

Ejercicio 6 – Visión 3D

Este ejercicio tiene como objetivo aplicar los conceptos aprendidos en el Tema 9: Visión 3D.

Hay que entregar un archivo comprimido **ejercicio6.zip** con el código del programa y ejemplos de su funcionamiento con las 3 nubes de puntos proporcionadas, el cual deberás subir al Aula Virtual.

Puntos totales posibles del ejercicio: 10

Instrucciones

En esta sesión vamos a trabajar con nubes de puntos 3D utilizando la librería PCL.

Instalación de PCL en Ubuntu

Para instalar la librería PCL en la máquina virtual, debes realizar lo siguiente:

```
$ sudo apt-get update
$ sudo apt-get install libpcl-dev
```

Crear un proyecto

```
$ mkdir /MiPath/MI_PROYECTO
$ vi CMakeLists.txt
```

El fichero **CMakeLists.txt** contiene algo similar a lo siguiente, al igual que los ejemplos vistos en clase. Debes cambiar MI_PROYECTO por el nombre del proyecto que estés construyendo:

```
cmake_minimum_required(VERSION 2.6 FATAL_ERROR)
project(MY_PCL_PROJECT)
find_package(PCL 1.4 REQUIRED COMPONENTS)
include_directories(${PCL_INCLUDE_DIRS})
link_directories(${PCL_LIBRARY_DIRS})
add_definitions(${PCL_DEFINITIONS})
add_executable(MI_PROYECTO main.cpp)
target_link_libraries(MiProyecto ${PCL_LIBRARIES})
```

Compilar y ejecutar el proyecto

```
$ cd /MiPath/MiProyecto
```

Crea tu programa MiProyecto.cpp en este directorio

```
$ mkdir build  
$ cd build  
$ cmake ..
```

Cada vez que modifiques tu programa, deberás realizar lo siguiente:

```
$ make  
$ ./MiProyecto
```

Si deseas consultar más detalles acerca de cómo usar PCL en tu propio proyecto, puedes hacerlo en http://pointclouds.org/documentation/tutorials/using_pcl_pcl_config.php#using-pcl-pcl-config

Ejercicio 1: Programa ejemplo de visualización

Crea un proyecto para **view3D.cpp** (proporcionado en el Aula Virtual) que te permitirá visualizar nubes de puntos y mover el punto de vista de la cámara.

El programa requiere un argumento con el nombre del fichero en formato .pcd que contiene la nube de puntos a visualizar. Opcionalmente se puede especificar una segunda nube de puntos, en cuyo caso se visualizan las dos nubes en distintas áreas de visualización.

Puedes probar a visualizar las nubes de ejemplo que tienes disponible en el Aula Virtual:

- Cloud0.pcd: representa una escena con una mesa (descargada del tutorial de PCL)
- Cloud1.pcd: escena de un despacho adquirida con la Kinect
- Cloud2.pcd: escena de un pasillo adquirida con la Kinect

Si deseas consultar la descripción del formato de fichero .pcd la encontrarás en: http://pointclouds.org/documentation/tutorials/pcd_file_format.php#pcd-file-format

Ejercicio 2: Programa ejemplo de filtrado

Crea un proyecto para **filter.cpp** (proporcionado en el Aula Virtual) que te permitirá especificar una nube de puntos y aplicar un filtro a dicha nube. El programa genera dos nubes de puntos con los inliers y outliers detectados por el filtro.

Para más detalles de este ejemplo, ver:

http://pointclouds.org/documentation/tutorials/statistical_outlier.php#statistical-outlier-removal

Ejecuta el programa para la nube Cloud0.pcd y comprueba sus resultados visualizándolos con el programa **view3D**.

Ejercicio 3: Extracción de planos

Puntuación: 10 puntos

Escribe un programa usando la librería PCL que permita extraer los n planos más grandes de una nube de puntos. La salida deben ser n nubes de puntos que representen cada uno de los planos detectados. Realiza pruebas con distintas nubes de puntos de ejemplo.

En el tutorial de PCL, tienes un ejemplo que te ayudará a implementar tu programa: https://pcl.readthedocs.io/projects/tutorials/en/latest/extract_indices.html

En dicho ejemplo, se utiliza un objeto **seg** de la clase **pcl::SACSegmentation<pcl::PointXYZ>** para la segmentación de un plano. El resultado de la segmentación **seg.segment()** son los coeficientes de la ecuación del plano y una lista de índices a los puntos de la nube que pertenecen a dicho plano.

Observa que para recuperar dichos puntos a partir de la lista de índices, se utiliza un filtro extract de tipo **pcl::ExtractIndices<pcl::PointXYZ>**