ORGANIZACIÓN DE DATOS 2021

Conceptos de compresión de datos

UNIVERSIDAD NACIONAL de MAR DEL PLATA

Compresión

¿Por qué es importante conocer sobre compresión?

- Casi toda la información se maneja en forma digital (Redes Sociales, digitalización de documentos, audio, video).
- Cada vez se necesita más capacidad de almacenamiento y transmisión de información (cloud computing)
- Año a año se requiere un mejor uso de los recursos (tv-digital, telefonía IP, teleconferencias, redes en general). El contenido en alta definición o 4k necesita de altísimos anchos de banda y tasas de datos sin comprimir.

Compresión

¿Qué área de la ciencia aplica?

- La compresión como álgebra, análisis matemático, probabilidad y estadística y algoritmia aplicados.
- Aplica a Teoría de la Información.

Teoría de la información

Según la teoría de la información de Shannon, definimos:

- **Fuente**: es todo aquello que emite el mensaje
- **Mensaje**: Es un conjunto de alfabetos que es transmitida por la fuente. En nuestro caso el alfabeto es 0 y 1.
- **Código**: Es el conjunto de alfabetos que representan el mensaje de acuerdo a reglas o convenciones.
- **Información**: Es el conjunto mínimo de alfabetos para que se requieren para representar un mensaje.

Compresión

Tipo de compresión

- Con perdida (lossy)
  - Imágenes
  - Audio
  - Video

- Sin perdida (lossless)

Compresión

Existen distintos tipo de fuentes de datos:

- Estructurada
- Desestructurada

Entropía de una fuente

Ya hablamos del concepto de información, veremos ahora como se mide la misma . De acuerdo a la teoría de la información, el nivel de información de una fuente se puede medir según la entropía de la misma. Los estudios sobre la entropía son de suma importancia en la teoría de la información y se deben principalmente a Shannon, existen a su vez un gran numero de propiedades respecto de la entropía de variables aleatorias debidas a Kolmogorov.

Definimos a la probabilidad de ocurrencia(Pi) de un mensaje en una fuente como la cantidad de apariciones de dicho mensaje dividido el total de mensajes. Li es la longitud del código utilizado para representar a dicho mensaje

H = Sumatoria (i=0 -> n) Pi * Li

"H" se lo denomina "Entropía de la fuente". La entropía de la fuente determina el nivel de compresión que podemos obtener como máximo para un conjunto de datos.

Código

Un código es decodificable sí y solo sí un código solo puede corresponder a un único mensaje.

Ejemplo:

a=0
b=01
C=10

Si el decodificador recibe el código: "0010" no puede distinguir si el mensaje original fue "aba" o "aac", ya que puede interpretarlo como 0 01 0 o como 0 0 10

En Cambio si los codigos se escribe asi:

a=0
b=10
c=11

El código "01011" solo puede corresponder a "abc"

Compresión de Datos

La idea fundamental es muy simple, guardar la mayor cantidad de datos posibles en la menor cantidad de espacio posible y que dado el archivo comprimido pueda recuperarse el archivo original.

Teorema fundamental de la compresión de datos.
"No existe un algoritmo que sea capaz de comprimir cualquier conjunto de datos"

Técnicamente:
"Si un compresor comprime un conjunto de datos en un factor F entonces necesariamente expande a algún otro conjunto de datos en un factor MAYOR que F"

Compresión RLE

El método de compresión RLE (Run Length Encoding, a veces escrito RLC por Run Length Coding) es utilizado por muchos formatos de imagen (BMP, PCX, TIFF). Se basa en la repetición de elementos consecutivos.
El principio fundamental consiste en codificar un primer elemento al dar el número de repeticiones de un valor y después el valor que va a repetirse

En comunicaciones : ITU T-REC-T.45, MNP5
-En su forma más básica se basa en “contar” la repetición de determinado símbolos

Ejemplo :

AAAAAAAAgh

Se comprime en

8Agh

Compresión RLE

En realidad, la compresión RLE está regida por reglas particulares que permiten que se ejecute la compresión cuando sea necesario y que se deje la cadena como está cuando la compresión genere pérdida. Las reglas son las siguientes:

Si se repiten tres o más elementos consecutivamente, se utiliza el método de compresión RLE.

De lo contrario, se inserta un carácter de control seguido del número de elementos de la cadena no comprimida.

Su principal ventaja radica en que RLE se puede usar aun cuando no se pueda analizar el texto o imagen completa a enviar

Compresión

Compresión estadística

Los compresores estadísticos conforman la columna vertebral de la compresión de datos. La idea básica es: El compresor "adivina" cual va a ser el input y escribe menos bits al output si adivinó correctamente. Notemos que el descompresor debe "adivinar" de la misma forma que el compresor para poder decodificar el archivo.

Los compresores estadísticos buscan representar a cada carácter con un número de bits proporcional a -log2(Pi) (longitud ideal de acuerdo a la entropía). Siendo Pi probabilidad de aparición de caracteres

Compresión Estádistica

Compresión estadística o basada en la entropía

El objetivo de la compresión estadística es en definitiva determinar cual deberá ser la longitud de los códigos correspondientes a los caracteres de un archivo, el valor de los códigos puede ser cualquiera siempre y cuando se respeten las longitudes, veremos a continuación una técnica que se utiliza frecuentemente para obtener las longitudes de estos códigos.

La compresión estadística se vale para poder comprimir una determinada fuente en el carácter estructurado de la misma, aquellas fuentes que son totalmente aleatorias, es decir de información pura, no pueden comprimirse.
El 90% de la información que se almacena en una computadora es de carácter netamente estructurado por lo que podremos comprimirla con facilidad. Cuanto mejor sea para archivos no aleatorios y peor para archivos random mejor será un algoritmo de compresión estadístico.

Compresión

Compresión estadística

Ejemplo:

Por ejemplo si solo tenemos cuatro caracteres a,b,c y d y el compresor sabe que la probabilidad de "a" es 0.5, "b" es 0.25, "c" es 0.25 y "d" es 0. Entonces intentaremos codificar de la forma:

- 1 bit para "a" (-log2(.5)=1)
- 2 bits para b y c (-log2(.25)=2)

Por ejemplo:

- a : 0
- b : 10
- C : 11

Como "d" tiene probabilidad de ocurrencia cero ni siquiera lo codificamos.

Si todos los caracteres ASCII son equiparables para un archivo entonces usaremos log2(1/256) = 8 bits para codificar cada uno, es decir que no podemos comprimir el archivo.

