

Identificación de un ambiente desconocido e implementación de un grafo de visibilidad para el cambio de configuración de un robot¹

Identification of an Unknown Environment and Implementation of a Visibility Graph to Change a Robot Configuration

Identification d'un environnement inconnu et implémentation d'un graphe de visibilité pour le changement de configuration d'un robot

Daniel Estiven Álvarez Gaviria

Estudiante de Ingeniería de Sistemas e Informática,
Universidad Nacional de Colombia
dalvarezg@unal.edu.co

Jovani Alberto Jiménez Builes

Licenciado en Docencia de Computadores
Magister en Ingeniería de Sistemas
Ph. D. en Ingeniería, pasantía pos doctoral Massachusetts,
Institute of Technology (USA) y Universidad Autónoma de
Madrid (España)
Profesor asociado, Universidad Nacional de Colombia
jajimen1@unal.edu.co

Jaime Alberto Guzmán Luna

Ingeniero Civil
Magíster en Ingeniería de Sistemas
Ph. D. en Ingeniería
Profesor asociado, Universidad Nacional de Colombia
jaguzman@unal.edu.co

Recibido: octubre 5 de 2012
Evaluado: noviembre 30 de 2012
Aprobado: diciembre 10 de 2012
Tipo de artículo: investigación científica y tecnológica

Contenido

1. Introducción
2. Materiales y métodos
3. Metodología
4. Resultados y discusión
5. Conclusiones
6. Referencias

¹ El trabajo descrito en este artículo hace parte del "Plan de acción para el fortalecimiento de los grupos de investigación Inteligencia Artificial en Educación y Diseño Mecánico Computacional" (código Hermes 14198) auspiciado por la Vicerrectoría de Investigación de la Universidad Nacional de Colombia.

Resumen

En este artículo se presenta una propuesta para la construcción de grafos de visibilidad como método de planificación de movimientos en un ambiente desconocido, mediante el uso de un sistema de sensores en un robot *Legó Mindstorm NXT*. La tarea a cumplir es la de obtener la ubicación de los diferentes vértices de los obstáculos, utilizando los datos proporcionados por el sistema de sensores. Estos datos adquiridos, tienen la finalidad de generar un grafo de visibilidad con todas las posibles trayectorias a recorrer por el robot en el ambiente identificado, el cual es representado gráficamente en un ordenador apoyándose en una comunicación vía *Bluetooth* constante entre el robot y el ordenador. Con el grafo de visibilidad generado se busca obtener la ruta óptima libre de colisiones a realizar por el robot desde una configuración inicial a una configuración final. Con relación a otras propuestas de geometría computacional el robot ya no es considerado como "un punto" sino que tiene la posibilidad de recorrer las rutas que su estructura le permite en el grafo de visibilidad evitando colisiones con el ambiente.

Palabras clave

Algoritmo de visibilidad, Grafo de visibilidad, Inteligencia artificial, Planificación de movimientos, Robótica.

Abstract

This paper presents a proposal for the construction of visibility graphs as a motion planning method in an unknown environment, using a *Legó Mindstorm NXT* robot equipped with a sensor system. The task to be accomplished is to obtain the location of the different vertex of the obstacles, using data provided by the sensor system. Acquired data serve to generate a visibility graph with all possible paths for the robot in the identified environment, which is represented graphically on a computer using a permanent *Bluetooth* communication between the robot and the computer. Having the generated visibility graph we look for the optimal collision-free path for the robot, from an initial configuration to a final configuration. Regarding other computational geometry proposals, the robot is no longer viewed as "a point", instead of that it has the possibility of traversing routes allowed by its structure, according to the visibility graph avoiding collisions with the environment.

Keywords

Artificial intelligence, Movement planning, Robotics, Visibility algorithm, Visibility graph.

Résumé

Cet article présente une proposition pour la construction de graphes de visibilité comme une méthode de planification de mouvements dans un environnement inconnu, au moyen de l'usage de un système de capteurs dans un robot *Legó Mindstorm NXT*. La tâche à réaliser est d'obtenir l'emplacement des différents sommets des obstacles, en utilisant les données trouvés par le système de capteurs. Les données acquises, servent pour générer un graphe de visibilité avec toutes les possibles trajectoires à parcourir par le robot dans un environnement identifié, lequel est représenté graphiquement sur un ordinateur en s'aidant sur une communication *Bluetooth* constant entre le robot et l'ordinateur. Avec le graphe de visibilité générée on cherche d'obtenir la route optimale qui soit libre de collisions pour être passé d'une configuration initial à une configuration finale. Par rapport à des autres propositions de géométrie computationnelle le robot n'est plus considéré comme « un point » mais il a la possibilité de parcourir les routes que son structure le permet dans le graphe de visibilité en évitant des collisions avec les objets de son environnement.

