

ACCADEMIA DI BELLE ARTI DI URBINO

Progettazione e Arti applicate

**NUOVE TECNOLOGIE DELL'ARTE**

# Codifica e compressione delle informazioni

# Rappresentazione digitale delle informazioni

## Definizione di informazione

In termini di economia del linguaggio possiamo affermare che **informazione** è tutto ciò che riduce incertezza. Esempi: una foto riduce l'incertezza di chi la osserva riguardo alla scena raffigurata, l'espressione del volto di un interlocutore riduce l'incertezza sul suo stato d'animo, un fruscio riduce l'incertezza sul luogo in cui cercare un fagiano, ogni parola riduce l'incertezza sul significato di una frase.

È una definizione limitata (in termini di antropologia culturale sappiamo che non sempre ciò che aggiunge sapere sottrae incertezza, anzi...) ma che ci aiuta ad introdurre la definizione successiva: la definizione di dato.

Per **dato** intendiamo un'informazione codificata. Quando l'informazione viene codificata in un linguaggio convenzionale per poter essere rappresentata, scambiata, memorizzata ed elaborata otteniamo un **dato**.

La codifica delle informazioni può rendere omogenea la rappresentazione e l'elaborazione di informazioni di natura diversa (suoni, immagini, testi, numeri) e renderli disponibili per l'elaborazione **automatica**.

Informatica significa **informazione automatica**. Più in particolare diciamo che l'informatica è quella disciplina che studia l'elaborazione automatica delle informazioni. Condizione prima perché le informazioni possano essere elaborate e che vengano codificate in un linguaggio convenzionale che l'informatica possa elaborare.

Ora vediamo di comprendere meglio il concetto di automazione. Per **automatico** si intende tutto ciò che compie un compito prestabilito senza l'intervento umano. Quindi quando si parla di *automazione* si fa riferimento alla realizzazione di strumenti per la soluzione di problemi o l'esecuzione di processi in maniera indipendente dall'attività umana.

In termini di economia dei processi un qualche processo viene automatizzato quando il numero di volte che esso deve essere eseguito è sufficientemente grande da rendere conveniente la progettazione e la costruzione di un sistema automatico che lo risolva. Gli stessi problemi che vengono trattati in maniera automatica sono stati originariamente trattati dall'uomo.

### Alcuni esempi

<b>Problema originale</b>	<b>Problema sistematico</b>
<i>Soluzione manuale</i>	<i>Soluzione automatica</i>
Determinare il diametro di una pallina	Classificare un insieme di palline in base al diametro
<i>Calibro</i>	<i>Griglie forate</i>
Scrivere una lettera	Riprodurre un testo in migliaia di copie
<i>Carta e penna</i>	<i>Stampa tipografica</i>
Regolare il traffico in caso di emergenza	Regolare il traffico ad ogni incrocio
<i>Vigile urbano</i>	<i>Semaforo</i>
Riempire un contenitore d'acqua	Riempire ripetutamente un contenitore
<i>Rubinetto manuale</i>	<i>Rubinetto a galleggiante</i>

## Che cosa è un computer

Un elaboratore elettronico è una macchina (elettronica) programmabile per l'elaborazione automatica di dati. Un elaboratore è costituito da una parte così detta Hardware (componenti tangibili e permanenti di un computer). E da una parte software (insieme dei programmi, componenti intangibili e temporanei, che controllano il funzionamento dell'hardware).

L'hardware è in grado di eseguire un insieme finito di operazioni elementari e di interpretare istruzioni in un linguaggio predefinito. Il software specifica sequenze di istruzioni da eseguire per arrivare alla soluzione di un problema.

Un computer è programmabile. È in grado, cioè, di svolgere compiti diversi in base ad un programma. Un elaboratore può essere utilizzato ripetutamente per risolvere problemi diversi o istanze diverse dello stesso problema. In molti casi, l'automazione di un processo è conveniente solo se si può realizzare con elaboratori programmabili già progettati e sviluppati per altre applicazioni. In tal caso solo il programma deve essere riprogettato. Il processo di codifica consente di utilizzare lo stesso elaboratore per manipolare informazioni diverse. Ogni informazione può (a meno di un'approssimazione) essere rappresentata come una sequenza di due soli simboli (di solito 0 e 1) che nel campo informatico vengono chiamati **BIT**.

La parola **BIT** nasce dalla contrazione delle parole inglesi Binary Digit e letteralmente BIT significa "cifra binaria". Cioè una cifra che può assumere solo due valori (di solito si parla di 0 e 1). La rappresentazione

fisica di un bit si può ottenere da un qualunque sistema in grado di trovarsi in due stati diversi mutuamente esclusivi. Ad esempio un bit può essere rappresentato da un interruttore che può trovarsi nei due stati acceso/spento. L' interruttore infatti si trova alla base del funzionamento dei circuiti logici che costituiscono le unità di elaborazione dei microprocessori.

Allo stesso modo un bit può essere rappresentato da un condensatore che si trova nei due stati carico/scarico. Tutte le memorie principali così dette dinamiche (come la **RAM**) si basano su questo principio, memorizzando i singoli bit in piccolissimi condensatori in modo tale da rappresentare un 1 se il condensatore è carico e uno 0 se questo è scarico.

D'altra parte, nelle memorie di massa non volatili come i dischi fissi, i bit vengono memorizzati attraverso particelle magnetiche orientabili. In questo modo per rappresentare un bit di valore 1 si orienta una particella in una direzione e per rappresentare uno 0 si orienta nel verso opposto.

La codifica delle informazioni ci consente, cioè, di utilizzare determinati supporti fisici (hardware) per rappresentarle.

