



Técnicas de los Sistemas Inteligentes.

Curso 2016-17.

Práctica 1: Robótica

Entrega 2: Explorador basado en fronteras.

1.1 Objetivo

El objetivo de esta entrega consiste en implementar un explorador basado en fronteras para que un robot pueda reconocer el mapa de su entorno. Los conceptos sobre exploración de mapas basada en fronteras se han explicado en clase y están brevemente explicados en el documento "*Explicacion_codigo_fuente.pdf*" en PRADO, además ahí se describe el código fuente básico a partir del cual el alumno deberá implementar el explorador de fronteras.

1.2 Tareas

Para facilitar la implementación, se recomienda seguir los siguientes pasos:

1. Realizar todos los ejercicios propuestos en el documento "*EjerciciosMapeo1*". Una vez realizados correctamente se tendrá un proceso capaz de detectar una frontera y seleccionar un punto de esta frontera siguiendo un criterio definido por el programador.
2. Realizar el ejercicio propuesto al final de la clase de la Sesión 5, en el que se propone añadir a la implementación de los ejercicios anteriores un proceso de envío de posición objetivo al robot. El envío de un objetivo se hace utilizando un cliente de acciones que se comunica con el servidor de acciones de move_base (paquete responsable de las operaciones de navegación del robot) tal y como se explica en las transparencias de clase de la Sesión 5. Para depurar el código es recomendable utilizar las funciones de visualización en **rviz** que se describen en la explicación del código fuente.
3. Una vez comprobado que todos los elementos componentes del explorador de fronteras funcionan, hay que implementar en la función main del código fuente proporcionado, una estrategia para la exploración de fronteras. Una propuesta es la siguiente:
 - a. Repetir sin fin
 - i. Girar 360º (quizá recomendable un par de veces)



- ii. Detectar la frontera
- iii. Seleccionar un punto de la frontera
- iv. Enviar al robot a ese punto objetivo.
- v. Adoptar una estrategia si no se alcanza el objetivo.

Es importante tener en cuenta que esta es una propuesta básica, pueden hacerse las modificaciones que se consideren oportunas para mejorar la eficiencia y efectividad de la exploración basada en fronteras.

- 4. La condición de parada del bucle principal dependerá de la comprobación de que la fase de detección de fronteras no encuentra ningún punto que cumpla la condición de

Experimentación. Se pide que se realicen al menos dos experimentaciones con la solución implementada, una con el mapa y mundo *corridor* y otra con el mapa y mundo *willow_garage*. Los archivos del mapa y del mundo están localizados en los directorios *map* y *worlds* respectivamente del paquete *mi_mapeo_stage*. En los distintos ficheros *launch* del directorio *launch* del paquete *mi_mapeo_stage* hay ejemplo de cómo utilizar ambos mapas y mundos. El análisis de la experimentación se escribirá en la memoria que debe entregarse, como se describe más abajo.

- 5. **Preparación del paquete ROS.** Para entregar el programa hay que crear un nuevo paquete ROS llamado *explorador_fronteras*, con las siguientes características:

- a. El fichero *CMakeLists.txt* debe estar configurado adecuadamente para que se compile un nodo con el nombre *explorador_fronteras_node* (usar el *CMakeLists.txt* del paquete *mi_mapeo_stage* como base para crearlo).
- b. En el manifiesto del paquete, escribir el nombre y apellidos de la persona que lo realiza.
- c. En el directorio *launch* del paquete debe aparecer un archivo *explorador_fronteras.launch* que incluya todos los nodos necesarios para ejecutar correctamente la exploración basada en fronteras. Puede mantenerse la misma estructura de directorios que hay en el paquete *mi_mapeo_stage*.
- d. Tener en cuenta que el nodo *explorador_fronteras_node* se debe ejecutar en una terminal aparte del fichero *launch*. Un ejemplo de llamada del nodo sería:

```
roslaunch explorador_fronteras explorador_fronteras_node
```

1.3 Material a entregar

El material entregado tiene que incluir

- 1. Una memoria en pdf de 5 páginas explicando el algoritmo en general que se ha implementado, indicando especialmente cómo se detecta la frontera, qué criterio se ha tenido en cuenta para seleccionar un nodo de la frontera, qué estrategia se ha seguido cuando se detecta que no se



ha alcanzado un objetivo enviado. La memoria tiene que incluir además un apartado para la experimentación en donde se muestre, por cada experimento, una imagen del mapa explorado con la máxima extensión de área conocida posible. Por cada experimento indicar el tiempo empleado en la exploración y una explicación sobre la cantidad de área descubierta (si ha descubierto el mapa en su totalidad o parcialmente y por qué)-

2. Un paquete ROS con los ficheros necesarios para que se pueda ejecutar el código. Es decir, el código fuente, fichero de manifiesto, CMakeFiles.txt y el fichero *launch*, organizados siguiendo la convención de paquetes ROS. La entrega debe ser simple de ejecutar, es decir, una vez se haya obtenido una copia del material, ejecutando

```
catkin-make --pkg=explorador_fronteras  
roslaunch explorador_fronteras explorador_fronteras.launch (en una  
terminal)
```

```
roslaunch explorador_fronteras explorador_fronteras_node <timeout> (en  
otra terminal)
```

debería ser suficiente para lanzar la exploración de fronteras

Entregar un fichero .zip que incluyados **ficheros**

1. el **fichero pdf** de la memoria,
2. y **un fichero zip** que incluye el paquete ROS comprimido.

1.4 Fecha de entrega

Todos los grupos: Hasta el 21 de Abril de 2017.