

Práctica 3: Navegación global

Lunes SGrupo 1.4:

Francisco Carrillo Perez.

Jose Antonio España Valdivia.

Rafael Leyva Ruiz.

Cristina Zuheros Montes.

30 Mayo 2016

Índice

1. Tarea 1: Extensión del algoritmo A*	2
2. Tarea 2: Mejora de tiempo del algoritmo	2
2.1. Uso de la distancia de Manhattan	2
2.2. Aumento de número de iteraciones	2
2.3. Añadimos unos pesos	2
3. Tarea 3: Experimentación	2
3.1. Experimentación en el mundo willow-garage	3
3.2. Experimentacion en simple-rooms	3
3.3. Experimentación en autolab	4

1. Tarea 1: Extensión del algoritmo A*

Solución

La realización del algoritmo A* consiste en lo siguiente:

- Se inicializa calculando los nodos calculando los nodos e introduciéndolos en la lista de abiertos. Mientras que la lista de nodos abiertos no esté vacía, se coge el primer nodo de la lista de abiertos y se coge su índice y se comprueba si es el nodo objetivo se hace una ruta y si no se introduce en cerrados y se saca de la lista de abiertos y se añaden en la lista de abiertos los vecinos que no están ni en abiertos ni en cerrados.
- Se accede a costmap a través de un puntero.
- Reajuste del coste de los nodos de la lista de abiertos en referencia al coste del mejor camino hasta ese nodo.
- Las heurísticas utilizadas consisten en primer lugar realizamos una estimación de la distancia euclídea, ya que en ningún momento vamos a sobreestimar el coste real. En segundo lugar, para calcular la seguridad del camino le vamos sumando el valor de los nodos sobre los que pasará el robot al ir por ese camino. Así obtenemos un camino más seguro.

Como hacemos un sort de la lista de nodos que tenemos, cuando saquemos el primero de la lista de abiertos sabemos que obtenemos aquél con el menor valor, ya que se les ordena en función de su valor F.

2. Tarea 2: Mejora de tiempo del algoritmo

2.1. Uso de la distancia de Manhattan

Para la optimización del tiempo del algoritmo A* hemos empleado como distancia para la heurística la distancia Manhattan cuya fórmula matemática es:

$$D(X_1, X_2) = \|X_1 - X_2\| = \sum_{i=1}^n |X_{1i} - X_{2i}|$$

Aligerando notablemente los cálculos realizados durante la planificación. Esta distancia es más rápida en tiempo y peor en la calidad de la ruta encontrada.

2.2. Aumento de número de iteraciones

Hemos aumentado el número de iteraciones del algoritmo A* permitiendo explorar más nodos y que pueda llegar a objetivos más lejanos en el caso de que la exploración de estos nodos sea más costosa.

```
if(currentIndex==cpgoal.index || explorados == 200000)
```

2.3. Añadimos unos pesos

Para que nuestra estimación heurística no tenga tanto peso, añadimos unos pesos, en este caso una función $w(n)$ que lo que hace es que divide el valor de la función $h(n)$ en el nodo actual entre el del nodo start. Con esto lo que conseguimos es que conforme nos vamos acercando más al objetivo, toma más importancia la $g(n)$ que nuestra estimación heurística.

```
w = calculateHCost(neighborCells[i],goalCell)/calculateHCost(start,goalCell);
```

```
CP.fCost=CP.gCost+w*CP.hCost;
```

3. Tarea 3: Experimentación

Solución

3.1. Experimentación en el mundo willow-garage



Figura 1:

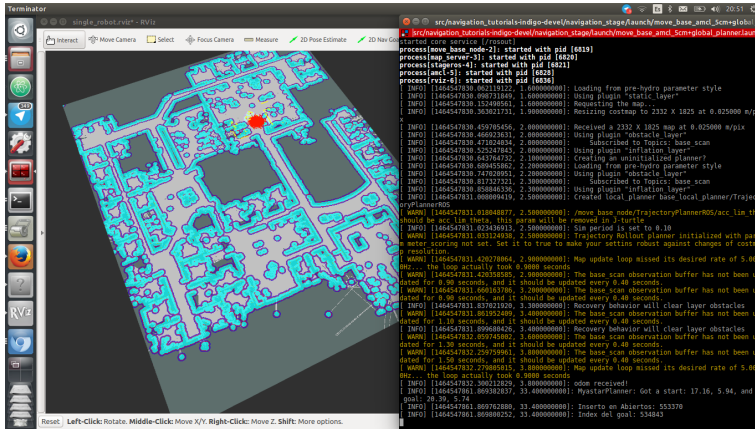


Figura 2:

3.2. Experimentacion en simple-rooms

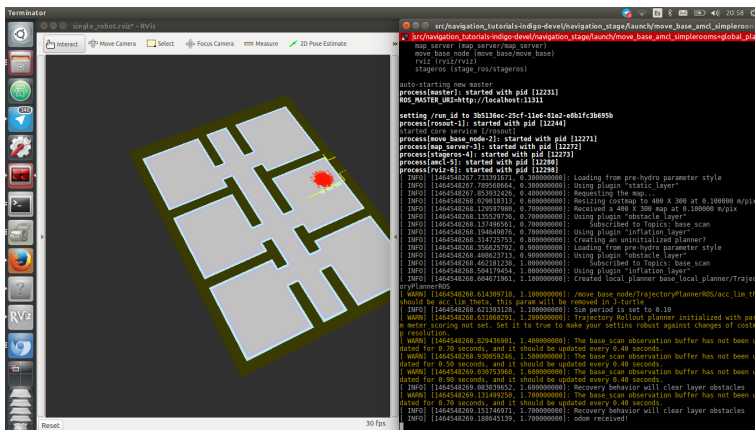


Figura 3:

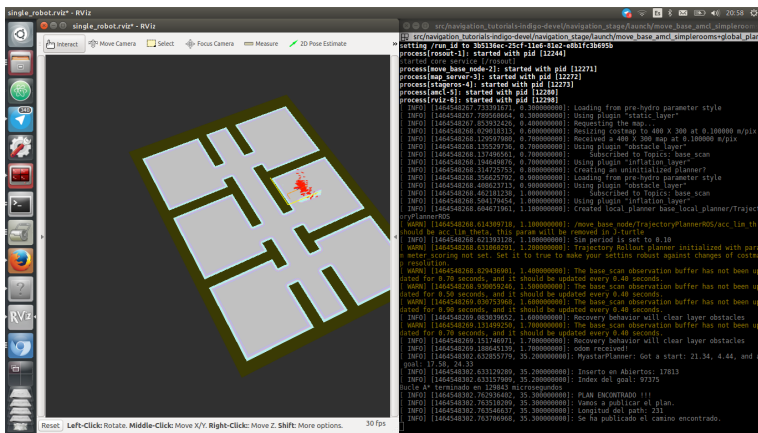


Figura 4:

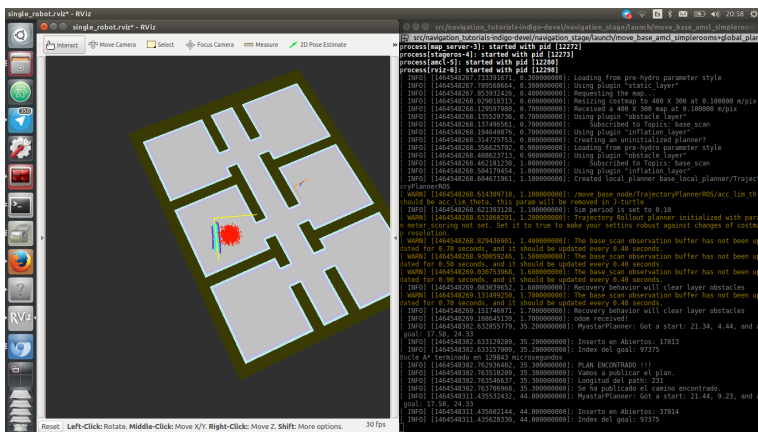


Figura 5:

3.3. Experimentación en autolab

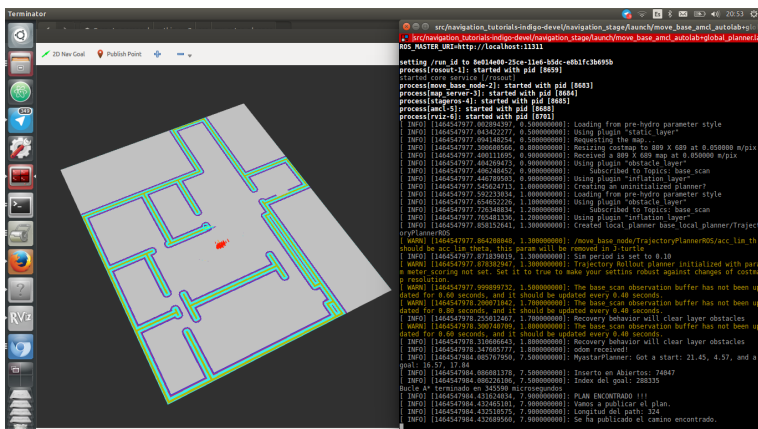


Figura 6:

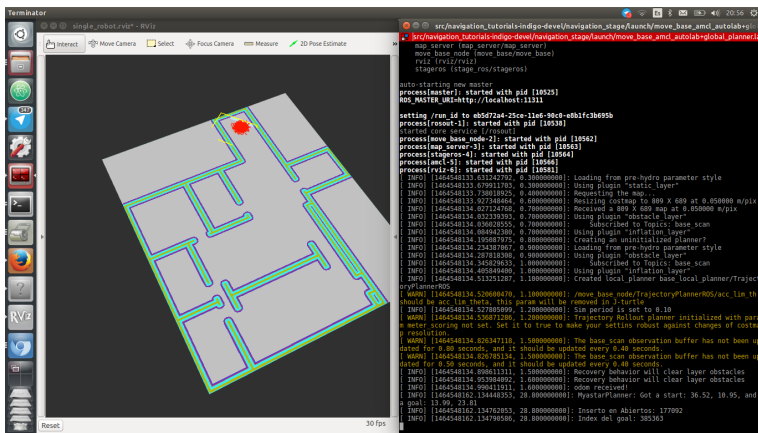


Figura 7: