los comportamientos del robot (navegar, generar ofertas y planificar trayectorias).
2. Comunicación: Es el encargado de enviar y recibir información de los otros miembros del equipo.
3. Generador de mapa: Es el encargado de integrar, en una sola representación, los datos recogidos por los sensores embarcados y la información recibida desde los otros miembros del equipo.

Mi propuesta es eficiente y robusta, ya que evita explorar espacios ya explorados y minimiza los viajes innecesarios. Asimismo, al ser una estrategia descentralizada se eliminan los puntos únicos de falla: por la manera en que se realiza la coordinación,

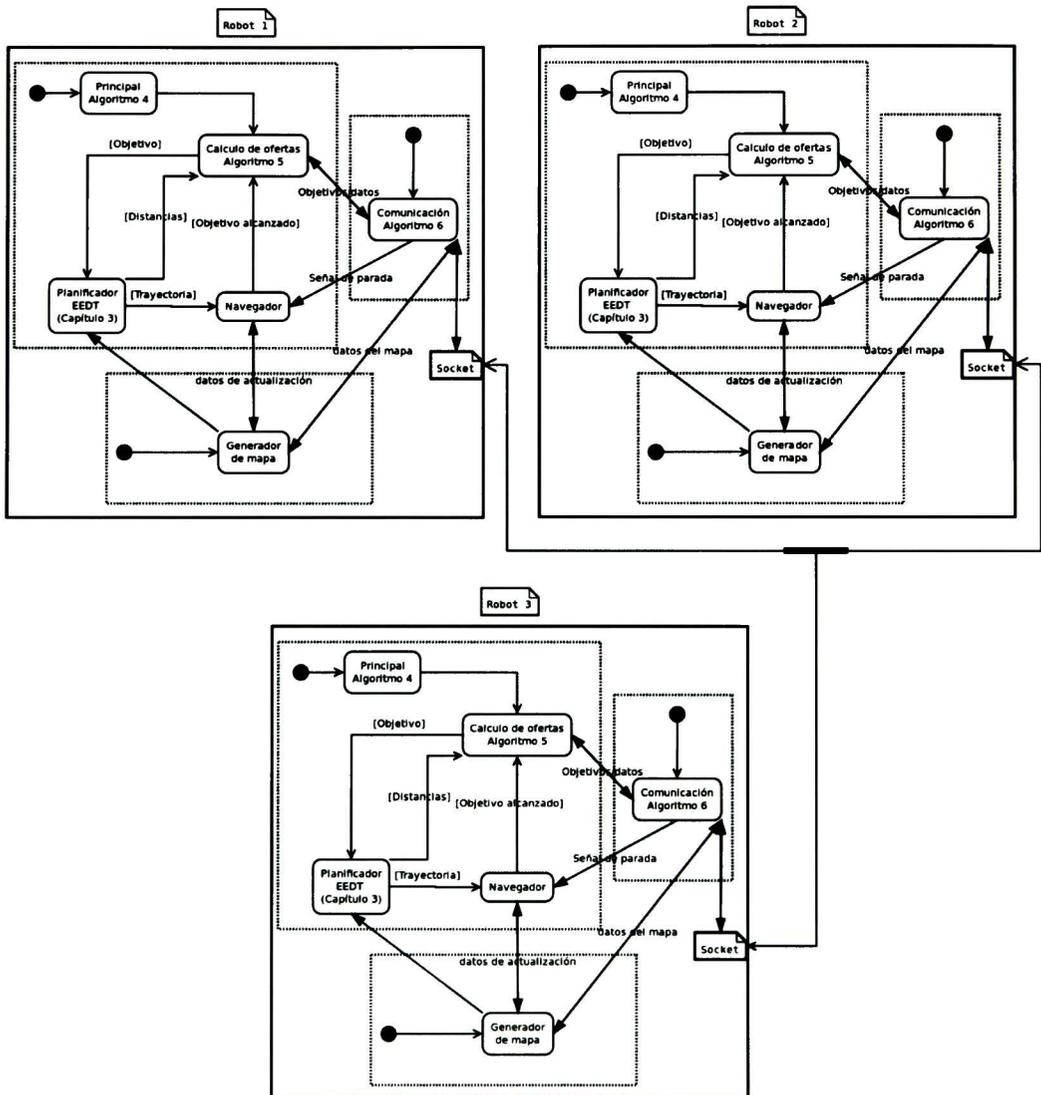


Figura 4.4: Ejemplo de mi propuesta usando 3 robots.

los robots pueden entrar y salir del proceso de ofertas sin comprometer el proceso de exploración final.

### 4.3.1 Localización y Generación de Mapas

Todos los robots en el equipo inician con un marco de referencia común, pero dado que se desea desarrollar un esquema descentralizado de información y de toma de decisiones, es necesario que cada robot mantenga y procese la información de su postura (localización y orientación) y de sus sensores láser. Las medidas láser y de posición deben estar sincronizadas para actualizar de manera eficiente el mapa.

En esta tesis usamos mapas basados en mallas de ocupación [50]. En las mallas de ocupación cada celda representa un espacio en el ambiente (por ejemplo  $2cm \times 2cm$ ). Para fines de navegación y planificación es necesario conocer si el estado de cada celda es libre, ocupado o desconocido.

El mapa se actualiza con los datos proporcionados por los sensores láser. Dada una medida láser y una posición, el mapa puede actualizarse utilizando la postura del robot y modificar el valor de la celda en la línea de visión del láser.

Para resolver la localización y mantenimiento de mapas decidimos usar el algoritmo MRICP (*Map Referenced Iterative Closest Point*) [27], ya que soluciona el problema de SLAM *online* en mapas de rejillas de ocupación, con lo cual se pueden obtener buenos mapas individuales [23, 45]. Es decir cada robot corrige individualmente su posición a la vez que integra información en su mapa, con lo cual esta información puede ser integrada por los otros miembros del equipo.

### 4.3.2 Cálculo de las Ofertas

La oferta para cada robot es una función ponderada de tres diferentes métricas: un parámetro de explotación, un parámetro de exploración y un parámetro de cohesión. Los

parámetros son definidos de la manera siguiente:

- **Explotación:** Incluye la distancia  $\rho_k$  desde la posición actual del robot  $r_k$  hasta su objetivo asignado, y la distancia  $\alpha_k^f$  desde el objetivo hasta la frontera considerada  $f$ . El parámetro de explotación es la suma de las dos distancias  $(\rho_k + \alpha_k^f)$ , y representa la distancia que el robot tendrá que recorrer para cumplir su objetivo actual y alcanzar después la frontera  $f$ . Con este parámetro se logra dar prioridad a las fronteras cercanas al robot, pues las fronteras lejanas tendrán un costo mayor.
- **Exploración:** Este parámetro representa la distancia promedio de la frontera considerada  $f$  a la posición actual de los demás miembros del equipo, incluyendo sus objetivos actuales, de la siguiente manera

$$\frac{\sum_{i=1, i \neq k}^n (\rho_i + \alpha_i^f)}{n - 1}$$

Este valor modela el costo promedio que le tomaría a otro miembro del equipo alcanzar la frontera considerada. Para lograr la dispersión de los robots, debemos dar prioridad a las fronteras que serían más costosas, en promedio, a cualquier otro miembro del equipo.

- **Cohesión:** La oferta que un robot  $k$  hace por una frontera  $f$  resulta penalizada si, para alcanzarla, el robot se ve forzado a romper el puente de comunicación con el resto del equipo. Como los robots sólo se comunican entre sí cuando alcanzan sus objetivos  $g(\cdot)$  para solicitar la información necesaria para sus respectivos procesos de subasta, consideramos que el puente de comunicación no se rompe si la trayectoria  $\gamma(r_k, f)$  que debe seguir el robot  $r_k$  para alcanzar la nueva frontera  $f$  queda en todo momento dentro del rango de alcance de comunicación  $\varphi$  de al menos uno de los objetivos de los demás miembros del equipo  $(\exists g(\cdot) \mid \text{distance}(g(\cdot), \gamma(r_k, f)) < 0.9\varphi)$ . Evidentemente, si sólo hay un robot activo ( $n = 1$ ), la penalización no aplica.

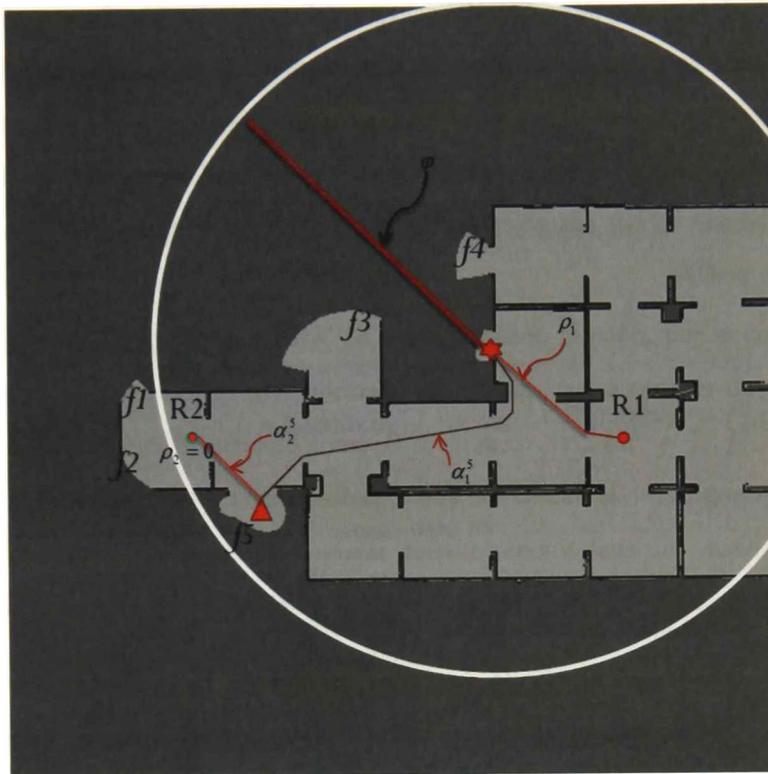


Figura 4.5: Ejemplo de los parámetros usados en mi propuesta usando 2 robots y en la cual el robot R2 calcula las ofertas para la frontera  $f_5$ . Explotación  $\rho_2 + \alpha_2^5$ , exploración  $\rho_1 + \alpha_1^5$  y cohesión  $\varphi$ .

Un esquema de cálculo de oferta se muestra en la Figura 4.3.2 (donde  $\varphi$  es una variable en función del rango de comunicación). En esta figura también podemos apreciar cada uno de los parámetros incluidos en mi propuesta (explotación, exploración y cohesión). Igualmente se muestran las variables al ofertar para la frontera  $f_5$ . Este proceso se repite para cada frontera excluyendo las que se encuentren fuera del diámetro con radio  $\varphi$ .

La oferta del robot  $k$  hacia la frontera  $f$ , denotada como  $b_k^f$ , se compone entonces por el costo para el robot para alcanzar la frontera considerada, menos el costo promedio para los otros robots, penalizada por la función de cohesión. La expresión se muestra en

la Ecuación 4.1, donde  $\varphi$  está en función del rango de comunicación  $\beta$ , en una primera instancia  $\varphi = 0.9\beta$  permitiendo incrementar el valor de  $\varphi$  cuando no existan fronteras dentro del rango de comunicación del robot, con la finalidad de permitir al robot seleccionar objetivos más allá del alcance de la comunicación permitiendo así el éxito de la exploración. Cada robot debe seleccionar sus objetivos de manera tal que el costo total para equipo sea el más bajo posible.

$$b_k^f = \begin{cases} (\rho_k + \alpha_k^f) - \frac{\sum_{i=1, i \neq k}^n (\rho_i + \alpha_i^f)}{n-1} & \text{si } \exists g(\cdot) \mid \text{distance}(g(\cdot), \gamma(r_k, f)) < 0.9\varphi \text{ o } n = 1 \\ \infty & \text{en otro caso} \end{cases} \quad (4.1)$$

Tal y como puede apreciarse, todos los parámetros de la función de utilidad de mi propuesta están basados en el cálculo de distancias precisas entre distintos puntos en cada mapa local. Si bien estas distancias pueden fácilmente calcularse por medio de un algoritmo clásico de frente de onda, es evidente que la precisión de los costos es tributaria de la precisión proporcionada por dicho algoritmo. Por esta razón, para el cálculo de las distancias de trayecto en el espacio libre y obtener la planificación de desplazamientos requerida, nosotros usamos el algoritmo EEDT que desarrollamos específicamente para aumentar la precisión, descrito en el Capítulo 3.

Como se explicará con mayor detalle en la sección 4.3.3, sólo el robot que necesita un nuevo objetivo realiza el proceso de ofertas. Durante este proceso el robot solicita la información a los otros miembros del equipo (mapas individuales y objetivos actuales). Basado en esta información, el robot estima las ofertas que harían los demás miembros del equipo, calcula las suyas y realiza un emparejamiento de mínimo costo para seleccionar su nuevo objetivo, buscando el mejor aporte para el equipo, pero sin interferir con las decisiones de los otros miembros del equipo.

