



# GRADO EN INGENIERÍA DE TECNOLOGÍAS DE TELECOMUNICACIÓN

## Compresión de Video

### Ejercicios Prácticos. Bloque 1. Tema 1

Nombre: \_\_\_\_\_

Fecha: \_\_\_\_\_

#### Funciones esenciales de procesamiento de una imagen

##### ➤ Mostrar información sobre una imagen.

```
imfinfo('pout.tif')
```

##### **Ejemplo 1: Generar una imagen que contenga una escala de grises.**

```
for x=1:256
    escalaGrises(1:256,x)=x-1; % Se genera escala de grises
end

% Visualización previa conversión a 1 byte
imtool(uint8(escalaGrises));

% Tools -> Choose Colormap
```

##### ➤ Lectura de una imagen

```
I=imread('pout.tif'); % Se lee una imagen (en este caso B&W)
[fil col dim]=size(I) % Tamaño de esa imagen. Si dim=1 indica monocromática.
imshow(I); % Muestra la imagen
```

##### **Ejemplo 2: La extracción de los planos R, G y B**

```
% Lectura de la imagen
I=imread('peppers.png');
[fil col dim]=size(I)

% Descomposición de las componentes RGB
I_red=I(:,:,1); % Componente R
I_green=I(:,:,2); % Componente G
I_blue=I(:,:,3); % Componente B

% Visualización de la imagen y sus componentes
subplot(2,2,1), imshow(I), title('ORIGINAL RGB');
subplot(2,2,2), imshow(I_red), title('Componente RED');
subplot(2,2,3), imshow(I_green), title('Componente GREEN');
subplot(2,2,4), imshow(I_blue), title('Componente BLUE');
```

### Ejemplo 3: Conversión de una imagen RGB a indexada

```
% Lectura de la imagen
I=imread('peppers.png');
whos I % Se observa el tamaño de I en formato RGB

[X map]=rgb2ind(I,256); % Se convierte RGB a una paleta de 256 colores
whos X map % Se observa el tamaño de ambas variables

% Visualización de las imágenes
subplot(1,2,1), imshow(I), title('ORIGINAL RGB: 2^{(8+8+8)} colores');
subplot(1,2,2), imshow(X, map), title('Imagen indexada: 256 colores');

% Ejecutar en la figura: Insert->ColorBar
```

Existen otras funciones especializadas para la lectura de imágenes:

- ✓ DICOM (*Digital Imaging and Communications in Medicine*) → **dicomread**
- ✓ NITF (*National Imagery Transmission Format*) → **nitfread**
- ✓ HDR (*High Dinamic Range*) → **hdrread**

### ➤ Conversión de datos.

Los formatos de datos más comunes son:

- **uint8**: 1 byte por pixel, rango [0 255]
- **double**: 8 bytes por pixel, rango [0.0 1.0]
- **logical**: 1 byte por pixel, 1 para *true* y 0 para *false*

Funciones de conversión de formato de datos	
Función	Formato de salida
<b>im2single</b>	single
<b>im2double</b>	double
<b>im2uint8</b>	uint8
<b>im2uint16</b>	uint16
<b>im2int16</b>	int16

### Ejemplo 4: Conversión de datos

```
A=[-6.0 4.1; 0 0.5]
whos A % Se observa el tamaño y número de bytes
C=im2uint8(A)
B=uint8(A) % Observar el redondeo
D=mat2gray(A) % Convierte una matriz a escala de grises rango [0.0 1.0]. Asigna el
valor mínimo de la matriz a 0.0 y el máxima a 1.0, escalando el resto de los valores
E=im2bw(D, 0.4) % Binariza la matriz D con valor umbral de 0.4
whos E
```

### Ejemplo 5: Conversión de de una imagen RGB a Imagen Gris

```
% Se lee la imagen
I=imread('peppers.png');

% función de conversión RGB -> Gray
I_gray=rgb2gray(I);

% Visualización de la imagen
imshow(I_gray);
```

## ➤ Visualización

```
imshow('peppers.png'); % Muestra una imagen con un conjunto de herramientas de manipulación
```

Operaciones básicas:

- Extraer el valor de un pixel: Posicionar el puntero sobre la imagen.
- *Tools-> Image Information*: Equivalente a la función **imfinfo**.
- *Tools-> Measure distance*: Medir la distancia entre dos pixels.
- *Tools-> Pixel region*: Valor de los pixels de una zona seleccionada.
- *Tools-> Crop Image*: Recorta una imagen.

