

## Práctica 2. Mapas métricos

### Descripción general

En esta práctica vamos a implementar algunas de las estructuras de datos que se utilizan habitualmente para almacenar la información de mapas de entorno métricos en robótica móvil.

### Python

Para el desarrollo de la práctica se utilizará Python como lenguaje de programación. No se va a proporcionar ningún código adicional. El alumno deberá crear todos los ficheros de código que necesite.

Sin embargo, sí que se proporcionarán ficheros con datos reales obtenidos mediante sensores de rango 3D montados sobre robots móviles en diferentes experimentos. Estos ficheros de datos estarán codificados en texto plano para facilitar la comprensión de su formato. El formato escogido es el de de datos 3D de la librería Point Cloud Library (PCL). Podemos ver un ejemplo de cabecera de un fichero .pcd en la Tabla 1.

```
# .PCD v0.7 - Point Cloud Data file format
VERSION 0.7
FIELDS x y z r g b
SIZE 4 4 4 1 1 1
TYPE F F F I I I
COUNT 1 1 1 1 1 1
WIDTH 58826
HEIGHT 1
VIEWPOINT 0 0 0 1 0 0 0
POINTS 58826
DATA ascii
```

Table 1: Ejemplo de cabecera en formato PCD

Como se puede observar, el número de puntos que contiene el fichero aparece tras los campos WIDTH Y DATA. El campo FIELDS nos indica la información que contiene el fichero, en este caso, las coordenadas (x, y, z) del punto y los valores de color RGB. Relacionados con este campo tenemos el campo SYZE y el campo TYPE que nos dicen el número de bytes que se dedica a cada elemento de FIELDS y el tipo. En este ejemplo, los datos de coordenadas son float de 4 bytes y los datos de color son enteros de 1 byte.

Tras la cabecera encontraremos los datos 3D obtenidos por el sensor tendremos las líneas formadas por los datos en sí. Siguiendo la información de la cabecera, normalmente encontraremos 6 valores por cada punto, tres números reales y tres números enteros. Los primeros se corresponderán con las coordenadas (x, y, z) del punto en un sistema de coordenadas centrado en el sensor. Los segundos suelen ser información de color, que no vamos a necesitar para la práctica.

## **Ejercicio 1. Almacenar los datos**

En el primer ejercicio de la práctica tendremos que implementar dos métodos para almacenar los datos métricos del entorno del robot. El objetivo es poder realizar una comparación analítica entre ambos métodos. Los métodos a implementar son una rejilla de ocupación y una estructura de datos tipo Oc-Tree, como la que se ha visto en clase de teoría.

En ambos casos, la información que tenemos que almacenar en las celdas será el número de puntos que caen dentro de la celda y la media de esos puntos. En caso de que la celda esté vacía, el número de puntos será 0. También, para los dos métodos estableceremos a priori el tamaño mínimo de celda.

Finalmente, una vez que los métodos estén implementados correctamente, se debe realizar un análisis de su funcionamiento. Para ello compararemos parámetros como el número total de celdas que utiliza cada método, la memoria utilizada, el número de celdas vacías, el número de celdas ocupadas, la media de puntos en las celdas ocupadas. Esta comparativa se puede hacer para diferentes tamaños mínimos de celda y así comprobar la respuesta de cada método para distintos tamaños de celda.

## **Ejercicio 1. Parte optativa**

Crear un visualizador de datos 3D en python que utilice la estructura de Oc-Tree para la visualización eficiente de los datos. Se podrá utilizar cualquier librería de visualización en python siempre que no implemente internamente la estructura Oc-Tree.

## **Ejercicio 1. Documentación**

Es obligatorio documentar todo el trabajo realizado. Si alguna parte del trabajo realizado no se documenta se considerará como que no se ha hecho, lo que puede llevar a suspender la práctica. La documentación debe incluir una comparación detallada de los métodos implementados y su rendimiento, así como de las pruebas que se hayan realizado para comprobar su correcto funcionamiento.

## **Ejercicio 1. Evaluación**

El ejercicio 1 se puntuará de 0 a 10 puntos. La parte optativa supondrá hasta un 20% más de nota.

## Entrega de la práctica

La fecha límite de entrega de esta segunda práctica será el día 14 de enero. La entrega se realizará a través de una tarea en la página moodle de la asignatura. Los alumnos deberán entregar un fichero comprimido que contenga el o los ficheros con el código fuente en python y el fichero con la documentación en **formato PDF**.

- **!!!IMPORTANTE!!!** Recordad que las prácticas son individuales y NO se pueden hacer en parejas o grupos. Cualquier código copiado supondrá un suspenso de la práctica para todas las personas implicadas en la copia. Se utilizarán herramientas para la detección de copia. Estos hechos serán comunicados a la Escuela Politécnica para que se tomen las medidas disciplinarias oportunas contra los infractores.
- **!!!IMPORTANTE!!!** El código fuente de los algoritmos a implementar en esta práctica se pueden encontrar fácilmente en Internet. No obstante, se espera una implementación original por parte de los alumnos. En caso de detectarse el plagio, la práctica quedará automáticamente suspendida con una nota de 0.