Las ofertas estimadas por el robot pueden ser vistas como los pesos de un grafo bipartita completo ponderado, con lo cual el problema de selección de la frontera para el robot considerado puede definirse como: Dado un grafo bipartita ponderado  $G = (R \cup F, R \times F)$ , donde el conjunto  $R$  representa robots, el conjunto  $F$  representa las fronteras disponibles y  $b_r^f \mid r \in R, f \in F$  son los pesos de las aristas, encontrar un emparejamiento  $M$  de  $R$  a  $F$  con el mínimo peso.

En general, los conjuntos  $R$  y  $F$  no son del mismo tamaño, por lo cual necesitamos añadir fronteras virtuales o robots virtuales en el grafo, hasta obtener dos conjuntos  $R$  y  $F$  del mismo tamaño  $m$ , definido como  $R = r_1, r_2, \dots, r_m$  y  $F = f_1, f_2, \dots, f_m$ , con un valor muy grande sobre las aristas virtuales que completan el nuevo grafo bipartita.

Matemáticamente, el problema puede definirse como: dada una matriz  $B$  de  $m \times m$ , encontrar una permutación  $\pi$  de  $\{1, 2, 3, \dots, m\}$  tal que la suma de los pesos  $B(i, \pi(i))$  es mínima, llamado *emparejamiento óptimo*.

En el caso de mi propuesta, el emparejamiento óptimo de  $R$  a  $F$  se obtiene aplicando el método Húngaro [49], por su probada eficiencia para resolver este problema.

### 4.3.3 Algoritmos Desarrollados

En mi propuesta, cada robot ejecuta el mecanismo de ofertas de la siguiente manera: Cuando el robot ha alcanzado su objetivo, solicita los mapas y los objetivos a los otros robots mediante el proceso de comunicación (Algoritmo 8). Con esta información, el robot actualiza su mapa en el proceso Generador de mapa, establece las fronteras candidatas a exploración en el proceso Cálculo de ofertas (Algoritmo 7), calcula la tabla de ofertas  $b_k^f$  y resuelve el problema de emparejamiento óptimo usando el método Húngaro [49], con lo cual el robot selecciona el objetivo que más beneficia a la exploración. El robot transmite el nuevo objetivo a los otros miembros del equipo (que se encuentran dentro de su rango de comunicación) mediante el proceso de comunicación, para poder ser notificado en el caso

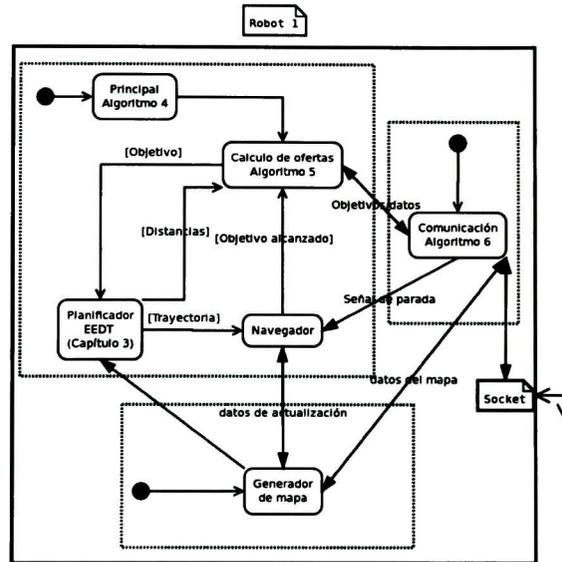


Figura 4.6: Esquema de mi propuesta en cada robot.

de que algún otro robot descubra dicho objetivo antes. Una vez seleccionado el objetivo se planifica su trayectoria con el Planificador EEDT (Capítulo 3) para después dirigirse hacia la frontera seleccionada mediante el proceso navegador *Nearness Diagram Navigation* [58]. Si durante su desplazamiento el robot recibe una señal de parada, significa que la frontera ha sido explorada o seleccionada recientemente por otro miembro del equipo, entonces el robot se detiene e inicia un nuevo proceso de ofertas. El esquema completo de mi propuesta para 1 robot se muestra en la Figura 4.6 y el procedimiento principal se resume en el Algoritmo 6.

El proceso para el cálculo de la tabla de ofertas  $b_k^f$  se describe en el Algoritmo 7. A partir del mapa actualizado, se detectan las celdas correspondientes a las fronteras inexploradas y se agrupan en *clusters* para ser consideradas como una única frontera  $f_i$  al centro de dicho cluster. De este conjunto de fronteras, solamente son consideradas, para el cálculo de ofertas, las fronteras que se encuentren a una distancia menor que  $\varphi$  (parámetro de

#### 4. Exploración Multi-Robot Usando Auto-Ofertas bajo Restricción en el Rango de la Comunicación

79

---

##### Algoritmo 6 Exploración Principal

---

**Input:** Mapa[n][m] inexplorado

**Output:** Mapa[n][m] con  $F = 0$

startMappingOnline()

send(MyGoal)

**repeat**

  solicita los otros mapas

  actualiza el mapa

  solicita los objetivos de los otros robots

  actualiza los objetivos de los otros robots

  Calcula las ofertas (Algoritmo 7)

  envía el objetivo seleccionado

**repeat**

    se dirige hacia el objetivo

**if** Si el objetivo de otro robot no es una frontera **then**

      envía una señal de parada

**end if**

**until** alcanzar el objetivo **OR** recibir una señal de parada

**until** mapa completo

envía señal de parada a los otros miembros del equipo

---

comunicación) de los otros robots; sin embargo, si todas las fronteras se encuentran a una distancia euclidiana mayor a la del parámetro  $\varphi$ , es necesario incrementar el valor de dicha variable ( $\varphi$ ) con la finalidad de integrar al proceso de ofertas al menos una frontera. Finalmente, se utiliza el método Húngaro para la selección de la frontera.

Si al robot se le asigna una frontera virtual (i.e. no existe una frontera que se le pueda asignar al robot y que beneficie a la exploración), entonces el robot selecciona la frontera más cercana como su objetivo. Es importante resaltar que dos o más robots pueden seleccionar la misma frontera como objetivo inmediato, con la finalidad de mantener el puente de comunicación.

Paralelo al Algoritmo 6 se ejecuta el proceso de comunicación (Algoritmo 8), el cual es responsable de recibir la información de mapas y objetivos de los otros robots, así como atender las solicitudes de información de los otros miembros del equipo.

**Algoritmo 7** Cálculo de tabla de ofertas  $b_k^f$  $f$  = fronteras detectadas $j = 1$ **repeat** $\varphi = \beta \times j$ 

actualiza las ofertas (Ecuación 4.1)

 $j = j + 0.5$ **until**  $\exists b_{ID}^f \neq \infty$ ;  $ID = \text{Robot actual}$ Encuentra una permutación  $\pi$  de  $R$  a  $F$  de emparejamiento óptimo.

Asigna un objetivo al robot actual

**if** el objetivo es asignado a una frontera real **then**

selecciona la frontera como objetivo

**else**

selecciona la frontera más cercana como objetivo

**end if**

Debido a que dos robots directamente comunicados entre sí pueden estar en contacto a través de otro(s) robot(s), se ha implementado en los Algoritmos 6 y 8 un *timestamp* con el que se marcan los mensajes, para verificar si la información recibida es más reciente que la información que ya se tiene, y descartar así información antigua. Este *timestamp* se actualiza en cada robot cada vez que el objetivo es enviado, con lo cual se busca tener siempre la información más actualizada acerca de los objetivos de cada miembro del equipo.

En mi propuesta, la planificación global de trayectorias para alcanzar los objetivos se obtiene (como ya se explicó con anterioridad) directamente del algoritmo EEDT [30], con lo cual obtenemos la mejor trayectoria como una secuencia de puntos intermedios en trayectorias rectilíneas sin colisión. Para realizar las operaciones de navegación, empleamos el algoritmo *Nearness Diagram Navigation* [58] como planificador local para controlar el desplazamiento real del robot y resolver localmente cualquier perturbación que pudiera aparecer durante el movimiento.

---

**Algoritmo 8** Comunicación

---

```
repeat
  data = receiveData()
  switch data
  case Solicitud de objetivos:
    envía mi objetivo y el objetivo conocido de los otros robots
  case Recibe objetivo:
    actualiza los objetivos de los otros robots
  case Solicitud de mapa:
    envía el mapa
  case Recibe mapa:
    actualiza el mapa
  case STOP:
    pone la señal de parada en (True)
  case FINISH:
    pone la señal de parada en (True)
until 1
```

---

## 4.4 Resultados Numéricos

En esta sección evalué el desempeño de mi propuesta. El simulador usado para realizar las pruebas es el ambiente Player/Stage<sup>1</sup> [12], el cual modela las interferencias entre los robots y las propiedades físicas de sensores y actuadores. Para realizar los experimentos, usé los modelos de los robots comerciales Pioneer 2-DX, equipados con sensores láser de espejo giratorio, con 360 grados de visión y un alcance de lectura de 4 metros, que son ejemplos típicos de robots y sensores encontrados en las aplicaciones reales.

### 4.4.1 Motivación para el Uso de un Simulador

Aunque la aplicación final de un algoritmo en robótica debe de probarse en un robot real, existen muchas buenas razones para examinar el comportamiento en simulador. La existencia de gran cantidad de simuladores de robótica (e.g. Player-Stage-Gazebo [12],

---

<sup>1</sup>Player/Stage está disponible de manera gratuita bajo Licencia Pública General GNU en la página <http://playerstage.sourceforge.net>.

Webots [57], Robot Operating System [71], Microsoft Robotics Developer Studio [41], entre otros) demuestra que el uso de los simuladores está muy extendido. Algunas de las ventajas del uso de un simulador generalmente incluyen:

**Costo** Los experimentos en robots reales pueden tomar mucho tiempo, energía y dinero. Puede ser mucho más barato mostrar que los algoritmos funcionan en simulación antes de implementarlos en robot reales.

**Simplicidad** Algunas complicaciones experimentadas en robots reales pueden abstraerse con el uso de un simulador, pudiendo así enfocarse el desarrollo en aspectos específicos de la propuesta.

**Control** Es más fácil examinar el comportamiento de un método, teniendo un control completo de las distintas variables y condiciones que afectan el problema.

**Repetibilidad** Los experimentos pueden llevarse a cabo varias veces y mostrar la consistencia del trabajo.

**Disponibilidad** En ocasiones no se cuenta con un robot real, por lo cual es necesario contar con herramientas de software en las cuales realizar las pruebas necesarias para probar las propuestas desarrolladas.

Por disponibilidad de material, el esquema propuesto en esta tesis fue probado en el simulador de comunidades de robots Player/Stage [12], ampliamente utilizado por la comunidad científica. El software Player/Stage proporciona herramientas de código abierto que proveen un ambiente de prueba para robots, particularmente para ambientes multi-robot. El proyecto proporciona el servidor de dispositivos robóticos Player, que da acceso a sensores y actuadores desde programas clientes. El simulador 2D para múltiples robots Stage y el simulador 3D Gazebo para múltiples robots, modelan de manera realista el comportamiento físico de motores y sensores.

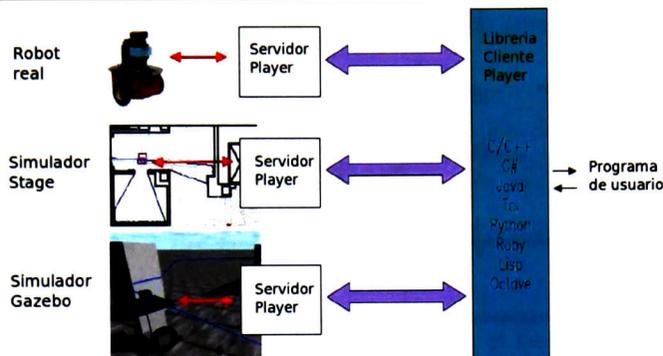


Figura 4.7: Abstracción de Hardware con Player.

El lenguaje de desarrollo para el ambiente Player/Stage es C y C++. Player/Stage es en realidad una máquina virtual de robots, es decir, la aplicación se escribe como si se estuviera programando el robot físico, por lo que se tiene una completa portabilidad del código directamente al hardware. Esta capacidad representa una de las principales ventajas del proyecto Player/Stage, ya que los programas escritos usando estas librerías pueden ser utilizados en forma transparente sobre simulador o sobre algunos tipos de robot reales, como son el Pioneer, Segway y Khepera (Figura 4.7).

### 4.4.2 Algoritmos Usados para la Comparación

Con la finalidad de realizar una comparativa del comportamiento de mi propuesta, se implementaron dos algoritmos para la exploración multi-robot:

- Frontier-Based Exploration Using Multiple Robots propuesto por Yamauchi en [95]
- Coordinated multi-robot exploration Propuesto por Burgard et al. en [16] y Stachniss et al. en [78].