## ➤ Escritura en un fichero

```
imwrite(I,nombre,fmt); % I es la matriz, nombre corresponde al nombre del fichero y
fmt los distintos formatos soportados
```

### Ejercicios prácticos 1.

La orden `imwrite(I,'pep75.jpg','quality',75);` permite guardar una imagen I en formato comprimido jpg con factor de calidad 75. Rellenar la siguiente tabla que indica el tamaño de la imagen 'peppers.png' comprimida de acuerdo a diferentes factores de calidad. Visualizar las imágenes para observar los diferentes factores de calidad.

Factor de calidad	Tamaño en bytes
5	
50	
75	
95	

## Procesado de imagen

## ➤ Operaciones de punto

### Ejercicios prácticos 2.

- 1.- Leer la imagen 'cameraman.tif'. Es una imagen monocromática.
- 2.- Realizar el negativo de esa imagen.

- 3.- Realizar una simetría en el eje X
- 4.- Realizar una simetría en el eje Y
- 5.- Realizar una rotación de 90°

La función `imrotate(I,angulo)` permite rotar una imagen `I` un ángulo especificado en grados. Repetir los apartados 3 a 5 anteriores utilizando esta función.

Las operaciones de rotación de matrices se pueden realizar en Matlab con las funciones: *flipdim*, *fliplr*, *flipud* y *rot*.

## ➤ Transformación de intensidad Gamma

### Ejemplo 6: Ajuste de una imagen mediante el parámetro gamma.

```
% Se lee una imagen
I=imread('peppers.png');

% Ajuste GAMMA
I_adj_2 = imadjust(I,[],[],2); % Ajuste gamma=2
I_adj_01 = imadjust(I,[],[],0.1); % Ajuste gamma=0.1

% Visualización de las imágenes
subplot(1,3,1), imshow(I), title('ORIGINAL RGB');
subplot(1,3,2), imshow(I_adj_2), title('\gamma=2');
subplot(1,3,3), imshow(I_adj_01), title('\gamma=0.1');
```

## ➤ Histograma

```
I=imread('cameraman.tif');
subplot(1,3,1), imshow(I), title('ORIGINAL');
subplot(1,3,2), imhist(I,256), axis tight, title('histograma');
```

### Ejemplo 7: Mejora contraste imagen mediante histograma.

```
% Leer la imagen
I=imread('cameraman.tif');

% Imagen ecualizada
I_eq=histeq(I,256);

% Visualización de las imágenes
subplot(2,2,1), imshow(I), title('ORIGINAL');
subplot(2,2,2), imhist(I,256),axis tight, title('Histograma Original');
subplot(2,2,3), imshow(I_eq), title('Imagen ecualizada');
subplot(2,2,4), imhist(I_eq,256),axis tight, title('Histograma Ecualizada');
```

## Ajuste manual del contraste

```
imtool('pout.tif');
Tools-> Adjust Contrast: Escoger un rango de valores de los pixels.
```

## ➤ Transformación espacial

### Ejemplo 8: Transformación espacial.

```
% Se lee una imagen
I=imread('cameraman.tif');

% Se genera una transformación T
T=maketform('affine',[0.94 0.34 0;-0.34 0.94 0; 0 0 1]);
I2=imtransform(I,T); % Se aplica esa transformación T a la imagen

% Visualización de las imágenes
subplot(1,2,1), imshow(I), title('ORIGINAL');
subplot(1,2,2), imshow(I2), title('Imagen transformada');
```

