

## Generación de Ruta Óptima para Robots Móviles a Partir de Segmentación de Imágenes

Holman Montiel, Edwar Jacinto y Fredy H. Martínez\*

Universidad Distrital Francisco José de Caldas, Facultad Tecnológica, Carrera 7 #40-53, Bogotá D.C.- Colombia (e-mail: hmontiela@udistrital.edu.co; ejacintog@udistrital.edu.co; fhmartinezs@udistrital.edu.co)

\* autor a quien debe ser dirigida la correspondencia

*Recibido Feb. 14, 2013; Aceptado May. 12, 2014; Versión final recibida Oct. 28, 2014*

---

### Resumen

Este artículo propone una estrategia geométrica para la generación de rutas óptimas de navegación para robots en ambientes estáticos y observables. Como principios de diseño del algoritmo se establecieron la robustez, la simplicidad y el bajo costo computacional. El algoritmo utiliza la segmentación de imágenes en regiones como arquitectura de soporte. Estas regiones son construidas a partir de los vértices de los obstáculos, los cuales definen dos conjuntos independientes de regiones. A partir de los límites entre regiones se construye un conjunto de puntos seguros de navegación, puntos que son la base para la generación de un banco de rutas seguras. Finalmente, un algoritmo de búsqueda selecciona de éste banco la ruta que seguirá el robot. El algoritmo se evaluó exitosamente por simulación sobre más de 30 ambientes, encontrando una ruta adecuada en el 100% de los casos.

*Palabras clave: robótica, planeación de rutas, navegación, esquema híbrido, segmentación*

## Optimal Path Generation for Mobile Robots from Cell Image Segmentation

### Abstract

This paper proposes a geometric strategy for generating optimal navigation paths for robots in static and observable environments. As design principles for the algorithm, robustness, simplicity and low computational cost were established. The algorithm uses image segmentation into regions as architecture support. These regions are constructed from the vertices of the obstacles, which define two independent sets of regions. From the boundaries between regions we build a set of secure navigation points, points that are the basis for generating a bank of secure paths. Finally, a search algorithm selects, from this bank, the path that the robot will follow. The algorithm has been successfully evaluated by simulation on over 30 environments, finding a suitable path in 100% of the cases.

*Keywords: robotics, path planning, navigation, hybrid scheme, segmentation*

## INTRODUCCIÓN

La planificación de la trayectoria de un robot móvil, o navegación, es una de las aplicaciones más importantes de los sistemas de control, y debido a sus implicaciones en aplicaciones reales, y a que se trata de un problema NP-duro (NP-completo), ha atraído la notable atención de un gran número de investigadores (LaValle, 2011, Egerstedt y Jensfelt, 2008, Moran et al., 2011, Vasile y Belta, 2013). El problema consiste en definir una ruta de navegación para un robot móvil en un ambiente observable, que pueda ser interpretada y ejecutada por el robot a lo largo de una secuencia de movimientos. El desplazamiento se realiza con movimientos seguros tanto para el robot como para el ambiente, evitando colisiones, con el objetivo de desplazarse desde un punto inicial  $p_i$  hasta un punto de destino o final  $p_f$  (LaValle, 2011). El diseño de los algoritmos contempla dos elementos centrales: la forma en que el ambiente es sensado, y si el ambiente es estático o dinámico. En el primer caso, se debe considerar si el ambiente ya se conoce como consecuencia de un sensado previo, o si es desconocido (Gonzalez et al., 2014). Además, se debe definir el nivel de autonomía del robot. En éste punto, se debe analizar el diseño del robot y sus capacidades, entre otras cosas para definir si el robot utiliza sensores y procesadores a bordo, o se cuenta con una unidad central de control (Belkhouche, 2009).

En el segundo caso, en cuanto a si el ambiente es estático o dinámico, el diseño del algoritmo puede considerar gran procesamiento de la información de los sensores (lo que es costoso en tiempo) en el caso de ambientes estáticos, o priorizar la velocidad de respuesta en ambientes dinámicos, lo que permite combinar algoritmos geométricos simples con alguna lectura del ambiente (Font y Batlle, 2006, Garrido et al., 2009). Los algoritmos tradicionales basados en la geometría del ambiente son ampliamente utilizados en ambientes estáticos (Xu y Shing, 2010, Cai y Ferrari, 2009, Bhattacharya y Gavrilova, 2008), mientras que los ambientes dinámicos prefieren el uso de estrategias reactivas (Egerstedt y Jensfelt, 2008, Paull et al., 2013, López y Gomez-Bravo, 2012). Finalmente, otro aspecto importante a considerar es la arquitectura hardware del robot, por ejemplo, el número de ejes y ruedas del robot, y cómo debe planearse el movimiento de cada una de ellas en el espacio libre (López y Gomez-Bravo, 2012, Vale et al., 2014).