Se decidió compararse con estas dos propuestas ya que la propuesta de Yamauchi [95] es con la cual se comparan muchas de las propuestas del estado del arte y la propuesta

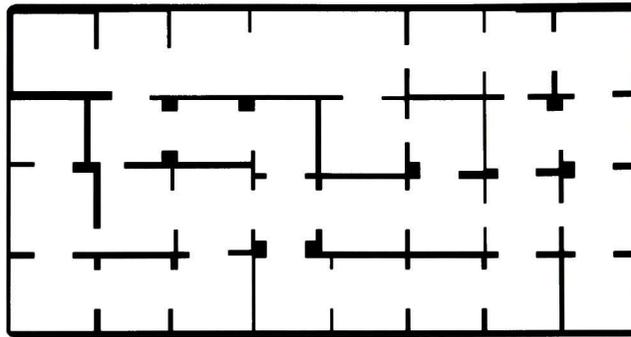


Figura 4.8: Mapa usado para las simulaciones (50 × 25 metros)

presentada en [16] y [78] se estudia el impacto de un rango de comunicación limitado, es tolerante a fallas y es de las propuestas más citadas en el estado del arte.

### 4.4.3 Pruebas Realizadas

Un ejemplo del medio ambiente a explorar se muestra en la Figura 4.8. El equipo de robots es simulado por programas idénticos, uno por robot, ejecutándose de forma paralela (el simulador permite incluso que los programas se ejecuten en computadoras diferentes, conectadas en red). Para validar mi algoritmo de exploración, llevé a cabo experimentos en los cuales variamos la cantidad de robots del equipo (de 1 a 5 robots) y el rango de comunicación (15 metros, 20 metros y 30 metros). Comparé mi propuesta con la propuesta reportada por Yamauchi [95], en la cual cada robot selecciona el área inexplorada más cercana, y con la propuesta de Stachniss [78], en la cual el líder asigna los objetivos tratando de maximizar la utilidad a la vez que minimiza la distancia recorrida.

Para cada caso de estudio realizamos 31 pruebas. Para evaluar la cohesión del equipo medimos el tiempo durante el cual alguno de los robots se mantuvo aislado el equipo (tiempo total fuera del rango de comunicación) y el tiempo máximo que algún robot se aisló del equipo (tiempo máximo fuera del rango de comunicación) y la distancia euclidiana

máxima que el robot aislado se alejó del resto del equipo (máxima distancia euclidiana fuera del rango de comunicación).

Por otro lado, para evaluar el parámetro de explotación de cada robot medimos la distancia total recorrida por el equipo de robots, lo que muestra si los robots eligen objetivos cercanos a sus posiciones actuales o si, por el contrario, eligen objetivos más alejados.

Para evaluar el parámetro de exploración, medimos el tiempo total necesario para completar el mapa, lo que muestra la eficiencia que tiene la estrategia para lograr la dispersión del equipo de robots en el medio ambiente a explorar.

Los resultados de estos experimentos se muestran en gráficas de caja y bigote, en las cuales mostramos la medida más pequeña (bigote inferior), el cuartil inferior (parte baja de la caja), la mediana (línea dentro de la caja), el cuartil superior (parte superior de la caja) y la medida mas grande registrada (bigote superior).

En la Figura 4.9 y la Tabla 4.1 podemos ver el tiempo total fuera del rango de comunicación, que es indicativo del tiempo total que la red de comunicación entre los robots se mantuvo rota. Es importante resaltar que si un robot se avería durante el tiempo que permanece aislado, toda la información nueva que haya podido recolectar sobre el ambiente se habrá perdido, pues ningún otro miembro del equipo posee dicha información.

Los tiempos son mayores usando la propuesta de Stachniss con un rango de comunicación de 15, 20 y 30 metros, ya que bajo el esquema de Stachniss los robots tienden a distribuirse sobre el medio ambiente. Con la propuesta de Yamauchi usando 2 y 3 robots el tiempo total fuera del rango de comunicación es menor que los tiempos obtenidos por la propuesta de Stachniss, pero usando cuatro y cinco robots con la propuesta de Yamauchi, los tiempos son similares a los tiempos obtenidos por la propuesta de Stachniss.

Usando mi propuesta con un rango de comunicación de 15 metros, el tiempo total

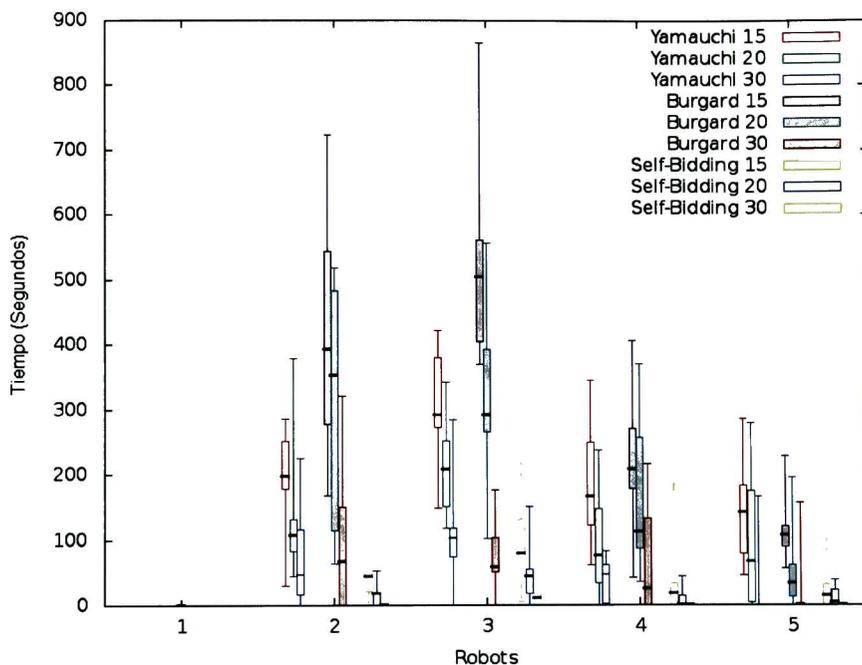


Figura 4.9: Tiempo total fuera del rango de la comunicación

Propuesta	Rango de comunicación	Robots			
		2	3	4	5
Yamauchi	15 m	201.1	290.9	166	141.05
	20 m	106.55	207.2	75.1	65.55
	30 m	47.7	101.9	47	0
Stachniss	15 m	391.4	503.1	207	105.4
	20 m	351.4	289.5	111.6	32.6
	30 m	66.5	57	24.65	0
Auto-Ofertas	15 m	43.2	77.7	16.9	14.7
	20 m	17.9	44	0	3.45
	30 m	0	11.2	0	0

Tabla 4.1: Tiempo total fuera del rango de la comunicación (segundos).

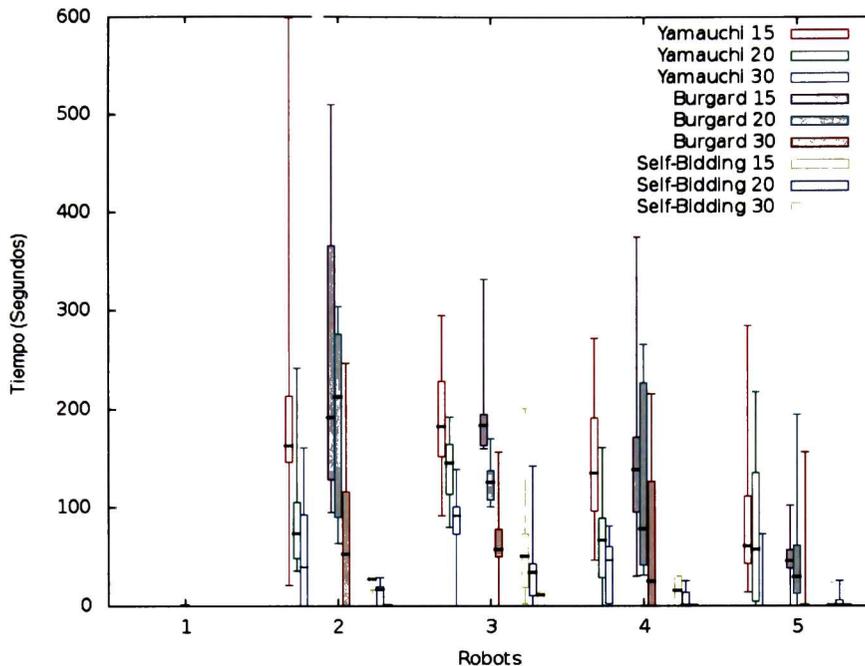


Figura 4.10: Tiempo máximo de aislamiento de algún robot (segundos).

fuera del rango de comunicación es menor que usando la propuesta de Yamauchi o la propuesta de Stachniss, este comportamiento se mantiene para 20 y 30 metros de rango de comunicación. Con esto se demuestra que mi propuesta logra mantener la cohesión del equipo con lo cual se intenta minimizar el aislamiento de alguno de los miembros del equipo. Si alguno de los robots se avería, poca o ninguna información se habrá perdido.

A continuación, analizamos el tiempo máximo que algún robot permaneció aislado, esto es, el tiempo durante el cual el mapa individual del robot aislado no integró información recabada por los otros robot y que a su vez los otros robot no integraron la información recabada por el robot aislado. La Figura 4.10 y la Tabla 4.2 muestran el tiempo máximo durante el cual algún robot se aisló de los otros miembros del equipo.

Propuesta	Rango de comunicación	Robots			
		2	3	4	5
Yamauchi	15 m	162.6	181.9	135	60.65
	20 m	72.8	145.55	66	56.4
	30 m	39.1	91.65	47	0
Stachniss	15 m	191.8	184	137.8	45.3
	20 m	212.7	125.1	77.65	29.2
	30 m	52.3	57	24.65	0
Auto-Ofertas	15 m	26.4	50.3	14.7	0.35
	20 m	16.15	33.95	0	1.45
	30 m	0	10.1	0	0

Tabla 4.2: Tiempo máximo de aislamiento (segundos).

De la Figura 4.10 y de la Tabla 4.2 podemos observar que en mi propuesta, cuando algún robot queda fuera del rango de comunicación, éste trata de reingresar a la red rápidamente. Por otra parte, la propuesta de Yamauchi y la propuesta de Stachniss no tratan de mantener al robot dentro del rango de comunicación de los otros robots, pudiendo así mantener a algún robot fuera del rango de comunicación de los otros por intervalos grandes ya que el robot aislado no intenta reingresar a la red de los otros miembros del equipo.

También podemos observar que el tiempo máximo fuera del rango de comunicación con la propuesta de Yamauchi y con la propuesta de Stachniss, son mayores que con mi propuesta y en consecuencia el riesgo de que algún robot falle cuando se encuentre aislado es mayor, provocando así que la información obtenida durante el aislamiento se pierda debido a que no se ha podido compartir.

Otra forma de analizar la cohesión del equipo, es estudiando la distancia euclidiana máxima más allá del rango de comunicación a la que se encuentra algún robot aislado, es decir, cuando un robot se aísla, qué tanto se aleja del resto del equipo. En la Figura 4.11 y la Tabla 4.3 podemos ver que en mi estrategia, el robot aislado viaja una distancia más corta fuera del rango de comunicación que con la propuesta de Yamauchi y con la propuesta de

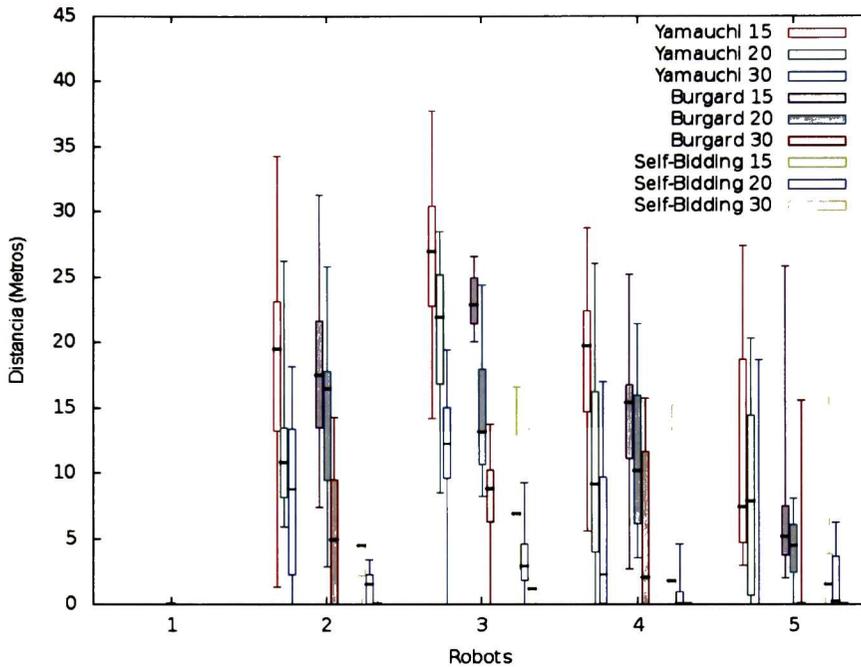


Figura 4.11: Distancia euclidiana máxima fuera del rango de comunicación

Stachniss. Esta situación demuestra que mi propuesta trata de mantener la cohesión del equipo evitando que los robots se dispersen más allá del rango de comunicación permitido.

Para evaluar el comportamiento durante la exploración analizamos el tiempo total requerido para completar la tarea. En la Figura 4.12 y en la Tabla 4.4 se muestra el tiempo total necesario para completar la exploración y en la cual podemos ver que para un equipo de 2 robot y un rango de comunicación de 15 metros, mi propuesta toma más tiempo para completar la exploración. Esto se debe a que los robots tratan de mantenerse dentro del rango de comunicación del otro, con lo cual los robots tienden a mantenerse juntos durante la exploración y no se dispersan adecuadamente en el medio ambiente. Por otra parte, dado que en la propuesta de Yamauchi y en la propuesta de Stachniss los robots no tratan de mantenerse dentro del rango de comunicación entre ellos, pueden separarse

Propuesta	Rango de comunicación	Robots			
		2	3	4	5
Yamauchi	15 m	19.478	26.997	19.725	7.4101
	20 m	10.794	21.946	9.138	7.828
	30 m	8.83	12.31	22.284	0
Stachniss	15 m	17.499	2.881	15.408	5.117
	20 m	16.423	13.122	10.207	4.421
	30 m	4.903	8.8	2.028	0
Auto-Ofertas	15 m	4.455	6.841	1.776	1.481
	20 m	1.514	2.846	0	0.154
	30 m	0	1.105	0	0

Tabla 4.3: Distancia euclidiana máxima fuera del rango de la comunicación (metros).

lo suficiente como para explorar áreas completamente distintas y acelerar el proceso de exploración. Sin embargo, como lo vimos en las estadísticas anteriores, ambas estrategias rompen los vínculos de comunicación por mucho tiempo, lo que puede ser una desventaja en caso de falla de alguno de los dos robots.

Es importante resaltar que mi propuesta, cuando el equipo de robots es más numeroso y/o cuenta con un mejor rango de comunicación, logra dispersar a los robots de manera más eficiente en el medio ambiente, comparada con las otras dos estrategias. Así, para un equipo de 3 robots y un rango de comunicación de 15 metros, mi propuesta, la propuesta de Yamauchi y la propuesta de Stachniss proporcionan tiempos similares, mientras que mi propuesta con un rango de comunicación de 30 metros arroja mejores resultados que los arrojados por la propuesta de Yamauchi y que la propuesta de Stachniss.

Mejor aún, para un equipo de 4 y 5 robots y rangos de comunicación de 15 metros, 20 metros y 30 metros, los tiempos de exploración usando mi propuesta son menores que los tiempos obtenidos con la propuesta de Yamauchi y con la propuesta de Stachniss. Finalmente, para un equipo de 5 robots y un rango de comunicación de 15 metros, podemos observar que la propuesta de Stachniss arroja resultados ligeramente mejores que mi propuesta y que la propuesta de Yamauchi, pero mi propuesta logra mantener los

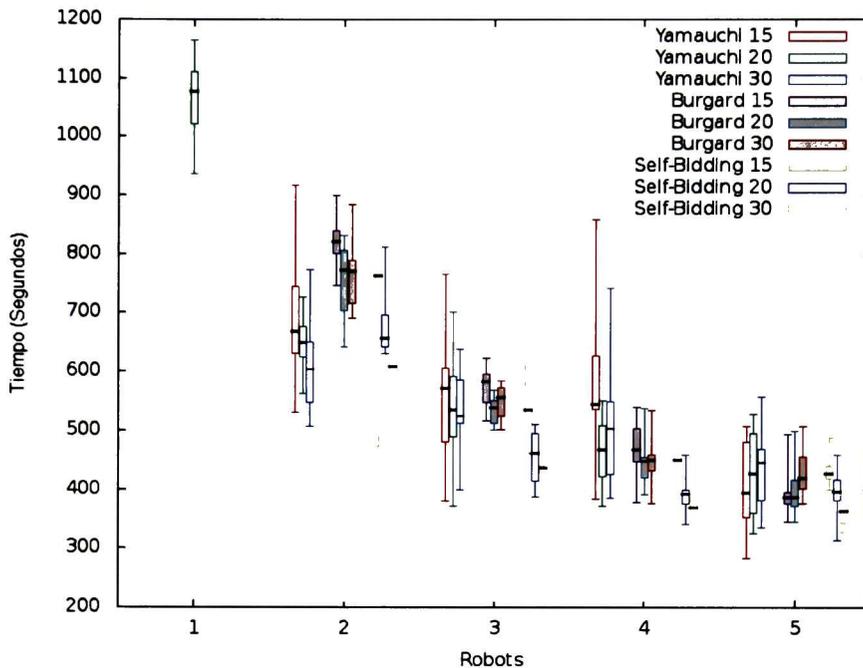


Figura 4.12: Tiempo total necesario para completar la exploración.

tiempos de desconexión de la red de comunicación cercanas a cero.

En la Figura 4.13 y en la Tabla 4.5 comparamos la distancia total recorrida por el equipo de robots, necesaria para completar la exploración. Es posible identificar este parámetro con la energía necesaria para realizar la exploración. Si la distancia total recorrida se mantiene constante, independientemente del número de robots, indica que la exploración se realiza de manera eficiente, pues no hay redundancia en explorar sitios que ya han sido explorados previamente.

En estas estadísticas podemos apreciar el mismo fenómeno que para el tiempo total requerido para la exploración: cuando mi propuesta cuenta con pocos robots o con un rango de comunicación muy corto, el desempeño de esta métrica se ve muy afectado, reflejando el costo de mantener la cohesión del equipo de robots. Sin embargo, al contar

Propuesta	Rango de comunicación	Robots			
		2	3	4	5
Yamauchi	15 m	668	572.1	544.1	393.5
	20 m	648.3	534.65	466.15	426.35
	30 m	604.6	525.5	504.5	445.3
Stachniss	15 m	820.9	582.2	467.45	386.4
	20 m	773.05	539.35	447.15	384.95
	30 m	769.9	555.05	450	419.5
Auto-Ofertas	15 m	762.25	535.3	449.6	426.55
	20 m	655.75	462.05	391.8	395.5
	30 m	608.8	435.9	368.2	363.4

Tabla 4.4: Tiempo total necesario para completar la exploración (segundos).

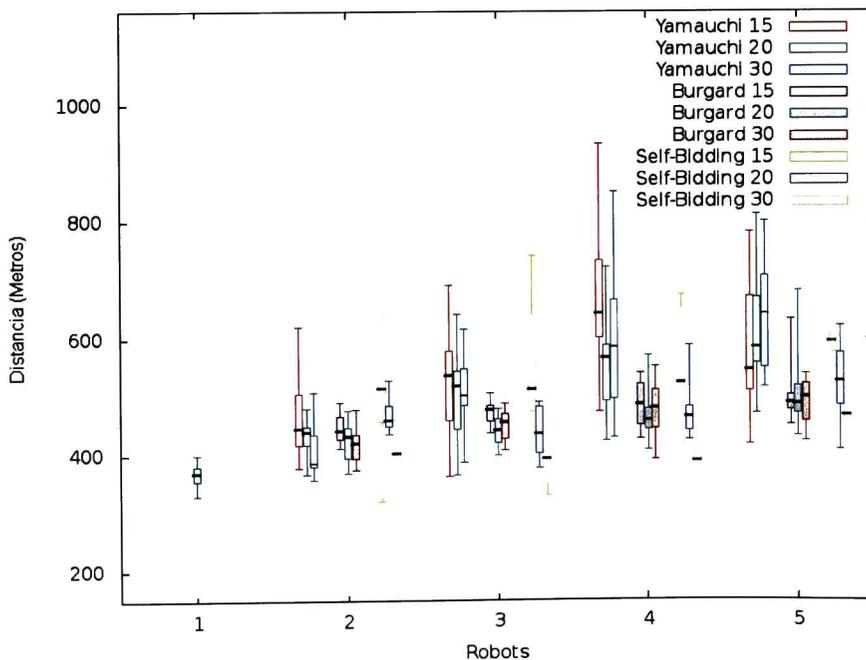


Figura 4.13: Distancia total recorrida por el equipo de robot

#### 4. Exploración Multi-Robot Usando Auto-Ofertas bajo Restricción en el Rango de la Comunicación

Propuesta	Rango de comunicación	Robots			
		2	3	4	5
Yamauchi	15 m	445.03	534.86	642.52	544.81
	20 m	438.22	517.3	566.63	583.497
	30 m	385.48	501.63	584.82	643.113
Stachniss	15 m	440.81	476.46	486.54	487.196
	20 m	430.83	440.94	458.38	485.486
	30 m	419.3	455.45	479.6	497.286
Auto-Ofertas	15 m	512.78	512.22	523.49	592.758
	20 m	459.24	435.25	464.05	525.477
	30 m	402.19	391.33	387.56	466.43

Tabla 4.5: Distancia total recorrida por el equipo de robot (metros).

con más robots o con un rango de comunicación mayor, mi propuesta logra distribuir eficientemente los robots en el medio ambiente, logrando mejores resultados que las propuestas de Yamauchi y de Stachniss.

Así por ejemplo, podemos ver que mi propuesta para un equipo de 2 robots con un rango de comunicación de 15 metros y 20 metros, la distancia total recorrida es mayor que en la propuesta de Yamauchi y que la propuesta de Stachniss, pues en mi propuesta los robots tienden a estar cerca entre sí para evitar los cortes en la comunicación, con lo cual se incrementa la redundancia durante la exploración. Con un rango de comunicación de 30 metros, la distancia total recorrida utilizando mi propuesta es muy cercana a la recorrida por un solo robot, con lo cual podemos deducir que la redundancia durante la exploración es mínima.

Para apreciar cómo los robots comparten la información utilizando las diferentes propuestas, en las Figuras 4.14-4.16, comparamos los mapas individuales de 3 robot usando la propuesta de Yamauchi, la propuesta de Stachniss y mi propuesta. En cada Figura se muestra el primer mapa tomado a los 2 segundos de iniciada la exploración, para permitir el primer intercambio de información.

En la Figura 4.14 podemos ver los mapas individuales de los robot usando la propuesta

de Yamauchi. Como puede apreciarse, el mapa del robot 3 es muy diferente a los mapas de los robots 1 y 2. En este caso, ya que el algoritmo no toma en cuenta el rango de comunicación para la selección de sus objetivos, el robot 3 se desplaza fuera del rango de comunicación de los otros miembros del equipo, causando así que se rompan todos los posibles puentes de comunicación y consecuentemente que los mapas individuales contengan información no actualizada. Si el robot 3 se avería mientras se encuentra fuera del rango de comunicación, toda la información que ya había recolectado se habrá perdido.

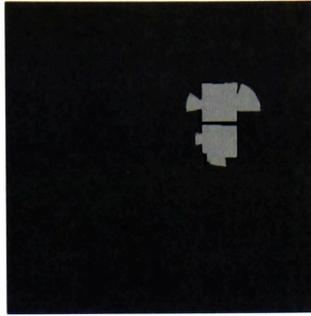
En la Figura 4.15 podemos ver los mapas individuales obtenidos con la propuesta de Stachniss. Como puede apreciarse, los tres robots poseen básicamente mapas distintos. En este caso los robots se coordinan y tratan de maximizar la utilidad lo cual implica que los robots tienden a dispersarse eficientemente en el área a explorar pero, ya que en esta propuesta no se trata de mantener dentro del rango de comunicación a los robots, éstos tienden a aislarse y consecuentemente, los mapas individuales contienen datos no actualizados. La pérdida de cualquiera de los robots involucra una pérdida importante de información sobre la exploración.

La Figura 4.16 muestra los mapas individuales de 3 robots usando mi propuesta. El comportamiento de cohesión de los miembros del equipo durante todo el proceso de exploración garantiza que los robots comparten toda la información recolectada con al menos un miembro del equipo en todo momento, manteniendo la conectividad de la red de comunicación. En consecuencia, los mapas individuales de cada robot son muy similares entre sí, y la pérdida de alguno de los robots no involucra una pérdida de información de exploración.

