



# GRADO EN INGENIERÍA DE TECNOLOGÍAS DE TELECOMUNICACIÓN

## Compresión de Video

### Ejercicios Prácticos Bloque 1. Tema 5

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

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

#### Ejemplo 1: Código Huffman

```
% Se definen los simbolos
symbolsTxt={'white','black','red','blue'};
symbols=[1:4];
%Probabilidad
prob = [14/25 6/25 3/25 2/25];

% Generacion deCodigo Huffman
[dict,avglen] = huffmandict(symbols,prob)

% Mostrar el codigo
fprintf('\n-----');
fprintf('\nNombre:Codigo');
fprintf('\n-----');

for x=1:length(symbols)
    fprintf('\n %5s: ',symbolsTxt{x});
    fprintf('%d',dict{x,2});
end
fprintf('\n-----');
fprintf('\n Longitud promedio: %5.3f',avglen);
fprintf('\n-----\n');

sig=[1 2 4 3 1];
cod=huffmanenco(sig,dict)
```

#### Ejercicios prácticos 1.

Realizar la codificación entrópica Huffman generando los códigos de la siguiente tabla utilizando lápiz y papel.

Símbolos	Probabilidad
Q1	0.4
Q2	0.2
Q3	0.12
Q4	0.08
Q5	0.08
Q6	0.08
Q7	0.03
Q8	0.01

Encodificar la siguiente secuencia: Q1Q1Q2Q1Q1Q1Q4Q7Q1Q6Q6Q1Q8  
Comprobar el resultado modificando el código del Ejemplo 1.

#### Ejemplo 4: Proceso de codificación de una imagen

```
%% Proceso de codificación
I = imread('cameraman.tif'); % Se lee la imagen
[M N]=size(I); % Se calcula el tamaño de la matriz
imhist(I,256), axis tight; % Se dibuja el histograma

% Se calcula valores de pixels (symbols) y probabilidad (p)
[p,symbols]=imhist(I);
prob=p/(M*N);

% Se construye la Tabla de Huffman: Código de Huffman
[dict,avglen]=huffmandict(symbols,prob);

% Se convierte una matriz 2D -> 1D (array)
for x=1:M
    A_1D((x-1)*N+1:x*N)=I(x,:);
end
% Codificación huffman de la matriz 1D
cod=huffmanenco(A_1D,dict);

% Compression ratio
fprintf('\nCompression ratio: %4.5f\n', (M*N*8)/length(cod));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Proceso de decodificación
% Decodificación huffman de la matriz 1D
decod=huffmandeco(cod,dict);
% Se convierte una matriz 1D -> 2D size MxN
for x=1:M
    D(x,:)=uint8(decod((x-1)*N+1:x*N));
end
subplot(1,2,1), imshow(I), title('IMAGEN ORIGINAL');
subplot(1,2,2), imshow(D), title('IMAGEN DECODIFICADA');
```

#### Ejercicios prácticos 2.

Observar si la tasa de compresión (compression ratio) aumenta o disminuye si se procesa la imagen 'lenna.tif'.

#### Ejercicios prácticos 3.

En el Ejemplo 4, con la imagen 'spice.tif' realizar las siguientes operaciones:

1. Para resetear las variables, ejecutar el comando para: **clear all**
2. Ejecutar el fichero del **Ejemplo4.m**. Guardar el Compression ratio proporcionado.
3. Comprobar el tamaño de las variables **p** y **symbols** retornadas por la función **imhist(I)**. Para ello, ejecutar **length(p)** y **length(symbols)**. Su valor es 255 porque la imagen utiliza datos **uint8** de 8 bits con un rango de valores de 0 a 255. Al observar el contenido de **p** (escribir **p** en la línea de comandos), se observa que hay muchos valores nulos. Eso significa que no hay pixels con un valor específico.
4. Para eliminar esos valores nulos, creamos una lista **p**, denominada **pr**, y otra para los **symbols** denominada **symbolsr**, mediante las siguientes instrucciones:

```
pr=p(find(p ~= 0));
symbolsr=symbols(find(p ~= 0));
```

Si observamos la dimensión de estas nuevas variables, **length(pr)** y **length(symbolsr)**, es mucho menor que la original debido a que se han eliminado los valores nulos de **p**.

Modificar el código del Ejemplo 5 para que trabaje sobre las variables **pr** y **symbolosr**.