Mots-clés

Algorithme de visibilité, graphe de visibilité, intelligence artificiel, planification de mouvements, robotique.

1. Introducción

En la robótica algunos de los desafíos más significativos frente a la autonomía están en el área de planificación automática de movimientos (Choset, 2005). Se busca poder llevar instrucciones generadas en un lenguaje de alto nivel a un robot convirtiéndolas automáticamente a comandos de movimiento de bajo nivel, de tal forma que se realice la tarea asignada. Esta tarea consiste típicamente en generar un cambio de *configuración* del robot de un lugar a otro, mientras se evitan los obstáculos presentes en el ambiente desconocido. Sin embargo, en la mayoría de la literatura se asume que el robot tiene un modelo preciso del ambiente antes de comenzar a desplazarse.

Es necesario entonces para resolver este último problema, el uso de sensores capaces de tomar medidas del ambiente mientras que a su vez el robot está en desplazamiento. Lo anterior permite generar un modelo del ambiente en el cual se pueden establecer todas las futuras decisiones que el robot debe tomar. El modelo propuesto se encarga de generar las principales características de lo que se conoce como grafo de visibilidad, y sobre el cual se planifican todas las posibles trayectorias que se pueden realizar en el ambiente de trabajo. Adicional al uso de los sensores para la identificación de los diferentes obstáculos presentes en el ambiente, también es fundamental considerar que en todo momento el robot debe conocer su localización en el mundo desconocido. El robot debe identificar constantemente su ubicación y la disposición geométrica de los obstáculos dentro del ambiente. Sumado a lo anterior, también es de gran relevancia considerar que el robot tiene un volumen específico, y por lo tanto no debe ser considerado como un punto en el espacio. Esto permite generar rutas que realmente pueden ser navegadas por el robot sin generar colisión alguna con los obstáculos en el ambiente.

En este artículo se presenta un modelo de identificación de un ambiente desconocido y su posterior utilidad en la generación de un grafo de visibilidad (Kavraki et al., 1996). Esta técnica de búsqueda de trayectorias libres de colisiones se implementó en un robot móvil Lego Mindstorm NXT (LaValle, 2006). El objetivo es construir una ruta libre de colisiones desde una posición inicial hasta una posición final en el ambiente previamente identificado tomando siempre en consideración las dimensiones físicas del robot. Es decir que el robot debe desplazarse por lugares que su tamaño se lo permita de tal forma que nunca colisione con los obstáculos.

2. Materiales y métodos

2.1. Identificación de obstáculos y generación de grafos de visibilidad

La planificación de movimientos se inquieta por la ejecución de una trayectoria geométrica desde una posición inicial en el espacio hasta un objetivo a través de un sistema móvil (Ghallab et al., 2004). La planificación distingue si el ambiente es conocido o no y dependiendo de esta clasificación deriva otros nombres, como son: planificación de movimientos en ambientes conocidos o fuera de línea y planificación de movimientos en ambientes desconocidos o en línea, respectivamente. En un problema de ambiente conocido, la configuración y orientación inicial, la configuración y orientación meta, así como el espacio de trabajo del robot son totalmente conocidos y estáticos, por lo que el robot puede moverse en él mismo basándose en su conocimiento previo. Mientras que en un problema de ambiente desconocido es necesario realizar un reconocimiento del espacio en el cual el robot va a navegar, ya que en este caso, solo se conoce una configuración inicial y final. En este trabajo, debido a que el grafo de visibilidad es un método de planificación de movimientos fuera de línea, es necesario identificar el ambiente previo a la realización del grafo de visibilidad. Esto se realiza mediante una implementación del algoritmo seguidor de paredes acorde a este caso, el cual se explica a continuación.

2.2 Algoritmo de identificación del ambiente

Para este caso se ha tomado el algoritmo ya existente del seguidor de paredes propuesto por Ferrari et al. (2002). El algoritmo se ha modificado de tal forma que el robot circunnavegue el área total del mundo desconocido. Este proceso de reconocimiento consta de dos partes (ver figura 1).

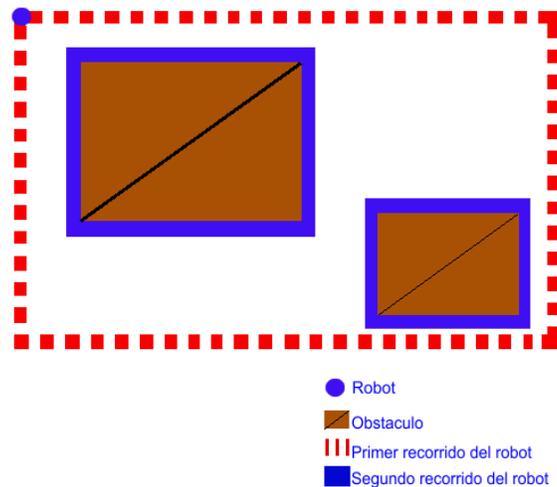


Figura 1. Ejemplo del algoritmo de navegación del ambiente.

