

<b>Práctica 3: Operadores puntuales</b>		<b>Grupo</b>	
		<b>Puesto</b>	
<b>Apellidos, nombre</b>		<b>Fecha</b>	
<b>Apellidos, nombre</b>			

El objetivo de esta práctica es presentar al alumno las técnicas para realizar operadores puntuales sobre imágenes en escala de grises.

Desarrolle cada ejercicio en un fichero de comandos ‘ejercicio\_X.m’ separado. Para conocer el funcionamiento preciso de los comandos que se introducen en este gui3n, utilice la ayuda de MATLAB. Para evitar posibles interferencias con otras variables o ventanas recuerde incluir siempre las instrucciones `clear all` y `close all` al principio de cada fichero de comandos.

Justo antes de finalizar la pr3ctica, comprima el gui3n y los ficheros ‘.m’ generados en un 3nico fichero con el nombre ‘Practica\_3\_ApellidosNombre.zip’, con3ctese al sistema de entrega de pr3cticas de la Intranet y entr3guelo en el grupo de pr3cticas (grupo 3nico).

## 1 Aproximaciones a la realizaci3n de operadores puntuales

La aplicaci3n 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 pr3ctica, asumiremos que un operador puntual queda definido por una transformaci3n  $T$  que toma como entrada el valor de un p3xel de la imagen original para generar un nuevo valor en el p3xel hom3logo de la imagen procesada:

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

, donde  $M \times N$  es el tama1o de la imagen original,  $r_k$  son los valores que toman los p3xeles de la imagen original,  $s_k$  los que toman los p3xeles de la imagen transformada y  $L$  el n3mero de posibles valores o niveles de intensidad.

Este operador, por lo tanto, se limita a modificar la amplitud de cada p3xel. En conclusi3n, la transformaci3n llevada a cabo por el operador quedar3 totalmente definida por la relaci3n entre los valores  $r_k$  de la imagen original y los valores  $s_k$  de la imagen procesada:

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

### 1.1 Ejercicio 1: modificaci3n directa de los p3xeles de la imagen

Este es el modo m3s inmediato y a su vez el m3s ineficaz de implementar un operador puntual; sin embargo, en ocasiones es el 3nico modo posible de hacerlo. Consiste en modificar uno por uno el valor de todos y cada uno de los p3xeles de la imagen original, aplic3ndoles la transformaci3n  $T$ .

Cargue y visualice la imagen de prueba `edificio_bw_512.bmp`<sup>1</sup> e indique si se trata o no de una

---

<sup>1</sup> Desc3rguese esta imagen de la p3gina *web* de la asignatura.

imagen indexada, si por lo tanto viene o no acompañada de una VLT y, en este caso, cuál es su número de niveles,  $L$ :

Implemente, modificando el valor de cada píxel de la imagen, la transformación:

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

Para modificar el valor de cada píxel existen al menos dos opciones:

- Opción 1: 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<sup>2</sup>, pero puede resultar difícil de aplicar si la función de transformación es una función definida por tramos:

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

- 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)
```

Obtenga las cuatro imágenes procesadas, dos utilizando la opción 1 y las otras dos utilizando la opción 2. Represente las cuatro imágenes obtenidas en una nueva figura (utilice `subplot` de 2 filas x 2 columnas y `subimage`, pasando a ésta función como parámetro la VLT de la imagen procesada, que es la misma que la de la imagen original). Observe y describa cuál es el efecto de esta transformación:

Represente en una nueva figura las cuatro funciones  $s_k = T_c(r_k)$ ,  $k \in \{0, \dots, L-1\}$  que ha llevado a cabo (utilice `subplot` y `plot`) y reproduzcalas en la siguiente tabla indicando los puntos más representativos:

---

<sup>2</sup> Recuerde que para poder operar con una imagen, ésta ha de ser de tipo `double` (si no lo es, utilice la función `double()`).


Observe que aunque en MatLab la operación realizada puede expresarse con una única línea de código, el cálculo de la imagen procesada exige a MatLab aplicar la transformación a todos los elementos o píxeles de la imagen original.

Para estimar el tiempo que tarda MatLab en realizar la operación, incluya al final de su código una instrucción `tic`; (almacena en una variable interna el tiempo actual) y a continuación incluya un bucle que realice 1000 veces una misma transformación. Tras el bucle incluya la instrucción `toc`; que imprimirá por la línea de comandos el tiempo transcurrido en segundos desde que se ejecutó la instrucción `tic`. Repita la operación con la imagen 'edificio\_bw\_1024.bmp' y rellene la tabla adjunta:

Tiempo estimado con imagen de 512x512 (s)	Opción 1:
	Opción 2:
Tiempo estimado con imagen de 1024x1024 (s)	Opción 1:
	Opción 2:

## 1.2 Ejercicio 2: modificación de la VLT

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  $R = \{r_0, \dots, r_k, \dots, r_{L-1}\}$  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:  $\psi[x, y] / \psi[x_n, y_m] = k, k \in \{0, \dots, L-1\}$ . 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, la imagen procesada será igual a la imagen original, pero su VLT,  $S$ , se habrá modificado:

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

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:

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

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:  $\mathbf{s}_k = T(\mathbf{r}_k)$ .

El objetivo es repetir el ejercicio 1 siguiendo esta aproximación. Para ello, en primer lugar aplique las cuatro transformaciones en él indicadas y represente las cuatro imágenes obtenidas (observe que en este caso a cada función `subimage` se le pasará la misma imagen original pero una VLT distinta).

Represente asimismo las cuatro funciones de transformación. Basta con representar la transformación de una de las tres componentes (por ejemplo  $s_k^R = T(r_k^R)$ ). Para obtener resultados similares, escale previamente los valores  $r_k^R$  multiplicándolos por  $L-1=255$  (ya que en la VLT toman valor entre 0 y 1, no entre 0 y  $L-1$ ). Consigne nuevamente el resultado obtenido:


Para estimar, comparativamente con el ejercicio 1, el tiempo que tarda MatLab en realizar la operación, incluya de nuevo al final de su código una instrucción `tic`; y un bucle que realice 1000 veces una misma transformación. Mida el número de segundos que tarda el programa en llevar a cabo el bucle. Repita la operación con la imagen 'edificio\_bw\_1024.bmp'.

Tiempo estimado con imagen de 512x512 (s)	
Tiempo estimado con imagen de 1024x1024 (s)	

Comente los resultados comparativos obtenidos en este apartado:

### 1.3 Ejercicio 3: representación de histogramas.

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.

Incluya al final de los ejercicios 1 y 2 (elimine previamente los bucles de estimación de tiempos) el código necesario para representar en una misma figura (utilice `subplot`) los histogramas de las cuatro imágenes procesadas en cada caso. Reproduzca aproximadamente los histogramas obtenidos; observe y justifique el aspecto relativo de cada uno de ellos, de acuerdo con la transformación realizada:

Histogramas del ejercicio 1		Histogramas del ejercicio 2	
Comentarios:		Comentarios:	

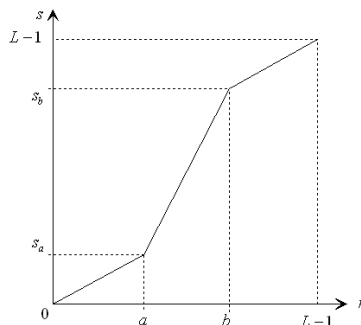
Indique finalmente, a la vista de los resultados de estos tres ejercicios, qué método utilizaría, y por qué, para aplicar un operador puntual:

## 2 Operadores puntuales

### 2.1 Ejercicio 4: ajuste de contraste por tramos.

Según se ha visto en las explicaciones teóricas, la expresión general de una operación de ajuste de contraste por tramos rectos viene dada por:

$$s = \begin{cases} \alpha \cdot r & 0 \leq r < a \\ \beta \cdot (r - a) + s_a & a \leq r < b \\ \gamma \cdot (r - b) + s_b & b \leq r < L \end{cases}$$



La figura presenta el aspecto de la transformación para unos valores concretos de sus parámetros, que modifica por tramos los valores de los píxeles de la imagen original, comprimiendo unos rangos ( $[0, a]$  y  $[b, L-1]$  en este ejemplo) y expandiendo otros ( $[a, b]$ ).

El fin de este ejercicio es mostrar la aplicación principal de esta transformación: el ajuste o realce de una imagen poco contrastada. Para poder observar el efecto tanto visualmente como en su histograma, utilice la primera aproximación (ejercicio 1) de realización de operadores puntuales.

Cargue la imagen de prueba `edificio_bc_512.bmp`<sup>3</sup>. Observe si se trata o no de una imagen indexada, si por lo tanto viene o no acompañada de una VLT y, en este caso, cuál es su número de niveles,  $L$ . Represente la imagen y su histograma e indique, observando éste, cuál es la característica que provoca el que una imagen tenga un bajo contraste:

Aplique la transformación de ajuste de contraste con parámetros:  $a = 80, b = 160, s_a = 30, s_b = 220$ . A la hora de obtener los  $L$  valores  $s_k = T(r_k)$ , tenga en cuenta que éstos deben ser números enteros, ya que representan índices de la VLT (para forzarlo, haga `s=round(s)` tras definir la transformación). Represente la imagen procesada, su histograma y la función de transformación. Indique, observando todos ellos, cuál ha sido el efecto visual de la transformación sobre la imagen, cómo se traduce este efecto en su histograma, y cómo la función de transformación consigue el efecto deseado:

---

<sup>3</sup> Descárguese esta imagen de la página *web* de la asignatura.

Intente ahora conseguir el efecto contrario: tome como imagen original la imagen `edificio_bw_512.bmp`<sup>4</sup> y aplíquelo una operación de ajuste de contraste con los parámetros necesarios para reducir su contraste (en vez de ampliarlo), es decir, para obtener una imagen similar a `edificio_bc_512.bmp`.

Indique los valores que ha utilizado para los cuatro parámetros de la transformación y dibuje dicha función:

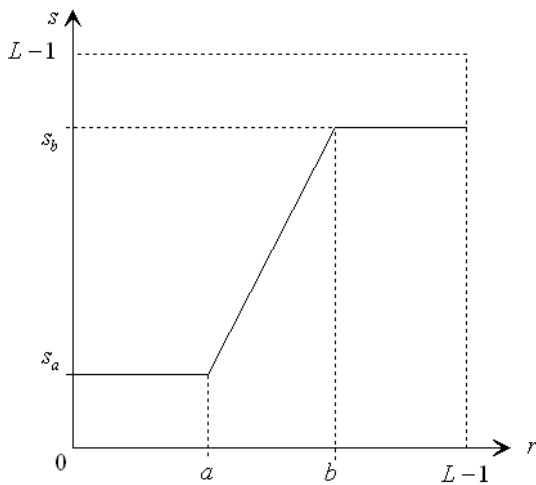
$a :$		$b :$		$s_a :$		$s_b :$	
-------	--	-------	--	---------	--	---------	--



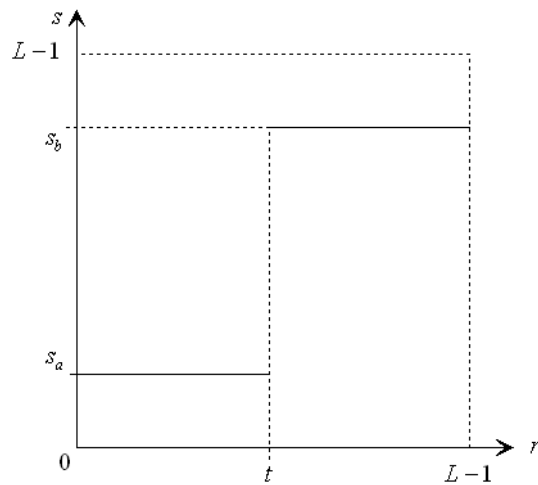
## 2.2 Ejercicio 5: recorte y umbralización.

Según se ha visto en las explicaciones teóricas, las operaciones de recorte y umbralización se realizan con transformaciones que pueden definirse con las curvas:

Recorte



Umbralización



<sup>4</sup> Descárguese esta imagen de la página *web* de la asignatura.

Para observar el efecto del recorte, cargue la imagen `cuadro_bw_512.bmp`<sup>5</sup> y visualice tanto la imagen como su histograma. Observe que en el histograma es posible identificar cuatro crestas claramente diferenciadas, correspondientes a otros tantos grupos de píxeles de la imagen. El primer objetivo de este ejercicio es utilizar esta transformación para identificar a qué grupo de píxeles corresponde cada cresta.

Para ello, diseñe cuatro operadores puntuales de recorte que mantengan inalterados los valores de los píxeles de cada una de las cuatro crestas (escoja valores de  $a$  y  $b$  en torno a cada una de ellas de modo que los cuatro intervalos formen una partición del rango de variación de los píxeles).

Represente en dos figuras de 2x2 subfiguras las cuatro imágenes procesadas y sus cuatro histogramas. Obsérvelas e indique a qué grupos de píxeles corresponde cada una de las crestas (numeradas de izquierda a derecha):

Cresta 1	
Cresta 2	
Cresta 3	
Cresta 4	

A continuación represente y reproduzca las cuatro funciones de transformación:


---

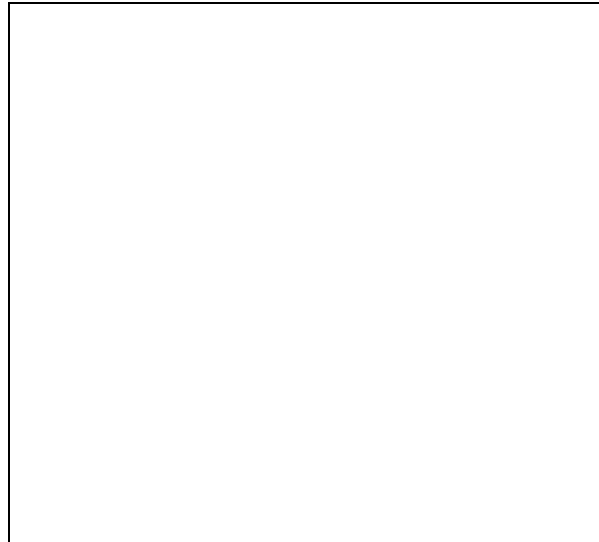
<sup>5</sup> Descárguese esta imagen de la página *web* de la asignatura.



La umbralización es un caso particular del recorte para  $a = b = t$  (donde  $t$  es el umbral). A la vista de los resultados observados al aplicar un operador puntual de recorte en las cuatro crestas de la imagen 'cuadro\_bw\_512.bmp', diseñe una función de transformación que umbralice la imagen asignando un valor  $s_a = 10$  a todos los trazos de la imagen y un valor  $s_b = 240$  al resto de las zonas.

Represente la imagen procesada y su histograma. Represente y reproduzca la función de transformación e indique el valor que ha seleccionado para el parámetro  $t$ .

$t$	
-----	--



### 2.3 Ejercicio 6: otros operadores puntuales.

A raíz de los ejercicios planteados hasta el momento, se asume que el alumno está en disposición de realizar sin especial complicación cualquier transformación puntual cuya expresión se aporte. Acuda a sus apuntes de las clases teóricas e intente realizar sobre cualquier imagen en escala de grises de que disponga las siguientes operaciones:

- Negativo o inversión del eje de intensidades.
- Resalte de una banda concreta de niveles (intente, por ejemplo resaltar los píxeles correspondientes a la tercera cresta del histograma de la imagen 'cuadro\_bw\_512.bmp').