



# Introducción a la compresión de información multimedia

<b>Apellidos, nombre</b>	Oliver Gil, José Salvador (joliver@disca.upv.es)
<b>Departamento</b>	Informática de Sistemas y Computadores
<b>Centro</b>	Escola Tècnica Superior d'Enginyeria Informàtica

## 1 Resumen de las ideas clave

Dos ideas claves subyacen en este artículo. Primero, la conveniencia de usar técnicas de compresión para transmitir o almacenar información multimedia, y después describir los sistemas de compresión y clasificarlos en base a varias taxonomías.

Dentro de estas clasificaciones, describiremos los codificadores basados en la fuente y aquellos basados en la entropía, concepto que definiremos y del que veremos un ejemplo.

## 2 Introducción

Seguramente, alguna vez hayas realizado la copia o transmisión de un archivo de vídeo. Habrás observado que requiere mucho más tiempo que otros tipos de ficheros. En general, las aplicaciones multimedia, por su naturaleza, gestionan un volumen importante de datos.

Si analizas una imagen verás que está compuesta por miles o millones de píxeles que deben ser procesados y transmitidos.

Además, un vídeo está compuesto por varias imágenes por segundo (habitualmente 25) con lo que genera un volumen de datos aún mayor.

En cuanto al audio, se recoge con un micrófono cuya señal se muestrea obteniendo un elevado número de muestras cada segundo (habitualmente entre 8000 y 44000 muestras por segundo y canal de audio)

Un CD puede almacenar unos 70 minutos de sonido sin comprimir, pero tan sólo 30 segundos de vídeo en calidad TDT estándar. En el caso del DVD, esta duración subiría tan sólo a unos 7 minutos si siguiéramos sin usar compresión.

Si tomas como ejemplo una película de menos de dos horas en alta definición, necesitaríamos más de un terabyte para su almacenamiento, y varios Gbps para su difusión en streaming.

Por tanto, al transmitir o almacenar información multimedia, es prácticamente necesario introducir un esquema de compresión para poder utilizar eficientemente los recursos: bien sea el ancho de banda de la red o los sistemas de almacenamiento disponibles.

Ten en cuenta que en general podemos pensar que un sistema de codificación es más eficiente cuando alcanza mayores niveles de compresión, es decir, es capaz de representar los datos con un menor número de bits. Sin embargo, a lo largo del artículo trataremos algunos aspectos que hay que tener en cuenta, como qué sucede cuando la compresión es con pérdidas, es decir, cuando el proceso de codificación introduce una degradación en la calidad de los contenidos. También podemos considerar otros aspectos aparte del nivel de compresión, como la complejidad del sistema, que puede incrementar latencias y consumo de energía en el sistema global.

### 3 Objetivos

Una vez que el alumno lea con detenimiento este documento, será capaz de:

- Justificar la conveniencia del uso de codificadores a la hora de almacenar y transmitir información multimedia.
- Identificar y clasificar los sistemas de codificación.
- Calcular la entropía de un conjunto de símbolos en base a sus características estadísticas.

### 4 Desarrollo

#### 4.1 Sistemas de compresión

Tal y como se observa en la Imagen 1, un sistema de compresión está formado por un compresor, que toma los datos originales y los representa eficientemente, y un descompresor, que hace el paso inverso. A veces, al compresor se le denomina simplemente codificador, y al descompresor, decodificador. Cuando se transmiten los datos en red, fíjate como el compresor se ubica en el extremo origen de los datos, mientras que el descompresor está en la parte destino.

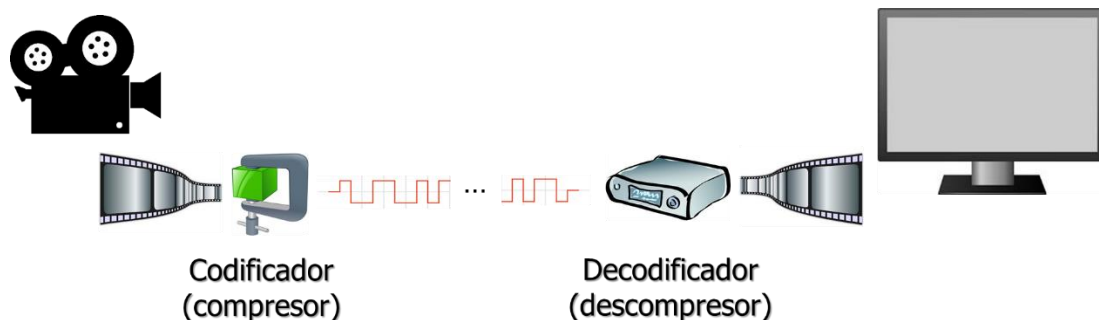


Imagen 1. Sistema de compresión

#### 4.2 Clasificaciones de los sistemas de compresión

En este punto vamos a estudiar diversas clasificaciones de los sistemas de compresión.

##### 4.2.1 En función de su complejidad

Podemos clasificar un compresor como **simétrico** o **asimétrico**, en función de su complejidad. En el primer caso, el codificador y el decodificador muestran una complejidad similar, mientras que en el segundo caso, uno de ellos es más complejo que el otro.

Habitualmente se evalúa la complejidad temporal. Pero también se pueden considerar complejidades espaciales como requerimientos de memoria, y consumos de energía.

Si nos centramos en la complejidad temporal, en el caso de un sistema asimétrico, el compresor suele ser el más costoso. ¿Por qué crees que es así?

Fíjate en aquellos pasos adicionales que un compresor puede necesitar completar.

Además, ¿en qué casos crees que es mejor un sistema simétrico y en qué casos uno asimétrico? Reflexiona sobre los sistemas de tiempo real (como videoconferencia) y escenarios como la distribución de películas, en soporte físico o bajo demanda.

#### 4.2.2 En función de su reversibilidad

También podemos clasificar los sistemas en función de si la codificación es reversible o no, es decir, si podemos garantizar que en la descompresión se recuperan exactamente los mismos datos (byte a byte) que fueron codificados. Si es así, el compresor es **sin pérdidas**, en inglés **lossless**. En caso contrario, el compresor es **con pérdidas**, en inglés **lossy**.

En compresión de información multimedia es habitual trabajar con sistemas con pérdidas, ya que estas pérdidas nos permiten una compresión mayor de los datos. La clave de estos sistemas es tratar de eliminar aquellos detalles que no percibimos los humanos. Si una frecuencia determinada no eres capaz de oírla, ¿para qué transmitirla?

En general, en un sistema de este tipo, multimedia con pérdidas, no podremos considerar tan sólo el nivel de compresión alcanzado, sino que deberemos relacionarlo con la calidad, tal y como se muestra en la Imagen 2. En esta imagen se observa que cuando el nivel de compresión es bajo (por ejemplo 1:8, es decir, reducir el tamaño de la imagen comprimida a la octava parte) la imagen obtenida es de mayor calidad, sin embargo al aumentar el nivel de compresión (en el ejemplo se comprime en 1:64, es decir, 64 veces más pequeña que la original), el deterioro de la calidad es evidente. Es decir, este tipo de curvas relaciona inversamente la calidad con el nivel de compresión.

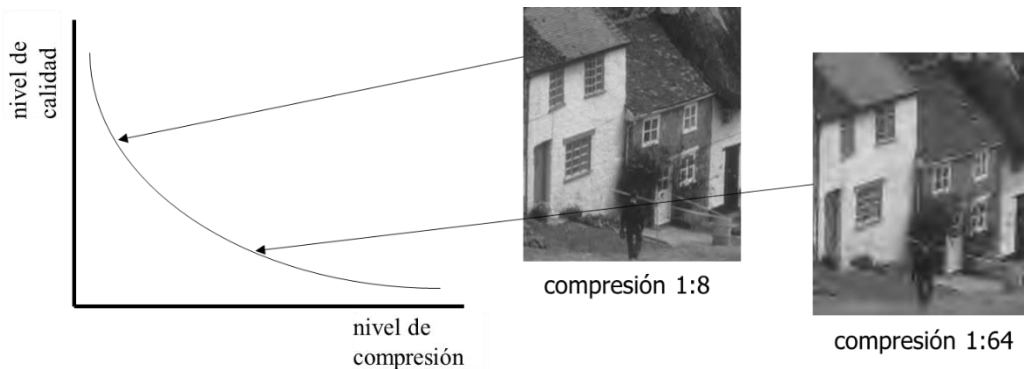


Imagen 2. Relación entre nivel de compresión y calidad (con imágenes)

#### 4.2.3 En función de su base de funcionamiento

Por último, vamos a estudiar los tipos de compresión que podemos definir en función de la esencia de los mecanismos en los que se basan los algoritmos de compresión. Así, podemos distinguir codificación basada en la fuente, en la entropía y en diccionarios.

- **Codificación basada en diccionarios**

Este tipo consiste en definir diccionarios de cadenas de texto, de forma que cuando aparecen esas cadenas en los datos, se sustituyen

por su índice en el diccionario. Un ejemplo de este tipo de codificación es LZ77, que es uno de los mecanismos que usa Winzip. La codificación basada en diccionarios no es muy común en multimedia, ya que presenta el **inconveniente** de que está orientada a un tipo de dato texto (y no a imagen, audio o vídeo), con lo que tan sólo estudiaremos en más profundidad los otros dos tipos.

- **Codificación basada en la fuente**

En la codificación basada en la fuente (en inglés, *source coding*) conocemos el tipo de los datos y los comprimimos en base al conocimiento de dicha fuente. Así, aplicamos algoritmos específicos según el tipo de fuente, que en multimedia pueden ser imagen, vídeo o audio.

Por ejemplo, en las imágenes hay determinadas componentes de frecuencia que el ojo no percibe. Sabiendo esto, podemos aplicar una transformada matemática para pasar de una representación de la imagen espacial a una representación en frecuencias, para así poder codificar solamente las frecuencias que sí que veamos. A este tipo de codificación se la denomina también codificación por transformación (o *transform coding*).

Como en este ejemplo que acabamos de ver, en muchas ocasiones estos compresores son con pérdidas, lo que nos permitirá alcanzar un nivel de compresión más elevado, lo que supone una **clara ventaja** respecto al resto de codificadores, con el **inconveniente** de introducir una pequeña pérdida de calidad en la señal (que será mayor cuanto más se comprima la fuente de datos).

Otro ejemplo de este tipo de codificación, en este caso sin pérdidas, es la cuenta de valores (o *run-length encoding*, con siglas RLE). En este compresor almacenamos el número de veces que se repite un valor. Este esquema es muy habitual en información multimedia y aparece, por ejemplo, en las muestras que representan un silencio en una señal de audio, en las que el valor 0 se repite mucho.

Una forma de implementar este tipo de codificación es la que hace JPEG para almacenar componentes de frecuencias en las que predomina el valor de 0. Consiste en almacenar cuantos ceros hay entre los números que son distintos de ceros. De esta forma se almacenan parejas de "cuenta de número de ceros" y "valores distintos de ceros".

Por ejemplo, dado el siguiente conjunto de valores:

(-2, 2, 4, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, -1, 0, 0, 0)

el algoritmo *run-length* de JPEG codificaría las siguientes parejas:

(0, -2) (0, 2) (0, 4) (10, 1) (5, -1) (3, n/d)

Así se indica que en primer lugar que no hay ningún 0 y aparece un -2, después tampoco hay ceros y aparece un 2, sigue sin haber ceros y tenemos un 4 y a continuación aparece una tira de 10 ceros antes del siguiente valor distinto de 0, que en este caso es un 1. Observa que en el ejemplo el último caso es especial, ya que sólo contiene cuentas de ceros.

- **Codificación basada en la entropía**

En la codificación basada en la entropía (*entropy coding* en inglés), se comprimen los datos en base a las meras propiedades estadísticas de los símbolos que los representan, desconociendo de qué naturaleza son estos datos. Al contrario que en la codificación basada en la fuente, aquí no sabemos si son parte de una imagen, de un vídeo, de un texto, o cualquier otro tipo de dato.

Su desarrollo se basa en un concepto de teoría de información denominado entropía, que nos permite conocer la cantidad de información que hay en un conjunto de símbolos de un alfabeto.

Estos compresores son de propósito general y sin pérdidas y básicamente tratan de identificar los símbolos o conjuntos de símbolos que más se repiten para representarlos con menos bits. Dos de los codificadores de este tipo más famosos son la codificación Huffman y la codificación aritmética.

El **principal inconveniente** de este tipo de codificación aplicada a la información multimedia es que alcanza unos niveles de compresión menores que los codificadores basados en la fuente, sin embargo, **la ventaja** es que es de uso general, y por tanto no es necesario conocer la naturaleza de la fuente, es decir, no hace falta saber si ese conjunto de datos representa una imagen, un audio, un vídeo, un texto, etc. Normalmente son codificadores que se usan como una parte de la codificación basada en la fuente.

### 4.3 Cálculo de la entropía

Dado un alfabeto  $S$ , del que conocemos la probabilidad de aparición de cada símbolo que lo forma, la entropía se calcula como la suma de la probabilidad de cada símbolo del alfabeto multiplicada por el logaritmo en base dos de la inversa de esa probabilidad. Es decir:

$$H(S) = \sum_i p_i \log_2 \frac{1}{p_i}$$

De forma práctica, la entropía nos indica el número de bits por símbolo que necesitamos, como mínimo, para codificar una cadena de ese alfabeto con codificación entrópica. Supone, por tanto, una cota sobre las prestaciones de este tipo de codificadores.

Veamos un ejemplo. Supón el siguiente alfabeto,

$$A=\{a, b, c\}$$

en el que la probabilidad de cada uno de los símbolos venga dada por las siguientes expresiones:

$$P(a)=4/7 \quad P(b)=1/7 \quad P(c)=2/7$$

Desarrollamos así la expresión para el cálculo de la entropía:

$$H = P(a) \log_2 \frac{1}{P(a)} + P(b) \log_2 \frac{1}{P(b)} + P(c) \log_2 \frac{1}{P(c)}$$

$$H = \frac{4}{7} \log_2 \frac{7}{4} + \frac{1}{7} \log_2 7 + \frac{2}{7} \log_2 \frac{7}{2} = 1,38$$

de forma que concluimos que las cadenas de este alfabeto no se pueden comprimir por debajo de 1,38 bits/símbolo con técnicas de codificación basadas en la entropía.

¿Sabrías calcular cuántos bits necesitamos, como mínimo, para poder codificar 10 símbolos de este alfabeto con un codificador entrópico? Ten en cuenta que el número de bits a emplear debe ser entero.

## 5 Cierre

Hemos justificado la necesidad de comprimir la información multimedia e introducido distintas clasificaciones. Como ejercicio final, puedes investigar a qué categorías pertenece el histórico código Morse en cada una de estas clasificaciones.

## 6 Bibliografía

Sayood, K: "Introduction to data compression", Morgan Kaufmann, 3<sup>rd</sup> edition, 2006, pág. 3-22.

Ghanbari, M: "Standard Codecs: Image Compression to Advanced Video Coding", IEE Telecommunications Series 49, 2003, pág. 25,41.