La primera de ellas se origina en la coordenada (0,0) donde el robot se basa en las dimensiones del área de trabajo previamente ingresadas por el usuario. El robot circunnavega el área en busca de los diferentes nodos de cada uno de los obstáculos ubicados en el ambiente desconocido (franja roja en la figura 1) con el fin de utilizarlos posteriormente en la generación del grafo de visibilidad. Debido a que esta primera etapa no garantiza la identificación total de los nodos ubicados en el ambiente, se realiza la segunda etapa del proceso de identificación de obstáculos. Esta segunda etapa se basa en los nodos identificados en la primera etapa y se realiza una búsqueda más exhaustiva. En esta vez se visitan los nodos que fueron encontrados en la primera etapa con el fin de verificar su posición y de identificar a que obstáculo pertenecen (franja azul en la figura 1). En todo este proceso es fundamental que el robot conozca su posición dentro del mapa en todo momento, con el fin de poder brindar unas coordenadas exactas de los nodos basándose en su posición.

En algunas propuestas donde es necesaria la navegación de un robot en un ambiente (Guzmán, et al., 2011a y 2011b) se considera al robot como un punto capaz de navegar por todos los espacios de ese mundo. Sin embargo, en el ambiente real el robot no es un punto, al contrario este tiene unas medidas específicas y ocupa un volumen en el espacio. Se considera tener en cuenta las dimensiones del robot dentro de los espacios que puede navegar sin colisionar con obstáculo alguno. Esto se logra agrandando cada uno de los obstáculos presentes en ambiente con respecto al radio del robot, el cual es de 15 cm (ver figura 2), de tal forma que no se generen incongruencias en la identificación de los vértices.

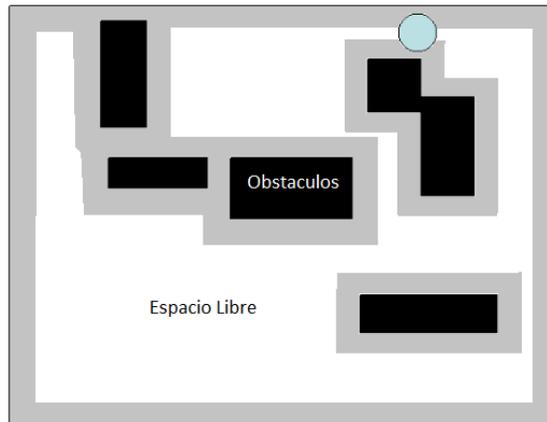


Figura 2. Modificación del espacio de configuración.

2.3 Grafos de visibilidad

Este es uno de los primeros métodos de planificación de movimientos (Latombe, 1991). Principalmente se aplica en espacios de configuración bidimensionales con regiones poligonales en los obstáculos. El grafo de visibilidad (ver figura 3) denotado por $G_{vis}(S)$, tiene un conjunto de nodos $V=\{v_1, \dots, v_n\}$, en el cual están incluidos los nodos de configuración inicial, final y cada uno de los vértices pertenecientes a los obstáculos. Los nodos v_i y v_j tienen un arco que los une si ambos están en la misma línea de visión, es decir, si el segmento $\overline{v_i v_j}$ no interseca el interior de un obstáculo en S . Las características que definen un mapa de visibilidad, son: (i) los nodos comparten un lado, si cada uno de ellos están en la misma línea de visión, (ii) todos los puntos en el espacio libre del robot están entre la línea de visión de por lo menos un nodo en el mapa de visibilidad, y (iii) las aristas del grafo son segmentos de línea que no pasan o no intersecan el interior de un obstáculo y que conectan dos nodos. La figura 3, detalla el proceso de creación de un grafo de visibilidad (Choset, 2005) y la figura 4 el algoritmo.

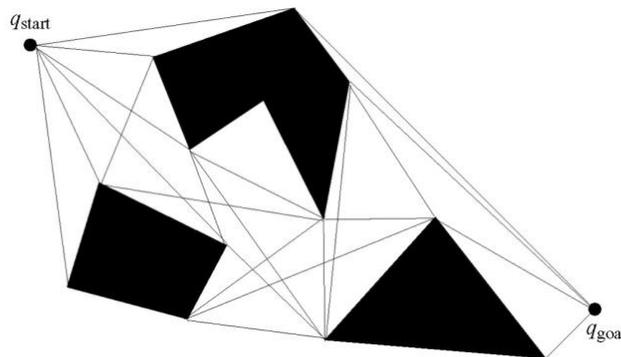


Figura 3. Grafo de visibilidad (Choset, 2005).

Algoritmo Grafo de Visibilidad

Entrada: Conjunto S de obstáculos.
Salida: Grafo de visibilidad $\mathcal{G}_{vis}(S)$.

1. Inicializar un grafo $\mathcal{G} = (V, E)$ donde V es el conjunto de todos los vértices de los polígonos en S y $E = \emptyset$
2. Para todos los vértices $v \in V$
3. Hacer $W \leftarrow \text{VERTICESVISIBLES}(v, S)$
4. Para cada vértice $w \in W$, agregar el arco (v, w) a E .

Figura 4. Algoritmo grafo de visibilidad (Choset, 2005).

La tarea primordial del grafo de visibilidad es, mediante las rutas identificadas, encontrar la más corta para la realización de un cambio de configuración del robot desde q_a (posición inicial) hasta q_f (configuración final) sin colisionar con los obstáculos en el ambiente. La figura 5, detalla el proceso identificación de la ruta más corta (Choset, 2005).

Algoritmo CaminoCorto

Entrada: Conjunto S de obstáculos, y dos puntos q_a y q_f en el espacio libre.
Salida: El camino libre de colisiones más corto conectando q_a y q_f .

1. $\mathcal{G}_{vis} \leftarrow \text{GRAFOVISIBILIDAD}(S \cup \{q_a, q_f\})$
2. Asignar a cada arco (v, w) en \mathcal{G}_{vis} un peso, el cual es la longitud Euclidiana del segmento \overline{vw}
3. Usar el algoritmo de Dijkstra para la computación del camino más corto entre q_a y q_f

Figura 5. Algoritmo del camino más corto (Choset, 2005).

La aplicación propone implementar el algoritmo de visibilidad sobre un robot Lego Mindstorms NXT 2.0 fabricado por Lego® (Oliveira et al, 2009), el cual es un *kit* de robótica programable lanzado al mercado en julio del 2006 (ver figura 6). Este tiene como componente principal un computador denominado *ladrillo*, en el cual se pueden conectar hasta tres motores y cuatro sensores. Con éste es posible desarrollar varios programas utilizando diferentes lenguajes de programación como C++, C# y Java, entre otros. La adaptación más conocida de este robot al lenguaje Java se denomina Java- Lejos NXJ.

Para el caso específico de este trabajo, son utilizados dos motores, dos sensores de ultrasonido y un compás magnético. El sensor de ultrasonido puede detectar objetos que estén de 0 a 255 cm de distancia con una precisión de ± 3 cm, y funciona igual que como lo hace un radar o un sonar. Envía ondas de sonido en una frecuencia por encima de la audible humanamente y mide el tiempo que el "eco" (el rebote de las ondas) de sonido tarda en regresar al sensor. En esta propuesta se utilizan dos sensores de ultrasonido. El primero se encarga de realizar la percepción total del mundo y de detectar la distancia entre el robot y cada uno de los obstáculos y el segundo para prevenir la colisión con obstáculos que no son visibles para el primer sensor. El otro sensor utilizado es el compás magnético, que como su nombre lo indica, mide el campo magnético de la Tierra y devuelve un valor representando la dirección actual del robot. La dirección magnética se calcula con precisión de 1° y devuelve un número entre 0 y 360.

La tarea de este último sensor es la de mejorar la precisión de los giros y de la navegación del robot para así poder generar una mejor identificación del ambiente desconocido.



Figura 6. Imagen del robot con el que se realizaron las pruebas.

3. Metodología

La secuencia metodológica aplicada en la investigación para la generación de grafos de visibilidad de un ambiente desconocido, con el objetivo de construir la ruta más corta para un cambio de *configuración* de un robot, se basa en el diagrama presentado en la figura 7. Allí se pueden distinguir cuatro etapas fundamentales: Especificaciones del ambiente, Identificación del ambiente desconocido, Planificación de movimientos para el cambio de configuración, y Ejecución del cambio de configuración del robot. Con el propósito de brindar una descripción más detallada, a continuación se cita cada uno de los elementos anteriores.

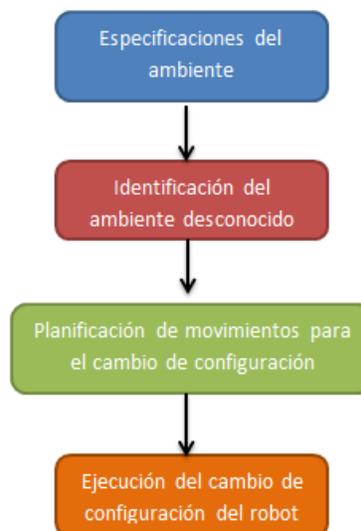


Figura 7. Diagrama de la metodología implementada.

