
Trabajo Fin de Máster: Navegación de robots usando un
algoritmo de segmentación y el centro de áreas



Trabajo Fin de Máster

Javier Fernández Pérez

Trabajo de investigación para el

Máster en I.A. Avanzada: Fundamentos, Métodos y Aplicaciones

Universidad Nacional de Educación a Distancia

Dirigido por el

Prof. D. José Manuel Cuadra Troncoso

Junio 2016

Agradecimientos

En primer lugar me gustaría expresar mi agradecimiento a los profesores D. José Manuel Cuadra Troncoso y D. Félix de la Paz López por todo el tiempo, paciencia y dedicación que me han brindado, así como la gran cantidad de conocimientos y recursos que han puesto a mi alcance y que han hecho mucho más fácil y llevadero el desarrollo de este trabajo final de Máster.

También me gustaría dar las gracias a todas esas personas que han formado parte de mi educación y formación durante toda mi vida, desde los profesores que tuve en el colegio, siguiendo por los que tuve en mi etapa universitaria hasta convertirme en Ingeniero Informático, como finalmente los que he tenido en estos años en la UNED durante mis estudios de Máster. Todos ellos me han transmitido una gran cantidad de conocimiento que yo he ido asimilando durante todos estos años y que, directa o indirectamente, he tratado de usar y plasmar en este trabajo.

De igual forma, me gustaría agradecer a las dos empresas en las que he trabajado, Zed Worldwide y Hemasoft, su gran predisposición para que siguiera con mis estudios académicos, permitiéndome ir a las reuniones con mis directores aún estando en horario de trabajo y dándome todas las facilidades posibles para compaginar mi vida laboral con mi vida académica.

No podría finalizar los agradecimientos sin citar a mi familia: mi padre Enrique, que, sin duda, ha sido el que me ha inculcado la pasión que tengo por todo lo relacionado con la informática y, en menor medida, por la inteligencia artificial y la robótica; mi madre Pilar, que siempre me ha brindado todo su amor y apoyado para alcanzar mis metas por muy difíciles que pareciesen; y mi hermano Héctor, al que espero algún día de vuelta por España para poder seguir compartiendo momentos muy felices juntos y que es un pilar fundamental en mi vida. A ellos, a mi novia Elena y a mis amigos debo en gran parte lo que soy ahora y que pueda estar presentando este proyecto final de Máster.

Resumen

El objetivo de este proyecto es desarrollar un método que permita a un robot moverse por un entorno desconocido de forma segura, y a la vez explorar el mundo que le rodea lo máximo posible. El robot se mueve en un plano bidimensional y percibe distancias en un plano paralelo a éste. La idea básica para el desarrollo de este método de navegación explorativa se fundamenta en cuatro fases: detección de bifurcaciones por parte del robot; obtención de características descriptivas del entorno a través de sus sensores láser; toma de decisión de movimiento del robot basado en este modelado del entorno; y ejecución del movimiento del robot.

Para que el robot detecte las bifurcaciones que se encuentra según se va moviendo se ha usado una variación del método del centro de áreas, donde el robot seguía la trayectoria del centroide del área libre frontal percibida, ampliando el número de centros de áreas de uno a tres (central, derecho e izquierdo). Tanto el centro de áreas derecho como el izquierdo son los dos centros de áreas que informan al robot de si se está cerca de una bifurcación, ya que si uno de estos dos centros de áreas (o ambos) pasan de ser accesibles a inaccesibles es una señal para el robot de que se produce este hecho.

Una vez que el robot sabe que existe una bifurcación cerca a su derecha o izquierda se aplica una técnica de segmentación para refinar el modelo del entorno y se lleva a cabo una descomposición del mismo según las características de los segmentos en áreas libres, segmentos no conectados y segmentos convexos. Las áreas libres se asemejan a zonas del entorno sin obstáculos, los segmentos no conectados son zonas donde hay obstáculos y existe la posibilidad de que entre estos obstáculos existan zonas de escape para el robot, mientras que los segmentos convexos son zonas en los que los obstáculos están muy juntos unos de otros o simplemente son el mismo obstáculo (por ejemplo, dos lados de una pared).

La decisión de hacia dónde se va a mover el robot se basa en las áreas resultantes de la descomposición previa, ya que, para llevar a cabo una mayor exploración del mapa, el robot debe dirigirse a áreas libres o áreas formadas por segmentos no conectados y, por tanto, huir de áreas formadas por segmentos convexos.

Finalmente, una vez que el robot sabe hacia dónde moverse es necesario colocar unos puntos de división virtuales (también llamados split points) en los sitios adecuados para que el robot se dirija de una manera efectiva hacia el sitio fijado. Por supuesto, la colocación de éstos depende en gran medida de las características del entorno.

Abstract

The aim of this masters final project is to develop a method that allows a robot to move around an unfamiliar environment safely, and at the same time be able to explore the world around him as much as possible. The robot moves in a two dimensional plane and perceives distances in a parallel plane to it. The basic idea for the development of this explorative navigation method is based on four phases: detection of forks by the robot; obtaining descriptive characteristics of the environment through laser sensors of the robot; the decision making of robot motion based on this environment modeling; and the execution of the movement of the robot.

To be able to detect forks when the robot moves, a variation of the area centre method has been used, where the robot followed the centroid of perceived frontal free area, expanding the number of area centres from one to three (central, right and left). Right and left area centres inform the robot if a fork is near, because if one of these two area centres (or both) becomes inaccessible the robot will know this fact.

When the robot knows that there is a fork near to the right or left, a segmentation technique is applied to refine the environmental model and a decomposition based on the characteristics of the segments is carried out, detecting free areas, not connected segments and convex segments. Free areas simulate unobstructed environment areas, not connected segments are spaces where there are some obstacles and there is a possibility that between these obstacles an escape zone exists for the robot, while convex segments are areas where there obstacles which are very close to each other or they are the same obstacle simply (for example, two sides of a wall).

The decision of the robot movement is based on the results of the previous division, because the robot has to go towards free areas or areas formed by unconnected segments to achieve a good exploration of the map and, therefore, the robot has to avoid areas formed by convex segments.

Finally, when the robot knows where it has to move it is necessary to place virtual split points in the right places so that the robot directs towards the fixed place effectively. Of course, the placement of these split points depends on the characteristics of the environment to a large degree.

Índice general

1. Introducción	1
2. Estado del arte	3
2.1. Segmentación	4
2.1.1. Clústering	4
2.1.2. Extracción de líneas	5
2.2. Métodos de navegación	6
2.2.1. Campos de fuerza virtual	6
2.2.2. Histograma del campo vectorial	7
2.2.3. Ventana dinámica	8
2.2.4. Métodos de navegación híbridos	8
2.3. Centro de áreas básico	8
2.4. Segmentación en el centro de áreas básico	13
3. Método propuesto	17
3.1. Fase 1: centro de áreas múltiple	19
3.2. Fase 2: centro de áreas múltiple con segmentación	22
3.3. Fase 3: mejoras al centro de áreas múltiple con segmentación	27
4. Resultados experimentales	35
4.1. Descripción del robot	35
4.2. Descripción de los mapas	35
4.3. Descripción de los parámetros	36
4.4. Experimentos	37
5. Conclusiones y trabajos futuros	43
5.1. Principales aportaciones y conclusiones	43
5.2. Trabajos futuros	44
Bibliografía	45

Nomenclatura

DWA Dynamic Window Approach

EVFF Extended Virtual Force Field

GLS Generalized Least Squares

IEPF Iterative End Point Fit

RANSAC RANdom SAmple Consensus

RRT Rapidly-exploring Random Tree

SEF Successive Edge Following

VFF Virtual Force Field

VFH Vector Field Histogram

Índice de figuras

2.1. Evitación de obstáculos usando el centro de áreas	10
2.2. Alcance de objetivos usando el centro de áreas	11
2.3. Selección del punto de división erróneo	12
2.4. Selección del punto de división correcto	12
2.5. Problema de puntos de división estáticos	13
2.6. Puntos de división dinámicos	14
2.7. Segmentos en coordenadas polares	14
2.8. Proceso de segmentación	15
3.1. Problema del cruce simétrico	18
3.2. Robot con varios centros de áreas	19
3.3. Posibles situaciones del robot y sus centros de áreas	21
3.4. Segmentación del entorno	23
3.5. Áreas libres	23
3.6. Segmentos no conectados	24
3.7. Segmentos conectados	25
3.8. Áreas convexas	25
3.9. Área libre con dos puntos de división	26
3.10. Punto de división en segmentos no conectados	28
3.11. Puntos de división en áreas convexas	28
3.12. Primer problema de la percepción del área libre	29
3.13. Segundo problema de la percepción del área libre estrecha	29
3.14. Segundo problema de la percepción del área libre estrecha	30
3.15. Segundo problema de la percepción del área libre estrecha	30
3.16. Problema de colocación de los puntos de división en áreas libres	31
3.17. Solución al problema de colocación de los puntos de división en áreas libres	32
3.18. Problema de elección de áreas convexas	33
3.19. Solución a la elección de áreas convexas	33
4.1. Mapas de los experimentos de navegación con el centro de áreas múltiple	36
4.2. Recorridos del robot en el mapa 1	40

4.3. Recorridos del robot en el mapa 2	41
4.4. Recorridos del robot en el mapa 3	41
4.5. Recorridos del robot en el mapa 4	42

Índice de cuadros

4.1. Resultados del centro de áreas básico	38
4.2. Resultados del centro de áreas múltiple con sector central de 1 grado	38
4.3. Resultados del centro de áreas múltiple con sector central de 61 grados	38
4.4. Resultados del centro de áreas múltiple con sector central de 121 grados	39

Capítulo 1

Introducción

Este trabajo tiene como objetivo la ampliación de métodos de navegación reactiva para robots autónomos tomando como base algoritmos basados en el centro de áreas. Cuando se habla de navegación no se debe caer en la tentación de pensar que lo que se pretende es única y exclusivamente que el robot se mueva por un entorno desconocido sin chocar contra los obstáculos con los que se va encontrando en su camino, sino que la navegación sea una navegación explorativa, es decir, sea capaz de descubrir todos los lugares y recovecos del mapa que le rodea. En definitiva, el robot debe ser capaz de desplazarse por un entorno desconocido y reconocer cuál es el movimiento más adecuado que debe realizar para ir a otra zona del mapa de la que no se conocen sus características. El tipo de robot que consideraremos en este trabajo es el robot terrestre equipado con sensores de rango 2D, que proporcionan al robot mediciones de las distancias a las que se encuentran los objetos que le rodean.

Con este objetivo en mente, se ha realizado una adaptación del algoritmo del centro de áreas para que el robot recopile la mayor información posible del entorno y así dar una mayor capacidad de exploración, pudiendo tomar diferentes decisiones en cuanto a su movimiento.

Un ser humano es capaz de examinar un sitio desconocido con muchas bifurcaciones o pasillos cuando puede moverse por el mismo y, además intuir cuál es la bifurcación que debe seguir para descubrir nuevos lugares. Supongamos que una persona llega a un centro comercial por primera vez y se encuentra que existen muchos pasillos por los que ir y que, además, estos pasillos confluyen unos con otros. Lo más conveniente es que si esta persona quiere explorar el centro comercial elija aleatoriamente uno de estos pasillos, empiece a andar por sí mismo y cuando vea que llega a una bifurcación, tome una decisión sobre qué pasillo seguir a partir de ese momento. Para tomar esta decisión las personas nos solemos guiar por lo que vemos más allá del cruce, es decir, si en lo que llegamos a ver de los pasillos nos da una idea de que ese pasillo sigue su trayectoria por otra zona del centro comercial o, por el contrario, ese pasillo termina unos metros más allá porque es un pasillo sin salida ninguna.

Si este problema lo trasladamos al mundo de la robótica la cosa se complica por diferentes motivos. En primer lugar la percepción que tiene un robot del entorno no es la misma que la de

un ser humano, ya sea por su limitación natural dada por los sensores que tiene, o por los errores en los datos captados por estos sensores (ruido). Otro problema que se plantea en esta nueva situación es cómo modelar los datos que el robot recibe por los sensores en algo que él pueda entender y tratar para lograr sacar las principales características identificativas del entorno, y que posteriormente le servirán para localizarse y saber donde está. Además, el desplazamiento propio del robot tiene ya de por sí un error dado por el tipo de suelo por el que se desplaza el robot o el desgaste de las ruedas, por lo que es necesario una continua relocalización del robot.

En los sucesivos capítulos de la memoria se detallan los estudios previos realizados en cuanto a la navegación de robots autónomos, conocido en el ámbito de la investigación como el estado del arte; la propuesta desarrollada en este trabajo, desglosada en las diferentes fases de las que se ha compuesto; los resultados experimentales que se han llevado a cabo, así como un análisis comparativo con resultados de otros investigadores; y finalmente se recopila las diferentes conclusiones y aportaciones extraídas del trabajo realizado, y se proponen algunas líneas de trabajo futuro.

Capítulo 2

Estado del arte

Se define navegación como el proceso de guiado de un robot móvil a través de un entorno con obstáculos. El principal objetivo en la navegación robótica consiste en que el robot se mueva de forma segura por el entorno, esquivando los obstáculos que se encuentra a su paso y reaccionando ante situaciones inesperadas que hacen peligrar su seguridad.

Según (Martínez, 1997), el problema de la navegación en robots se puede descomponer en tres tareas:

1. Percepción del entorno a través de los sensores y creación de una abstracción (modelo) del entorno.
2. Planificación de una trayectoria libre de obstáculos.
3. Guiado del robot basándose en la trayectoria planificada.

En cuanto a la percepción del entorno por parte de un robot móvil para que pueda moverse por el mismo, existen multitud de sensores que son capaces de medir estímulos muy variados: color, distancia, fuerza, presión, inclinación, etc.

Estos sensores se pueden clasificar de muchas maneras, pero una de las más comunes es en sensores propioceptivos y exteroceptivos.

Dentro de los sensores propioceptivos se encuentran todos los sensores que captan el estado interno del robot (medidas de carga de baterías, posición del robot, etc.), como son los encoders, los acelerómetros, los giroscopios y la brújula y el inclinómetro.

Los sensores exteroceptivos obtienen información del entorno del robot, como la distancia a objetos. Las medidas de este tipo de sensores normalmente son interpretadas por el robot para extraer características del entorno y construir un modelo del mismo. Los sensores exteroceptivos más usados en la navegación de robots son: los sensores de choque, los de visión, el sónar y los sensores láser.

2.1. Segmentación

Una vez que se tienen los datos provenientes de los sensores es necesario simplificar los modelos generados, ya que obtener información de los mismos es más difícil según el entorno es más complejo. Por ello no es de extrañar que una de las técnicas más utilizadas en robótica para modelar la información proveniente de los sensores del robot sea la segmentación, ya que es capaz de modelar de forma bastante sencilla la información proveniente de los sensores de los robots sin, por ello, perder información relevante.

La segmentación es el proceso que divide un conjunto de mediciones en segmentos, y se basa en tres aspectos fundamentales según (Forsyth y Ponce, 2002): el número de líneas existentes, la pertenencia de un determinado punto a un segmento y la estimación de los parámetros de las líneas. El número de líneas existentes y la pertenencia de un determinado punto a un segmento es, en resumidas cuentas, detectar los extremos de todos los segmentos.

2.1.1. Clústering

Si la búsqueda de los extremos de los segmentos se hace a partir de medidas de rango se le denomina clústering. En general, esta técnica usa un valor umbral para decidir si un punto pertenece al segmento formado por los puntos anteriores del haz.

El clústering no adaptativo es aquél en el que el valor umbral es constante durante todo el proceso y, por lo tanto, es crucial el valor umbral que se escoja. Ejemplos de ello se pueden encontrar en (Mastrogiovanni, Sgorbissa, y Zaccaria, 2005) para la construcción dinámica de mapas de interiores; (Nguyen et al., 2005) para comparar el rendimiento de varios algoritmos de extracción de líneas y (Keat, Pradalier, y Laugier, 2005) para la localización de vehículos y construcción de mapas de un aparcamiento.

En contraposición al clústering no adaptativo, en el clústering adaptativo se calcula un valor umbral diferente en cada iteración, tomando la decisión de si un punto pertenece al segmento o no dependiendo de la distancia al punto anterior. Si esta distancia no supera el umbral formará parte de este segmento; si es así, forma parte de un nuevo segmento. El ejemplo más sencillo es el encontrado en (Xu et al., 2002) porque utiliza un valor umbral variable mediante una función lineal. Otros estudios que aplican esta técnica son (Dietmayer, Sparbert, y Streller, 2001; Santos et al., 2003; Borges y Aldon, 2004); variando en cada uno de ellos la forma de calcular este umbral.

Otra técnica de clústering es el scale-space, que se basa en encontrar la escala adecuada para reconstruir el mundo como un conjunto de segmentos. La primera referencia a los operadores multiescala se encuentra en (Witkin, 1983), en el que se dice que la convolución Gaussiana es el único operador que satisface los principios de la simetría general y la invarianza de escala. Posteriormente se ha desarrollado la teoría matemática del scale-space en (Lindeberg, 1990; Lindeberg, 1993; Lindeberg, 1994), donde en lugar de especificar un operador de scale-space en el dominio continuo para luego discretizarlo, se discretiza el scale-space requerido. En (Weickert,

Ishikawa, y Imiya, 1997) se muestra que el concepto de scale-space lineal en un sentido es sobredeterminado; mientras que en (Pauwels et al., 1995) se encuentra una buena cantidad de operadores lineales multiescala.

2.1.2. Extracción de líneas

La extracción de líneas estima la ecuación del segmento a medida que se incorporan nuevos puntos a éste. La gran mayoría de estos algoritmos usan la distancia del punto al segmento para decidir si va a formar parte del segmento. Existen dos tipos de extracción de líneas: robustas y no robustas.

Las técnicas de extracción de líneas no robustas usan estimadores que, si tratan con algún dato que se encuentra distante numéricamente con respecto al resto de observaciones, proporcionan una estimación poco o nada acertada, como son las basadas en el seguimiento de extremos consecutivos, line tracking, iterative end point fit o split and merge.

En la técnica de extracción de líneas no robustas basada en el seguimiento de extremos consecutivos (en inglés Successive Edge Following, SEF), se considera que un segmento termina cuando la diferencia entre una distancia y su siguiente excede un determinado valor umbral. Posteriormente hay que realizar un postprocesado para ajustar una línea a cada conjunto mediante un método de regresión (Borges y Aldon, 2000; Siadat et al., 1997).

En line tracking se estima la recta que une los dos primeros puntos, añadiendo cada uno de los siguientes puntos si se cumple cierto criterio; si no, se comienza un nuevo segmento (Nguyen et al., 2005; Martinez-Cantin et al., 2006).

En la técnica llamada Iterative End Point Fit (IEPF) primero se halla la línea que une los puntos inicial y final para posteriormente buscar el punto más alejado con respecto a esa línea. Si este punto está a mayor distancia que un determinado valor umbral, se divide en dos segmentos; y así recursivamente. El proceso acaba cuando la distancia al punto más alejado sea menor que el valor umbral para todos y cada uno de los segmentos (Borges y Aldon, 2004; Siadat et al., 1997).

Finalmente, dentro de las técnicas de extracción no robusta, está la técnica split and merge que se compone de dos pasos. En el primero se estima un segmento con todos los puntos y se calcula qué punto queda a mayor distancia. Si esta distancia es mayor a un determinado umbral, se divide el segmento en dos partes por este punto. Dicho proceso se repite hasta que sea imposible dividir el segmento. En el segundo paso se unen los segmentos adyacentes que cumplan un criterio de colinealidad (Keat, Pradalier, y Laugier, 2005; Mastrogiovanni, Sgorbissa, y Zaccaria, 2005; Sack y Burgard, 2004).

Las técnicas de extracción de líneas robustas, a diferencia de las anteriores, tratan con estimadores que proporcionan una estimación adecuada independientemente del valor de los datos con los que trabaje. Algunas de estas técnicas son: la mínima mediana de cuadrados, random sample consensus o la transformada de Hough.

La solución de mínima mediana de cuadrados se basa en minimizar las desviaciones con

respecto a la mediana. Para ello es necesario ajustar los datos y, luego, identificar los datos anómalos. Uno de los principales problemas de esta técnica es que es muy costoso computacionalmente, por lo que muchas veces se opta por dar soluciones aproximadas (Rousseeuw y Leroy, 1987; Olson, 1997).

En Random Sample Consensus (RANSAC) de forma aleatoria se escogen dos puntos y se halla la línea que pasa por ellos. Después se separan los demás puntos en dos clases según su distancia a la recta. Si existen muchos puntos cercanos a la recta, se fija otro umbral de distancia y se recalcula el segmento. Este proceso continúa hasta un número máximo de iteraciones o que queden pocos puntos cercanos a la recta (Fischler y Bolles, 1981).

La transformada de Hough es una técnica en la que se usa una malla que discretiza los valores para el radio y ángulo de las mediciones. Una celda de la malla se incrementa si una medición verifica la ecuación de la recta para los valores de los parámetros correspondientes a la celda. Las celdas con mayor número de votos en la malla definen los valores de los parámetros de los segmentos (Duda y Hart, 1972; Norouzi et al., 2009).

2.2. Métodos de navegación

Cubierto el tema de cómo los robots perciben el entorno y lo modelan, toca hablar de los métodos de navegación de robots móviles más extendidos: el campo de fuerza virtual, el histograma del campo vectorial y la ventana dinámica. También se va a comentar en detalle otro método de navegación, el centro de áreas básico, pero al ser una parte fundamental del desarrollo de este trabajo se ha tomado la decisión de explicarlo, por separado, en otra sección aparte.

2.2.1. Campos de fuerza virtual

La técnica de navegación conocida como campos de fuerza virtual (VFF) o campos de potencial consiste en que el robot está sometido a un campo potencial, donde el punto de destino del robot ejerce una fuerza de atracción sobre el robot y los obstáculos generan una fuerza de repulsión sobre el robot, por lo que su velocidad y dirección vendrá determinada por la fuerza resultante de sumar sus fuerzas de atracción y repulsión. Dicha técnica fue usada por primera vez por (Hogan y Andrews, 1983) y más tarde por (Khatib, 1985). Otra propuesta con este mismo enfoque, fue desarrollada por (Krogh, 1984).

A pesar de su sencillez, este método tenía una serie de importantes inconvenientes como pusieron de manifiesto (Koren y Borenstein, 1991), destacando entre estos inconvenientes la trampa del mínimo local producido por un obstáculo con forma cóncava entre el robot y el objetivo, ocasionando una continua navegación del robot dentro de esta trampa. (Moravec y Elfes, 1985) desarrollaron el concepto de mallas de certeza que representa un mapa de obstáculos donde cada rejilla tiene un valor que indica la probabilidad de que exista un obstáculo o no en

esa localización, a fin de superar las inexactitudes que provienen de los datos sensoriales. Sin embargo, una limitación de esta técnica es que se asume que el entorno es estático.

(Khosla y Volpe, 1988) propusieron las funciones potenciales supercuadráticas, donde se emplea un amplio conjunto de formas descrito por fórmulas supercuadráticas. (Kim y Khosla, 1991) y (Connolly, Burns, y Weiss, 1990) hacen uso de las funciones potenciales armónicas para solventar el problema de los mínimos locales. En el caso de tener obstáculos en movimiento (Ge y Cui, 2000) proponen un nuevo conjunto de funciones potenciales que dependen de la posición y de la velocidad de los obstáculos.

(Chengqing et al., 2000) fue pionero en el concepto de obstáculo virtual como una estrategia para solventar el problema de los mínimos locales en la navegación basada en campos de potencial. (Im y young Oh, 2000) desarrollaron la idea de campos de fuerza virtuales extendidos (EVFF) mediante la integración de teoría de redes neuronales y programación evolutiva. (Zou y Zhu, 2003) fueron los primeros investigadores en introducir la idea de tener un destino virtual para solucionar el problema de los mínimos locales. (Nourani-vatani et al., 2006) presentó una plataforma robótica donde combinaba localización basada en sensores láser y mapeado usando el framework Atlas con árboles aleatorios de exploración rápida (RRT), planificación de caminos y esquivación de obstáculos mediante campos de fuerza virtual.

En los campos potenciales globales, donde se considera la totalidad de los obstáculos como también el objetivo para construir el campo potencial como un todo y no como la adición de las partes, se busca eliminar el problema de los mínimos locales. Bajo este nuevo enfoque aparecen las funciones de navegación propuestas por (Rimon y Koditschek, 1992), que son explícitamente diseñadas para presentar un único mínimo local en el objetivo y, si existe otro tipo de puntos de equilibrio, se enmascara (rellena).

2.2.2. Histograma del campo vectorial

A fin de superar las deficiencias que tenía el método del campo de fuerzas virtual, (Borenstein y Koren, 1991) desarrollaron el histograma del campo vectorial (VFH), un método que busca sectores con baja densidad de obstáculos en histogramas polares construidos localmente. Con este nuevo método era mucho menos probable el quedar atrapado en mínimos locales y permitía al robot moverse más rápido.

Este método fue mejorado y renombrado como VFH+ (Ulrich y Borenstein, 1998) gracias a la simplificación en cuanto a la capacidad de movimiento del robot, permitiéndole únicamente movimientos en rectas o arcos e incluyendo la norma de que los obstáculos que bloquearan una dirección dada también bloqueaban todas las posibles trayectorias a través de esa dirección.

Otra variante a este método puede ser encontrado en (Ulrich y Borenstein, 2000) donde se verifica que una dirección candidata guía al robot alrededor de un obstáculo usando el algoritmo de búsqueda A* y funciones de coste y heurística apropiadas.

2.2.3. Ventana dinámica

A diferencia de otros métodos, el método de la ventana dinámica (DWA) introducido por (Fox, Burgard, y Thrun, 1997) se deriva directamente de la dinámica del robot, y está diseñado para hacer frente a las restricciones impuestas por las velocidades y aceleraciones limitadas del robot. Este método se compone de dos componentes: el primero genera un espacio de búsqueda válido compuesto por las trayectorias circulares seguras que se pueden alcanzar en un intervalo corto de tiempo y están libres de colisiones con obstáculos; y el segundo selecciona una solución óptima en este espacio de búsqueda (maximizar el espacio libre desde cualquier obstáculo).

Este método se ha integrado con un planificador de movimientos por (Thrun et al., 1998) y ampliado para usar un mapa en conjunción con información sensorial y así generar movimientos libres de colisiones en (Fox et al., 1998).

En (Brock y Khatib, 1999) se propone un método de ventana dinámica global como una generalización del método de ventana dinámica, combinando métodos de planificación de movimientos y evitación de obstáculos en tiempo real. También en (Ogren y Leonard, 2005) se usa el método de ventana dinámica como modelo de problema de control predictivo.

2.2.4. Métodos de navegación híbridos

Los métodos de navegación explicados anteriormente se pueden combinar con diferentes técnicas de aprendizaje para mejorar su rendimiento, como por ejemplo: lógica difusa ((Pradhan, Parhi, y Panda, 2009; Raudonis y Maskeliunas, 2011; Abdessemed, Benmahammed, y Monacelli, 2004)), enjambre de partículas ((Doctor, Venayagamoorthy, y Gudise, 2004; Smith, Venayagamoorthy, y Holloway, 2006; Masehian y Sedighizadeh, 2010)), algoritmos genéticos ((Yang et al., 2007; Yun, Parasuraman, y Ganapathy, 2011; Tuncer y Yildirim, 2012)) o redes neuronales ((Chohra, Farah, y Benmehrez, 1998; Silva, Crisostomo, y Ribeiro, 2000; Billard y Hayes, 1999)).

2.3. Centro de áreas básico

El método de navegación denominado método del centro de áreas básico (Cuadra Troncoso, 2011) es la base de este proyecto, por lo que su conocimiento y comprensión es fundamental para poder entender la investigación aquí realizada, y por ello se ha separado en una sección aparte. Se han cogido algunas figuras de este trabajo para poder explicar mejor los conceptos básicos de tal investigación.

La fuente de inspiración del algoritmo de centro de áreas básico para la esquivación de obstáculos en un entorno desconocido es la siguiente: los humanos modelamos el área libre que nos rodea, no el área ocupada (obstáculos), cuando navegamos reactivamente, y usamos el centro de áreas de dicha área libre como una referencia de movimiento.

Supongamos que una persona conduce un coche y se encuentra con un camión grande que

viene en dirección opuesta. El conductor de dicho coche lo que hace instintivamente es buscar y seguir el centro de áreas del espacio libre circundante. Áreas más concurridas harán bajar la velocidad al conductor y áreas más despejadas permitirán incrementar la velocidad.

Si el área percibida es convexa, volverse hacia la posición del centro de áreas y avanzar en línea recta es la opción de navegación más segura. Un área es convexa cuando la línea que une cualquier par de puntos de dicha área está contenida en ella, por esto las áreas convexas no contienen obstáculos dentro y son seguras. En cambio, cuando el área no es convexa el centro de áreas puede estar fuera de ella y, por lo tanto, ser inaccesible. Básicamente, la solución propuesta consiste en dividir el área percibida en dos, con la esperanza de que alguna de las dos partes tenga un centro de áreas accesible hacia el que encaminarse.

El proceso para encontrar centros de áreas accesibles para una navegación segura se describe a continuación y puede verse visualizado en la figura 2.1, denominado a partir de este momento como sector de avance completo a la zona completa del área percibida frontal, y como sector de avance restringido al subsector seleccionado tras una operación de división:

1. El robot comienza a seguir el centro de áreas.
2. Mientras que el centro de áreas sea accesible, el robot lo sigue.
3. Cuando el centro de áreas se vuelve inaccesible, se determinan los dos puntos del polígono que definen el sector angular que contiene a la recta que une el origen con el centro de áreas. De entre los dos se elige el punto más cercano al robot como punto de división (split point) y se divide el polígono en dos. Luego los polígonos resultantes son comprimidos. Se calculan sus centros de áreas y se determina su accesibilidad:
 - a) Si los dos centros de los dos sectores son accesibles se elige uno de ellos, aleatoriamente o mediante una preferencia externa.
 - b) Si sólo uno de los centros es accesible se elige su sector.
 - c) Si ambos centros son inaccesibles, no se puede continuar el camino. Entonces se determina el centro de áreas de la parte libre percibida posterior y se gira hacia él; es una maniobra de escape.

Durante el giro, vértices del polígono descartado pasan al lado del punto de división donde se encuentra el sector de avance restringido. Estos vértices se añaden a dicho sector y son eliminados del descartado.

Este proceso incrementa la amplitud del sector de avance restringido, por lo que se incrementa el alcance máximo considerado proporcionalmente; así cuando la amplitud del sector restringido alcance a la del completo, el rango máximo considerado habrá alcanzado su valor inicial.

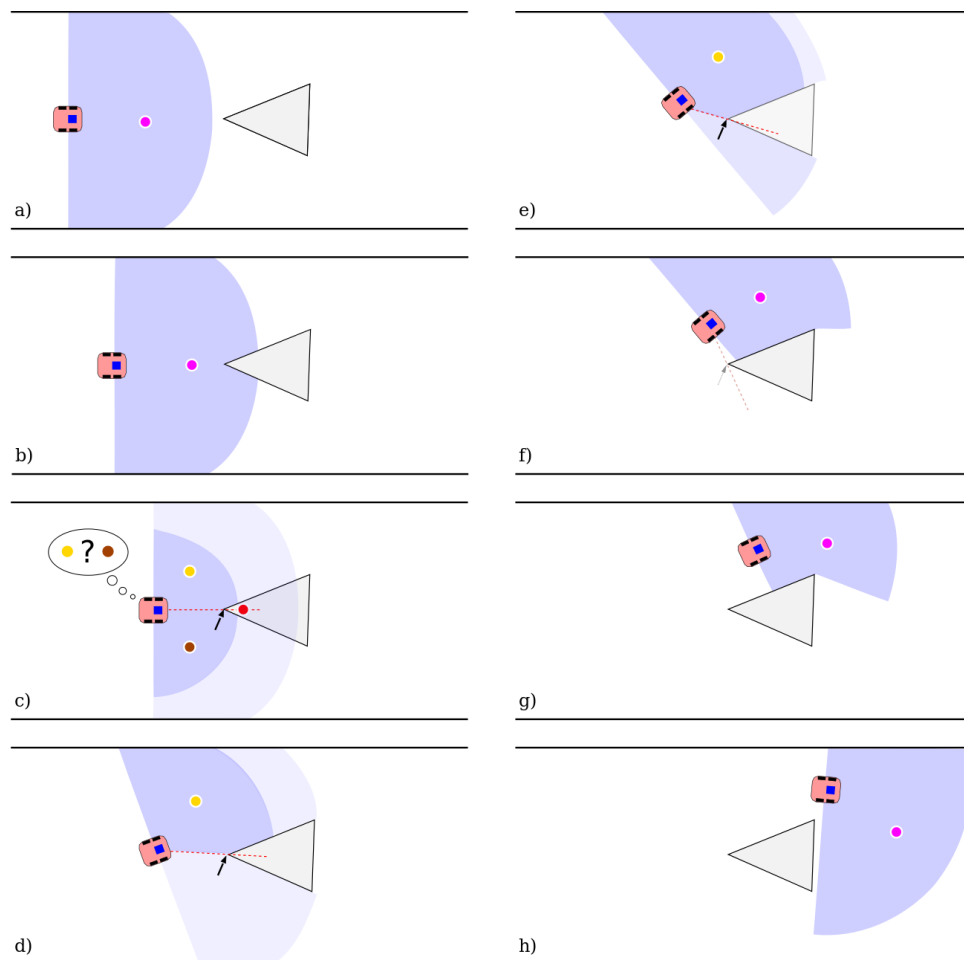


Figura 2.1: Estas imágenes forman una secuencia mostrando cómo el robot supera un obstáculo usando el método del centro de áreas básico. Las áreas sombreadas, oscuras o claras, representan el sector de avance completo. El área oscura es el sector de avance actual, las claras son sectores descartados. Los círculos pequeños son los centros de áreas; el color amarillo indica la selección del subsector izquierdo y el marrón la del derecho, el violeta indica que se usa el sector de avance completo y el rojo que el centro de áreas es inaccesible. El punto de división se indica con una flecha.

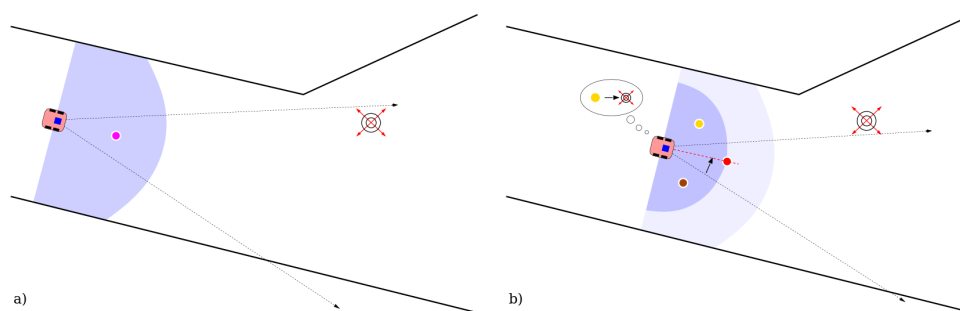


Figura 2.2: Los nuevos símbolos de estas figuras son: el sector angular frontal, delimitado por flechas largas y punteadas; y el objetivo, los dos círculos concéntricos con un aspa de flechas en su interior. Mientras que el objetivo se encuentra dentro del sector angular frontal el robot sigue el centro de áreas (figura a), pero cuando sale de dicho sector angular se considera que existe un obstáculo virtual sobre el centro de áreas y se aplica el proceso de división, eligiendo el sector del lado orientado al objetivo (figura b).