## Codifica delle informazioni

La codifica delle informazioni in termini di bit è l'operazione che consente di utilizzare i calcolatori elettronici per elaborare dati di qualsiasi natura. Abbiamo già detto che un bit può rappresentare solamente due diverse informazioni (1 o 0, acceso o spento, ecc.). Per codificare insieme con un numero maggiore di informazioni/stati usiamo stringhe di bit di lunghezza variabile. Una stringa è una sequenza di  $n$  bit e viene anche detta parola dall'inglese word. Con un numero  $n$  bit si possono formare  $2^n$  configurazioni diverse e quindi si possono rappresentare  $2^n$  informazioni diverse. La codifica è una convenzione che associa un significato ad ogni configurazione di bit.

I calcolatori sono oggetti finiti che elaborano e memorizzano un numero finito di bit. Un numero finito di bit permette di codificare un numero finito di informazioni diverse. Se l'insieme delle informazioni da rappresentare è superiore alle combinazioni di BIT che ho a disposizione non avrò una rappresentazione esatta delle informazioni ma una rappresentazione approssimata.

Un insieme di informazione può essere limitato o illimitato (infinito) e può essere discreto a continuo. Facciamo alcuni esempi.

L'insieme dei **caratteri** di una lingua è limitato e discreto in quanto composto da un numero definito di segni.

L'insieme dei **numeri interi** è infinito e discreto: la sequenza dei numeri è infinita, ma, dati due valori

qualsiasi, il numero degli elementi compresi tra i due valori è sempre finito.

L'insieme dei **numeri reali** è invece infinito e continuo: la sequenza dei numeri reali è infinita e il numero di elementi compresi tra due elementi qualsiasi della sequenza anche molto vicini tra loro è sempre infinito.

L'insieme dei **colori** è limitato e continuo: la sensibilità del mio occhi limita la sequenza dei colori dal rosso al violetto, ma tra due elementi della sequenza potrò sempre immaginare un numero infinito di sfumature.

La strategia adottata nella codifica cambia a seconda delle caratteristiche dell'insieme di informazioni da rappresentare.

- Normalmente gli insiemi finiti e discreti (come i caratteri di una lingua) vengono rappresentati esattamente.
- Gli insiemi illimitati vengono limitati. Nella convenzione che definisce la codifica si stabilisce che venga rappresentato un sottoinsieme di informazioni la cui dimensione dipende dal numero di bit che si dedicano alla codifica (gli elementi del sottoinsieme sono rappresentati esattamente, gli altri non sono rappresentati).
- Gli insiemi continui vengono partizionati. Normalmente l'insieme viene suddiviso in intervalli di uguale ampiezza e viene codificato come se fosse un insieme discreto.

Gli insiemi illimitati e continui saranno sia limitati che partizionati.

## Codifica del Testo

Un testo è una sequenza di caratteri alfabetici, separatori e caratteri speciali come punteggiatura, accentuazioni, ecc. La codifica **ASCII** (American Standard Code for Information Interchange) prevede l'uso di 128 caratteri diversi. Ogni carattere è associato ad una diversa configurazione di 7 bit.

La codifica ASCII si è piuttosto limitata. Modellata sulle lingua inglese non comprende le lettere con modificatori quali accenti, dieresi, tilde ecc. Per queste limitazioni sono nate estensioni non standard del codice ASCII che utilizzano 8 bit (1 byte) per carattere, portando a 256 il numero di caratteri disponibili. Si sono diffuse quindi diverse tabelle di caratteri (specializzate per gruppi di lingue) che hanno in comune i primi 128 caratteri (ASCII standard) e usano i rimanenti per offrire alle lingue di riferimenti tutti i caratteri necessari. Questo processo ha reso la codifica dei caratteri relativa. Il codice di un carattere non è univoco ma dipende dalla tabella che viene impiegata. Ad esempio in un computer che esegue Microsoft Windows utilizzando la tabella codici predefinita 1252 e in un computer Apple Macintosh che utilizza la tabella codici Macintosh Roman, la posizione 232 di queste tabelle dei caratteri corrisponde rispettivamente alla lettera è nel sistema Windows ed alla lettera Ë in quello Macintosh.

Con l'entrata delle lingue asiatiche nel mondo dei personal computer è nata l'esigenza di aumentare in maniera sensibile il numero dei segni da codificare. Nasce quindi due progetti col fine di costruire un'unica



tabella universale dei caratteri, il progetto ISO 10646 e il progetto Unicode.

Lo standard ISO10646/Unicode si basa su una codifica a 32 bit che consente oltre due miliardi di possibili caratteri, numero che dovrebbe essere decisamente sufficiente per realizzare una tavola di codifica dei caratteri veramente universale.

L'implementazione informatica di Unicode tuttavia comporta dei problemi, in particolar modo per quanto riguarda la crescita della dimensione dei files, per tal motivo sono stati creati dei charset detti UTF, *Universal Character Set Transformation Format*.

UTF usa una tecnica di bit-shifting (spostamento dei bit) per codificare i caratteri Unicode. In sostanza l'UTF usa 7 bit per carattere per codificare i primi 127 caratteri corrispondenti all'ASCII standard, e attiva l'ottavo bit solo quando serve la codifica Unicode. Ogni carattere Unicode viene quindi codificato con un numero variabile di byte che va da 1 a 3. Il vantaggio è che il testo è visualizzabile con un normale editor di testi, anche se naturalmente i caratteri diversi dall'ASCII standard potrebbero apparire non correttamente. Per i testi comuni utilizzati in occidente consente di mantenere più compatte le dimensioni dei files rispetto ad UCS in quanto la maggior parte dei caratteri sono ASCII. Inoltre, l'UTF-8 attraversa indenne, come se fosse un normale testo, anche i sistemi e i programmi che non supportano Unicode. Per questi motivi l'UTF è molto usato nella posta Internet ed in alcuni sistemi operativi.

Tra le codifiche UTF la più comune è la UTF-8, che tra l'altro è la codifica di default di XML.

Indipendentemente dalla codifica usata un testo è rappresentato dalla sequenza di byte associati ai caratteri che lo compongono, nell'ordine in cui essi compaiono. Un testo di 1000 caratteri richiede 1000 byte (1 Kb) se viene utilizzata una tabella di testo ANSI, 4000 byte (4 Kb) se viene utilizzata la codifica Unicode e un numero variabile di byte, che dipende dai caratteri codificati se si usa una codifica UTF.

## Codifica dei numeri

Per la rappresentazione dei numeri è particolarmente diffusa la codifica binaria, che fornisce una diretta trasposizione in un alfabeto a due valori della notazione decimale da noi comunemente utilizzata. Il numero di simboli dell'alfabeto è detto base di numerazione. La codifica binaria e quella decimale utilizzano basi diverse (2 e 10, rispettivamente) ma sono entrambe notazioni posizionali, in quanto permettono di rappresentare ogni numero come sequenza di cifre del tipo:

$$c_n \ c_{n-1} \ \dots \ c_1 \ c_0 \ c_{-1} \ c_{-2} \ \dots \ c_{-m}$$

Il valore ( $v$ ) del numero rappresentato è la somma pesata delle cifre, dove il peso di ogni cifra dipende dalla base di numerazione ( $B$ ) e dalla posizione della cifra:

$$v = c_n B^n + c_{n-1} B^{n-1} + \dots + c_1 B + c_0 + c_{-1} B^{-1} + c_{-2} B^{-2} + \dots + c_{-m} B^{-m}$$

Es:  $(101.1)(\text{base } 2) = 1 * 2^2 + 0 * 2^1 + 1 * 2^0 + 1 * 2^{-1} = (5.5)(\text{base } 10)$ .

## Numeri interi.

I numeri interi sono un insieme discreto illimitato. Per poter essere codificati devono essere limitati. In genere ci si restringe ad un sottoinsieme simmetrico rispetto allo 0. Avendo a disposizione  $n$  bit, se ne usa uno per rappresentare il segno e i restanti  $n-1$  per rappresentare il modulo. Se  $a = n-1$  (il numero dei bit a disposizione meno uno) il massimo numero rappresentabile è (in modulo)  $2^{a-1}$ . Se il risultato di un'operazione superiore a  $2^{a-1}$  o inferiore a  $-2^{a-1}$  non

può essere codificato e il calcolatore restituisce un messaggio di **overflow**.

## Numeri reali.

I numeri reali sono un insieme continuo e illimitato. Per poterli rappresentare occorre limitarli (in modo simmetrico rispetto allo 0) e partizionarli. Esistono due tipi di rappresentazione dei numeri reali:

rappresentazione in virgola fissa e rappresentazione in virgola mobile o *floating-point*.

### Rappresentazione in virgola fissa.

Nella rappresentazione a virgola fissa un bit viene serve a indicare il segno (+ o -) un certo numero di bit (diciamo *a*) vengono dedicati a rappresentare le cifra prima della virgola e i rimanenti (diciamo *b*) le cifre dopo la virgola. Più *a* sarà grande più potremo rappresentare numeri positivi grandi e numeri negativi piccoli, più *b* sarà grande più aumenterà la precisione. Il difetto della rappresentazione dei numeri reali in virgola fissa è la rigidità.

### Rappresentazione in virgola mobile (floating-point).

Il numero da rappresentare viene espresso nella forma

$$S \ 0.M \ B^{sE}$$

dove *s* rappresenta il segno, *M* la mantissa, *E* l'esponente, se il segno dell'esponente e *B* la base di numerazione. Non tutti i numeri hanno rappresentazione univoca. Questo comporta perdita di efficienza e difficoltà di confronto. Quando esistono rappresentazioni equivalenti dello stesso numero, tra queste si definisce normale quella con l'esponente più piccolo. Da una rappresentazione non normale a quella

normale equivalente si passa traslando verso sinistra le cifre della mantissa e sottraendo 1 all'esponente.

## Codifica delle immagini

Le immagini sono informazioni continue in tre dimensioni: due spaziali ed una colorimetrica. Per codificarle occorre operare tre discretizzazioni. Le due discretizzazioni spaziali riducono l'immagine ad una matrice di punti colorati, detti pixel. La terza discretizzazione limita l'insieme di colori che ogni pixel può assumere.

### Immagini bitmap

Il modo più elementare per rappresentare un'immagine consiste nel rappresentare il colore di ogni pixel. Il numero di bit necessari a rappresentare ogni pixel dipende dal numero di diversi colori disponibili. Un byte consente di rappresentare 256 colori. Un'immagine a 256 colori è una sequenza di byte.

Un'immagine di 100X100 pixel a 256 colori richiede 10000 byte (10 Kb) per essere rappresentata. Un'immagine a due soli colori (ad esempio bianco e nero) richiede un solo bit per pixel. Le immagini codificate secondo questa tecnica vengono dette anche immagini raster o immagini bitmap. Questo tipo di immagini vengono solitamente usate per fotografie, scansioni e in genere per tutte le immagini che prevedono sottili gradazioni nei colori e nelle forme.

Le immagini raster sono caratterizzate da tre informazioni principali che ne consentono la corretta interpretazione: dimensioni, numero dei colori, risoluzione.

Le dimensioni consentono di ricostruire correttamente l'immagine collocando i pixel al loro posto.

Il numero dei colori è definito dal numero di bit che viene dedicato alla descrizione del colore di ogni pixel:

- **immagini monocromatiche:** un bit per ogni pixel (se il bit è 0 è bianco, se no nero)
- **immagini a 256 colori:** 8 bit (un byte) per ogni pixel. A ogni immagine è associata una tabella di 256 colori (detta tavolozza o palette) in cui sono descritti i contenuti **RGB** (Red, Green, Blue) di ognuno dei 256 colori usati. In altre parole i 256 sono numerati da 0 a 255 e l'elaboratore, quando decodifica l'immagine controlla a che colore corrisponde il numero che descrive ogni pixel. In questa maniera si possono avere in ogni immagine al massimo solo 256 colori diversi ma scelti
- **immagini a 16 bit:** vengono riservati 16 bit (2 byte) per descrivere il colore di ogni pixel. Ogni colore è descritto dalle sue componenti **RGB** (Red, Green, Blue). Vengono messi a disposizione 5 bit per ogni componente (l'ultimo bit non è usato).  
Avremo cioè colori descritti con  $2^5$  (cioè 32) livelli di Rosso, Verde e blue per un totale di 32.768 colori.
- **immagini a 24 bit:** vengono riservati 24 bit (3 byte) per descrivere il colore di ogni pixel. Ogni colore è descritto dalle sue componenti **RGB** (Red, Green, Blue). Vengono messi a disposizione 8 bit (un byte) per ogni componente. Avremo cioè colori descritti con  $2^8$  (cioè 256) livelli di Rosso, Verde e blue per un totale di 16.777.216 colori.
- **immagini a 32 bit:** vengono riservati 32 bit (4 byte) per descrivere il colore di ogni pixel. Come nelle

immagini a 24 bit vengono messi a disposizione 8 bit (un byte) per ogni componente. Il quarto byte descrive 256 livelli di trasparenza (il cosiddetto canale alpha) che può avere il pixel.

Infine la risoluzione definisce il rapporto che c'è tra la dimensione dell'immagine in pixel e la dimensione reale che l'immagine ha quando viene resa da un dispositivo (stampante, fax, schermo, ecc). La risoluzione di misura normalmente in **punti (pixel) per pollice** o in **punti per centimetro**.

## Immagini vettoriali

La grafica vettoriale scompone l'immagine in gruppi logici di componenti (linee, cerchi, rettangoli, ecc.) e memorizza le forme in termini di coordinate e colori dei vari elementi geometrici che le compongono. In definitiva un file che memorizza un'immagine di tipo vettoriale è costituito da un'insieme di coordinate e dalle informazioni sul colore di ciascuna forma. Nel momento della visualizzazione i punti descritti dalle coordinate vengono disegnati e colorati a dovere. La grafica vettoriale viene comunemente usata nel disegno animato, nella grafica lineare e negli strumenti CAD.

Un grande vantaggio delle immagini vettoriali rispetto a quelle raster/bitmap sta nella minor dimensione del file che le memorizza e nella capacità di essere variate di dimensioni senza subire alcuna distorsione. Le immagini bitmap, invece, se ridimensionate rispetto alle dimensioni originali di acquisizione, hanno la tendenza a perdere di risoluzione risultando distorte o sfocate.



Filmati video (immagini in movimento). I filmati video possono essere considerati sequenze di fotogrammi. Abbiamo già visto come codificare immagini, e quindi sappiamo come codificare ogni fotogramma. Il numero di fotogrammi nell'unità di tempo determina la qualità del movimento. E' stato sperimentato che l'occhio umano non percepisce discontinuità nel movimento al di sotto del venticinquesimo di secondo. Pertanto, una frequenza di riproduzione dei fotogrammi superiore a 25Hz sarebbe superflua se lo scopo fosse la riproduzione di un video. D'altro canto potrebbe non esserlo se servisse a documentare un esperimento di brevissima durata o se il video dovesse essere rivisto al rallentatore.

Es: Filmato di 10 minuti, con risoluzione di 100x100 pixel a 256 colori: dimensione complessiva di  $600 \times 25 \times 100 \times 100 \times 8 = 1.2\text{Gbit}$ .

## Codifica di segnali analogici

Un segnale analogico è una grandezza fisica che varia nel tempo e che può assumere un insieme continuo (e quindi infinito) di valori. Un segnale digitale invece può assumere solo un insieme discreto (e generalmente finito) di valori. Un segnale analogico ha anche la proprietà di essere **tempo-continuo**: cioè il suo valore è significativo (e può variare) in qualsiasi istante di tempo.

Diversamente, un segnale digitale viene anche detto segnale **tempo-discreto**: poiché il suo valore ha interesse solo in istanti di tempo prestabiliti, generalmente equidistanziati, mentre non ha alcun valore tra un istante ed un altro.

La rappresentazione più naturale di un segnale  $s$  è la sequenza dei suoi valori istantanei  $s(t)$ . Ci poniamo prima il problema di rappresentare un valore istantaneo, e poi quello di rappresentarne la sequenza.

Il valore istantaneo di un segnale può essere rappresentato da un numero, e come tale può essere codificato utilizzando notazione in virgola fissa o in virgola mobile. Tali codifiche inducono limitazioni e partizionamenti sullo spazio dei possibili valori che il segnale può assumere (quantizzazione del segnale).

Una volta codificato il valore istantaneo del segnale si ottiene la sequenza temporale ripetendo la codifica ad intervalli prestabiliti. In generale, se il segnale da codificare era **tempodiscreto**, la scelta naturale è quella di rappresentare il valore del segnale in tutti e soli gli istanti in cui esso è significativo. Per la codifica dei segnali analogici, che come abbiamo detto sono

**tempocontinui** occorre operare una discretizzazione dell' intervallo temporale per ottenere una rappresentazione finita. L'operazione di discretizzazione più comune è il campionamento, che consiste nell'osservare il segnale tempo-continuo ad istanti di tempo prefissati (generalmente equidistanziati) in modo da ottenere un segnale **tempodiscreto** che assume lo stesso valore di quello originale nei punti di campionamento.

In definitiva, le operazioni di campionamento e quantizzazione trasformano un segnale analogico **tempocontinuo** in un segnale digitale **tempodiscreto**.

## Codifica del suono

Il suono è un segnale analogico tempo-continuo. Ci sono diversi modi per descrivere la natura di tale segnale. Per fissare le idee pensiamo che il suono prodotto da un altoparlante è prodotto dalla vibrazione di una membrana. Descrivendo la posizione della membrana nel tempo (e quindi il suo spostamento) a tutti gli effetti descriviamo il suono. Assumiamo allora che il segnale analogico  $s(t)$  esprima la posizione della membrana (rispetto ad un riferimento fissato) all'istante di tempo  $t$ . Tale segnale deve essere campionato e discretizzato per poter essere codificato in termini di bit. Per valutare la qualità della codifica possiamo pensare di usare il segnale quantizzato e **tempo-discreto** (quello cioè effettivamente rappresentato dalla nostra codifica) per muovere la membrana di un altoparlante. Se il campionamento è troppo rado e vengono usati pochi bit per codificare ogni valore istantaneo, la perdita di informazione degrada la qualità del suono (il suono riprodotto è sensibilmente diverso da quello originale).

Per quanto riguarda il campionamento, è dimostrato che il suono percepibile dall'orecchio umano viene riprodotto fedelmente se la frequenza di campionamento (il numero di campioni in un secondo) è non inferiore a 30KHz. Per rendere intelligibile il parlato è sufficiente una frequenza di campionamento di 8KHz. La quantizzazione introduce comunque una distorsione. Questa tecnica di codifica del suono è quella comunemente usata nella codifica di tipo wave che troviamo nei file .wav. Questa codifica tenta di ricostruire fedelmente la forma d'onda del suono

campionandola adeguatamente. Il file che si ottiene è normalmente di grandi dimensioni.

Es: Telefono. La comunicazione telefonica deve riprodurre la voce umana ai due estremi del collegamento garantendo la comprensione del parlato. A tal fine lo standard prevede un campionamento a 8KHz ed una quantizzazione a 256 livelli (codificati con 8 bit). Il segnale digitale che ne risulta usa 64Kbit per codificare ogni secondo di conversazione. 10 minuti di conversazione richiedono 38.4Mbit.

La codifica di tipo MIDI, invece ha tra i formati audio lo stesso ruolo che il formato vettoriale ha tra i formati grafici. Infatti, il formato MIDI acronimo di Musical Instrument Digital Interface, non è una registrazione sonora ma contiene dei comandi da impartire ad un sintetizzatore sonoro (software o hardware) che riproduce il suono di uno strumento musicale. I file MIDI per questo motivo sono molto piccoli ed hanno il formato del tipo:

suona la nota x per un tempo y con lo strumento z.
--

La codifica MIDI, quindi riproduce un suono in maniera sintetica. Le nuove suonerie polifoniche dei telefoni cellulari di ultima generazione si basano su una codifica di tipo MIDI.

# La compressione dei dati

Spesso risulta utile ridurre le dimensioni dei dati su si sta lavorando in modo da renderne più agevole l'archiviazione e la trasmissione. Una condizione tipica in cui viene applicata la compressione sui dati è la ridondanza. In poche parole, dei dati che contengono le stesse informazioni in qualche modo ripetute (ridondanza) sono facilmente comprimibili. Ad esempio, quando ci si trova di fronte a un testo scritto si incorre in esempi di ridondanza lampanti. Si consideri, per esempio, l'uso che la lingua italiana fa della lettera 'q' che è sempre seguita da una 'u': nessuno avrebbe difficoltà nel capire il significato della parola 'q\*adro"', nonostante essa non sia scritta correttamente. Quindi, una prima tecnica di compressione della lingua italiana potrebbe essere basata sull'omissione del carattere 'u' tutte le volte che questo è preceduto da un carattere 'q'.

La presenza di ridondanza in una lingua, in un testo scritto, o in una qualunque stringa di simboli non è però nefasta, anzi, essa svolge il compito fondamentale di rendere **robusto** il messaggio contenuto in quella stringa. Per robusto si intende facilmente comprensibile e non incline ad essere male interpretato anche nel caso in cui il messaggio venga trasmesso in modo solo parziale. Errori di trasmissione su codici ridondanti sono molto più facilmente correggibili.

D'altra parte, un testo che sia stato compresso al massimo è molto fragile, in quanto, per definizione di massima compressione, ciascun simbolo sarà portatore di informazione, e quindi una sua alterazione porterà ad una perdita irrimediabile di quella stessa informazione.

In definitiva, l'uso di codifiche ridondanti può avere due motivazioni: la **flessibilità** e l'**affidabilità**.

- La **flessibilità** indica la possibilità di utilizzare la stessa codifica in situazioni diverse. Es: il cosiddetto 'baco del millennio" (millenium bug) che costa al mondo migliaia di miliardi è dovuto alla scarsa lungimiranza con cui i produttori di software hanno deciso di utilizzare due sole cifre decimali per rappresentare l'anno nelle date, dando per scontate (e quindi prive di contenuto informativo) le prime 2, fino ad ora sempre uguali a 19.
- L'**affidabilità** deriva dalla capacità di alcune codifiche ridondanti di rivelare o correggere errori. Un codice non ridondante associa univocamente una configurazione ad un significato. Un errore nella configurazione comporta un errore di interpretazione del significato. Una codifica ridondante a rivelazione d'errore associa significati utili solo ad un sottoinsieme delle configurazioni possibili. Se un errore trasforma una configurazione significativa in una non significativa, chi la interpreta riconosce la presenza dell'errore. Una codifica ridondante a correzione d'errore associa molte configurazioni allo stesso significato. Se un errore trasforma

una configurazione in una a cui è associato lo stesso significato, l'interpretazione resta corretta.

Codifiche ridondanti si prestano a compressione, come dimostrano gli esempi precedenti, ma esistono anche tecniche di compressione che agiscono su dati irridondanti sfruttando proprietà tipiche dell'informazione da rappresentare.

## Tecniche generiche di compressione

Per quanto riguarda le tecniche di compressione una prima distinzione che bisogna fare è tra compressione **lossy** e **lossless**. La prima tecnica effettua una compressione dei dati definita "con perdita d'informazione" o anche distruttiva in quanto, una volta applicata questa tecnica, non è più possibile ricostruire in maniera esatta i dati di partenza attraverso il processo di decompressione.

In definitiva, c'è stata, una perdita irrimediabile di informazione.

Le tecniche di tipo **lossy** sono legate a specifiche codifiche delle informazioni (immagini, suoni, filmati, ecc.) e sfruttano le loro specifiche caratteristiche. In questi casi perdita di informazione significa perdita di qualità.

Questa tecnica è molto utilizzata nel campo della compressione dei formati multimediali soprattutto laddove è spesso inutile ricostruire, come nel caso delle fotografie, l'informazione iniziale con una risoluzione maggiore di quella che l'occhio umano può apprezzare o di quella che uno specifico mezzo (ad esempio il video) può restituire.



Le tecniche di compressione di tipo **lossless** permettono invece di recuperare interamente l'informazione contenuta nel documento codificato prima della sua compressione, ma la loro efficienza di compressione è minore. Queste tecniche sono comunque le uniche utilizzabili in quei casi in cui non è possibile accettare neppure la minima perdita delle informazioni. Ad esempio la tecnica di compressione che sta alla base dei compressori tipo Winzip o Winrar è una tecnica di tipo **lossless** in quanto non sarebbe accettabile perdere dell'informazione comprimendo ad esempio un file di installazione di un programma o un file di testo.

## Tecniche di tipo lossless

### Compressione run-length

La compressione di tipo run-length fa parte delle tecniche di compressione di tipo lossless cioè senza perdita di dati. Essa si basa su un algoritmo denominato di *Run Length Encoding* (RLE) che è uno dei più semplici algoritmi di compressione mai progettati. Si basa sul fatto che nei dati da comprimere esistono sequenze, definite run, che si ripetono costantemente. Una volta individuate le sequenze ripetute, vengono sostituite da un unico simbolo e dal numero delle ripetizioni presenti.

Ad esempio data la stringa di bit 01110000 l'algoritmo RLE la comprimerebbe codificandola in 03\*14\*0 che sta ad indicare '0 tre volte 1 quattro volte 0' .

Esistono numerose varianti di RLE le cui principali differenze consistono nella lunghezza minima da attribuire ad un run. Nell' esempio appena descritto abbiamo usato una lunghezza del run pari a uno cioè si prendeva in considerazione un carattere alla volta e si cercava se questo veniva

ripetuto consecutivamente. Allo stesso modo si possono prendere in considerazione gruppi di simboli e cercare se l' intero gruppo viene ripetuto all' interno della stringa.

### Compressione a codifica differenziale

In alcuni casi, le informazioni sono costituite da blocchi di dati, ognuno dei quali differisce leggermente dal precedente come, ad esempio, i fotogrammi successivi di un filmato. In questo caso sono utili le tecniche di

compressione che utilizzano la codifica differenziale. L'approccio di queste tecniche è quello di memorizzare non il blocco stesso ma le sue differenze rispetto al precedente. È evidente che in questo caso se si dovesse corrompere un blocco compresso durante la trasmissione, risulterebbe compromessa la ricostruzione/decompressione di tutti i blocchi successivi.

## Codifiche basate su dizionari

Il termine dizionario si riferisce all'insieme di elementi di base sui quali viene ricostruito il messaggio compresso. In pratica viene utilizzato un insieme di simboli (dizionario) per codificare un messaggio. I simboli del dizionario rappresentano particolari sequenze di bit e durante la compressione, ad ogni sequenza riconosciuta viene sostituito il simbolo corrispondente. La particolarità di queste tecniche sta nel fatto che il dizionario viene creato dinamicamente durante il processo di compressione. Tale tecnica sta alla base dell'algoritmo di compressione Lempel-Ziv che troviamo nella maggior parte dei tools di compressione quali WinZip.

## Compressione delle immagini

### Formato GIF

La sigla GIF è acronimo di Graphic Interchange Format. Questo tipo di compressione rientra nelle tecniche di tipo lossless, cioè senza perdita di dati. La caratteristica del formato GIF è che esso può esportare solo immagini che contengono al massimo 256 colori. Se l'originale contiene un numero più elevato di colori quindi, è necessario effettuare una riduzione e la perdita di qualità sarà significativa. Il formato GIF usa colori a 8 bit ed è efficace per comprimere immagini vettoriali, geometriche o testo. Questo formato fu diffuso negli anni Ottanta come metodo efficiente di trasmissione delle immagini su reti di dati. All'inizio degli anni Novanta i progettisti originali del web lo adottarono per l'efficienza che offriva. Oggi la stragrande maggioranza delle immagini sul web è in questo formato ed è supportato da tutti i browser web.

Il formato GIF usa una forma di compressione LZW che mantiene inalterata la qualità dell'immagine, ovvero riduce le dimensioni del file senza pregiudicare la qualità grafica dell'immagine. Questo formato è basato sull'uso di una tavolozza di colori: anche se il singolo colore può essere uno fra milioni di sfumature, solo un certo numero di essi è disponibile (al massimo  $256=8\text{bit}$ ). I colori sono memorizzati in una 'tavolozza', una tabella che associa un numero ad un certo valore di colore.

La limitazione a 256 colori appariva ragionevole all'epoca della creazione del formato GIF perché non erano ancora diffusi dispositivi in grado di visualizzarne un numero superiore. Per disegni al

tratto, fumetti, fotografie in bianco e nero sono di regola sufficienti 256 colori. Minore sarà il numero di colori presenti nell'immagine e maggiori saranno le possibilità di compressione, ovvero minori saranno le dimensioni del file in quanto sarà ridotta la dimensione della tavolozza e quindi il numero di codici per descrivere i colori.

Il formato GIF consente anche di salvare le immagini in un formato interlacciato. Il formato a interlacciamento produce una visualizzazione graduale di un'immagine in una serie di passate sempre più definite a mano a mano che i dati arrivano al browser. Ogni nuovo passo crea un'immagine più nitida fino al completamento dell'intera immagine.

Il formato GIF consente anche di definire un colore come trasparente. Nelle aree di colore contrassegnato come trasparente, verrà visualizzato il colore di sfondo. Questa proprietà viene utilizzata per le animazioni e per la sovrapposizione di più immagini.

Il formato GIF, infine prevede una serie di semplici comandi con cui si può stabilire la frequenza con cui il fotogrammi di una animazione sono visualizzati, se l'animazione è continua o viene ripetuta un numero finito di volte, se le immagine visualizzate vengono sovrapposte a quelle precedenti o lo sfondo viene cancellato. In altre parole GIF è anche un formato programmabile.

Con l'evoluzione dei personal computer (maggiore risoluzione dei video, maggior numero di colori riproducibili), oggi si usano principalmente immagini a 16, 24 o 32 bit. Ciò nonostante, anche se in misura minore, il formato GIF è ancora molto utilizzato sia in

quanto formato che consente animazioni programmate, sia perché, con determinate immagini risulta più efficiente del formato JPEG (oggi sicuramente il formato di compressione più utilizzato). In realtà oggi utilizzare il formato GIF significa perdita di informazione. Il processo di conversione avviene di fatto in due fasi: una prima fase (**lossy**) in cui l'immagine originale (normalmente a 24 o 32 bit) viene ridotta a 256 colori creando una palette ottimizzata per rappresentarla al meglio e una seconda fase (**lossless**) in cui all'immagine viene applicata la compressione GIF.

Molto spesso il formato GIF risulta particolarmente efficiente quando si convertono in **bitmap** disegni realizzati con programmi per disegno vettoriale con limitato numero di colori e uso di colori pieni.

## Formato JPEG

Un formato grafico utilizzato più frequentemente sul Web per ridurre le dimensioni dei file grafici è lo schema di compressione JPEG (*Join Photographic Expert Group*). A differenza delle immagini GIF, le immagini JPEG sono policrome (24 bit, o 16,8 milioni di colori). Questo tipo di immagini ha generato un altissimo interesse tra fotografi, artisti, progettisti grafici, specialisti della composizione di immagini mediche, storici dell'arte e altri gruppi per i quali la qualità dell'immagine è d'importanza fondamentale e per i quali non è possibile accettare compromessi sulla fedeltà dei colori tramite retinatura di un'immagine a colori a 8 bit.

Una forma più recente di JPEG, chiamata JPEG **progressivo**, conferisce alle immagini JPEG la stessa

gradualità di visualizzazione delle immagini GIF interlacciate; al pari di queste ultime, le immagini JPEG progressive impiegano spesso un tempo maggiore per lo scaricamento sulla pagina rispetto ai JPEG standard, ma offrono al lettore un'anteprima più rapida.

La compressione JPEG utilizza una sofisticata tecnica matematica, chiamata trasformazione discreta del coseno, per produrre una scala scorrevole di compressione delle immagini. Tale tecnica si basa su di una codifica dell'immagine **percettiva** in cui viene distinta la luminosità dei pixel dal loro colore. Il motivo di tale distinzione è che l'occhio umano è più sensibile alle variazioni di luminosità che non a quelle di colore. Lo standard base di JPEG trae vantaggio da questo fenomeno codificando ogni componente della luminosità, ma dividendo l'immagine in blocchi di quattro pixel e registrando solo il colore medio di ogni blocco. La rappresentazione finale preserva dunque i cambiamenti di luminosità, attenuando i repentini mutamenti cromatici dell'immagine originale. Il vantaggio è che ogni blocco di quattro pixel è rappresentato soltanto da 6 valori (4 di luminosità e 2 di colore) anziché 12 valori che sarebbero necessari in un sistema a 3 valori per pixel (immagini a 24 bit RGB).

È possibile scegliere il grado di compressione che si desidera applicare a un'immagine in formato JPEG, ma in questo modo si determina anche la qualità dell'immagine. Più si comprime un'immagine con la compressione JPEG, più si riduce la qualità dell'immagine stessa.

## Formato PNG

Il formato PNG (Portable Network Graphic) è stato sviluppato appositamente per il Web.

Questo formato è stato disponibile fin dal 1995 ma ha stentato ad acquisire popolarità a causa della mancanza di un supporto generalizzato da parte dei browser. Si tratta di un formato che secondo le intenzioni degli autori doveva sostituire il formato GIF. Questo formato senza perdita di informazioni comprime le immagini a 8 bit producendo file di dimensioni inferiori rispetto a GIF ma supporta anche immagini a 16 e 24 bit.