Este artículo presenta una estrategia geométrica híbrida, soportada en la segmentación de imágenes en celdas, para la planeación global de una ruta en ambientes estáticos. Como tal, la estrategia se soporta en los principios de los grafos de visibilidad, es decir, un grafo construido a partir de lugares intervisibles (Latombe, 1991), pero desde la perspectiva de puntos seguros de navegación en el espacio libre del ambiente, puntos que se definen geoméricamente a partir de descomposiciones trapezoidales (Latombe, 1991). A partir de la segmentación se define una matriz general de trayectos, los cuales luego son evaluados para la selección de la ruta sub-óptima/óptima de forma eficiente.

El esquema contempla una unidad de control central para el procesamiento de la información, comunicación con el robot y sensado del ambiente. La utilización de una única cámara como sensor global del ambiente es una alternativa eficiente y robusta, que elimina además el problema de la óptima localización de sensores, y la complejidad inherente a la combinación de datos de cada sensor (Krause et al., 2008), y va de la mano con el criterio de localización de sensores basado en alta entropía (Paull et al., 2013).

## FORMULACIÓN DEL PROBLEMA Y SUPUESTOS

El objetivo de la estrategia de planeación de ruta global, es definir un camino óptimo y seguro para el desplazamiento de un robot móvil a lo largo del espacio libre de obstáculos en el ambiente de navegación, desde un punto inicial  $p_i$  hasta un punto destino  $p_f$ . Esta estrategia debe diseñar la ruta a partir de las medidas del sensor (información contenida en el ambiente).

Un robot (no un punto, con un área definida en  $\mathbb{R}^2$ ) es colocado en un punto  $p_i$  dentro de un ambiente de navegación  $W \subset \mathbb{R}^2$  plano, conectado y compacto. Se define a  $\mathcal{O}$  como el conjunto de obstáculos en el ambiente, en el que cada obstáculo  $O \in \mathcal{O}$  es cerrado con un límite (frontera) conectado y analítico a pedazos, y de longitud finita. Además, los obstáculos en  $\mathcal{O}$  son pares disjuntos (mutuamente disjuntos) entre sí (no comparten puntos comunes) y finitos en número (Figura 1(a)). El límite o frontera de  $W$  se denota por  $\partial W$ , y los límites de cada obstáculo se consideran parte de  $\partial W$ . El espacio libre en el cual puede navegar libremente el robot se denota por  $E$ , quien también está contenido en  $\mathbb{R}^2$ , y se define como  $W - \mathcal{O}$ .

Como es común en muchos esquemas (Fainekos, 2011, Bhatia et al., 2010, Kloetzer y Belta, 2010), el ambiente se divide en un conjunto finito de regiones conectadas, sobre las cuales se define un sistema de transición discreto (Figuras 1(b) y 1(c)). A lo largo de los límites entre regiones se localizan posibles puntos de tránsito (puntos en azul), que son seleccionados, o no, como punto de la ruta de navegación por parte del algoritmo de planeación.

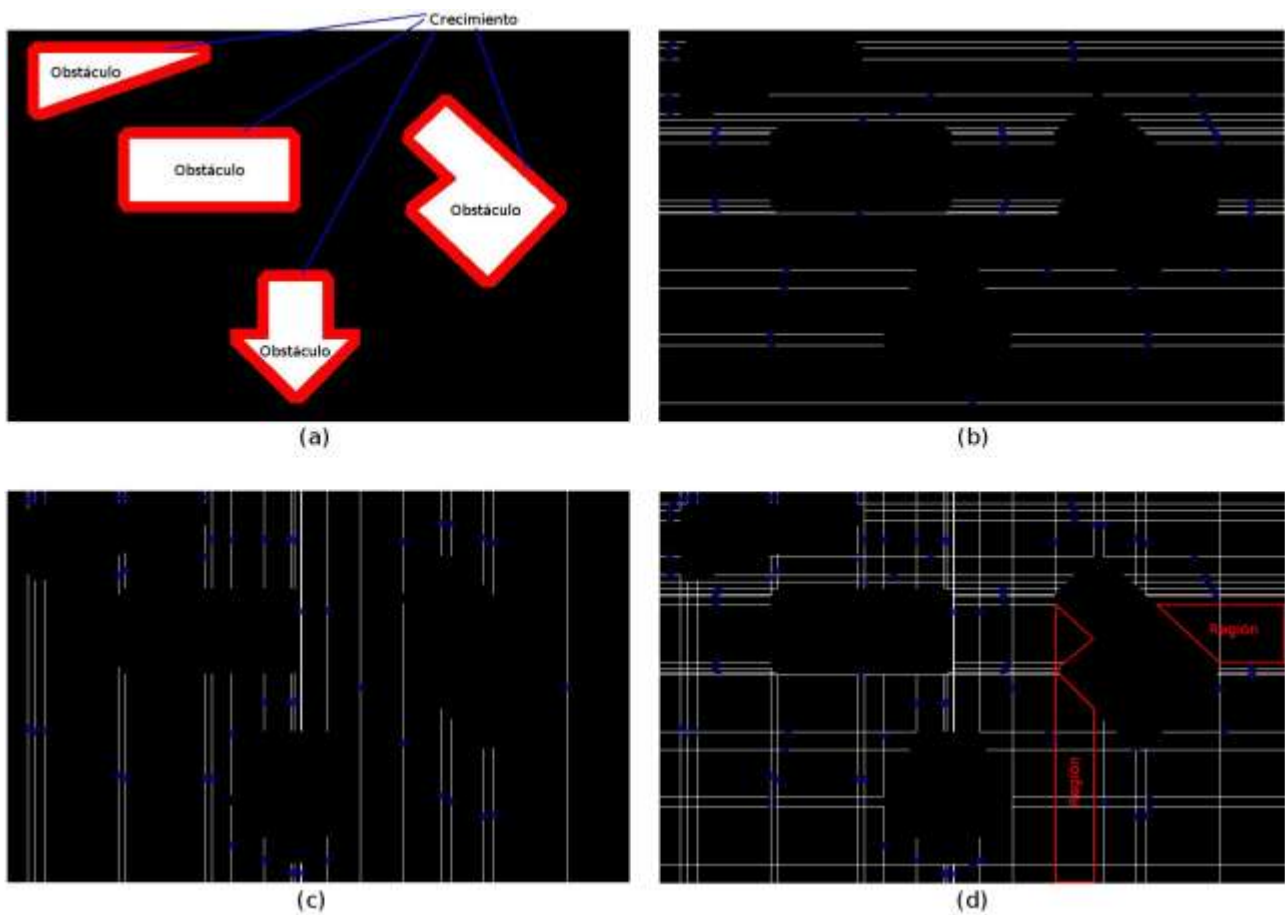


Fig. 1: Ambiente de navegación con detalle de formulación. (a) Obstáculos (regiones internas en blanco) y crecimiento (borde rojo) en un ambiente de navegación  $W$ . (b) Límites entre regiones  $\gamma$  conformados por segmentos horizontales. (c) Límites entre regiones  $\gamma$  conformados por segmentos verticales. (d) Límites horizontales y verticales con detalle de dos regiones (en rojo), una con límites horizontales y una con límites verticales.

Las regiones o celdas que descomponen a  $W$ , son generadas o inducidas por los obstáculos en  $W$ . Cada límite entre regiones se denota por  $\gamma$ , y al conjunto de todos estos límites entre regiones se denota por  $\Gamma$ . Cada  $\gamma_i \in \Gamma$  es la imagen de una curva inyectiva y rectificable definida entre límites de obstáculos y/o el límite del ambiente  $\partial W$ . Si los  $\gamma_i \in \Gamma$  son pares disjuntos entre sí, entonces cada  $\gamma_i$  separa dos regiones, y cada región está acotada por dos límites  $\gamma_i$  e intervalos de  $\partial W$  (Figura 1(d)). Las regiones se denotan por  $r$ , y al conjunto de todas las regiones en el ambiente se denota por  $\mathcal{R}$ .

El robot se considera pequeño con respecto a  $W$  y a las regiones  $r \in \mathcal{R}$ . Se considera también que el tamaño del robot está limitado en  $\mathbb{R}^2$  por un círculo de diámetro  $d$ . Éste robot debe desplazarse desde  $p_i$  hasta un punto de destino final  $p_f$  navegando a lo largo de  $E$ , es decir, esquivando los obstáculos. Además, se desea que la ruta seleccionada sea lo más corta posible, y no presente ángulos de giro pequeños que dificulten la maniobrabilidad (segura para el robot). El algoritmo de navegación debe planear las rutas de navegación de tal forma que se seleccionen los puntos centrales de los límites entre regiones  $\gamma_i$ , siempre y cuando las distancias y la geometría permitan el paso del robot. El ambiente será completamente observable por parte de la unidad central de control, la cual se comunica con el robot para transmitir la ruta de navegación.

## METODOLOGÍA Y DISEÑO

El algoritmo propuesto se basa en la segmentación del ambiente de navegación  $W$  en regiones  $r \in \mathcal{R}$ , y en el análisis de éstas regiones para la definición de la ruta. El trabajo se soporta en gran parte en procesamiento digital de imágenes que contienen la información del ambiente. Estas imágenes son capturadas por una única cámara instalada sobre el ambiente de navegación, que sirve de sensor para la unidad central de control.

Las imágenes se someten inicialmente a un proceso básico de identificación de obstáculos, y del espacio libre de navegación  $E$ . Cada imagen se convierte a escala de grises, y luego a binaria (blanco y negro) con

umbrales adecuados al nivel de iluminación. Una vez diferenciados claramente los obstáculos del fondo, se realiza un proceso de dilatación o crecimiento de los obstáculos con el fin de identificar el área segura de navegación para el robot. Éste proceso de dilatación aumenta el tamaño de los obstáculos para procesamiento en  $d/2$ , esto garantiza que el robot no se estrellará con los obstáculos a pesar de que la ruta se calcule para un objeto puntual (Figura 1(a)).

La definición de las regiones se realiza de acuerdo a los vértices identificados en cada uno de los obstáculos  $O \in \mathcal{O}$ . Cada vértice produce límites de regiones  $\gamma_i \in \Gamma$  tanto horizontales (Figura 1(b)) como verticales (Figura 1(c)). Los  $\gamma_i$  horizontales en conjunto con los bordes de los obstáculos (que hacen parte de  $\partial W$ ) producen un primer conjunto de regiones; lo mismo ocurre con los  $\gamma_i$  verticales en conjunto con los obstáculos, los cuales producen un segundo conjunto de regiones. Esto significa que la imagen posee una doble segmentación en regiones (Figura 1(d)). Esta doble segmentación es independiente la una de la otra, dado que ambas se realizan a partir de la misma imagen (una segmentación produce regiones horizontales, y la otra regiones verticales). La importancia de estas regiones es que cada curva  $\gamma$  permite definir su punto medio como un punto seguro de navegación (puntos azules en las Figuras 1(b), 1(c) y 1(d)). Las coordenadas de estos puntos son almacenadas en una matriz, de la cual se deben seleccionar los puntos que conforman una ruta óptima de navegación.

La segunda parte del proceso consiste en la selección de la ruta. El primer paso para ello consiste en definir todas las posibles rutas, y entre ellas seleccionar la mejor. Con los puntos medios de todas las curvas  $\gamma_i$  (Figura 2(a)) más los puntos  $p_i$  y  $p_t$ , se trazan todas las posibles rectas que unen dichos puntos (uno contra todos), eliminando aquellas que se superponen con los obstáculos dilatados (Figura 2(b)). Se calcula luego todas las distancias euclidianas de cada una de estas rectas.

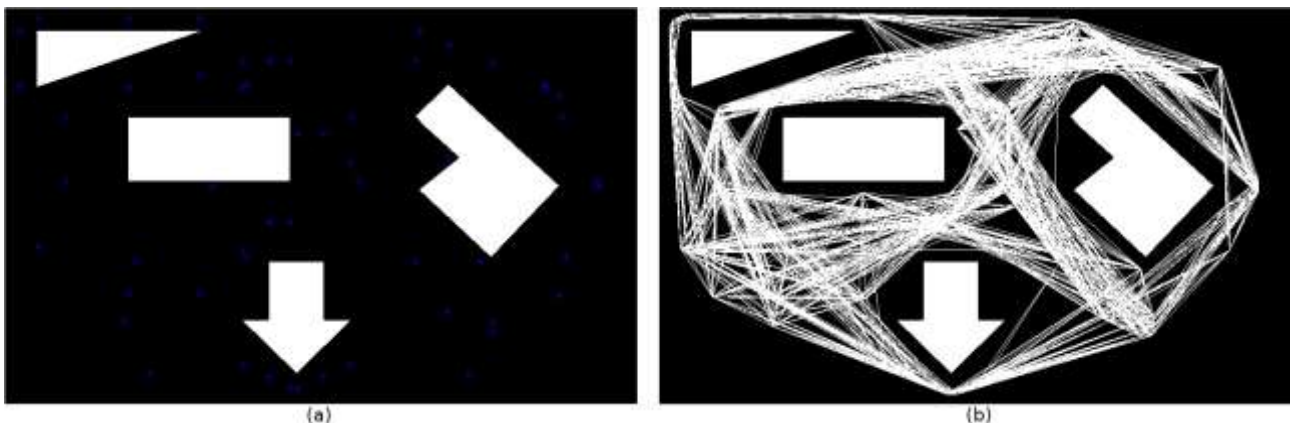


Fig. 2: Ambiente de navegación con detalle de diseño. (a) Puntos medios de las curvas  $\gamma$  generadas a partir de los vértices geométricos de los obstáculos. (b) Todos los posibles caminos de un punto a otro que no se cruzan con obstáculos dilatados.

