

Codificación de fuente

Francesc Tarrés
Margarita Cabrera

PID_00185031

Índice

Introducción	5
Objetivos	7
1. Fuentes de mensajes y codificación de fuente	9
2. Codificación de fuente y codificación de canal	11
3. Codificación PCM de fuentes analógicas	13
3.1. Ejemplo 1. Codificación PCM de señales de audio: formato CD-Audio	16
3.2. Ejemplo 2. Codificación PCM de imágenes: formato bitmap	17
3.3. Ejemplo 3. Codificación PCM de vídeo: formato ITU-601, Norma 4:2:0	19
4. Compresión de fuentes.	
Necesidad y conceptos básicos	21
4.1. La transformación de los datos	21
4.2. Cuantificador	22
4.3. Codificador entrópico	24
5. Medida de la información	27
5.1. Información asociada a un mensaje	27
5.2. Entropía de una fuente de mensajes	30
6. Códigos de longitud variable. Conceptos básicos	32
7. Códigos de Huffman	34
7.1. Obtención sistemática de los códigos de Huffman	34
7.2. Códigos de Huffman truncados	38
8. Transformaciones de datos	40
8.1. Codificación diferencial. Aplicación a la codificación de imagen	40
8.2. Predicción lineal. Una generalización de la codificación diferencial	43
8.3. Codificación diferencial y predicción de una señal de audio	44
8.4. Codificación diferencial adaptativa	49
9. Codificación de voz en telefonía móvil GSM	51
Actividades	53
Bibliografía	54

Introducción

El objetivo de los sistemas de comunicaciones digitales es enviar, con la máxima **fiabilidad** y **eficiencia**, una secuencia de bits que proporcionan un flujo de información desde el emisor al receptor. La mayor parte de los contenidos de esta asignatura se centran en el estudio de sistemas y procedimientos que permiten mejorar esta fiabilidad y eficiencia del sistema de comunicación, entendiendo que deseamos transmitir un máximo de bits por segundo (eficiencia) con la mínima degradación (fiabilidad). Por tanto, casi siempre supondremos que nuestro objetivo es transmitir datos sin cuestionarnos el origen de estos datos ni las características de la fuente que representan.

Para optimizar el sistema de comunicaciones completo, es conveniente utilizar procedimientos que permitan realizar una representación **eficiente** de la información en forma de secuencias de bits. En algunos casos, la propia información que deseamos transmitir es de naturaleza digital (ficheros de datos, documentos de texto, programas informáticos, etc.) por lo que la obtención de la secuencia digital es directa. En cambio, en otras aplicaciones es necesario definir un proceso que nos permita representar una información de naturaleza continua mediante una secuencia de bits. Además, deseamos que esta representación sea eficiente por lo que deberemos intentar **compactar (comprimir)** la secuencia de bits que representa la información. Así pues, entendemos como una representación eficiente del mensaje la reducción del número de bits que deben usarse para la transmisión del mensaje.

Esta última afirmación puede tener algunos matices en función de las aplicaciones. En efecto, en algunos casos es posible que la carga computacional requerida para que el equipo transmisor pueda realizar la compresión de los mensajes sea excesiva, aumentando el coste del transmisor más allá de su posible explotación comercial. En otras aplicaciones es posible que se disponga de un canal con la suficiente capacidad para transmitir la información sin necesidad de comprimirla. En cualquier caso, aunque el grado de compresión final que se aplica a los mensajes depende de un gran número de factores, es incuestionable que debemos conocer las técnicas que permitan realizar esta compresión con la mayor efectividad posible.

En este módulo estudiaremos el proceso mediante el cual se convierte una información de naturaleza analógica a una secuencia de bits y posteriormente analizaremos diversas técnicas para representar de forma compacta esta secuencia de bits. Para verificar que nuestra representación de la información es la adecuada, deberemos establecer mecanismos para “medir” la cantidad de información proporcionada por un mensaje. Con este objetivo introducimos algunos conceptos elementales de la teoría de la información como la defini-

ción formal de información, la entropía (o información media proporcionada por una fuente) y la representación de los mensajes mediante códigos de longitud variable. Finalmente, se introducen algunos ejemplos sencillos de sistemas de compresión de información para señales de audio y vídeo.

Objetivos

Los objetivos que debe alcanzar el estudiante con este módulo didáctico son los siguientes:

1. Comprender el concepto de codificación de fuente de forma eficiente.
2. Saber situar la codificación del mensaje dentro de un sistema de comunicaciones genérico.
3. Conocer elementos y bloques integrantes que intervienen en un compresor.
4. Comprender los fundamentos de la representación de información mediante secuencias de bits: PCM.
5. Distinguir entre entropía y tasa de bits media de un código.
6. Saber calcular códigos de longitud variable.
7. Comprender el proceso de codificación y decodificación mediante códigos de longitud variable.
8. Entender los mecanismos básicos para la compresión de información en sistemas diferenciales y predictivos.
9. Conocer los principios de codificación de la señal de voz en sistemas de telefonía GSM.

1. Fuentes de mensajes y codificación de fuente

En este apartado veremos que la naturaleza de los mensajes que deseamos transmitir puede ser analógica o digital. Los sistemas de comunicaciones digitales están pensados para transmitir información en formato digital, por lo que, cuando la fuente se presenta en este formato, puede ser transmitida de forma directa. No obstante, cuando la fuente se nos presente en formato analógico necesitaremos definir un procedimiento para convertirla en digital. En este apartado se definen las fuentes analógicas y los elementos que intervienen en la conversión y codificación de las mismas a formatos digitales.

Las fuentes que generan la información pueden ser de naturaleza analógica o digital. Las **fuentes digitales** pueden provenir de documentos de texto, hojas de cálculo, programas o cualquier otro tipo de datos informáticos. Su principal característica es que se representan, de manera natural, como secuencias de bits y que por lo tanto pueden ser transmitidas directamente mediante un sistema de comunicaciones digital.

En el caso de las **fuentes analógicas**, el mensaje que se desea transmitir suele ser una variable real que varía de forma continua en el tiempo (u otra variable independiente). Los ejemplos más típicos serían una señal de voz o de audio, la tensión proporcionada por un sensor de temperatura, el nivel de gris o de una componente de color en una imagen, etc.

En la figura 1 se representa una señal analógica típica que se corresponde con el nivel de tensión obtenido en la salida de un micrófono durante un intervalo de tiempo de 120 ms. La característica principal de las señales analógicas es que son funciones reales de variable real* que representan un fenómeno o magnitud física. Esto significa, para nuestro ejemplo, que en cualquier instante de tiempo el sistema de captura de la señal de voz proporciona un nivel de tensión que puede tomar cualquier valor real. A partir de esta definición, resulta evidente que las señales analógicas deberán convertirse en secuencias de bits antes de poder ser transmitidas mediante un sistema de comunicaciones digital.

* Esta definición puede extenderse al caso complejo, pero no lo consideraremos en esta asignatura.

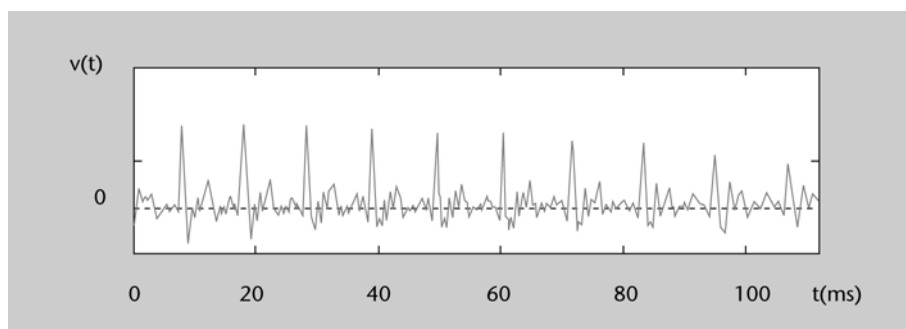


Figura 1. Representación de una señal analógica correspondiente a un fragmento de 120 ms de una señal de voz (vocal /o/).

Una **señal analógica** puede considerarse como una función real de variable real que representa un fenómeno o magnitud física.

2. Codificación de fuente y codificación de canal

En la figura 2 se representa un diagrama de bloques conceptual con los elementos básicos que intervienen en un sistema de comunicaciones digital. La primera etapa del emisor consiste en convertir cualquier fuente de señal, sea analógica o digital, en una secuencia de dígitos binarios aptos para su transmisión. Esta etapa se conoce como **codificación de fuente** o también como **compresión de datos**, y consiste en extraer toda la redundancia posible de la señal reduciendo el número de bits con el que se representa la información, sin que ello suponga una pérdida aparente de calidad para el receptor final del mensaje.

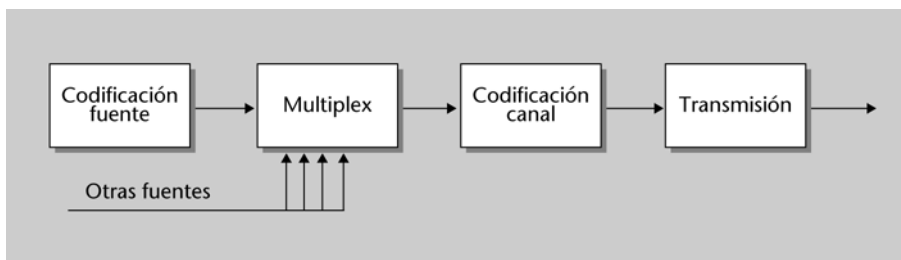


Figura 2. Elementos básicos de un sistema de comunicaciones digitales.

El proceso de codificación de una fuente analógica es más complejo que el de una fuente digital. En este último caso, la codificación de fuente consiste, si es que se considera necesario, en comprimir el flujo de bits de la fuente digital original.

En el caso de fuentes analógicas, es necesario definir primero un procedimiento para convertir la información analógica en digital, lo que siempre supone una posible pérdida de calidad o, en otras palabras, la introducción de **ruido** o **distorsión**. En el apartado siguiente describiremos brevemente la codificación PCM, que es el procedimiento más utilizado para convertir una fuente analógica en digital.

Una vez la señal analógica se ha convertido en digital, todavía es posible procesarla para reducir el número de bits que la representan. Los métodos y estrategias empleados para la compresión de la señal son muy variados, por lo que resulta difícil establecer un procedimiento óptimo, ya que los resultados dependen, en gran medida, de las características de las señales. Generalmente, los métodos de compresión avanzados combinan simultáneamente diferentes métodos para codificar la señal.

La **codificación de fuente** es el proceso por el cual se convierte una fuente digital o analógica en una secuencia de dígitos binarios que la representa. Uno de los objetivos de la codificación de fuente es extraer la redundancia existente en la fuente para representarla de la forma más eficiente posible.

El bloque de **multiplexación de fuente** incluido en la figura 2 sólo está presente en algunos sistemas de comunicación digital que generalmente son servicios audiovisuales avanzados. El concepto de multiplexación es simple: las tramas de bits procedentes de fuentes diferentes (audio, vídeo) comparten un mismo recurso del medio de transmisión.

Un ejemplo típico en el que se realiza la multiplexación de varias fuentes es la televisión digital terrestre (TDT). En TDT, un canal proporciona información sobre varios programas de televisión mientras que en televisión analógica cada programa está asociado a un único canal. Así, por ejemplo, la información que se transmite a través de un canal de TDT incluye diversas fuentes de mensajes: 4 o 5 fuentes de vídeo (una por programa), varias fuentes de audio multicanal con la banda sonora y la versión original de cada programa, fuentes de audio con las versiones dobladas, información de datos de teletexto avanzado, datos interactivos, etc. El conjunto de todas estas fuentes multiplexadas se denomina **flujo de transporte** (TS, *transport stream*).

Es importante distinguir entre la multiplexación de fuentes y la multiplexación de canales o de portadoras. La TDT puede ser un buen ejemplo para facilitar esta distinción. En efecto, las portadoras están multiplexadas en el espectro radioeléctrico y se corresponden a los programas de TV analógica convencional. Para cambiar el programa en TV analógica, debe sintonizarse en el receptor una portadora (canal) distinta. En cambio, en TDT cada portadora contiene distintos programas de televisión con las fuentes multiplexadas.

Ejemplo

La portadora TDT de Televisión Española contiene los programas de TVE1, TVE2, 24H TVE, Clan 50 y Teledeporte.

Refiriéndonos de nueva a la figura 2, encontramos el bloque de la **codificación de canal**. Esta codificación se utiliza para acondicionar las señales a las características del canal. En esta etapa se introducen códigos de protección frente a errores, códigos de encriptación de la información y se adecua la forma de onda de las señales para que puedan ser utilizadas por el bloque de modulación o transmisión. Se trata, por tanto, de adaptar los datos a las características del medio de transmisión, para garantizar una correcta recepción de las señales. El medio de transmisión condiciona el tipo de **modulación** que se utiliza y la codificación de canal. Así, en un medio como el cable, los sistemas de detección y corrección de errores pueden ser más simples que en el caso de la comunicación vía satélite debido a que la relación señal a ruido es mucho mayor y, por tanto, se reduce la probabilidad de error.

3. Codificación PCM de fuentes analógicas

En este apartado veremos el proceso más utilizado para convertir una señal analógica en digital, conocido como codificación PCM. El esquema básico de este procedimiento se muestra en la figura 3.

La señal analógica original se aproxima mediante una retícula rectangular. La separación entre los elementos de la retícula en el eje de abscisas es constante y se conoce como el **periodo de muestreo** de la señal. El teorema del muestreo establece que si el número de muestras por segundo es mayor que el doble del ancho de banda de la señal, ésta podrá recuperarse a partir de sus muestras sin ambigüedades. Bajo estas condiciones, en principio, la representación de la señal no mejora aunque aumentemos la densidad de muestras en el eje temporal.

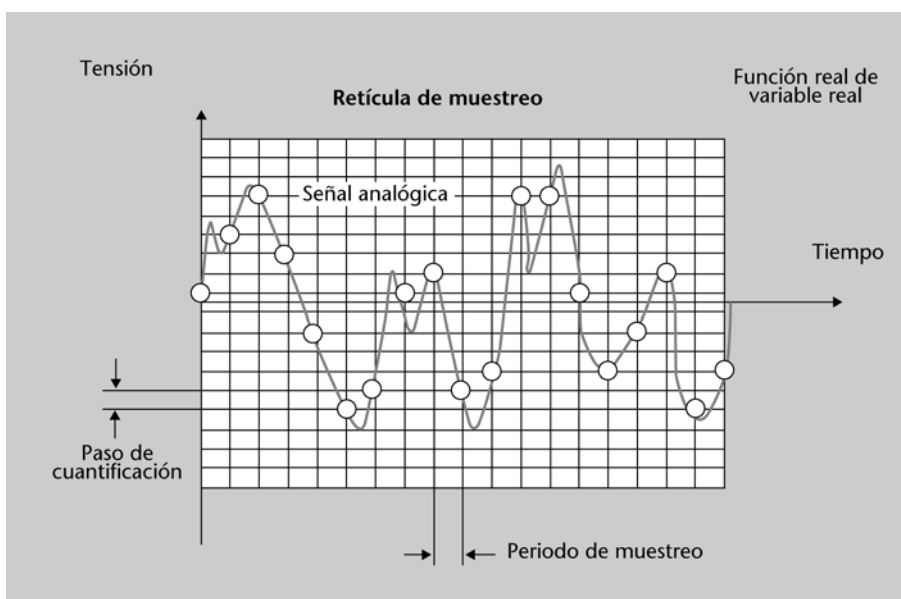


Figura 3. Muestreo y cuantificación de una fuente analógica.

El intervalo de separación en el eje vertical se denomina **paso de cuantificación**. En este caso, al reducir el intervalo entre dos pasos de cuantificación consecutivos, se mejora la representación de la señal. Un intervalo de cuantificación excesivamente elevado introduce un error significativo en la representación de la señal que, a efectos prácticos, se traduce en la presencia de efectos indeseados. Estos efectos se conocen como **ruido de cuantificación** y se muestran gráficamente en la figura 4. El nombre de *ruido de cuantificación* proviene de aproximar el nivel de la señal por uno de los posibles niveles de cuantificación, lo que puede considerarse equivalente a haber sumado un ruido a la señal.

Una vez fijado el paso de cuantificación y los niveles máximo y mínimo que puede tomar la señal, también queda determinado el número total de

posibles niveles de cuantificación que se utilizarán para la codificación. Ahora bien, como el número de niveles de cuantificación es finito, podemos asociar una palabra código distinta a cada uno de ellos, por lo que el valor de la señal puede especificarse mediante palabras código de un alfabeto finito.

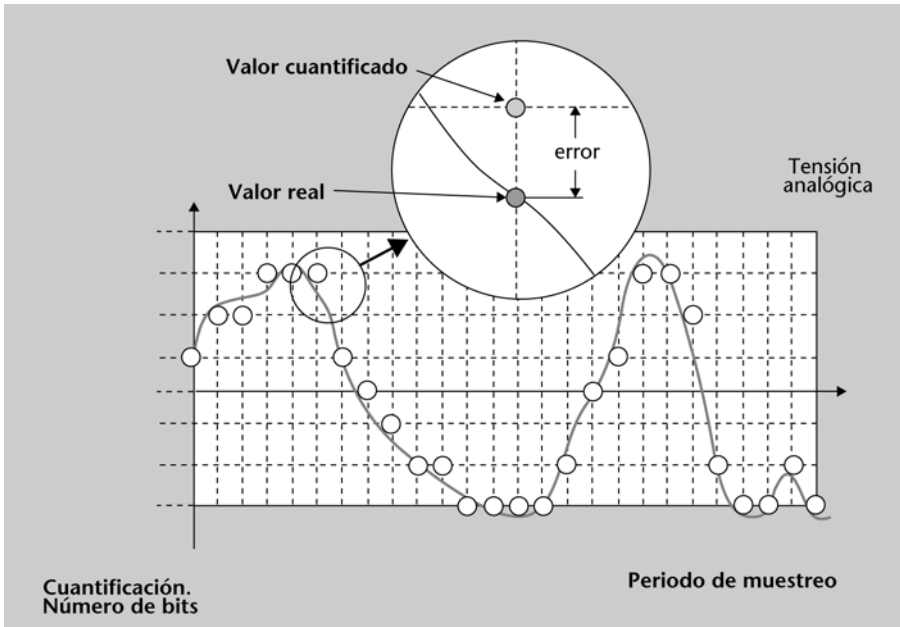


Figura 4. El error que se comete al aproximar el valor de la señal por uno de los niveles de cuantificación es equivalente a introducir un ruido (ruido de cuantificación).

En la figura 5 se muestra este proceso de forma simplificada utilizando un alfabeto reducido de 8 posibles niveles de cuantificación. Cada nivel de cuantificación tiene asignado un código binario (palabra código). La transmisión de este código binario proporciona información del nivel de tensión de la señal en cada instante de muestreo. Este proceso de codificación de la fuente se conoce como PCM (*pulse code modulation*).

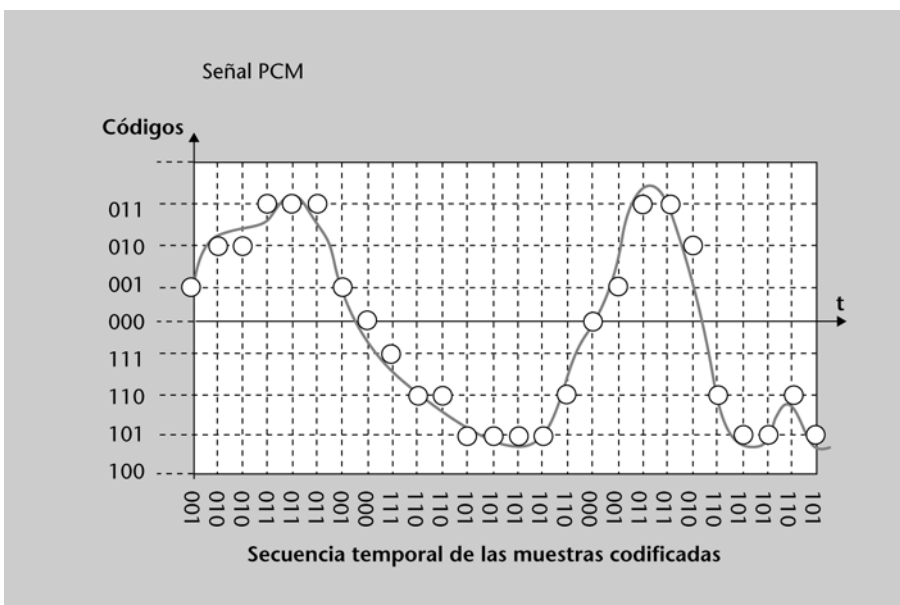


Figura 5. Codificación de una fuente analógica en PCM.

El número de niveles del cuantificador (L) se elige como una potencia de 2. De esta forma, cada nivel de cuantificación puede asociarse a una palabra código de B bits. La relación entre el número de niveles del cuantificador (L) y el número de bits por muestra (B) viene dada por:

$$L = 2^B.$$

Ejemplo

En un sistema PCM de 16 bits por muestra, la señal analógica se cuantifica en $2^{16} = 65.536$ niveles distintos.

Mediante este proceso de muestreo y cuantificación, la fuente analógica original ha sido convertida en una fuente digital. En efecto, al realizar el muestreo temporal, la señal analógica se convierte en una secuencia de muestras. El valor de amplitud de cada muestra puede codificarse mediante la selección de una palabra código de un alfabeto finito, de manera que el envío del mensaje se convierte en la transmisión de una secuencia determinada de palabras código. En el ejemplo de la figura 5, podríamos transmitir una aproximación a la fuente analógica original enviando la siguiente secuencia de mensajes 001 010 010 011 011... (que son los valores que toman los códigos de la señal en el eje horizontal de la figura).

En la práctica, el paso de cuantificación idóneo viene determinado por las características del receptor del mensaje, que para señales de audio y vídeo suele ser el ser humano. Consideremos como ejemplo la digitalización de una señal de audio para aplicaciones de reproducción de sonido en alta fidelidad. En este caso, las características del sistema auditivo humano son las que determinan el paso de cuantificación máximo para que la señal pueda reproducirse sin que se aprecie ninguna diferencia en relación con la señal original. El valor del paso de cuantificación óptimo se determina efectuando pruebas exhaustivas de calidad subjetiva obtenidas con distintos oyentes e intervalos de cuantificación. El principio básico es que no tiene sentido reducir el paso de cuantificación más allá de las capacidades del sistema auditivo.

En el caso de que el destinatario final de la información no sea el ser humano, el procedimiento es parecido. Consideremos como ejemplo un sistema de visión artificial que debe realizar el reconocimiento óptico de matrículas de automóvil. Ahora, el criterio para determinar el paso de cuantificación es que las capacidades de reconocimiento del sistema no se vean limitadas por el hecho de utilizar un número excesivamente reducido de niveles. El intervalo de cuantificación adecuado es, nuevamente, el máximo valor que permite mantener las prestaciones de reconocimiento de caracteres del sistema.

En los casos considerados, las prestaciones del sistema no se ven afectadas si se utiliza un paso de cuantificación menor que el que hemos definido como idóneo. No obstante, reducir el paso de cuantificación supone aumentar el número de bits, lo que en general representa un incremento en el coste tecnológico del sistema. En efecto, veremos a continuación algunos ejemplos de codificación en PCM de señales de audio, imágenes o vídeo, y determinaremos para cada uno de ellos el número de bits asociado a la fuente. Observaremos que, en todos los casos, la codificación de la fuente en PCM supone un número de bits muy ele-

vado que será conveniente comprimir si deseamos transmitir la información de manera eficiente.

3.1. Ejemplo 1. Codificación PCM de señales de audio: formato CD-Audio

El formato **CD-Audio** (*compact disc digital audio*) fue el primer sistema que popularizó el consumo de información audiovisual en formato digital. La aparición del estándar es el resultado de una estrecha colaboración entre las empresas Philips y Sony, en la que se definieron las características técnicas del sistema. Los objetivos iniciales eran conseguir un sistema de reproducción de audio de alta calidad y que pudiera soportar un gran número de reproducciones sin deteriorarse, ya que éste era el principal problema de los discos de vinilo. En pocos años, el sistema **CD-Audio** se convirtió en el estándar de la industria discográfica y desplazó casi por completo a los discos de vinilo y a las cassetes compactas.

El sistema está basado en una digitalización directa de la señal de audio analógico mediante una codificación PCM. Así pues, tenía que especificarse la frecuencia a la que se tomarían las muestras de la señal de audio y el número de niveles de cuantificación con los que se codificaría cada muestra. Para la frecuencia de muestreo se fijaron 44.100 Hz (muestras por segundo). Este número cumple con el **criterio de Nyquist**, ya que es superior al doble del ancho de banda de la señal de audio, que puede considerarse de 20 kHz. El valor concreto de 44.100 muestras por segundo se debe a que los primeros sistemas de registro de audio digital, desarrollados por la BBC a finales de la década de 1960, ya utilizaron esta frecuencia de muestreo.

Para determinar el número de niveles de cuantificación, se hicieron numerosas pruebas de calidad con especialistas en audio. Se aceptó que, con un total de 14 bits (16.384 niveles = 2^{14}), podía obtenerse una calidad de audio indistinguible de los originales analógicos, aunque finalmente se acordó tomar un total de 16 bits por muestra como estándar ($2^{16} = 65.536$ niveles). Las palabras de 16 bits (2 bytes), además de ofrecer una mayor calidad teórica, son compatibles con todas las memorias y procesadores digitales. No obstante, en la primera época de comercialización de discos compactos, algunas grabaciones fueron realizadas con 14 bits debido a la complejidad tecnológica asociada a las precisiones de 16 bits (los dos bits de menor peso estaban siempre a cero).

La **tasa de bits** (*bitrate*) de una señal PCM se define como el número medio de bits que se requieren para codificar un fragmento con una duración de un segundo. La tasa de bits se determina como el producto entre la frecuencia de muestreo y el número de bits por muestra:

$$r_b = f_m \cdot B.$$

Teniendo en cuenta la frecuencia de muestreo, el número de bits por muestra y que el sistema CD-Audio utiliza dos canales de información (estéreo), podemos calcular la tasa de bits que supone el almacenamiento o la transmisión de esta información:

$$r_b = 2 \text{ canales} \times \frac{44.100 \text{ muestras}}{\text{canal}} \times \frac{16 \text{ bits}}{\text{muestra}} = 1.411.200 \text{ bps} .$$

El valor de 1,4 Mbps obtenido hace que el sistema de CD-Audio no pueda ser utilizado de forma práctica en un gran número de aplicaciones en las que el ancho de banda del canal de transmisión o la capacidad de memoria del sistema sean recursos escasos. Por ello, han aparecido un gran número de alternativas para intentar comprimir el flujo total de bits sin que se aprecie una disminución de la calidad de la señal. Entre estas alternativas destacan sistemas como el *minidisc*, MP3, AAC (*advanced audio coding*), etc. Mediante el sistema de compresión MP3 o el AAC, podemos conseguir la misma calidad perceptual que con CD-Audio con tasas de 128.000 bps para señales estereofónicas, lo que significa una compresión aproximada en un factor 12 (la señal comprimida ocupa una doceava parte de la señal original).

3.2. Ejemplo 2. Codificación PCM de imágenes: formato bitmap

La digitalización de una imagen también consiste en un muestreo y una cuantificación. En este caso el muestreo se realiza en el plano de imagen y queda determinado por el número de filas y de columnas en las que se descompone una imagen. La imagen puede interpretarse como una matriz, tal y como se representa en la figura 6. Los elementos de la matriz se denominan **elementos de imagen** o **píxeles** (*picture elements*).

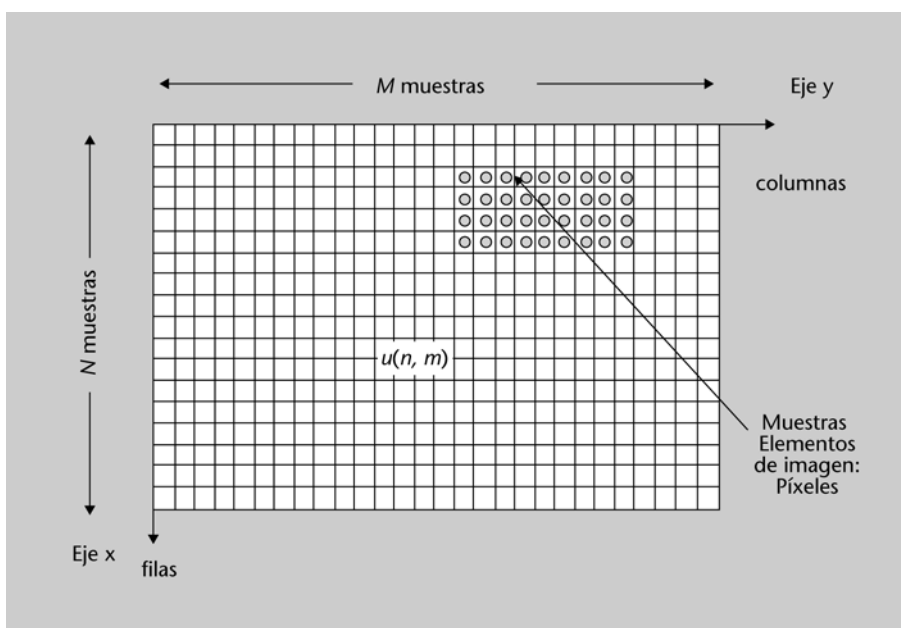


Figura 6. Interpretación de una imagen como una matriz.

El número de filas y columnas en las que se descompone una imagen depende de la aplicación. En televisión de definición convencional se toman 576 filas \times 720 columnas (Europa), en televisión de alta definición 1.080 filas \times 1.920 columnas mientras que en una cámara de fotografía de 6 Mpíxeles se tomarían unas 2.000 filas \times 3.000 columnas.

Del mismo modo, el número de niveles en los que se cuantifica cada elemento de imagen (píxel) depende de las características de la aplicación. En blanco y negro, el sistema visual sólo puede distinguir alrededor de unos 40 a 45 niveles de gris distintos. En la práctica, las imágenes en blanco y negro se cuantifican con 256 niveles de gris (8 bits) que, aunque son más que los necesarios, facilita estructurar la memoria y los ficheros asociados a imágenes en palabras de 8 bits. Este formato de codificación PCM se conoce como **BITMAP de 8 bits** y se utiliza para imágenes en blanco y negro. La codificación de imágenes en color suele realizarse tomando también 256 niveles para cada una de las componentes primarias: rojo, verde y azul (RGB, *red, green, blue*). Esta codificación da lugar a las denominadas imágenes de 24 bits por píxel o **BITMAP de 24 bits** (8 bits para cada componente de color). Existen otros formatos alternativos como el BITMAP de 15 bits o el de 16 bits, aunque cada vez son menos utilizados. Así, como ejemplo, el formato BITMAP de 16 bits consiste en codificar las componentes de rojo y azul con 5 bits, mientras que la componente verde está codificada con 6 bits. Los **BITMAP de 32 bits** codifican cada una de las componentes con 8 bits y utilizan 8 bits adicionales para codificar la “transparencia” de la imagen (grado en que la imagen se superpone con el fondo).

El número de bits necesario para almacenar una imagen sin comprimir puede calcularse como el número total de píxeles, es decir, el número de filas por el número de columnas ($N_{filas} \times M_{col}$) por el número de bits que ocupa cada píxel (*bpp, bits per pixel*)

$$N_{bits} = N_{filas} \cdot M_{col} \cdot bpp.$$

Como un ejemplo para determinar el número total de bits que puede ocupar una fotografía, calcularemos el tamaño de memoria necesario para almacenar o transmitir una imagen en color tomada por una cámara de fotografía digital de 6 Mpíxeles.

$$N_{imagen} = 2.000 \text{ filas} \times 3.000 \text{ columnas} \times 24 \text{ bits} = 144.000.000 \text{ bits} .$$

Nuevamente, el número de bits resultante es considerable (tardaríamos aproximadamente 2,5 minutos en transmitir esta imagen a través de una línea ADSL de 1 Mbps), lo que nos vuelve a indicar la importancia de comprimir la información antes de almacenarla o transmitirla. Los formatos de compresión de imagen más populares son el JPEG y el TIFF comprimido. El JPEG permite

comprimir con factores del orden de 10 o superiores sin que puedan apreciarse diferencias con los archivos originales en PCM.

3.3. Ejemplo 3. Codificación PCM de vídeo: formato ITU-601, Norma 4:2:0

Consideraremos ahora la digitalización de una señal de vídeo y calcularemos el número total de bits que requiere el almacenamiento o transmisión de una secuencia de vídeo digital sin comprimir.

El estándar ITU-601 define distintos modos para codificar la señal de vídeo digital sin comprimir. Estos modos dependen del estándar analógico de procedencia (NTSC, PAL) y de las aplicaciones a las que se orienta (estudios de televisión, difusión audiovisual). Las normas más utilizadas actualmente son la 4:2:2 y la 4:2:0. La primera se utiliza principalmente en estudios de televisión y para el intercambio de materiales entre productoras. La norma 4:2:0 es la que se utiliza en la difusión de señales de televisión o DVD.

Los sistemas de televisión en color, por cuestiones de compatibilidad con los sistemas en blanco y negro, no transmiten las tres componentes primarias (R, G y B), sino la componente de luminancia (Y, señal de blanco y negro) más dos componentes de croma adicionales (Cr y Cb), que proporcionan la información de color al receptor (componentes diferencia de color). El receptor de color puede calcular las componentes RGB a partir de las señales recibidas YCrCb. Obsérvese que las componentes RGB y las componentes YCrCb pueden considerarse como dos sistemas de coordenadas de color distintos, teniendo tres variables independientes en cada uno de ellos. Podremos pasar de un sistema de coordenadas a otro mediante matrices de cambios de base (sistemas de ecuaciones).

El sistema de coordenadas basado en las componentes YCrCb resulta más útil que el sistema RGB debido principalmente a dos motivos:

- 1) Por cuestiones de compatibilidad con los antiguos sistemas de televisión en blanco y negro, que nos obligaban a transmitir la información de luminancia.
- 2) El más importante, el sistema visual humano, presenta una mayor resolución a la componente de luminancia que a las componentes de color.

Podemos comprobar este último hecho situando dos hilos de diferente color a cierta distancia de un espectador. Si la distancia es la adecuada, podremos comprobar que el espectador es capaz de discriminar que existen dos hilos pero no puede distinguir sus colores. Este experimento nos permite inferir que deberemos “tratar” mejor, es decir, dedicarle más bits, a la componente de luminancia que a las componentes de croma.

Las normas 4:2:2 y 4:2:0 que hemos comentado anteriormente utilizan este principio para almacenar/transmitir una cantidad de información menor para las componentes de croma que para las de luminancia. En concreto, la norma 4:2:2 dedica la mitad de información a cada una de las componentes de croma respecto a la luminancia. La norma 4:2:0 dedica una cuarta parte de la información a las componentes de croma respecto a la luminancia.

En el caso del sistema de televisión europeo, la norma 4:2:0 tiene las siguientes características de codificación para cada una de las componentes:

	Y	Cr	Cb
Número de imágenes por segundo	25 imágenes	25	25
Número de líneas por imagen	576 líneas	288	288
Número de columnas por imagen	720 columnas	360	360
Número de bits por componente	8/10/12	8/10/12	8/10/12

Tabla 1. Características de codificación para la norma 4:2:0 del sistema de televisión europeo.

Si suponemos que cuantificamos cada componente con 256 niveles (8 bits), la tasa de bits necesaria para transmitir una señal de televisión en formato 4:2:0 vendrá dada por:

$$r_b = 25 \frac{\text{imágenes}}{\text{s}} \times (576 \times 720 + 2 \times 288 \times 360) \frac{\text{componentes}}{\text{imagen}} \times 8 \frac{\text{bits}}{\text{componente}} = 124,416 \cdot 10^6 \text{ bits/s.}$$

Por tanto, la señal de vídeo sin comprimir requiere un número de bits muy importante por lo que la norma 4:2:0 (y también la 4:2:2) sólo se utiliza en intercambios de señal entre equipos de vídeo. Si seguimos poniendo nuestra línea ADSL de 1 Mbps, necesitaríamos alquilar un total de 125 líneas para transmitir una señal de vídeo en tiempo real de acuerdo con esta norma. Este hecho nos indica que antes de transmitir o almacenar la información de vídeo, deberemos comprimirla, para lo que generalmente se utiliza el estándar de compresión MPEG-2. La norma 4:2:0 es especialmente importante debido a que es el formato de origen (es decir, antes de comprimir) que se utiliza en la difusión de televisión digital, tanto en su versión cable como en las versiones satélite o terrestre. También es el formato que se utiliza en el sistema DVD-Vídeo antes de realizar la compresión en MPEG-2.

4. Compresión de fuentes.

Necesidad y conceptos básicos

La digitalización de señales analógicas aporta considerables ventajas entre las que pueden citarse la protección frente a ruidos, la facilidad para encriptar las señales, la posibilidad de procesar digitalmente los datos, etc. No obstante, también adolece de algunos inconvenientes. Hemos visto que, en general, la digitalización directa mediante PCM supone un elevado número de bits. En efecto, la señal de audio analógica en banda base tiene un ancho de banda de unos 20 kHz, pero al digitalizarla y codificarla en PCM se convierte en una señal de unos 700 kbps (hemos calculado unos 1,4 Mbps para los dos canales estereofónicos). Del mismo modo, una señal de vídeo analógico tiene un ancho de banda aproximado de 5 MHz. En cambio, la digitalización de la secuencia de vídeo en un formato PCM como el 4:2:0 requiere transmitir una tasa de bits de $124,4 \cdot 10^6$ bits/s = 124,4 Mbps.

Aunque se usen códigos y modulaciones digitales de gran eficiencia espectral, los recursos espectrales (anchos de banda) necesarios para transmitir las tasas de bits asociadas a las señales digitalizadas serán muy superiores a los que se requieren con los sistemas analógicos.

Tomando la codificación PCM como punto de partida, es necesario aplicar un proceso de reducción de la tasa de bits que permita la transmisión de las señales sin una pérdida aparente de calidad. Todos los sistemas de compresión de información utilizados actualmente pueden descomponerse en los tres módulos o elementos que se muestran en la figura 7, y que analizaremos con cierto detalle en los siguientes apartados.

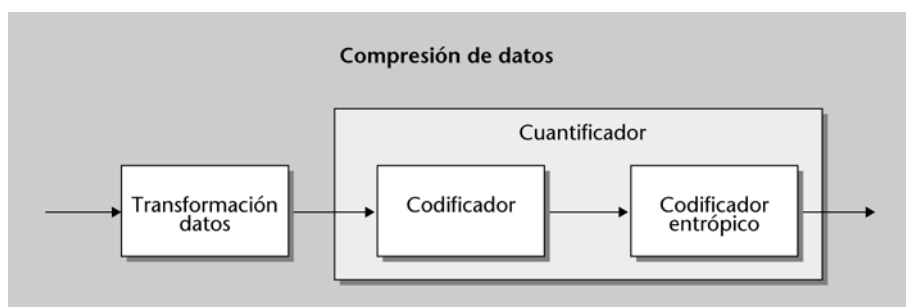


Figura 7. Elementos de un sistema de compresión de datos.

4.1. La transformación de los datos

El bloque de transformación de datos es un procedimiento genérico que se utiliza para representar la información en una forma alternativa y en la que resulta más evidente la redundancia existente en los datos originales. Es

necesario que esta transformación sea invertible, es decir, que a partir de los datos transformados podamos recuperar exactamente la información original. Todas las transformaciones utilizadas en los estándares actuales de compresión de audio y vídeo son invertibles, lo que significa que en este bloque concreto no se introduce ningún tipo de pérdida.

Consideremos como ejemplo que aplicamos una transformada de Fourier a las señales que representan la información de un fragmento de audio. Es posible que resulte más eficiente codificar los datos resultantes de esta transformación que los datos originales. En efecto, en el dominio transformado de Fourier, es más simple identificar las regiones espectrales que tienen una energía por debajo del umbral de audición, o analizar las bandas que resultarán enmascaradas por otras señales y que, por lo tanto, no serían audibles. Si este análisis de la señal es factible, no tendrá sentido codificar toda aquella información que el sistema auditivo humano es incapaz de escuchar y resultará más rentable dedicar los bits a las zonas audibles. El resultado final es que, en este dominio transformado, podemos codificar la información de forma más eficaz. El sistema MP3 utiliza los principios de codificación expuestos para realizar la compresión de señales de audio de alta fidelidad.

La transformación de los datos puede ser de naturaleza muy distinta (Fourier, coseno, *wavelet*, diferencia, etc.) o incluso puede no resultar necesario realizar la transformación si es que existen procedimientos eficientes para la compresión directa o los recursos computacionales de la aplicación son escasos. En algunos casos, es posible que existan varias transformaciones realizándose simultáneamente para diferentes componentes de la señal y que la codificación se realice tomando datos parciales de cada una de ellas. Éste es el caso, por ejemplo, del estándar MPEG-2 vídeo, en el cual se combinan distintas estrategias de codificación en función del tipo de información y de las características de la señal.

4.2. Cuantificador

El bloque de cuantificación es el principal responsable de la introducción de las pérdidas en el sistema de compresión. Esencialmente, su función es la de aproximar los resultados obtenidos en la salida del bloque de transformación mediante un conjunto finito de posibles valores (alfabeto finito de posibles resultados). Esta aproximación realizada sobre los resultados de la transformación impide que pueda recuperarse de forma exacta la señal original. Por tanto, en las aplicaciones en las que se desee recuperar la información de forma exacta, el bloque de cuantificación no deberá estar presente. La existencia o no de este bloque en el esquema general del compresor determina una posible clasificación de los métodos de compresión.

La compresión puede ser con pérdidas (*lossy*) o sin pérdidas (*lossless*) en función de si la información que se recupera coincide exactamente con la original

o es sólo una aproximación. Los métodos de **compresión sin pérdidas** se utilizan sobre todo en aplicaciones de codificación de datos binarios de aplicaciones informáticas, en los que es absolutamente necesario recuperar la información original. Los formatos más populares son el zip, el rar y el arj. No obstante, todos estos métodos de compresión (winzip, winrar) presentan unos resultados extremadamente ineficaces cuando se aplican a señales audiovisuales. Es fácil comprobar que cuando se intenta comprimir un fichero bitmap (*.bmp) o un archivo de audio PCM (*.wav) mediante este tipo de utilidades los factores de compresión obtenidos son muy pequeños. Para resolver este problema, es necesario desarrollar algoritmos sin pérdidas, específicos para las aplicaciones de audio, imagen y vídeo. Generalmente, estos sistemas específicos consiguen obtener factores de compresión del orden de 2 o 3, mejores que los genéricos winzip, winrar pero insuficientes para muchas aplicaciones de transmisión o almacenamiento de datos.

En tratamiento de imágenes, los métodos de compresión sin pérdidas encuentran su aplicación en la codificación de imágenes médicas o científicas en las que puede resultar crítica la pérdida de parte de la información. El formato JPEG dispone de una versión de codificación sin pérdidas que suele emplearse en estas aplicaciones. En audio también existen algunas aplicaciones que permiten comprimir la información pudiendo recuperar de forma exacta el original. Su aplicación más conocida es la transferencia de archivos musicales, especialmente de música clásica en calidad CD-Audio, a través de Internet. El formato de compresión de audio sin pérdidas más extendido es el APE.

La **compresión con pérdidas** es la más habitual en la codificación de señales de vídeo y audio. Las pérdidas son tolerables siempre que la calidad de las señales descodificadas sean aceptables. El principio general sobre el que se sustenta la codificación con pérdidas es que no resulta necesario codificar aquellas componentes de la información que no son audibles o visibles por los sistemas de percepción humana. Por lo tanto, estos métodos se basan en las características psico-fisiológicas de los sistemas auditivo y visual que son, en última instancia, los que deben evaluar la calidad del algoritmo de compresión. Por este motivo, es fundamental comprender las limitaciones y características de estos sistemas de percepción para diseñar codificadores en los que las pérdidas de información resulten poco evidentes o incluso inapreciables.

La principal ventaja de estas estrategias de codificación es que consiguen unos factores de compresión sensiblemente mayores que los que se obtienen con los métodos sin pérdidas. Además, suelen ser métodos escalables con la aplicación, es decir, el grado de pérdida de calidad que se tolera depende del ámbito al que se destine el codificador. Así, la calidad que se exige en una aplicación de vídeo en directo a través de una red de telefonía móvil es mucho menor que la que se exige para la difusión de televisión. Por lo tanto, un codificador de vídeo para aplicaciones de videoconferencia en redes móviles deberá realizar una compresión de los datos mucho mayor que un codificador

para aplicaciones de televisión ya que las tasas de transferencia de bits que admiten ambas aplicaciones son claramente diferentes.

Cuantificación, compresión con pérdidas y calidad

Consideremos un esquema de compresión como el de la figura 7 en el que la transformación que se aplica a los datos es la transformada coseno. Los resultados de aplicar esta transformada dependen de los valores de la señal PCM de entrada (número de bits con la que ha sido codificada) y de la precisión numérica con la que se ha realizado la transformada. Un ejemplo típico sería que los números obtenidos en la salida del bloque transformador sean enteros comprendidos entre -8.192 y 8.191 (la transformada se realiza en coma fija con una precisión de 14 bits). En una primera aproximación podemos suponer que si mantenemos esta precisión podremos recuperar de forma prácticamente exacta la información de la señal original.

La función del bloque cuantificador es reducir el número total de posibles palabras que deberán codificarse. En nuestro ejemplo, cada elemento de la transformada puede tomar 2^{14} posibles valores. El bloque cuantificador puede aproximar el resultado de la transformada en, por ejemplo, pasos de 8, de manera que el número obtenido después de la transformación siempre se aproxima al múltiplo de 8 más cercano. Así, en este ejemplo, los únicos resultados posibles serían..., -24 , -16 , -8 , 0 , 8 , 16 , ... con lo que el número total de posibles mensajes a codificar se reduce de 16.384 a $2.048 = 2^{11} = 2^{14}/2^3$.

Esta reducción en el número de posibles palabras a transmitir supondrá una disminución del número total de bits a enviar. Así, por ejemplo, ahora sólo se requieren 11 bits para enviar uno de los resultados mientras que originalmente se requerían 14. El coste de esta compresión es que los valores de la transformada han sido aproximados y la información original sólo podrá recuperarse de forma aproximada.

El factor de compresión está directamente relacionado con el paso de cuantificación. Estándares como el JPEG, o las diferentes versiones de los MPEG audio y vídeo, permiten que el usuario especifique el factor de compresión que desea obtener. Aumentar el factor de compresión es equivalente a aumentar el paso de cuantificación. En efecto, si en nuestro ejemplo ponemos un paso de cuantificación de 32, ahora todos los resultados de la transformada deberán aproximarse por múltiplos de 32. El número total de posibles resultados se reduce a $512 (= 2^9 = 2^{14}/2^5)$, con lo que cada mensaje podrá codificarse con menos bits. No obstante, aparece una pérdida de calidad debido a que los resultados han sido aproximados con un error mayor.

La relación existente entre el factor de compresión y la calidad no siempre es evidente. En algunos casos, puede irse aumentando el factor de compresión sin que el resultado afecte, al menos aparentemente, a la calidad. En otros casos, un pequeño cambio en el factor de compresión puede suponer una pérdida muy importante de calidad. En general, el margen de trabajo ideal para el paso de cuantificación es aquél en el que aparece una relación directa entre el factor de compresión y la calidad (*graceful degradation regions*).

4.3. Codificador entrópico

El último bloque de un compresor que nos queda por discutir es el codificador entrópico. En general, el bloque de codificación consiste en asignar una palabra de código binario a cada uno de los posibles mensajes que puede generar una fuente. En nuestro caso, atendiendo a la disposición de los elementos en la figura 7, la fuente que debemos codificar es la salida del cuantificador.

En el ejemplo anterior, el resultado de transformar y cuantificar la señal nos daba números enteros, múltiplos de 8 (o múltiplos de 32, en el segundo caso) que estaban comprendidos entre -8.192 y 8.191 . Teníamos, por tanto, un total de 2^{11} (2^9 , en el segundo caso) posibles mensajes, de manera que, en principio, parecería natural asignar a cada uno de ellos una palabra código de 11 bits (9 bits). Los co-

dificadores entrópicos intentan determinar las asignaciones de las palabras código de una manera más eficiente que mediante esta estrategia tan directa.

La idea general de un codificador entrópico es intentar aprovechar la estadística con la que se producen los mensajes. Los codificadores entrópicos más utilizados son los **códigos de longitud variable** en los que el número de bits de una palabra código depende de las probabilidades con las que se producen los mensajes. La idea básica es la de asignar palabras de **corta longitud** (pocos bits) a aquellos mensajes que tienen una **mayor probabilidad**, y palabras de mayor longitud a los que se producen con menor frecuencia. Este argumento parece muy intuitivo, ya que estamos asignando pocos bits a aquellos mensajes que ocurren muy a menudo y que, por lo tanto, es de esperar que aporten poca información. En cambio, los mensajes que se producen con muy poca frecuencia tendrán asignadas palabras código con un número mayor de bits, lo que se corresponde con una mayor información.

En general, podemos afirmar que la estadística de los mensajes de una fuente se modifica cuando se aplican transformaciones a las señales, lo que nos sugiere concentrar los esfuerzos en buscar aquellas transformaciones que permitan obtener una estadística de mensajes que pueda codificarse con un número reducido de bits.

En los apartados siguientes estudiaremos con detalle la relación entre la información aportada por un mensaje y su probabilidad. También estableceremos el límite teórico de bits con el que puede codificarse una fuente digital y veremos algunas estrategias para la construcción de códigos de longitud variable.

Las tarifas telefónicas y los códigos de longitud variable

Consideremos un sencillo ejemplo sobre tarifas telefónicas que nos servirá para ilustrar de forma intuitiva las ideas básicas de los códigos de longitud variable.

La compañía telefónica X ofrece una tarifa plana para poder llamar a 256 números con coste reducido de 0,20 €/min. La compañía telefónica Y contrataca con otra oferta para llamar a 256 números a un coste de 0,35 €/min, pero permitiendo seleccionar un total de 4 números con un coste de 0,10 €/min.

Para saber qué compañía nos conviene más, necesitamos hacer un estudio de nuestras “estadísticas” de llamada. Podremos comprobar que si tenemos 256 amigos a los que llamamos con la misma frecuencia resulta más rentable la tarifa de la compañía X. En cambio, si nuestras llamadas se centran, en un 90%, en el novio/a, el amigo y la amiga íntima y en nuestra madre resultará más rentable utilizar la tarifa de la compañía Y.

Problema 1

Determinad el coste de realizar 100 llamadas de 1 minuto con las dos tarifas anteriores, sabiendo que nuestra “estadística” de llamadas se centra en un 90% en 3 números y que el resto de números es equiprobable.

Solución

El coste si utilizamos la compañía X es:

$$100 \text{ min} \times 0,2 \text{ €/min} = 20 \text{ €}$$

En cambio, con la compañía Y, nuestro coste será:

$$90 \text{ min} \times 0,1 \text{ €/min} + 10 \text{ min} \times 0,35 \text{ €/min} = 12,5 \text{ €/min.}$$

En los códigos de longitud variable...

... (VLC, *Variable Length Codes*) los mensajes con mucha probabilidad tienen asignadas palabras código de poca longitud, mientras que los mensajes con una probabilidad baja tienen palabras código de mucha longitud.

Problema 2

Determinad cuál es el porcentaje mínimo de llamadas a números seleccionados a partir del cual resulta rentable la tarifa de la compañía Y.

Solución

Para determinar el porcentaje crítico debemos hallar el valor en el que las dos tarifas tendrán el mismo coste:

$$A \text{ min} \times 0,2 \text{ €/min} = A \text{ min} \times P \times 0,1 \text{ €/min} + A \text{ min} \times (1 - P) \times 0,35 \text{ €/min}$$

Despejando el valor de P obtenemos:

$$P = 0,6 \text{ (60 \%)}$$

5. Medida de la información

En este apartado nos centraremos en los conceptos esenciales de la teoría de la información: la medida de la información asociada a un mensaje y la entropía asociada a la fuente. Dichos conceptos fueron desarrollados por Claude E. Shannon.

Claude E. Shannon (1916-2001) fue un gran matemático e ingeniero que debe considerarse como el padre de la teoría de la información. La pasión por los ingenios mecánicos y la ciencia se la inculcó su abuelo, un granjero que en su tiempo libre construía máquinas para la agricultura y que fue el inventor de la lavadora. Desde muy joven Shannon mostró una excelente aptitud para las matemáticas y para la ingeniería, graduándose en ambas disciplinas en la Universidad de Michigan y ampliando posteriormente sus estudios en el MIT (*Massachusetts Institute of Technology*), donde colaboró con Norbert Wiener y Vannevar Bush. Este último era el inventor del analizador diferencial, un aparato calculador analógico que permitía resolver ecuaciones complejas mediante la acción de un elevado número de elementos mecánicos que incluían muelles, engranajes, ruedas dentadas, etc. La solución de una ecuación se encontraba midiendo los cambios de posición que habían experimentado las distintas partes de la máquina. El proceso de “programación” de la máquina para afrontar un nuevo problema requería desmontar y volver a montar varias piezas según una configuración mecánica adecuada, tarea que podía durar varios días. Shannon financió sus estudios como “programador” del analizador diferencial y su función principal era prepararlo para que otros científicos pudieran resolver sus ecuaciones.

En paralelo realizó sus estudios centrándose en el análisis de los relés del analizador diferencial. Este estudio fue el eje principal de su tesis de máster, donde aplica por primera vez el álgebra de Boole a las máquinas computadoras, identificando la máquina de computación con la lógica de Boole. El trabajo concluye con el artículo *A Symbolic Analysis of Relay and Switching Circuits* (1938), que habría de tener una gran influencia en el posterior desarrollo de sistemas y máquinas de procesamiento de información. Su tesis doctoral la realizó aplicando conceptos de álgebra en genética teórica.

Trabajó durante muchos años en los Laboratorios Bell, donde colaboró con otros matemáticos y científicos en aspectos clave de las comunicaciones, mejorando la eficiencia y la fiabilidad de las transmisiones telefónicas y telegráficas a larga distancia. Sus trabajos sobre criptología constituyeron la base del que se considera como su artículo más importante: *A Mathematical Theory of Communications* (1945) en el que se introducen los conceptos de *entropía*, *medida de la información*, *capacidad de un canal*, *distorsión* y *códigos redundantes* para la protección frente a errores.

En 1950 publica otro de sus artículos clásicos: *Programming a Computer for Playing Chess* que fue, durante muchos años, una referencia en la programación de autómatas y ordenadores para el juego del ajedrez.

Su pasión por la ingeniería mecánica y la computación le mantuvo ocupado durante gran parte de su vida en el diseño de máquinas que él mismo denominaba “inútiles”. Por poner sólo unos ejemplos, construyó una máquina calculadora en números romanos, unas tortugas autómatas que caminaban solas por su casa, una máquina que realizaba malabarismos con tres pelotas, una máquina que permitía resolver el cubo de Rubik, un roncito automático que ‘aprendía’ técnicas para salir de laberintos, etc. Su gran pasión fue el monociclo, en el cual pasaba muchas horas practicando malabarismos.



Figura 8. Claude E. Shannon.

5.1. Información asociada a un mensaje

Una fuente de mensajes se caracteriza por el número de mensajes que puede producir y por la probabilidad con la que se produce cada mensaje. En la figura 9 se representa la idea básica de la caracterización de una fuente de mensajes en la que se definen los diferentes mensajes m_k y sus probabilidades asociadas

p_k . En la práctica, es difícil conocer *a priori* la probabilidad con la que se produce cada uno de los mensajes. No obstante, siempre puede realizarse un estudio en el que se determinen las frecuencias de aparición de cada mensaje y estimar su probabilidad a partir de estos resultados.

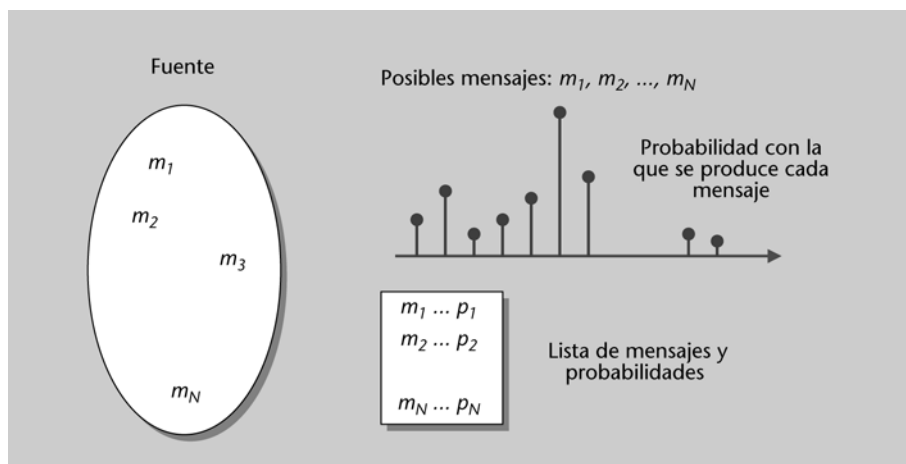


Figura 9. Caracterización de una fuente de mensajes.

Una vez especificada o estimada la estadística de la fuente, deberemos plantearnos establecer una medida cuantitativa de la cantidad de información asociada a cada mensaje. La medida debe verificar ciertos requisitos que parecen naturales y que generalmente están asociados al concepto cualitativo de información. Estos postulados son:

1) **La información asociada a un mensaje es positiva.** Es lógico suponer que siempre que se produce uno de los posibles mensajes, éste aporta alguna clase de información. No tiene sentido que se produzcan pérdidas de información si recibimos nuevos datos.

2) **Los mensajes con poca probabilidad aportan más información que los mensajes con mucha probabilidad.** Los mensajes que se producen frecuentemente nos aportan muy poca información porque son esperados de antemano. Así, cuando se nos informa de que hay circulación lenta en los accesos de entrada a Barcelona entre las 8 y las 9 de la mañana, nos encontramos ante un mensaje que nos aporta muy poca información. En cambio, la noticia de una gran nevada en Barcelona aparecerá siempre con grandes titulares en los periódicos, ya que, debido a su baja probabilidad, nos aportará mucha información.

3) **La información que aportan dos mensajes simultáneos estadísticamente independientes es la misma que la que aportan cuando se producen por separado.** Esto significa que la información que nos aportan dos eventos independientes que se producen simultáneamente coincide con la suma de las informaciones que nos aportarían si se produjeran por separado.

Un ejemplo de fuente de mensajes...

... podría ser una imagen en blanco y negro. Los posibles mensajes que genera esta fuente son el nivel de gris de los píxeles. Tendremos por tanto, en el caso más habitual, un total de 256 posibles mensajes. Podemos estimar la probabilidad de cada mensaje determinando el porcentaje de píxeles que tienen un determinado nivel respecto al total de píxeles de la imagen.

Podemos formular estos postulados mediante las siguientes ecuaciones:

$$I(m_k) \geq 0 \quad (1)$$

$$I(m_i) > I(m_j) \quad \text{si} \quad p_i < p_j \quad (2)$$

$$I(m_i, m_j) = I(m_i) + I(m_j) \quad \text{si} \quad P(m_i, m_j) = P(m_i) \cdot P(m_j) \quad (3)$$

En virtud del teorema fundamental del cálculo, únicamente existe una función que verifique las tres propiedades simultáneamente. De este modo, la medida cuantitativa de la información es totalmente axiomática pues se encuentra directamente a partir de las propiedades que queremos que verifique.

Así pues, la información asociada al mensaje m_k vendrá determinada por la única función que verifica las tres propiedades anteriores:

$$I(m_k) = -\log_b p_k \quad (4)$$

Es fácil comprobar que la definición anterior verifica las propiedades para cualquier base b del logaritmo. Cuando la base del logaritmo es 2, la unidad de información se denomina **bits**. Si la base del logaritmo es el número de Euler e , la unidad de información se mide en **nats**.

Consideremos como ejemplo la información que proporciona una moneda al caer. Existen dos posibles mensajes (cara y cruz), los dos con la misma probabilidad ($1/2$). La información asociada al mensaje cara será:

$$I(\text{cara}) = -\log_2(1/2) = 1 \text{ bit}$$

Análogamente, si consideramos una fuente que puede transmitir 8 mensajes equiprobables, la información asociada a cada uno de estos mensajes viene dada por:

$$I(m_i) = -\log_2(1/8) = \log_2 2^3 = 3 \text{ bit}$$

De acuerdo con estos resultados, para estimar la cantidad de información ligada a uno de los mensajes de la fuente es necesario establecer en primer lugar las características estadísticas de la fuente de mensajes. Estas características pueden estimarse de forma experimental dejando que la fuente genere mensajes y aproximando su probabilidad por la frecuencia relativa de los mismos. En general, si la estimación se realiza sobre una muestra suficientemente alta, este procedimiento suele proporcionar resultados satisfactorios.

5.2. Entropía de una fuente de mensajes

Conociendo la información asociada a cada mensaje podemos determinar la información media que proporciona la fuente. Para ello debemos ponderar la información de cada mensaje por la probabilidad de que se produzca:

$$H = \sum_{k=1}^N p_k \cdot I(m_k) = -\sum_{k=1}^N p_k \cdot \log_2 p_k. \quad (5)$$

Esta función se conoce con el nombre de **entropía** y establece un límite al número de bits medio con el que podrán codificarse los mensajes de una fuente. Ningún codificador podrá obtener códigos cuyo número de bits medio sea inferior a la entropía de la fuente. El nombre de *entropía* de la función anterior se debe a que su fórmula es muy parecida a la entropía termodinámica de L. Boltzman. Recientemente, algunas teorías físicas han relacionado las dos entropías (J. D. Bekenstein).

Mensajes equiprobables y mensajes con probabilidades dispares

Comprobaremos mediante un ejemplo que la entropía de una fuente disminuye cuando uno (o unos cuantos mensajes) tienen una probabilidad de ocurrencia mucho mayor que los demás. Este resultado puede generalizarse (véase problema 3) para demostrar que la entropía de una fuente es máxima cuando todos los mensajes que produce son equiprobables.

Supongamos dos fuentes F_1 y F_2 que pueden producir un total de 256 mensajes cada una de ellas. En la fuente F_1 todos los mensajes son equiprobables ($p_k = 1/256$) de manera que la información asociada a cada mensaje es de 8 bits y la entropía de la fuente será también de 8 bits.

$$H(F_1) = \sum_{n=1}^{256} \frac{1}{256} \cdot \log_2 \left(\frac{1}{256} \right) = 8 \text{ bits}.$$

La fuente F_2 tiene un mensaje con una probabilidad muy alta y el resto de mensajes tienen la misma probabilidad:

$$\begin{aligned} m_1 &\rightarrow p(m_1) = p_1 \\ m_k &\rightarrow p(m_k) = \frac{(1-p_1)}{255} \quad k = 2, \dots, 256 \end{aligned}$$

En este caso la entropía viene dada por:

$$H(F_2) = -p_1 \cdot \log_2 p_1 - (1-p_1) \cdot \log_2 \left(\frac{1-p_1}{255} \right). \quad (6)$$

En el gráfico de la figura 10 se muestra el valor de la entropía de F_2 en función del valor de la probabilidad p_1 . Podemos observar que a medida que la probabilidad p_1 se acerca a la unidad, la entropía de la fuente disminuye. Esto coincide con la idea intuitiva de que, a medida que p_1 aumenta, el resultado que proporcionará la fuente es más previsible (el mensaje m_1 tiene una probabilidad de ocurrencia mayor que el resto) con lo que la información media que proporciona es menor. En el caso límite en que la probabilidad p_1 sea la unidad, la fuente no proporciona ninguna información. Otro aspecto que resulta interesante observar es que el valor máximo de la entropía se produce cuando $p_1 = 1/256$. En efecto, aunque en la figura 10 pueda dar la sensación de que el máximo de la entropía está en el origen, sustituyendo en la ecuación (6) $p_1 = 0$ y $p_1 = 1/256$ obtenemos, respectivamente 7,9943 bits y 8 bits. Este resultado también parece intuitivo, ya que cuando todos los símbolos son equiprobables, nuestra incertidumbre sobre cuál será el mensaje que producirá una fuente es máxima; de este modo es natural que la información media proporcionada por la fuente también tome su valor máximo. Demostraremos este resultado matemáticamente en el problema 3.

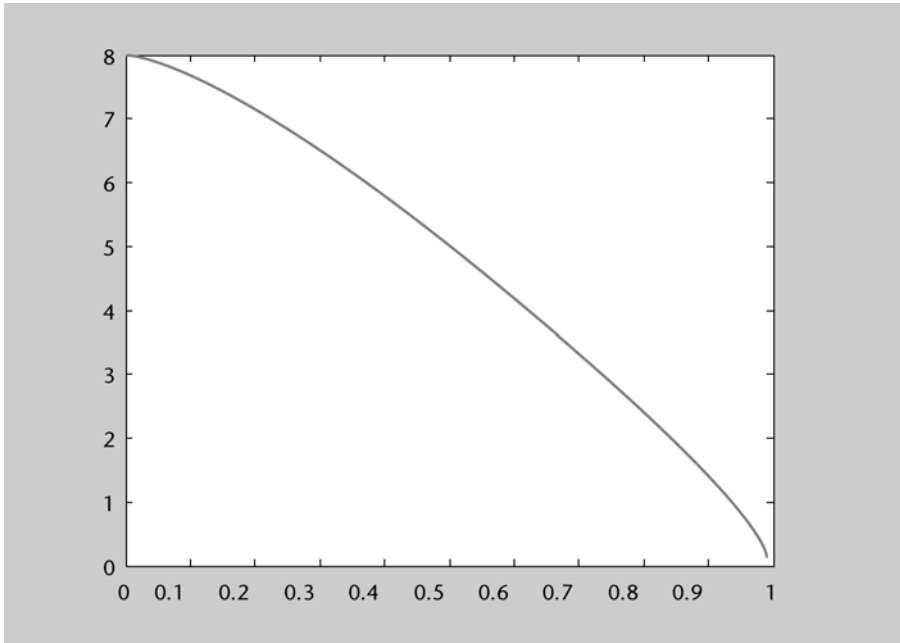


Figura 10. Evolución de la entropía con la probabilidad de uno de los mensajes (resto de mensajes equiprobables).

Problema 3

Demostred que para una fuente de M mensajes la entropía es máxima cuando todos los mensajes son equiprobables.

Solución

Cuando los mensajes son equiprobables la entropía viene dada por:

$$H = \sum_{i=1}^M \frac{1}{M} \cdot \log_2(M) = \log_2(M) .$$

Por lo tanto, debemos demostrar que en un caso general:

$$H(F_M) \leq \log_2 M$$

donde F_M indica una fuente genérica de M mensajes.

Consideremos la siguiente diferencia:

$$H(F_M) - \log_2(M) = \sum_{i=1}^M p_i \cdot \log_2\left(\frac{1}{p_i}\right) - \sum_{i=1}^M p_i \cdot \log_2(M) = \sum_{i=1}^M p_i \cdot \log_2\left(\frac{1}{Mp_i}\right)$$

y utilicemos la desigualdad:

$$\ln y \leq y - 1 \Rightarrow \log_2 y \leq (y - 1) \cdot \log_2 e$$

con lo que finalmente obtenemos:

$$H(F_M) - \log_2(M) \leq \log_2 e \left(\sum_{i=1}^M p_i \left(\frac{1}{Mp_i} - 1 \right) \right) = \log_2 e \cdot \left[\sum_{i=1}^M \frac{1}{M} - \sum_{i=1}^M p_i \right] = 0$$

lo que demuestra la proposición.

6. Códigos de longitud variable. Conceptos básicos

Tal como hemos introducido anteriormente, la idea básica de los códigos de longitud variable es asignar palabras código de longitudes distintas en función de la probabilidad de los mensajes. Los mensajes más probables se codificarán con palabras con un menor número de bits que los mensajes menos probables.

Considerar como ejemplo los mensajes de la fuente que se resume en la tabla 2 y en la que los códigos asignados a cada mensaje han sido elegidos para que el código cumpla una serie de restricciones que detallaremos posteriormente.

mensajes	probabilidad	código
m_1	0,4	0
m_2	0,3	10
m_3	0,1	1100
m_4	0,1	1101
m_5	0,1	1110

Tabla 2. Ejemplo de códigos de longitud variable.

Podemos calcular el número medio de bits que utiliza el código propuesto ponderando el número de bits que se utiliza para cada mensaje con la probabilidad de que se produzca el mensaje.

$$N_{\text{medio}} = \sum_{k=1}^N p_k \cdot N_k = 1 \cdot 0,4 + 2 \cdot 0,3 + 4 \cdot 0,3 = 2,2 \text{ bits}$$

El número de bits medio debe ser superior a la entropía. En efecto, la entropía representa la información media de la fuente, por lo que cualquier código práctico que utilicemos tendrá un número de bits medio mayor o igual que la entropía. En nuestro caso:

$$H = -\sum_{k=1}^N p_k \cdot \log_2 p_k = 2,04 \text{ bits}$$

Problema 4

Determinad, utilizando la tabla 2, la secuencia de bits con la que quedaría codificada la siguiente secuencia de mensajes: $m_3 m_5 m_1 m_1 m_3 m_2 m_1 m_1$.

Solución

Para obtener la secuencia de bits simplemente debemos sustituir cada mensaje por su código correspondiente. En nuestro ejemplo obtenemos:

1100 1110 0 0 1100 10 0 0

Problema 5

Supongamos una fuente con 5 mensajes que ha sido codificada mediante la tabla 2. Determinad la secuencia de mensajes a la que corresponde la siguiente secuencia de bits:

010011010110010

Solución

La forma más usual para determinar la secuencia de mensajes consiste en ir descodificando paquetes de bits una vez se encuentran en la tabla. Si un paquete de bits no se encuentra en la tabla, se aumenta el paquete tomando un bit adicional. Así:

Paquete 1:	0	Corresponde al mensaje m_1 .
Paquete 2:	1	No corresponde a ningún mensaje, aumentamos el paquete con el siguiente bit.
Paquete 2:	10	Corresponde al mensaje m_2 .
Paquete 3:	0	Mensaje m_1 .
Paquete 4:	1	No corresponde a ningún mensaje.
Paquete 4:	11	No corresponde a ningún mensaje.
...		
Paquete 4:	1101	Corresponde al mensaje m_4 .

Utilizando este procedimiento la secuencia de mensajes final es:

$$m_1 m_2 m_1 m_4 m_1 m_3 m_2.$$

El código que hemos utilizado en estos ejemplos ha sido propuesto de antemano. Hemos visto que su número de bits medio está muy próximo a la entropía y que es sencillo obtener secuencias de bits a partir de una secuencia de mensajes y viceversa. Sin embargo, deberíamos ser conscientes de que no todos los códigos tendrán estas dos propiedades y que deberíamos encontrar un procedimiento sistemático que nos permitiera generar este tipo de códigos. En el apartado siguiente examinaremos con detalle la construcción y algunas propiedades de los códigos de Huffman. Dichos códigos y sus variantes son utilizados en la mayor parte de compresores de señales de audio y vídeo (MPEG-1, MPEG-2, MPEG-4, MP3, etc.).

7. Códigos de Huffman

En este apartado estudiaremos los códigos de Huffman, que se han convertido en una de las técnicas más utilizadas en las tecnologías de la comunicación digital. Estos códigos deben su nombre a David A. Huffman.

David A. Huffman (1925-1999) ha sido uno de los científicos que mayor influencia ha tenido en las tecnologías de la información, aportando contribuciones significativas en distintos campos como la teoría de máquinas de estado finito, las metodologías de diseño de circuitos lógicos, el diseño de señales para radar y, sobre todo, por su aportación a la compresión de señales mediante los códigos de longitud variable que llevan su nombre. Actualmente, los códigos de Huffman o algunas de sus múltiples variantes se utilizan en prácticamente todas las aplicaciones que requieren la compresión y transmisión de datos digitales: módems, fax, redes de ordenadores, archivos de audio MP3, archivos de vídeo (MPEG-2, DIVX), etc.

En 1951 David A. Huffman era un estudiante de la asignatura de Teoría de la información impartida por el profesor Robert M. Fano en el MIT (*Massachusetts Institute of Technology*). Durante ese curso, el profesor Fano ofreció a sus estudiantes la posibilidad de realizar un trabajo con el que podían quedar exentos de presentarse al examen final de curso. El trabajo consistía en determinar los códigos más eficientes para representar números, letras u otros símbolos (mensajes) utilizando códigos binarios. Lo que el profesor Fano no mencionó era que tanto Claude E. Shannon como él mismo llevaban varios años colaborando e intentando obtener un método óptimo para la obtención de estos códigos binarios sin que por el momento hubieran tenido éxito alguno. Huffman trabajó en el problema durante varios meses, desarrollando distintas alternativas para la obtención de códigos sistemáticos sin poder llegar a demostrar que alguna de las soluciones propuestas fuese óptima. Finalmente, desanimado, y cuando ya estaba empezando a estudiar para presentarse al examen final, dio con la demostración definitiva.

Actualmente, los códigos de Huffman constituyen una de esas pocas ideas fundamentales que se utilizan en prácticamente todas las tecnologías de comunicación de datos digitales. Otra de las contribuciones especialmente importantes de David A. Huffman, y de la que él se sentía particularmente orgulloso, fue la primera técnica de metodología formal para el diseño de circuitos lógicos secuenciales que desarrolló en su tesis doctoral.

Aunque los códigos de Huffman han ayudado a muchos inversores y tecnólogos a hacerse millonarios, el único provecho que Huffman obtuvo por los códigos fue la dispensa al examen final de Teoría de la información. Huffman nunca trató de obtener una patente por alguna de sus contribuciones, aunque es evidente lo sustanciosas que le hubieran resultado estas patentes, especialmente la de los códigos. Finalmente, mencionar que Huffman fue discípulo directo de Shannon, ya que este último pasó varias horas enseñándole a montar en monociclo, hasta que finalmente también se convirtió en un experto.



Figura 11. David A. Huffman.

7.1. Obtención sistemática de los códigos de Huffman

Los códigos de Huffman (1951) representan hoy en día la técnica más utilizada para la eliminación de la redundancia estadística en la codificación. El procedimiento para obtener estos códigos es conceptualmente simple, aunque, como veremos, desde el punto de vista computacional resulta complejo. Es posible demostrar que los códigos de Huffman son óptimos en el sentido de que consiguen el número más pequeño de símbolos de código por símbolo de fuente con la restricción de que los mensajes deben codificarse uno a uno (cada mensaje tiene asociada una única palabra código).

También puede demostrarse que el número de bits medio que se consigue obtener con un código de Huffman está siempre comprendido entre la entropía de la fuente y la entropía más uno:

$$H < \bar{N}_{\text{Huffman}} < H + 1 \quad (7)$$

El procedimiento para obtener un código de Huffman puede resumirse en las siguientes etapas:

- 1) Ordenar los mensajes por probabilidades decrecientes.
- 2) Crear un árbol combinando siempre los dos nodos con menor probabilidad. El resultado de la combinación de los dos nodos es un nuevo nodo cuya probabilidad es la suma de las probabilidades de los nodos que lo han construido. Continuar con este procedimiento hasta obtener un único nodo.
- 3) Para asignar los códigos a los mensajes, basta con recorrer el árbol desde la raíz hasta cada uno de los mensajes, asignando un 0 o un 1 en función de si tomamos una rama hacia arriba o hacia abajo (la asignación de ceros y unos a las ramas puede ser arbitraria).

El procedimiento anterior resulta bastante más claro si se ilustra con un ejemplo sencillo como el que se muestra en la figura 12. Podemos observar cómo los nodos van combinándose de dos en dos, eligiendo siempre los de probabilidad más pequeña. Si existen varios nodos con la misma probabilidad, pueden combinarse dos cualesquiera. La probabilidad asignada al nodo raíz deberá ser siempre la unidad.

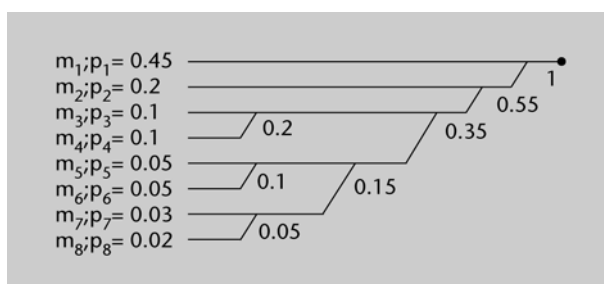


Figura 12. Proceso de creación del código de Huffman.

En la figura 13 se muestra el mismo árbol pero ahora se utiliza para la asignación de códigos a los mensajes. Así, para determinar el código que asignamos al mensaje m_4 deberemos partir del nodo raíz, realizar dos pasos hacia abajo y después dos pasos más hacia las ramas superiores. El código resultante para este mensaje es 1100. Obsérvese que, si seguimos esta construcción del árbol de Huffman, el mensaje con menor probabilidad siempre queda codificado con todos los símbolos igual a la unidad.

Una vez determinado el código de Huffman para esta fuente, podemos calcular el número medio de bits que se utilizan obteniendo un total de 2,4 bits. Si calculamos la entropía de la fuente original obtendremos 2,344 bits, lo que da una idea de la calidad del código.

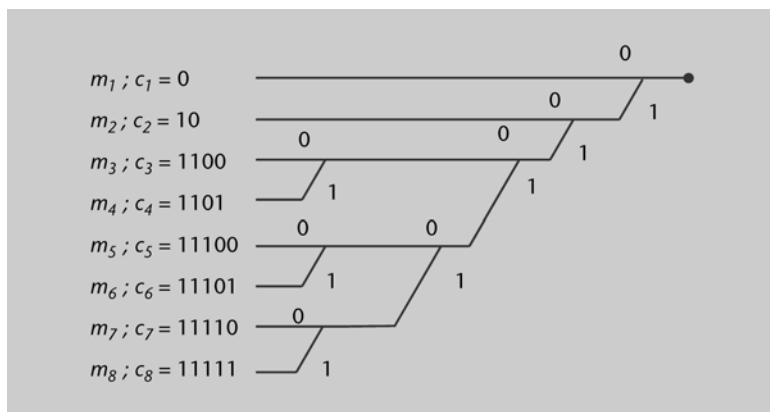


Figura 13. Proceso de generación del código a partir del árbol de Huffman.

Una de las características de los códigos de Huffman es que permiten descodificar la secuencia de datos a partir de la concatenación directa de los códigos y sin necesidad de “señalizar” el principio y el final de un mensaje. Esto puede parecer trivial, ya que los códigos son de longitud variable, por lo que podría no resultar evidente determinar cuándo empieza y cuándo acaba cada código. No obstante, la descodificación es posible y además es sencilla debido a que, por construcción del árbol de Huffman, ninguno de los códigos largos tiene como subcadena una palabra código más corta. Todos los códigos que cumplen que una palabra código nunca es el prefijo de otra palabra código pueden descodificarse sin ambigüedad. Estos códigos se denominan de prefijo y puede demostrarse que todos ellos cumplen la ecuación (7). Los códigos de Huffman son los códigos de prefijo que tienen un número de bits medio más reducido.

Para ilustrar el procedimiento de descodificación consideremos la cadena de bits 111011101001011111..., cuyo proceso se representa esquemáticamente en la figura 14. La descodificación empezaría comprobando si el primer bit (1) es una palabra código, buscándolo en la tabla de palabras código con un único bit. Como no es así, comprobaríamos si los dos primeros bits (11) pueden ser una palabra código. Continuaríamos con este procedimiento hasta que encontrásemos la primera palabra código 11101, que sí que se encuentra en la tabla, y que corresponde al mensaje m_6 . Procediendo de este modo, llegaríamos a descodificar la secuencia de mensajes completa que se corresponde con los códigos c_6, c_4, c_1, c_1, c_2 y c_8 .

Uno de los aspectos más críticos en los códigos de Huffman y, en general, en los códigos de longitud variable, es la aparición de errores en el canal. En efecto, cuando se produce algún error de bit en el canal, puede producirse un cambio de palabra código que supone la pérdida del sincronismo y la posibilidad de que aparezca un gran número de errores consecutivos. Algunas veces, el receptor puede detectar de forma automática la aparición de los errores debido

a que no encuentra las palabras código en las tablas (o que la estadística de aparición no se corresponde con la esperada). Aunque existen algoritmos para que el descodificador pueda resincronizarse con la cadena de datos después de un error, estos algoritmos son complejos y su coste computacional muy elevado, por lo que se usan muy poco en aplicaciones prácticas.

TABLA	Secuencia de bits
$c_1 = 0$	111011101001011111...
$c_2 = 10$	1 1 1 0 1 → $c_6 = 11101$
$c_3 = 1100$	111011101001011111...
$c_4 = 1101$	1 1 0 1 → $c_4 = 1101$
$c_5 = 11100$	111011101001011111...
$c_6 = 11101$	
$c_7 = 11110$	0 → $c_7 = 0$
$c_8 = 11111$	

Figura 14. Ejemplo de descodificación de una cadena de mensajes.

La solución más habitual para protegerse frente a los posibles errores de bits en el canal es introducir **palabras de resincronización** con una determinada frecuencia (palabra única, cuyo valor y longitud es conocida por el descodificador y que sirve para buscar un punto de resincronización cuando se han producido errores). Las palabras de resincronización permiten utilizar algoritmos muy simples de descodificación: detectar pérdida de sincronismo (errores) y buscar la palabra única que nos permitirá volver a sincronizarnos. La frecuencia con la que se introduce la palabra de resincronización depende de la probabilidad con la que se produzcan errores. Si los errores son muy probables, conviene introducirla con mayor frecuencia. No obstante, es evidente que introducir la palabra de *resincronización* con mucha frecuencia reduce la eficiencia del compresor. Los estándares de compresión MPEG para señales de audio y vídeo utilizan esta técnica para recuperar el sincronismo del descodificador si se han producido errores. La frecuencia con la que se inserta la palabra de resincronización en MPEG depende de la aplicación. Así, en difusión vía satélite la palabra se inserta muchas más veces que en difusión por cable, ya que en este último caso las señales están más protegidas frente a errores e interferencias.

La obtención del código de Huffman una vez se dispone de la estadística de la señal es conceptualmente muy sencilla. No obstante, los algoritmos asociados no son triviales y requieren realizar bastantes simplificaciones y comparaciones en la tabla de mensajes. Otro de los problemas importantes es la estimación de las estadísticas con las que se producen los mensajes. Las estimaciones de los mensajes menos probables son poco fiables y se requiere trabajar con muestras muy grandes para obtenerlos. A menudo, se obtiene que la probabilidad estimada de algunos mensajes es nula, ya que estos mensajes no se han producido ninguna vez en la secuencia que se ha utilizado para determinar las estadísticas. Aunque el procedimiento de Huffman

puede aplicarse de la misma forma en estos casos, es posible que los valores de la tasa de bits media con que se codifica la fuente no sean tan óptimos como cabría esperar. Existen varias alternativas para simplificar el diseño de la tabla de codificación de Huffman. Todas ellas son soluciones sub-óptimas que simplifican el análisis de la fuente y la construcción de la tabla de codificación. Un ejemplo de estas soluciones subóptimas son los códigos de Huffman truncados que expondremos posteriormente.

El estudiante puede encontrar descritos los códigos de Huffman truncados en el subapartado 7.2 de este mismo módulo.

Unos códigos de longitud variable especialmente interesantes son los **códigos reversibles** (*reversible variable length codes*), que se utilizan en codificación de vídeo en el estándar MPEG-4. Estos códigos tienen la propiedad de que pueden ser descodificados cuando se leen en el sentido inverso al habitual (*backward*). De esta forma, cuando se produce un error en el canal, debemos esperar hasta la próxima palabra de *resincronización* y realizar la lectura en sentido inverso, con lo que seremos capaces de descodificar parte del mensaje. Los códigos reversibles son menos eficientes que los códigos de Huffman convencionales en cuanto al número medio de bits con el que codifican la fuente. En la figura 15 se muestra gráficamente la idea de los códigos reversibles.

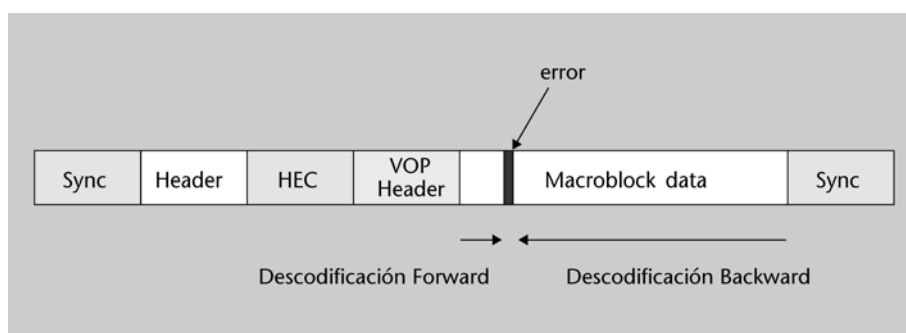


Figura 15. Códigos de longitud variable reversibles.

En aplicaciones de almacenamiento de datos en ficheros, es recomendable que la codificación y la decodificación se realicen empleando *buffers* temporales de memoria con una longitud que sea un múltiplo de 8 bits. Con ello, cuando el *buffer* está lleno, se pueden almacenar la secuencia de bits, aprovechando una palabra de un byte o múltiplo del byte.

7.2. Códigos de Huffman truncados

Una de las variantes más comunes del método de Huffman son los códigos truncados, que se basan en descartar los mensajes menos probables del proceso de análisis del código de Huffman. Estos mensajes se codifican posteriormente utilizando palabras de longitud fija precedidas de un prefijo proporcionado por el mismo análisis de la fuente simplificada. Las etapas de análisis del código se resumen en los siguientes puntos:

1) Se ordenan los N mensajes de la fuente por orden de probabilidad decreciente.

2) Se eligen los M mensajes menos probables y se genera un nuevo mensaje L que representa los $(N - M)$ mensajes descartados. La probabilidad de L es la suma de las probabilidades de los mensajes con baja probabilidad que representa.

3) Se construye el código de Huffman asociado a la nueva fuente de $M + 1$ mensajes.

4) Se construye el código de los $N - M$ mensajes de menor probabilidad, empleando como prefijo el código de Huffman asignado al nuevo mensaje L , seguido de un código binario convencional de longitud constante.

El procedimiento se resume esquemáticamente en la figura 16, donde hemos supuesto que se descartan los 5 mensajes con menor probabilidad. La suma de las probabilidades de estos mensajes se inserta en la tabla de mensajes previamente ordenados (en nuestro ejemplo, es el 5.º mensaje más probable) y se construye el código de Huffman con esta nueva estadística. El código resultante se utiliza como prefijo para asignar las palabras que representan los mensajes originales. En este ejemplo, debemos añadir tres bits adicionales para poder representar los 5 mensajes.

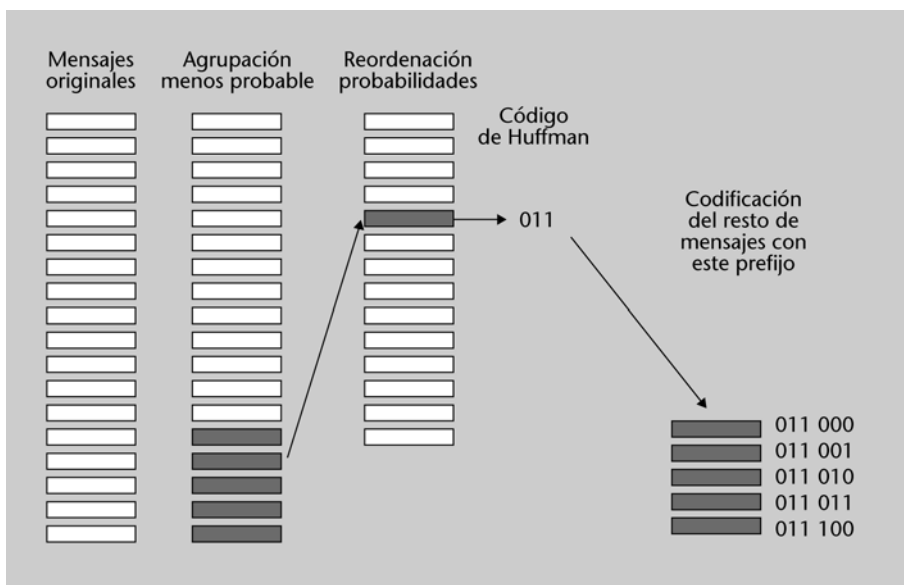


Figura 16. Construcción de los códigos de Huffman truncados.

8. Transformaciones de datos

Hemos visto que los códigos de longitud variable obtienen una reducción eficiente del número de bits medio de una fuente siempre que existan diferencias importantes entre las probabilidades con las que se producen los mensajes. Por ello, es evidente que uno de los aspectos más importantes en el diseño de cualquier compresor es utilizar transformaciones de la fuente de mensajes para que produzcan unos mensajes con mucha mayor probabilidad que otros. Generalmente, en la codificación de señales de audio y vídeo se utilizan varios tipos de transformaciones: diferencial, predicción, transformada coseno, etc., dependiendo de las características de las componentes que se codifican. Las transformaciones concretas dependen de la aplicación, incluso en algunos casos pueden combinarse varios métodos. Así, por ejemplo, en los casos en los que los terminales tengan restricciones de coste computacional, las transformaciones serán más simples y seguramente con una menor eficiencia desde el punto de vista de compresión. Para ilustrar la idea general consideraremos con cierto detalle la transformación diferencial y describiremos cualitativamente los métodos predictivos, concluyendo con algunos ejemplos muy simplificados de la codificación de voz en telefonía móvil GSM (*global system to mobile communication*).

El estudio detallado de las transformaciones concretas utilizadas en cada uno de los sistemas audiovisuales escapa a los propósitos de este módulo.

8.1. Codificación diferencial. Aplicación a la codificación de imagen

El esquema básico de un codificador diferencial se representa en la figura 17. Como ejemplo, la señal de entrada podría ser la luminancia de una línea de la imagen. La transformación utilizada consiste simplemente en calcular la diferencia entre dos muestras consecutivas. En principio, desde un punto de vista intuitivo, esta transformación parece natural debido a la redundancia que existe entre dos elementos consecutivos. Es de esperar que, una vez realizada la transformación, la mayoría de muestras se encuentren situadas en un entorno próximo al cero por lo que, seguramente, los mensajes con valores próximos a cero tendrán probabilidades mucho mayores que los otros mensajes.

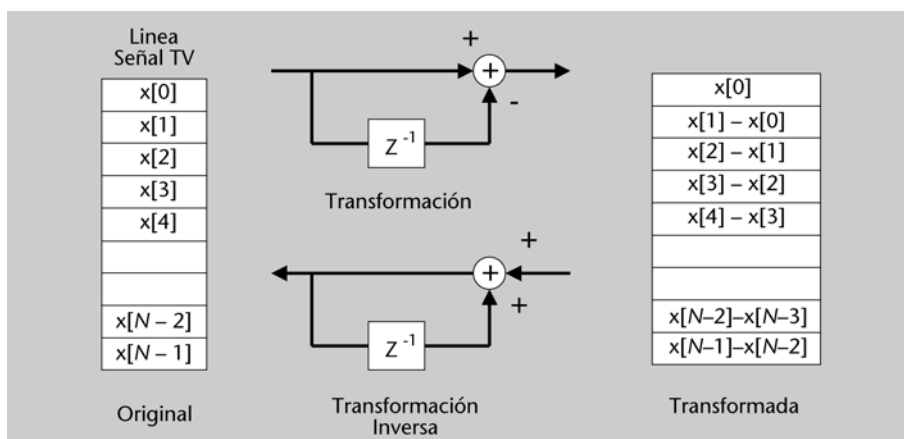


Figura 17. Ejemplo de transformación diferencial.

Podemos observar que la transformación que hemos utilizado es invertible y que se puede recuperar de forma exacta la información original a partir de los resultados transformados. Analíticamente, la transformación diferencial viene dada por la expresión:

$$y[n] = x[n] - x[n-1],$$

mientras que la transformación inversa se obtiene como:

$$z[n] = y[n] + z[n-1].$$

Puede verificarse que una transformación es inversa de la otra con sólo aplicar las ecuaciones anteriores a las tablas de valores que se muestran en la figura 17.

Los resultados de aplicar el codificador diferencial a una línea de la imagen se muestran en la figura 18, donde se representa el nivel de gris de una línea original y los valores resultantes después de la transformación. Podemos ver cómo los valores transformados se concentran alrededor del cero y que únicamente en los contornos de la imagen se producen valores significativos de la señal diferencia.

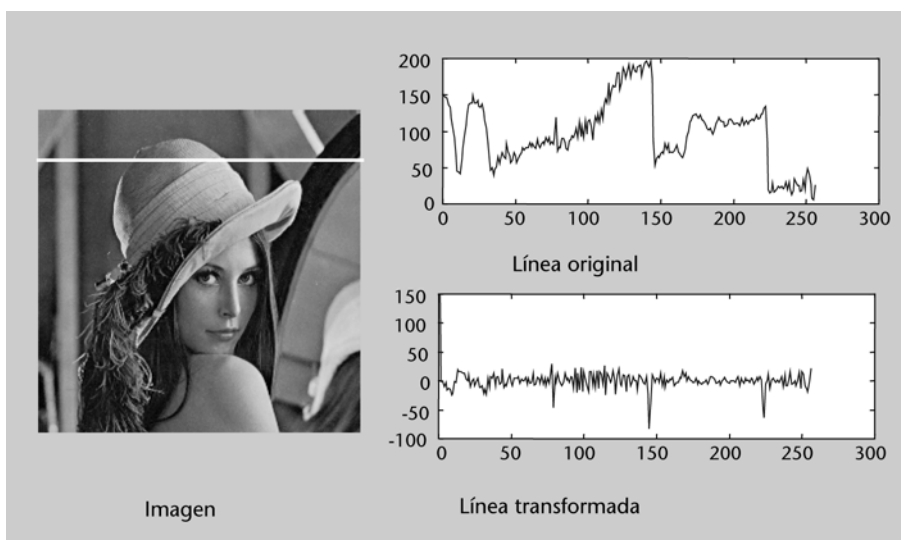


Figura 18. Representación de los niveles de una línea original y una transformada o diferencial.

En la figura 19 se representan los resultados del codificador diferencial aplicados a una imagen en blanco y negro. La imagen original está codificada con 256 niveles de gris (8 bits por píxel) y se representa asignando el negro al nivel 0 y el blanco al nivel 255. La imagen diferencial se ha representado tomando el nivel 0 como un gris neutro. Los valores negativos son más oscuros que el gris neutro mientras que los valores positivos son más claros. Esta imagen se ha obtenido aplicando el codificador diferencial a cada una de las líneas de la imagen. A partir de estos resultados visuales, es evidente que hemos conseguido el propósito que perseguíamos, es decir, la mayoría de píxeles de la imagen diferencial están próximos al nivel de gris medio y sólo en los contornos de la imagen aparecen píxeles cuyo valor se aparta apreciablemente del cero.



Figura 19. Imagen original e imagen diferencial.

Este efecto puede apreciarse más claramente en las gráficas de la figura 20, donde se representan los histogramas de la imagen original y de la imagen diferencial. El **histograma** es una representación bidimensional de la frecuencia relativa con la que aparece cada nivel de gris en la imagen. Así, para cada posible nivel de gris en el eje de abscisas encontramos el número de píxeles de la imagen que toman ese nivel de gris. Por tanto, el histograma es una representación aproximada de la probabilidad de que un píxel tome un determinado nivel de gris en la imagen. Nótese que en estas gráficas se han normalizado los niveles de gris de forma que el negro corresponde al cero mientras que el blanco corresponde a la unidad. En el histograma de la izquierda, que se corresponde con la imagen original, los niveles de gris se distribuyen dentro de toda la gama sin mostrar una preferencia directa por ningún valor. En cambio, en la gráfica de la derecha, que corresponde a la imagen diferencial, existe una gran concentración de elementos de imagen con un nivel de gris próximo a 0,5 (gris neutro con el que hemos representado el valor cero).

