

# Tema 0: Compresión y Recuperación de Información Multimedia

Rafael Molina

Depto. de Ciencias de la Computación  
e Inteligencia Artificial  
Universidad de Granada

# Contenidos

- I. Multimedia
- II. Bases Tecnológicas para la comunicación multimedia
- III. Compresión una Tecnología Posibilitadora
  - I. Compresión ¿Por qué?.
  - II. ¿Por qué es posible la compresión?
  - III. Técnicas de compresión
  - IV. Compresión sin pérdida
  - V. Compresión con pérdida
- IV. Recuperación de Información Multimedia
  - I. Definición
  - II. Problemas de recuperación de información
  - III. Desafíos en el indexado multimedia automático
- V. Bibliografía

# I. Multimedia

***¿Qué lleva asociado las tecnologías multimedia?***

*representación, transmisión y almacenamiento conjunto de medios de comunicación, como texto, audio, imágenes y vídeo, así como su posterior recuperación y utilización en aplicaciones de los datos multimedia.*

## II. Bases tecnológicas para la comunicación multimedia

¿En qué se basa la comunicación multimedia (entendida como comunicación que integra diferentes medios)?

- **Sistemas**

Sistemas que integren audio, vídeo, texto, etc.

- **Codificación**

Algoritmos de compresión para texto, audio, imágenes y vídeo

- **Comunicaciones/Redes**

Transmisión multimedia sobre redes con errores,  
Streaming,  
Calidad del servicio (QoS)

- **Sistemas de Recuperación de la Información Multimedia**

### III Compresión una tecnología posibilitadora

Cualquier avance en las tecnologías posibilitadoras (enabling technologies): desarrollo de sistemas, codificación, desarrollo en redes de comunicaciones y recuperación de la información marcará el avance multimedia.

Dentro del conjunto de tecnologías que afectan al desarrollo multimedia nos concentraremos en este curso en: **la compresión y recuperación de información multimedia.**

Para algunos: **La COMPRESIÓN es LA tecnología posibilitadora detrás de la revolución multimedia que estamos experimentando.**

Existen además otras tecnologías posibilitadores de interés relacionadas, por ejemplo, con los sistemas y las redes.

## III.1. Compresión ¿por qué?.

En los últimos años hemos visto una transformación (o revolución) en la forma que utilizamos para comunicarnos.

Esta transformación incluye: Internet, comunicaciones móviles y sin lugar a duda vídeo.

La compresión de datos es una de las llamadas tecnologías posibilitadoras (enabling technologies) para estos tres elementos que son parte de la revolución multimedia.

Sin compresión no tendría sentido subir imágenes, audio o vídeo a la red, la calidad de las comunicaciones celulares no sería la misma y desde luego la TV digital no sería posible.

Conceptos como JPEG o MPEG no son nuevos para nadie.

La compresión de datos se usa en estos estándares para reducir el número de bits necesarios para representar imágenes, vídeo o música.

Podría decirse que la compresión de datos es *“El arte o la ciencia de representar información de una forma compacta”* [Sayood, 2012 página 1].

Una pregunta:

¿Por qué no nos centramos en el desarrollo de mejores técnicas de transmisión y almacenamiento?.

Con varios ejemplos:

1 segundo de video sin compresión utilizando el formato CCIR 601 que veremos necesita más de 20 MBs.

2 minutos de música con calidad CD sin compresión, (44100 muestras por segundo, 16 bits por muestra, estéreo) necesita más de 20MBs.

ESA, NASA, CSA, JAXA monitorizarán conjuntamente cambios medioambientales. Diariamente producirán Terabytes de datos.

etc. , etc. , etc.



# ¿Por qué comprimir?

Telefonía (220-3400 Hz)	8000 muestras/s x 12 b/muestra = 96 kbps
Habla banda ancha (50-7000 Hz)	16000 muestras/s x 14 b/muestra = 224 kbps
Audio banda ancha (20-20.000 Hz)	44100 muestras/s x 2 canales x 16 b/muestra = 1412 Mbps
Imágenes color	512x512 píxeles x 24 bpp = 768 kBs
Vídeo (CCIR601)	720x480 gris x 8 bpp x 30 frames/s + 2x360x480 chroma píxeles x 8 bpp x 30 frames = 166 Mbps 720x576 gris x 8 bpp x 25 frames/s + 2x360x576 chroma píxeles x 8 bpp x 25 frames = 166 Mbps
HDTV	1280x720 píxeles color x 24 bpp x 60 frames/s = 1.3 Gbps 1920x1080 píxeles color x 24 bpp x 60 frames/s = 2.9 Gbps

CCIR601: Sin compresión podríamos almacenar

- 31 s de video en un CD-ROM (650 MBs)
- 4 minutos de video en un DVD-5 (40 Gbits)

HDTV Terrestre

- 19.3 Mbits/s (canales de 6 MHz)
- Necesita una razón de compresión de ~ 70

**4K, PEOR TODAVÍA: 4096X2160 (Full 4K) 3840x2160 (4K UHDV)**  
**8K, MUCHO PEOR**

Mientras que podemos afirmar que la capacidad de transmisión y almacenamiento crece constantemente, un corolario de la Primera Ley de Parkinson establece:

*las necesidades de transmisión y almacenamiento crecen a una velocidad que es el doble de la mejora en capacidad de transmisión y almacenamiento.*

Primera Ley de Parkinson: “el trabajo crece para llenar todo el tiempo disponible” [Sayood, 2012, página 2]

## Un poco de Historia:

Uno de los primeros ejemplos de compresión de datos es el desarrollado por Samuel Morse a mediados del siglo XIX, (ver también el código Braille) en el que las letras enviadas por el telégrafo son codificadas utilizando puntos (.) y rayas (-) siendo la secuencia de símbolos más corta para las letras más frecuentes. ¿Por qué crees que es esto?

Este tipo de modelos para la compresión son llamados modelos estadísticos.

Existen además modelos que tienen en cuenta el proceso físico de formación del dato (voz) o modelos que analizan la capacidad de percepción del receptor (imágenes).

## III.2. ¿Por qué es posible la compresión?

Consideremos  
una imagen

Redundancia  
espacial

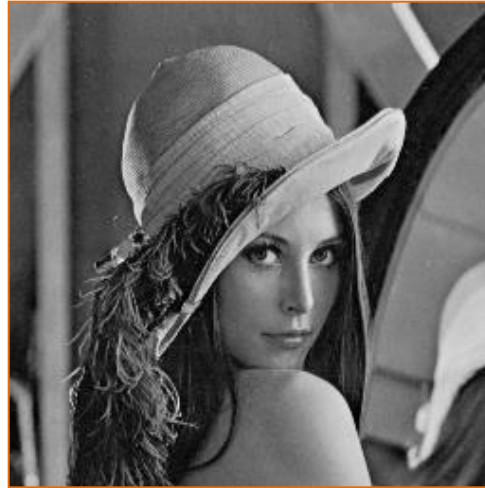


Imagen original  $s[x,y]$

Imagen diferencia  $u_D[x,y] = s[x,y] - s[x-1,y]$

# Redundancia temporal

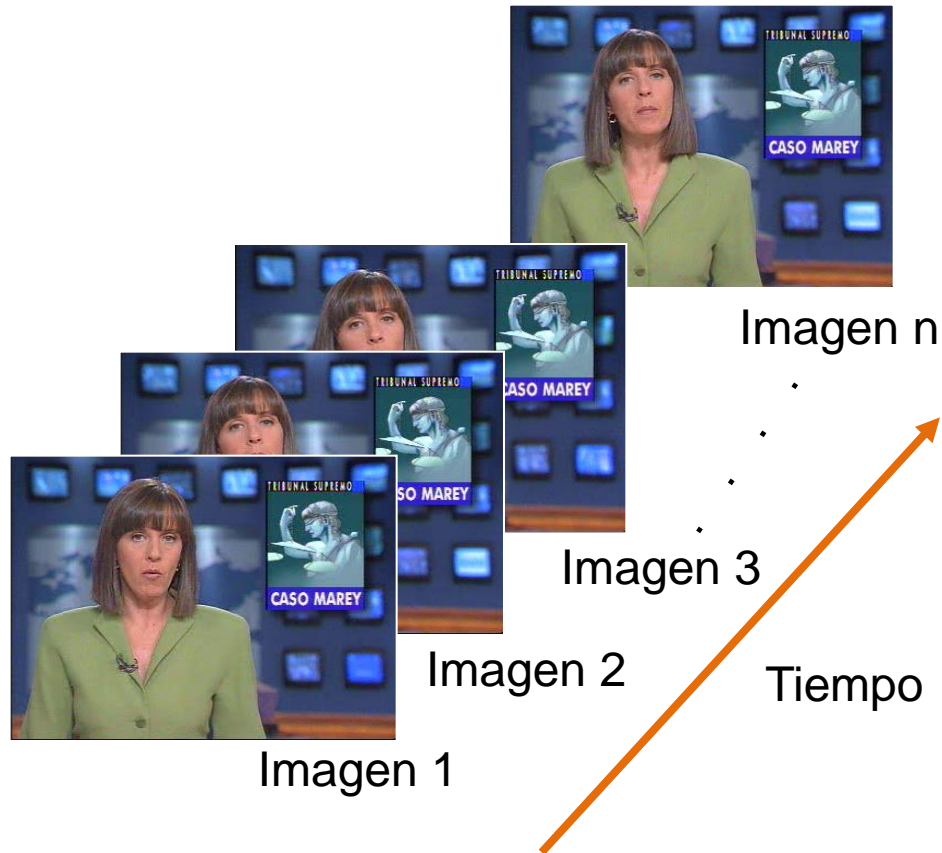


Imagen 1  
(25 imágenes/seg)



Imagen 2  
(25 imágenes/seg)

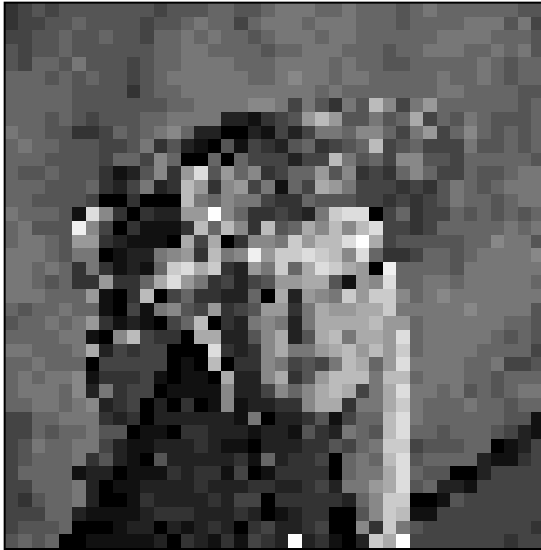
# Formas obvias de comprimir una imagen o vídeo

- Reducir la frecuencia de muestreo
  - Menos píxeles por línea y menos líneas
  - Menos imágenes por segundo (vídeo)
- Reducir el número de niveles de amplitud en la señal muestreada
  - Pulse coded modulation – PCM

**Obviamente se usan técnicas más avanzadas**

# Ejemplos de muestreo

- Reducir la frecuencia de muestreo (espacial)



40x40



80x80

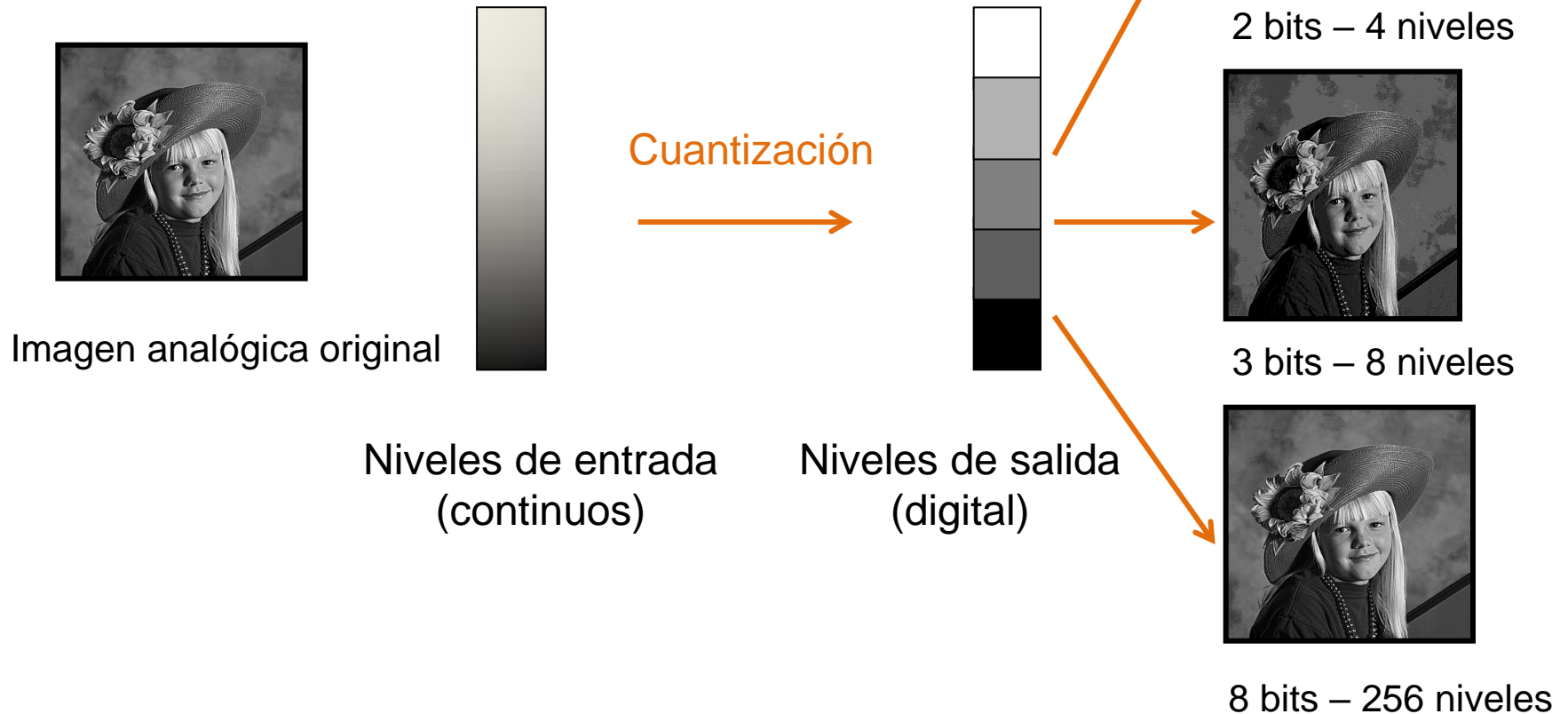


324x324



# Cuantización de imagen

- Reducir el número de niveles de amplitud en la señal muestreada





# Reducción del número de valores de representación



Imagen original  
24 bits/pixel - 16777200 colores diferentes

# Reducción del número de valores de representación



Imagen codificada con 8 bits/pixel  
256 colores diferentes. Factor de compresión 3



# Reducción del número de valores de representación



Imagen codificada con 6 bits/pixel  
64 Colores diferentes. Factor de compresión 4

# Reducción del número de valores de representación



Codificada con 4 bits/píxel  
16 colores diferentes. Factor de compresión 6

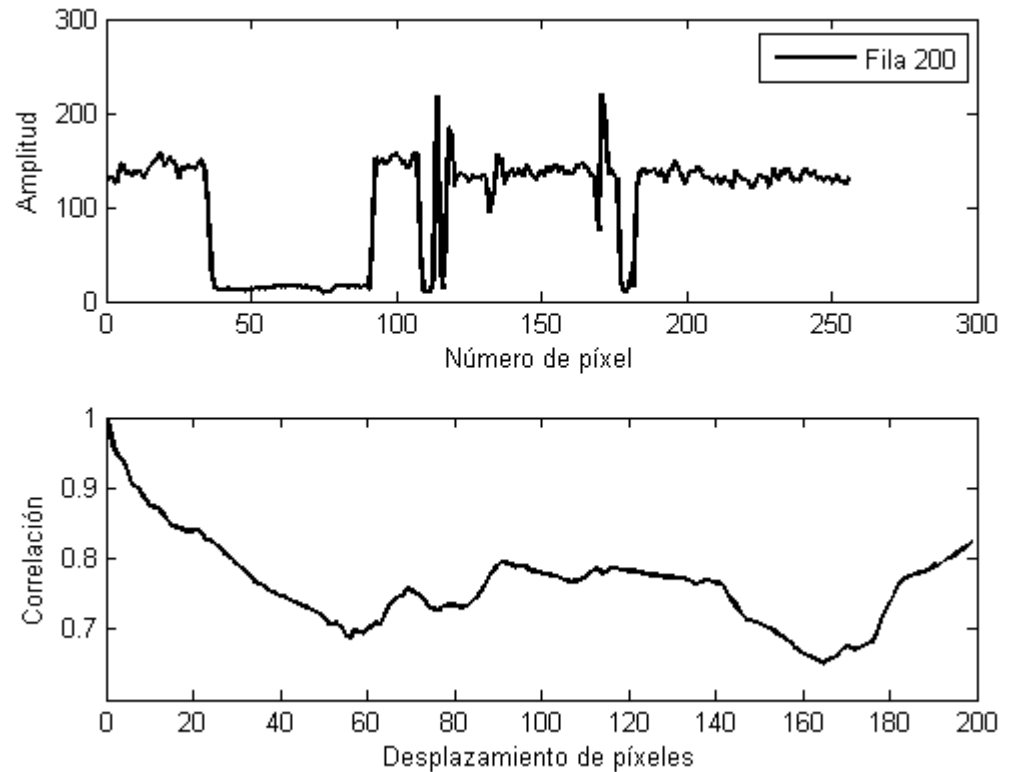
# Reducción del número de valores de representación



Imagen codificada con 2 bits/píxel  
2 colores diferentes. Factor de compresión 24

# ¿Por qué es posible reducir el espacio que ocupa una imagen?. Ejemplos.

Imagen de entrada



## Ejecuta el contenido del guion figura0\_1

A la izquierda tenemos la imagen original y a la derecha la fila 200 y debajo la correlación entre píxeles en función de la distancia entre los píxeles, en esa fila.

Observa, un píxel se parece mucho al que está a su lado (su correlación normalizada es casi uno). Esta es la base de Differential Pulse Code Modulation (DPCM)

```

% Guion figura0_1.m
% Dibuja, para una línea de una imagen, sus niveles de gris y la
% correlación

clear; close all
I = imread('cameraman.tif');
figure,imshow(I),title('Imagen de entrada')
%
Row = 200; % fila de la que dibujaremos su perfil
x = double(I(Row,:));
Col = size(I,2);
%
MaxN = 200; % número de puntos de correlación que calcularemos
Cor = zeros(1,MaxN); % almacena los valores de correlación
for k = 1:MaxN
    v=x(1:Col-(k-1));
    u=x(k:Col) ;
    Cor(k) = u*v'/(norm(u)*norm(v));
end
figure,subplot(2,1,1),plot(1:Col,x,'k','LineWidth',2)
xlabel('Número de píxel'), ylabel('Amplitud')
legend(['Fila' ' ' num2str(Row)])
subplot(2,1,2),plot(0:MaxN-1,Cor,'k','LineWidth',2)
xlabel('Desplazamiento de píxeles'), ylabel('Correlación')

```

# Veamos ahora un ejemplo basado en JPEG

Original Image



DCT compressed Image

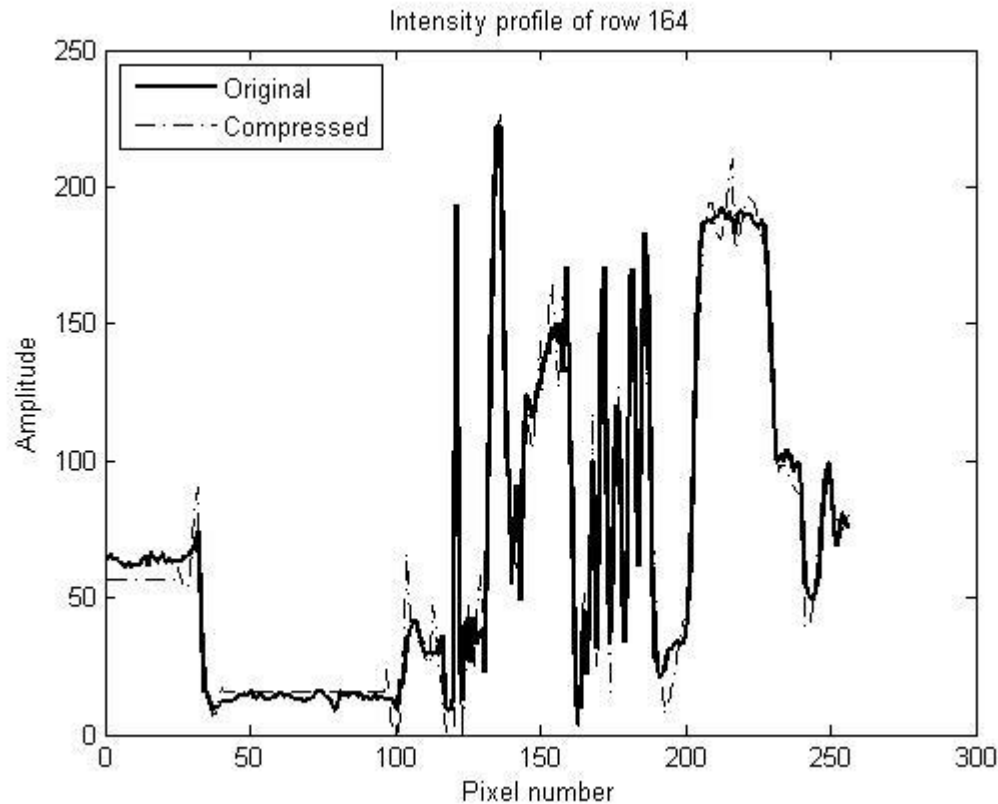


## Ejecuta el contenido del guion figura0\_2

A la izquierda tenemos la imagen original y a la derecha su versión comprimida usando un modelo simplificado de JPEG. En el código se observa como la utilización de una tabla de cuantización produce la compresión.

Fíjate en los artificios de compresión





Guion figura0\_2

Observa que hay diferencias importantes entre la imagen original y la comprimida.

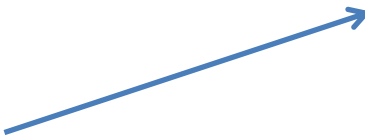
Lee el código y observa que la compresión está basada en la transformación de cuadrados 8x8 de la imagen y en su cuantización

```

% Script figura0_2.m
% Example to show blockiness in DCT compression
% Quantizes and dequantizes an intensity image using
% 8x8 DCT and JPEG quantization matrix
close all
clear
I = imread('cameraman.tif');
figure,imshow(I), title('Original Image')
%
fun = @dct2; % 2D DCT function
N = 8; % block size of 2D DCT
T = blkproc(I,[N N],fun); % compute 2D DCT of image using
NxN blocks
%
Scale = 4.0; % increasing Scale quantizes DCT coefficients
heavily
% JPEG default quantization matrix
jpgQMat = [16 11 10 16 24 40 51 61;
12 12 14 19 26 58 60 55;
14 13 16 24 40 57 69 56;
14 17 22 29 51 87 80 62;
18 22 37 56 68 109 103 77;
24 35 55 64 81 194 113 92;
49 64 78 87 103 121 120 101;
72 92 95 98 121 100 103 99];

```

Cambia la escala y  
observa lo que pasa

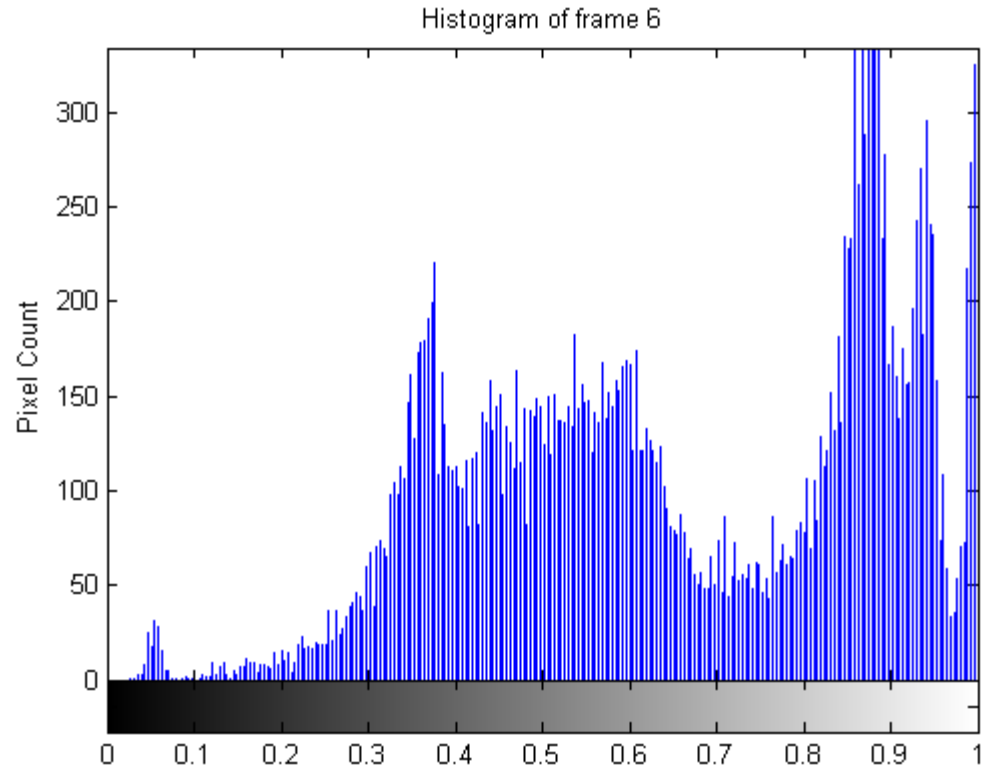


```

Qstep = jpgQMat * Scale; % quantization step size
% Quantize and dequantize the coefficients
for k = 1:N:size(I,1)
    for l = 1:N:size(I,2)
        T1(k:k+N-1,l:l+N-1) = round(T(k:k+N-1,l:l+N-1)./Qstep).*Qstep;
    end
end
% do inverse 2D DCT
fun = @idct2;
y = blkproc(T1,[N N],fun);
y = uint8(round(y));
figure,imshow(y), title('DCT compressed Image')
% Plot image profiles before and after compression
ProfRow = 164;
figure,plot(1:size(I,2),I(ProfRow,:),'k','LineWidth',2)
hold on
plot(1:size(I,2),y(ProfRow,:),'-k','LineWidth',1)
title(['Intensity profile of row ' num2str(ProfRow)])
xlabel('Pixel number'), ylabel('Amplitude')
legend('Original','Compressed')

```

# El último ejemplo que vamos a ver utiliza una secuencia de vídeo



B

Ejecuta el código del guión figura0\_3

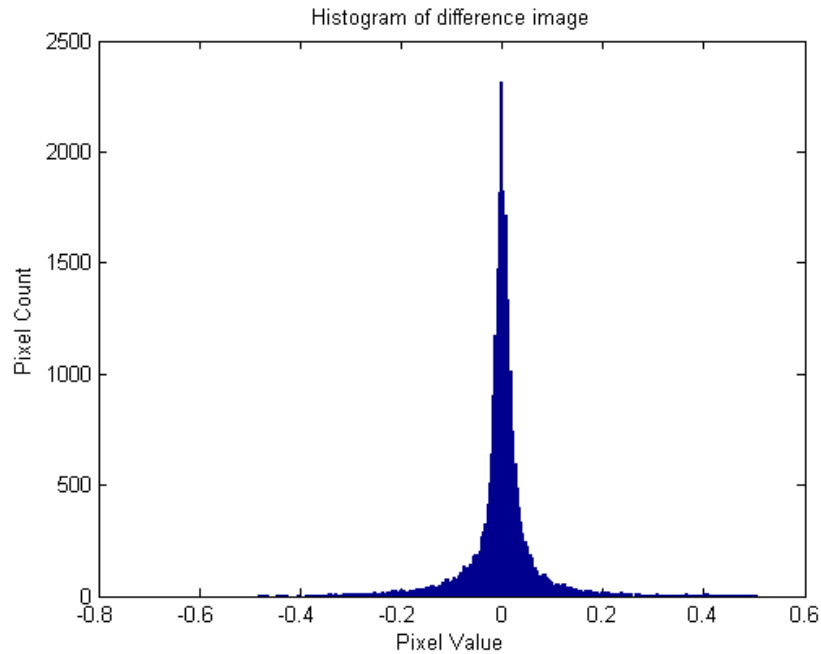
Dos imágenes consecutivas (A y B) y el histograma de la segunda (B).

Vamos a utilizar la predicción de la segunda imagen, B, a partir de la primera, A, para comprimir el espacio que ocupa la segunda

```

% Script figure0_3.m
% generates a differential frame by subtracting two
% temporally adjacent intensity image frames
% quantizes the differential frame and reconstructs
% original frame by adding quantized differential frame to the other frame.
close all; clear
file_name = 'foreman.yuv';
file_format = 'QCIF_PAL';
num_of_fotograma = 25;
[yuv_movie, yuv_array] = readYUV(file_name, num_of_fotograma, ...
file_format);
imshow(yuv_movie);
%%
I1=frame2im(yuv_movie(5));
I2=frame2im(yuv_movie(6));
Frm1=I1(:,:,1);
Frm2=I2(:,:,1);
%%
I1 = im2single(Frm1); % convert from uint8 to float single
I2 = im2single(Frm2); % convert from uint8 to float single
figure,imshow(I1,[]), title([num2str(5) 'th frame'])
figure,imshow(I2,[]), title([num2str(6) 'th frame'])
figure,imhist(I2,256),title(['Histogram of frame ' num2str(6)])
xlabel('Pixel Value'), ylabel('Pixel Count')

```



Difference image



Quantized Difference image



### Guion figura0\_3

Hemos construido a continuación el histograma de la diferencia entre las dos imágenes B-A y hemos cuantizado dicha diferencia.

```
%%  
Idiff = imsubtract(I2,I1); % subtract frame 5 from 6  
figure,hist(Idiff(:),256),title('Histogram of difference image')  
xlabel('Pixel Value'), ylabel('Pixel Count')  
figure,imshow(Idiff,[]),title('Difference image')  
% quantize and dequantize the differential image  
IdiffQ = round(Idiff/4)*4;  
figure,imshow(IdiffQ,[]),title('Quantized Difference image')
```

### Guion figura0\_3

Por último le hemos sumado a A la diferencia cuantizada. Observa, en la siguiente transparencia, donde no coinciden

```
%%  
y = I1 + IdiffQ; % reconstruct frame 6  
figure,imshow(y,[]),title('Reconstructed image')
```

6th frame



Reconstructed image

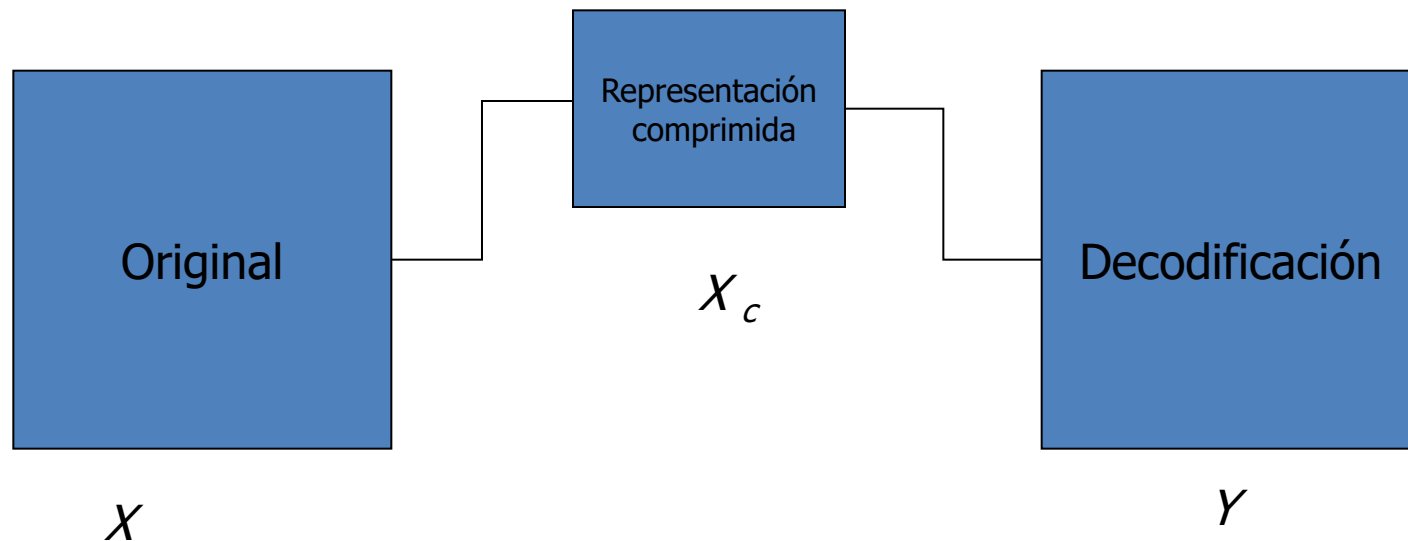




## III.3. Técnicas de compresión

Cualquier algoritmo o técnica de compresión tiene dos partes:

- Un **algoritmo de compresión** que toma una entrada  $X$  y genera una representación  $X_c$  que necesita menos bits.
- Un **algoritmo de reconstrucción** que trabaja en la representación comprimida  $X_c$  y genera la decodificación  $Y$ .



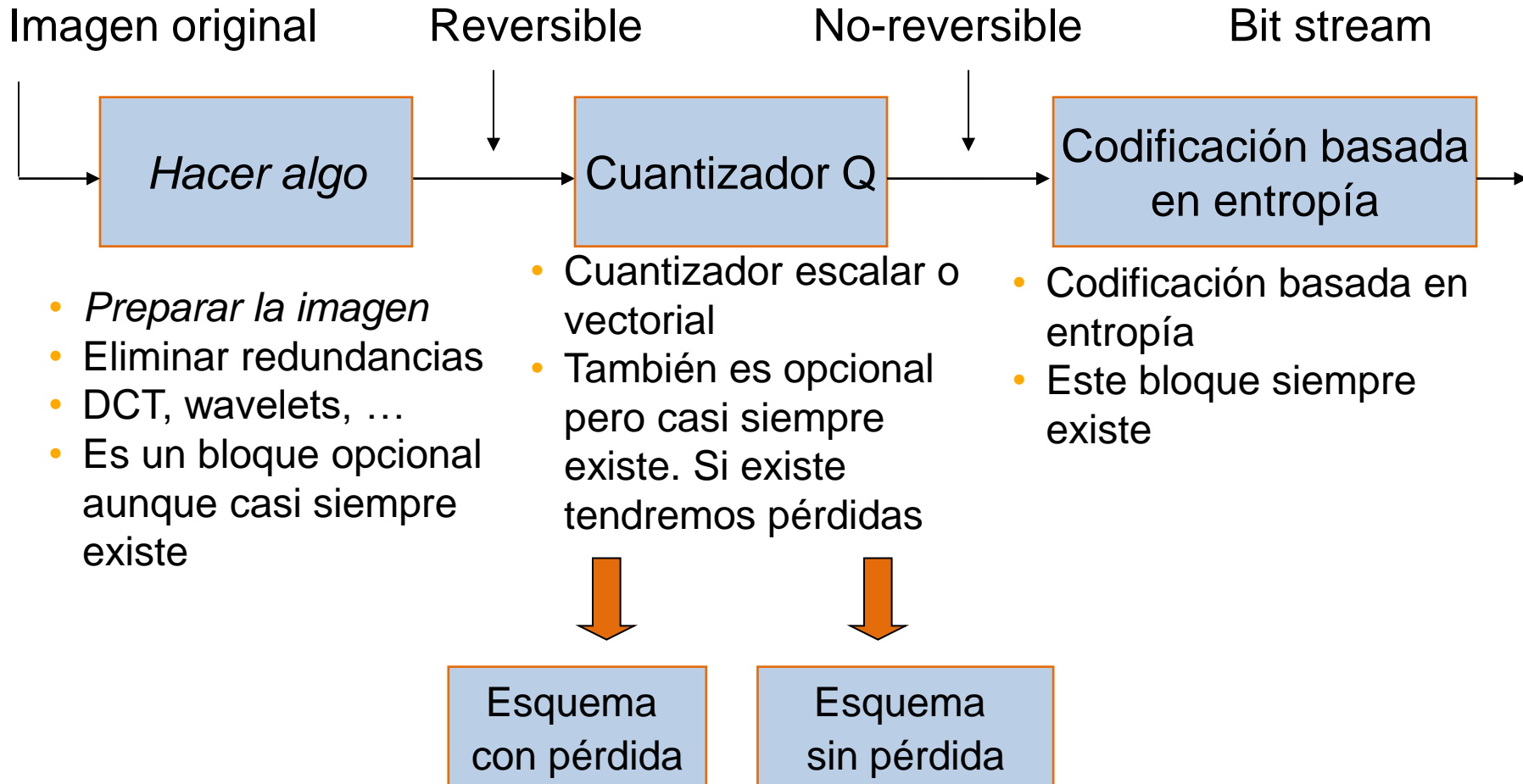
Un esquema de compresión consta tanto de la parte de compresión como de la de decodificación.

Normalmente se utiliza el término algoritmo de compresión para denotar el esquema completo de compresión.

Los algoritmos de compresión se dividen en dos grandes clases:

- ❖ Algoritmos sin pérdida, en los que la entrada al codificador,  $X$ , y la salida del decodificador,  $Y$ , coinciden.
- ❖ Algoritmos con pérdida, que suelen proporcionar mayor compresión que los sin pérdida, pero en los que  $X$  e  $Y$  no coinciden, aunque se parecen (concepto a definir).

# Esquema general para la codificación (compresión) de imagen/vídeo



## III.4 Compresión sin pérdida

Como indica su nombre, no hay pérdida de información. Se utiliza en aplicaciones donde no se permite ninguna diferencia entre los datos originales y los decodificados.

Son campos de aplicación, entre muchos otros:

- Compresión de texto

- Compresión de datos bancarios

- Compresión de datos empresariales/financieros

- Compresión de binarios/ejecutables

- Compresión de imágenes médicas

## Técnicas estadísticas

1. Código de Huffman.
2. Codigos aritméticos.
3. Código de Golomb.

## Técnicas basadas en diccionarios

1. LZW, LZ77.

## Técnicas predictivas

1. PPM, Método de Burrows-Wheeler.

**Estándares:** Morse, Braille, Unix compress, gzip, zip, bzip, gif (sin pérdida hasta 256 colores), png, bmp, jbig, jpeg sin pérdida,...

## III.5. Compresión con pérdida

Estas técnicas llevan asociadas una pérdida de información, los datos originales no pueden, normalmente, ser recuperados exactamente.

Voz e imágenes (vídeo) son ejemplos claros de campos que toleran pérdida en la compresión.

Incluye técnicas como:

1. Cuantización escalar y de vectores.
2. Wavelets.
3. Transformaciones por bloques.
4. Estándares: JPEG, JPEG 2000, MPEG (1, 2, 4).

**Factor de compresión** = cociente entre el número de bits necesarios para representar los datos antes de la compresión y el número de bits necesarios para representar los datos después de la compresión.

**Ejemplo:** dada una imagen de tamaño 256x256 con un byte de información por píxel, si tras la compresión ocupa 16.384 bytes su factor de compresión será 65.536: 16.384=4:1.

**También** podemos usar el número medio de bits necesarios para representar cada dato. En nuestro ejemplo, si inicialmente usábamos 8 bits: 2 bits/elemento.

## Para imágenes (en general, sustituimos imagen por datos)

$$\text{Factor de compresión} = \frac{\text{Bits de la imagen original}}{\text{Bits de la imagen comprimida}}$$

$$\text{Bits/pixel} = \frac{\text{Bits en la imagen comprimida}}{\text{Número de píxeles}}$$



# Ejemplo



Imagen original  
256 x 256 x 8 bits

$$\text{Bits/pixel} = \frac{40.000}{256 \times 256} = 0.61 \text{ bpp}$$

Si el número de bits por píxel en la imagen original es 16, ¿cuál sería el factor de compresión?

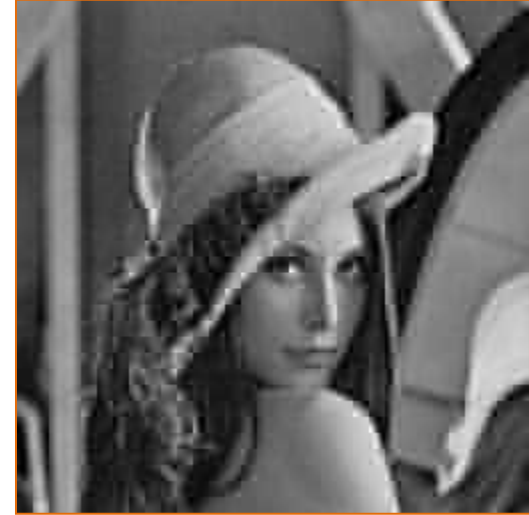


Imagen comprimida con  
40.000 bits

$$\text{F.C} = \frac{256 \times 256 \times 8}{40.000} = 13.1$$

$$\text{F.C} = \frac{8 \text{ bpp}}{0.61 \text{ bpp}} = 13.1$$

Cuando la compresión es con pérdida tenemos que utilizar, además de la cantidad de compresión obtenida, una medida para determinar la diferencia entre los datos originales y reconstruidos. Esta diferencia recibe el nombre de **distorsión**.

Las medidas de distorsión podrían ser a su vez basadas en criterios “matemáticos” o perceptuales. Lo discutiremos cuando veamos la compresión con pérdida.

Veamos sólo unas ideas

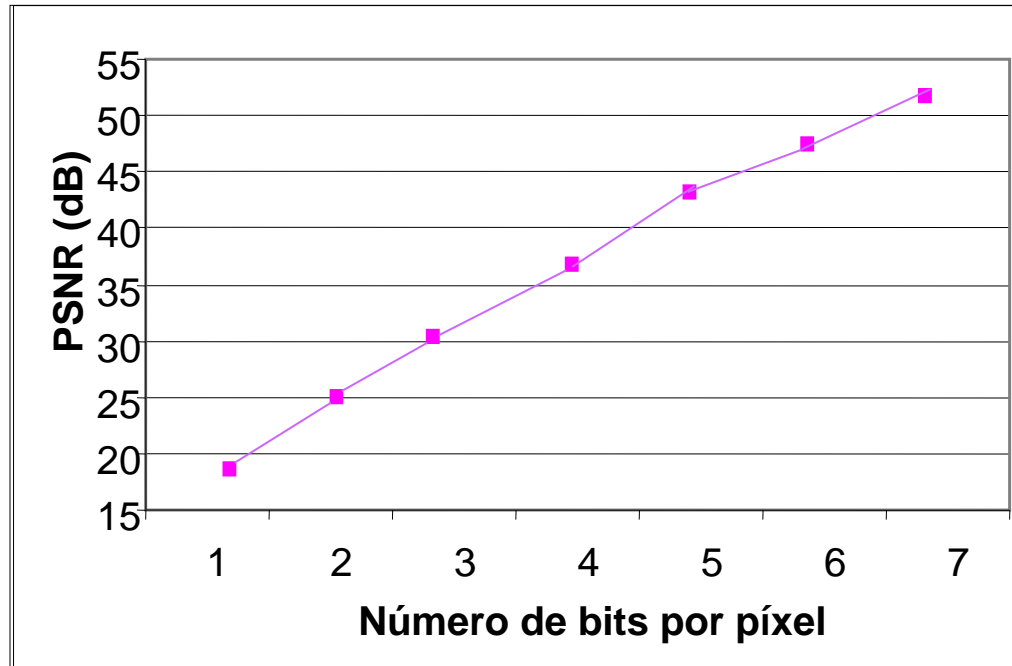
# Estos conceptos los veremos detenidamente con la compresión con pérdida



Medida subjetiva: muy buena  
PSNR: alto (30dB)



Medida subjetiva: Medio-bajo  
PSNR: alto (23dB)



**Curva razón (tasa)-calidad**  
Para una curva (tasa)-distorsión el eje y sería el MSE

$X$  y  $\hat{X}$  denotan la imagen original y la reconstruida (decodificada)

CRIM

Visualmente:

Excelente-bueno

Bueno-Moderado

-Pobre

Pobre-muy malo

$$MSE = \frac{1}{N} \sum_{i=0}^{N-1} |X_i - \hat{X}_i|^2$$

$$MAD = \frac{1}{N} \sum_{i=0}^{N-1} |X_i - \hat{X}_i|$$

$$PSNR = 10 \log_{10} \frac{255^2}{MSE};$$

# IV. Recuperación de Información Multimedia

## IV.I Definición

En su definición más básica la recuperación de información multimedia es el **proceso de búsqueda y localización de documentos (en un sentido muy amplio) multimedia.**

**Su campo de investigación asociado es la construcción de los mejores motores de búsqueda.**

Curiosamente la consulta puede ser multimedia

Tomar una foto y buscar información sobre el sitio

Piensa en otros ejemplos

Realicemos un breve recorrido por la recuperación de información

## IV.II. Problemas de Recuperación de Información

Todos experimentamos la recuperación de información como un proceso que lleva asociado motores de búsqueda, por ejemplo Google, Yahoo, Bing, etc

Durante miles de años, los métodos tradicionales de recuperación de información han sido proporcionados por los bibliotecarios: tarjetas de referencias con metadatos en catálogos. Objetos en localizaciones físicas que siguen un cierto esquema y una mesa de información

Uno de los ejemplos más antiguos de recuperación.

Mesopotamia, siglo XXIV AC



Como la potencia de los ordenadores y la disminución del coste del almacenamiento han facilitado la búsqueda, por ejemplo de texto, en grandes colecciones.

Simplificando (mucho), cada palabra de un documento es insertada en un índice que apunta de vuelta al documento. Al introducir la palabra, simplificando mucho, obtenemos una lista con todos los documentos que la contienen.

Si tenemos dos palabras, el sistema devuelve la intersección de las dos listas asociadas a las dos palabras.