Un primer criterio de selección de ruta, es el de seleccionar las distancias más cortas entre dos puntos. Si bien esta estrategia no tiende a seleccionar la ruta más corta, si ayuda al robot a bordear de forma segura los obstáculos. La Figura 3(a) muestra la aplicación de éste criterio a las rectas definidas en la Figura 2(b). La ruta desde  $p_i$  hasta  $p_t$  se determinó con estas rectas utilizando un simple algoritmo de primero en profundidad (Depth-First Search - DFS) (Tarjan, 1971, Hanzhou y Hongxia, 2013). Éste algoritmo, ampliamente utilizado para la detección de componentes conexas, permite una muy rápida selección de ruta con un muy bajo costo computacional. Dado que el algoritmo propuesto de generación de rutas permite la obtención de un conjunto optimizado de rutas, la selección final puede realizarse sin mayor costo por medio de cualquier algoritmo de búsqueda. Si se asume como peso de cada recta su distancia euclidianas, es posible utilizar el algoritmo Dijkstra o A\* para encontrar la ruta más corta. También es posible la implementación de búsqueda bi-direccional o incluso de búsquedas aleatorizadas. Finalmente, la selección del algoritmo depende de la escala y el tipo de problema, y en el caso aquí propuesto, un algoritmo minimalista de rápida respuesta y bajo costo computacional resulta ser la opción más acertada.

Este criterio de distancias cortas es bastante eficiente computacionalmente, y permite bordear con facilidad los obstáculos, pero no siempre encuentra las rutas más cortas, en particular cuando existen muchos obstáculos. A fin de mejorar la selección de rutas cortas, a las rectas definidas en la Figura 3(a) se adicionó un conjunto de rectas conformadas por las distancias más largas entre dos puntos de la matriz, bajo el concepto de que ellas reducen las rutas en espacios libres de obstáculos. Sin embargo, solamente se toman las rectas del 20% de las distancias más cortas entre dos puntos que no se traslapan con los

obstáculos dilatados (Figura 3(c)). Durante las pruebas se observó que incluir un mayor porcentaje de rectas no aporta información relevante (ni para trayectos abiertos, ni para reducción de distancia) y en cambio sí incrementa el costo computacional. Esta estrategia define un nuevo conjunto de rectas, que al sumarse a las rectas de la Figura 3(a), no solo cierra los trayectos abiertos, sino que agrega rutas directas que en algunos casos reduce la distancia de la ruta. El siguiente pseudocódigo presenta una descripción más detallada del método de planificación de forma global.

#### Algoritmo propuesto para la planeación de la ruta

```

1  Definición posición final (usuario);
2  Inicializar cámara digital;
3  Identificación de posición inicial mediante procesamiento digital de la imagen (PDI):
   Captura de la imagen, filtrado e identificación;
4  (PDI) Identificación de obstáculos;
5  (PDI) Crecimiento de obstáculos;
6  (PDI) Segmentación horizontal y definición de puntos medios (puntos de navegación);
7  (PDI) Segmentación vertical y definición de puntos medios (puntos de navegación);

# Algoritmo de selección de ruta
8  begin
9      Definición matriz puntos de transito (puntos de navegación +  $p_i$  +  $p_i$ );
10     Definición de rectas entre puntos – rectas que se cruzan con obstáculos;
11     Calcular la distancia euclidiana de cada recta;
12     # Matriz de trayectos
13     while Robot no esta en  $p_i$ ;
14         for  $i = 1$  hasta recorrer matriz puntos de transito;
15             Calcular punto más cercano a  $p_i$  y agregar la recta que los une
               a matriz de trayectos;
16             Calcular 20% de trayectos adicionales ordenados de menor a
               mayor longitud, y agregarlos a la matriz de trayectos;
17         end
18     Algoritmo primero en profundidad para selección de trayectos que
               conforman la ruta;
19     Establecer ruta y transmitirla a actuador;
20 end

```

#### ANÁLISIS DE RESULTADOS

La cámara utilizada es la Bosch NBC-265-W, cámara HD (720p, sensor CMOS 1280x800 pixeles) IP inalámbrica, con capacidad de captura de hasta 30 fps, con lente varifocal de 2.8 a 12 mm (es decir, alcance de 0 a 35 m). Como plataforma robótica se utilizó el Single-Board RIO (sbRIO) de National Instruments (32 cm de frente por 22 cm de ancho y 14 cm de alto), con procesador y FPGA para procesamiento embebido. La cámara se localizó a una altura de 7.4 m sobre la plataforma, lo que permitió la captura de todo el ambiente (áreas de trabajo desde 5 m<sup>2</sup> hasta 64 m<sup>2</sup>), y la clara identificación del robot y los obstáculos en las imágenes en todas las pruebas.

