

Práctica 4

Creación de una mapa de celdas de ocupación mediante el paquete `gmapping`

M.I. Marco Negrete

Robots Móviles y Agentes Inteligentes

Objetivos

- Crear un mapa del Laboratorio de Biorrobótica que sea útil para las tareas de navegación que se implementarán en las siguientes prácticas.
- Familiarizar al alumno con el uso del paquete `GMapping` para creación de mapas geométricos.
- Familiarizar al alumno con la adquisición de lecturas del sensor láser Hokuyo UG01.

1. Introducción

Las celdas de ocupación son un tipo de mapa geométrico en el que el espacio se discretiza con una determinada resolución. Cada una de las celdas (o píxeles en el caso de espacios bidimensionales) tiene un valor asignado que indica el nivel de ocupación del espacio que ésta representa. En general el nivel de ocupación es booleano, esto es, cada celda puede tener sólo dos valores: espacio libre o espacio ocupado. En caso de usar un enfoque probabilístico, el valor de cada celda indica la probabilidad de que esté ocupada.

ROS tiene el mensaje `OccupancyGrid`, dentro del paquete `nav_msgs`, para representar este tipo de mapas. En este mensaje, el valor de ocupación de cada celda se representa con un entero en el intervalo $[0, 100]$. El valor de -1 indica que no se tiene información sobre la celda.

El paquete `gmapping` sirve para crear celdas de ocupación a partir de las lecturas de un sensor láser y de la odometría calculada por la base móvil. El mapa generado se puede guardar en una imagen utilizando el paquete `map_server`.

2. Desarrollo

Nota: Para esta práctica se asume que el alumno ejecutó correctamente las prácticas 1 y 2.

2.1. Prerrequisitos

Como primer paso, asegúrese de actualizar y recompilar el repositorio mediante los comandos:

```
cd ~/RoboticsCourses
git pull origin master
cd catkin_ws
catkin_make
```

Antes de poder utilizar el hardware, es necesario configurar los nombres que el sistema operativo asignará al sensor láser, joystick y base móvil, para ello ejecute los siguientes comandos:

```
sudo cp ~/RoboticsCourses/Prerequisites/80-justinaRobot.rules /etc/udev/rules.d/
sudo udevadm control --reload-rules
sudo service udev restart && sudo udevadm trigger
```

Además, es necesario instalar el paquete `gmapping` con el comando:

```
sudo apt-get install ros-indigo-gmapping
```

2.2. Creación del mapa

En el laboratorio se dispone de dos bases móviles para realizar las pruebas: JustinaCN y JustinaNL. Las figuras 1a y 1b son fotos de las dos bases disponibles.

Para comenzar a crear el mapa con JustinaCN, conecte el cable USB etiquetado como “USB_Main” a cualquier puerto de la computadora y ejecute el comando:

```
roslaunch bring_up mapping_justinaCN
```

Si se utiliza a JustinaNL, ejecute el comando:

```
roslaunch bring_up mapping_justinaNL
```

En ambos casos se ejecutarán varios nodos, de los cuales, los más importantes son: `gmapping`, `hokuyo_node` y `mobile_base`. El visualizador debe mostrar algo parecido a la figura 2.

Las bases móviles se teleoperan con el *stick* derecho del joystick. Para crear el mapa, basta con mover la base por todo el espacio que se quiera representar. El mapa se actualiza cada que el robot se mueve una distancia de 30 [cm] o gira un ángulo de 0.2 [rad]. Es importante no mover la base muy rápido para que el mapa se pueda crear correctamente.

Si el mapa dibujado en el visualizador comienza a mostrar espacios incongruentes será necesario comenzar el mapeo nuevamente. Basta con detener el proceso y ejecutar nuevamente el comando `roslaunch`. Es probable que se requieran varios intentos para obtener un buen mapa.

Una vez que se ha mapeado todo el espacio deseado, ejecute, **desde otra terminal, sin detener los procesos previos**, el siguiente comando:



(a) JustinaCN



(b) JustinaNL

Figura 1: Bases móviles disponibles en el Laboratorio de Biorrobótica

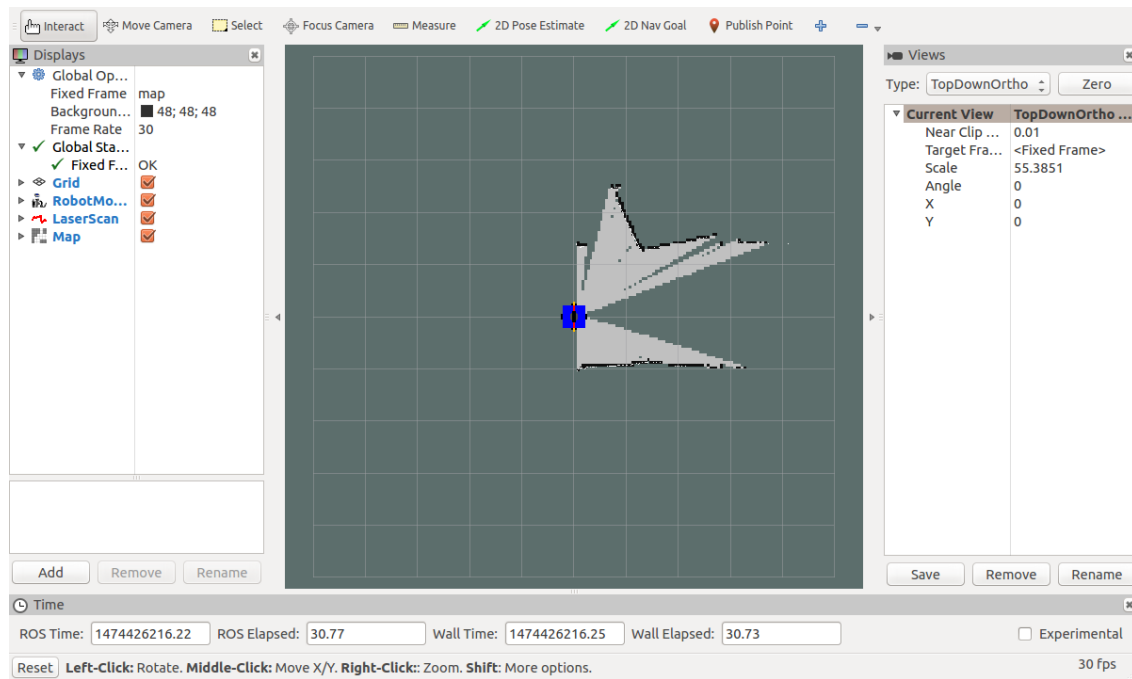


Figura 2: Ejemplo de lo que muestra el visualizador al comienzo del mapeo

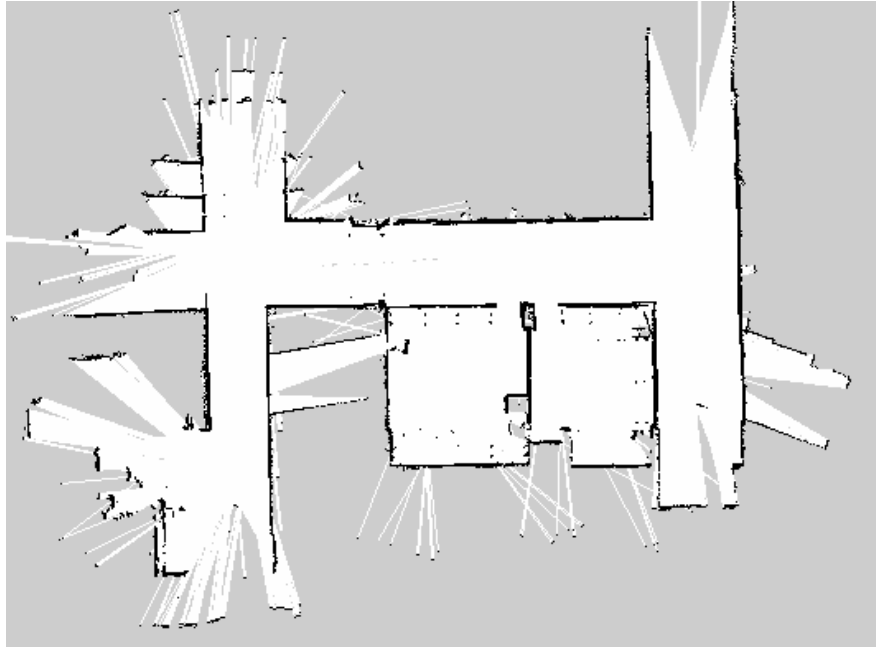


Figura 3: Ejemplo de un mapa

```
roslaunch map_server map_saver -f ~/RoboticsCourses/catkin_ws/src/student/biorobotics_map
```

Este comando guarda el mapa en la carpeta `student` (creada en clase); dicho mapa se guarda en dos archivos, uno con extensión `.pgm` (el mapa en sí), que puede ser abierto por un visualizador de imágenes, y un archivo `.yaml`, que contiene información relativa al mapa.

Compare la imagen con el ambiente real. Si la representación no es suficientemente buena, repita el proceso completo. La figura 3 es un ejemplo de un mapa obtenido con este proceso.

3. Evaluación

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