

# PRÁCTICA 2 ROS

## Detección de dígitos y ArUcos en ROS

17 de Enero de 2023

Esta práctica consiste en la detección de dígitos y ArUcos en ROS. El objetivo principal es determinar las luminosidades a las que determinadas cámaras son capaces de detectar ArUcos. Para ello, se realiza el experimento que se muestra en la Figura 1, es decir, partiendo de la oscuridad total se incrementa la luminosidad abriendo progresivamente una caja, que contiene:

- Un luxómetro para medir la luminosidad.
- Una cámara apuntando al luxómetro para registrar los valores de luminosidad.
- Una cámara a analizar (de la cual queremos obtener a qué valores de luminosidad se pueden detectar los ArUcos) que apunta hacia un tablero de 16 ArUcos.

Como resultado de dicho experimento, se extrae un bag que contiene las imágenes de cada cámara, las que registra el luxómetro y las correspondientes al tablero de ArUcos, junto con la marca de tiempo correspondiente a cada imagen. Para el procesamiento de dichas imágenes, se han creado dos paquetes en ROS:

- *aruco\_markers*: En este paquete se encuentran los *scripts* necesarios para leer los arucos, contar los arucos detectados en cada imagen y escribir un archivo *.csv* con la marca de tiempo y el número de ArUcos detectados.
- *text\_from\_images*: En este paquete se encuentran los *scripts* necesarios para leer los dígitos, y escribir un archivo *.txt* con la marca de tiempo y los dígitos que representan la luminosidad. Para detectar los dígitos a partir de las imágenes, se incluye un script para entrenar una pequeña red neuronal (una red neuronal convolucional o CNN) que reconoce los diferentes dígitos a partir de las imágenes que recibe como entrada.

El objetivo de esta práctica es que el alumno sea capaz de inspeccionar paquetes de ROS, entendiendo la estructura de un proyecto y para qué sirven sus *scripts*, además de detectar los módulos de interés para su aplicación. El alumno deberá completar o modificar el código necesario para realizar los ejercicios evaluables.

### EJERCICIOS

- **Ejercicio 1** (3 pts): Clonar el repositorio de GitHub que contiene los paquetes necesarios para la realización de esta práctica en el siguiente Enlace al repositorio. Seguir los pasos que aparecen en su descripción para su configuración e instalación.

- **Ejercicio 2** (3 pts): En el paquete *text\_from\_images*, buscar el archivo que crea el modelo que clasifica los dígitos. En dicho archivo, completar el código que viene marcado con *GRADED*, definiendo y entrenando la red neuronal. Lanzar dicho *script* con python3 para entrenar la red y comprobar que el modelo se ha guardado en la carpeta *model* del paquete *text\_from\_images*.
- **Ejercicio 4** (1 pt): Para detectar los dígitos, se ha seguido el siguiente procedimiento. i) Se detecta el cuadrado iluminado del luxómetro, donde están los dígitos, binarizando la imagen. ii) Se aplica el detector de bordes Canny. iii) Se aplica sobre dicha imagen la transformada de Hough para detectar las líneas correspondientes al los bordes del cuadrado. iv) Se hallan los vértices del cuadrado hallando la intersección entre las líneas. Para realizar todo lo anterior, es necesario lanzar el archivo *detect\_corners.launch*, seleccionando la cámara deseada. El procedimiento se ilustra en la Figura 1, a la derecha.
- **Ejercicio 5** (1 pt): Tras lanzar el *script* anterior, se imprimen por pantalla los vértices del cuadrado detectado. Sin embargo, estos vértices pueden cambiar de posición si movemos la cámara en otro experimento. Para que el paquete sea robusto a diferentes distancias entre cámara y luxómetro y las imágenes de entrada de la CNN tengan siempre el mismo tamaño, realizamos homografía. Es decir, mediante una matriz de homografía se relaciona la transformación entre dos planos, en este caso el “cuadrado” que contiene los dígitos pasa a formar un cuadrado de tamaño 100x100. En la Figura 1, a la derecha, se muestra dicho cuadrado superpuesto a la imagen original.  
 Para más adelante realizar la homografía mientras leemos los dígitos, son necesarios los vértices del cuadrado original y los del cuadrado final. Estos son imprimidos por pantalla tras lanzar el *launch* anterior. Copiar y pegar dichos puntos en el archivo de configuración correspondiente.
- **Ejercicio 6** (1 pt): Encuentra el *script* que se llama en el archivo *dr\_experiments.launch* encargado de leer y escribir los dígitos. Completa la parte donde aparezca *GRADED*. Finalmente, lanza el launch y observa los resultados.
- **Ejercicio 7** (1 pt): Tras analizar los resultados para la cámara blue, ¿detectas algún problema? ¿Se te ocurre alguna solución? Si no pudieses utilizar una red neuronal para clasificar los dígitos, ¿qué harías? Puedes escribir tu respuesta y entregarla o comentarla con el profesor si está disponible.



Figura 1: **Izquierda:** Montaje experimental. A la izquierda, el luxómetro y la cámara apuntándolo. A la derecha el tablero de ArUcos y cámara a analizar apuntando a ellos. **Derecha:** La imagen sobre la que se realiza la detección del cuadrado y la imagen resultante tras realizar la homografía.