La estrategia de navegación se probó con un total de 30 escenarios, 20 de ellos capturados en laboratorio de prototipos reales, y 10 generados por software. La mayoría de los ambientes contempló obstáculos geométricos, los cuales se definieron como objetivo en el perfil inicial. Sin embargo, las pruebas de desempeño contemplaron también laberintos. La Figura 4 muestra dos casos de navegación resueltos por la estrategia para diferentes posiciones iniciales y finales del robot.

A lo largo de más de 50 pruebas diferentes (diferentes ambientes, diferentes dimensiones de ambiente, diferente complejidad de obstáculos y diferentes tamaños de robots), el algoritmo fue capaz de encontrar en el 100% de los casos una ruta de navegación adecuada entre el punto inicial y el punto final. Si bien la ruta encontrada no siempre está entre las consideradas óptimas (más cortas, Figura 4(a)), siempre trata de encontrarlas navegando entre los obstáculos del ambiente, y nunca utiliza las periferias, a menos de que no exista otra opción.

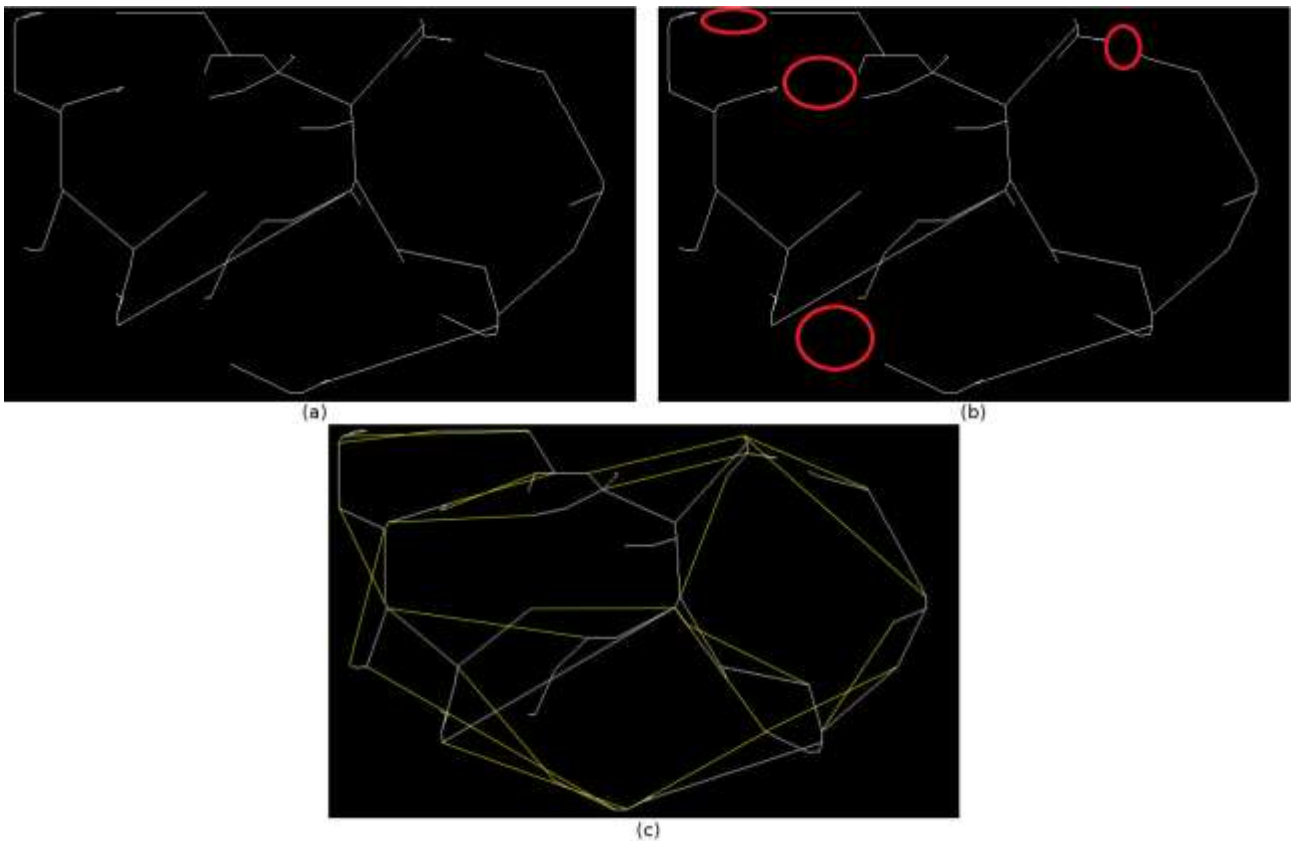


Fig. 3: Ambiente de navegación con detalle de distancias cortas. (a) Rectas formadas por las distancias más cortas de un punto a su punto más cercano. Se coloca una recta por cada punto. (b) Las rectas no cubren todos los espacios alrededor de los obstáculos, por lo que se presentan caminos abiertos en los posibles trayectos. (c) Se definen nuevas rectas (rectas amarillas) con distancias más largas entre puntos.

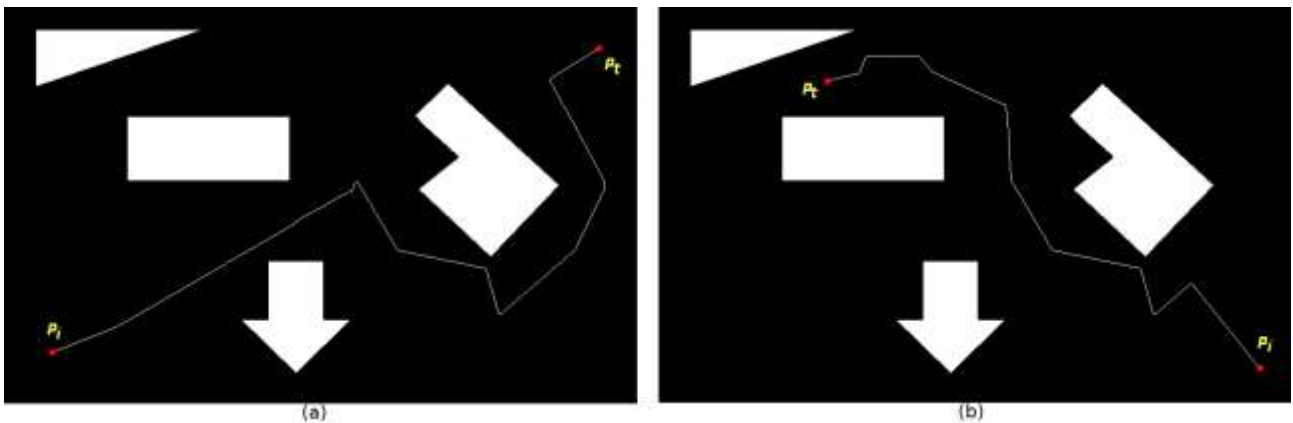


Fig. 4: Ambiente de navegación con dos ejemplos de la estrategia propuesta. (a) El punto inicial del robot  $p_i$  se localiza en la parte inferior izquierda, y el punto final  $p_f$  se localiza en la parte superior derecha. (b) El punto inicial del robot  $p_i$  se localiza en la parte inferior derecha, y el punto final  $p_f$  se localiza en la parte superior izquierda.

Frente a otros esquemas geométricos previamente evaluados en la investigación (Voronoi, Bug, Delaunay y otros tipos de segmentación), el algoritmo mantiene un bajo consumo computacional (entre 2 y 5 s, utilizando un Intel Core i7 con 8 Gigabytes de RAM corriendo Windows 7 64 bits y MatLab R2011a), y sobresale por una alta confiabilidad (siempre encuentra rutas seguras de navegación).

## CONCLUSIONES

El artículo propone un esquema de navegación para robots en ambientes estáticos y observables que se basa en estrategias geométricas de segmentación de imágenes en celdas o regiones. El algoritmo identifica los vértices extremos de los objetos, y a partir de ellos define dos conjuntos de regiones, uno con límites

horizontales y otro con límites verticales. En los límites de estas regiones, define puntos seguros de navegación que utiliza para establecer secciones de rutas de navegación. Estas secciones son seleccionadas de acuerdo a las distancias más cortas a fin de facilitar la navegación a través de los obstáculos. Distancias más largas son también seleccionadas para reducir la distancia total de navegación. La selección de la ruta se realiza mediante un algoritmo de búsqueda de primero en profundidad. Durante la evaluación de desempeño, a lo largo de muchas simulaciones, el algoritmo demuestra ser extremadamente robusto al encontrar en el 100% de los casos una ruta adecuada, aún en ambientes de gran complejidad. A diferencia de otras estrategias geométricas evaluadas previamente, como la triangulación de Delaunay, el algoritmo nunca trata de navegar por las periferias, a menos que no exista otra opción. Otra característica importante es que el costo computacional se mantiene bastante bajo.

## NOTACIÓN

|                |   |
|----------------|---|
| $\Gamma$       | Conjunto de límites entre regiones. Es un conjunto de curvas definidas en el ambiente entre obstáculos y los límites del ambiente.                |
| $\gamma$       | Límite entre regiones. Es una curva que separa dos regiones en el ambiente.   |
| $d$            | Diámetro del círculo más pequeño que contiene al robot en $\mathbb{R}^2$ .  |
| $E$            | Espacio libre en el ambiente de navegación, en el cual el robot puede desplazarse.  |
| $O$            | Obstáculo en $W$ .  |
| $\mathcal{O}$  | Conjunto de obstáculos.   |
| $p_i$          | Un punto en $W \subset \mathbb{R}^2$ .  |
| $\mathbb{R}^q$ | El $q$ -dimensional espacio Euclidiano.   |
| $\mathcal{R}$  | Conjunto de todas las regiones $r$ de $W$ .   |
| $r$            | Región en $W$ . Cada región está definida por curvas $\gamma$ e intervalos de $\partial W$ .  |
| $W$            | Conjunto que tiene un interior abierto conectado, con obstáculos que representan áreas inaccesibles. Área de navegación en el espacio Euclidiano. |
| $\partial W$   | Límite o frontera de $W$ .  |

## REFERENCIAS

- Belkhouche, F. *Reactive path planning in a dynamic environment*. IEEE Transactions on Robotics, 25(4), 902-911. (2009).
- Bhatia, A., Kavraki, L. E. y Vardi, M. Y. *Sampling-based motion planning with temporal goals*. Proceedings IEEE International Conference on Robotics & Automation, 2689-2696, (2010).
- Bhattacharya, P. y Gavrilova, M. L. *Roadmap-based path planning - using the Voronoi diagram for a clearance-based shortest path*. IEEE Robotics & Automation Magazine, 15(2), 58-66, (2008).
- Cai, C. y Ferrari, S. *Information-driven sensor path planning by approximate cell decomposition*. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 29(3), 672-689, (2009).
- Egerstedt, M. y Jensfelt, P. *A control theoretic formulation of the generalized slam problem in robotics*. American control conference, 2409-2414, (2008).
- Fainekos, G. E. *Revising temporal logic specifications for motion planning*. Proceedings IEEE International Conference on Robotics & Automation, 40-45, (2011).
- Font, J. y Batlle, J. *Posicionamiento de robots móviles mediante un filtro de Kalman angular y triangulación*. Información Tecnológica, 17(5), 9-14, (2006).
- Garrido, S., Moreno, L., Blanco, D. Martín, F. *Smooth path planning for non-holonomic robots using fast marching smooth path planning for non-holonomic robots using fast marching*. IEEE International Conference on Mechatronics ICM 2009, 1-6, (2009).
- Gonzalez, A., Ghaffarkhah, A. y Mostofi, Y. *An integrated framework for obstacle mapping with see-through capabilities using laser and wireless channel measurements*. IEEE Sensors Journal, 14(1), 25-38, (2014).
- Hanzhou, W. y Hongxia, W. *Multibit color-mapping steganography using depth-first search*. 2013 International Symposium on Biometrics and Security Technologies (ISBAST), 224-229, (2013).
- Kloetzer, M. y Belta, C. *Automatic deployment of distributed teams of robots from temporal logic motion specifications*. IEEE Transactions on Robotics and Automation, 26(1), 48-61, (2010).

- Krause, A., Singh, A. y Guestrin, C. *Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies*. The Journal of Machine Learning Research, 9(1), 235-284, (2008).
- Latombe, J. C. *Robot motion planning*. Kluwer Academic Publishers, (1991).
- LaValle, S. M. *Motion planning*. IEEE Robotics & Automation Magazine, 18(1), 79-89, (2011).
- López, D. A. y Gomez-Bravo, F. *Vodec: A fast Voronoi algorithm for car-like robot path planning in dynamic scenarios*. Robotica, 30(7), 1189-1201, (2012).
- Moran, O., Künning, F. y Cuello, J. *Implementación en el Robot Antropomorfo CXN-I del Control Cinemático de Trayectorias Rectilíneas*. Información Tecnológica, 22(1), 93-100, (2011).
- Paull, L., Saeedi, S., Seto, M. y Li, H. *Sensor-driven online coverage planning for autonomous underwater vehicles*. IEEE/ASME Transactions on Mechatronics, 18(3), 1827-1838, (2013).
- Tarjan, R. *Depth-first search and linear graph algorithms*. 12th Annual Symposium on Switching and Automata Theory, 114-121, (1971).
- Vale, A., Fonte, D., Valente, F. Ribeiro, I. *Trajectory optimization for autonomous mobile robots in ITER*. Robotics and Autonomous Systems, 62(6), 871-888, (2014).
- Vasile, C. I. y Belta, C. *Sampling-based temporal logic path planning*. 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 4817-4822, (2013).
- Xu, L. y Shing, T. W. S. *Self-organizing potential field network: A new optimization algorithm*. IEEE Transaction on Neural Networks, 21(9), 1482-1495, (2010).