



UNIVERSIDAD DEL BÍO-BÍO

14 Umbralización escala de grises

Tutorial Procesamiento de Imagen con webcam

Estudiantes Practicantes:

Javiera Gutiérrez

Javiera Henríquez

Teresa Vidal

Profesor:

Luis Vera

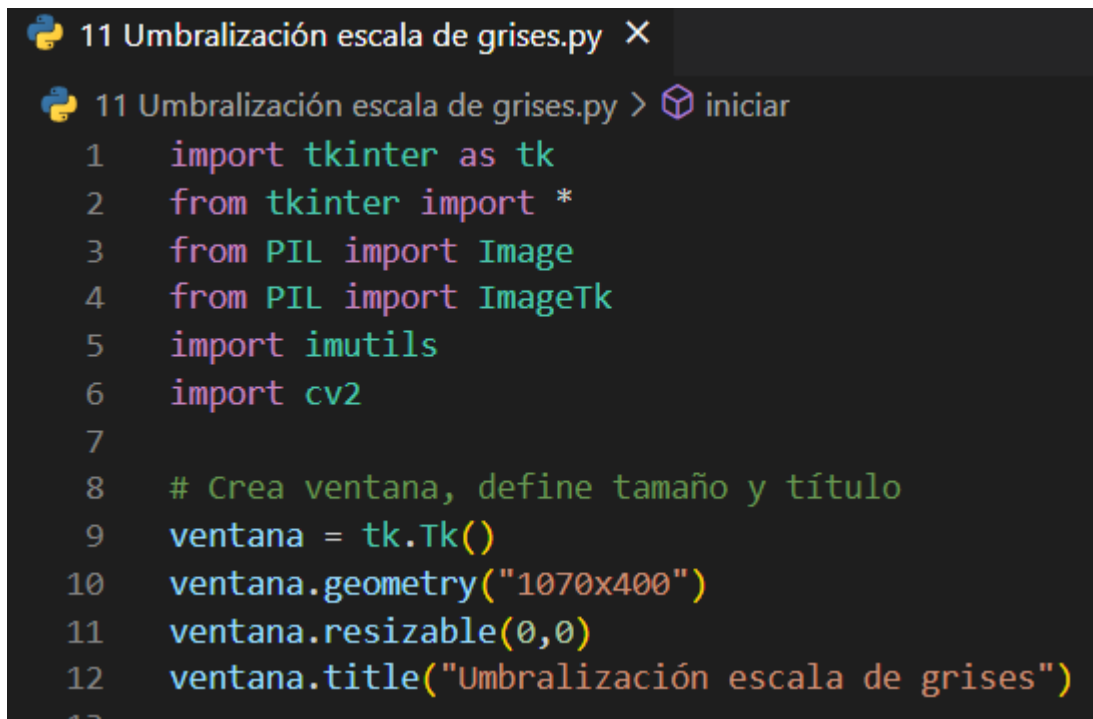
Laboratorio CIMUBB

Umbralización escala de grises

Para umbralizar una imagen que está en escala de grises se usa un proceso diferente, ya no se piden sus tres valores RGB, si no, solo un valor que puede ser entre 0 a 255.

Creamos un nuevo archivo .py y añadimos las siguientes librerías.

Añadimos el código para crear nuestra ventana, y le damos un tamaño, que no se pueda modificar y título.

A screenshot of a code editor window titled "11 Umbralización escala de grises.py". The editor shows a Python script with the following code:

```
1  import tkinter as tk
2  from tkinter import *
3  from PIL import Image
4  from PIL import ImageTk
5  import imutils
6  import cv2
7
8  # Crea ventana, define tamaño y título
9  ventana = tk.Tk()
10 ventana.geometry("1070x400")
11 ventana.resizable(0,0)
12 ventana.title("Umbralización escala de grises")
```

Creamos las funciones para iniciar la cámara web y para tomar una foto en escala de grises, ambos códigos vistos anteriormente en el tutorial.

```

14 #Funciones cámara web
15 def camara():
16     global capture
17     capture = cv2.VideoCapture(0)
18     iniciar()
19
20 def iniciar():
21     global capture
22     if capture is not None:
23         ret, frame = capture.read()
24         if ret == True:
25             frame = imutils.resize(frame, width=311)
26             frame = imutils.resize(frame, height=241)
27             ImagenCamara = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
28             im = Image.fromarray(ImagenCamara)
29             img = ImageTk.PhotoImage(image= im)
30             LImagen.configure(image= img)
31             LImagen.image = img
32             LImagen.after(1,iniciar)
33         else:
34             LImagen.image = ""
35             capture.release()
36
37 #Función para tomar una foto
38 def Capturar():
39     global CapturaG
40     camara = capture
41     return_value, image = camara.read()
42     frame = imutils.resize(image, width=301)
43     frame = imutils.resize(frame, height=221)
44     CapturaG = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
45     imG = Image.fromarray(CapturaG)
46     imgG = ImageTk.PhotoImage(image= imG)
47     LImagenROI.configure(image= imgG)
48     LImagenROI.image = imgG
49

```

Creamos la función **umbralizacion()**, obtenemos el valor entre 0 y 255 que nos da el usuario y lo definimos como mínimo, quedando 255 como máximo. Se le aplica la Umbralización a la imagen en escala de grises y se muestra en el tercer cuadro gris de nuestra ventana.

```

50 def umbralizacion():
51     global thresh1
52     valor = int(numeroUmbra.get())
53     ret, thresh1 = cv2.threshold(CapturaG, valor, 255, cv2.THRESH_BINARY)
54     Umbral = Image.fromarray(thresh1)
55     Umbral = ImageTk.PhotoImage(image=Umbral)
56     ImagenUmbra.configure(image = Umbral)
57     ImagenUmbra.image = Umbral
58

```

Se crea un **SpinBox** que es donde el usuario ingresará su valor, validamos que solo pueda elegir valores entre 0 y 255.

```
59  #SpinBox
60  numeroUmbra = tk.Spinbox(ventana, from_=0,to=255)
61  numeroUmbra.place(x=800, y=331, width=42, height=23)
```

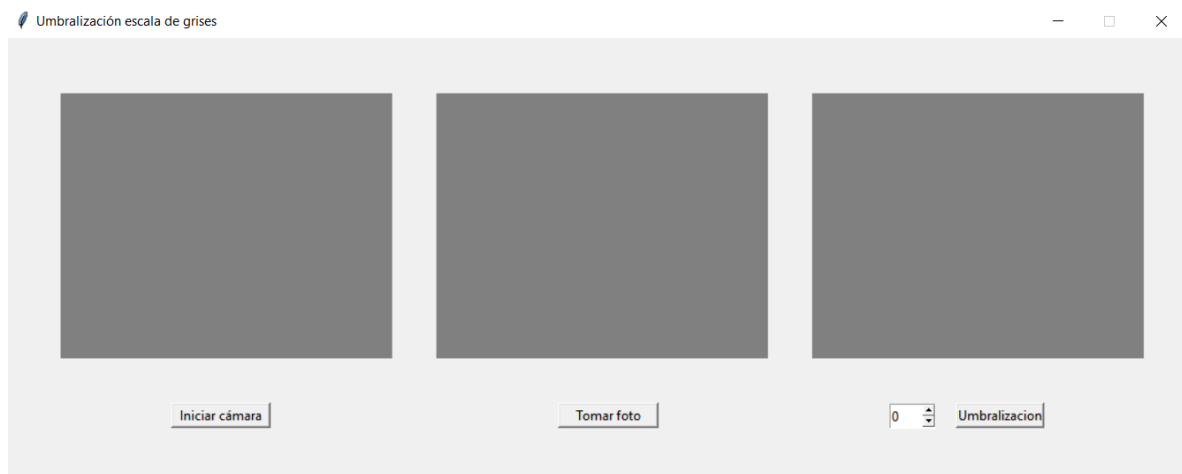
Se añaden tres botones para iniciar la cámara, tomar una foto y para umbralizar la imagen.

Se añaden tres cuadros grises para mostrar la webcam, desplegar la captura que tomamos y el tercero para mostrar la imagen umbralizada.

Finalmente, escribimos **ventana.mainloop()** para que nuestro programa se muestre correctamente.

```
63  #Botones
64  BCamara = tk.Button(ventana, text="Iniciar cámara", command=camara)
65  BCamara.place(x=150,y=330,width=90,height=23)
66  BCapturar = tk.Button(ventana, text="Tomar foto", command=Capturar)
67  BCapturar.place(x=500,y=330,width=91,height=23)
68  Umbra = tk.Button(ventana, text="Umbralizacion", command=umbralizacion)
69  Umbra.place(x=860,y=330,width=80,height=23)
70
71  #Cuadros de imagen gris
72  LImagen = tk.Label(ventana, background="gray")
73  LImagen.place(x=50,y=50,width=300,height=240)
74  LImagenROI = tk.Label(ventana, background="gray")
75  LImagenROI.place(x=390,y=50,width=300,height=240)
76  ImagenUmbra = tk.Label(ventana, background="gray")
77  ImagenUmbra.place(x=730,y=50,width=300,height=240)
78
79
80  ventana.mainloop()
```

Una vez compilado y ejecutado, nuestro programa se verá así:



Y al umbralizar una captura se verá así:

