

# Bloque 1: Operadores puntuales

## MATERIAL EXPERTOS

### Experto 3: Introducción al histograma, estirado vs ecualizado.

### *Histograma*

El histograma de una imagen permite observar cuantitativamente el efecto de los operadores puntuales; de ahí el interés de saber representarlo e interpretarlo. La función que ofrece MatLab para este fin es `imhist`, a la que hay que pasar como parámetros la imagen en cuestión y su VLT (si ésta está disponible).

Puede encontrar una extensa descripción de la función `imhist`, incluyendo parámetros potenciales, requisitos de entrada y ejemplos invocando el comando:

```
doc imhist
```

, en la consola de Matlab.

### *Estirado de histogramas*

El estirado o normalización de histograma es un proceso por el cual se cambia el rango de intensidad de los píxeles de la imagen. Aunque la transformación puede no ser lineal, la forma más común de estirar el histograma es aplicando la transformación:

$$s_k = T(r_k) = (r_k - m) \frac{nM - nm}{M - m} + nm$$

Donde  $m$  y  $M$  suelen representar los límites del rango de la imagen original  $r_k$ ,  $k \in [m, M]$

Por otro lado,  $nm$  y  $nM$  representan los límites del rango tras el procesado, es decir  $s_k$ ,  $k \in [nm, nM]$ .

Nótese que aunque los nuevos límites pueden fijarse libremente, en general  $m \geq nm, nm = 0$  y  $M \leq nM, nM = 255$ .

### *Ecualizado o igualación de histogramas*

Por otro lado, como han visto en la parte teórica de la asignatura, la igualación o ecualización de histograma es un proceso distinto que, si bien también resulta en el estirado del histograma, tiene como efecto principal el producir una uniformización del histograma, es decir cambiar su forma de manera que el histograma resultante tienda a ser *plano o equi-distribuido*. Este proceso, en general, incrementa globalmente el contraste de la imagen.

Como han visto en teoría, la transformación a realizar para la ecualización de histograma viene definida por la expresión, tomando el histograma normalizado como estimador de la *f.d.p.*

$$p_r(r_k) = \frac{np_k}{np}, s_k = T(r_k) = \sum_{i=0}^k p_r(r_i) = \sum_{i=0}^k \frac{np_i}{np}$$

, donde  $np_k$  es la altura del histograma en el nivel  $k$  y  $np = \sum_k np_k$ .

Para realizar los sumatorios, Matlab dispone de la función `cumsum`, una extensa descripción de esta función, incluyendo parámetros potenciales, requisitos de entrada y ejemplos invocando el comando:

`doc cumsum`

, en la consola de Matlab.

Nótese que el rango de  $s_k$  será en general:  $s_k \in [0,1]$ , por lo tanto, se han de escalar los valores de  $s_k$  al intervalo de números enteros  $[0, L-1]$ ,  $L = 256$ .

Para mejorar la visualización de la imagen ecualizada, al ser ésta una imagen discreta, suele realizarse un estirado posterior sobre la imagen ecualizada para que el rango de la imagen sea el total disponible (generalmente  $[0, 255]$ ).

### ***Ecualizado o igualación de histogramas para imágenes en color***

La aplicación de estas técnicas sobre imágenes en color no debe realizarse individualmente para cada banda, puesto que esto puede derivar en cambios en la temperatura del color de la imagen.

Para realizar adecuadamente este proceso, lo mejor es:

- i. Cambiar a un espacio de color que aísle en un solo canal la información de luminancia, intensidad o brillo (YCbCr, Luv, Lab, HSV, HSI,...).
- ii. Aplicar la técnica deseada sobre este canal.
- iii. Volver al espacio cromático original (generalmente RGB).

Experimente con la función `imhist`, para ello puede hacer uso de las imágenes demo de Matlab:

```
ima1 = imread('pears.png');  
ima2 = imread('rice.png');
```

, puede convertirlas a escala de grises mediante la instrucción:

```
ima1_g = rgb2gray(ima1);
```

Puede pasarlas a tipo double invocando al comando `ima1d = im2double(ima1);`, y obtendrá el resultado escalado entre 0 y 1.

También puede invocar al comando `ima1d = double(ima1);`, y obtendrá el resultado escalado entre 0 y 255:

Puede encontrar una imagen indexada en el material de prácticas de la asignatura:

`edificio_bw_512.bmp`.