Compresión

Compresión estadística

Una forma de evitar el redondeo es ablocando símbolos, por ejemplo:

AA : 0
AB : 10
BA : 110
BB : 111

Con lo cual A es representado en alrededor de 0.7 bits y B en 1.4 bits (valores no enteros). El ablocar códigos para lograr valores enteros para los bloques que correspondan a valores no-enteros de los caracteres es, sin embargo, una tarea que puede llegar a resultar mucho mas costosa que el proceso de compresión en sí, por lo que su utilidad es únicamente teórica.

Entropía de una fuente

Ya hablamos del concepto de información, veremos ahora como se mide la misma . De acuerdo a la teoría de la información, el nivel de información de una fuente se puede medir según la entropía de la misma. Los estudios sobre la entropía son de suma importancia en la teoría de la información y se deben principalmente a Shannon, existen a su vez un gran numero de propiedades respecto de la entropía de variables aleatorias debidas a Kolmogorov.

Definimos a la probabilidad de ocurrencia(Pi) de un mensaje en una fuente como la cantidad de apariciones de dicho mensaje dividido el total de mensajes. Li es la longitud del código utilizado para representar a dicho mensaje

H = Sumatoria (i=0 -> n) Pi * Li

"H" se lo denomina "Entropía de la fuente". La entropía de la fuente determina el nivel de compresión que podemos obtener como máximo para un conjunto de datos.

Compresión estadística

Huffman estático

Este algoritmo desarrollado por Huffman en 1952 es el mas popular y utilizado de los compresores estadísticos existiendo numerosas variantes, su popularidad se basa en que se demuestra que un árbol de Huffman es óptimo para la compresión de una cierta cantidad de caracteres ('cualquier' otro árbol que se use para comprimir genera un archivo igual o mayor al generado por Huffman).

Método :

Se le asigna a cada carácter que aparece en el archivo su frecuencia de aparición, caracteres con frecuencia cero se descartan pues no van a aparecer.

Se tienen pues n conjuntos de un caracter cada uno, con su respectiva frecuencia, a continuación mientras la cantidad de conjuntos sea mayor que uno se toman los dos conjuntos cuya sumatoria de frecuencias sea menor y se unen. Cuando queda un único conjunto, se ha formado un árbol binario (con el código Huffman).

Compresión

Huffman estático

Algoritmo

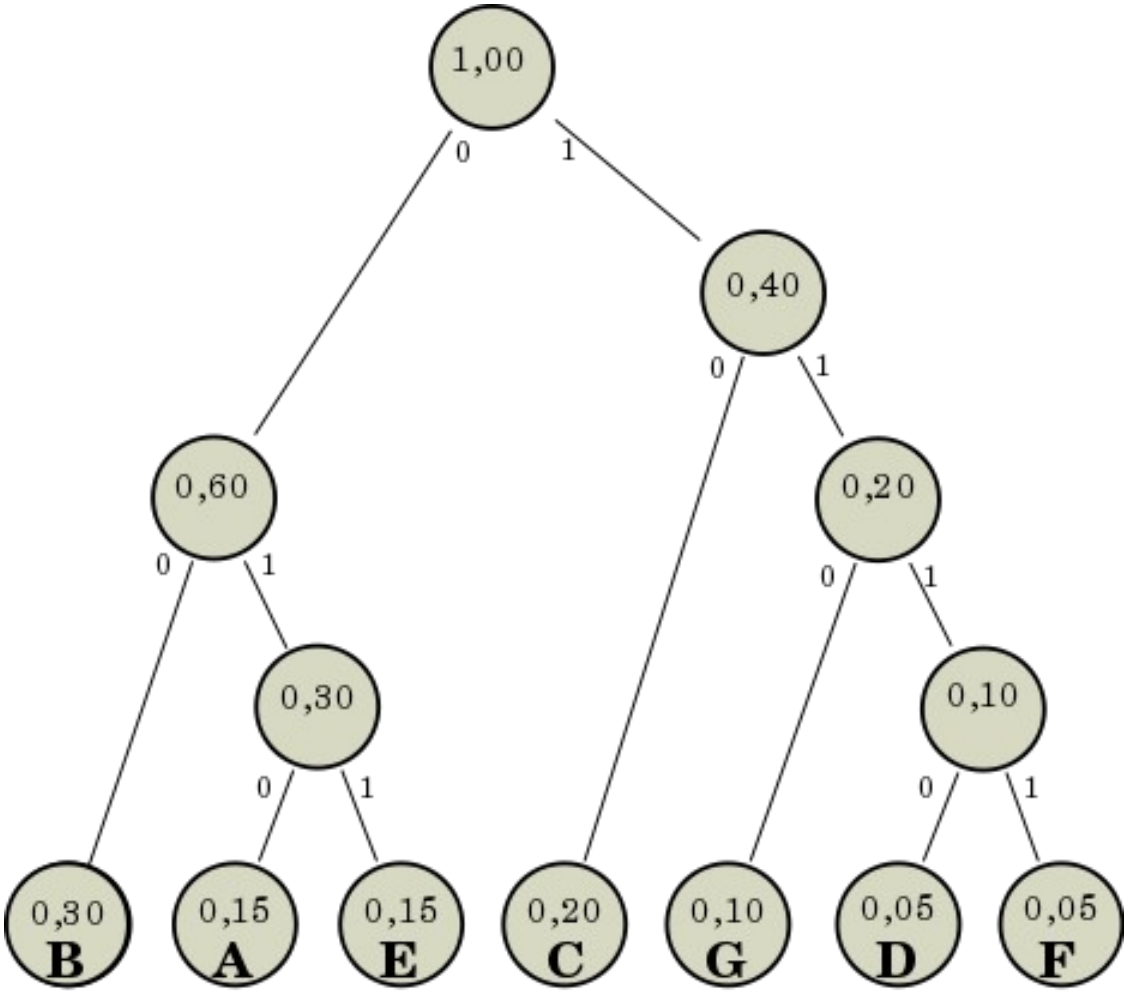
1.- Armar Histograma

2.- Agrupar de a dos caracteres siempre que su frecuencia de aparición sea menor a todos y se arma un nuevo símbolo, con la frecuencia de aparición de los dos anteriores.

3.- Repetir el punto dos hasta que quede un árbol binario donde el nodo principal tenga cadena con todos los caracteres.

El descompresor deberá primero recuperar el árbol y luego va descomprimiendo procesando bit por bit, comienza en la raíz del árbol y va doblando según encuentre ceros o unos, cuando llega a una hoja emite el carácter encontrado y vuelve a comenzar desde la raíz. El fin del archivo se indica al descompresor pre-concatenando al archivo comprimido la longitud del archivo original o bien inventando un carácter nuevo (código 256) que indique el fin de archivo.

Código de Huffman - Ejemplo



Compresión – Clasificación según su input

Básicamente cualquier algoritmo de compresión puede clasificarse en tres categorías según como se procesa el archivo de entrada.

- **Algoritmos estáticos**: Leen una vez el archivo para recolectar estadísticas y luego otra vez para comprimirlo.
- **Algoritmos semi-estáticos**: Levantan en memoria un bloque del archivo, lo procesan en forma estática y luego graban el bloque comprimido. Solo hacen una lectura del archivo original pero cada bloque es procesado dos veces (en memoria).
- **Algoritmos dinámicos**: Leen una vez el archivo de entrada y lo van comprimiendo a medida que lo leen.

Compresión – La necesidad de comprimir con pérdida - (loosy)

Tecnología de Red	Ancho de banda
Dial Up	56 kbps
ADSL lite	1.5 Mbps
T1	1.54 Mbps
ADSL 1	4 Mbps
Ethernet classic	10 Mbps
Wifi 802.11b	11 Mbps
ADSL 2+	24 Mbps
T3	44 Mbps
Wifi 802.11g	54 Mbps
Fast ethernet	100 Mbps
802.11n	150-300 Mbps
Gigabit ethernet	1 Gbps

Imagenes

PNG ~ 2 – 4 kB

GIF ~ 6 – 800 kB
JPG ~ 9 – 10 MB
TIFF ~ 900 – 10 MB
BMP ~ 900 – 50 MB

Documentos

DOCX ~4 – 8 kB
PDF ~ 18 – 20 kB
ODT ~ 80 – 90 kB

Archivos multimedia

eBook ~ 1 – 5 MB
MP3 song ~ 3 – 15 MB
DVD Movie ~ 4 GB
HD Movie ~ 5 – 8 GB
Blu-Ray Movie ~ 20 – 25 GB

Sin comprimir

24-bit, 1080i @ 60 fps: 24 × 1920×540[b] × 60 = 1.49 Gbit/s

24-bit, 1080p @ 60 fps: 24 × 1920×1080 × 60 = 2.98 Gbit/s.

Compresión - JPEG

Las imágenes en formato crudo (RAW), almacenan los píxeles en formato RGB, por lo que su tamaño es proporcional a la resolución

Compresión de imágenes

El estándar JPEG (Grupo Unido de Expertos en Fotografía, del inglés Joint Photographic Experts Group) para comprimir imágenes fijas de tono continuo (es decir, fotografías) se desarrolló gracias a los expertos fotográficos que trabajan bajo los auspicios unidos de la ITU, ISO e IEC, otro organismo de estándares

Compresión – Archivos RAW

Los archivos RAW contienen la información necesaria para producir una imagen visible a partir de los datos del sensor de la cámara.

Utilidades

- exifprobe
- nconvert -fullinfo

Compresión - JPEG

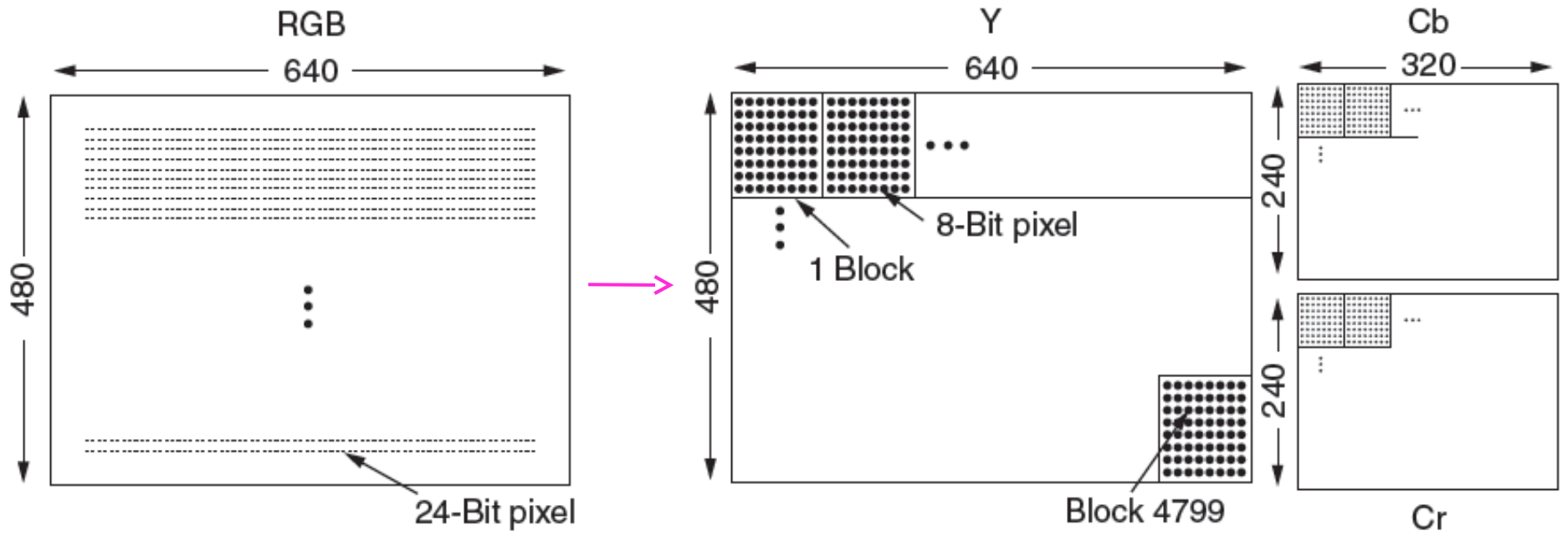
El paso 1 es la preparación del bloque. Con fines de especificación, supongamos que la entrada JPEG es una imagen RGB de 640 3 480 con 24 bits/píxel. RGB no es el mejor modelo de color para utilizar en la compresión.
El ojo es mucho más sensible a la luminosidad o brillo de las señales de video que a la crominancia o color de las mismas. Por ende, calculamos primero la luminosidad Y y las dos crominancias, Cb y Cr, a partir de los componentes R, G y B. Las siguientes fórmulas se utilizan para valores de 8 bits que varían de 0 a 255:

Y = 165 + 0.26R + 0.50G + 0.09B
Cb = 128 + 0.15R + 0.29G +2 0.44B
Cr = 128 + 0.44R + 0.37G + 0.07B

Compresión - JPEG

Los pixeles se mapean al espacio de luminancia/crominancia (YCbCr) y se sub-samplean

- El ojo es menos sensitivo a la crominancia



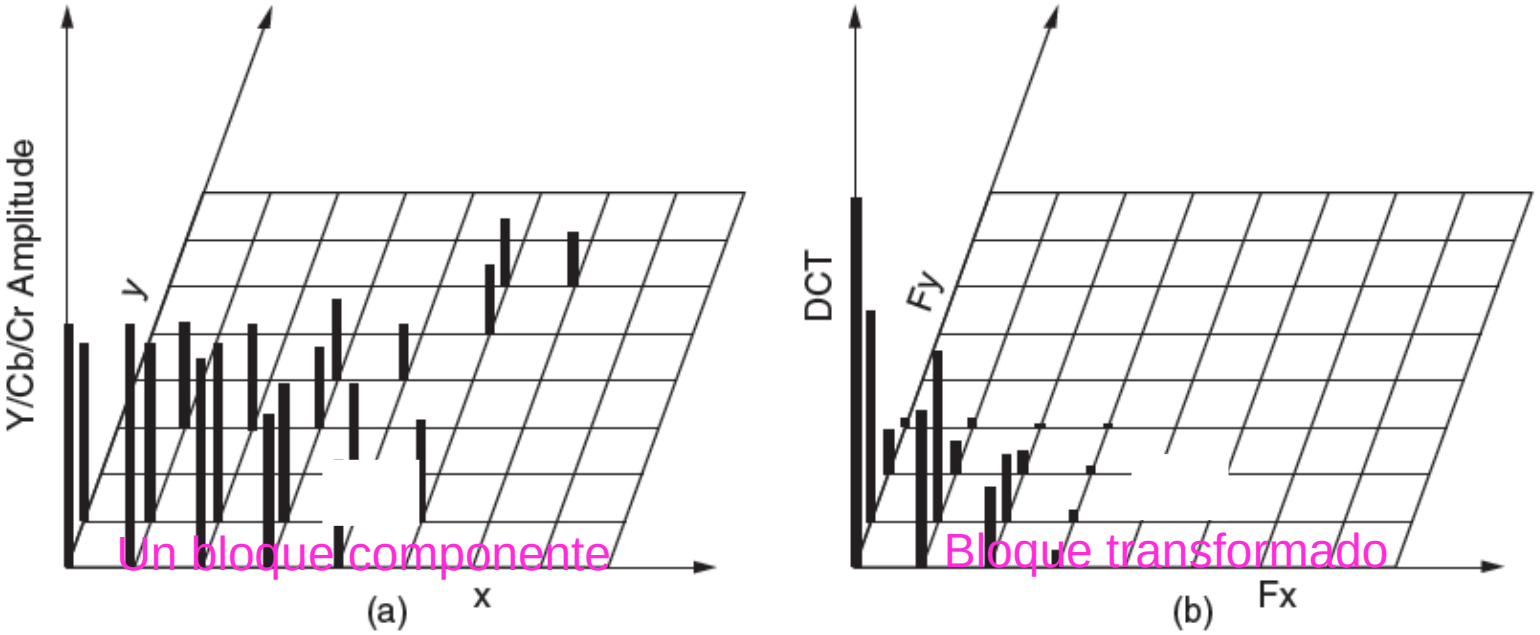
Entrada 24-bit RGB pixels

8-bit luminance pixels

8-bit chrominances for every 4 pixels

Compresión - JPEG

Paso 2: Cada bloque componente se transforma a frecuencias en el domino espacial con DCT (Discrete Cosine Transformation), esto comprime por perder precisión(datos), pero captura los componentes clave



Compresión - JPEG

Paso 3: Los coeficientes DCT se cuantizan diviendo por los valores en la tabla de cuantización; reduces bits in higher spatial frequencies

- El elemento superior izquierdo se diferencia sobre los bloques (paso 4)

DCT coefficients

150	80	40	14	4	2	1	0
92	75	36	10	6	1	0	0
52	38	26	8	7	4	0	0
12	8	6	4	2	1	0	0
4	3	2	0	0	0	0	0
2	2	1	1	0	0	0	0
1	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Input

/

Quantization table

1	1	2	4	8	16	32	64
1	1	2	4	8	16	32	64
2	2	2	4	8	16	32	64
4	4	4	4	8	16	32	64
8	8	8	8	8	16	32	64
16	16	16	16	16	16	32	64
32	32	32	32	32	32	32	64
64	64	64	64	64	64	64	64

Thresholds

=

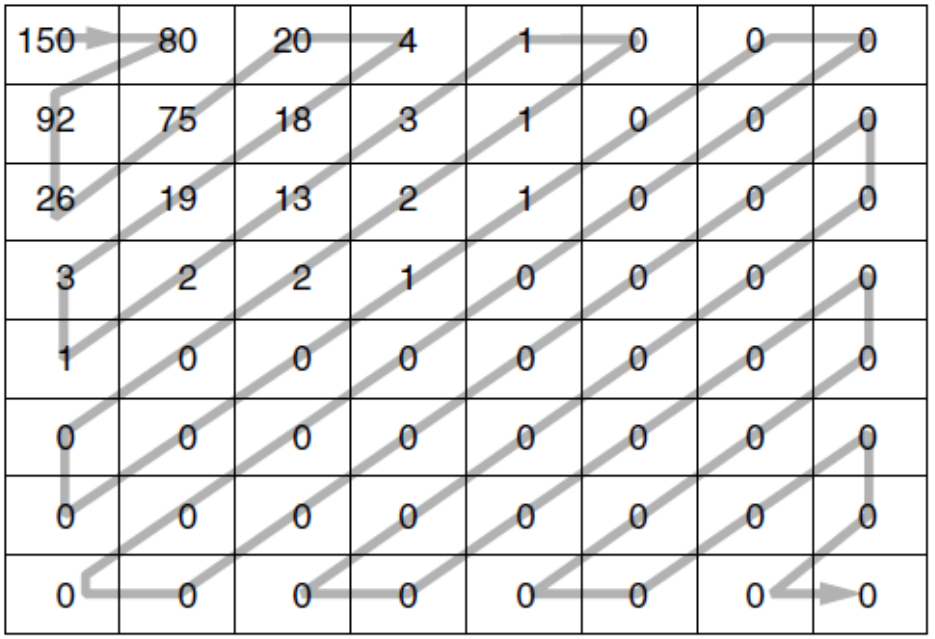
Quantized coefficients

150	80	20	4	1	0	0	0
92	75	18	3	1	0	0	0
26	19	13	2	1	0	0	0
3	2	2	1	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
0	0	0	0	0	0	0	0

