

Robótica I

Tarea 3. Polígono de Visualización

10 de mayo de 2020

Marco Antonio Esquivel Basaldua

I. INTRODUCCIÓN

En robótica móvil y planificación de movimientos, el polígono de visualización en un espacio plano en \mathbb{R}^2 se define como el espacio que el robot puede "ver" desde su posición actual suponiendo que éste cuenta con un sensor omnidireccional de rango.

En este documento se considera el caso de un espacio poligonal simplemente conectado y en el que el robot es un punto y cuenta con un sensor de rango infinito.

II. ALGORITMO

De manera análoga a una implementación real, el algoritmo desarrollado crea un disco unitario virtual de sensado al rededor de la posición del robot con una resolución de dos rayos de sensado por grado en la circunferencia del disco. Estos rayos de sensado encuentran la intersección más cercana de la recta desde la ubicación del robot, en dirección al rayo de sensado, con las aristas del espacio libre poligonal. Cada intersección encontrada es guardada como un vértice en el polígono de visualización.

La implementación de este algoritmo requiere como entradas dos archivos de texto. El primero de ellos, llamado "map.txt", contiene las dimensiones del mapa, la cantidad de vértices en el espacio libre y las coordenadas de estos vértices ordenados de manera a que cada punto ordenado está conectado al punto siguiente y al anterior formando un arista. El segundo archivo de texto, llamado "robot.txt", contiene la ubicación del robot dentro del espacio libre. El usuario debe tener el cuidado de ubicar al robot dentro del espacio libre del mapa.

El algoritmo usado en esta implementación se describe a continuación.

Algorithm 1: Visual Polygon

Data: Free space vertices, robot location and sensor points ubication
for 1 to 720 **do**
 Get closest intersection to free space in robot – sensor direction
 Add intersection to visual polygon vertices
return Vertices on visual polygon

III. RESULTADOS

A continuación se muestran los resultados obtenidos al aplicar este algoritmo para dos mapas distintos y la posición del robot ubicado en distintas coordenadas.

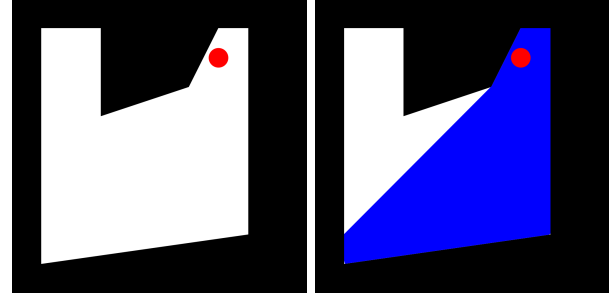


Figura 1. Mapa 1 de dimensiones 100×100 con posición del robot en las coordenadas (70, 80) y polígono de visualización generado.

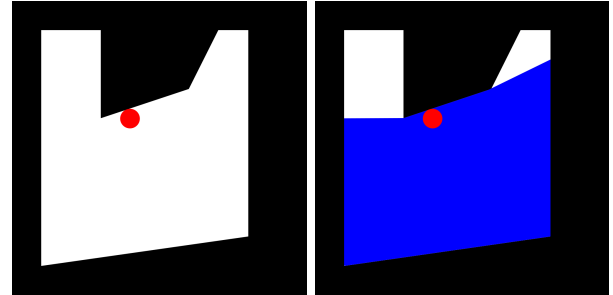


Figura 1. Mapa 1 de dimensiones 100×100 con posición del robot en las coordenadas (40, 60) y polígono de visualización generado.

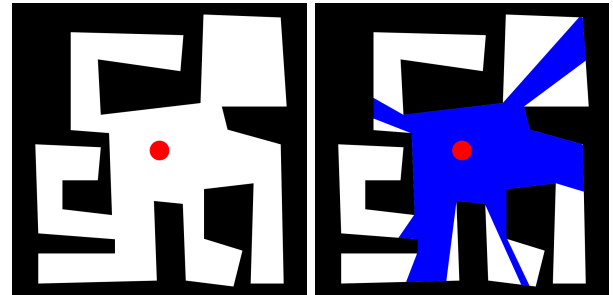


Figura 1. Mapa 2 de dimensiones 100×100 con posición del robot en las coordenadas (50, 50) y polígono de visualización generado.

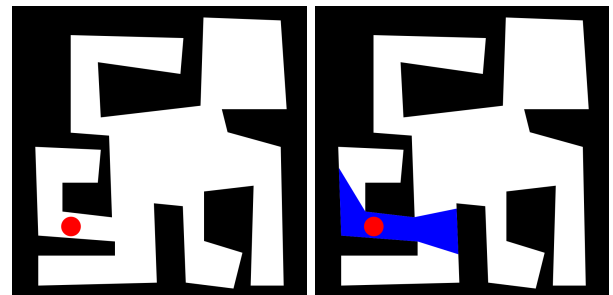


Figura 1. Mapa 2 de dimensiones 100×100 con posición del robot en las coordenadas (20, 25) y polígono de visualización generado.

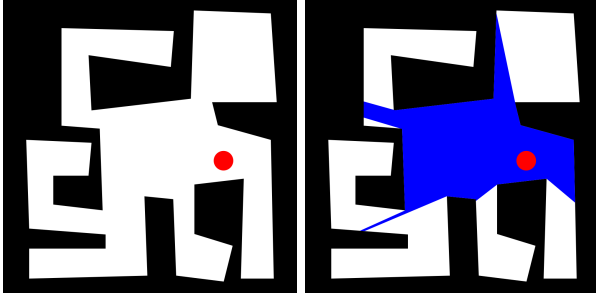


Figura 1. Mapa 2 de dimensiones 100×100 con posición del robot en las coordenadas $(75, 45)$ y polígono de visualización generado.

IV. CONCLUSIONES

Al utilizar un algoritmo que asemeja la implementación de un sensor real omnidireccional de resolución finita, se genera un polígono de 720 vértices cualquiera que sea la configuración del mapa en el que se genera el polígono.

Cada rayo de sensado presente debe buscar las intersecciones que se generan con las aristas del espacio libre poligonal por lo que la complejidad del algoritmo es de $\mathcal{O}(720 * n)$, siendo n la cantidad de vértices en el espacio libre. En el caso en el que $n \leq 720$ la complejidad del algoritmo sería igual o mayor a $\mathcal{O}(n^2)$, en el caso en que $n > 720$ se estaría logrando una complejidad lineal.