

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.

Documentación y material disponible:

1. Sobre Matlab
 - a. Documentación facilitada en Práctica 0
 - b. www.mathworks.com
 - c. Ayuda de la herramienta
2. Sobre *Image Processing*
 - a. Ayuda de la herramienta

Imágenes a utilizar: “P1_1.jpg”.

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

Ejercicios

1. Con la instrucción `imfinfo` de Matlab obtén la siguiente información de la imagen “P1_1.jpg”: anchura en píxeles, altura en píxeles, tipo de imagen y profundidad de bit.
2. Lee la imagen y guárdala en una variable de Matlab de nombre `Imagen1`.
3. Visualiza esta imagen con la instrucción `imtool` y con la instrucción `imshow`. Familiarízate con los entornos gráficos de salida de cada una de estas instrucciones.
4. Con la instrucción `whos` obtén la siguiente información de la variable Matlab `Imagen1`: tipo de dato y rango.

5. Utilizando la instrucción `max`, calcula el valor máximo de la variable `Imagen1` (máximo nivel de intensidad).
6. Calcula en Matlab la imagen complementaria de `Imagen1`, denominándola `Imagen2`. Visualiza esta imagen y guárdala en un fichero de imagen del mismo formato que la imagen original empleando la instrucción `imwrite`.
7. Crea y visualiza una matriz, de nombre `Imagen3`, con los niveles de rojo de la imagen `Imagen1`. Notar que esta nueva matriz es una imagen en niveles de gris.
8. Utiliza la función `imadjust` con la configuración `ImagenSalida=imadjust (ImagenEntrada, [], [], gamma)` para, mediante la modificación del parámetro `gamma`, obtener una imagen `Imagen4` más clara (asignar `gamma=0.5`) y una imagen `Imagen5` más oscura (asignar `gamma=1.5`) que `Imagen3`.

Visualiza estas imágenes y representa su histograma mediante la función Matlab `imhist`. Interpreta cualitativamente los resultados de la operación realizada.

9. Utiliza la función `imabsdiff`, para crear una nueva imagen `Imagen6` que refleje la diferencia absoluta 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 el contenido de dos matrices de la misma dimensión es el mismo:

```
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`)

10. 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`.