Output

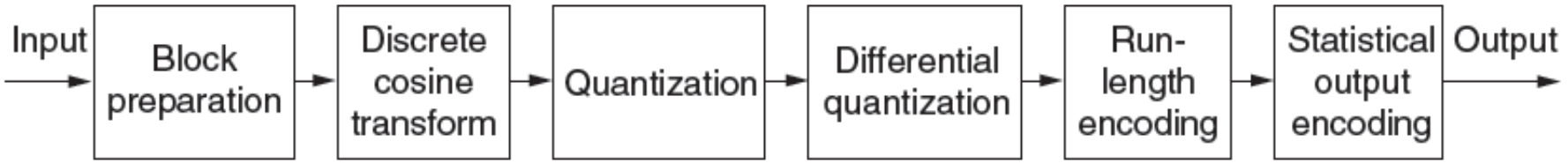
Compresión - JPEG

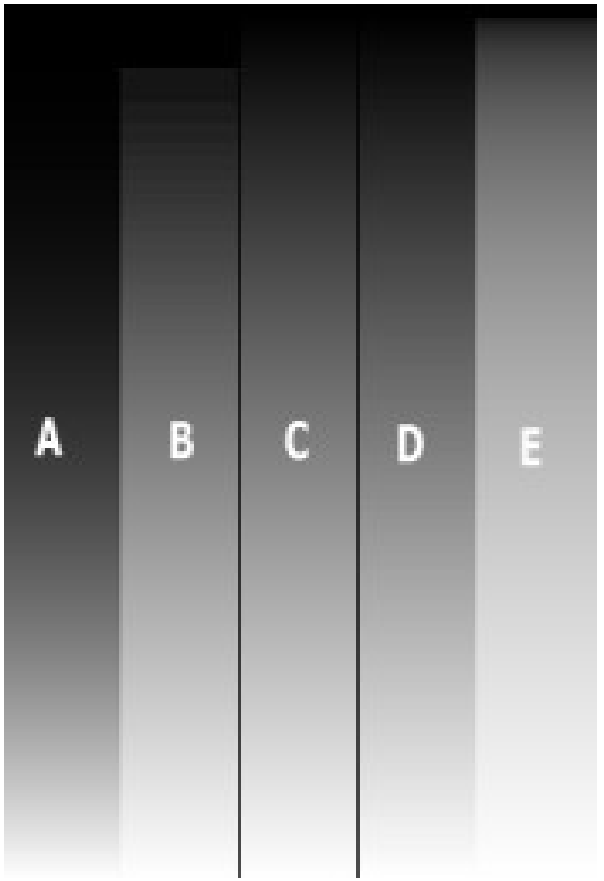
Paso 5: Se le aplica al bloque run-length codificada en zig-zag. Luego se codifica con Huffman (Paso 6)



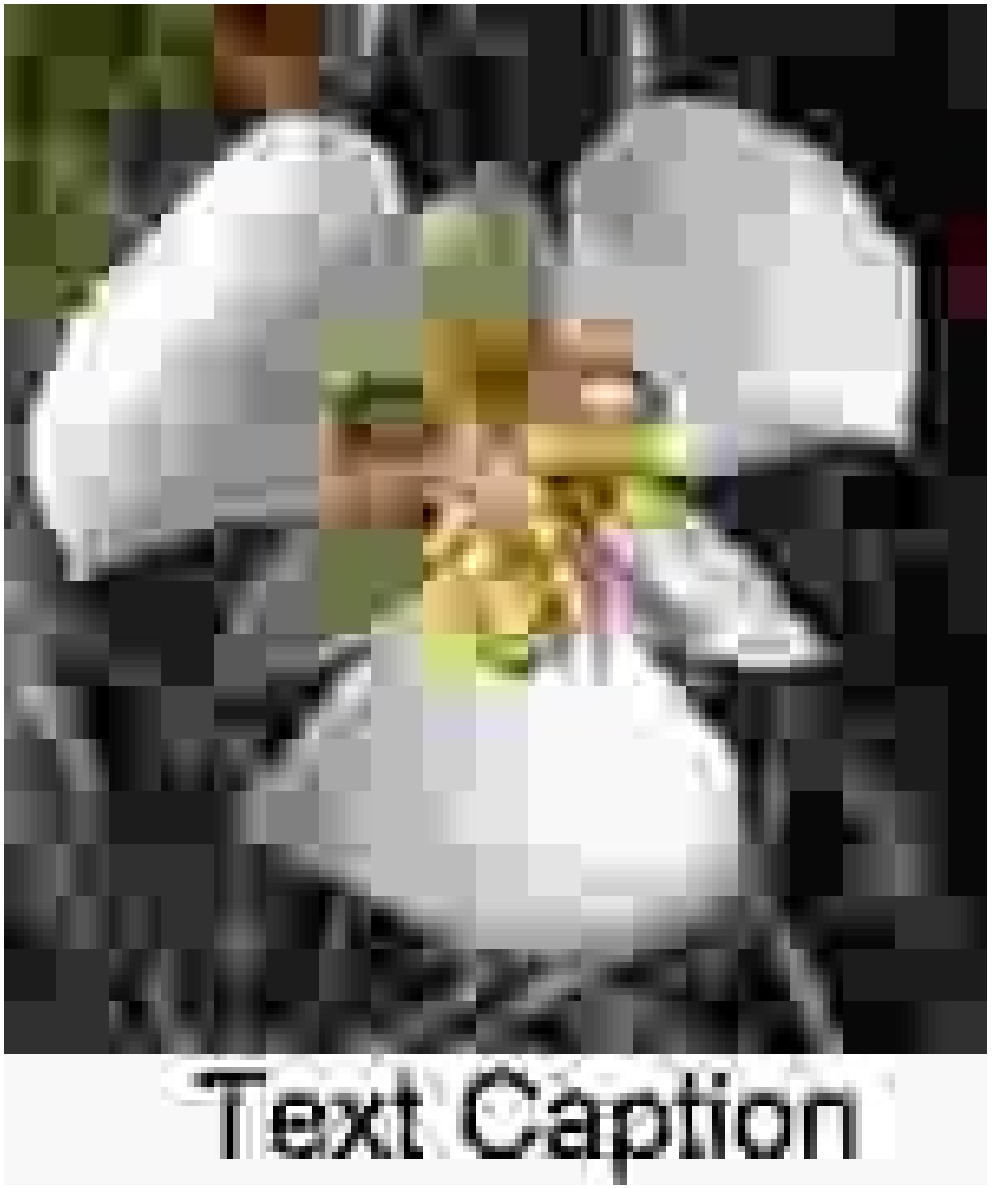
Orden en que los coeficientes son enviados

Compresión - JPEG





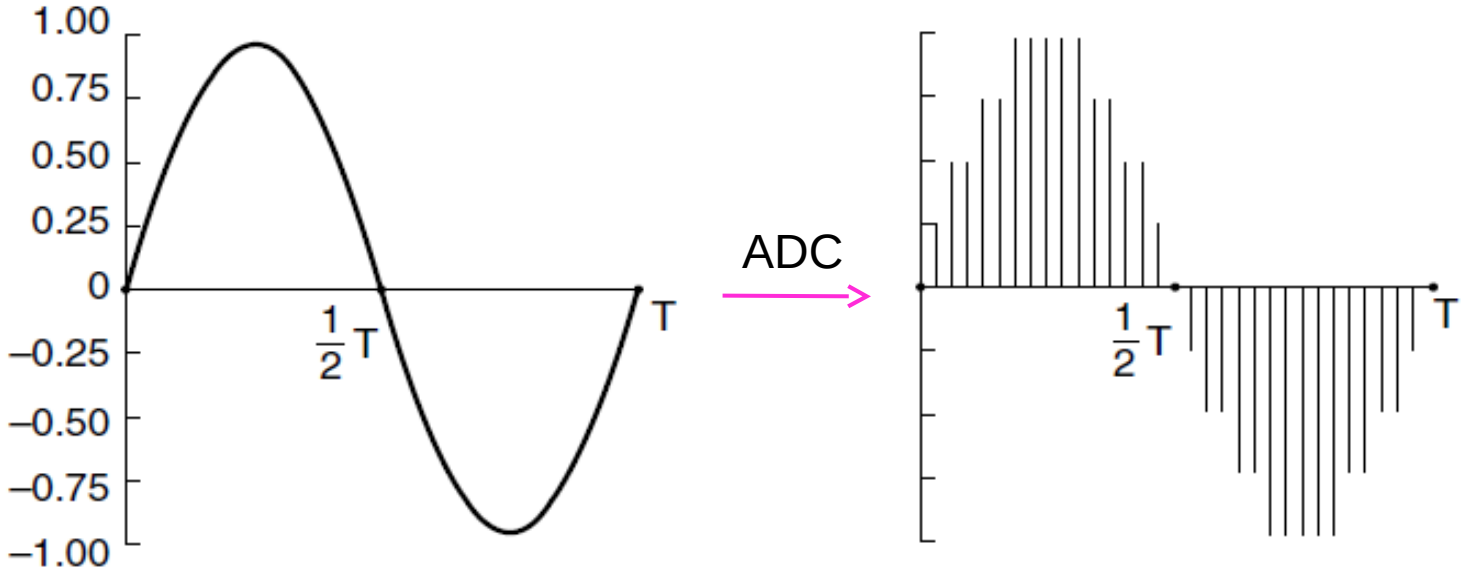
Compresión - JPEG



Compresión – Audio digital

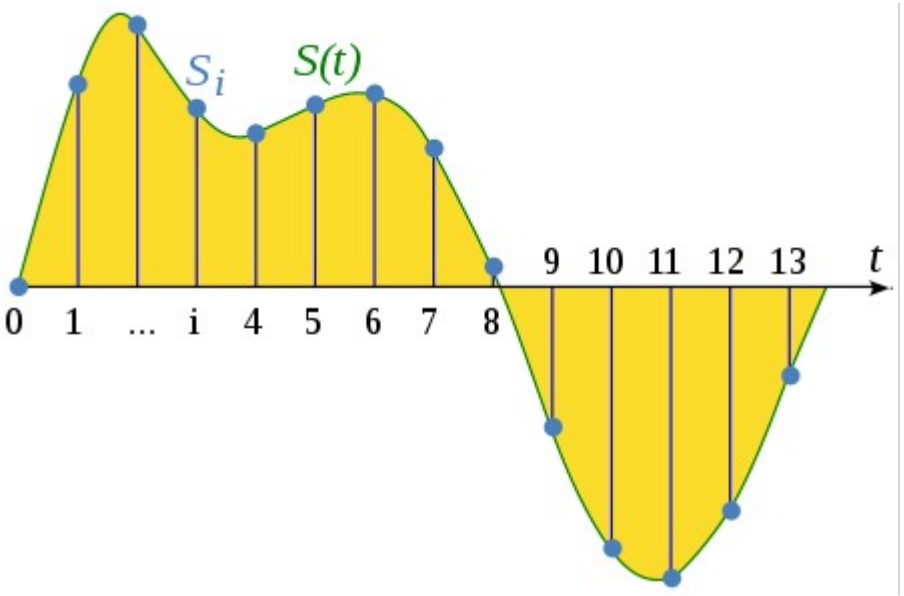
Los ADC (Analog-to-Digital Converter) producen audio digital desde un microfono – Teorema del muestreo : 2 x ancho de banda

- Telefono: 8000 8-bit samples/segundo (64 Kbps); el audio en la computadora es usualmente mejor (e.g., 16 bit)



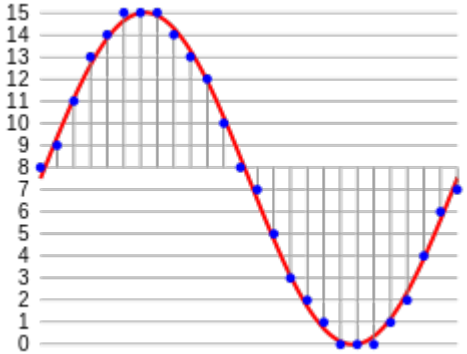
Audio analógico

Audio digital
(sampled, 4-bit quantized)


S_i
$S(t)$
0 1 ... i 4 5 6 7 8 9 10 11 12 13 t

Compresión – Audio digital

El rango de frecuencia del oído humano es de 20 Hz a 20 000 Hz. Algunos animales, en particular los perros, pueden escuchar frecuencias más altas. El oído escucha el ruido en forma logarítmica, por lo que la proporción de dos sonidos con potencia A y B se expresa de manera convencional en dB (decibeles) como la cantidad 10 log 10 (A/B)

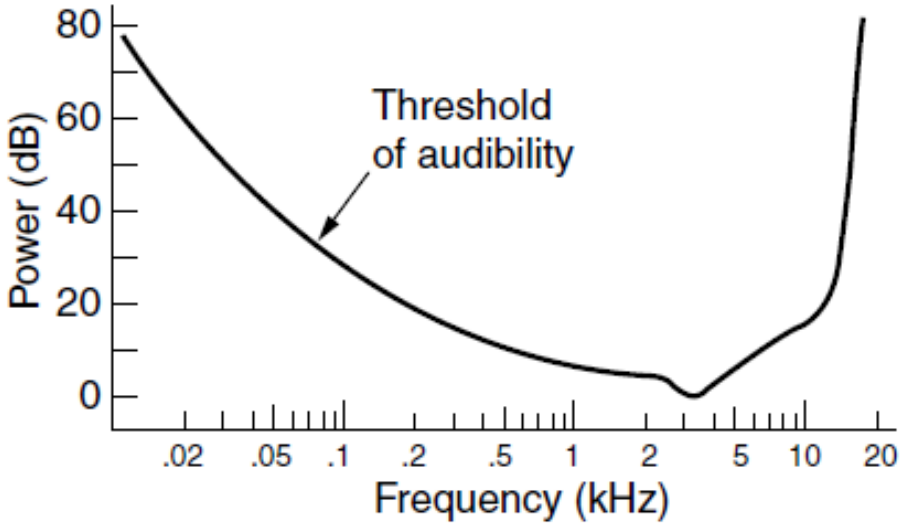


Ejemplo audio en calidad CD:
44000 muestras/seg * 16 bit muestra => 704000 bits/seg * 2 canales = >1.4 Mbits / seg

