

Introducción al procesamiento de imágenes

Ejercicio 1: creación de vectores y matrices

1.1 Crea un vector fila de 10 elementos con valores enteros aleatorios (*random*) entre 0 y 100 y realiza las siguientes operaciones sobre el vector creado:

- a) accede a la tercera posición;
- b) obtén el vector formado por las 5 primeras posiciones del vector original;
- c) obtén el vector formado por las posiciones impares del vector original;
- d) suma todos los elementos del vector.

1.2 Crea 2 matrices 'a' y 'b', la primera de dos filas y 3 columnas (2x3): [1 2 3; 4 5 6] y la segunda de (3x2): [2 4; 5 7; 9 3], a continuación realiza las siguientes operaciones:

- a) multiplica las dos matrices;
- b) convierte la matriz 'a' en una nueva matriz 'a2' (3x2) utilizando la función RESHAPE (utiliza la función help en la Command Window para utilizar RESHAPE correctamente); qué diferencia hay entre la matriz a2 y la traspuesta de a?
- c) suma las matrices 'a2' y 'b' y guarda el resultado en la variable 'c';
- d) multiplica, elemento por elemento, las matrices 'a2' y 'b'.

Ejercicio 2: creación de imágenes

Crea una función `edgeImg()` que implemente los siguientes puntos:

- a) crear una imagen de tamaño 256x256 px de un solo canal de tipo 8-bit unsigned mitad blanca y mitad negra, tal como se muestra en Figura 1;
- b) visualizar la imagen y guardarla como `edge_img.jpg`.

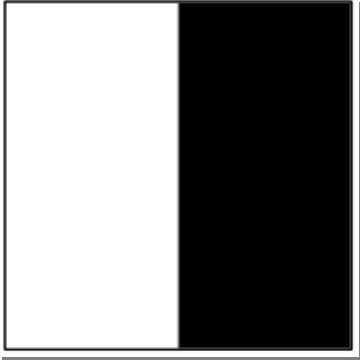


Figure 1: Edge Image

Ejercicio 3: tratamiento de imágenes en escala de grises

Dada la imagen en escala de grises hand_1C.jpg, crear la función `rotatelm()` que implemente los siguientes puntos:

- abrir el archivo y guardarlo en una variable;
- visualizar la imagen;
- visualizar en una gráfica de tipo 'plot' los valores de los niveles de gris de la fila 200 en el rango de columnas [310, 330];
- convertir la imagen original de manera que resulte especular respecto al eje vertical (Fig. 2 a > Fig. 2 b);
- rotar la imagen original de 90° en dirección contraria a las agujas del reloj;
- guardar las imágenes creadas en d) y e) y guardarlas con el nombre `hand_flip.jpg` y `hand_rot.jpg`, respectivamente.

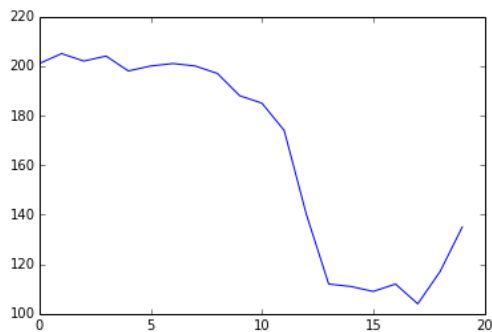
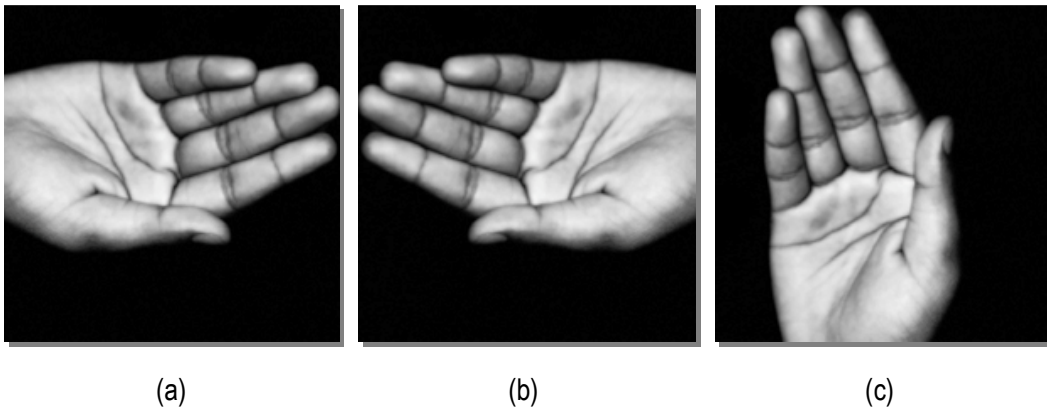


Figura 2

Ejercicio 4: binarización de imágenes

La binarización $B_I(x,y)$ de una imagen $I(x,y)$ a partir de un umbral TH consiste en convertir la imagen en una imagen binaria (de 0s y 1s), dependiendo de si el nivel de intensidad de cada pixel de la imagen original es mayor o menor al umbral TH :

$$B_I(x,y) = \begin{cases} 0, & \forall(x,y): I(x,y) < Th \\ 1, & \forall(x,y): I(x,y) \geq Th \end{cases}$$

Dada la imagen `hand_1C.jpg`, crear la función $B = \text{thresholdImg}(I, th)$ que aplica el umbral th a la imagen I para crear su versión binaria B (como en Figura 3).

