

# Práctica #5



Materia: Sistemas de Visión Artificial

Grupo: 7°E1

Isaac Alejandro Gutiérrez Huerta 19110198

11/05/2022

## **Práctica #5**

### ***Objetivo:***

Umbrales.

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 numpy as np
```

```
from matplotlib import pyplot as plt
```

```
libro = cv2.imread('ImgUmbrales.jpg')
```

```
ret, threshold1 = cv2.threshold(libro, 170, 255, cv2.THRESH_BINARY)
```

```
ret, threshold2 = cv2.threshold(libro, 170, 255, cv2.THRESH_BINARY_INV)
```

```
ret, threshold3 = cv2.threshold(libro, 170, 255, cv2.THRESH_TRUNC)
```

```
ret, threshold4 = cv2.threshold(libro, 170, 255, cv2.THRESH_TOZERO)
```

```
ret, threshold5 = cv2.threshold(libro, 170, 255, cv2.THRESH_TOZERO_INV)
```

```
grises = cv2.cvtColor(libro, cv2.COLOR_BGR2GRAY)
```

```
ret, threshold6 = cv2.threshold(grises, 170, 255, cv2.THRESH_BINARY)
```

```
threshold7 = cv2.adaptiveThreshold(grises, 255, cv2.ADAPTIVE_THRESH_GAUSSIAN_C,  
cv2.THRESH_BINARY, 115, 1)
```

```
ret, threshold8 = cv2.threshold(grises, 20, 255, cv2.THRESH_BINARY + cv2.THRESH_OTSU)
```

```
threshold9 = cv2.adaptiveThreshold(grises, 255, cv2.ADAPTIVE_THRESH_MEAN_C,  
cv2.THRESH_BINARY, 115, 1)
```

```
orig = cv2.cvtColor(libro, cv2.COLOR_BGR2RGB)  
th1 = cv2.cvtColor(threshold1, cv2.COLOR_BGR2RGB)  
th2 = cv2.cvtColor(threshold2, cv2.COLOR_BGR2RGB)  
th3 = cv2.cvtColor(threshold3, cv2.COLOR_BGR2RGB)  
th4 = cv2.cvtColor(threshold4, cv2.COLOR_BGR2RGB)  
th5 = cv2.cvtColor(threshold5, cv2.COLOR_BGR2RGB)  
th6 = cv2.cvtColor(threshold6, cv2.COLOR_BGR2RGB)  
th7 = cv2.cvtColor(threshold7, cv2.COLOR_BGR2RGB)  
th8 = cv2.cvtColor(threshold8, cv2.COLOR_BGR2RGB)  
th9 = cv2.cvtColor(threshold9, cv2.COLOR_BGR2RGB)
```

```
res, gr = plt.subplots(2,5)  
gr[0,0].imshow(orig)  
gr[0,0].set_title('Original')  
gr[0,0].axis('off')
```

```
gr[0,1].imshow(th1)  
gr[0,1].set_title('Binary')  
gr[0,1].axis('off')
```

```
gr[0,2].imshow(th2)  
gr[0,2].set_title('Binary Inv')  
gr[0,2].axis('off')
```

```
gr[0,3].imshow(th3)  
gr[0,3].set_title('Trunc')  
gr[0,3].axis('off')
```

```
gr[0,4].imshow(th4)
gr[0,4].set_title('To Zero')
gr[0,4].axis('off')

gr[1,0].imshow(th5)
gr[1,0].set_title('To Zero Inv')
gr[1,0].axis('off')

gr[1,1].imshow(th6)
gr[1,1].set_title('Binary Grises')
gr[1,1].axis('off')

gr[1,2].imshow(th7)
gr[1,2].set_title('Gaus')
gr[1,2].axis('off')

gr[1,3].imshow(th8)
gr[1,3].set_title('Otsu')
gr[1,3].axis('off')

gr[1,4].imshow(th9)
gr[1,4].set_title('Mean')
gr[1,4].axis('off')

plt.savefig("Resultado.jpg")
plt.show()
```

### ***Resultados de Impresión:***

A continuación, se muestran los resultados. Se pueden observar las diferentes funciones de umbrales que se utilizaron:



### ***Conclusiones:***

Me pareció una herramienta muy interesante para discriminar colores o para separar la imagen para poder cambiar sus colores para que sean los óptimos en los siguientes procesos que se quieran hacer con la imagen.

### ***Enlace de GitHub:***

<https://github.com/IsaacGutierrezCETI/Practica-5.-Umbrales>