

PRÁCTICA 1

“Introducción al tratamiento de imágenes digitales en Matlab”

Objetivos.

Familiarizarse con el entorno, la programación y el tratamiento matricial de imágenes en Matlab.

Imagen a utilizar: “P1.jpg”.

Instrucciones a utilizar: `imfinfo`, `whos`, `imread`, `max`, `imtool`, `imshow`, `imwrite`, `stem`, `imhist`

Ejercicios

1. Con la instrucción `imfinfo` de Matlab obtén información de la imagen de la práctica (anchura en píxeles, altura en píxeles, profundidad de bit).
2. Lee la imagen y guárdala en una variable de Matlab (matriz) de nombre `Imagen1`. Para ello, haz uso de la instrucción `imread`.
3. Con la instrucción `whos` obtén la siguiente información de la variable Matlab `Imagen1`: tipo de dato y rango.
4. Calcula el mayor elemento de la variable `Imagen1` (máximo nivel de intensidad). Utiliza para ello la instrucción `max`.
5. Calcula en Matlab la imagen complementaria de `Imagen1`, denominándola `Imagen2`. Se debe calcular de forma independiente la imagen complementaria de cada canal de la `Imagen1` y después, componer la `Imagen2`.

6. Crea una matriz, de nombre `Imagen3`, con los niveles de rojo de la imagen `Imagen1`. Esta nueva matriz es una imagen en niveles de gris. Utiliza la función `imadjust` con la configuración `ImagenSalida=imadjust (ImagenEntrada, [], [], gamma)` para, mediante la modificación del parámetro `gamma`, obtener una imagen `Imagen4` más clara (`gamma = 0.5`, por ejemplo) y una imagen `Imagen5` más oscura (`gamma = 1.5`, por ejemplo) que `Imagen3`. Visualiza estas imágenes. Representa el histograma de estas tres imágenes mediante la función Matlab `imhist`. Interpreta cualitativamente los resultados de la operación realizada.
7. Utiliza la función `imabsdiff`, para crear una nueva imagen `Imagen6` que refleje el valor absoluto de la diferencia de `Imagen4` e `Imagen5`. Interpreta los resultados. Realiza la misma operación sin utilizar la función `imabsdiff` y comprueba que obtienes los mismos resultados. Para ello, implementa y utiliza la siguiente función que permite saber si dos matrices son iguales:

```
varLogica = funcion_compara_matrices(M1, M2)
```

donde `varLogica` es una variable lógica indica si `M1` y `M2` son iguales (valor `true`) o diferentes (valor `false`).

8. Implementación de histograma de una imagen:
- a. Implementa una función que tenga como objetivo calcular el histograma de una imagen de intensidad `I`. La función debe devolver un vector `h` con la información numérica del histograma.

```
h = función_histograma(I)
```

Deben implementarse dos versiones de la función: la primera, que realiza un recorrido por cada píxel de la imagen para generar el histograma; la segunda, que realiza un recorrido por cada posible nivel de gris que puede estar presente en la imagen de entrada.

- b. Aplica la función anterior para generar y visualizar el histograma de la componente verde de la imagen de la práctica.

- c. Comprueba que obtienes los mismos resultados que genera la función Matlab `imhist`. La comprobación debe realizarse visualmente, representando los histogramas en una misma gráfica, y numéricamente, utilizando la función `funcion_compara_matrices`.