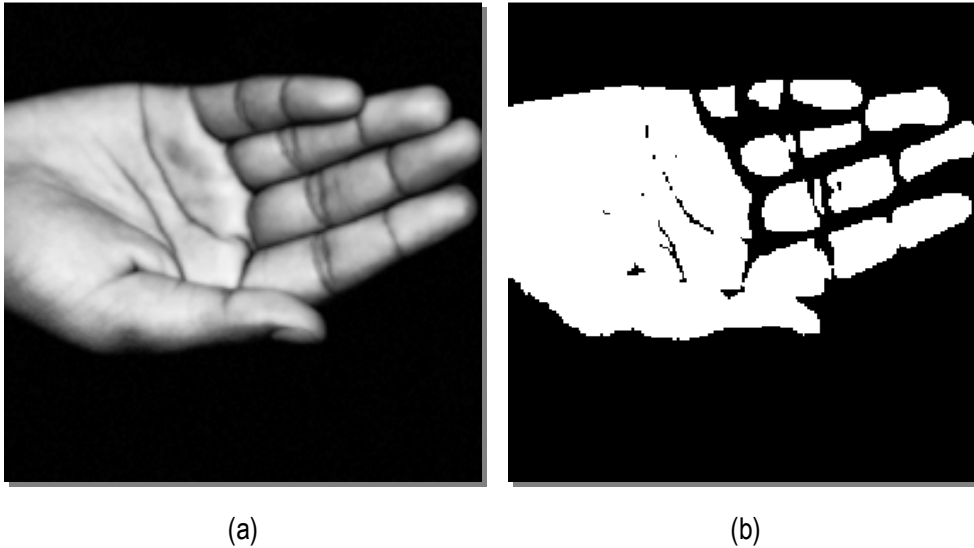


Figura 3: ejemplo de imagen binaria cuando se aplica el valor del umbral $TH = 128$.

Ejercicio 5: creación de imágenes de 3 canales (en color)

Las imágenes RGB están formadas por 3 matrices, llamadas comúnmente canales. Como ejercicio práctico para familiarizar con las imágenes de 3 canales, implementar los siguientes puntos:

- crea las imágenes en escala de grises mostradas en Figura 4 (a-c) (tamaño 128x128 px);
- combina las 3 imágenes de forma que se obtenga la imagen mostrada en la Figura 4d.

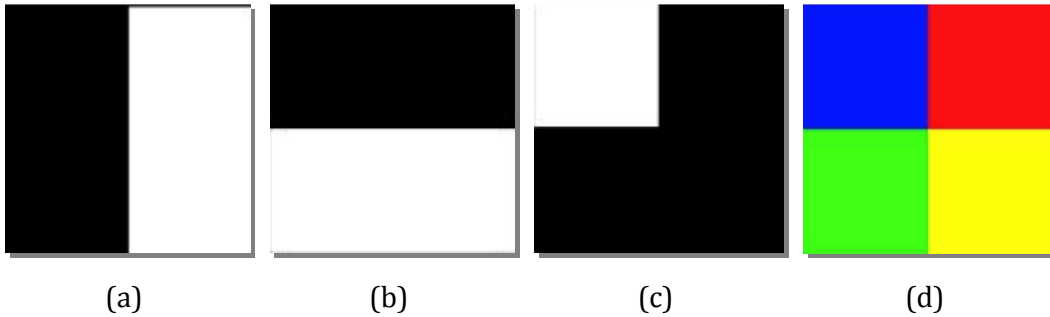


Figure 4: formación de una imagen de 3 canales (RGB)

Ejercicio 6: tratamiento de imágenes en color RGB

Dados los dos archivos de imagen hand.jpg (Figura 5 a) y mapfre.jpg (Figura 5 b), crear la función `fuselmg()` que implemente los siguientes puntos:

- abrir los dos archivos
- convertir la imagen hand.jpg en escala de grises utilizando la función `rgb2gray`;
- realizar la binarización sobre la imagen resultante en b) para conseguir 2 regiones: una perteneciente a la mano (foreground) y la otra al fondo (background);
- utilizar la matriz binaria creada en c) para fusionar las imágenes hand y Mapfre (Fig. 5 c)



(a)

(b)

(c)

Figura 5

Esta practica no se tiene que entregar.