3.1 Especificaciones del ambiente

Para permitir que un robot pueda realizar la planificación de movimientos en un ambiente desconocido, es necesario brindarle a éste de manera simple algunos datos que describan o permitan la inicialización de lo que el usuario desea hacer, de manera tal que el robot pueda comenzar la planificación de rutas de trayectoria a partir de esta información.

En este trabajo los datos de entrada son tres: coordenadas de un punto inicial, coordenadas de un punto final y la especificación de las dimensiones de la zona de trabajo. Los tres son enviados al robot vía *Bluetooth*. Los dos primeros son expresados como coordenadas de un plano cartesiano de dos dimensiones. Mientras que el tercero, corresponde a las medidas del área sobre la cual se va a generar el espacio de búsqueda.

3.2 Identificación del ambiente desconocido

Corresponde a dos componentes que trabajan conjuntamente.

(i) *Modelo matemático*. Este módulo es una parte fundamental en la generación del grafo de visibilidad, ya que mediante el modelo matemático se generan todas las coordenadas de los vértices que han sido identificados, genera la dirección y la distancia que debe recorrer el robot, con el fin de ir a la siguiente posición y luego culminar el análisis del obstáculo. Con este módulo también se calcula la posición absoluta del robot utilizando la información percibida por el sistema sensorial NXT y algunos conceptos propios de la trigonometría básica (Rosen, 2007) (ver figura 8).

La posición absoluta de los obstáculos detectados puede ser calculada mediante las funciones trigonométricas del seno y el coseno; utilizando las expresiones (1) y (2), para encontrar el x y el y respectivamente, asociados con el vértice identificado en un obstáculo;

$$x = x_i + \text{distancia} \times \cos \alpha \quad (1)$$

$$y = y_i + \text{distancia} \times \sin \alpha \quad (2)$$

Donde x_i e y_i son las coordenadas anteriores del robot, distancia, corresponde al valor numérico calculado como la distancia desde la posición actual del robot hasta el vértice encontrado, y α es la orientación del robot expresada en ángulos. Esta distancia es obtenida mediante la medición realizada por el sensor de ultrasonido desde el punto actual hasta el vértice del obstáculo al cual esta apuntado.

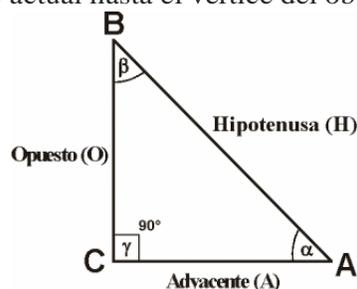


Figura 8. Ejemplo conceptos de trigonometría aplicados.

(ii) *Proceso de sentido*. Los datos empleados en las expresiones (1) y (2), como son la distancia y la orientación del robot son obtenidos en la tarea de identificación del mundo. El primero de ellos es hallado mediante el uso del sensor de ultrasonido a medida que el robot está en el proceso de reconocimiento del área desconocida, ya que está constantemente realizando mediciones para lograr hallar los diferentes obstáculos. En el caso del ángulo, se halla generando un ángulo perpendicular al ángulo de posicionamiento del robot.

La aplicación de las ecuaciones (1) y (2), permiten de manera recursiva construir el mapa con todas las características para posteriormente generar el grafo de visibilidad y así hallar la ruta óptima a realizar por el robot para trasladarse del punto inicial al punto final.

3.3 Módulo de planificación de movimientos

(i) *Algoritmo grafo de visibilidad.* En este módulo se construye el grafo de visibilidad con los obstáculos hallados en el módulo de identificación del ambiente. En este grafo se obtienen todas las posibles rutas que el robot puede utilizar para llegar de la *configuración* inicial a la final (Muñoz, 1997). Luego Implementando el algoritmo de Dijkstra se obtiene la ruta óptima para ese cambio de *configuración*.

(ii) *Traductor lejos.* Este es el encargado de llevar los datos generados por el grafo de visibilidad y el algoritmo de Dijkstra expresados como un conjunto de coordenadas a visitar para la realización de la ruta óptima a comandos de bajo nivel, expresados como instrucciones del lenguaje LeJOS (Oliveira et al., 2009), las cuales finalmente son interpretadas y ejecutadas por el robot. En este se le informa al robot la velocidad, el ángulo y la distancia que debe recorrer para lograr llegar al objetivo final.