4. Cuando el movimiento del robot hace que todos los vértices hayan pasado el punto de división, el polígono descartado es eliminado. Ahora, siguiendo el centro de áreas el robot puede sobrepasar el obstáculo.

Para conseguir que el robot alcance un determinado objetivo sólo es necesario realizar unas cuantas modificaciones al método del centro de áreas. La primera modificación es que si se produce un proceso de división, debido a un obstáculo real, la dirección del objetivo respecto a la situación del robot indica cuál es la preferencia por un determinado lado; y la segunda es que si la dirección del objetivo se desvía respecto a la del robot, determinado esto por un sector angular frontal cuya amplitud es inversamente proporcional a la distancia a la que se encuentre el robot de su objetivo, se sitúan obstáculos virtuales en la posición del centro de áreas causando un proceso de división con una preferencia de lado definida (ver figura 2.2).

La selección del punto de división cuando el centro de áreas se vuelve inaccesible tiene un problema que se visualiza sobre todo cuando el robot persigue un objetivo, y es que, en ocasiones, la posición del centro de áreas no tiene por qué ser la del punto más intrusivo de su borde, pudiéndose colocar el punto de división en un sitio que no es el óptimo e incrementando las posibilidades de choque del robot. Esta situación se puede observar en la figura 2.3.

La solución dada a este problema usa la segmentación de las medidas de rango para disponer de un conjunto poco numeroso de segmentos bien definidos. Se seleccionan los extremos izquierdos de los segmentos desde el que contiene al centro de áreas hacia la derecha y todos los extremos derechos de los segmentos del centro de áreas hacia la izquierda. Para acabar de los extremos seleccionados se elige el más próximo al robot como punto de división (ver figura 2.4).

Otro problema vislumbrado en este método, que se acrecienta cuando el robot persigue un objetivo, es que, aunque la posición inicial del punto de división esté en el pico más peligroso, el posterior movimiento del robot puede hacer que sea otro pico el que entrañe más peligro, pero,

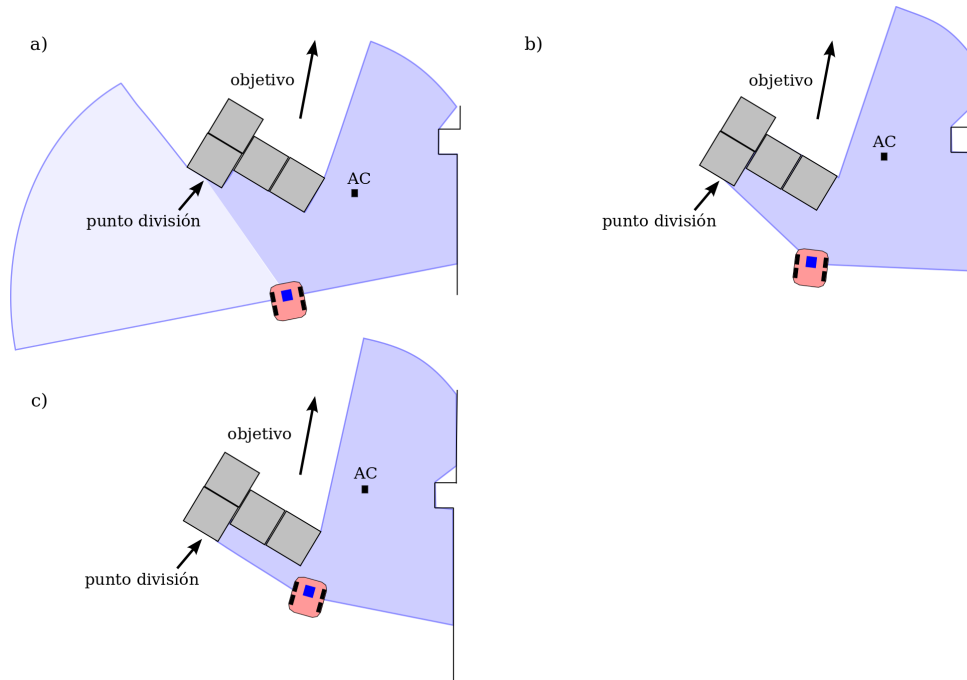


Figura 2.3: Como la situación del centro de áreas no es la ideal cuando se vuelve inaccesible, se coloca el punto de división en un sitio que produce que uno de los dos subsectores tenga un pico acusado, pudiendo producir una situación peligrosa si se escoge este subsector.

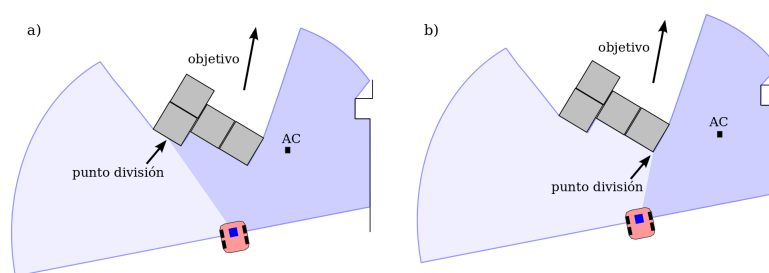


Figura 2.4: Tras realizar el proceso de segmentación se escoge el extremo derecho que se encuentra a la izquierda del robot como sitio para colocar el punto de división, ya que es el más cercano al robot, sitúandolo en el pico más saliente de la trayectoria del robot y conduciéndolo por una vía segura.

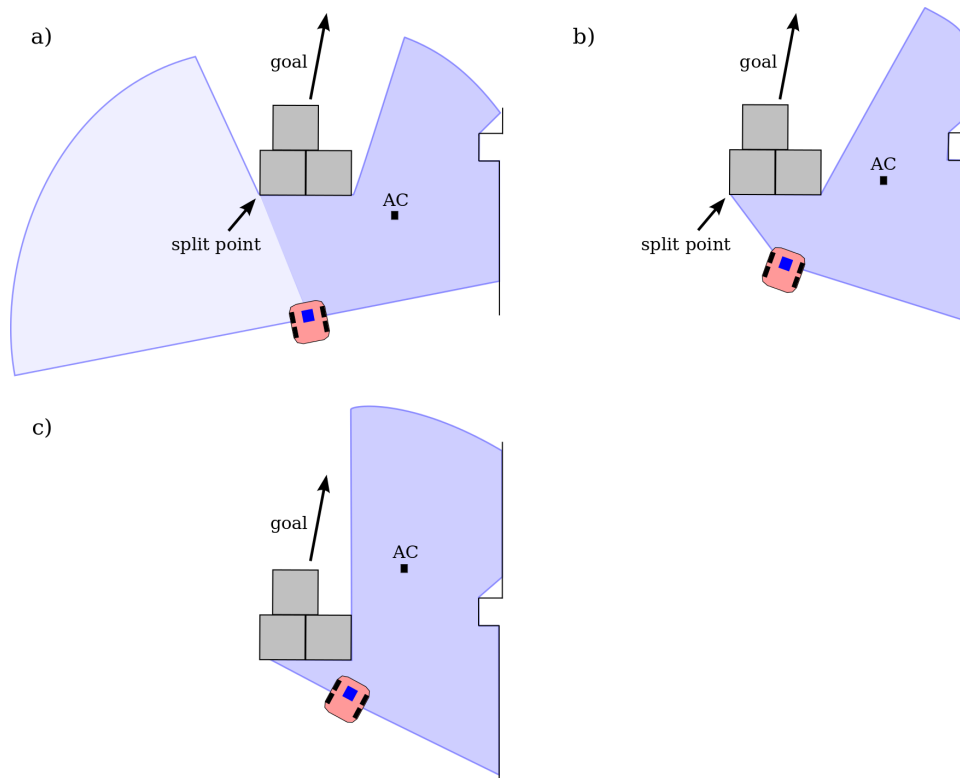


Figura 2.5: Inicialmente se coloca el punto de división en el pico más peligroso, pero al variar el robot su trayectoria sin hacerse el centro de áreas inaccesible, no se actualiza la posición del punto de división y la situación del robot pasa a ser peligrosa.

al no tener por qué volverse el centro de áreas inaccesible, puede no producirse un nuevo proceso de división, como se da en la figura 2.5. De ahí que es necesario que los puntos de división tengan que ser dinámicos, desplazando la posición del punto de división hacia el extremo del segmento que se encuentre más cercano al robot sin tener que mediar proceso de división (ver figura 2.6).

2.4. Segmentación en el centro de áreas básico

Una vez explicado el método del centro de áreas básico, se va a detallar el proceso de segmentación llevado a cabo en este método de navegación por la importancia del mismo (Cua-dra Troncoso, 2011).

La estimación de líneas del proceso de segmentación se consigue gracias a la extensión del filtro de Kalman con el procedimiento de Mínimos Cuadrados Generalizados (en inglés Generalized Least Squares, GLS) lineales y varianza del error no constante, modelando las ecuaciones de las rectas en coordenadas polares (ver figura 2.7). Además, se obtiene un intervalo de valores del factor de normalización de la ganancia del filtro para su adecuado rendimiento.

El procedimiento de clústering usado es una adaptación del scale-space para la obtención

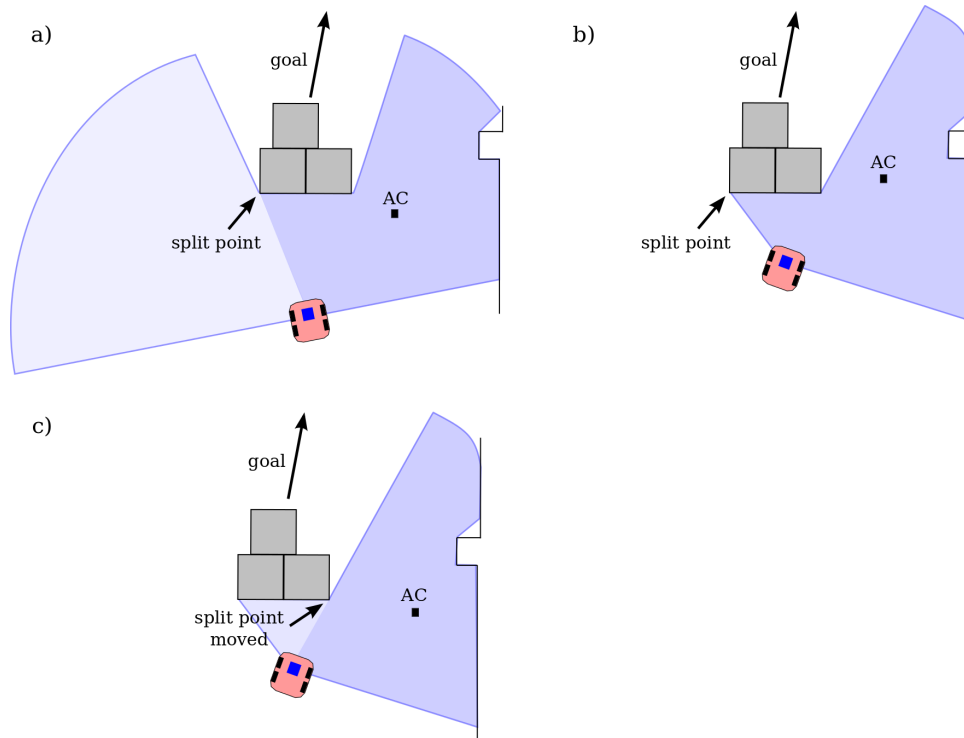


Figura 2.6: Como se actualiza la posición de los puntos de división del robot a la vez que el robot cambia su trayectoria, el punto de división se desplaza a una ubicación que permite al robot esquivar el obstáculo sin peligro aparente.

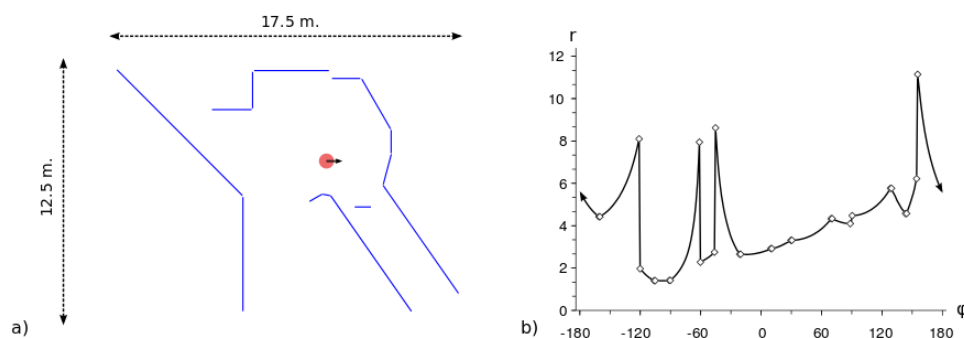


Figura 2.7: En la figura a) se muestra la visión que tiene el robot tomando medidas en 360 grados del mundo en coordenadas rectangulares, mientras que la figura b) es su correspondencia en coordenadas polares.

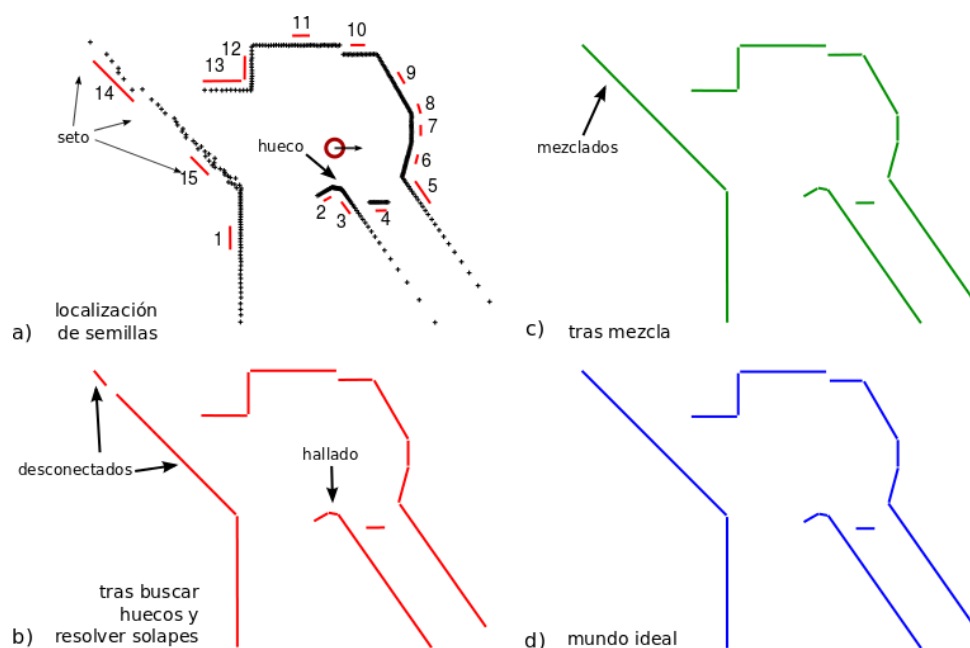


Figura 2.8: Proceso de segmentación formado por la estimación, resolución de solapamientos y mezcla de segmentos. Además se muestra el mundo ideal.

de segmentos en coordenadas polares, pudiendo determinar zonas interiores de estos segmentos a partir de los cuales inicializar el filtro de Kalman diseñado gracias a la primera y segunda derivada de la representación scale-space de las mediciones del robot. Primero se aplica un filtro de reducción de ruido, basado en la caracterización de las curvas de nivel obtenidas por el procedimiento del scale-space al alterar valores equiespaciados de un segmento, refinándolo posteriormente mediante un filtrado estadístico online que lo adapta al nivel de ruido presente en cada momento. Por último se procede a fusionar la información obtenida de ambas derivadas sobre los posibles extremos de los segmentos, y obteniendo las zonas interiores de los posibles segmentos que constituirán las semillas para la inicialización de los filtros de Kalman.

Una vez obtenidas las semillas mediante clústering, se aplica un procedimiento de la mediana repetida, con un criterio distinto de minimización distinto al de mínimos cuadrados, a cada una de estas semillas para estimar los parámetros de las rectas. Al aplicar el test de detección de outliers se extraen segmentos fiables más cortos. También se establece un criterio basado en varianzas unitarias para resolver posibles solapamientos entre segmentos, así como zonas que hayan quedado sin asociar a segmentos para realizar en ellas un proceso de estimación. Una vez hecho esto todos los puntos del haz están asociados a un segmento, aislados o como medidas que faltan. Finalmente, se establece un criterio de mezcla de segmentos adyacentes colineales para reducir las posibilidades de fragmentación de la representación del entorno concebida.

En la figura 2.8 se muestra un ejemplo de todo este proceso de segmentación.

Capítulo 3

Método propuesto

El método del centro de áreas básico, expuesto en las secciones 2.3 y 2.4, funciona bastante bien cuando el robot deambula por un entorno desconocido esquivando obstáculos o cuando debe alcanzar un objetivo, pero tiene problemas cuando lo que se pretende es que el robot explore el entorno. Este comportamiento inadecuado se debe a que cuando el robot llega a un punto del mapa donde existe una bifurcación o cruce, el centro de áreas que sigue el robot puede no convertirse en inaccesible, impidiendo el proceso de división y, por tanto, un cambio en la dirección del movimiento del robot. Este problema se puede ver perfectamente cuando hay un cruce simétrico, como el de la figura 3.1, donde el robot sigue su camino hacia adelante sin darse cuenta de que existen otros caminos alternativos por los que podría ir.

