Capítulo 5

Representación de [datos] multimedia

El diccionario de la lengua de la Real Academia Española define el *adjetivo* «multimedia»:

> *1. adj. Que utiliza conjunta y simultáneamente diversos medios, como imágenes, sonidos y texto, en la transmisión de una información.*

El FOLDOC define «multimedia» como *sustantivo:*

> *<multimedia> Any collection of data including text, graphics, images, audio and video, or any system for processing or interacting with such data. Often also includes concepts from hypertext.*

En todo caso, es bien conocida la importancia actual de la representación de estos tipos de contenidos, que, además de textos (e hipertextos), incluyen sonidos (audio), imágenes (gráficos y fotografías) e imágenes en movimiento (animaciones y vídeos).

Salvo los textos e hipertextos, ciertas imágenes (las generadas por aplicaciones gráficas) y ciertos sonidos (los generados por sintetizadores digitales), la mayoría de las fuentes de datos multimedia son señales analógicas. Para almacenar, procesar y transmitir en formato digital una señal analógica es necesario *digitalizarla*, es decir, convertirla en una sucesión de datos numéricos representables con un número limitado de bits.

En la asignatura «Sistemas y señales» de segundo curso habrá usted estudiado los principios teóricos de este proceso de digitalización, en particular el teorema del muestreo y las conversiones de señales analógicas a digitales y viceversa. En el apartado siguiente los resumiremos, poniendo énfasis en las señales multimedia.

Por otra parte, dado el gran volumen de datos numéricos que resultan para representar digitalmente sonidos e imágenes, normalmente se almacenan y se transmiten comprimidos. El asunto de la compresión lo trataremos en el capítulo siguiente, limitándonos en éste a las representaciones numéricas sin comprimir. Pero hay que tener presente que en la práctica ambos procesos (digitalización y compresión) van unidos. Se llama **codec** (codificador/decodificador) al conjunto de convenios (frecuencia de muestreo, algoritmo de compresión, etc.) y al sistema hardware y/o software que se utiliza.

5.1. Digitalización

La cadena analógico-digital-analógico

Cuando la fuente de los datos es una señal analógica y el resultado ha de ser también una señal analógica, el procesamiento y/o la transmisión digital requiere conversiones de analógico a digital y viceversa. La figura 5.1 ilustra el proceso para el caso particular del sonido. En este caso el procesamiento previo a la conversión a digital y el posterior a la conversión a analógico suelen incluir filtrado y amplificación.

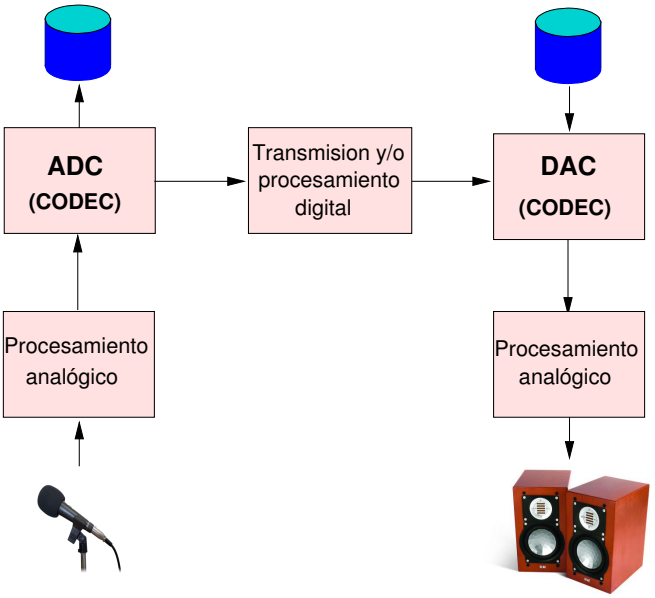


Figura 5.1: Procesamiento del sonido

El caso de las imágenes es similar si la fuente de tales imágenes es una cámara que genera señales analógicas. Las cámaras fotográficas y de vídeo actuales incorporan ya en sus circuitos los codecs adecuados, de manera que la imagen o el vídeo quedan grabados digitalmente en su memoria interna.

Los codecs, como ya hemos dicho, incluyen tanto los elementos que realizan las conversiones de analógico a digital (ADC: Analog to Digital Converter) y de digital a analógico (DAC: Digital to Analog Converter) como los que se ocupan de la compresión y la descompresión. Recordemos brevemente los principios de los primeros.

Analógico a digital y digital a analógico

La conversión de una señal analógica en una señal digital se realiza en tres pasos (figura 5.2):

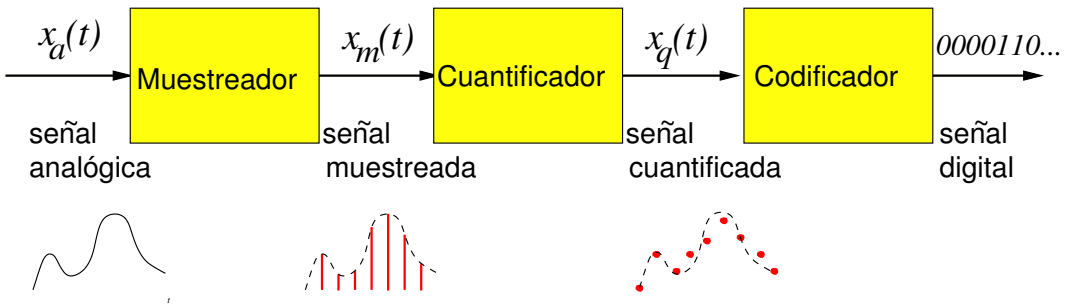
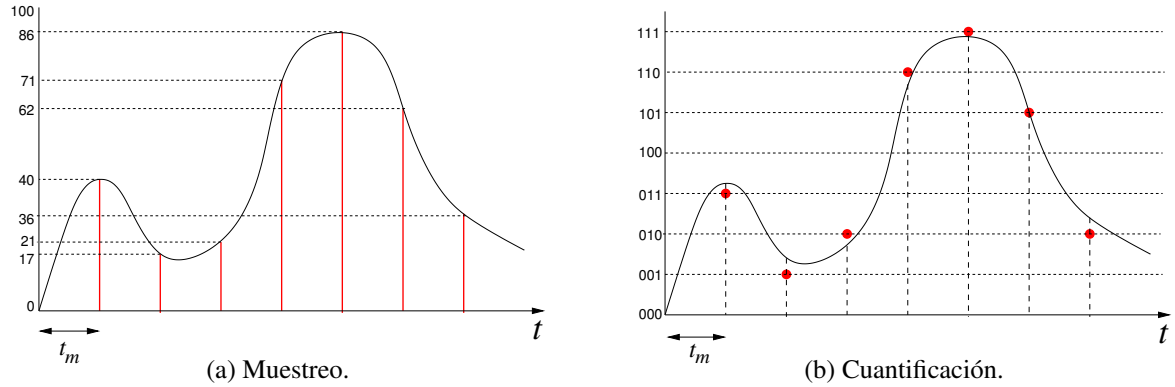


Figura 5.2: Conversión de analógico a digital.

1. El **muestreo** es una *discretización en el tiempo*[1]. El resultado es una sucesión temporal de **muestras**: valores de la señal original en una sucesión de instantes separados por el **período de muestreo**, t_m. En el ejemplo de la figura 5.3a, si $t_m = 1$, estos valores son 0 en el instante 0, 40 en el 1, 17 en el 2, 21 en el 3, etc.

2. La **cuantificación** es una *discretización de la amplitud*. Las muestras son valores reales que han de representarse con un número finito de bits, n. Con n bits podemos representar 2^n valores distintos o **niveles**, y cada muestra original se representa por su nivel más cercano. El número de bits por muestra se llama **resolución**. La figura 5.3b indica el resultado de cuantificar las muestras del fragmento de señal del ejemplo con 8 niveles (resolución $n = 3$).



(a) Muestreo. (b) Cuantificación.

Figura 5.3: Muestreo y cuantificación.

3. En su función de **codificación**, un códec aplica (mediante software o mediante hardware) determinados algoritmos a las muestras cuantificadas y genera un flujo de bits que puede transmitirse por un canal de comunicación *(streaming)* o almacenarse en un fichero siguiendo los convenios de un formato. En el ejemplo anterior, y en el caso más sencillo (sin compresión), este flujo sería la codificación binaria de la secuencia de muestras: 000011001010...

La conversión inversa (de digital a analógico) se realiza con un descodificador acorde con el codificador utilizado y una interpolación para reconstruir la señal analógica. Lo ideal sería que esta señal reconstruida fuese idéntica a la original, pero en los tres pasos del proceso de digitalización y en la interpolación se pueden perder detalles que lo impiden, como ilustra la figura 5.4, en la que la función de interpolación es simplemente el mantenimiento del nivel desde una muestra a la siguiente *(zero-holder hold)*. Veamos cómo influye cada uno de los pasos del proceso.

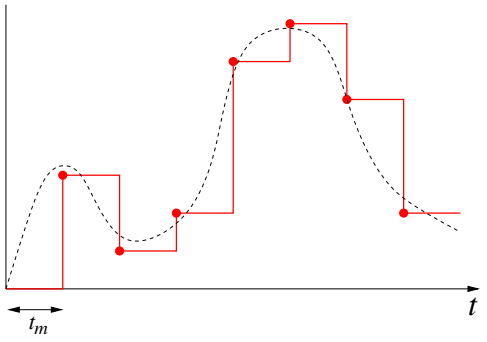


Figura 5.4: Reconstrucción con *zero-holder hold*.

[1] «Discretizar» una magnitud continua es convertirla en otra discreta. «Discreto» es (R.A.E.):

«5. adj. Mat. Dicho de una magnitud: Que toma valores distintos y separados. La sucesión de los números enteros es discreta, pero la temperatura no.»

Muestreo

Si el período de muestreo es t_m, la frecuencia de muestreo es $f_m = 1/t_m$. Como debe usted saber, según el teorema del muestreo de Nyquist-Shannon [2], si la señal analógica tiene una frecuencia máxima f_M basta con muestrear con $f_m \geq 2 \times f_M$ para poder reconstruir *exactamente* la señal original a partir de las muestras. A $f_m/2$ se le llama **frecuencia de Nyquist**. Si la señal original contiene componentes de frecuencia superior a la de Nyquist al reconstruirla aparece el fenómeno llamado ***aliasing***: cada uno de esos componentes genera otro espurio (un «alias») de frecuencia inferior a f_M que no estaba presente en la señal original. Éste es el motivo por el que generalmente se utiliza un filtro paso bajo analógico antes del muestreo.

Cuantificación

La figura 5.3b ilustra el caso de una cuantificación lineal: los valores cuantificados son proporcionales a las amplitudes de las muestras. Pero se utiliza más la *cuantificación logarítmica*, en la que los valores son proporcionales a los logaritmos de las amplitudes.

A diferencia del muestreo, la cuantificación introduce *siempre* una distorsión (**ruido de discretización**) tanto mayor cuanto menor es la **resolución** (número de bits por muestra). Depende de la aplicación, pero resoluciones de 8 (256 niveles) o 16 (65.536 niveles) son las más comunes.

Codificación

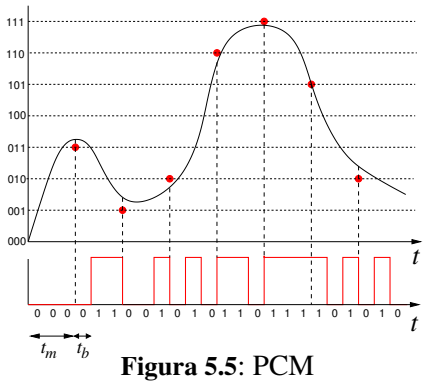


Figura 5.5: PCM

La forma de codificación más sencilla es **PCM** (Pulse-Code Modulation): en el intervalo de tiempo que transcurre entre la muestra n y la muestra $n+1$ (período de muestreo, t_m) se genera un tren de impulsos que corresponde a la codificación binaria de la muestra n. En la figura 5.5 puede verse la sucesión de bits generados para el ejemplo anterior.

Con esta codificación el período del flujo de bits es $t_b = t_m/R$, donde R es la resolución. Por tanto, el *bitrate* resultante es $f_m \times R$. En el caso de que la señal corresponda a una voz humana se puede aprovechar que hay una correlación entre muestras y codificar con DPCM (Diferencial PCM) o ADPCM (Adaptive DPCM), reduciéndose notablemente el *bitrate* a costa de una pequeña pérdida de calidad. Pero esto ya es una forma de compresión, asunto que trataremos en el capítulo siguiente.

Interpolación

El *zero-holder hold* (figura 5.4) introduce bastante distorsión. En los DAC se suele utilizar interpolación lineal (*first-holder hold*) o polinómica. En teoría, si la frecuencia de muestreo es mayor que la de Nyquist (y obviando el ruido de discretización), se podría reconstruir perfectamente la señal original, pero la fórmula exacta es prácticamente irrealizable[3].

[2] «Theorem 1: If a function $f(t)$ contains no frequencies higher than W cps, it is completely determined by giving its ordinates at a series of points spaced $1/(2W)$ seconds apart.» C. E. Shannon: Communication in the presence of noise. Proc. Institute of Radio Engineers, 37, 1 (Jan. 1949), pp. 10–21. Reproducido en Proc. IEEE, 86, 2 (Feb. 1998).

[3] En el artículo citado en la nota anterior se demuestra esta fórmula de interpolación: $f(t) = \sum_{n=-\infty}^{\infty} x_n \frac{\sin \pi(2Wt-n)}{\pi(2Wt-n)}$, donde x_n es el valor de la muestra enésima.

5.2. Representación de sonidos

Ya sea la fuente del sonido analógica o un secuenciador digital, normalmente el sonido queda representado, a efectos de almacenamiento o transmisión digital, como una secuencia de bits. La secuencia resultante para una determinada señal analógica depende de la frecuencia de muestreo y de la resolución. Suponiendo que la codificación es PCM, el *bitrate* es:

$$f_b = f_m \times R \times C \quad \text{kbps}$$

donde f_m es la frecuencia de muestreo en kHz (miles de muestras por segundo), R la resolución en bits por muestra y C el número de canales.

El límite superior de frecuencias humanamente audibles es aproximadamente 20 kHz, por lo que sería necesaria una frecuencia de muestreo de $f_m \geq 40$ kHz (40.000 muestras por segundo) y una resolución de 16 bits por muestra para permitir una reconstrucción prácticamente perfecta. No obstante, ciertas aplicaciones no son tan exigentes, y relajando ambos parámetros se pierde calidad pero se reduce la necesidad de ancho de banda en la transmisión y de capacidad de almacenamiento (que es el producto del *bitrate* por la duración de la señal). En la tabla 5.1 se resumen valores típicos para varias aplicaciones.

Aplicación	f_m **(kHz)**	R **(bits)**	C	f_b **(kbps)**	**En 1 minuto...**
Telefonía	8	8	1	64	480 kB (≈468 KiB)
Radio AM	11	8	1	88	660 kB (≈644 KiB)
Radio FM	22,05	16	2	705,6	5.292 kB (≈5 MiB)
CD	44,1	16	2	1.411,2	10.584 kB (≈10 MiB)
TDT	48	16	2	1.456	11.520 kB (≈11 MiB)

Tabla 5.1: Tasas de bits y necesidades de almacenamiento para algunas aplicaciones de sonido.

En las aplicaciones que pretenden una reconstrucción perfecta la frecuencia de muestreo es algo superior a 40 kHz[4]. Esto se explica porque aunque la señal original se someta a un filtrado para eliminar las frecuencias superiores a 20 kHz, el filtro no es perfecto: una componente de 21 kHz aún pasaría, aunque atenuada, lo que provocaría el *aliasing*.

Representación simbólica

La representación en un lenguaje simbólico de algunas características de los sonidos no es nada nuevo: los sistemas de notación musical se utilizan desde la antigüedad. Aquí nos referiremos a lenguajes diseñados para que las descripciones de los sonidos sean procesables por programas informáticos. Nos limitaremos a citar algunos ejemplos (si está usted interesado no le resultará difícil encontrar abundante información en Internet):

- La **notación ABC** es un estándar para expresar en texto ASCII la misma información que la notación gráfica de pentagrama.
- **MIDI** (Musical Instrument Digital Interface) es otro estándar que no solamente incluye una notación, también un protocolo e interfaces para la comunicación entre instrumentos electrónicos.
- **MusicXML** es un lenguaje basado en XML con mayor riqueza expresiva que MIDI.
- **VoiceXML** está más orientado a aplicaciones de síntesis y reconocimiento de voz.

[4]El que el estándar de audio CD determine precisamente 44,1 kHz se debe a una historia interesante. Si tiene usted curiosidad puede leerla en `http://www.cs.columbia.edu/~hgs/audio/44.1.html`.

5.3. Representación de imágenes

Como con el sonido, podemos distinguir entre la representación binaria de una imagen y la descripción de la misma mediante un lenguaje simbólico. En el primer caso se habla de «imágenes matriciales», y en el segundo de «gráficos vectoriales». Estos últimos tienen un interés creciente, sobre todo para las aplicaciones web, por lo que nos extenderemos en ellos algo más de lo que hemos hecho para los sonidos.

Imágenes matriciales

Una imagen matricial *(raster image)* es una estructura de datos que representa una matriz de píxeles. Los tres parámetros importantes son el ancho y el alto en número de píxeles y el número de bits por píxel, **bpp**. A veces se le llama en general «*bitmap*», pero conviene distinguir entre:

- **bitmap** para imágenes en blanco y negro (1 bpp),
- **greymap** para imágenes en escala de grises (n bpp; el número de tonos es 2^n), y
- **pixmap** para imágenes en color (n bpp es la **profundidad de color**; el número de colores es 2^n)

La **resolución** de la imagen matricial mide la calidad visual en lo que respecta al grado de detalle que puede apreciarse en la misma. Lo más común es expresar la resolución mediante dos números enteros: el de columnas de píxeles (número de píxeles de cada línea) y el de líneas. O también, mediante el producto de ambos. Así, de una cámara con una resolución 10.320 por 7.752 se dice que tiene $10.320 \times 7.752 = 80.000.640 \approx 80$ megapíxeles.

A veces con el término «resolución» se designa a la **densidad de píxeles**, medida en píxeles por pulgada (ppi).

Los conceptos sobre digitalización resumidos antes se aplican también a las imágenes matriciales, cambiando el dominio del tiempo por el del espacio: la «pixelación» es un muestreo en el espacio. Pero hay un cambio en la terminología: lo que en la señal temporal muestreada se llama «resolución» (número de bits por muestra) aquí es «profundidad de color», y lo que en imágenes se llama «resolución» corresponde a la frecuencia de muestreo.

El teorema del muestreo normalmente se formula para funciones de una variable, el tiempo, pero es aplicable a funciones de cualquier número de variables, y, por tanto, a las imágenes digitalizadas. Y fenómenos como el *aliasing* aparecen también en estas imágenes: en la televisión convencional podemos observarlo (en alta definición menos) cuando una persona viste una camisa de rayas delgadas y muy juntas (frecuencia espacial grande).

Gráficos vectoriales

La representación «vectorial» de las imágenes es un enfoque totalmente distinto al de la representación «matricial». En lugar de «ver» la imagen como una matriz de píxeles, se *describe* como un conjunto de objetos primitivos (líneas, polígonos, círculos, arcos, etc.) definidos matemáticamente en un lenguaje.

Para entenderlo, veamos un ejemplo concreto en el lenguaje SVG *(Scalable Vector Graphics)*, que es un estándar del W3C (Consorcio WWW). En la figura 5.6 puede usted ver el código en SVG y el resultado visual de una imagen sencilla. Aun sin saber nada del lenguaje (que está basado en XML), es fácil reconocer la correspondencia entre las sentencias y la imagen. Después de la cabecera (las cinco primeras líneas), aparecen tres declaraciones de rectángulos («`<rect... />`») con sus propiedades (posición, dimensiones y color).

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg xmlns="http://www.w3.org/2000/svg"
     width="100%" height="100%">
<rect x="0" y="0" width="240" height="50"
      stroke="red" stroke-width="1" fill="red" />
<rect x="0" y="50" width="240" height="70"
      stroke="yellow" stroke-width="1" fill="yellow" />
<rect x="0" y="120" width="240" height="50"
      stroke="red" stroke-width="1" fill="red" />
</svg>
```

Figura 5.6: Bandera.svg.

Salvo que se trate de un dibujo muy complicado, el tamaño en bits de una imagen descrita en SVG es mucho menor que el necesario para representarla como imagen matricial. El ejemplo de la figura, con una resolución 241 por 171 y una profundidad de color de 8 bits, necesitaría 241×171 = 41.211 bytes. Utilizando un formato comprimido (PNG) con la misma resolución y la misma profundidad de color (así se ha hecho en el original de este documento) ocupa solamente 663 bytes. Pero el fichero de texto en SVG tiene sólo 486 bytes (y además, se puede comprimir, llegando a menos de 300 bytes).

La representación vectorial tiene otra ventaja: es independiente de la resolución. Una imagen matricial tiene unos números fijos de píxeles horizontales y verticales, y no se puede ampliar arbitrariamente sin perder calidad (sin «pixelarse»). La imagen vectorial se puede ampliar todo lo que se necesite: la calidad sólo está limitada por el hardware en el que se presenta.

Sin embargo, la representación vectorial sólo es aplicable a los dibujos que pueden describirse mediante primitivas geométricas. Para imágenes fotográficas es absolutamente inadecuada.

5.4. Representación de imágenes en movimiento

La propiedad más importante de la visión humana aprovechable para la codificación de imágenes en movimiento es la de **persistencia**: la percepción de cada imagen persiste durante, aproximadamente, 1/25 seg. Esto conduce a lo que podemos llamar el *«principio de los hermanos Lumière»*: para conseguir la ilusión de movimiento continuo basta con presentar las imágenes sucesivas a un ritmo de 30 fps. «fps» es la abreviatura de «frames por segundo»; en este contexto, una traducción adecuada de «frame» es «**fotograma**»[5].

Observe que ahora hay un muestreo en el tiempo (añadido, en su caso, a la digitalización de cada fotograma), y el teorema del muestreo sigue siendo aplicable. Habrá visto, por ejemplo, las consecuencias del *aliasing* cuando la escena contiene movimientos muy rápidos con respecto a la tasa de fotogramas (por ejemplo, en el cine, cuando las ruedas de un vehículo parecen girar en sentido contrario).

El problema de aumentar la tasa de fotogramas es que aumenta proporcionalmente la tasa de bits. En animaciones en las que se puede admitir una pequeña percepción de discontinuidad la tasa puede bajar a 12 fps. En cinematografía son 24 fps, pero duplicados o triplicados ópticamente (la película

[5]Es importante la matización «en este contexto». En transmisión de datos «frame» se traduce por «trama», y en otros contextos por «marco»

avanza a 24 fps, pero con esa tasa se percibiría un parpadeo; el proyector repite dos o tres veces cada fotograma). En televisión analógica los estándares son 25 fps (PAL) o 30 fps (NTSC), pero en realidad se transmiten 50 o 60 campos por segundo entrelazados (un campo contiene las líneas pares y otro las impares). La HDTV en Europa utiliza 50 fps. Los monitores LCD suelen configurarse para 60 o 75 fps.

Sabiendo la duración de un vídeo, el número de fps y la resolución de cada fotograma es fácil calcular la capacidad de memoria necesaria para su almacenamiento y de ancho de banda para su transmisión. Por ejemplo, la película «Avatar» tiene una duración de 161 minutos. Si la digitalizamos con los parámetros de HD (1.980×1.080, 50 fps, 32 bpp), pero sin compresión, la tasa de bits resulta $1.980 \times 1.080 \times 50 \times 32 = 3.421{,}44 \times 10^6$ bps. Es decir, necesitaríamos un enlace de más de 3 Gbps para transmitir en tiempo real solamente el vídeo. Toda la película (sin sonido) ocuparía $3.421{,}44 \times 10^6 \times 161 \times 60/8 \approx 4{,}13 \times 10^{12}$ bytes. Es decir, $4{,}13 \times 10^{12}/2^{40} \approx 3{,}76$ TiB. Un disco de 4 TiB sólo para este vídeo. Es evidente la necesidad de compresión para estas aplicaciones.