

CENTRO DE ENSEÑANZA TÉCNICA INDUSTRIAL



REPORTE DE PRÁCTICA 5

Profesor: Mauricio Alejandro Cabrera Arellano

Alumno: Ángel de Jesús Mejía Suarez

21310182

6°G

Materia: Vision Artificial

12 de mayo de 2025

PRÁCTICA 5: Umbrales.

OBJETIVO: Objetivo: Utilizar las funciones de umbrales para la recuperación de información.

Threshold1 binary, b_inv, Trunc, To Zero, Tz_inv, Mean, Gaus, Otsu.

CÓDIGO:

```
import cv2
import matplotlib.pyplot as plt

# --- Cargar imagen a color ---
img_color = cv2.imread('BanderaMexico.png')
if img_color is None:
    print("✗ No se pudo cargar la imagen. Verifica el nombre y la ubicación del archivo.")
    exit()

# Convertir a RGB (porque OpenCV carga en BGR)
img_rgb = cv2.cvtColor(img_color, cv2.COLOR_BGR2RGB)

# Convertir a escala de grises solo para los umbrales
img_gray = cv2.cvtColor(img_color, cv2.COLOR_BGR2GRAY)

# ----- Umbrales -----
_, binary = cv2.threshold(img_gray, 127, 255, cv2.THRESH_BINARY)
_, binary_inv = cv2.threshold(img_gray, 127, 255, cv2.THRESH_BINARY_INV)
_, trunc = cv2.threshold(img_gray, 127, 255, cv2.THRESH_TRUNC)
_, tozero = cv2.threshold(img_gray, 127, 255, cv2.THRESH_TOZERO)
_, tozero_inv = cv2.threshold(img_gray, 127, 255, cv2.THRESH_TOZERO_INV)

# Umbrales adaptativos
mean_c = cv2.adaptiveThreshold(img_gray, 255, cv2.ADAPTIVE_THRESH_MEAN_C,
                               cv2.THRESH_BINARY, 11, 2)
gauss_c = cv2.adaptiveThreshold(img_gray, 255, cv2.ADAPTIVE_THRESH_GAUSSIAN_C,
                                cv2.THRESH_BINARY, 11, 2)

# Otsu
_, otsu = cv2.threshold(img_gray, 0, 255, cv2.THRESH_BINARY + cv2.THRESH_OTSU)

# ----- Mostrar resultados -----
titulos = ['Imagen Original (Color)', 'Binary', 'Binary Invertido', 'Truncado',
           'To Zero', 'To Zero Inv', 'Adaptativo Media', 'Adaptativo Gaussiano',
           'Otsu']
imagenes = [img_rgb, binary, binary_inv, trunc, tozero, tozero_inv, mean_c,
            gauss_c, otsu]
```

```
plt.figure(figsize=(12, 10))
for i in range(len(imagenes)):
    plt.subplot(3, 3, i+1)
    cmap = 'gray' if i != 0 else None
    plt.imshow(imagenes[i], cmap=cmap)
    plt.title(titulos[i])
    plt.axis('off')

plt.tight_layout()
plt.show()
```

DEMOSTRACIÓN:

