Bloque 1: Operadores puntuales

MATERIAL EXPERTOS	**Experto 2: Aproximaciones para la realización de operadores puntuales**

La aplicación de un operador sobre una imagen que denominaremos *original* ($\psi[x,y]$), da lugar a una nueva imagen que denominaremos *procesada* ($\theta[x,y]$). A efectos de esta práctica, asumiremos que un operador puntual queda definido por una transformación *T* que toma como entrada el valor de un píxel de la imagen original para generar un nuevo valor en el píxel homólogo de la imagen procesada:

$$\psi[x,y]/\psi[x_n,y_m]=r_k \xrightarrow{T} \theta[x,y]/\theta[x_n,y_m]=s_k=T(r_k),$$
$$m\in\{0,...,M-1\}, n\in\{0,...,N-1\}, k\in\{0,...,L-1\}$$

, donde *MxN* es el tamaño de la imagen original, r_k son los valores que toman los píxeles de la imagen original, s_k los que toman los píxeles de la imagen transformada y *L* el número de posibles valores o niveles de intensidad.

Este operador, por lo tanto, se limita a modificar la amplitud de cada píxel. En conclusión, la transformación llevada a cabo por el operador quedará totalmente definida por la relación entre los valores r_k de la imagen original y los valores s_k de la imagen procesada:

$$s_k=T(r_k), \quad k\in\{0,...,L-1\}$$

Mediante MatLab, podemos diseñar al menos tres estrategias para la realización de operaciones puntuales:

1. Modificación directa de los píxeles de la imagen.
2. Modificación del vector de valores de la imagen.
3. Modificación de la VLT (*Video Lookup Table*).

Comparemos las tres aproximaciones en función de los resultados obtenidos para cada una de ellas y en términos de eficiencia o coste computacional. Nótese que la tercera estrategia sólo podrá realizarse si la imagen de entrada es indexada, es decir tiene mapa asociado VLT.

Sea la transformación deseada:

$s_k=T_c(r_k)=\min(r_k+c,L-1)$, para distintos valores de $c=\{8,32,64,128\}$.

Pueden describirse tres estrategias a seguir para realizar esta transformación

Modificación directa de los píxeles de la imagen (opción 1)

Este es el modo más inmediato y a su vez el más ineficaz de implementar un operador puntual; sin embargo, en ocasiones es el único modo posible de hacerlo. Consiste en modificar uno por uno el valor de todos y cada uno de los píxeles de la imagen original, aplicándoles la transformación *T*.

Consiste en aplicar directamente la transformación sobre las matrices que definen la imagen original (ima) y la imagen procesada (ima_proc). Es la opción más directa, pero puede resultar difícil de aplicar si la función de transformación es una función definida por tramos:

```
ima_proc=min(ima+c,L-1);
```

Modificación del vector de valores de la imagen (opción 2)

Consiste en definir primero el vector de valores r_k y calcular el vector de valores transformados s_k. A continuación, la imagen procesada se obtiene utilizando la original para referenciar los índices del vector de valores transformados.

Esta opción permite definir con relativa sencillez cualquier función de transformación:

```
r=[0:L-1];
s=min(r+c,L-1);
ima_proc=s(ima+1);      % Sumamos 1 porque el valor 0 corresponde a s(1)
```

Modificación del vector de la VLT (opción 3)

Este es el modo más eficaz de implementar un operador puntual, pero sólo es válido en el caso de imágenes indexadas, es decir, imágenes cuyos píxeles referencian posiciones de una VLT.

Sea $R=\{r_0,...,r_k,...,r_{L-1}\}$ la VLT de la imagen original. Los valores de sus píxeles serán referencias o posiciones de esta VLT. Por lo tanto, una imagen representada de este modo es posible formalizarla según: $\psi[x,y]/\psi[x_n,y_m]=k,\ \ k\in\{0,...,L-1\}$.

Dado que la transformación que lleva a cabo un operador puntual sólo afecta a los valores de los píxeles, y éstos están definidos por la VLT, bastará con aplicar la transformación a la VLT.

En conclusión, **<u>la imagen procesada será igual a la imagen original</u>**, pero su VLT, *S*, se habrá modificado:

$$\begin{array}{ccc} \psi[x,y]/\psi[x_n,y_m]=k & \xrightarrow{T} & \theta[x,y]/\theta[x_n,y_m]=k=\psi[x_n,y_m] \\ R=\{r_0,...,r_k,...,r_{L-1}\} & \xrightarrow{T} & S=\{s_0,...,s_k,...,s_{L-1}\}/\ s_k=T(r_k) \end{array}$$

Para modificar una VLT debe tener en cuenta que en MatLab cada valor r_k de una VLT es en realidad una matriz fila $\mathbf{r_k}$ de tres valores de tipo **`double`**, que indican respectivamente la cantidad de las componentes roja, verde y azul del color que representa:

$\mathbf{r_k} = [r_k^R \quad r_k^G \quad r_k^B],\ 0 \le r_k^i \le 1$

En el caso de VLTs en escala de grises, las tres componentes tienen igual valor.

Para llevar a cabo la transformación de la VLT, deberá aplicarla sobre la terna de valores de cada matriz fila: $\mathbf{s_k} = T(\mathbf{r_k})$. Para obtener resultados similares a las aproximaciones anteriores, escale previamente los valores de cada componente, p.e. r_k^R, multiplicándolos por *L-1=255* (ya que en la VLT toman valor entre 0 y 1, no entre 0 y L-1).

Experimente con las diferentes técnicas para la realización de operaciones puntuales, para ello puede hacer uso de las imágenes demo de Matlab:

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

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

```
edificio_bw_512.bmp.
```

,para cargar la imagen y su mapa invoque:

```
[ima_i,map]  = imread('edificio_bw_512.bmp');
```

Puede pasarlas a tipo double invocando al comando:

```
ima1d = im2double(ima1);
```

, en este caso el rango de la imagen cambiará a [0,1], alternativamente, puede realizar un casting manteniendo el rango mediante el comando

```
ima1d = double(ima1);
```