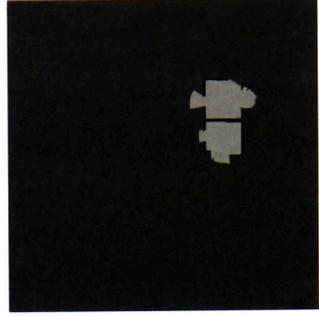
De las Figuras 4.14-4.16, podemos ver que mi propuesta realmente logra mantener a los robots dentro del rango de comunicación de forma más eficiente que la propuesta de Yamauchi y que la propuesta de Stachniss. Con esta estrategia, se logra que el intercambio de información pueda darse con mayor frecuencia durante la exploración, por lo que cada



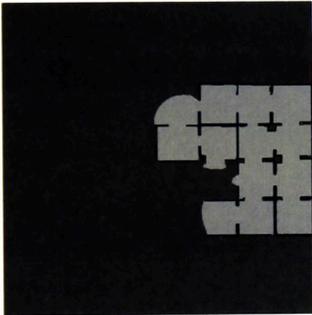
(a) Mapa local del robot 1 al inicio de la exploración (tiempo = 2 segundos)



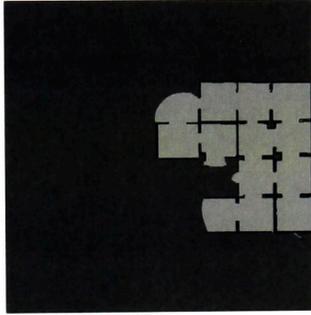
(b) mapa local del robot 2 al inicio de la exploración (tiempo = 2 segundos)



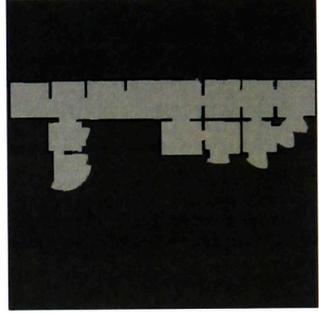
(c) mapa local del robot 3 al inicio de la exploración (tiempo = 2 segundos)



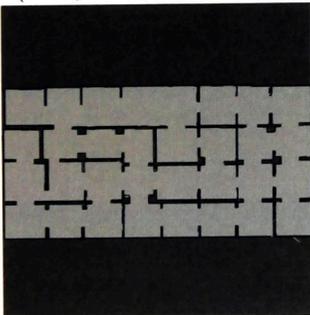
(d) Mapa local del robot 1 a mitad de la exploración (tiempo = 240 segundos)



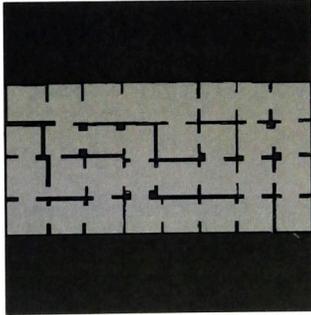
(e) mapa local del robot 2 a mitad de la exploración (tiempo = 240 segundos)



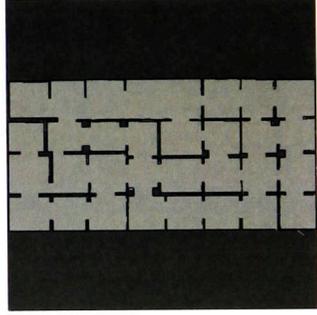
(f)mapa local del robot 3 a mitad de la exploración (tiempo = 240 segundos)



(g) Mapa local del robot 1 al final de la exploración (tiempo = 572 segundos)

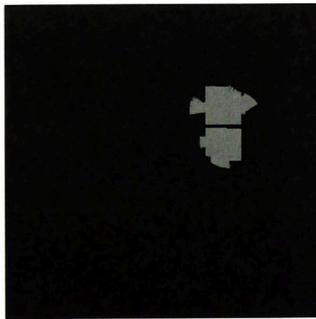


(h) mapa local del robot 2 al final de la exploración (tiempo = 572 segundos)

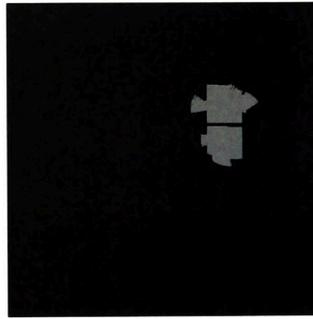


(i)mapa local del robot 3 al final de la exploración (tiempo = 572 segundos)

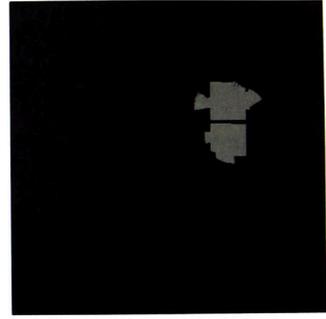
Figura 4.14: Mapas locales usando la propuesta de Yamauchi (15 metros de rango de comunicación).



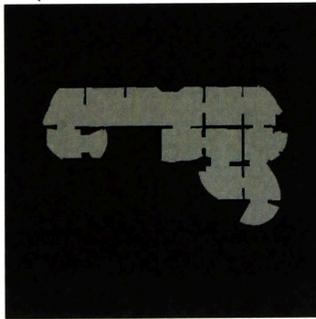
(a) Mapa local del robot 1 al inicio de la exploración (tiempo = 2 segundos)



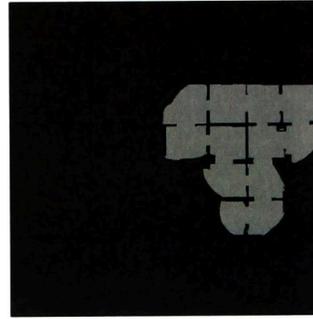
(b) mapa local del robot 2 al inicio de la exploración (tiempo = 2 segundos)



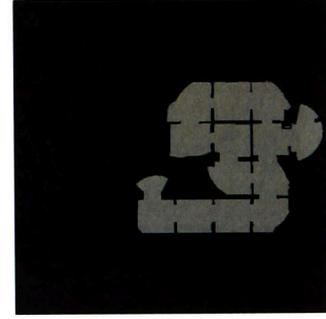
(c) mapa local del robot 3 al inicio de la exploración (tiempo = 2 segundos)



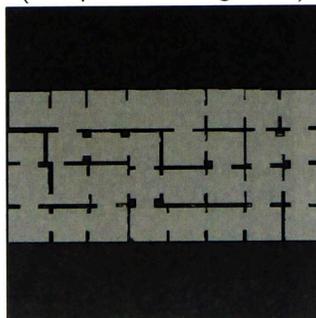
(d) Mapa local del robot 1 a mitad de la exploración (tiempo = 266 segundos)



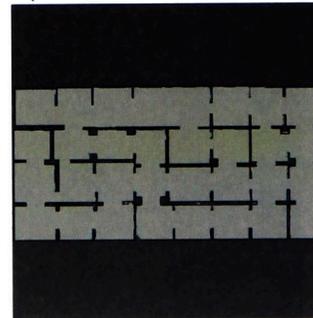
(e) mapa local del robot 2 a mitad de la exploración (tiempo = 266 segundos)



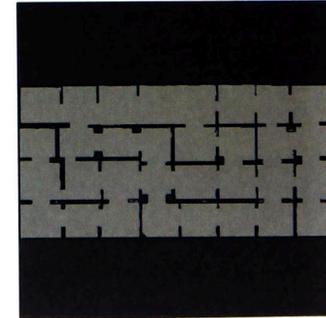
(f)mapa local del robot 3 a mitad de la exploración (tiempo = 266 segundos)



(g) Mapa local del robot 1 al final de la exploración (tiempo = 580 segundos)



(h) mapa local del robot 2 al final de la exploración (tiempo = 580 segundos)



(i)mapa local del robot 3 al final de la exploración (tiempo = 580 segundos)

Figura 4.15: Mapas locales usando la propuesta de Stachniss (15 metros de rango de comunicación).

#### 4. Exploración Multi-Robot Usando Auto-Ofertas bajo Restricción en el Rango de la Comunicación



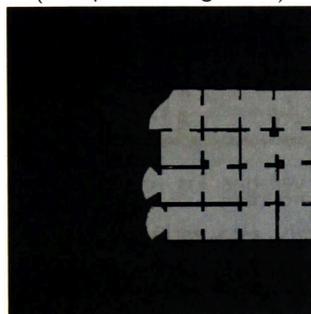
(a) Mapa local del robot 1 al inicio de la exploración (tiempo = 2 segundos)



(b) mapa local del robot 2 al inicio de la exploración (tiempo = 2 segundos)



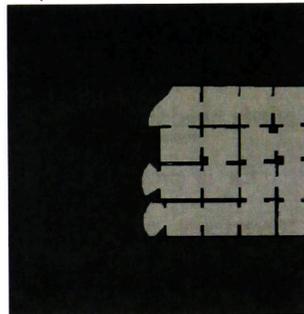
(c) mapa local del robot 3 al inicio de la exploración (tiempo = 2 segundos)



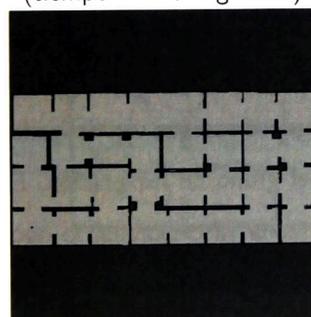
(d) Mapa local del robot 1 a mitad de la exploración (tiempo = 219 segundos)



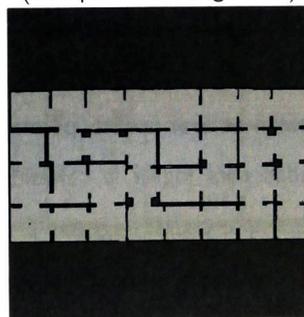
(e) mapa local del robot 2 a mitad de la exploración (tiempo = 219 segundos)



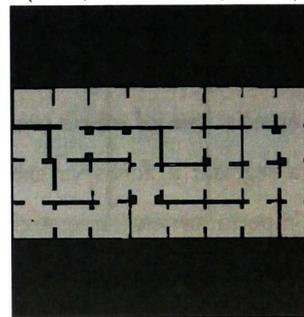
(f)mapa local del robot 3 a mitad de la exploración (tiempo = 219 segundos)



(g) Mapa local del robot 1 al final de la exploración (tiempo = 535 segundos)



(h) mapa local del robot 2 al final de la exploración (tiempo = 535 segundos)



(i)mapa local del robot 3 al final de la exploración (tiempo = 535 segundos)

Figura 4.16: Mapas locales usando Auto-ofertas.

robot posee un mapa actualizado con la información recientemente recolectada por los otros miembros del equipo.

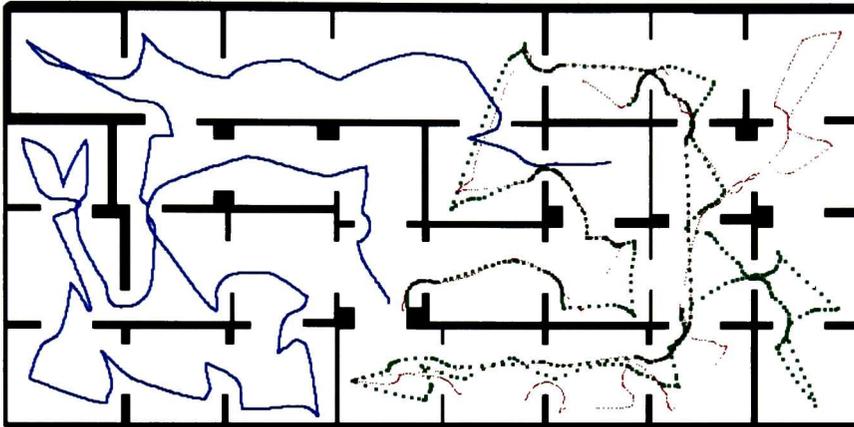
Finalmente, para ejemplificar el tipo de trayectorias seguidas por los robots, mostramos la exploración con 3 robots y un rango de comunicación de 20 metros y 30 metros, usando la propuesta de Yamauchi (Figura 4.17), la propuesta de Stachniss (Figura 4.18), y con mi propuesta (Figura 4.19).

En la propuesta de Yamauchi (Figura 4.17) podemos ver que en ambos casos (rango de comunicación = 20 y rango de comunicación = 30) dos robots seleccionan objetivos muy similares, estorbándose continuamente durante la exploración, mientras que el tercer robot se separa demasiado de los otros miembros del equipo, con lo cual este tercer robot puede quedar aislado.

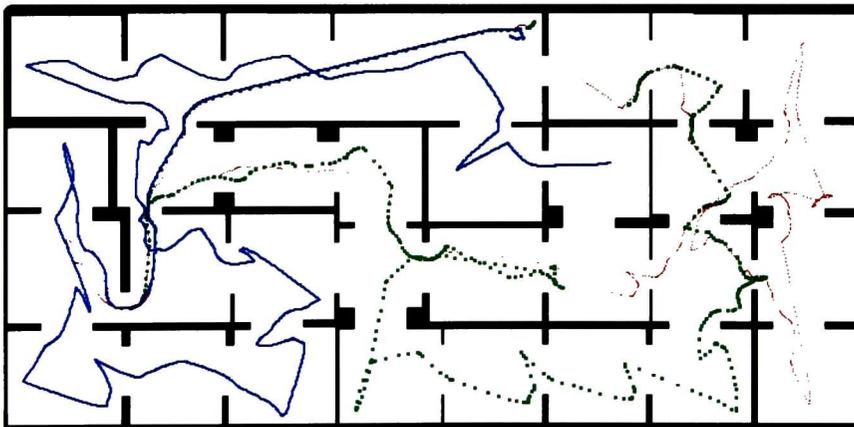
En la propuesta de Stachniss (Figura 4.18) podemos ver que los robots generan trayectorias que los dispersan en el ambiente. Sin embargo, esta estrategia tiene la tendencia a seleccionar sus objetivos en puntos muy cercanos a los obstáculos, lo que dificulta el trabajo al módulo de navegación del robot y vuelve lento el proceso de exploración. Por otra parte, la dispersión excesiva de los robots rompe rápidamente la red de comunicación.

En la Figura 4.19 se muestran los trayectos resultantes usando mi propuesta. Se puede apreciar que los robots se dispersan en el ambiente, pero en algunos casos se ven obligados a regresar a zonas ya exploradas para lograr la cohesión del grupo de exploración. Es importante resaltar que el robot que decide regresar a una zona ya explorada, lo hace porque sus objetivos posibles actuales lo alejan del grupo de robots o porque no tiene objetivos que sean benéficos al estado actual de la exploración.

En aplicaciones en el mundo real no podemos asumir que no existen errores en la información de odometría a la que tiene acceso el robot. Estos errores afectan la manera en cómo se integran en el mapa local los datos observados con los sensores láser, obteniéndose

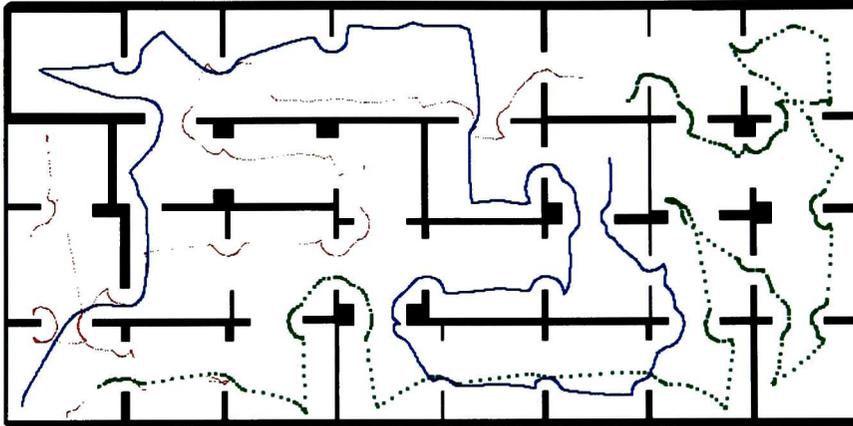


(a) Alcance de la comunicación = 20 metros

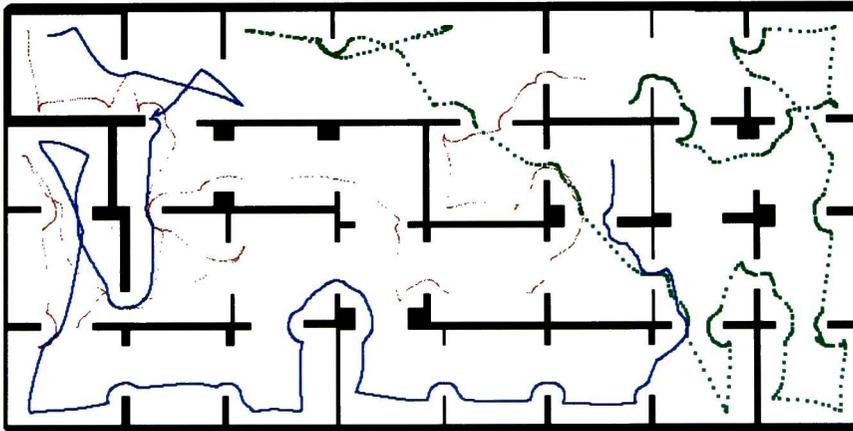


(b) Alcance de la comunicación = 30 metros

Figura 4.17: Exploración por un equipo de tres robot usando la propuesta de Yamauchi.

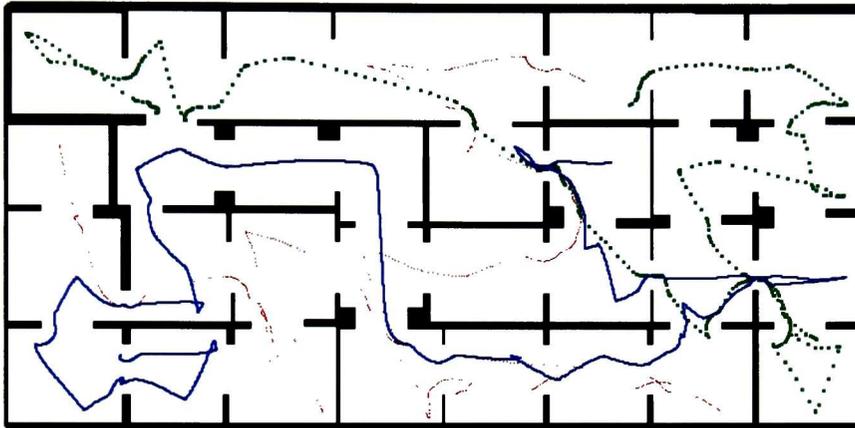


(a) Alcance de la comunicación = 20 metros

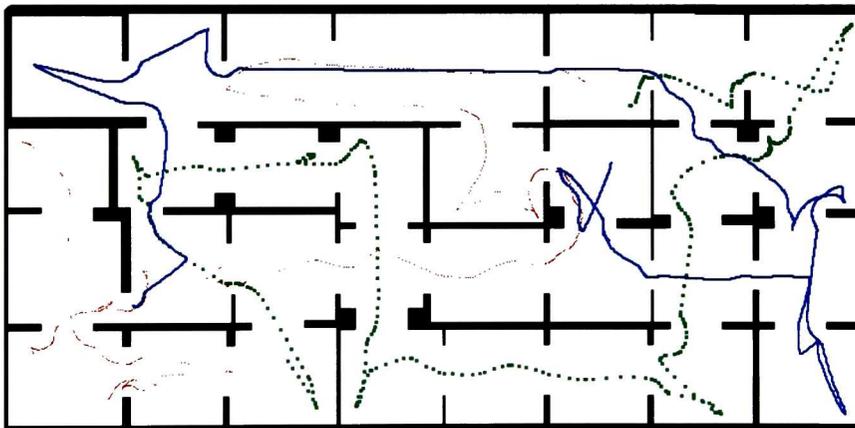


(b) Alcance de la comunicación = 30 metros

Figura 4.18: Exploración por un equipo de tres robot usando la propuesta de Stachniss.



(a) Alcance de la comunicación = 20 metros



(b) Alcance de la comunicación = 30 metros

Figura 4.19: Exploración por un equipo de tres robot usando mi propuesta.

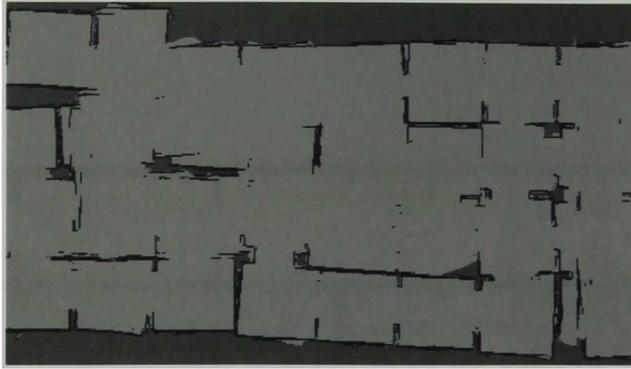


Figura 4.20: Mapa generado a partir la odometría con error error gaussiano.

mapas con un alto grado de distorsión, como puede apreciarse en la Figura 4.20. Una consecuencia directa es que la fusión de nueva información proveniente de los demás miembros del equipo con la información propia del robot se torna sumamente difícil.

Para comprobar el buen funcionamiento de mi propuesta ante incertidumbres en odometría, incluimos un error sobre el modelo diferencial de desplazamiento del robot, para los movimientos de rotación y de translación, definida como una función gaussiana con media igual a 0% y una desviación estándar de 5%.

Para corregir el error, decidimos usar el algoritmo "Map Referenced Iterative Closest Point" (MRICP) [27] ya que soluciona el problema *online* en cada uno de los robots, reduciendo la distorsión y logrando obtener buenos mapas individuales [23, 45] con lo cual el emparejamiento de los mapas se realiza de la siguiente manera, cuando un robot recibe un mapa ( $\omega$ ), cada punto  $(x, y)$  es promediado con el mapa actual ( $v$ ). El nuevo valor de ocupación  $v'_{xy}$  para el mapa se obtiene entonces:

$$v'_{xy} = \begin{cases} \frac{v_{xy} + \omega_{xy}}{2} & \text{si } v_{xy} \neq \text{explorado} \\ \omega_{xy} & \text{si } v_{xy} = \text{explorado} \end{cases} \quad (4.2)$$

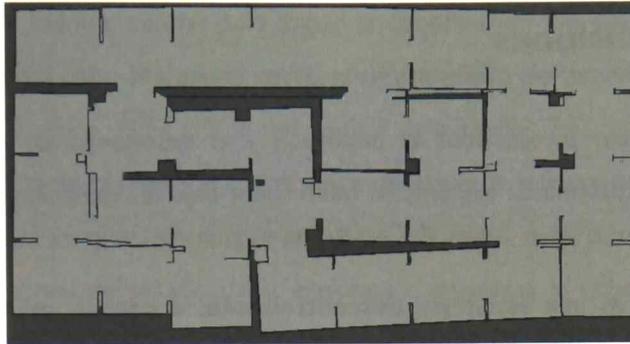


Figura 4.21: Mapa con error en odometría, construido utilizando el algoritmo MRICP

Al fusionar de esta forma la información del ambiente, la información de cada celda que tendrá mayor peso será la que provenga del robot que ha actualizado por mayor tiempo dicha celda, es decir la que ha corregido más su error en dicha posición.

En la Figura 4.21 se muestra el mapa resultante al incluir la corrección de error en cada robot para la prueba que se muestra en la Figura 4.19(b). Como podemos ver a diferencia del mapa generado sin corrección de error (Figura 4.20) en el cual los muros desaparecen a consecuencia de la mala localización, al realizar la corrección de error se mantiene una buena representación del ambiente.

De las Figuras 4.21 y 4.20 podemos ver que mi propuesta integra de manera eficiente la corrección de error ya que genera un buen mapa donde todos los muros están presentes, además, al continuar siendo descentralizada no sacrifica la tolerancia a fallos. Por otra parte, si no incluimos la corrección de error el mapa generado (Figura 4.20) no incluye algunos de los muros con lo cual la planificación de trayectorias se torna muy complicado en este mapa.

## 4.5 Conclusiones

En este capítulo presentamos el desarrollo y el desempeño de mi estrategia de solución para el problema de exploración multi-robot bajo la restricción en el rango de la comunicación.

Mi propuesta es una estrategia **descentralizada**, donde los robots operan como agentes económicos que buscan realizar el mejor aporte posible al proceso de exploración, sin depender de un módulo central para asegurar la coordinación del equipo. A diferencia de la mayoría de los métodos descentralizados propuestos en la literatura, el esquema de coordinación que proponemos es **asíncrono**, es decir, no requiere que los robots establezcan ventanas de tiempo para el intercambio de información, en las que tienen que esperar a los demás miembros del equipo para intercambiar información y establecer nuevos objetivos a explorar, por lo que tenemos una estrategia **sin tiempos muertos** durante la exploración. En mi propuesta, cuando un robot ha terminado su misión y requiere de un nuevo objetivo, analiza la situación actual de la exploración y decide, sin interferir con las misiones de los demás robots, la mejor acción a tomar.

Puesto que mi estrategia se basa en agentes independientes que buscan realizar el mejor aporte a una tarea en común, nos hemos inspirado de las heurísticas poblacionales empleadas en optimización multi-objetivo, para proponer una función de utilidad que permita al robot sacar ventaja de su posición actual (**explotar** su posición), que favorezca la dispersión de los robots (para **explorar** las zonas más alejadas de los demás miembros del equipo y evitar explorar zonas ya exploradas), pero que mantenga la **cohesión** del grupo y asegurar un intercambio eficiente de la información, pues de ello depende que el robot pueda analizar correctamente la situación actual de la exploración al momento de elegir un nuevo objetivo.

Esta estrategia de exploración ofrece además **robustez**, pues el proceso de exploración

no depende de un módulo central para lograr la coordinación del equipo de robots. Es posible que un robot, por una avería total, resulte excluido del proceso de exploración sin comprometer el éxito de la misión del equipo. Igualmente, es posible incluir nuevos miembros al proceso de exploración cuando este último ya se encuentra muy avanzado, y los nuevos robots se integran sin ningún problema a la tarea, pues contarán rápidamente con una versión actual del estado de la exploración gracias a la cohesión de la red de comunicación. Tenemos entonces una estrategia que es **flexible** y, al mismo tiempo, **tolerante a fallos** individuales de los robots.

De los experimentos en simulación, podemos concluir que mi propuesta es una alternativa eficiente a los esquemas actualmente desarrollados por la comunidad científica en robótica móvil, para resolver el problema de exploración de medios ambientes desconocidos, que es uno de los problemas más importantes en el área.

# 5

## Conclusiones y Trabajo Futuro

La generación de mapas es uno de los principales problemas de la robótica, ya que muchas aplicaciones requieren de una representación del espacio donde se desenvuelven. En esta tesis doctoral presentamos el desarrollo de una estrategia de coordinación descentralizado de un equipo de robots para la generación de mapas en el cual se toma en cuenta la restricción en el rango de comunicación.

La estrategia de exploración se ve beneficiada con el algoritmo novedoso para la planificación global de trayectorias desarrollado en esta tesis, ya que de él depende que se alcancen de forma eficiente los objetivos dados, además de que la distancia dada por el planificador hacia cada uno de los objetivos se toma como una medida de utilidad en mi algoritmo de exploración. El algoritmo de planificación global que desarrollamos tiene la ventaja de obtener la distancia exacta para cada una de las celdas del mapa, además de que nos proporciona el próximo mejor punto visible, todo esto sin necesidad de realizar un post procesamiento y minimizando la necesidad de verificar la línea de visión de cada celda al explotar el procesamiento realizado por las celdas vecinas. Otra ventaja es que

la trayectoria generada por mi propuesta no tiene restricciones en el ángulo al momento de seleccionar la siguiente celda a visitar durante la construcción del camino, es decir el robot puede dirigirse a cualquier celda sin limitarse a ángulos de  $\pi/4$ .

En esta tesis proponemos una estrategia para la exploración multi-robot totalmente descentralizada, en la cual cada uno de los robots operan como agentes económicos que buscan realizar el mejor aporte posible al proceso de exploración. Al no depender de un módulo central nos proporciona tolerancia a fallas completas de alguno de los miembros del equipo. Una ventaja importante de mi estrategia es que los robots no requieren el establecimiento de ventanas de tiempo para la coordinación (asíncrono), lo que evita la generación de tiempos muertos durante la exploración. En mi propuesta, cuando un robot ha terminado su misión y requiere de un nuevo objetivo, analiza la situación actual de la exploración y decide, sin interferir con las misiones de los demás robots, la mejor acción a tomar.

Mi propuesta contempla tres características basadas en las heurísticas de optimización multi-objetivo: **Explotación** con lo cual el robot se dirige a objetivos más cercanos a él (explotando su posición actual), **exploración** con lo cual el robot trate de separarse de los otros miembros del equipo (evitando el trabajo redundante y las interferencias entre éstos) y **cohesión** con lo cual los robots traten de evitar el romper con los puentes de comunicación, ya que de la comunicación depende que el robot pueda analizar correctamente la situación actual de la exploración al momento de elegir un nuevo objetivo.

De los experimentos en simulación, se puede concluir que mi propuesta es una alternativa eficiente a los esquemas actualmente desarrollados por la comunidad científica en robótica móvil, para resolver el problema de exploración de medios ambientes desconocidos, que es uno de los problemas más importantes en el área.

## 5.1 Trabajo Futuro

Debido a la naturaleza del problema abordado en esta tesis, existen distintos aspectos que podemos mejorar. Los aspectos principales son abordados a continuación.

### 5.1.1 Planificación de Trayectorias

Hemos desarrollado el algoritmo EEDT para ambientes bidimensionales donde cada celda puede contener solo dos tipos de estados (libre u ocupado). Una dirección interesante de investigación sería extender mis resultados a ambientes tridimensionales. En este caso el problema se vuelve más complejo ya que se incrementa considerablemente el número de celdas necesarias para representar el ambiente. Por otra parte, creemos que es posible extender esta propuesta a medios ambientes heterogéneos, de manera a incluir en el costo de distancia, el costo de traversabilidad del terreno, por ejemplo, ocupado, piso liso, arena, grava, concreto, etc.

### 5.1.2 Exploración Multi-robot

Una forma de mejorar la calidad del mapa generado sería el incluir en la estrategia de exploración un parámetro que favorezca la selección de objetivos cercanos a zonas que contengan la mayor cantidad de marcas, las cuales puedan ayudar a mejorar la localización del robot, es decir penalizar áreas abiertas y de objetivos donde la localización pueda verse afectada.

Otra mejora interesante es considerar el caso en el que los robots inician con marcos de referencia distintos, es decir, pueden comunicarse dentro de un rango de comunicación pero no conocen sus posiciones relativas. En este caso la exploración se vuelve mucho más difícil ya que además de integrar nueva información y mantenerse dentro del alcance de la comunicación de los otros robots, es necesario resolver la posición relativa entre ellos. Una

idea sería utilizar la intensidad de la señal en la comunicación para estimar una distancia entre los robots.

# Bibliografía

- [1] Susanne Albers and Monika R. Henzinger. Exploring Unknown Environments. *SIAM J. Comput.*, 29(4):1164–1188, February 2000.
- [2] Susanne Albers, Klaus Kursawe, and Sven Schuierer. Exploring unknown environments with obstacles. In *SODA '99: Proceedings of the tenth annual ACM-SIAM symposium on Discrete algorithms*, pages 842–843, Philadelphia, PA, USA, 1999. Society for Industrial and Applied Mathematics.
- [3] R. de Almeida and C. Melin. Exploration of Unknown Environments by a Mobile Robot. In *Intelligent Autonomous Systems 2, An International Conference*, pages 715–725, Amsterdam, The Netherlands, The Netherlands, 1989. IOS Press.
- [4] Juan Andrade-Cetto, T. Vidal-Calleja, and A. Sanfeliu. Multirobot C-SLAM: Simultaneous localization, control and mapping. In *In Proceedings of the IEEE ICRA Workshop on Network Robot Systems*, Barcelona, 2005.
- [5] Baruch Awerbuch, Margrit Betke, Ronald L. Rivest, and Mona Singh. Piecemeal graph exploration by a mobile robot (extended abstract). In *Proceedings of the eighth annual conference on Computational learning theory, COLT '95*, pages 321–328, New York, NY, USA, 1995. ACM.
- [6] M. Baglietto, M. Paolucci, L. Scardovi, and R. Zoppoli. Information-based multi-agent exploration. In *Robot Motion and Control, 2002. RoMoCo '02. Proceedings of the Third International Workshop on*, pages 173–179, 2002.

- [7] Rodrigo Benenson, Stéphane Petti, Thierry Fraichard, and Michel Parent. Towards urban driverless vehicles. *International Journal of Vehicle Autonomous Systems*, 1/2(6):4–23, 2008.
- [8] P.J. Besl and Neil D. McKay. A method for registration of 3-D shapes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 14(2):239–256, 1992.
- [9] A. Birk and S. Carpin. Merging Occupancy Grid Maps From Multiple Robots. *Proceedings of the IEEE*, 94(7):1384–1397, 2006.
- [10] S. Blackmore. New concepts in agricultural automation. In *HGCA conference—Stoneleigh Park, Kenilworth, Warwickshire, UK*, volume 28, 2009.
- [11] J.E. Bresenham. Ambiguities in Incremental Line Rastering. *Computer Graphics and Applications, IEEE*, 7(5):31–43, 1987.
- [12] Richard T. Vaughan Brian P. Gerkey and Andrew Howard. The Player/Stage Project: Tools for Multi-Robot and Distributed Sensor Systems. *Proc. of the Intl. Conf. on Advanced Robotics (ICAR)*, pages 317–323, July 2003.
- [13] R.A. Brooks. A robust layered control system for a mobile robot. *Robotics and Automation, IEEE Journal of*, 2(1):14–23, 1986.
- [14] Lloyd Arnold Brown. *The Story of Maps*. Dover Publications, INC. New York, 1980.
- [15] W. Burgard, M. Moors, D. Fox, R. Simmons, and S. Thrun. Collaborative multi-robot exploration. In *Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on*, volume 1, pages 476–481 vol.1, 2000.
- [16] W. Burgard, M. Moors, C. Stachniss, and F. Schneider. Coordinated multi-robot exploration. *IEEE Transactions on Robotics and Automation.*, 21:376–386, June 2005.

- [17] Wolfram Burgard and Frank Schneider. Collaborative exploration of unknown environments with teams of mobile robots. In *In dagstuhl*, pages 52–70. Springer Verlag, 2002.
- [18] Tom Bylander. The Computational Complexity of Propositional STRIPS Planning. *Artificial Intelligence*, 69(1–2):165–204, 1994.
- [19] Stefano Carpin. Fast and accurate map merging for multi-robot systems. *Autonomous Robots*, 25(3):305–316, 2008.
- [20] A. Chohra, A. Farah, and C. Benmehrez. Neural Navigation Approach for Intelligent Autonomous Vehicles (IAV) in Partially Structured Environments. *Applied Intelligence*, 8(3):219–233, 1998.
- [21] C.Y. Choo, J.M. Smith, and N.M. Nasrabadi. An efficient terrain acquisition algorithm for a mobile robot. In *Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on*, pages 306–311 vol.1, 1991.
- [22] Per-Erik Danielsson. Euclidean Distance Mapping. *Computer Graphics And image Processing*, 14:227–248, 1980.
- [23] Julian de Hoog. *Role-Based Multi-Robot Exploration*. PhD thesis, University of Oxford, 2011.
- [24] Julian de Hoog, Stephen Cameron, and Arnoud Visser. Role-Based Autonomous Multi-Robot Exploration. In *International Conference on Advanced Cognitive Technologies and Applications (COGNITIVE)*, November 2009.
- [25] Xiaotie Deng, Tiko Kameda, and Christos Papadimitriou. How To Learn An Unknown Environment I: The Rectilinear Case. *Journal of the ACM*, 45:215–245, 1997.

- [26] P. Dinnissen, S.N. Givigi, and H.M. Schwartz. Map merging of Multi-Robot SLAM using Reinforcement Learning. In *Systems, Man, and Cybernetics (SMC), 2012 IEEE International Conference on*, pages 53–60, 2012.
- [27] G. Dissanayake, J. Paxman, J.V. Miro, O. Thane, and H.-T. Thi. Robotics for Urban Search and Rescue. In *Industrial and Information Systems, First International Conference on*, pages 294–298, 2006.
- [28] A. Doniec, N. Bouraqadi, M. Defoort, Van Tuan Le, and S. Stinckwich. Distributed Constraint Reasoning Applied to Multi-robot Exploration. In *Tools with Artificial Intelligence, 2009. ICTAI '09. 21st International Conference on*, pages 159–166, 2009.
- [29] G. Dudek, M. Jenkin, E. Miliotis, and D. Wilkes. Robotic exploration as graph construction. *Robotics and Automation, IEEE Transactions on*, pages 859–865, 1991.
- [30] Juan C. Elizondo-Leal and Gabriel Ramirez-Torres. An Exact Euclidean Distance Transform for Universal Path Planning. *Electronics, Robotics and Automotive Mechanics Conference*, 0:62–67, 2010.
- [31] Juan C. Elizondo-Leal, Gabriel Ramírez-Torres, and Gregorio Toscano Pulido. Multi-robot Exploration and Mapping Using Self Biddings. In Hector Geffner, Rui Prada, Isabel Machado Alexandre, and Nuno David, editors, *IBERAMIA*, volume 5290 of *Lecture Notes in Computer Science*, pages 392–401. Springer, 2008.
- [32] Juan C. Elizondo-Leal, Gabriel Ramírez-Torres, and Gregorio Toscano Pulido. Multi-robot Exploration and Mapping Using Self Biddings and Stop Signals. In Alexander F. Gelbukh and Eduardo F. Morales, editors, *MICAI*, volume 5317 of *Lecture Notes in Computer Science*, pages 615–625. Springer, 2008.

- [33] D. Fox, J. Ko, K. Konolige, B. Limketkai, D. Schulz, and B. Stewart. Distributed Multirobot Exploration and Mapping. *Proceedings of the IEEE*, 94(7):1325–1339, 2006.
- [34] Yoav Freund and Robert E Schapire. A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.
- [35] Birgit Graf, Matthias Hans, and Rolf D. Schraft. Care-O-bot II: Development of a Next Generation Robotic Home Assistant. *Auton. Robots*, 16:193–205, March 2004.
- [36] J. Hancock and C. Thorpe. ELVIS: Eigenvectors for Land Vehicle Image System. *Intelligent Robots and Systems, IEEE/RSJ International Conference on*, 1:35, 1995.
- [37] Andrew Howard. Multi-robot Simultaneous Localization and Mapping using Particle Filters. *I. J. Robotic Res*, 25(12):1243–1256, 2006.
- [38] M. Jager and B. Nebel. Dynamic decentralized area partitioning for cooperating cleaning robots. In *Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on*, volume 4, pages 3577–3582 vol.4, 2002.
- [39] R. A. Jarvis. collision-free trajectory planning using distance transform. *Proc. National Conference and Exhibition on Robotics, Melbourne, 20-24th Aug. 1984*, also in *Mechanical Engineering Transactions, Journal of the Institution of Engineers*, ME10(3):187–191, 1985.
- [40] Ray Jarvis. Robot path planning: complexity, flexibility and application scope. In *PCAR '06: Proceedings of the 2006 international symposium on Practical cognitive agents and robots*, pages 3–14, New York, NY, USA, 2006. ACM.

- [41] Kyle Johns and Trevor Taylor. *Professional Microsoft Robotics Developer Studio*. Wrox Press Ltd., Birmingham, UK, UK, 2008.
- [42] J.L. Jones. Robots at the tipping point: the road to iRobot Roomba. *Robotics & Automation Magazine, IEEE*, 13(1):76–78, 2006.
- [43] Miguel Julia, Óscar Reinoso, Arturo Gil, Mónica Ballesta, and Luis Payá. A hybrid solution to the multi-robot integrated exploration problem. *Eng. Appl. of AI*, 23(4):473–486, 2010.
- [44] Miguel Julia, Oscar Reinoso, Arturo Gil, Monica Ballesta, and Luis Paya. BEHAVIOUR BASED MULTI-ROBOT INTEGRATED EXPLORATION. *INTERNATIONAL JOURNAL OF INNOVATIVE COMPUTING INFORMATION AND CONTROL*, 7(9):5225–5244, SEP 2011.
- [45] Adrian Jimenez-Gonzalez Stephen Cameron J. Ramiro Martinez de-Dios Julian de Hoog and Anibal Ollero. Using Mobile Relays in Multi-Robot Exploration. In *Proceedings of the Australasian Conference on Robotics and Automation*, December 2011.
- [46] J. Ko, B. Stewart, D. Fox, K. Konolige, and B. Limketkai. A practical, decision-theoretic approach to multi-robot mapping and exploration. In *Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, volume 4, pages 3232–3238 vol.3, 2003.
- [47] Sven Koenig, Craig Tovey, and William Halliburton. Greedy Mapping of Terrain. In *In Proceedings of the International Conference on Robotics and Automation*, pages 3594–3599. IEEE, 2001.

- [48] Kurt Konolige, Dieter Fox, Benson Limketkai, Jonathan Ko, and Benjamin Stewart. Map merging for distributed robot navigation. In *In Intl. Conf. on Intelligent Robots and Systems*, pages 212–217, Las Vegas, Nevada, 2003.
- [49] H. W. Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistic Quarterly*, 2:83–97, 1955.
- [50] J. C. Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, Boston, MA, 1991.
- [51] Steven M. Lavalle. Rapidly-Exploring Random Trees: A New Tool for Path Planning. Technical report, 1998.
- [52] David Lee. *The map-building and exploration strategies of a simple sonar-equipped mobile robot*. Cambridge University Press, New York, NY, USA, 1996.
- [53] Tomás Lozano-Pérez and Michael A. Wesley. An algorithm for planning collision-free paths among polyhedral obstacles. *Commun. ACM*, 22:560–570, October 1979.
- [54] Alexei A. Makarenko, Stefan B. Williams, Frederic Bourgault, and Hugh F Durrant-Whyte. An Experiment in Integrated Exploration. In *In Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 534–539, 2002.
- [55] E Masehian and D. Sedighzadeh. Classic and heuristic approaches in robot motion planning—a chronological review. In *World Academy of Science, Engineering and Technology*, volume 23, pages 101–106, 2007.
- [56] Jean-Arcady Meyer and David Filliat. Map-based navigation in mobile robots. II. A review of map-learning and path-planning strategies. *Cognitive Systems Research*, 4(4):283–317, 2003.

- [57] O. Michel. Webots: Professional Mobile Robot Simulation. *Journal of Advanced Robotics Systems*, 1(1):39–42, 2004.
- [58] Javier Minguez, Associate Member, and Luis Montano. Nearness Diagram (ND) Navigation: Collision Avoidance in Troublesome Scenarios. *IEEE Transactions on Robotics and Automation*, 20:2004, 2004.
- [59] Seungbin Moon and G.S. Virk. Survey on ISO standards for industrial and service robots. In *ICCAS-SICE, 2009*, pages 1878–1881, 2009.
- [60] P. Mukhija, K.M. Krishna, and V. Krishna. A two phase recursive tree propagation based multi-robotic exploration framework with fixed base station constraint. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 4806–4811, 2010.
- [61] R.R. Murphy. Human-robot interaction in rescue robotics. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 34(2):138–153, 2004.
- [62] Nils J. Nilsson. A Mobile Automaton: An Application of Artificial Intelligence Techniques. In Donald E. Walker and Lewis M. Norton, editors, *IJCAI*, pages 509–520. William Kaufmann, 1969.
- [63] Shimon Y. Nof. *Handbook of Industrial Robotics*. John Wiley & Sons, Inc., New York, NY, USA, 2nd edition, 1999.
- [64] Giuseppe Oriolo, Giovanni Ulivi, and Marilena Vendittelli. Real-time map building and navigation for autonomous robots in unknown environments. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 28(3):316–333, 1998.

- [65] Anshika Pal, Ritu Tiwari, and Anupam Shukla. Multi-Robot Exploration in Wireless Environments. *Cognitive Computation*, 4(4):526–542, 2012.
- [66] Ezra Federico Parra-González, Gabriel Ramírez-Torres, and Gregorio Toscano Pulido. Motion Planning for Cooperative Multi-robot Box-Pushing Problem. In Hector Geffner, Rui Prada, Isabel Machado Alexandre, and Nuno David, editors, *Advances in Artificial Intelligence - IBERAMIA 2008, 11th Ibero-American Conference on AI, Lisbon, Portugal, October 14-17, 2008. Proceedings*, volume 5290 of *Lecture Notes in Computer Science*, pages 382–391. Springer, 2008.
- [67] Luis Bergasa Pascual, Angel Leon, Rafael Barea, Elena Lopez, and Manuel Ocaña. SLAM and Map Merging. *Journal of Physical Agents*, 3(1), 2009.
- [68] A.K. Poernomo and Huang Shell Ying. New Cost Function for Multi-Robot Exploration. *Control, Automation, Robotics and Vision, 2006. ICARCV '06. 9th International Conference on*, pages 1–6, 5-8 Dec. 2006.
- [69] JONATHAN POTTER. *Country Life Book Of Antique Maps*. Country Life, 1989.
- [70] Saroj Kumar Pradhan, Dayal Ramakrushna Parhi, Anup Kumar Panda, and Rabindra Kumar Behera. Potential field method to navigate several mobile robots. *Applied Intelligence*, 25(3):321–333, December 2006.
- [71] Morgan Quigley, Ken Conley, Brian P. Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y. Ng. ROS: an open-source Robot Operating System. In *ICRA Workshop on Open Source Software*, 2009.
- [72] Martijn N. Rooker and Andreas Birk. Multi-robot exploration under the constraints of wireless networking. *Control Engineering Practice*, 15(4):435–445, April 2007.

- [73] A. Rosenfeld and J. Pfaltz. Sequential operations in digital picture processing. *Journal of the ACM*, 13(4):471–494, 1966.
- [74] Marcel Schoppers. Universal Plans for Reactive Robots in Unpredictable Environments. In *IJCAI*, pages 1039–1046, 1987.
- [75] Weihua Sheng, Qingyan Yang, Jindong Tan, and Ning Xi. Distributed multi-robot coordination in area exploration. *Robot. Auton. Syst.*, 54(12):945–955, 2006.
- [76] Reid G. Simmons, David Apfelbaum, Wolfram Burgard, Dieter Fox, Mark Moors, Sebastian Thrun, and Hakan Younes". "Coordination for Multi-Robot Exploration and Mapping". In *AAAI / IAAI* ",, pages "852–858",, "2000",.
- [77] C. Stachniss, O. Martínez-Mozos, and W. Burgard. Speeding-Up Multi-Robot Exploration by Considering Semantic Place Information. *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pages 1692–1697, 2006.
- [78] Cyrill Stachniss. Coordinated Multi-Robot Exploration. In *Robotic Mapping and Exploration*, volume 55 of *Springer Tracts in Advanced Robotics*, pages 43–71. Springer Berlin Heidelberg, 2009.
- [79] S. Thrun. Learning Metric-Topological Maps for Indoor Mobile Robot Navigation. *Artificial Intelligence*, 99(1):21–71, 1998.
- [80] Sebastian Thrun. Robotic Mapping: A Survey. In *Exploring Artificial Intelligence in the New Millenium*. Morgan Kaufmann, 2002.
- [81] Sebastian Thrun. Simultaneous mapping and localization with sparse extended information filters. In *Proceedings of the Fifth International Workshop on Algorithmic Foundations of Robotics (WAFR)*, Nice, France, 2002.

- [82] Sebastian Thrun and John J. Leonard. Simultaneous Localization and Mapping. In Bruno Siciliano and Oussama Khatib, editors, *Springer Handbook of Robotics*, pages 871–889. Springer, 2008.
- [83] Sebastian Thrun and Yufeng Liu. Multi-robot SLAM with Sparse Extended Information Filters. In Paolo Dario and Raja Chatila, editors, *ISRR*, volume 15 of *Springer Tracts in Advanced Robotics*, pages 254–266. Springer, 2003.
- [84] T. Tong, H. Yalou, Y. Jing, and S. Fengchi. Multi-robot cooperative map building in unknown environment considering estimation uncertainty. In *Control and Decision Conference, 2008. CCDC 2008. Chinese*, pages 2896–2901. IEEE, 2008.
- [85] Benjamín Tovar, Lourdes Muñoz-Gómez, Rafael Murrieta-Cid, Moises Alencastre-Miranda, Raul Monroy, and Seth Hutchinson. Planning exploration strategies for simultaneous localization and mapping. *Robotics and Autonomous Systems*, 54(4):314–331, 2006.
- [86] Jose Vazquez and Chris Malcolm. Distributed multirobot exploration maintaining a mobile network. In *In Proceedings of Second IEEE International Conference on Intelligent Systems*, pages 113–118, 2004.
- [87] Arnoud Visser, Julian de Hoog, Adrian Jimenez-Gonzalez, and Jose Ramiro Martinez de Dios. Discussion of Multi-Robot Exploration in Communication-Limited Environments. In *Workshop on "Towards Fully Decentralized Multi-Robot Systems: Hardware, Software and Integration"*, *IEEE International Conference on Robotics and Automation (ICRA)*., May 2013.
- [88] Z. Wang, S. Huang, and G. Dissanayake. D-SLAM: A decoupled solution to simultaneous localization and mapping. *The International Journal of Robotics Research*, 26(2):187–204, 2007.

- [89] Z. Wang, S. Huang, and G. Dissanayake. Multi-robot simultaneous localization and mapping using D-SLAM framework. In *Intelligent Sensors, Sensor Networks and Information, 2007. ISSNIP 2007. 3rd International Conference on*, pages 317–322. IEEE, 2007.
- [90] Stefan B. Williams, Gamini Dissanayake, and Hugh F. Durrant-Whyte. Towards Multi-Vehicle Simultaneous Localisation and Mapping. In *ICRA*, pages 2743–2748. IEEE, 2002.
- [91] Stefan B. Williams and Hugh Durrant-whyte. An efficient approach to the simultaneous localisation and mapping problem. In *In Proc. IEEE Int. Conf. Robotics and Automation*, pages 406–411, 2002.
- [92] Ling Wu, Domènec Puig Valls, and Miguel Angel Garcia. Balanced Multi-Robot Exploration through a Global Optimization Strategy. *Journal of Physical Agents*, 4(1), 2010.
- [93] Jing Xiao, Zbigniew Michalewicz, Lixin Zhang, and Krzysztof Trojanowski. Adaptive Evolutionary Planner/Navigator for Mobile Robots. *IEEE Transactions on Evolutionary Computation*, 1(1):18–28, April 1997.
- [94] B. Yamauchi. A frontier-based approach for autonomous exploration. In *CIRA '97: Proceedings of the 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation*, page 146, Washington, DC, USA, 1997. IEEE Computer Society.
- [95] Brian Yamauchi. Decentralized coordination for multirobot exploration. *Robotics and Autonomous Systems*, 29(2-3):111–118, 1999.
- [96] Bum-Jae You, Myung Hwangbo, Sung-On Lee, Sang-Rok Oh, Young Do Kwon, and San Lim. Development of a home service robot ISSAC. In *Intelligent Robots and*

- 
- Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, volume 3, pages 2630–2635 vol.3, oct. 2003.
- [97] A. Zelinsky. A mobile robot exploration algorithm. *Robotics and Automation, IEEE Transactions on*, 8(6):707–717, 1992.
- [98] R. Zlot, A. Stentz, M. Dias, and S. Thayer. Multi-Robot Exploration Controlled By A Market Economy. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3016–3023, 2002,.



# CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS DEL IPN

## UNIDAD TAMAULIPAS

Cd. Victoria, Tamaulipas, a 28 de junio de 2013.

Los abajo firmantes, integrantes del jurado para el examen de grado que sustentará el C. Juan Carlos Elizondo Leal, declaramos que hemos revisado la tesis titulada:

**“Estrategia descentralizada para la exploración multi-robot, incluyendo restricciones en rango de comunicación”**

Y consideramos que cumple con los requisitos para obtener el grado de Doctor en Ciencias en Computación.

Atentamente,

**Dr. Eduardo Arturo Rodríguez Tello**



---

**Dr. Wilfrido Gómez Flores**



---

**Dr. Guadalupe Acosta Villarreal**



---

**Dr. Himer Ávila George**



---

**Dr. José Gabriel Ramírez Torres**



---



CINVESTAV - IPN  
Biblioteca Central



SSIT0011723