

Taller 3
Ejercicios en Imágenes

Natalia Prada Sosa
Cristian Martinez
Laura Daniela Parga
Maria Jose Calderon

Programación de computadores

Diego Camilo Otalero Osorio

Universidad Nacional de Colombia

Bogota D.C

1. Explicación de procesos.

- Cree una imagen de un canal, es decir con una forma (alto, ancho) del tamaño que desee, donde la diagonal principal hacia arriba sea blanca y desde la diagonal hacia abajo sea negra.

Primero se crean variables que servirán posteriormente, la variable *ancho* que determina el tamaño de la imagen (20) y la variable *imagen* que está vacía pero se irá llenando con la variable *fila* a medida que el código se ejecute.

Se usa un ciclo for anidado con un range de (0 a ancho) para recorrer los elementos de la imagen, tanto las *filas* como las *columnas*, a medida que estas variables van tomando valores del ciclo se evalúan dos situaciones, la primera cuando las *filas* son menores o iguales a la *columnas* y la segunda cuando las *filas* son mayores a las *columnas*.

Cuando se cumple la primera condición se usan unas variables de colores (*rojo, verde, azul*) que se les asignan el valor 0 ya que eso determina el color negro y esta información se guarda en la variable *pixel*, esta variable llenará a la variable *fila* a través de la función `.append`; pero si por el contrario se cumple la segunda condición sucederá el mismo proceso con la variación de que las variables de colores se les asignará un valor de 255 lo que determina el color blanco.

Luego se crea otra variable llamada *imagen_2* a la cual se le asigna el valor de *imagen*, posteriormente a través de la función `plt.imshow(imagen_2)`, se muestra la imagen y se imprime la matriz con `print`.

- Use una imagen de la librería *skimage*, en la base de datos “data”. e implemente alguno de los siguientes procesos: **Suavizar y Resaltar contornos**

Primero se importa de la librería *skimage* la opción *feature*, de *skimage.color* se importa *rgb2gray* y se importa *matplotlib.pyplot* en la variable *plt*.

Se inicia la variable *image* dándole el valor de una imagen proveniente de *data*, luego a la variable *imagen* se le aplica la opción *rgb2gray* que le da a la imagen solo en la escala de colores rojo, verde y azul, esta se guarda en la variable *img*.

Posteriormente se crea otra variable llamada *borde* que guardará la información de `feature.canny(img)`, lo que quiere decir que tomará las características de la imagen para reducir el ruido y luego busca los gradientes en la intensidad de los píxeles, lo que permite ver los contornos de la imagen potenciados.

Luego se usa la función `plt` que permite visualizar la imagen que está guardada en la variable *borde*.