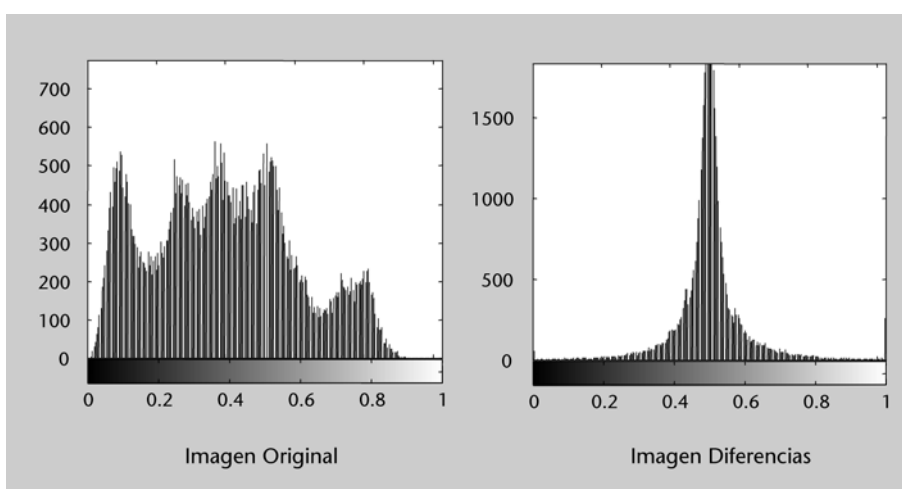


Figura 20. Histogramas de la imagen original y la imagen transformada o diferencial.

Cuando una señal tiene una estadística en la que predominan unos pocos valores, que se presentan con mucha probabilidad, frente a otros muchos posibles niveles que se presentan con poca probabilidad, es cuando resulta

rentable utilizar códigos de longitud variable en los que los valores más probables se codifican empleando pocos bits, mientras que los valores con poca probabilidad utilizan una gran cantidad de bits. Para tener una idea de la ganancia que hemos obtenido al aplicar la transformación diferencial, podemos comparar la entropía de la fuente original y de la fuente diferencial, suponiendo que las probabilidades de cada mensaje pueden aproximarse por los histogramas calculados anteriormente. En este caso obtenemos que la entropía de la imagen original es de 7,58 bits, mientras que la entropía de la imagen diferencial es de 5,8 bits.

Es importante remarcar que la codificación que hemos realizado hasta este momento es una codificación sin pérdidas, siempre que mantengamos los cálculos de la diferencia con la precisión adecuada. En este ejemplo, la imagen original estaba codificada con 256 niveles de gris (8 bits por muestra) por lo que las señales diferencia pueden tomar valores comprendidos entre -255 y 255, ya que es posible que un píxel tome el valor nulo cuando el anterior era blanco o viceversa. Esto significa que si queremos reconstruir la imagen original con los niveles de gris exactos (sin pérdidas), deberemos mantener los 511 posibles valores que puede tomar la señal diferencia (9 bits por muestra). Si estos valores se aproximan mediante una cuantificación más tosca, ya no será posible recuperar la imagen original de forma exacta.

8.2. Predicción lineal. Una generalización de la codificación diferencial

La codificación diferencial puede considerarse un caso particular de la predicción de muestras. Los **predictores lineales** son uno de los métodos más utilizados para la codificación de señales de audio y vídeo y se basan en estimar el valor de la muestra actual a partir de una combinación lineal de las muestras anteriores. La idea general es que combinando varias de las muestras anteriores podemos obtener una mejor estimación de la muestra actual. Por lo tanto, si mejoramos la estimación de la muestra, la señal diferencia tendrá una probabilidad muy alta para los mensajes próximos a cero y podrá ser codificada de forma muy eficiente mediante códigos de longitud variable.

La ecuación que rige el filtro predictor es:

$$\tilde{x}[n] = \sum_{k=1}^P a_k \cdot x[n-k]$$

donde P es el orden del predictor y a_k son los coeficientes de predicción. Los coeficientes de predicción pueden determinarse en base a las características estadísticas de la señal o ser establecidos de antemano entre el codificador y el decodificador.

El orden del predictor...

... es el número de muestras anteriores que se utilizan para realizar la predicción de la muestra actual.

La idea básica para codificar la señal es que tanto el transmisor como el receptor conozcan los coeficientes que se utilizarán en el predictor y utilicen ambos los mismos coeficientes en todo momento. Una posibilidad es que el transmisor calcule los coeficientes y los envíe al receptor. Otra alternativa es que ambos calculen los coeficientes, utilizando el mismo algoritmo y con las mismas muestras de señal, para tener siempre la garantía de que el predictor usado en el transmisor y el usado en el receptor sean el mismo. De esta forma, tanto el transmisor como el receptor utilizarán estos coeficientes para predecir la siguiente muestra a partir de las muestras anteriores. Esta predicción tiene un error, que será conocido por el transmisor y que lo enviará al receptor para que éste pueda determinar el valor exacto de la muestra actual. Así pues, la señal que se transmite es el error de predicción que esperamos que, con elevada probabilidad, tome valores próximos a cero.

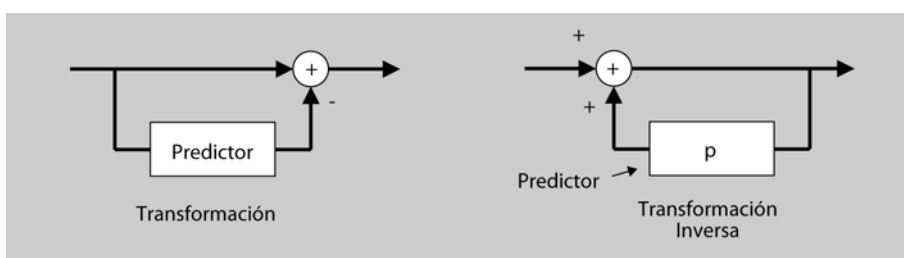


Figura 21. Esquema genérico de un predictor y de la transformación inversa.

La codificación diferencial...

... puede considerarse un caso especial de un predictor que utiliza un único coeficiente (la muestra anterior) con valor unidad.

8.3. Codificación diferencial y predicción de una señal de audio

Consideraremos un ejemplo utilizando un fragmento de señal de audio para ilustrar el efecto de las transformaciones de los datos sobre la entropía. En la figura 22 se muestra la forma de onda de un fragmento de señal de audio de aproximadamente 3 segundos de duración. El fragmento tiene un solo canal (monofónico) y está muestreado a 44.100 muestras por segundo. La codificación original es en PCM utilizando 8 bits por cada muestra.

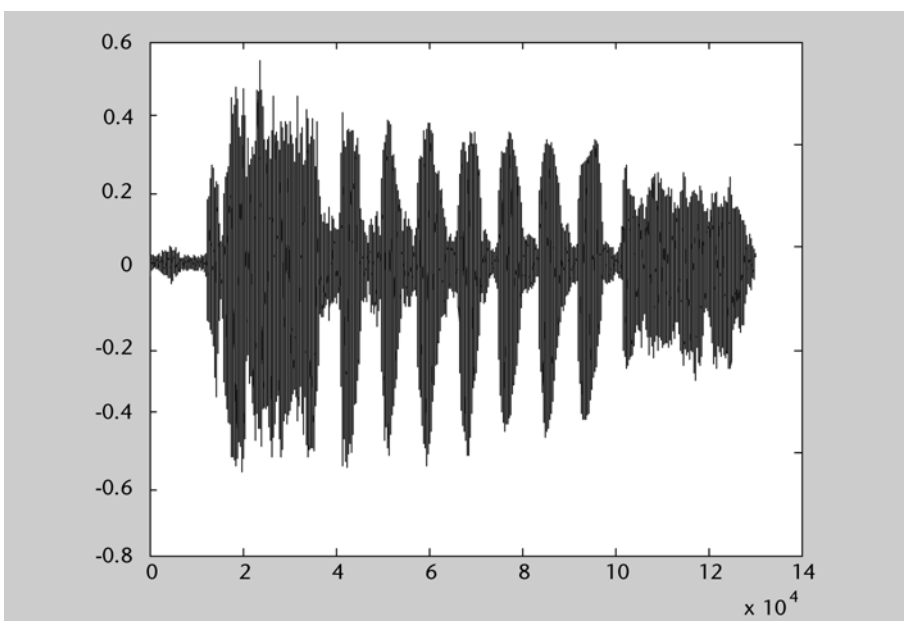


Figura 22. Forma de onda de un fragmento de audio ($f_m = 44.100$ Hz, 8 bits por muestra).

A partir de este fragmento original realizamos la codificación diferencial, es decir, la información que transmitimos o almacenamos es la diferencia entre dos muestras consecutivas. La forma de onda de la señal diferencia obtenida se muestra en la figura 23. Deberíamos observar que las formas de onda de las dos señales tienen un cierto parecido que, por otra parte, resulta lógico. De hecho, la energía de la señal diferencia aumenta en aquellos intervalos en los que la energía de la señal original también aumenta, lo cual resulta perfectamente natural, ya que nuestro error en la estimación aumenta cuando el nivel de la señal es mayor. Aparte de esto, apreciaríamos diferencias significativas en la forma de onda precisa de las dos señales si las observáramos en detalle (realizando un zoom de las gráficas para ver su comportamiento en un intervalo de corta duración). No obstante, la diferencia más importante que existe entre ambas formas de onda es el nivel de señal. Una de las características que podemos apreciar es que la señal diferencia tiene un nivel muy inferior al de la señal original (escala del eje vertical), lo que nos indica que nuestra estimación ha funcionado de la forma esperada.

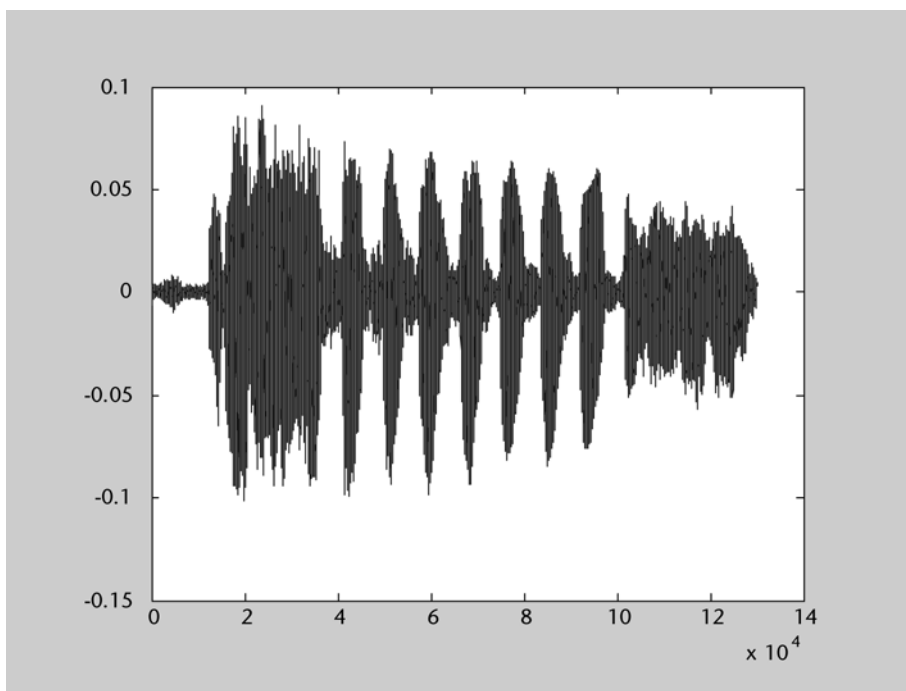


Figura 23. Señal diferencia obtenida a partir del fragmento de audio original de la figura 20.

Para comparar estos resultados, en la figura 24 se representan los histogramas de las dos señales resultantes. Los histogramas se han calculado suponiendo que las señales resultantes se dividen en 256 posibles niveles de cuantificación (256 mensajes). En el eje de abscisas de estas gráficas aparece, por tanto, el número del mensaje y no el valor numérico de la señal. Además, los histogramas están centrados aproximadamente sobre el mensaje 128, que se corresponde con niveles de señal próximos a cero.

Resulta inmediato comprobar que la transformación diferencial aumenta de forma considerable la probabilidad de transmitir los niveles más bajos. Además, si calculamos la entropía de la señal original y de la señal diferencia, ob-

tenemos 6,12 bits y 3,54 bits respectivamente, lo que nos indica que podremos encontrar códigos eficientes para codificar esta última señal por debajo de la tasa media de bits necesaria para codificar la señal original (los criterios de Huffman nos garantizan que la señal diferencia podrá codificarse entre 3,54 y 4,54 bits por muestra de promedio).

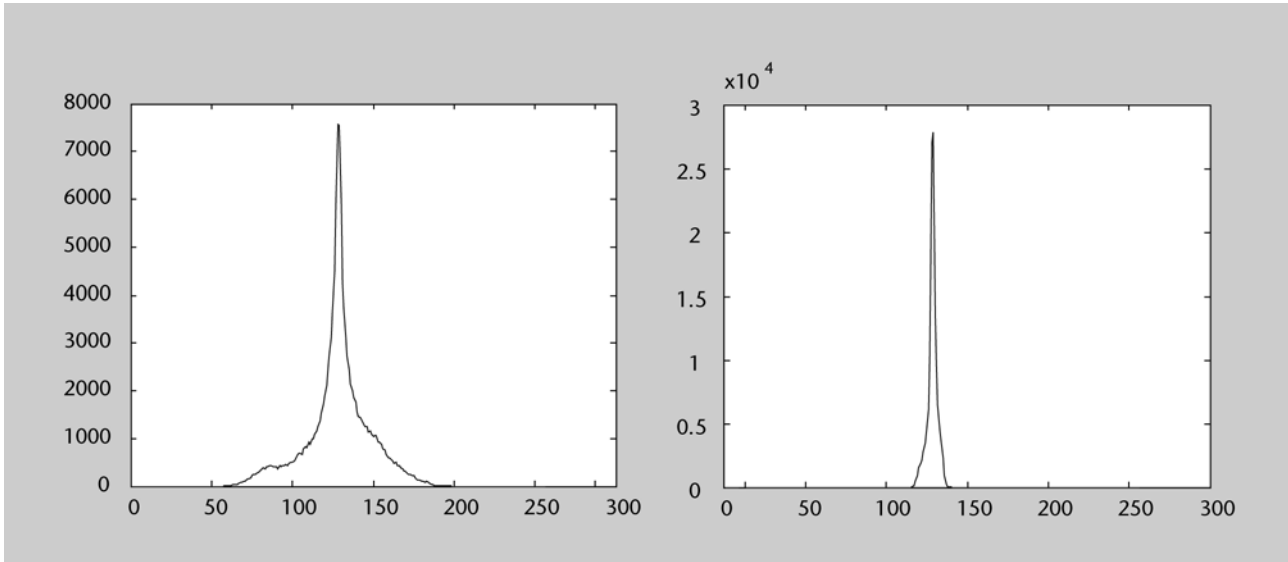


Figura 24. Comparativa entre el histograma de la señal de audio original (izquierda) y la señal diferencia (derecha). El histograma de esta última tiene la mayor parte de sus valores concentrados en la región central.

Finalmente, veamos el resultado que se obtiene cuando utilizamos un predictor. En este ejemplo concreto vamos a utilizar un predictor con 4 coeficientes, es decir, que la estimación de la muestra actual se obtiene ponderando las 4 muestras anteriores con unos coeficientes $\{a_i\}$:

$$\tilde{x}[n] = \sum_{i=1}^4 a_i \cdot x[n-i].$$

En este caso, la señal que transmitimos o almacenamos será la diferencia entre la muestra original y la muestra estimada mediante la ecuación anterior:

$$s[n] = x[n] - \tilde{x}[n].$$

Los coeficientes del predictor pueden calcularse de distintas maneras. Una posibilidad, la más sencilla, consiste en fijar de antemano su valor de acuerdo con razonamientos intuitivos. Así, por ejemplo, podríamos tomar todos los valores a_i iguales a 0,25, con lo que el valor estimado para la muestra actual no es más que el valor promedio de las muestras anteriores. Esta aproximación resulta bastante razonable cuando sabemos que la señal varía muy lentamente y que existe un nivel de ruido significativo en las muestras. Otra posible aproximación sería $a_1 = 0,4$, $a_2 = 0,3$, $a_3 = 0,2$, $a_4 = 0,1$, donde también estamos realizando una especie de promedio, pero dando ahora un mayor peso a las muestras más recientes.

Otra manera de enfocar el problema, más matemática y con mejores resultados, consiste en determinar los coeficientes del predictor que resultarán ópti-

mos para esta señal concreta. Para poder resolver este problema, deberemos definir el concepto de *optimización (optimación)*, es decir, cuál es el criterio que utilizamos para decir que un determinado conjunto de coeficientes es mejor que otro. Generalmente, el criterio de optimización se toma como el valor mínimo del error cuadrático medio de la señal diferencia, esto es:

$$\min \left\{ E[s^2[n]] = \frac{1}{M} \sum_n s^2[n] \right\} .$$

Utilizando este criterio de minimización es posible determinar los coeficientes del predictor óptimo para nuestro fragmento de señal. En nuestro ejemplo, los coeficientes que obtenemos al resolver este problema son:

$$a_1 = 3,2528; a_2 = -4,1701; a_3 = 2,5477; a_4 = -0,6394$$

En la figura 25 se muestra esquemáticamente como se construye a partir de la señal original la señal de error (la señal que transmitiremos o almacenaremos).

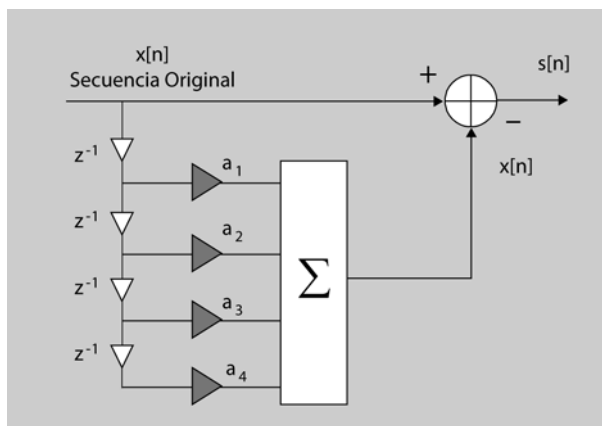


Figura 25. Obtención de la señal diferencia utilizando un predictor lineal de 4 coeficientes.

En la figura 26 hemos representado la forma de onda de la señal $s[n]$ que se obtiene al aplicar el predictor. Como en el caso de la codificación diferencial, la forma genérica se parece a la de la señal original ya que, como hemos visto, ambas deben tener los máximos de energía en los mismos intervalos. No obstante, los detalles de la forma de onda son distintos y, lo que es más importante, el nivel de la secuencia $s[n]$ es considerablemente menor que el de la señal original y que el de la señal diferencia, lo que nos indica que la predicción mediante 4 coeficientes ha supuesto una mejora respecto a la codificación diferencial.

En la figura 27 se muestra el histograma de la señal $s[n]$, mucho más concentrado en unos pocos mensajes que los histogramas mostrados en la figura 24. Para calcular este histograma hemos supuesto que los valores de la señal se dividen en 256 posibles niveles de cuantificación.

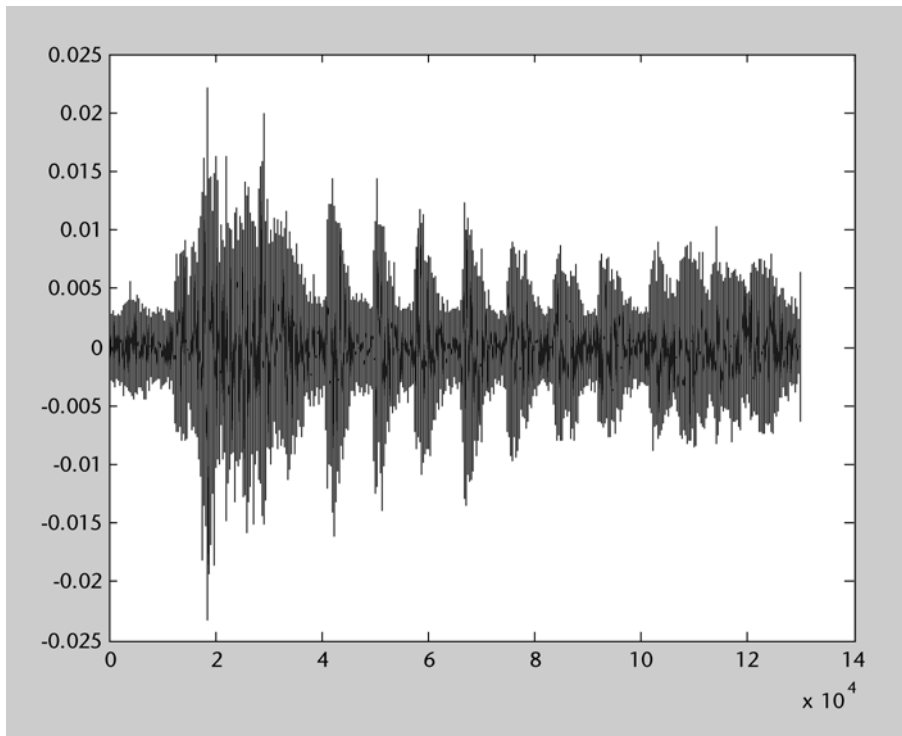


Figura 26. Forma de onda de la señal $s[n]$ obtenida mediante el uso de un predictor de 4 coeficientes.

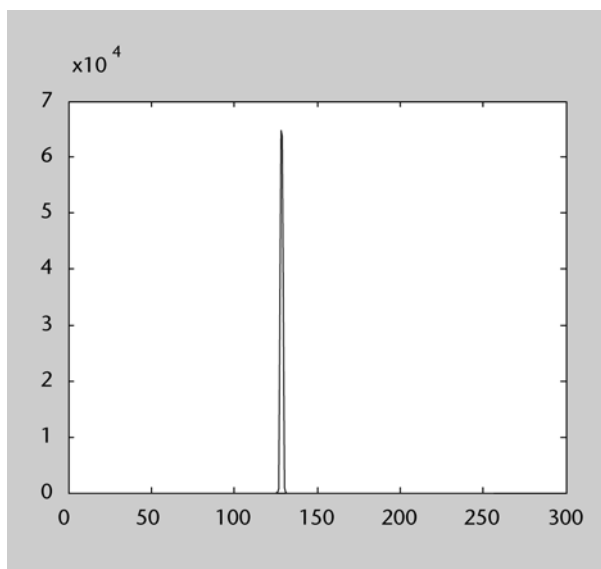


Figura 27. Histograma de la señal $s[n]$ obtenida como la diferencia entre la señal original y la predicción con 4 coeficientes.

También se incluye una tabla comparativa de las entropías de las tres fuentes: original, diferencial y diferencial con predictor. Para determinar las entropías, hemos estimado las probabilidades de las fuentes a partir de los histogramas. Puede resultar curioso “escuchar” las formas de onda de la señal original, el diferencial y la diferencia de predicción utilizadas en este ejemplo*. Al escuchar estas señales se reconoce claramente la melodía original en los ficheros que contienen la señal diferencia y el error de predicción, aunque con una calidad considerablemente inferior. Esta última tiene un nivel muy bajo, debido a que la predicción es bastante buena y, por tanto, el nivel de la diferencia es muy reducido.

* Se proporcionarán los archivos RNoche_orig.wav, RNoche_dif.wav y RNoche_pred.wav en el Aula de la asignatura.

Codificador	Entropía
Señal original (PCM)	6,12
Señal diferencia	3,54
Señal diferencia con predicción 4 coeficientes	1,09

Tabla 3. Comparativa entre las entropías de las 3 fuentes.

Los codificadores que hemos utilizado en el ejemplo anterior son codificadores con pérdidas. La señal original está codificada con 8 bits en PCM (256 niveles), por lo que la señal diferencia, es decir, la resta entre dos muestras consecutivas, puede tomar 511 valores distintos (cualquier número entero entre -255 y 255). Por lo tanto, cuando volvemos a aproximar los resultados obtenidos a 256 niveles, estamos introduciendo pérdidas, con lo que la señal original ya no podrá ser recuperada de forma exacta.

Al reducir el número de posibles mensajes a transmitir, también se reduce la entropía. Cuando el número total de mensajes se compacta a la mitad, la entropía se reduce, como máximo, en un bit. Así, para nuestro problema, puede comprobarse que la entropía que hubiéramos obtenido si se mantiene la señal diferencial sin errores (511 niveles) es de 4,51 bits. Este resultado nos indica que la codificación diferencial, aun cuando se realice sin pérdidas, supone una mejora en compresión respecto a la codificación PCM original, pues conseguimos reducir la entropía en aproximadamente 1,5 bits.

8.4. Codificación diferencial adaptativa

En el ejemplo desarrollado en el apartado anterior, la señal de audio tiene una duración aproximada de 3 segundos y hemos utilizado un único predictor de 4 coeficientes para realizar la estimación de la muestra siguiente. Los coeficientes del predictor se han calculado para que con ellos se obtenga un valor cuadrático mínimo de la señal diferencia. Evidentemente, esta idea puede generalizarse y aplicarla a fragmentos de señal más pequeños, calculando el predictor óptimo para cada uno de estos fragmentos. Esta filosofía de codificación es muy utilizada y recibe el nombre de **codificación adaptativa** (ADPCM, *adaptive differential pulse code modulation*), ya que los coeficientes del predictor van modificándose y adecuándose a las características de la señal a lo largo del tiempo.

Veamos un posible ejemplo simplificado de cómo podría estructurarse un codificador con estas características:

- 1) El codificador analiza la señal en paquetes de longitud preestablecida (por ejemplo, 1.024 muestras).
- 2) Para cada paquete, el codificador determina los coeficientes del predictor óptimo.
- 3) Se transmiten los coeficientes del predictor óptimo al decodificador. También se transmite la señal diferencia de predicción que se ha obtenido en el codificador.
- 4) El decodificador recupera la señal original a partir de las muestras anteriores, los coeficientes del predictor y la señal diferencia.

Esta idea general admite múltiples variantes tanto en los parámetros del sistema (longitud del paquete de muestras, número de coeficientes del predictor, número de mensajes con el que se cuantifica la señal diferencia, etc.) como en la estrategia de cálculo (admitir solapamientos entre paquetes, estrategias de adaptación de los coeficientes del predictor muestra a muestra, etc.).

9. Codificación de voz en telefonía móvil GSM

En este apartado describiremos de manera genérica el sistema de codificación de canal que se utiliza en telefonía móvil GSM. Algunos detalles de este sistema de codificación son muy complejos y escapan a los objetivos de este texto por lo que nos limitaremos a exponer las ideas más generales.

En aplicaciones de telefonía, para obtener una calidad suficientemente satisfactoria mediante una codificación PCM es necesario muestrear la señal de voz a razón de 8.000 muestras por segundo tomando un total de 8 bits por muestra (256 niveles). Esto significa que la tasa de bits para telefonía digital PCM es:

$$R = 8.000 \frac{\text{muestras}}{\text{s}} \times 8 \frac{\text{bits}}{\text{muestra}} = 64.000 \text{ bps}.$$

La RDSI (Red digital de servicios integrados) y otros sistemas de multiplexación troncal de canales digitales utilizan este tipo de codificación para transmitir la señal de voz. No obstante, en telefonía móvil digital, este flujo de datos es elevado y es conveniente comprimirlo.

Además, debemos tener en cuenta que el sistema de compresión no puede ser excesivamente complejo, ya que los procesadores de los teléfonos móviles deben ser simples y de coste reducido: una mayor complejidad computacional se traduce en un mayor consumo de energía, y la duración de la batería es una de las características máspreciadas en un teléfono móvil.

Cuando se elaboró el estándar GSM, se estudiaron y evaluaron diferentes alternativas para la codificación de la señal de voz. Principalmente se tuvieron en cuenta la calidad subjetiva y la complejidad. Finalmente, se eligió un codificador basado en un *vocoder* de predicción lineal, el RPE-LPC-LTP (*Regular pulse excited-linear predictive coder with long term prediction*), al que se le añadieron algunas mejoras adicionales. La idea general de este codificador consiste en dividir la señal de voz en fragmentos de corta duración: 160 muestras, el que corresponde a 20 ms. Cada uno de estos paquetes se codifica con 260 bits, lo que permite codificar la señal de audio a una tasa de 13.000 bps.

Para cada uno de los paquetes:

- 1) Se determinan los coeficientes del predictor óptimo junto con una posible forma de onda que permita reproducir la señal de entrada en el decodificador.
- 2) Se codifican los coeficientes del predictor más los parámetros de la forma de onda con que se excitará el predictor.

Por último, la parte "*Long Term Prediction*" del codificador se refiere a la utilización de una predicción adicional de la forma de onda a partir de los paquetes transmitidos anteriormente.

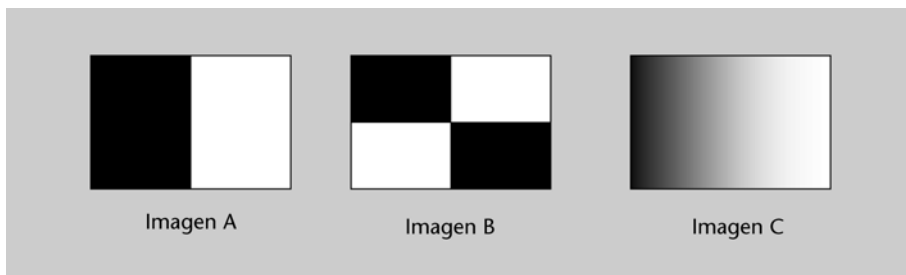
Actividades

1. Intentad justificar por qué las señales diferencia y diferencia de predicción mencionadas al final del apartado 8.3 son inteligibles.

2. Una fuente puede proporcionar 8 posibles mensajes con las siguientes probabilidades: 0,6, 0,1, 0,1, 0,1, 0,025, 0,025, 0,025, 0,025. Se pide:

- a) Determinad el código de Huffman.
- b) Determinad la entropía de la fuente.
- c) Determinad el número medio de bits que utilizará el código de Huffman.

3. Considerad las tres imágenes de la figura 28 e indicad si las siguientes afirmaciones son verdaderas o falsas.



La imagen C es la que tiene mayor entropía.	
La imagen A y la B tienen la misma entropía.	
La imagen B es la que tiene mayor entropía.	
La entropía de la imagen A es de 1 bit.	

4. Comprobad que para la fuente de la figura 12 la entropía es 2,344 y que el número medio de bits del código obtenido en la figura 11 es 2,4.

5. Considerad el código de longitud variable que se proporciona en la tabla siguiente:

Mensaje	Código	Mensaje	Código
M_1	00	M_5	101
M_2	010	M_6	110
M_3	011	M_7	1110
M_4	100	M_8	1111

a) Determinad la palabra código con la que se codificaría la siguiente secuencia de mensajes:

$$M_3 M_5 M_1 M_8 M_1 M_1 M_6 M_7$$

b) Determinad la secuencia de mensajes a la que corresponde la siguiente secuencia de bits:

$$010100101111000000101111100000$$

c) Determinad el valor medio de bits del código suponiendo que todos los mensajes son equiprobables. Razonad intuitivamente por qué el resultado es superior a 3.

6. Considerad el código de longitud variable proporcionado en la tabla del ejercicio anterior. Determinad qué condición deben cumplir las probabilidades de M_1 , M_7 y M_8 para que el código resultante tenga una tasa de bits media menor que 3.

7. Determinad cuánto tiempo se necesitaría para transmitir una imagen de 3.000 filas por 4.000 columnas, codificada en PCM con 24 bpp (bit por píxel), a través de una línea ADSL de 1 Mbps. Suponed que el rendimiento de la línea ADSL es máximo y que se aprovecha todo el flujo de 1 Mbps para transmitir la información útil.

Bibliografía

Básica

Benedetto S.; Biglieri, E. (1999). *Principles of Digital Transmission*. Kluwer Academic Press / Plenum publishers.

Proakis, J. G. (2003). *Digital Communications* (4.ª ed.). McGraw Hill.

Complementaria

Carlson, A. B. (2001). *Communication Systems: An Introduction to Signals and Noise in Electrical Communication* (4.ª ed.). McGraw Hill.

Stix, G. (1991, sept.). "Encoding the Neatness of Ones and Zeroes". *Scientific American* (pág. 54-58).

Segal, J. "El geómetra de la información". *Temas Investigación y Ciencia: La Información* (núm. 36, pág. 12-15).

Bekenstein, Jacob D. "La información en el Universo Holográfico". *Temas Investigación y Ciencia: La Información* (núm. 36, pág 16-23).

Gibson, Jerry D. y otros (1998). *Digital Compression for Multimedia: Principles & Standards*. Morgan Kauffman.

Tarrés, F. (2001). *Sistemas Audiovisuales I: Televisión Analógica y Digital*. Ediciones UPC.