## ➤ Operaciones de entorno

### Ejemplo 9: Operaciones de filtrado

```
% Se lee una imagen
I=imread('peppers.png');

% Definición de diferentes máscaras
f1=1/9*[1 1 1; 1 1 1; 1 1 1]; % Máscara de desenfoque
f2=[0 -1 0; -1 5 -1; 0 -1 0]; % Máscara de enfoque
f3=[1 1 1; 1 -8 1; 1 1 1]; % Laplaciana
f4=[-1 -2 -1; 0 0 0; 1 2 1]; % Máscara de bordes Vertical
f5=transpose(f4); % Máscara de bordes Horizontal

% Filtrado aplicando las máscaras anteriores
I1=imfilter(I,f1);
I2=imfilter(I,f2);
I3=imfilter(I,f3);
I4=imfilter(I,f4);
I5=imfilter(I,f5);

% Visualización de las imágenes
subplot(3,2,1), imshow(I), title('ORIGINAL');
subplot(3,2,2), imshow(I1), title('Desenfoque');
subplot(3,2,3), imshow(I2), title('Enfoque');
subplot(3,2,4), imshow(I3), title('Laplaciana');
subplot(3,2,5), imshow(I4), title('Borde Vertical');
subplot(3,2,6), imshow(I5), title('Borde Horizontal');
```

La función *fspecial* permite crear filtrados de imagen 2D predeterminados: *average*, *disk*, *gaussian*, *laplacian*, *log*, *motion*, *prewitt*, *sobel*, *unsharp*.

### Ejemplo 10: Operaciones de filtrado usando filtros predefinidos

```
% Leer una imagen
I = imread('cameraman.tif');

% Filtrado con movimiento
fm = fspecial('motion',20,45);
Im = imfilter(I,fm,'replicate');

% Filtrado circular
fc = fspecial('disk',10);
```

```

Ic = imfilter(I,fc,'replicate');

% Filtro de contraste
fs = fspecial('unsharp');
Is = imfilter(I,fs,'replicate');

% Visualización de las imágenes
subplot(2,2,1), imshow(I), title('ORIGINAL');
subplot(2,2,2), imshow(Im), title('Filtro motion');
subplot(2,2,3), imshow(Ic), title('Filtro circular');
subplot(2,2,4), imshow(Is), title('Filtro contraste');

```

### Ejemplo 11: Operaciones de filtrado para eliminar ruido

```

% Lectura de la imagen
I=imread('coins.png');

% Se añade ruido
In=imnoise(I,'salt & pepper');

% Filtro especial para eliminar ruido salt&pepper
If= medfilt2(In);

% Visualización de las imágenes
subplot(1,3,1), imshow(I), title('ORIGINAL');
subplot(1,3,2), imshow(In), title('Imagen con ruido');
subplot(1,3,3), imshow(If), title('Imagen filtrada');

```

### Ejercicios práctico 3.

Con objeto de mejorar la calidad imagen, modificar el código anterior para que a la imagen filtrada se aplique un filtro tipo 'unsharp' o de contraste. Observar el resultado.

## ➤ Operaciones aritméticas y lógicas

```

Z= imadd(X,Y); % Suma de dos imágenes X e Y
Z= imsubtract (X,Y); % Resta de dos imágenes X e Y
Z= imabsdiff (X,Y); % Resta en valor absoluto de dos imágenes X e Y
Z= imcomplement (X); % Complementa una imagen X. Imagen negativa.
Z= immultiply (X,Y); % Multiplica la imagen X por otra Y. Y puede ser escalar.

```

Los resultados de las operaciones aritméticas y lógicas se truncan para que cualquier valor se encuentre dentro del rango especificado.

### Ejemplo 12: Suma de dos imágenes

```

% Lectura de imágenes
I = imread('rice.png');
J = imread('cameraman.tif');

% Se suman ambas imágenes
K = imadd(I,J,'uint16');

% Visualización de las imágenes
subplot(1,3,1), imshow(I), title('Imagen 1');
subplot(1,3,2), imshow(J), title('Imagen 2');
subplot(1,3,3), imshow(K,[]), title('Imagen suma');

```

### ➤ Detección de contornos

`Z= edge (I, method); % Detecta el contorno en la imagen I usando diferentes métodos: sobel, prewitt, Roberts, laplacian, zero-corss y canny.`

#### **Ejemplo 13: Detección de contornos**

```
% Lectura de una imagen
I = imread('circuit.tif');

% Dos métodos de detección de bordes
I1 = edge(I,'prewitt');
I2 = edge(I,'canny');

% Visualización de las imágenes
subplot(1,3,1), imshow(I), title('ORIGINAL');
subplot(1,3,2), imshow(I1), title('prewitt');
subplot(1,3,3), imshow(I2), title('canny');
```

#### **Ejercicios prácticos 4.**

- 1.- Cambiar el threshold de la aproximación 'canny' a 0.8 y comprobar su efecto.
- 2.- Sumar la imagen resultante de utilizar la función edge a la imagen original y visualizar el resultado. Para ello, usa la siguiente instrucción que permite convertir una imagen 'logical' en una uint8:

```
I2 = uint8( edge(I,'canny') ) * 255;
```

### ➤ Redimensionado de imágenes

`Z= imresize (I, scale, method); % Modifica el tamaño de una imagen I por un factor de escala usando diferentes métodos de interpolación: nearest, bilinear y bicubic.`

#### **Ejemplo 14: Redimensionado de una imagen**

```
% Lectura de una imagen
I = imread('cameraman.tif');

% Cambio de dimensionado de las imágenes
Ismall = imresize(I, 0.5);
Ilarge1 = imresize(I, 2, 'nearest');
Ilarge2 = imresize(I, 2, 'bilinear');

% Visualización de las imágenes
figure, imshow(I), title('ORIGINAL');
figure, imshow(Ismall), title('Imagen x 0.5');
figure, imshow(Ilarge1), title('Imagen x 2 nearest');
figure, imshow(Ilarge2), title('Imagen x 2 bilinear');
```

### ➤ Medida de la calidad de una imagen

La función `[PSNR MSE E]=errorMedida (I, I')` proporciona parámetros de medida de error entre dos imágenes I e I'. Retorna 3 parámetros numéricos: PSNR, MSE y E. PSNR es el que suele utilizar como referencia.

### Ejemplo 15: Medida de calidad de una imagen

```
% Lectura de una imagen
I=imread('peppers.png');
ImagInfo=imfinfo('peppers.png');
fprintf('\n peppers.png size=%d bytes',ImagInfo.FileSize);

% Conversión a JPEG
imwrite(I,'peppers15.jpg','quality',15);

% Medidas de calidad de las imágenes
ImagInfo=imfinfo('peppers15.jpg'); %Información de la imagen
Ijpg=imread('peppers15.jpg'); % Lectura de la imagen en formato jpeg

% Calculo medida calidad
[PSNR MSE E] = errorMedida(I, Ijpg);
fprintf('\n peppers15.jpg size=%d bytes PSNR=%f dB dB MSE=%f',ImagInfo.FileSize,
PSNR, MSE);

% Visualización de las imágenes
subplot(2,2,1), imshow(I),title('IMAGEN ORIGINAL');
subplot(2,2,2), imshow(Ijpg),title('IMAGEN COMPRIMIDA');
subplot(2,2,3), imshow(abs(I-Ijpg),[]),title('IMAGEN DIFERENCIA');
subplot(2,2,4), imshow(log(1+abs(double(I-Ijpg)))),title('IMAGEN DIFERENCIA
LOG');
```

### Ejercicios práctico 4.

Utilizando como referencia el código del ejemplo 15, rellenar los datos de la siguiente tabla:

Imagen	Factor de calidad	Tamaño (bytes)	MSE	PSNR (dB)	Calidad Subjetiva
peppers.png	N/A	287677	N/A	N/A	Muy Buena
peppers05.jpg	5				
peppers15.jpg	15				
peppers25.jpg	25				
peppers50.jpg	50				
Peppers75.jpg	75				