

---

# SESION 5

---

PROCESADO DE IMAGEN Y VISIÓN POR COMPUTADOR



Universidad  
de Alcalá

8 DE NOVIEMBRE DE 2023

Ricardo Martínez Guadalajara

## Introducción

En esta sesión se va a trabajar con detección de bordes y umbrales en imágenes y vídeos con OpenCV

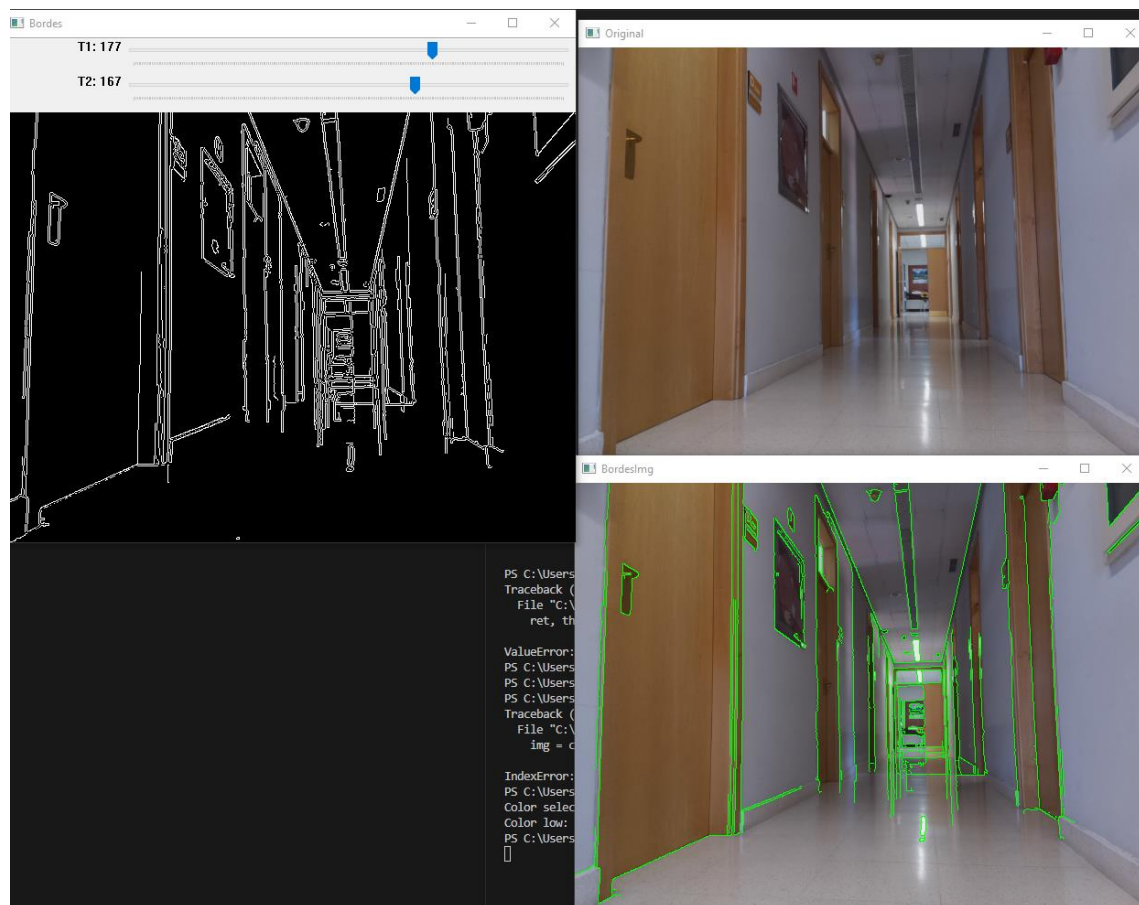
## Objetivo

Desarrollar los apartados del guion de prácticas como programas de Python

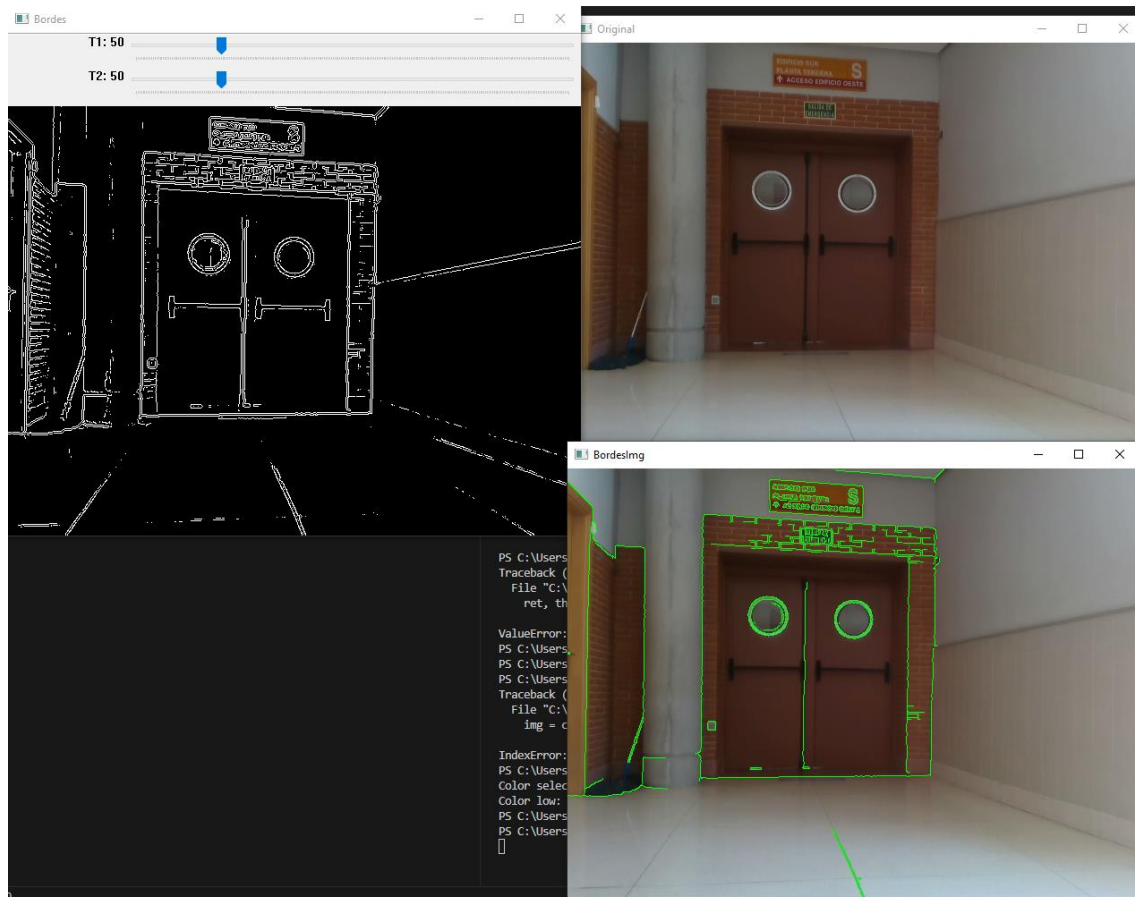
## Contenido

### Apartado 1

Se presenta el fichero Apartado1.py que contiene la representación de imágenes su detección de bordes con dos trackbars para controlar los parámetros T1 y T2. Unos valores razonables son 177 y 167 ya que se consigue una correcta detección de los bordes sin captar ruido.



Comparando ambas imágenes (Politecnica1 y Politecnica2) se puede comprobar que el algoritmo de Canny funciona un poco pero en detección de bordes sobre superficies brillantes o irregulares.



## Apartado 2

En este apartado jugamos con el umbralizado de videos. Se aporta un programa llamado apartado2Mod.py en el que se realizan dos ejercicios, el primero con un umbral fijo se observa que con los cambios de iluminación se produce efectos no deseados como la no detección de diferencias entre la pelota y el fondo y que umbraliza todo el video.

Sin embargo, en el segundo ejercicio con el la técnica de Otsu, se observa como adapta el umbral de forma dinámica para equilibrar el resultado.

## Apartado 3

En este apartado se pide comprender el código del fichero Apartado3.py. En análisis es el siguiente:

Define una función EventoRaton que se activa cuando se hace clic en la imagen. Esta función toma el color del píxel en el que se hizo clic, crea una máscara que identifica los píxeles en la imagen que están dentro de un cierto umbral de ese color, y luego colorea esos píxeles en verde.

Carga una imagen del sistema de archivos. El nombre del archivo de la imagen se pasa como argumento al script.

Lee un umbral de la línea de comandos. Este umbral se utiliza para determinar qué píxeles son "suficientemente cercanos" al color seleccionado para ser coloreados.

Crea una imagen vacía del mismo tamaño que la imagen original. Esta imagen se utiliza para almacenar el color seleccionado.

Muestra la imagen original en una ventana y establece la función EventoRaton como el controlador de eventos del ratón para esa ventana.

Entra en un bucle infinito que espera a que el usuario haga clic en la imagen. Cuando el usuario hace clic en la imagen, la función EventoRaton se activa, colorea los píxeles correspondientes en verde y actualiza la imagen en la ventana.

Una vez que el usuario ha seleccionado un color, el script calcula los umbrales de color superior e inferior para la segmentación. Estos umbrales se calculan sumando y restando el umbral del color seleccionado.

El script luego utiliza la función `cv2.inRange` para crear una máscara que identifica los píxeles en la imagen original que están dentro del rango de color especificado por los umbrales.

Los píxeles en la imagen original que corresponden a la máscara se colorean en verde.

Finalmente, el script muestra la imagen segmentada en una ventana,