3.4 Robot móvil

Este módulo representa la parte visual del algoritmo, ya que en él se puede ver al robot en pleno movimiento, ejecutando las instrucciones, resultado de todos los procesos previos de los demás módulos. En esta parte el robot recibe desde el ordenador vía *Bluetooth*, comandos de bajo nivel para realizar la ruta óptima encontrada en la planificación de movimientos.

4. Resultados y discusión

Con el fin de evaluar la precisión del algoritmo de visibilidad acá descrito, se realizaron varias pruebas experimentales.

Para la primera prueba experimental, se ha diseñado el ambiente mostrado en la figura 9, en el cual el robot comienza el reconocimiento del ambiente con un par de obstáculos. El robot comienza desde las coordenadas (0,0) a circunnavegar el ambiente (Pilot, 2012) y a identificar los obstáculos (Rectangle, 2012) del mapa mediante el sensor de ultrasonido de tal forma que halle todos sus vértices. Una vez obtenidos estos datos se pueden generar las diferentes trayectorias y encontrar la ruta óptima. Debido a las características de este escenario y a la implementación de la geometría, el robot no pasará entre los dos obstáculos debido a que dicho espacio es inferior a su tamaño (De la Rosa, 2004), por lo tanto el robot tomará a ambos obstáculos como uno solo. El área de trabajo corresponde a un cuadrado cuyo lado mide 90 cm.

La evaluación consiste en analizar si el modelo a escala construido por el robot en el mapa de visibilidad, se ejecuta de manera precisa en el mundo real. Para ello, una vez el robot ejecute el correspondiente movimiento en el mundo, se medirá la distancia posición actual de ejecución, respecto la posición esperada definida por los puntos fijos de los vértices de cada obstáculo del mundo real.

Al poner a correr el robot Lego Mindstorms, implementado con la arquitectura propuesta se obtuvo el siguiente mapa del ambiente en el ordenador:

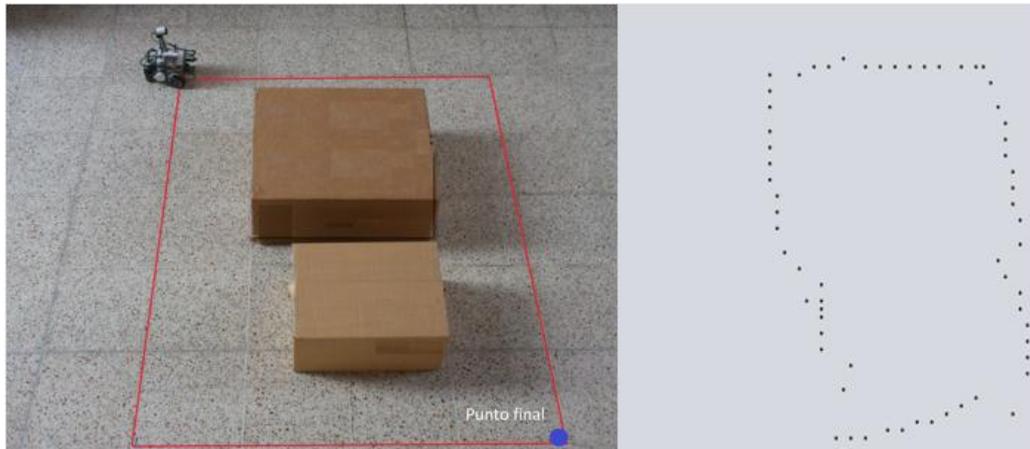


Figura 9. Imagen ambiente de prueba1 y mapa obtenido por las lecturas del sensor de ultrasonido.

Luego de la identificación total del ambiente, en el ordenador se generó el siguiente grafo de visibilidad del ambiente descubierto junto a su ruta más corta identificada, realizada por el algoritmo de Dijkstra y pintada de azul (ver figura 10).

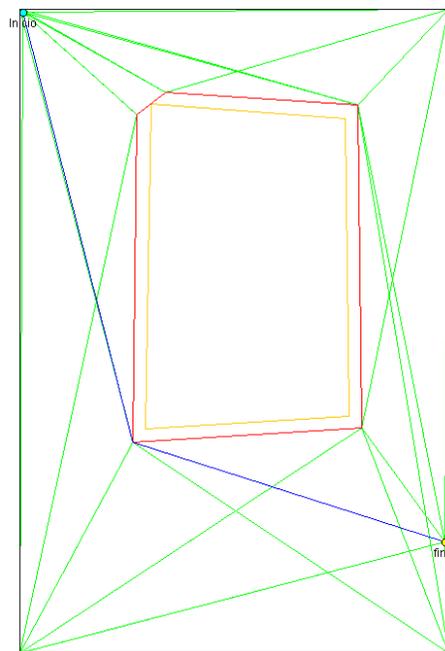


Figura 10. Grafo de visibilidad generado.