Sin embargo, ¡no es tan simple !!.

ii Existen numerosos problemas a resolver!!

Estaría bien que pudiésemos tomar una foto de un monumento, la Alhambra, por ejemplo y que al buscarla nos diesen información sobre ella. Prueba con **Google images**. Mira también **Google Goggles** Las aplicaciones en turismo son obvias.

Busca la aplicación desarrollada por **Snaptell** para encontrar CDs, películas, libros, ... .

Podríamos también querer comparar rayos-X de pacientes con los que hubiera en una base de datos



Intentemos analizar con ejemplos la recuperación de información desde el punto de vista de la consulta (columna) y el tipo de documento que devuelve el sistema (fila)

Documento \ Consulta	emoción	tararear	sonido	imagen	texto	habla	bosquejo
texto					A		
video	C		E	C	C		
imagen							
habla					D		
música		B					
bosquejo							



Entrada A: tradicional motor de búsqueda de texto

Entrada B: tararear una canción y el sistema hace sonar la canción

Entrada C: búsqueda de vídeo basado en multimodalidad.  
Ejemplo: devuélveme videos de escenas tristes a partir de una imagen de un castillo derruido y texto vampiro.

Entrada D: a partir de un texto podemos querer que nos devuelvan los programas de radio de debate.

Piensa en ejemplos que rellenen el resto de las casillas

Obviamente, no todas las casillas son iguales de fáciles o de útiles.

En E podríamos tener una entrada en la que imitamos el rugido de un león y queremos documentales de vida salvaje.

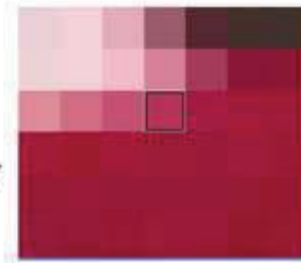
No todos los sistemas de recuperación de información tienen el mismo éxito:

- El indexado de texto lleva siendo aplicado con éxito durante más de dos décadas
- El indexado multimedia por contenidos está menos maduro.

## IV.III Desafíos en el indexado multimedia automático

- Un tipo de medio tiene que localizarse (acoplarse) en otro – cross-media retrieval -.
- Las imágenes y vídeos no contienen normalmente metadatos
- Rasgos de bajo nivel no se correlacionan fácilmente con el significado de alto nivel de, por ejemplo, imágenes –el gap semántico-.

# El Gap Semántico



?

podemos localizar las caras, a lo mejor la copa pero qué hay sobre la descripción del momento: victoria de los ingleses sobre los alemanes en 1996, alegría de los ingleses, tristeza de los alemanes, etc.

En algunos casos el gap semántico puede saltarse, como en las descripciones de hierba, cielo, gente, etc.



**Polisemia:** multitud de significados

Buscamos imágenes de la misma persona, del mismo tipo de arte, transiciones graduales de luz.

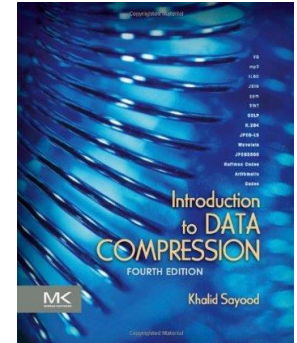
En estos casos es fundamental la retroalimentación del usuario.

**Problema de tiempo de respuesta.** Es obvio que no podemos hacer comparación rasgo a rasgo. Las estructuras de árbol de dimensión muy elevada no son muy eficientes en la búsqueda, se necesita compresión, vecinos más próximos, etc.

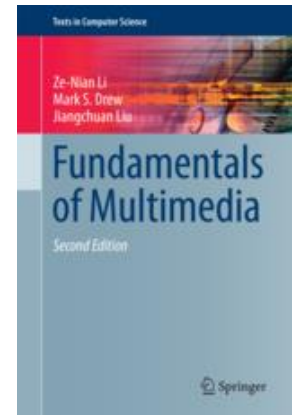
**Resistencia a la utilización.** Parece que estamos muy dispuestos a usar sistemas como los de Google, Yahoo o Microsoft mientras que otros no son usados. Esto podría ser debido a como de amigable percibimos el sistema.

## V. Bibliografía

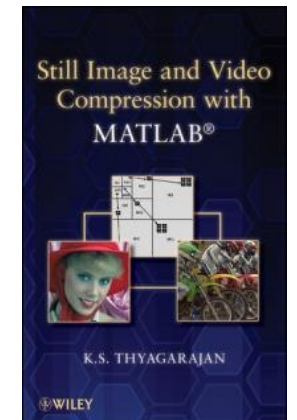
K. Sayood, Introduction to Data Compression, cuarta edición, 2012. <http://www.mkp.com>



Li, Ze-Nian, Drew, Mark S., Liu, Jiangchuan  
Fundamentals of Multimedia, 2014



K.S. Thyagarajan, Still Image and Video  
Compression with Matlab, 2012



# VI. Bibliografía

- Curso Multimedia del Prof. A.D. Marshall  
<http://www.cs.cf.ac.uk/Dave/Multimedia/>
- Image and Video Compression Learning Tool  
VcDemo  
<http://insy.ewi.tudelft.nl/content/image-and-video-compression-learning-tool-vcdemo>
- Apuntes de Prof. Luis Torres, Codificación de Contenidos Audiovisuales, Escuela Técnica Superior de Ingeniería de Telecomunicación de Barcelona, Universidad Politécnica de Catalunya



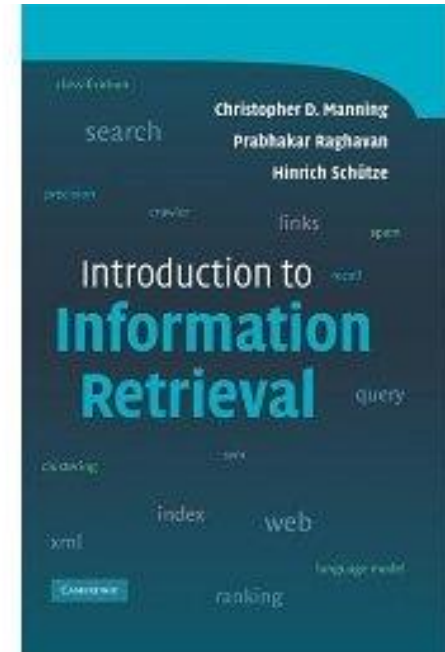


## VI. Bibliografía

Material básico del curso para recuperación de información multimedia:

### **Introduction to Information Retrieval,**

Hinrich Schütze, Christopher D. Manning, Prabhakar Raghavan, CUP, 2008



### **Recuperación de información. Un enfoque práctico y multidisciplinar**

Fidel Cacheda, Juan Manuel Fernández Luna, Juan Francisco Huete Guadix, Ra-Ma 2011.