La solución adoptada para el problema de la captación del entorno por parte del robot consiste primero en una ampliación del algoritmo del centro de áreas, extendiendo el número de sectores de avance del robot de uno (el central) a tres (central, derecho e izquierdo). La consecuencia directa de esta ampliación en el número de sectores de avance es que el número de centros de áreas también se incrementa de uno a tres y, por tanto, el robot tiene la capacidad de captar información del mundo que le rodea según la accesibilidad o no de estos tres centros de áreas. Si lo llevarámos al ámbito humano sería como darle al robot las ventajas de tener un cuello que puede girar una serie de grados para mirar nuevas zonas accesibles cercanas a él pero que no se encuentran justo enfrente suya, sino a alguno de sus dos lados.

Otra técnica que se aplica para obtener información del entorno consiste en, una vez que el robot detecta que existen otros caminos alternativos al que está siguiendo, realizar un proceso de segmentación de las mediciones del robot. Observando las características de estos segmentos obtenidos (como distancias entre ellos o puntos de cruce entre ellos) se puede obtener información de si existen zonas libres (sin obstáculos), de si existen zonas con obstáculos pero en los que existe la posibilidad de que haya una zona de escape entre ellos, o si por el contrario, el robot está rodeado completamente de obstáculos en la parte frontal del robot.

Los experimentos realizados con el robot han puesto de manifiesto la importancia de la colocación de los puntos de división (o split points). Estos puntos de división ayudan a que

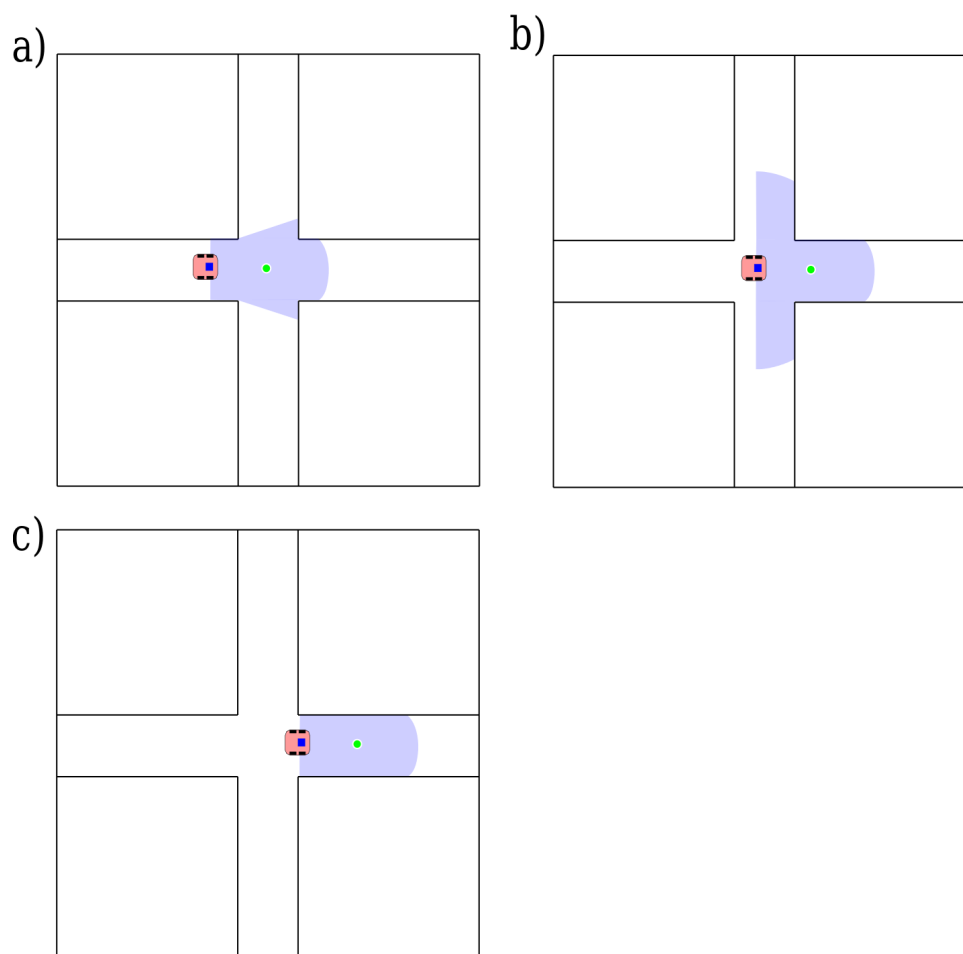


Figura 3.1: En todas las figuras el círculo de color verde representa el centro de áreas y el área sombreada es el sector de avance. Esta secuencia de figuras representa el problema del cruce simétrico, donde el robot con el método del centro de áreas es incapaz de detectar cuando existe un cruce, debido a que el centro de áreas siempre es accesible y, por consiguiente, el robot siempre va en la misma dirección.

el robot se mueva hacia el sitio fijado, y, si no se colocan en el sitio correspondiente, pueden hacer que, aunque el robot sepa correctamente a qué zona del mapa debe dirigirse, el propio movimiento del robot dirigido por estos puntos de división lleve a que tome un camino no adecuado o que no se pretendía inicialmente.

Para una mejor comprensión del método propuesto, se va a descomponer éste en las diferentes fases de construcción del mismo.

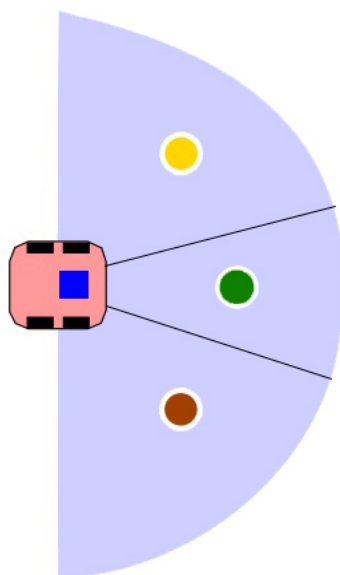


Figura 3.2: El rango de datos de los sensores del robot se decompone en tres sectores (polígonos): el izquierdo cuyo centro de áreas está representado con el círculo de color amarillo, el central cuyo centro de áreas está representado con el círculo de color verde, y el izquierdo cuyo centro de áreas está representado con el círculo de color marrón.

3.1. Fase 1: centro de áreas múltiple

En el algoritmo del centro de áreas básico sólo se tiene en cuenta lo que tiene el robot enfrente suya (sector central con una amplitud de 180°) para tomar una decisión de por dónde seguir. En cambio, para poder implementar el algoritmo del centro de áreas múltiple es necesario que el robot reconozca cuándo se está muy próximo de una bifurcación a la derecha, a la izquierda o a ambas, por lo que es necesario realizar una serie de modificaciones para que también recopile información de los sectores izquierdo y derecho. Para ello el robot necesita de dos nuevos polígonos de centro de áreas: el derecho e izquierdo (ver figura 3.2). La manera de definir la amplitud de estos dos sectores viene determinada por la amplitud del sector central:

- El sector central tiene X grados.
- El sector izquierda irá desde -90 grados hasta $(0 - \frac{X}{2})$ grados.
- El sector central irá desde $(0 - \frac{X}{2})$ grados hasta $(0 + \frac{X}{2})$ grados.
- El sector derecho irá desde $(0 + \frac{X}{2})$ grados hasta 90 grados.

También es necesario cambiar respecto al algoritmo del centro de áreas básico la política que decide hacia dónde debe dirigirse el robot. La política tomada para el algoritmo del centro de áreas múltiple es la siguiente:

1. Si el centro de áreas del sector central es inaccesible, quiere decir que el robot tiene un obstáculo justo enfrente que debe esquivar y, por tanto, se sigue la política tradicional consistente en dividir el área percibida en dos y calcular el centro de áreas de los dos polígonos resultantes. Si ambos centros de áreas son accesibles el robot se dirige hacia uno al azar; si sólo uno es accesible se dirige hacia éste, y si ninguno es accesible se busca el centro de áreas del sector trasero del robot (para dar media vuelta y escapar).
2. Si el centro de áreas del sector central es accesible, pero el sector de avance está comprimido más de un 85 %, caso que se da cuando el robot está en plena esquivación del obstáculo detectado, se sigue dicho centro de áreas, ya que lo primordial en ese estado es que el robot no choque contra el obstáculo y ya se preocupará de seguir explorando el entorno cuando lo esquive.
3. Si el centro de áreas del sector central es accesible y el sector de avance está completo o no está comprimido más de un 85 %, quiere decir que no hay obstáculos cercanos al robot en su parte frontal. Por ello comprueba la información que le llega de sus sectores derecho e izquierdo para tomar una decisión de movimiento (ver figura 3.3):
 - a) Si el centro de áreas del sector izquierdo y derecho es accesible significa que no se ha detectado bifurcación ninguna próxima, por lo que el robot sigue el centro de áreas del sector central.
 - b) Si el centro de áreas del sector izquierdo no es accesible pero sí el del derecho quiere decir que en el sector izquierdo hay una bifurcación. Una vez detectado este evento se procede igual que si el centro de áreas del sector central fuera inaccesible, pero con el sector izquierdo: se divide el centro de áreas del sector izquierdo en dos y se calcula el centro de áreas de los dos polígonos resultantes. Si el centro de áreas de los dos polígonos es accesible se elige el centro de áreas de un polígono u otro al azar (cuanto mayor sea el área del sector mayor es la probabilidad de ser el elegido). Si sólo es accesible uno de los dos centros de áreas se elige éste para ser seguido por el robot; y si ninguno es accesible se sigue el centro de áreas del sector central.
 - c) Si el centro de áreas del sector derecho no es accesible pero sí el del izquierdo quiere decir que en el sector derecho hay una bifurcación. Una vez detectado este evento se procede igual que anteriormente, pero con el sector derecho.
 - d) Si ni el centro de áreas del sector izquierdo ni el del sector derecho son accesibles significa que en ambos sectores hay una bifurcación. Una vez detectado este evento se elige completamente al azar uno de los dos sectores y se procede igual que anteriormente: si se elige el sector derecho se dividirá el derecho y si se elige el izquierdo se dividirá el izquierdo.

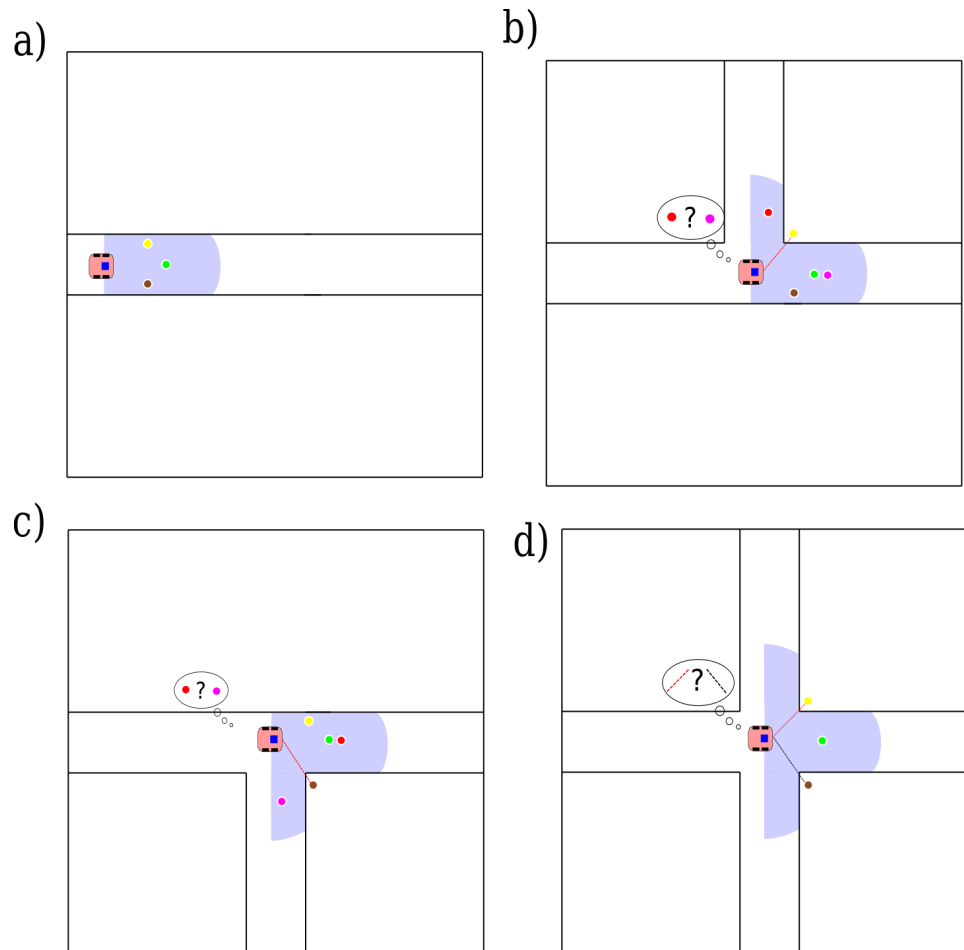


Figura 3.3: En todas las figuras el círculo de color verde representa el centro de áreas del sector central, el de color amarillo es el centro de áreas del sector izquierdo, el de color marrón el del sector derecho, y los círculos de color rojo y fucsia son los centros de áreas de los subsectores resultantes tras el proceso de división. En la figura a) los centros de áreas derecho, central e izquierdo son accesibles; en la figura b) el centro de áreas izquierdo es inaccesible y se produce la división; en la figura c) el centro de áreas derecho es inaccesible y se produce la división; y en la figura d) tanto el centro de áreas derecho como el izquierdo son inaccesibles, debiendo elegir uno de los dos para llevar a cabo la correspondiente división.

3.2. Fase 2: centro de áreas múltiple con segmentación

Como se ha podido comprobar en 2.1 existe una gran cantidad de técnicas de segmentación que pueden ser aplicadas a este problema.

En primer lugar, se había pensado usar técnicas que midan, de forma cuantitativa, la forma del polígono de avance del robot; y así, poder sacar conclusiones de las características del entorno que rodean al robot. Una primera aproximación a esta idea era usar momentos estadísticos de diferentes órdenes, pero se vio que los cálculos a realizar eran bastante complicados según se iba incrementando el orden, y se decidió rechazar este procedimiento. Por otro lado, en (de la Paz López, 2003) se habla de cómo construir un modelo del medio a través del conjunto de puntos dados por el valor de los sensores de un robot, discretizando en zonas según el valor de la primera y segunda derivada. La complejidad de cálculo de primeras y segundas derivadas es mucho menor que de momentos de orden superior, pero, al aplicarla sobre el problema que se plantea, se comprobaba que las zonas abarcaban muy pocos sensores y, por tanto, la amplitud no era lo suficiente como para que el robot tuviera la confianza de seguir esa zona.

Debido a estos fracasos y tras analizar de forma detallada los experimentos realizados en (Cuadra Troncoso, 2011) se decidió usar el proceso de segmentación de los datos de los sensores del robot que resulta de ese estudio, y que ya se ha explicado en la sección 2.4. Este proceso de segmentación se produce cuando se detecta que existe una bifurcación próximamente a la derecha o a la izquierda del robot gracias a los tres centros de áreas, y tiene en cuenta que para que se forme un segmento éste debe estar compuesto por cinco o más puntos (medidas de los sensores láser).

Una vez que se ha realizado el proceso de segmentación sobre los datos provenientes de los sensores láser del robot se tiene un conjunto de segmentos que se relacionan con una serie de sensores láser. Si se analiza, convenientemente estos segmentos, el robot puede tener información muy valiosa sobre el entorno que le rodea, como por ejemplo si existen zonas sin obstáculos que esquivar o si se intuye alguna posible bifurcación (ver figura 3.4).

El primer caso que se puede dar es que una serie de sensores consecutivos no tenga ningún segmento asociado (los sensores del robot dan el valor máximo de alcance), existiendo un área libre. La información que se puede sacar de este caso es que existe una zona sin obstáculos (libre) delimitado por estos sensores, y que si decide ir hacia allí el conocimiento que tiene sobre su entorno aumentará. Un ejemplo de este caso es cuando el robot está en una bifurcación y no llega a ver el fin de uno de los pasillos (ver figura 3.5).

El segundo caso que puede ocurrir es que entre dos segmentos consecutivos no exista conexión ninguna aunque se prolonguen indefinidamente (es decir, son segmentos paralelos), o que ambos segmentos se corten en su prolongación fuera del rango delimitado por el último sensor del primer segmento y el primer sensor del segundo segmento más un pequeño margen (ya que puede existir pequeños errores). Si sucede esto a la conclusión a la que se llega es que existen dos obstáculos y que entre ellos existe una zona que puede ser una bifurcación o no (dependiendo del caso).

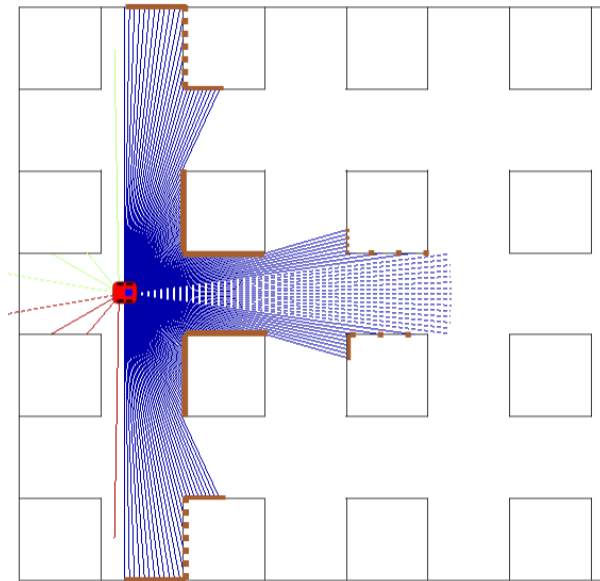


Figura 3.4: Esta imagen muestra los segmentos que se obtienen de los valores de los sensores del robot tras el proceso de segmentación. Las líneas continuas marrones son segmentos que se corresponden a varios sensores, mientras que la línea discontinua marrón son segmentos que se corresponden a un único sensor del robot. Las líneas azules son los sensores del robot.