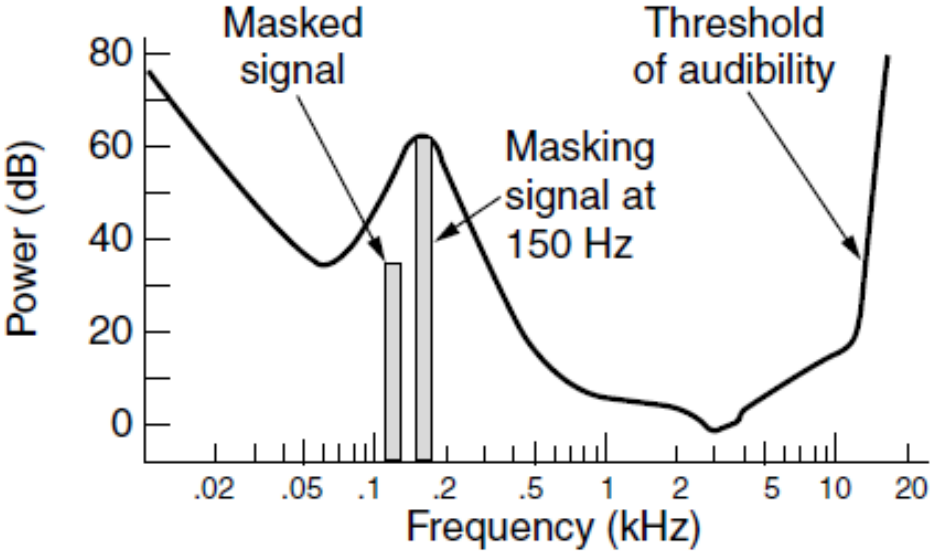
Compresión – Audio digital

El audio digital es usualmente comprimido con modelos psicoacústicos

- Los encoders (como AAC o MP3) explotan la percepción humana

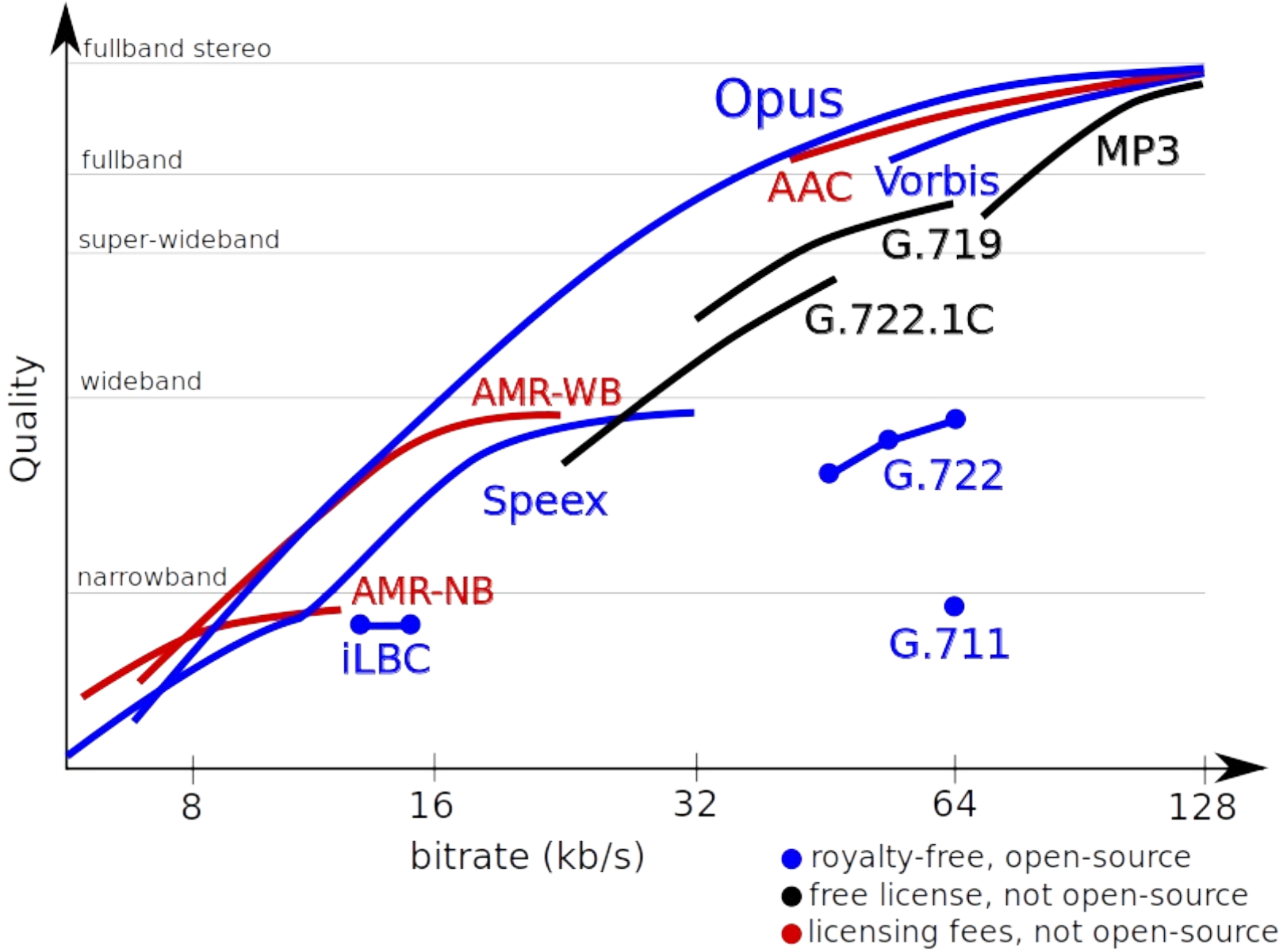


La sensibiidad del oido varía con la frecuencia



Un tono fuerte puede enmascarar un todo suave

Compresión – Audio digital



Compresión – video digital

Requerimientos de ancho de banda del video sin comprimir (RAW), segun su resolución, norma y profundidad de colores

525 NTSC uncompressed (YCbCr 4:2:2 chroma subsampling)

8-bit @ 720x486 @ 29.97 fps = 20 MB/s, or 70 GB/h.

10-bit @ 720x486 @ 29.97 fps = 27 MB/s, or 94 GB/h.

625 PAL uncompressed

8-bit @ 720x576 @ 25 fps = 20 MB/s, or 70 GB/h.

10-bit @ 720x576 @ 25 fps = 26 MB/s, or 93 GB/h.

720p HDTV uncompressed

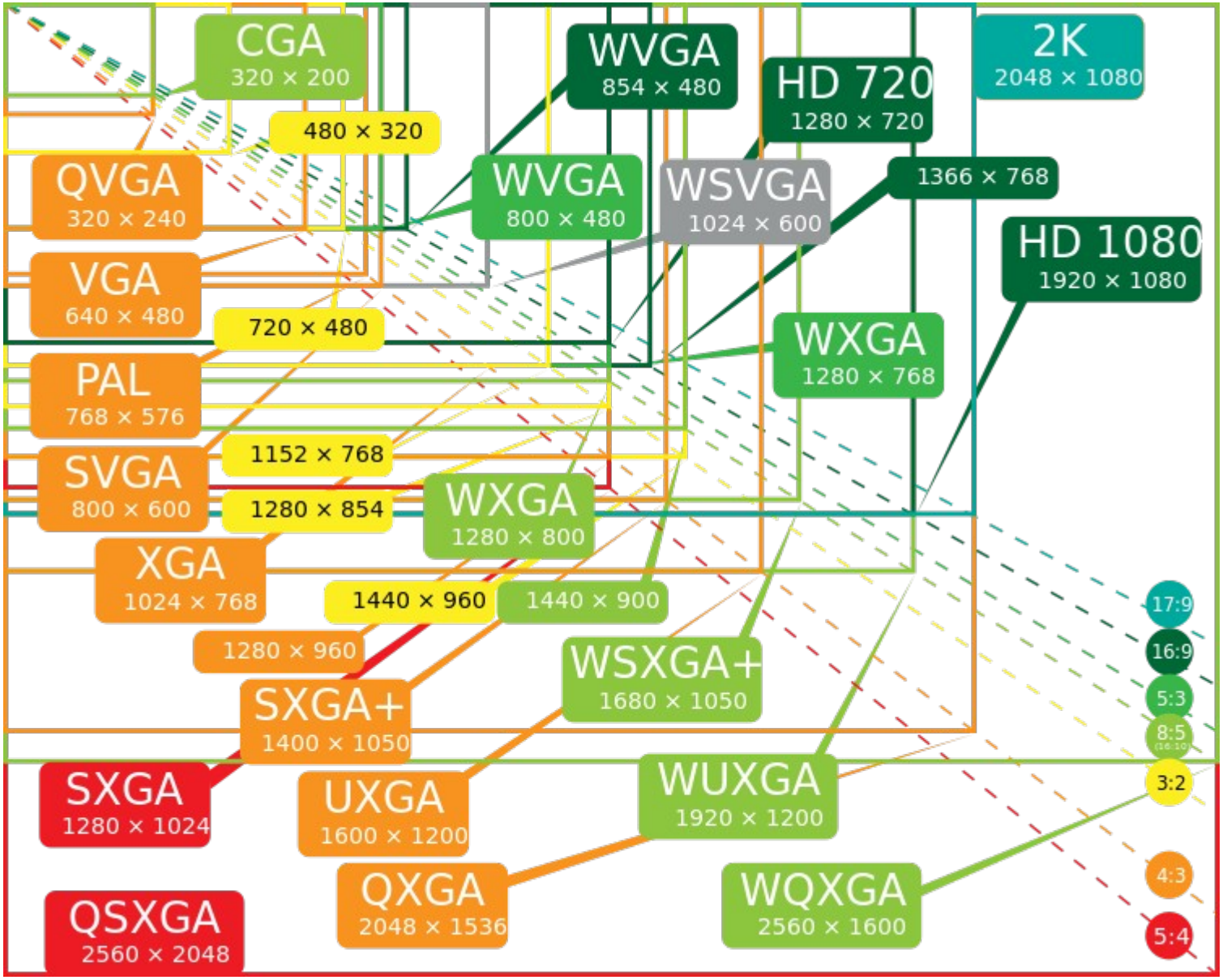
8-bit @ 1280x720 @ 59.94 fps = 105 MB/s, or 370 GB/h.

10-bit @ 1280x720 @ 59.94 fps = 140 MB/s, or 494 GB/h

8-bit @ 1920x1080 @ 29.97 fps = 119 MB/s, or 417 GB/h.

10-bit @ 1920x1080 @ 29.97 fps = 158 MB/s, or 556 GB/h.

Compresión – video digital



Compresión – Video digital

El objetivo es aprovechar la redundancia espacial y temporal

La redundancia espacial se resuelve aplicando JPEG a cada cuadro (DCT)

Compresión – video digital

MPEG comprime sobre una secuencia de cuadros, usando una técnica llamada motion tracking para remover redundancia temporal

- I (Intra-coded) cuadros son autocontenidos
- P (Predictive) cuadros usan predicción de block motion
- B (Bidirectional) cuadros pueden ser la prediccion base para un cuadro futuro

Tres cuadros consecutivos con componentes estacionarios y en movimiento