Los datos obtenidos para esta prueba son registrados en la tabla 1.

Tabla 1. Prueba de mapas de visibilidad con un obstáculo.

| Punto | Coordenadas Teóricas (cm) | Coordenadas Experimentales (cm) | Error en x (cm) | Error en y (cm) |
|----------------------|----------------------------------|--|------------------------|------------------------|
| Punto Inicial | (0, 0) | (0, 0) | 0 | 0 |

| | | | | |
|----------------|--------------|---------------|-----|-----|
| Punto 1 | (30.5, 30.5) | (31.8, 31.5) | 6.8 | 4 |
| Punto 2 | (90.5, 30.5) | (91.3, 30.6) | 1.3 | 0.1 |
| Punto 3 | (87, 129) | (92.2, 119) | 5.2 | 10 |
| Punto 4 | (42, 127) | (35.9, 117.5) | 6.1 | 9.5 |

La segunda prueba consistió en acercar uno de los obstáculos un poco más al punto final, esta vez con espacio suficiente para que el robot identificara en el ambiente, a diferencia de la prueba anterior, dos obstáculos independientes (ver figura 11).

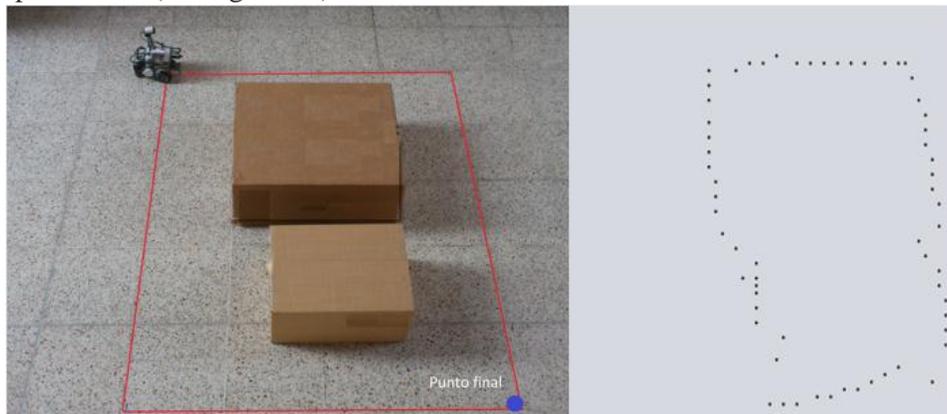


Figura 11. Imagen ambiente de prueba 2 y mapa obtenido por las lecturas del sensor de ultrasonido.

A continuación se presenta el grafo de visibilidad generado del ambiente (ver figura 12).

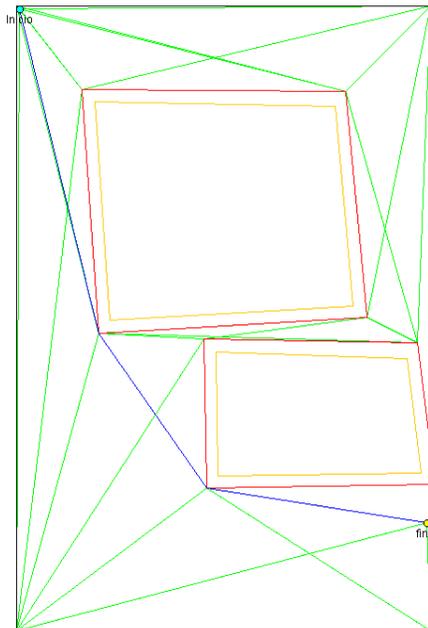


Figura 12. Grafo de visibilidad del ambiente.

Los datos obtenidos para esta prueba son registrados en las tabla 2.

Tabla 2.

| Punto | Coordenadas Teóricas (cm) | Coordenadas Experimentales (cm) | Error en x (cm) | Error en y (cm) |
|----------------------|----------------------------------|--|------------------------|------------------------|
| Punto Inicial | (0, 0) | (0, 0) | 0 | 0 |
| Punto 1 | (30.5, 30.5) | (31.4, 30.9) | 0.9 | 0.4 |
| Punto 2 | (90.5, 30.5) | (91.4, 27.9) | 0.9 | 2.6 |
| Punto 3 | (90.5, 80) | (95.5, 84.4) | 5 | 4.4 |
| Punto 4 | (30.5, 80) | (26.5, 33, 85.9) | 4 | 5.9 |
| Punto 5 | (77.5, 114.6) | (73, 110.4) | 4.5 | 4.2 |
| Punto 6 | (77.5, 146) | (75.1, 142.5) | 2.4 | 4.6 |
| Punto 7 | (90.5, 146) | (91.4, 141.3) | 0.9 | 4.7 |
| Punto 8 | (90.5, 114.6) | (87.4, 113.3) | 3.1 | 1.3 |

En las tablas anteriores se muestran los resultados obtenidos al ejecutar la aplicación y las coordenadas que le son enviadas al computador desde el robot para la generación del mapa. Estas coordenadas representan la ubicación de cada uno de los vértices en el mundo, es decir tanto los vértices del obstáculo como el punto final e inicial. Como se puede ver en cada una de las tablas, al ser ejecutada la aplicación el robot visita cada uno de los vértices de los objetos en el mundo sin explorar, con el fin de encontrar obstáculos que aún no han sido identificados y así obtener las aristas entre cada vértice. Mientras más preciso sea el desplazamiento del robot respecto a las coordenadas teóricas, más exacto será el mapa que se está realizando y se podrá confirmar más acertadamente que la ruta que el robot eligió como la más corta sí es efectivamente la óptima.

5. Conclusiones

Se realizó la experimentación de una técnica de planificación de movimientos, a través de la construcción de un robot Lego NXT 2.0, implementado bajo el lenguaje Java Lejos.

Debido a que el robot está expuesto a errores de odometría, imprecisión de sus diferentes partes, como lo son los motores y los sensores, en esta propuesta se utilizó la tarea de la planificación de movimientos que se torna en un problema arduo, y en la cual la inteligencia que se le es otorgada al robot mediante la programación y el manejo de los diferentes errores juega un papel crucial. Era entonces la meta de esta propuesta otorgarle al robot estas capacidades tan importantes, de tal forma que se pudiera lograr un método de planificación exitoso y totalmente independiente del manejo del usuario. Se considera que esto ha sido logrado.

Con la propuesta expuesta en este artículo se ha demostrado como mediante la identificación de un ambiente desconocido y su posterior uso con el método de grafos de visibilidad, se puede hallar una ruta óptima para lograr un cambio de configuración de un robot.

Un aspecto que se debe considerar en esta propuesta es la utilidad de la plataforma de programación LeJOS, ya que su gran variedad de bibliotecas siempre permitió encontrar las herramientas apropiadas para lograr la completitud de esta aplicación.

Como trabajo futuro se propone implementar esta propuesta para otros escenarios con el fin de poder llegar a otras conclusiones. También se puede realizar una comparación con otros métodos de *roadmaps* para lograr mejores resultados.

6. Referencias

- Choset, K. (2005). Principles of Robot Motion: Theory, Algorithms, and Implementations. Boston: MIT Press.
- De la Rosa, E. (2004). Heurística para la generación de configuraciones en pasajes estrechos aplicada al problema de los clavos. Capítulo I. Planificación de movimientos. Tesis Maestría en Ciencias con Especialidad en Ingeniería en Sistemas Computacionales. Universidad de las Américas Puebla.
- Ferrari, M., Ferrari, G. & Hempel, R. (2002). Building robots with LEGO Mindsorms. USA: Syngress.
- Ghallab, M., Nau, D. & Traverso, P. (2004). Automated Planning, theory and practice. San Francisco: Morgan Kaufman.
- Guzmán, J. & Torres, I. Vélez, C. (2011a). Experiencias en la Implementación de las Técnicas de Descomposición Aproximada y Trapezoidal para la Navegación en Robots Móviles Lego. XI Latin American Robotics Competition, Bogotá, Colombia.
- Guzmán, J.; Torres, I. & Álvarez, D. (2011b). Aplicando Algoritmos de Visibilidad para la Planificación de Movimientos en un Robot Lego. XI Latin American Robotics Competition, Bogotá, Colombia.
- Kavraki, L.; Svestka, P.; Latombe, J. & Overmars, M. (1996). Probabilistic Roadmap for Path Planning in High-Dimensional Configuration Spaces, IEEE. Transactions on Robotics and Automation, 12(4), 566-580.
- Latombe, J. (1991). Kluwer Academic Publishers.
- LaValle, S. (2006). Planning Algorithms. Cambridge: Cambridge University Press.
- Muñoz, V. (1997) Planificación de Trayectorias para Robots Móviles. Tesis Doctoral. Departamento de Ingeniería de Sistemas y Automática. Planificación de Trayectorias para Robots Móviles.
- Oliveira, G.; Silva, R.; Lira, T. & Reis, L. (2009). Environment Mapping using the Lego Mindstorms NXT and LeJS NXJ. Fourteenth Portuguese Conference on Artificial Intelligence EPIA.
- Rosen, K. (2007). Discrete Mathematics and its Applications. Nueva York: McGrawHill.