



# GRADO EN INGENIERÍA DE TECNOLOGÍAS DE TELECOMUNICACIÓN

## Compresión de Video

### Ejercicios Prácticos Bloque 1. Tema 4

---

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

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

#### Ejemplo 1: Visualización de una base 2D DCT

```
M=1000;  
y=0:M-1;  
x=0:M-1;  
  
% Selección de bases  
v=2;  
u=5;  
  
% Creación de la base  
for i=0:M-1  
    for j=0:M-1  
        C(i+1,j+1)=cos((2*i+1)*u*pi/M) * cos((2*j+1)*v*pi/M);  
    end  
end  
  
mesh(C); % Visualización
```

#### Ejercicios prácticos 1.

Repetir el Ejemplo 1 con varios valores de u y v.

#### Ejemplo 2: Proceso de transformada DCT

```
% Lectura de la imagen  
I=imread('lenna.tif'); % Se lee la imagen  
Id=im2double(I); % Se convierte a double  
  
% Procesado DCT  
If=dct2(Id); % Transformada DCT 2-D  
Ii=idct2(If); % Transformada inversa DCT 2-D  
Ip=im2uint8(Ii); % Conversión a uint8  
  
% Visualización de las imágenes  
subplot(1,3,1), imshow(I), title('IMAGEN ORIGINAL');  
subplot(1,3,2), imshow(log(1+abs(If)),[]), title('DCT 2-D');  
subplot(1,3,3), imshow(Ip), title('IMAGEN RECUPERADA');
```

#### Ejemplo 4: Transformada 2D DCT via 1D DCT

```
% Base DCT 8  
T=dctmtx(8); % Matriz de T  
  
f=rand(8,8); % Generación de una matriz aleatoria  
  
% Procesado 2D-DCT en base a dos 1D-DCT  
Fr=T*f; % 1D DCT  
F=T*transpose(Fr); % 1D DCT
```

```

% 2D DCT directa
Fd=dct2(f);

% Visualización coeficientes
disp(F);
disp(Fd);
disp(Fd);

```

### Ejemplo 5: Eliminación de coeficientes 2D DCT

```

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

% Seleccion valor threshold
Umbral=-1;
while (Umbral > 1 || Umbral < 0)
    Umbral = input('Introduce un threshold [0.0 1.0]: ');
end

T=dct2(I); % Transformada directa 2D DCT

% Se seleccionan los coeficientes superiores a un
% umbral. Escala logarímicamente.
Ts=(log(abs(T)) > Umbral*log(max(max(abs(T))))) .* T;

% Contar coeficientes no nulos. Si su valor es 10000 veces menor al valor
% del coeficiente máximo
Tnn=(abs(T) > max(max(abs(T)))/10000.0);
fprintf('\n Imagen original: coeficientes no nulos =%d',sum(sum(Tnn)));
Tns=(abs(Ts) > max(max(abs(T)))/10000.0);
fprintf('\n Imagen procesada: coeficientes no nulos =%d',sum(sum(Tns)));

% Se aplica la transformada inversa
Ii=idct2(Ts);

% Visualización imágenes y resultados
subplot(2,2,1), imshow(I), title('IMAGEN ORIGINAL');
subplot(2,2,2), imshow(mat2gray(Ii)), title('IMAGEN RECUPERADA');
fprintf('\n PSNR=%f', PSNR(I,Ii));
fprintf('\n');

```

### Ejercicios prácticos 2.

Rellenar la tabla inferior utilizando el código del Ejemplo 5.

Umbral	Coeficientes no nulos	PSNR	Calidad de la imagen
0.1			
0.2			
0.3			
0.4			
0.5			
0.6			
0.7			
0.8			
0.9			

#### Ejemplo 6: Ejemplo de aplicación blockproc

```
% Definición de función
fun=@(block_struct) dct2(double(block_struct.data));

% Ejecución operación func a los bloques de 8x8
K=blockproc('lenna.tif',[8 8],fun);

% Visualización de coeficientes
imshow(log(1+abs(K)),[], title('COEFICIENTES DCT'));
```

#### Ejercicios prácticos 3.

Modificar el código del Ejemplo 5 para que la operación de umbral se realice sobre bloques de 8x8. Tener en cuenta que la partición de bloques 8x8 se aplica tanto a la transformada directa como inversa. Rellenar la tabla inferior.

Umbral	Coeficientes no nulos	PSNR	Calidad de la imagen
0.1			
0.2			
0.3			
0.4			
0.5			
0.6			
0.7			
0.8			
0.9			

#### Ejemplo 7: Representación de scale en términos del parámetro quality

```
quality=1:99; % rango de valores de quality

% Cálculo de escale
scale=zeros(1,length(quality)); % Inicialización
scale=scale + (quality > 50) .* (2-2.*quality/100);
scale=scale + (quality <= 50) .* (50 ./ quality);

% Representación gráfica
p=plot(quality,scale,'Color','red','LineWidth',2);
set(p,'Color','red','LineWidth',2);
% set(p,'xlabel','FontSize',12); %,'Fontweight','b');
xlabel('quality','FontSize',12,'Fontweight','b','Color','blue');
xlabel('scale','FontSize',12,'Fontweight','b','Color','blue');
grid on
```

#### Ejemplo 8: Ordenación zig-zap del JPEG

```
% Patron para reordenacion en zig-zag: order
order = [ 1 2 9 17 10 3 4 11 ...
          18 25 33 26 19 12 5 6 ...
          13 20 27 34 41 49 42 35 ...
          28 21 14 7 8 15 22 29 ...
          36 43 50 57 58 51 44 37 ...
          30 23 16 24 31 38 45 52 ...
          59 60 53 46 39 32 40 47 ...
          54 61 62 55 48 56 63 64];
```

```
% Transformada cuantificada
DCTquant=[-3 -2 2 0 0 0 0 0; ...
    4 -1 -1 0 0 0 0 0; ...
    7 2 0 0 0 0 0 0; ...
    1 1 1 0 0 0 0 0; ...
    -3 -1 0 0 0 0 0 0; ...
    -1 0 0 0 0 0 0 0; ...
    0 0 0 0 0 0 0 0; ...
    0 0 0 0 0 0 0 0];
```

#### Ejercicios prácticos 4.

En el Ejemplo 8 se encuentra el array **order** que contiene la ordenación en zig-zag usado por el JPEG. Esta variable especifica el orden de proceso de los coeficientes cuantificados. Utilizar **order** para reordenar la matriz de coeficientes DCTquant usada como ejemplo y generar un array denominado **zigzagOrder** que contenga los coeficientes ordenados de acuerdo al estándar

#### Ejemplo 9: Análisis comparativo JPEB básico -> JPEG standard

```
% Lectura de una imagen
I=imread('peppers.png'); % Se abre la imagen

quality=40; % Se define el factor de calidad de compresión

% Se guarda la imagen MATLAB JPEG
imwrite(I,'peppers.jpg','Quality',quality);

% Se lee la imagen jpeg
Imat=imread('peppers.jpg');
PSNR_JPEGmatlab=PSNR(I,Imat); % Calculo PSNR

% Se procesa con el JPEG básico
[Ibas, PSNR_JPEGbasico]=jpegBasicoFuncion(I, quality);

fprintf('\n PSNR ORIGINAL -> JPEG MATLAB: %5.2f',PSNR_JPEGmatlab);
fprintf('\n PSNR ORIGINAL -> JPEG BASICO: %5.2f',PSNR_JPEGbasico);
fprintf('\n PSNR JPEG MATLAB -> JPEG BASICO: %5.2f\n',PSNR(Ibas,Imat));

% Visualización de ambas imagenes procesadas
compareImagesTool(Ibas,Imat);
```