

# Práctica 5

## Cálculo de rutas utilizando A\* y celdas de ocupación

M.I. Marco Negrete

Robots Móviles y Agentes Inteligentes

### Objetivos

- A partir del mapa creado en la práctica 4, calcular una ruta mediante el algoritmo A\*.
- Suavizar la ruta utilizando descenso del gradiente.
- Publicar la ruta en un tópico y desplegarla en el visualizador `rviz`.

## 1. Introducción

### 1.1. El algoritmo A\*

La planeación de rutas consiste en la obtención de un movimiento continuo que conecte una configuración inicial, con una final mientras se evaden obstáculos al mismo tiempo. Se parte del supuesto de que se dispone de una representación del ambiente en la cual hay información sobre el espacio navegable y el espacio ocupado por los obstáculos. En esta práctica se asume que el robot sólo se mueve sobre un plano y que se tiene una representación del ambiente que consiste en un mapa de celdas de ocupación (obtenido en la práctica 4).

Una forma de encontrar una ruta es aplicando un algoritmo de búsqueda en grafos. En el caso de las celdas de ocupación, cada celda representa un nodo en el grafo y se considera que cada celda (nodo) está conectada únicamente con sus celdas vecinas. Para determinar los nodos vecinos se puede utilizar conectividad cuatro u ocho. En esta práctica se utilizará la conectividad cuatro.

A\* es un algoritmo de búsqueda que explora la ruta con el menor costo esperado. Para un nodo  $n$ , el costo esperado  $f(n)$  se calcula como

$$f(n) = g(n) + h(n)$$

donde  $g(n)$  es el costo de la ruta desde el nodo origen hasta el nodo  $n$  y  $h(n)$  es una heurística que determina un costo que se esperaría tener desde el mismo nodo  $n$  hasta el nodo objetivo. Este costo esperado de hecho subestima el valor real, es decir, se debe cumplir que  $h(n) \leq g(n) \quad \forall n \in \text{Grafo}$ .

En la búsqueda por  $A^*$  se manejan dos conjuntos principales, uno llamado *lista abierta* y el otro, *lista cerrada*. La lista abierta contiene todos los nodos que han sido visitados pero no expandidos y la cerrada, aquellos que han sido visitados *y* expandidos (también llamados nodos conocidos). El algoritmo 1 muestra los pasos en pseudocódigo para implementar  $A^*$ .

## 2. Desarrollo

**Nota:** Para esta práctica se asume que el alumno ejecutó correctamente la práctica 4.

### 2.1. Suavizado

### 2.2. Visualización

## 3. Evaluación

- La práctica se considera entregada si el mapa representa al ambiente de manera satisfactoria.

---

**Algoritmo 1:** Búsqueda con  $A^*$ 

---

**Datos:** Grafo, nodo inicial, nodo meta

**Resultado:** Ruta óptima expresada como una secuencia de nodos

Cerrado  $\leftarrow \emptyset$

Abierto  $\leftarrow \{\text{nodo\_inicial}\}$

previo(nodo\_inicial)  $\leftarrow \emptyset$

**mientras** *Abierto*  $\neq \emptyset$  **hacer**

    nodo\_actual  $\leftarrow$  nodo con el menor valor  $f$  del conjunto *Abierto*

    Abierto  $\leftarrow$  Abierto - {nodo\_actual}

    Cerrado  $\leftarrow$  Cerrado  $\cup$  {nodo\_actual}

**si** *nodo\_actual es nodo\_meta* **entonces**

        Anunciar éxito y salir de este ciclo

**fin**

**para cada** *nodo\_vecino* de *nodo\_actual* **hacer**

**si** *nodo\_vecino*  $\in$  *Cerrado* **entonces**

            Continuar con el siguiente nodo\_vecino

**fin**

**si** *nodo\_vecino*  $\in$  *Abierto* **entonces**

            costo\_temporal  $\leftarrow g(\text{nodo\_actual}) + d(\text{nodo\_actual}, \text{nodo\_vecino})$

**si** *costo\_temporal*  $< g(\text{nodo\_vecino})$  **entonces**

$g(\text{nodo\_vecino}) \leftarrow \text{costo\_temporal}$

$f(\text{nodo\_vecino}) \leftarrow \text{costo\_temporal} + \text{heurística}(\text{nodo\_vecino}, \text{nodo\_meta})$

                previo(nodo\_vecino)  $\leftarrow$  nodo\_actual

**fin**

**en otro caso**

$g(\text{nodo\_vecino}) \leftarrow g(\text{nodo\_actual}) + d(\text{nodo\_actual}, \text{nodo\_vecino})$

$f(\text{nodo\_vecino}) \leftarrow g(\text{nodo\_vecino}) + \text{heurística}(\text{nodo\_vecino}, \text{nodo\_meta})$

            previo(nodo\_vecino)  $\leftarrow$  nodo\_actual

            Abierto  $\leftarrow$  Abierto  $\cup$  {nodo\_vecino}

**fin**

**fin**

**fin**

**si** *nodo\_actual*  $\neq$  *nodo\_meta* **entonces**

    Anunciar falla

**en otro caso**

    RutaOptima  $\leftarrow \emptyset$

**mientras** *nodo\_actual*  $\neq \emptyset$  **hacer**

        //El nodo actual se inserta al principio de la ruta

        RutaÓptima  $\leftarrow \{\text{nodo\_actual}\} \cup \text{RutaÓptima}$

        nodo\_actual  $\leftarrow$  previo(nodo\_actual)

**fin**

    Regresar RutaÓptima

**fin**

---