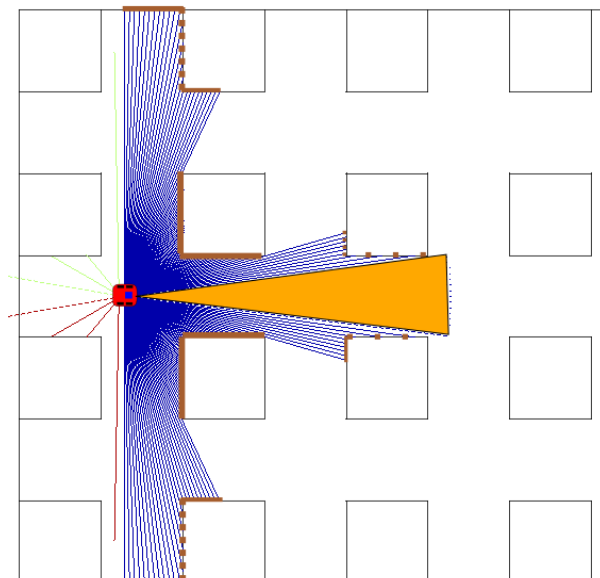


Figura 3.5: El triángulo naranja se corresponde con el área libre que ha detectado el robot tras la segmentación, ya que en esa zona no se ha obtenido ningún segmento para esos sensores.

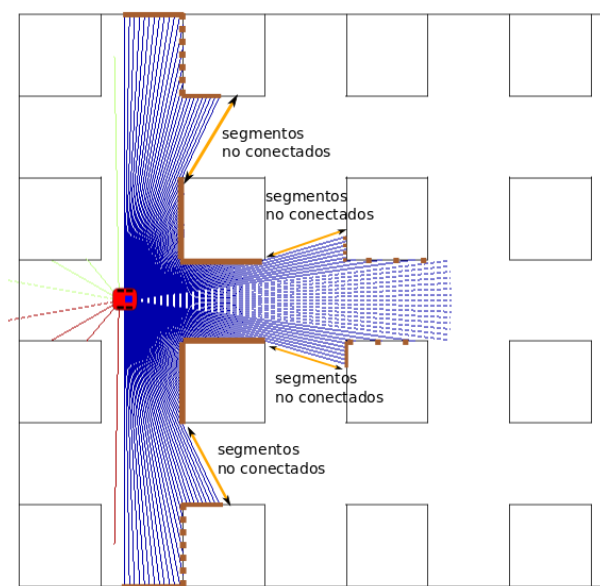


Figura 3.6: Las líneas naranjas marcan el extremo de los pares de segmentos que no están conectados. En esta imagen hay un total de cuatro pares de segmentos no conectados.

Por ejemplo, el robot está en un pasillo largo y este pasillo se bifurca en otros a su derecha o izquierda. En este caso los segmentos no están conectados (ver figura 3.6).

El tercer y último caso es que dos segmentos consecutivos (o su prolongación) se corten dentro del rango delimitado por el último sensor del primer segmento y el primer sensor del segundo segmento más un pequeño margen (ver figura 3.7). Se dice que los segmentos están conectados. Si es así se puede decir que existen dos obstáculos que se encuentran extremadamente cerca (o juntos) el uno del otro, como puede ser el caso de dos paredes con una esquina común.

Como se ha mencionado anteriormente, los segmentos conectados indican al robot que hay obstáculos a su alrededor muy juntos, por lo que el robot debe tener mucho cuidado con los movimientos que haga si quiere ir hacia esa zona. Por ello, los segmentos conectados hay que dividirlos en subconjuntos de segmentos que conformen un polígono convexo con respecto al robot, ya que por definición todos los movimientos que se hagan dentro del área de un polígono convexo son seguros. En resumen, cada conjunto de segmentos conectados que se tengan tras la segmentación se debe dividir en subconjuntos de segmentos que cumplan que el polígono formado por estos segmentos y el robot forme un polígono convexo (ver figura 3.8).

Una vez que se ha detectado una posible bifurcación y que se ha descompuesto la percepción que tiene el robot de su entorno cercano, primero en segmentos, y luego estos segmentos en áreas libres, áreas convexas y segmentos no conectados, se procede a tomar la decisión de movimiento del robot basado en este modelo del entorno.

El razonamiento que se sigue es el siguiente:

1. Si existe una o varias áreas libres se descartan aquéllas que no alcanza una determinada amplitud. Se pueden dar dos casos:

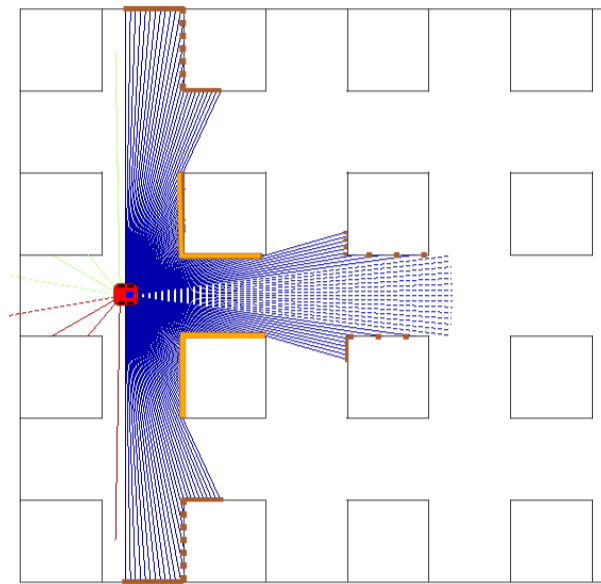


Figura 3.7: Las líneas naranjas marcan los segmentos que están conectados. En este ejemplo se ven que hay dos segmentos conectados por un lado y otros dos por otro.

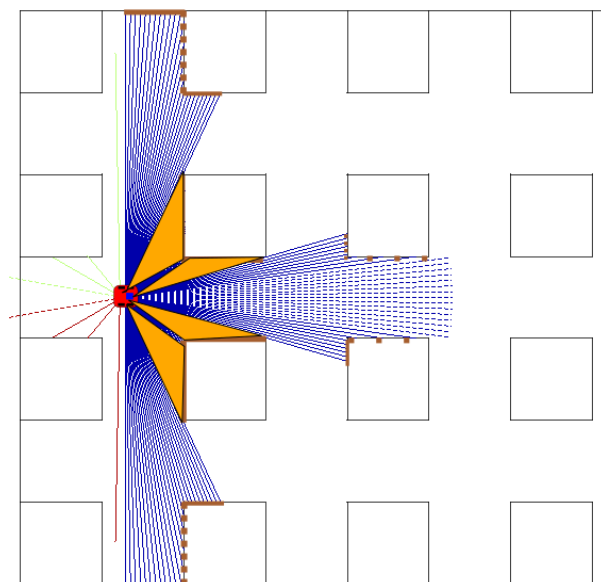


Figura 3.8: Las zonas naranjas marcan las distintas áreas convexas que existen a partir de los segmentos conectados. En este ejemplo hay un total de cuatro áreas convexas.

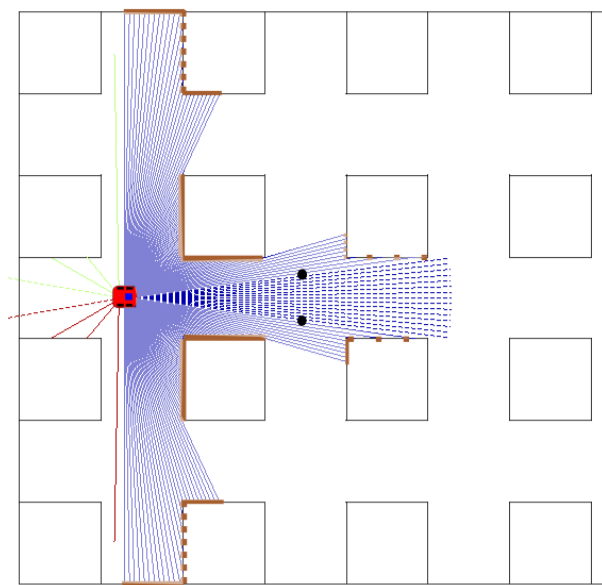


Figura 3.9: Los dos puntos de división son los puntos en color negro, mientras que los sensores con un color azul más claro son los que forman parte de los sectores de avance descartados, y los de un color azul más oscuro forman parte del sector de avance actual. Los puntos de división se colocan a la misma distancia del robot, a ambos extremos del área libre.

- a) Tras los descartes ya no quedan áreas libres candidatas: ir al paso 2.
 - b) Quedan áreas libres candidatas: se escoge una de forma aleatoria para ser por la que el robot deberá dirigirse con la ayuda de los split points o puntos de división.
 - Si el área libre coincide con el inicio de la visión del robot se coloca un único split point o punto división a una distancia determinada del robot (siempre menor que el rango máximo de los sensores) en la dirección del último sensor que se encuentre dentro del área libre escogida.
 - Si el área libre coincide con el fin de la visión del robot se coloca un único split point o punto división a una distancia determinada del robot (siempre menor que el rango máximo de los sensores) en la dirección del primer sensor que se encuentre dentro del área libre escogida.
 - Si el área libre no coincide ni con el inicio ni con el fin de visión del robot se ponen dos split points a la misma distancia del robot robot (siempre menor que el rango máximo de los sensores), uno en la dirección del primer sensor que se encuentra dentro del área libre escogida y otro en la dirección del último sensor dentro de esta área libre (ver figura 3.9).
2. Si se han detectado pares de segmentos no conectados se descartan aquellos donde la distancia entre el final del primer segmento y el inicio del siguiente segmento no conectado sea menor a un valor dado en función de la amplitud del robot. Se pueden dar dos casos:

- a) Tras los descartes ya no quedan huecos entre conjuntos de segmentos no conectados: ir al paso 3.
 - b) Quedan huecos entre conjuntos segmentos no conectados: se escoge uno de los huecos de forma aleatoria para ser por el que el robot deberá dirigirse. Para conseguir que el robot se mueve hacia allí se coloca un único split point, que debe situarse en el extremo no conectado más próximo al robot. Con esto se consigue que el robot no tenga en cuenta nada de lo que hay en esa zona desde donde está posicionado hasta llegar al split point (ver figura 3.10).
3. Si existen áreas convexas se descartan aquéllas cuya amplitud sea menor a un valor dado. Se pueden dar dos casos:
- a) Tras los descartes ya no quedan áreas convexas candidatas: ir al paso 4.
 - b) Quedan áreas convexas candidatas: se escoge una de forma aleatoria para ser por la que el robot deberá dirigirse con la ayuda de los split points o puntos de división. Los dos split points se colocan en ambos extremos de los segmentos que forman el área convexa (ver figura 3.11).
4. No se ha encontrado una escapatoria que satisfaga los requisitos del robot, por lo que se sigue el centro de áreas original, en búsqueda de otra bifurcación.

3.3. Fase 3: mejoras al centro de áreas múltiple con segmentación

A veces ocurre que el robot se da cuenta que existe una bifurcación a su derecha o a su izquierda cuando ya ha sobrepasado parte de la bifurcación, produciéndose los dos siguientes comportamientos no deseados.

Para simular el primer comportamiento no deseado se va a suponer el siguiente caso (ver figura 3.12): el robot detecta que existe una bifurcación y comprueba cuáles son las características del entorno que le rodea en ese momento. Una vez realizado el proceso de segmentación y descomposición en áreas libres, convexas y conjuntos de segmentos no conectados observa que no existen áreas libres con una amplitud suficiente como para poder pasar por ellas. Sin embargo, si se observa la figura 3.12 se ve que existe suficiente espacio como para que el robot pase por ahí sin ningún problema.

El otro comportamiento no deseado es el que se muestra en la figura 3.13: tras detectar el robot que existe una bifurcación y tras el proceso de segmentación, observa que existe un área libre con una amplitud suficiente como para poder pasar por ella. Como se puede observar la percepción del robot le hace creer que el área libre seguir es mucho más pequeño de lo que realmente es, situando los split points en sitios nada convenientes.

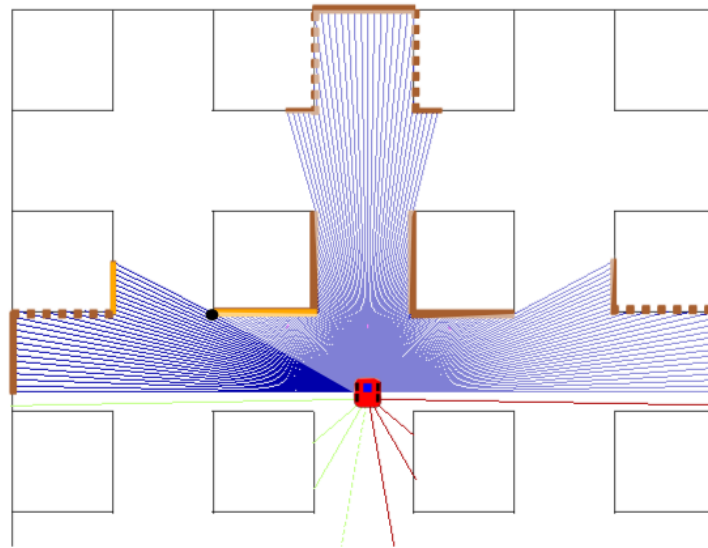


Figura 3.10: El punto de división es el punto en color negro, los segmentos naranjas son los segmentos no conectados por los que el robot ha decidido guiarse para seguir su camino al no existir ningún área libre, mientras que los sensores con un color azul más claro son los que forman parte del sector de avance descartado, y los de un color azul más oscuro forman parte del sector de avance actual. Se coloca el punto de división en el extremo no conectado más próximo al robot.

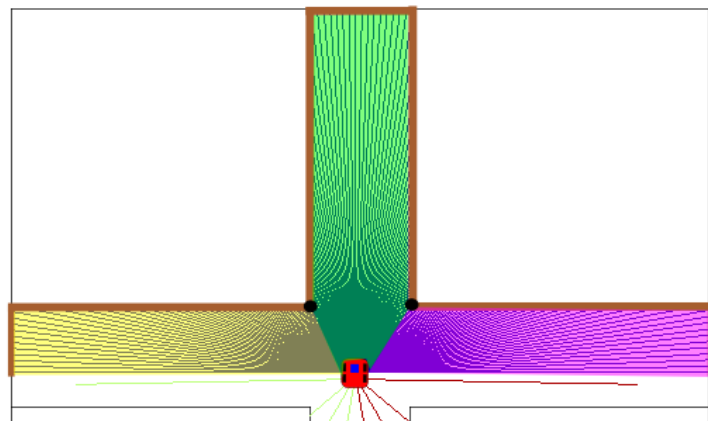


Figura 3.11: Los dos puntos de división son los puntos en color negro; mientras que en colores amarillo, verde y rosa se marca las áreas convexas que se forman a partir de los segmentos conectados. En este caso el robot escoge seguir por el área convexa de color verde (no hay ni áreas libres ni segmentos no conectados), por lo que se colocan dos puntos de división en ambos extremos de los segmentos que forman el área convexa.

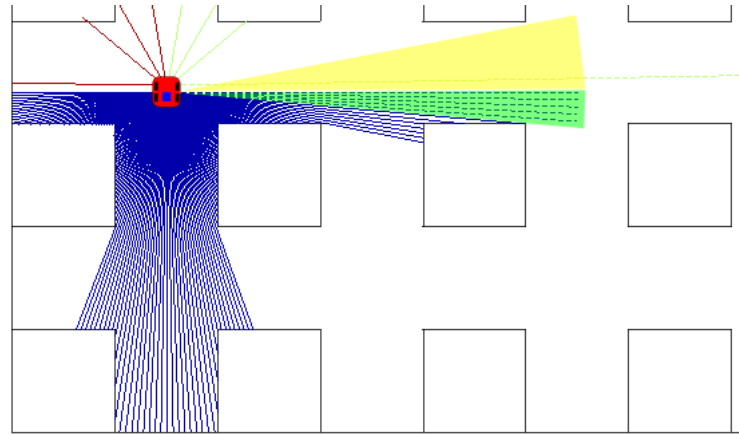


Figura 3.12: En verde se marca el área libre detectada por el robot, mientras que en amarillo el espacio que el robot no detecta como área libre. Cuando el robot se da cuenta de que existe una bifurcación y observa el entorno para ver si existe algún área libre por donde ir ya es demasiado tarde porque parte de ese área libre ya lo ha dejado atrás.

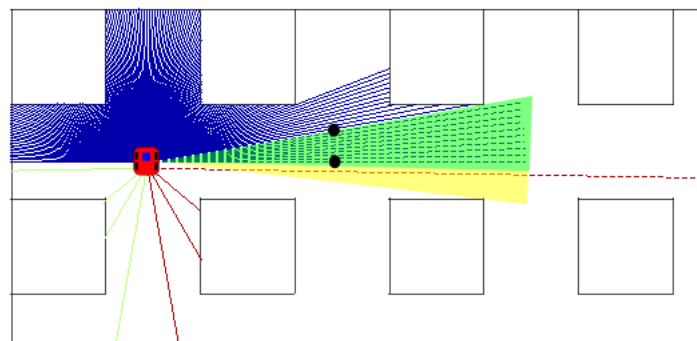


Figura 3.13: Los puntos negros son los puntos de división, en verde se marca el área libre detectada por el robot, mientras que en amarillo el espacio que el robot no detecta como área libre. El robot se da cuenta a tiempo de que existe un área libre hacia donde puede dirigirse, pero debido a que parte de este área libre ya ha sido sobrepasada por él se sitúan los split points en lugares que no son los óptimos.

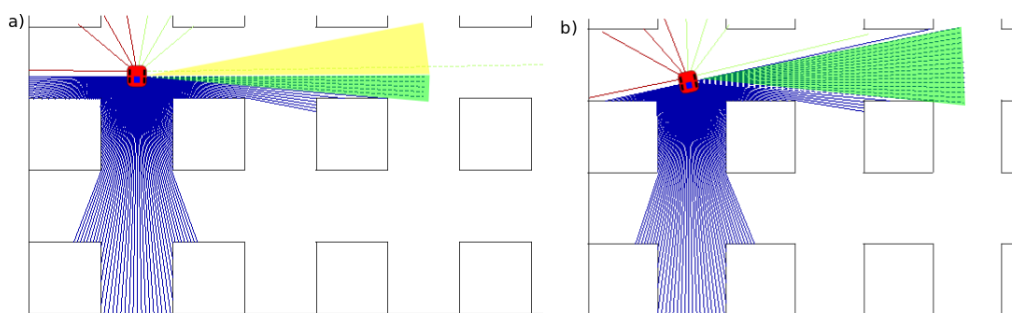


Figura 3.14: En verde se marca el área libre detectada por el robot, mientras que en amarillo el espacio que el robot no detecta como área libre. En la figura a) el robot detecta una bifurcación y observa un área libre muy estrecha. En la figura b) cuando el robot se para y empieza a girar sobre sí mismo descubre que el área libre es mucho más ancha de lo que realmente pensaba en un principio.

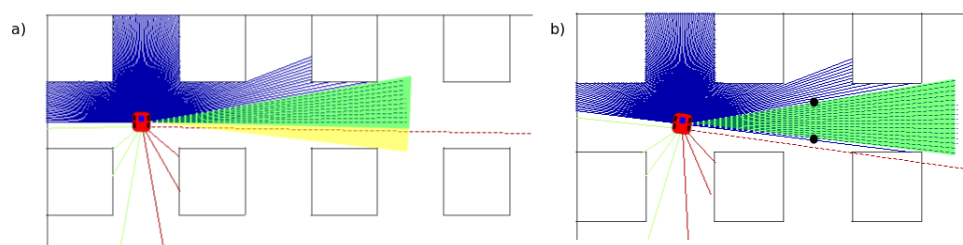


Figura 3.15: Los puntos negros son los puntos de división, en verde se marca el área libre detectada por el robot, mientras que en amarillo el espacio que el robot no detecta como área libre. En la figura a) el robot no tiene una visión total del área libre. En la figura b) cuando el robot se para y empieza a girar sobre sí mismo descubre que el área libre es más ancha, poniendo los puntos de división más separados entre ellos.

Estos dos problemas detectados pueden solucionarse si, una vez detectado una bifurcación próxima a su derecha o a su izquierda y el proceso de segmentación ha producido áreas libres en los extremos derecho o izquierdo del campo de visión del robot, éste es capaz de pararse y girar sobre sí mismo hacia esa zona libre (o ambas) y encontrar dónde acaba ese área libre, teniendo así una visión total del área libre percibida. Una vez que tenga esta información ya puede decidir si las áreas libres siguen siendo lo suficientemente pequeñas como para no decidir ir por ese camino o si el área libre es más amplio de lo que realmente se pensaba en un principio y poder separar más los split points (ver figuras 3.14 y 3.15).

Otro aspecto que se ha visto que era necesario mejorar del algoritmo básico del centro de áreas múltiple con segmentación era dónde había que colocar los puntos de división o split points tras tomar la decisión de seguir un área libre.

El problema de esto se puede ver en la figura 3.16. Los split points hacen que el robot tenga en cuenta sólo la información de los sensores que hay entre ambos split points (o entre el split point y el extremo si sólo hay uno) hasta que los sobrepase, pudiéndose dar el caso de que

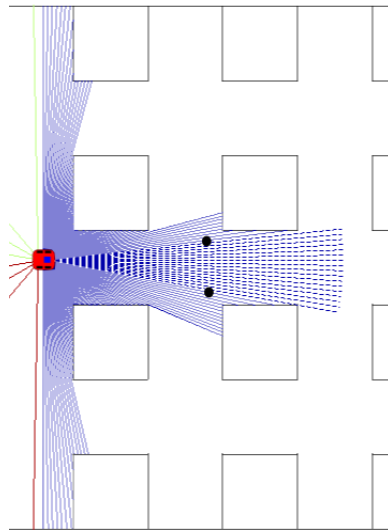


Figura 3.16: Los puntos negros son los puntos de división, mientras que los sensores con un color azul más claro son los que forman parte del sector de avance descartado, y los de un color azul más oscuro forman parte del sector de avance actual. Al situarse los split a una distancia predefinida se da el caso de que se sitúan muy cerca del fin de la próxima bifurcación, lo que impide al robot darse cuenta de la misma

haya otra bifurcación cercana y que el robot no sea capaz de detectarla porque está centrado en alcanzar los split points colocados, o que la detecte cuando ya es tarde.

La solución a este problema es que los split points se coloquen en un sitio u otro dependiendo del entorno que rodea al robot en ese momento, y que no sean colocados a una distancia prefijada; es decir, sean adaptativos al cambio según la situación que se le presente al robot en un determinado momento.

La nueva colocación de los split points cuando el robot quiera seguir un área libre va a consistir en detectar el obstáculo más cercano a cada uno de los lados del área libre y que esté a una distancia menor que una dada del robot, haciendo un barrido hacia la derecha en el caso del lado derecho del área libre y un barrido hacia la izquierda en el caso del lado izquierdo del área libre. Una vez detectados los obstáculos se colocarán ahí los split points (ver figura 3.17). Sólo se colocaran los split points a una distancia prefijada si no se ha encontrado ningún obstáculo que cumpla con esa premisa.

Además, con esta solución se consigue que los split points se correspondan con elementos físicos del mapa, que siempre es mucho más aconsejable que que no haya correspondencia.

Como se ha comentado anteriormente, cuando el robot se encuentra encerrado entre obstáculos y el resultado de aplicar la técnica de segmentación es un conjunto de segmentos conectados, se aplica la descomposición en áreas convexas para que el robot tenga una mayor seguridad al moverse. El problema viene que, como puede haber varias zonas convexas y el robot elige aleatoriamente una de ellas para dirigirse por ahí, puede elegir un área convexa cuyo fin esté

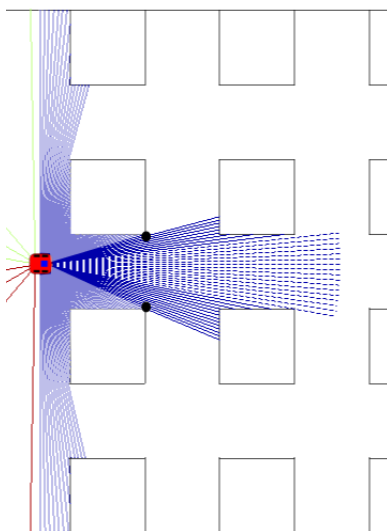


Figura 3.17: Los puntos negros son los puntos de división, mientras que los sensores con un color azul más claro son los que forman parte del sector de avance descartado, y los de un color azul más oscuro forman parte del sector de avance actual. Con la nueva política de colocación de split points, éstos se sitúan en un sitio mucho más conveniente y alcanzables por el robot más rápidamente.

a poca distancia de donde se encuentre el robot, produciendo que al cabo de unos instantes el robot dé media vuelta cuando llegue al final de este área (ver figura 3.18).

Sin embargo, si el robot tomara la decisión de a qué área convexa dirigirse, en vez de aleatoriamente, basado en las informaciones que capta de cada área convexa, el comportamiento es mucho más adecuado. Como lo que importa en este caso es que el fin del área convexa esté lejos para que el robot tenga que hacer un mayor recorrido y así poder descubrir otras vías de exploración, cuando el robot tiene que tomar una decisión sobre a qué área dirigirse lo hace en función de dos aspectos: amplitud del área convexa y distancia del robot al punto más lejano de los segmentos que componen el área libre. Si se multiplican estos dos valores lo que está haciendo el robot es una ponderación, o lo que es lo mismo, da un valor numérico a cada área convexa en función de esa distancia y amplitud. Cuánto mayor sea este valor numérico significa que es un área convexa más recomendable para ser seguida por el robot. Una vez que se ha calculado este valor para todas las áreas convexas, el robot escoge el área convexa con este valor numérico más alto para seguir su camino (ver figura 3.19).

Finalmente, el comportamiento que un robot autónomo debe tener siempre es ser capaz de no chocar contra los obstáculos que hay a su alrededor. Aunque en el método del centro de áreas múltiple con segmentación se busca que los movimientos del robot sean lo más seguros posibles, existen casos aislados en los que el robot se encuentra en una situación de bastante peligro porque se encuentra con un obstáculo muy próximo que, o bien los sensores no han sido capaces de detectar a tiempo, o bien el afán explorativo del robot ha hecho que tome alguna decisión

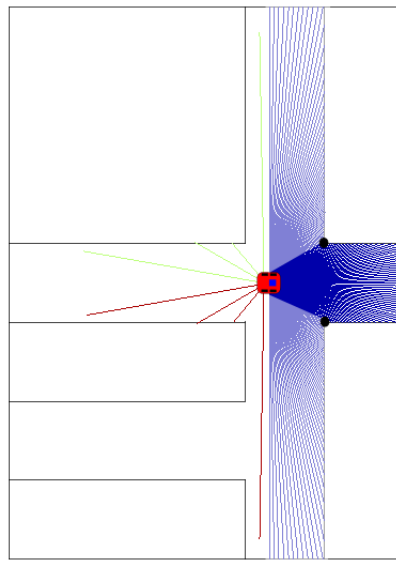


Figura 3.18: Los puntos negros son los puntos de división, mientras que los sensores con un color azul más claro son los que forman parte del sector de avance descartado, y los de un color azul más oscuro forman parte del sector de avance actual. Al elegir de forma aleatoria una de las áreas convexas sin tener en cuenta sus características se puede elegir erróneamente cuál debe seguir el robot.

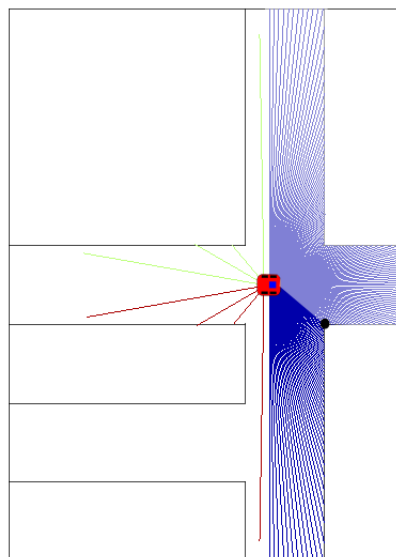


Figura 3.19: El punto negro es el punto de división, mientras que los sensores con un color azul más claro son los que forman parte del sector de avance descartado, y los de un color azul más oscuro forman parte del sector de avance actual. En este ejemplo el área convexa que se encuentra justo enfrente del robot se descarta porque su fin está muy próximo a donde se encuentra el robot, eligiendo en este caso el área convexa derecha.

arriesgada que le lleve a este punto. Por ello, cualquier cambio de estado de comportamiento de robot como, por ejemplo, el robot pasa de buscar una bifurcación a moverse a una que ha encontrado, debe tener siempre presente antes de ejecutar cualquier movimiento si es accesible o no su centro de áreas central para, en el caso de que no lo sea, pasar a un comportamiento de esquivar el obstáculo, siguiendo el método de centro de áreas básico.

Capítulo 4

Resultados experimentales

El método descrito en las secciones anteriores se implementó en el simulador Cybersim (Cuadra, 2011) para comprobar su validez bajo diversas circunstancias. Cybersim es un software escrito en C++ y que usa la librería gráfica de Qt-4 orientado a la simulación y el control de conductas en robots autónomos en desarrollo construido por la UNED.

4.1. Descripción del robot

El robot simulado es similar a un Pioneer-3AT (MobileRobots Inc., 2011a)(Inc., 2011), con dos ruedas cuyas velocidades se controlan independientemente. Los sensores usados por el robot para las simulaciones realizadas son sus sensores láser, aunque por defecto también viene equipado con otro tipo de sensores. En total el robot tiene 181 sensores láser equiespaciados 1 grado en su portal frontal, desde los -90 grados hasta los 90 grados.

En cuanto a la cinemática simulada del robot, tanto el ruido como la inercia son configurables en Cybersim. El ruido es el error en los datos captados por los sensores del robot (por ejemplo, el robot detecta a través de uno de sus sensores que existe un obstáculo a un metro de distancia, cuando en realidad está a medio metro de distancia); mientras que la inercia es la incapacidad que tiene el robot de modificar por sí mismo el estado de reposo o movimiento en el que se encuentra (por ejemplo, la distancia o tiempo que tarda en frenarse completamente). En las pruebas que se van a detallar en este capítulo tanto el ruido como la inercia toman valor 0 (caso ideal), ya que inicialmente lo que se pretende es probar que el nuevo método de navegación funciona correctamente en un entorno ideal, y luego probarlo en un entorno real.

4.2. Descripción de los mapas

Se usaron cuatro clases de mapas para las pruebas (ver imagen 4.1). El primer mapa se compone de un conjunto de pasillos en los que unos se comunican con otros en distintos cruces. El segundo mapa es aún un poco más complicado, ya que hay zonas que son menos perceptibles

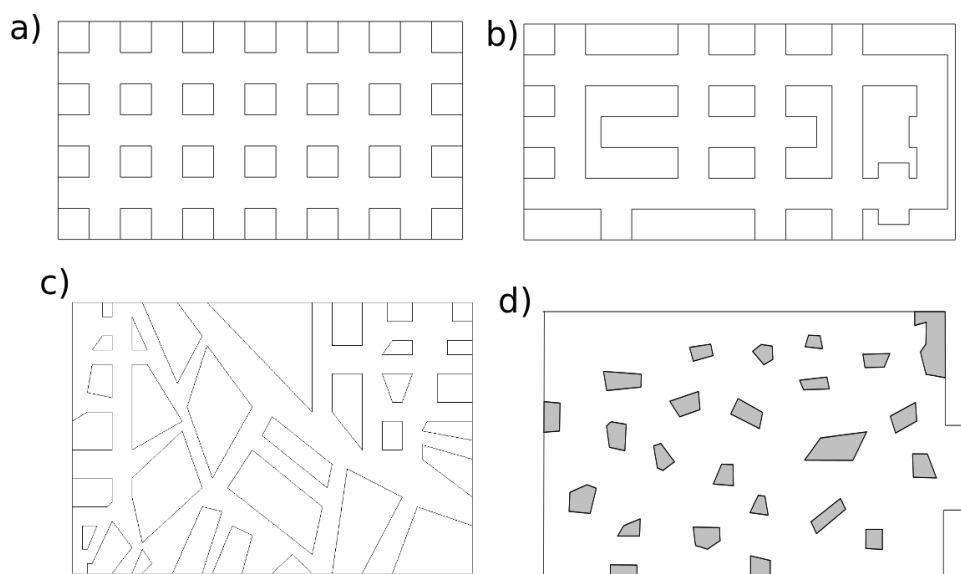


Figura 4.1: a) Mapa 1, b) Mapa 2, c) Mapa 3, d) Mapa 4

por el robot a simple vista y se necesita una exploración mayor, con las paredes aún en ángulo recto. En el tercer mapa los pasillos ya no son rectos en su mayoría, sino que se asemejan más a una distribución del plano de una ciudad. Finalmente el último de los mapas es un espacio abierto en el que hay distintos obstáculos, con diferentes tamaños y formas.

4.3. Descripción de los parámetros

En las distintas pruebas que se han realizado en los mapas se ha variado una serie de parámetros que afectan al rendimiento del método de navegación descrito. Los parámetros que se han ido modificando son los siguientes:

- Amplitud del sector central: la amplitud del sector central afecta en la concepción que el robot tiene de cuál es su sector central y los sectores laterales izquierdo y derecho. Trasladado esto a un ser humano, sería cuánto se gira el cuello para mirar a los dos lados. Cuanto mayor sea el sector central, menor serán los sectores laterales y, por tanto, el robot observará más lateralmente (“gira más el cuello”) y viceversa, cuanto menor sea el sector central, mayores serán los sectores laterales y, por tanto, el robot observará más céntricamente (“gira menos el cuello”).
- Proporción de los subsectores centrales: cuando el sector central se va a dividir en dos subsectores debido a que el centro de áreas es inaccesible, existe un parámetro que indica el tamaño mínimo que deben tener ambos subsectores respecto al polígono de avance para que pueda producirse tal división, si no superan ese tamaño mínimo el robot pasa a dar media vuelta sobre sí mismo. Este parámetro se incluye para que el robot no pueda elegir

un sector libre que sea demasiado pequeño para él y haya alta probabilidad de que choque contra el obstáculo que pretende esquivar.

- Proporción de los subsectores derecho e izquierdo: al igual que pasa con el sector central, cuando el sector derecho o izquierdo se va a dividir en dos subsectores debido a que el centro de áreas es inaccesible, existe otro parámetro que indica el tamaño mínimo que deben tener ambos subsectores respecto al polígono de avance para que pueda producirse tal división. Al igual que el anterior parámetro, éste se incluye para que el robot no pueda elegir un sector libre que sea demasiado pequeño para él y haya alta probabilidad de que choque contra alguna pared u obstáculo cercano.
- Proporción del sector de avance elegido: a veces sucede que el sector de avance elegido por el robot no viene limitado por un sólo split point (punto de división), sino por varios; por lo que es necesario comprobar también en este punto que dicho sector de avance tiene un tamaño adecuado para que el robot pueda pasar por él y no chocar con objetos cercanos.

4.4. Experimentos

Se han realizado multitud de experimentos con diferentes valores para los parámetros explicados en la sección 4.3 y en los diferentes mapas descritos en la sección 4.2. Es importante destacar en este punto que todos los experimentos han tenido la misma duración, para así poder ser evaluados bajo las mismas condiciones.

En las pruebas se han usado 4 configuraciones distintas para los parámetros que se usan con el centro de áreas múltiple:

- Configuración 1
 - Proporción de los subsectores centrales: 0,07
 - Proporción de los subsectores derecho e izquierdo: 0,15
 - Proporción del sector de avance elegido: 0,07
- Configuración 2
 - Proporción de los subsectores centrales: 0,15
 - Proporción de los subsectores derecho e izquierdo: 0,15
 - Proporción del sector de avance elegido: 0,07
- Configuración 3
 - Proporción de los subsectores centrales: 0,15
 - Proporción de los subsectores derecho e izquierdo: 0,15

	Central
Mapa 1	49/63
Mapa 2	41/60
Mapa 3	71/180
Mapa 4	94/102
Total	255/405

Cuadro 4.1: Resultados del centro de áreas básico

	Sector central de 1 grado			
	Configuración 1	Configuración 2	Configuración 3	Configuración 4
Mapa 1	63/63	63/63	61/63	60/63
Mapa 2	50/60	57/60	49/60	50/60
Mapa 3	141/180	166/180	129/180	122/180
Mapa 4	88/102	95/102	82/102	80/102
Total	342/405	381/405	321/405	312/405

Cuadro 4.2: Resultados del centro de áreas múltiple con sector central de 1 grado

- Proporción del sector de avance elegido: 0,15
- Configuración 4
 - Proporción de los subsectores centrales: 0,07
 - Proporción de los subsectores derecho e izquierdo: sin límite
 - Proporción del sector de avance elegido: sin límite

Para evaluar la proporción de mapa que ha sido explorado por el robot en cada una de las pruebas realizadas se ha dividido cada mapa en rejillas de ocupación, a cuántas más rejillas de un mapa alcancen los sensores del robot querrá decir que ha explorado más el mapa que otra prueba en la que hayan llegado a menos rejillas del mismo mapa.

En las tablas 4.1, 4.2, 4.3 y 4.4 se indica la proporción $\text{rejillasAlcanzadas}/\text{rejillasMapa}$ por prueba.

Al ver las tablas se observa que el parámetro que más influye en los resultados es, sin duda, la amplitud del sector central (y por extensión de los sectores laterales derecho e izquierdo).

	Sector central de 61 grados			
	Configuración 1	Configuración 2	Configuración 3	Configuración 4
Mapa 1	62/63	63/63	60/63	60/63
Mapa 2	52/60	54/60	53/60	47/60
Mapa 3	74/180	85/180	78/180	68/180
Mapa 4	68/102	75/102	71/102	65/102
Total	216/405	277/405	262/405	240/405

Cuadro 4.3: Resultados del centro de áreas múltiple con sector central de 61 grados

	Sector central de 121 grados			
	Configuración 1	Configuración 2	Configuración 3	Configuración 4
Mapa 1	45/63	49/63	47/63	41/63
Mapa 2	54/60	54/60	48/60	40/60
Mapa 3	80/180	114/180	88/180	78/180
Mapa 4	73/102	82/102	64/102	60/102
Total	252/405	299/405	247/405	219/405

Cuadro 4.4: Resultados del centro de áreas múltiple con sector central de 121 grados

Variando el resto de parámetros se obtienen pequeñas variaciones en los resultados (con alguna excepción), pero el cambio más significativo se produce al variar este parámetro. Con estas tablas se comprueba que la amplitud del sector central que mejor resultados ha dado para el centro de áreas múltiple ha sido con una amplitud del sector central de 1 grado. Dentro de las pruebas realizadas con este sector central, la configuración con mayor porcentaje de exploración de media ha sido la configuración 2 en la que el método de navegación utiliza proporción de los subsectores centrales 0,15; proporción de los subsectores derecho e izquierdo 0,15 y proporción del sector de avance elegido 0,07.

En el mapa 1 (ver imagen 4.2) si el robot usa el método de navegación básico vemos cómo es capaz de observar parte del mundo, pero al ver el comportamiento que sigue el robot destaca que para la exploración sólo vaya por dos caminos repetidamente. La justificación a este comportamiento es que el algoritmo del centro de áreas básico fue pensado para esquivar los obstáculos que un robot se encuentra a su paso, y no tiene capacidad explorativa.

Si en vez de usar el algoritmo de centro de áreas básico se usa el nuevo método de navegación, se comprueba que los resultados son muy buenos, ya que el robot ha sido capaz de conocer todo el mapa en el tiempo dado, tanto cuando el sector central es de 1 grado como cuando es de 61 grados. En cambio, al modificar el sector central a 121 grados se observa un cambio brusco en el comportamiento del robot y nada positivo, ya que ha sido incapaz de seguir otro pasillo diferente del que había partido en un inicio, haciendo que aparezcan bastantes zonas sin que el robot tenga conocimiento de ellas al acabar el experimento.

En el mapa 2 (ver imagen 4.3) si el robot usa el método de navegación básico, se observa más o menos el mismo comportamiento que en el anterior mundo para el mismo método. El robot en líneas generales realiza un movimiento en línea recta y realiza muy pocos giros, dando como resultado un mapa que ha sido explorado escasamente.

Si se prueba este mapa con el algoritmo del centro de áreas múltiple los resultados mejoran ostensiblemente respecto al método de navegación básico para la mayoría de los casos.

Cabe destacar los resultados que se obtienen cuando el nuevo método de navegación se configura para que use una amplitud del sector central de 1 grado, una proporción de los subsectores centrales 0,15; proporción de los subsectores derecho e izquierdo 0,15 y proporción del sector de avance elegido 0,07. En este caso el área explorada por el robot llega al 95%, viéndose cuando se observa la trayectoria del robot el comportamiento que se pretendía buscar

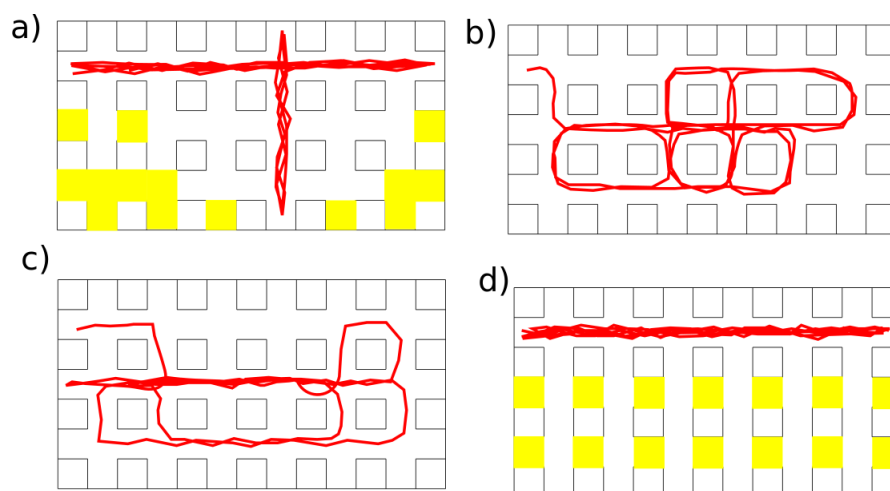


Figura 4.2: a) centro de áreas básico, b) sector central 1 grado y configuración 2, c) sector central de 61 grados y configuración 2, d) sector central de 121 grados y configuración 2. En rojo aparece la trayectoria que ha seguido el robot y en amarillo las rejillas de ocupación que se han quedado sin explorar por el robot.

con este nuevo método de navegación: que el robot fuera capaz de tomar diferentes direcciones cuando llegara a un cruce en distintos instantes de tiempo. Gracias a los diversos cambios de dirección que el robot es capaz de realizar, el mapa va siendo explorado poco a poco por el robot, llegando incluso a explorar zonas de difícil acceso (pasillo central).

Las pruebas realizadas con el tercer mapa corroboran algo que ya se vislumbraba en los mapas anteriores: que el algoritmo del centro de áreas múltiple mejora al centro de áreas básico en dos aspectos fundamentales en cuanto a la capacidad explorativa del mismo: la detección de cruces y su posterior gestión para que el robot se dirija a ellos.

Al ver la imagen 4.4 se ve que con el centro de áreas básico la mayor parte del mapa queda sin ser conocida por el robot, ya que se encuentra la mayor parte del tiempo realizando un recorrido de ida y vuelta por un mismo pasillo.

Como en anteriores situaciones cabe destacar los resultados obtenidos con el centro de áreas múltiple y un sector central de 1 grado; teniendo especial importancia esta prueba debido a la complejidad del mapa. El robot ha sido capaz de llegar o, al menos visualizar a través de sus sensores, zonas del mapa que eran de muy difícil acceso; y ha sido capaz de realizar una gran variación de movimientos y direcciones; haciendo posible estos muy buenos resultados.

Finalmente se va a realizar una comparativa del método de áreas básico con el método de áreas múltiple en un mapa en el que los cruces y bifurcaciones no son tan obvios ni es un mapa tan estructurado como son los anteriores, por lo que a priori este mapa no se adapta tan bien como podían ser los anteriores al método de áreas múltiple.

Al realizar una simulación de este mapa (ver imagen 4.5) con el método de navegación básico se comprueba que el robot se comporta de una forma bastante correcta, esquivando los obstáculos y recorriendo casi todo el mapa, aunque se sabe que la principal finalidad de este método de

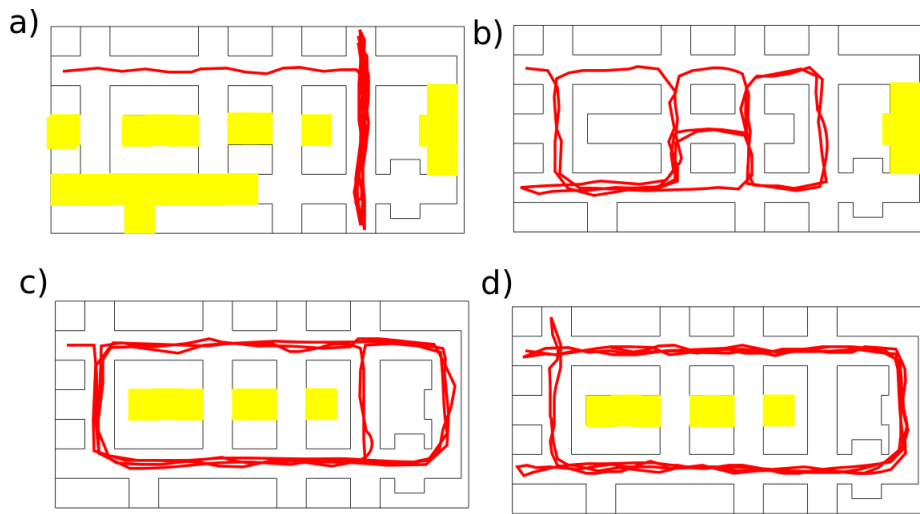


Figura 4.3: a) centro de áreas básico, b) sector central 1 grado y configuración 2, c) sector central de 61 grados y configuración 2, d) sector central de 121 grados y configuración 2. En rojo aparece la trayectoria que ha seguido el robot y en amarillo las rejillas de ocupación que se han quedado sin explorar por el robot.

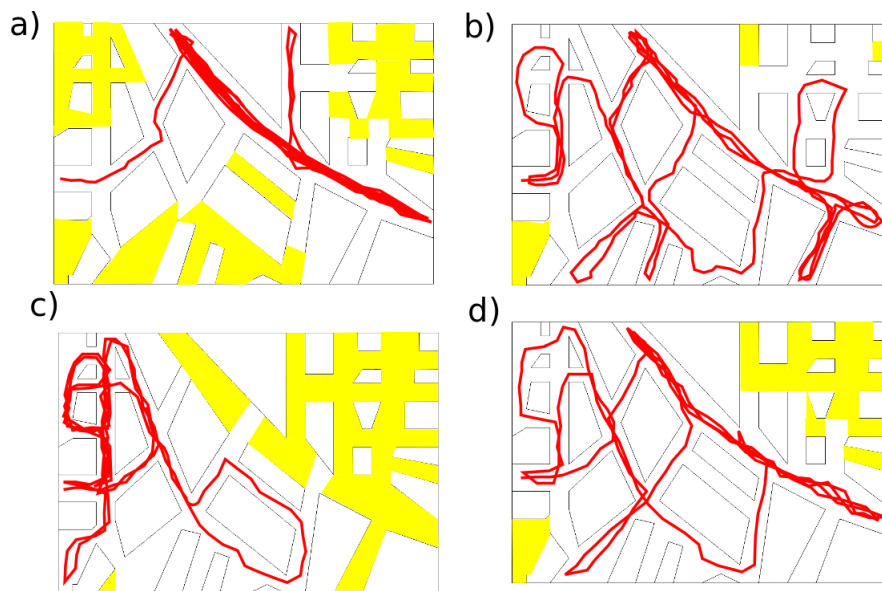


Figura 4.4: a) centro de áreas básico, b) sector central 1 grado y configuración 2, c) sector central de 61 grados y configuración 2, d) sector central de 121 grados y configuración 2. En rojo aparece la trayectoria que ha seguido el robot y en amarillo las rejillas de ocupación que se han quedado sin explorar por el robot.

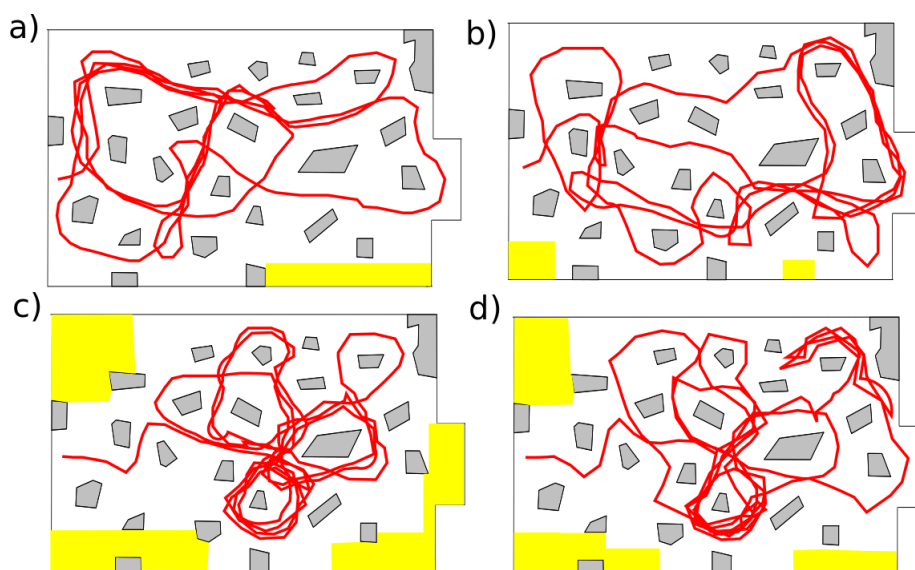


Figura 4.5: a) centro de áreas básico, b) sector central 1 grado y configuración 2, c) sector central de 61 grados y configuración 2, d) sector central de 121 grados y configuración 2. En rojo aparece la trayectoria que ha seguido el robot y en amarillo las rejillas de ocupación que se han quedado sin explorar por el robot.

navegación es saber cómo debe comportarse el robot cuando se encuentra un obstáculo delante (algoritmo del centro de áreas básico) y no realizar una exploración en profundidad del mundo.

Al experimentar con este mapa y el nuevo método de navegación con una amplitud del sector central de 1 grado se ve que el comportamiento del robot es mejor del que se esperaba en un principio, ya que es capaz de esquivar los obstáculos y al mismo tiempo ir explorando el mapa, sin dar muchos rodeos y giros inesperados.

En cambio, si se cambia el sector central de 1 grado a 61 grados se comprueba que el comportamiento del robot es algo peor ya que no es capaz de explorar tanto el mapa, debido a que el robot se queda moviéndose por la misma zona bastante rato, impidiendo el descubrimiento de nuevas zonas del mapa. Igual pasa cuando se prueba con un sector central de 121 grados.

Capítulo 5

Conclusiones y trabajos futuros

5.1. Principales aportaciones y conclusiones

- Se le ha dotado al método del centro de áreas de una capacidad explorativa que en su desarrollo inicial no tenía, gracias a la división del área libre percibida frontal en tres sectores de avance (central, derecho e izquierdo) y en el que cada uno tiene su propio centro de áreas.
- A partir de un proceso de segmentación de los datos de los sensores láser del robot, basado en filtros de Kalman, se ha conseguido descomponer el entorno percibido en áreas libres, convexas y segmentos no conectados; permitiendo al robot tomar una decisión de por qué camino continuar.
- Al basar la colocación de los split points tanto en las zonas resultantes del proceso de segmentación como en pequeños movimientos que el robot puede hacer sobre sí mismo para captar mayor información del entorno, se consigue que el robot tome mejores decisiones en cuanto a su movimiento y sea capaz de explorar mundos complejos.
- Los experimentos realizados han demostrado que cuanto más complejo es el mundo debido a que existen más caminos posibles a seguir, mayor es la mejora experimentada con el centro de áreas múltiple. Además, se ha realizado algún experimento en mundos donde los cruces o bifurcaciones no son tan obvios, sino que el mundo es más bien un mapa con obstáculos distribuidos aleatoriamente sin seguir una geometría específica, y los resultados han sido igual de satisfactorios.
- Se ha constatado que cuanto más pequeño es el sector de avance central y más grande los sectores de avance derecho e izquierdo, mayor es la capacidad explorativa del robot. La justificación de este comportamiento se basa en la situación de los centros de áreas de estos sectores de avance respecto al robot, anteponiéndose al mismo y, por tanto, anticipando posibles cruces antes de que el robot llegue a los mismos, posibilitando el giro.

5.2. Trabajos futuros

Una de las posibles extensiones a este trabajo es que para ampliar la capacidad explorativa del robot éste sea capaz de localizarse y reconocer cuándo se encuentra en un sitio por el que ha pasado previamente. Este problema es uno de los más complejos en robótica y se conoce como SLAM, que es cuando el robot necesita ir construyendo un mapa a la vez que se localiza dentro del mismo. Una de las posibilidades que se podrían plantear para el tema de la localización es llevar a cabo un análisis que compare la forma de los diferentes segmentos resultantes del proceso de segmentación con los segmentos de los procesos previos. Si este análisis da mucha semejanza entre los dos conjuntos de segmentos se podría suponer que el robot se encuentra en el mismo sitio. Como posible alternativa para escapar de la problemática del SLAM sería dotar al robot de un GPS y así conocería en todo momento sus coordenadas, posibilitando su localización.

Una vez que el robot es capaz de saber si está en un sitio por el que ha pasado previamente y que existe una bifurcación cercana, debe recordar qué acciones ha tomado en ese mismo sitio en instantes de tiempo pasado, para así no repetir la misma acción e ir por otro camino no explorado; es decir, el robot debe tener memoria. Se podría ir construyendo un grafo dirigido donde los nodos fueran zonas donde el robot ha detectado un cruce y las aristas estuvieran etiquetadas con la decisión tomada y el tiempo que ha tardado en ir entre los nodos conectados. Con esto se conseguiría una estructura muy útil para, por ejemplo, conocer cuál es el camino más corto entre dos puntos del mapa.

Otro estudio que se podría realizar es que la descomposición en los sectores derecho, izquierdo y central no sea siempre fijo en función de la amplitud del sector central, sino que se adapte al medio que se va encontrando el robot a su paso. También se podría analizar que el alcance máximo de los sectores derecho e izquierdo sea mayor que el del sector central, ya que los seres humanos cuando miramos a ambos lados de una calle alzamos mucho más la vista que cuando vamos pendiente de lo que tenemos justo delante para no chocar contra ello.

Finalmente se podría extender el método del centro de áreas múltiple a 3D, dotando al robot de un sistema de visión y realizando el mismo procesamiento pero sobre la imagen dada por este sistema de visión. Una vez segmentada la información el razonamiento a seguir sería similar al expuesto en este trabajo, teniendo en cuenta las particularidades de la problemática en 3D.

Bibliografía

Bibliografía

- Abdessemed, Foudil, Khier Benmahammed, y Eric Monacelli. 2004. A fuzzy-based reactive controller for a non-holonomic mobile robot. *Robotics and Autonomous Systems*, 47(1):31 – 46. [2.2.4](#)
- Billard, Aude y Gillian M. Hayes. 1999. Drama, a connectionist architecture for control and learning in autonomous robots. *Adapt. Behav.*, 7(1):35–63, Diciembre. [2.2.4](#)
- Borenstein, J. y Y. Koren. 1991. The vector field histogram-fast obstacle avoidance for mobile robots. *Robotics and Automation, IEEE Transactions on*, 7(3):278–288, Jun. [2.2.2](#)
- Borges, G. A. y M. J. Aldon. 2000. A split-and-merge segmentation algorithm for line extraction in 2d range images. En *Pattern Recognition, 2000. Proceedings. 15th International Conference on*, volumen 1, páginas 441–444 vol.1. [2.1.2](#)
- Borges, Geovany Araujo y Marie-José Aldon. 2004. Line extraction in 2d range images for mobile robotics. *Journal of Intelligent and Robotic Systems*, 40(3):267–297. [2.1.1](#), [2.1.2](#)
- Brock, O. y O. Khatib. 1999. High-speed navigation using the global dynamic window approach. En *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, volumen 1, páginas 341–346 vol.1. [2.2.3](#)
- Chengqing, Liu, V.M.H. Ang, H. Krishnan, y Ser Yong Lim. 2000. Virtual obstacle concept for local-minimum-recovery in potential-field based navigation. En *Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on*, volumen 2, páginas 983–988 vol.2. [2.2.1](#)
- Chohra, A., A. Farah, y C. Benmehrez. 1998. Neural navigation approach for intelligent autonomous vehicles (iav) in partially structured environments. *Applied Intelligence*, 8(3):219–233, Mayo. [2.2.4](#)
- Connolly, C.I., J.B. Burns, y R. Weiss. 1990. Path planning using laplace's equation. En *Robotics and Automation, 1990. Proceedings., 1990 IEEE International Conference on*, páginas 2102–2106 vol.3, May. [2.2.1](#)

- Cuadra, José Manuel. 2011. Cybersim. <http://www.ia.uned.es/personal/jmcuadra/cybersim/cybersim-edu/cybersim-edu-1.0.0.tar.gz>. 4
- Cuadra Troncoso, José Manuel. 2011. *Modelado adaptativo del medio para la navegación de robots autónomos utilizando algoritmos basados en el centro de áreas*. Ph.D. tesis, Universidad Nacional de Educación a Distancia. 2.3, 2.4, 3.2
- de la Paz López, Félix. 2003. *Una arquitectura que integra el modelado endógeno del medio y la navegación para un robot genérico de ruedas*. Ph.D. tesis, Universidad Nacional de Educación a Distancia. 3.2
- Dietmayer, K., J. Sparbert, y D. Streller. 2001. Model Based Object Classification and Tracking in Traffic Scenes from Range Images. En *International Conference on Information Visualisation*. 2.1.1
- Doctor, S., G.K. Venayagamoorthy, y V.G. Gudise. 2004. Optimal pso for collective robotic search applications. En *Evolutionary Computation, 2004. CEC2004. Congress on*, volumen 2, páginas 1390–1395 Vol.2, June. 2.2.4
- Duda, Richard O. y Peter E. Hart. 1972. Use of the Hough transformation to detect lines and curves in pictures. *Commun. ACM*, 15(1):11–15, Enero. 2.1.2
- Fischler, Martin A. y Robert C. Bolles. 1981. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, Junio. 2.1.2
- Forsyth, David A. y Jean Ponce. 2002. *Computer Vision: A Modern Approach*. Prentice Hall Professional Technical Reference. 2.1
- Fox, D., W. Burgard, y S. Thrun. 1997. The dynamic window approach to collision avoidance. *Robotics Automation Magazine, IEEE*, 4(1):23–33, Mar. 2.2.3
- Fox, Dieter, W. Burgard, Sebastian Thrun, y A. B. Cremers. 1998. A hybrid collision avoidance method for mobile robots. En *Proc. of the IEEE International Conference on Robotics and Automation '98*. 2.2.3
- Ge, S.S. y Y.J. Cui. 2000. New potential functions for mobile robot path planning. *Robotics and Automation, IEEE Transactions on*, 16(5):615–620, Oct. 2.2.1
- Hogan, N y JR Andrews. 1983. Impedance control as a framework for implementing obstacle avoidance in a manipulator. En *Control of Manufacturing Processes and Robotic Systems: Presented at the Winter Annual Meeting of the American Society of Mechanical Engineers, Boston, Massachusetts, November 13-18, 1983*, página 243. American Society of Mechanical Engineers. 2.2.1

- Im, Kwang-Young y Se young Oh. 2000. An extended virtual force field based behavioral fusion with neural networks and evolutionary programming for mobile robot navigation. En *Evolutionary Computation, 2000. Proceedings of the 2000 Congress on*, volumen 2, páginas 1238–1244 vol.2. [2.2.1](#)
- Inc., MobileRobots. 2011. Pioneer3at. <http://mobilerobots.com/ResearchRobots/P3AT.aspx>. [4.1](#)
- Keat, C. T. M., C. Pradalier, y C. Laugier. 2005. Vehicle detection and car park mapping using laser scanner. En *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, páginas 2054–2060, Aug. [2.1.1](#), [2.1.2](#)
- Khatib, O. 1985. Real-time obstacle avoidance for manipulators and mobile robots. En *Robotics and Automation. Proceedings. 1985 IEEE International Conference on*, volumen 2, páginas 500–505, Mar. [2.2.1](#)
- Khosla, P. y R. Volpe. 1988. Superquadric artificial potentials for obstacle avoidance and approach. En *Robotics and Automation, 1988. Proceedings., 1988 IEEE International Conference on*, páginas 1778–1784 vol.3, Apr. [2.2.1](#)
- Kim, J. y P. Khosla. 1991. Real-time obstacle avoidance using harmonic potential functions. En *Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on*, páginas 790–796 vol.1, Apr. [2.2.1](#)
- Koren, Y. y J. Borenstein. 1991. Potential field methods and their inherent limitations for mobile robot navigation. En *Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on*, páginas 1398–1404 vol.2, Apr. [2.2.1](#)
- Krogh, B. H. 1984. A generalized potential field approach to obstacle avoidance control. *International Journal of Robotic Research*. [2.2.1](#)
- Lindeberg, T. 1990. Scale-space for discrete signals. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(3):234–254, Mar. [2.1.1](#)
- Lindeberg, Tony. 1993. Discrete derivative approximations with scale-space properties: A basis for low-level feature extraction. *Journal of Mathematical Imaging and Vision*, 3(4):349–376. [2.1.1](#)
- Lindeberg, Tony, 1994. *Shape in Picture: Mathematical Description of Shape in Grey-level Images*, capítulo Scale-Space for N-Dimensional Discrete Signals, páginas 571–590. Springer Berlin Heidelberg, Berlin, Heidelberg. [2.1.1](#)
- Martínez, V.F.M. 1997. *Planificación de trayectorias para robots móviles*. Departamento de Ingeniería de Sistemas y Automática de la Universidad. [2](#)

- Martinez-Cantin, R., J. A. Castellanos, J. D. Tardos, y J. M. M. Montiel. 2006. Adaptive scale robust segmentation for 2d laser scanner. En *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, páginas 796–801, Oct. [2.1.2](#)
- Masehian, E. y D. Sedighizadeh. 2010. Multi-Objective PSO- and NPSO-based Algorithms for Robot Path Planning. *Advances in Electrical and Computer Engineering*, 10(4):69–76. [2.2.4](#)
- Mastrogiovanni, F., A. Sgorbissa, y R. Zaccaria. 2005. On the tips of one’s toes: self-localization in a dynamic environment. En *2005 International Symposium on Computational Intelligence in Robotics and Automation*, páginas 341–346, June. [2.1.1](#), [2.1.2](#)
- Moravec, H.P. y A. Elfes. 1985. High resolution maps from wide angle sonar. En *Robotics and Automation. Proceedings. 1985 IEEE International Conference on*, volumen 2, páginas 116–121, Mar. [2.2.1](#)
- Nguyen, V., A. Martinelli, N. Tomatis, y R. Siegwart. 2005. A comparison of line extraction algorithms using 2d laser rangefinder for indoor mobile robotics. En *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, páginas 1929–1934, Aug. [2.1.1](#), [2.1.2](#)
- Norouzi, M., M. Yaghoobi, M. R. Siboni, y M. Jadaliha. 2009. Recursive line extraction algorithm from 2d laser scanner applied to navigation a mobile robot. En *Robotics and Biomimetics, 2008. ROBIO 2008. IEEE International Conference on*, páginas 2127–2132, Feb. [2.1.2](#)
- Nourani-vatani, Navid, Michael Bosse, Jonathan Roberts, y Matthew Dunbabin. 2006. Practical path planning and obstacle avoidance for autonomous mowing. En *In Proc. of the Australasian Conference of Robotics and Automation*. [2.2.1](#)
- Ogren, P. y N.E. Leonard. 2005. A convergent dynamic window approach to obstacle avoidance. *Robotics, IEEE Transactions on*, 21(2):188–195, April. [2.2.3](#)
- Olson, Clark F. 1997. An approximation algorithm for least median of squares regression. *Information Processing Letters*, 63(5):237 – 241. [2.1.2](#)
- Pauwels, E. J., L. J. van Gool, P. Fiddelaers, y T. Moons. 1995. An extended class of scale-invariant and recursive scale space filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(7):691–701, Jul. [2.1.1](#)
- Pradhan, Saroj Kumar, Dayal Ramakrushna Parhi, y Anup Kumar Panda. 2009. Fuzzy logic techniques for navigation of several mobile robots. *Applied Soft Computing*, 9(1):290 – 304. [2.2.4](#)
- Raudonis, V. y R. Maskeliunas. 2011. Trajectory based fuzzy controller for indoor navigation. En *Computational Intelligence and Informatics (CINTI), 2011 IEEE 12th International Symposium on*, páginas 69–72, Nov. [2.2.4](#)

- Rimon, E. y D.E. Koditschek. 1992. Exact robot navigation using artificial potential functions. *Robotics and Automation, IEEE Transactions on*, 8(5):501–518, Oct. [2.2.1](#)
- Rousseeuw, P. J. y A. M. Leroy. 1987. *Robust Regression and Outlier Detection*. John Wiley & Sons, Inc., New York, NY, USA. [2.1.2](#)
- Sack, Daniel y Wolfram Burgard. 2004. A comparison of methods for line extraction from range data. En *In Proc. of the 5th IFAC Symposium on Intelligent Autonomous Vehicles (IAV)*. [2.1.2](#)
- Santos, Sérgio, JE Faria, F Soares, R Araujo, y U Nunes. 2003. Tracking of multi-obstacles with laser range data for autonomous vehicles. *Proc. 3rd National Festival of Robotics Scientific Meeting (ROBOTICA)*, páginas 59–65. [2.1.1](#)
- Siadat, Ali, Axel Kaske, Siegfried KLAUSMANN, Michel DUFAUT, y RenÅ© HUSSON. 1997. An optimized segmentation method for a 2d laser-scanner applied to mobile robot navigation. [2.1.2](#)
- Silva, C., M. Crisostomo, y B. Ribeiro. 2000. Monoda: a neural modular architecture for obstacle avoidance without knowledge of the environment. En *Neural Networks, 2000. IJCNN 2000, Proceedings of the IEEE-INNS-ENNS International Joint Conference on*, volumen 6, páginas 334–339 vol.6. [2.2.4](#)
- Smith, Venayagamoorthy, y Holloway. 2006. Obstacle avoidance in collective robotic search using particle swarm optimization. En *IEEE Swarm Intelligence Symposium*, May. [2.2.4](#)
- Thrun, S., A. Bücken, W. Burgard, D. Fox, T. Fröhlinghaus, D. Henning, T. Hofmann, M. Krell, y T. Schmidt. 1998. Map learning and high-speed navigation in RHINO. En D. Kortenkamp R.P. Bonasso, y R. Murphy, editores, *AI-based Mobile Robots: Case Studies of Successful Robot Systems*. MIT Press. [2.2.3](#)
- Tuncer, Adem y Mehmet Yildirim. 2012. Dynamic path planning of mobile robots with improved genetic algorithm. *Computers & Electrical Engineering*, 38(6):1564 – 1572. [2.2.4](#)
- Ulrich, I. y J. Borenstein. 1998. Vfh+: reliable obstacle avoidance for fast mobile robots. En *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, volumen 2, páginas 1572–1577 vol.2, May. [2.2.2](#)
- Ulrich, I. y J. Borenstein. 2000. Vfh*: local obstacle avoidance with look-ahead verification. En *Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on*, volumen 3, páginas 2505–2511 vol.3. [2.2.2](#)
- Weickert, Joachim, Seiji Ishikawa, y Atsushi Imiya, 1997. *Gaussian Scale-Space Theory*, capítulo On the History of Gaussian Scale-Space Axiomatics, páginas 45–59. Springer Netherlands, Dordrecht. [2.1.1](#)

- Witkin, Andrew P. 1983. Scale-space filtering. En *Proceedings of the Eighth International Joint Conference on Artificial Intelligence - Volume 2, IJCAI'83*, páginas 1019–1022, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc. [2.1.1](#)
- Xu, Zezhong, Jilin Liu, Zhiyu Xiang, y Han Li. 2002. Map building for indoor environment with laser range scanner. En *Intelligent Transportation Systems, 2002. Proceedings. The IEEE 5th International Conference on*, páginas 136–140. [2.1.1](#)
- Yang, Shiqiang, Weiping Fu, Dexin Li, y Wen Wang. 2007. Research on application of genetic algorithm for intelligent mobile robot navigation based on dynamic approach. En *Automation and Logistics, 2007 IEEE International Conference on*, páginas 898–902, Aug. [2.2.4](#)
- Yun, Soh Chin, S. Parasuraman, y V. Ganapathy. 2011. Dynamic path planning algorithm in mobile robot navigation. En *Industrial Electronics and Applications (ISIEA), 2011 IEEE Symposium on*, páginas 364–369, Sept. [2.2.4](#)
- Zou, Xi-yong y Jing Zhu. 2003. Virtual local target method for avoiding local minimum in potential field based robot navigation. *Journal of Zhejiang University SCIENCE A*, 4(3):264–269. [2.2.1](#)