Anche se il formato PNG supporta il colore a 24 bit, la sua routine di compressione senza perdita di informazioni non è in grado di raggiungere l'efficienza del formato JPEG. Il formato PNG supporta, inoltre, le funzionalità di trasparenza e interallacciamento ma non l'animazione.

Un'utile caratteristica del formato PNG è la capacità di incorporare del testo per offrire la possibilità di eseguire ricerche sulle immagini; è infatti possibile memorizzare nel file dell'immagine una stringa che identifica l'immagine stessa. Purtroppo il formato grafico PNG non è ampiamente supportato e l'implementazione corrente delle immagini PNG in Netscape Navigator e Microsoft Internet Explorer non supporta completamente tutte le sue funzioni.



## Compressione audio

Come nel caso delle immagini si tratta di distinguere tecniche di compressione senza perdita (lossless) e con perdita (lossy).

### Compressione senza perdita

Usando un algoritmo di compressione senza perdita, dal risultato della compressione si può riottenere tutta l'informazione originaria. In questo caso la riduzione massima generalmente ottenibile, utilizzando algoritmi studiati appositamente per l'audio è all'incirca del 60%, ma solo con alcuni tipi di suono. Si possono utilizzare gli stessi algoritmi generali di compressione (come LZW) ma i risultati in termine di riduzione sono inferiori.

Un buon esempio di algoritmo di compressione lossless è il **FLAC** (Free Lossless Audio Codec). **FLAC** è un diffuso codec audio libero di tipo lossless, cioè senza perdita di qualità. La compressione non rimuove informazioni dal flusso audio, ed è quindi adatto sia all'ascolto normale che per l'archiviazione. Il formato **FLAC** attualmente ha un buon supporto da parte di vari software audio.

Altri esempi di compressione lossless sono **APE**, **ALF**.

### Compressione con perdita

Gli studi di psicoacustica hanno permesso di accertare che l'uomo non è sensibile nello stesso modo a tutte le frequenze e che un suono ad alta intensità ne maschera uno con frequenza vicina ma intensità più bassa. Sfruttando queste ed altre considerazioni, si può pensare di eliminare l'informazione che non

verrebbe comunque percepita ed ottenere quindi un buon rapporto di compressione.

In questo modo sono stati sviluppati algoritmi di compressioni specifici per l'audio ad altissima efficienza che sono in grado di ottenere riduzione della lunghezza dei file dell'ordine di 10 a 1 praticamente senza perdita di qualità.

Facciamo alcuni esempi.

### **MP3**

**MP3** (o, più esattamente "MPEG-1/2 Audio Layer 3") è un algoritmo di compressione audio in grado di ridurre drasticamente la quantità di dati richiesti per riprodurre un suono, rimanendo comunque una riproduzione fedele del file originale non compresso. È lo standard de facto della compressione audio.

La qualità di un file MP3 dipende dalla qualità della codifica e dalla difficoltà con il quale il segnale deve essere codificato. Buoni codificatori hanno una qualità accettabili da 128 a 160 kbit/s, la chiarezza perfetta di un brano si ottiene da 160 a 192 kbit/s. Un codificatore che ha bassa qualità lo si riconosce ascoltando persino un brano a 320 kbit/s. Per questo non ha senso parlare qualità di ascolto di un brano di 128 kbit/s o 192 kbit/s. Una buona codifica MP3 a 128 kbit/s prodotta da un buon codificatore produce un suono migliore di un file MP3 a 192 kbit/s codificato con uno scarso codificatore.

Oggi esistono buone alternative a MP3 anche se molto meno diffuse.

## Vobis

**VORBIS** è un algoritmo di compressione orientato alla compressione mono, stereo o 5.1 surround di segnale audio PCM campionato a 44.1 kHz o 48 kHz, con una profondità di campionamento di 16 bit o 32 bit. È comunque in grado di gestire anche segnali in ingresso differenti da quelli raccomandati.

Nel trattamento di segnale stereo musicale Vorbis ha il suo bit rate ideale intorno ai 128 kbit/s, risultando estremamente difficoltoso da distinguere rispetto all'originale in un ascolto cieco già da 192 kbit/s.

Trattandosi di un algoritmo di compressione lossy, cioè a perdita di informazioni, è l'encoder a svolgere il compito più delicato in assoluto, dovendo scegliere *quale* parte di informazione acustica sacrificare.

Vorbis è un algoritmo dall'approccio pesantemente VBR, ovvero a bit rate estremamente variabile in base al tipo di segnale sonoro che è chiamato a codificare. Per questo motivo al posto di riferirsi al valore di kbit/s Xiph.Org raccomanda di usare la nomenclatura  $q$ , ovvero il livello di qualità con cui è stata eseguita la codifica.

Tra i pregi che comunemente si attribuiscono a Vorbis, soprattutto riferendosi all'inevitabile paragone con lo standard de-facto MP3, vanno ricordati la maggior estensione e pulizia delle alte frequenze (sopra i 16 kHz), il supporto multicanale a livello nativo e in generale una migliore conservazione delle microinformazioni di spazialità sonora del segnale originario.

Tra i difetti attribuiti vanno citati la relativa pesantezza dell'algoritmo di decodifica rispetto al collaudato MP3, e soprattutto una certa tendenza al pre-echo, ovvero un'innaturale *fantasma sonoro* che sembra precedere di alcuni brevi istanti ogni brusco aumento di pressione sonora. L'esempio che tipicamente viene portato è quello di una sonata di pianoforte con attacchi di *fortissimo* dal silenzio, oppure il suono delle nacchere.

### Advanced Audio Coding

Il formato **Advanced Audio Coding (AAC)** è un formato di compressione audio creato dal consorzio MPEG e incluso ufficialmente nell'MPEG-4. L'AAC fornisce una qualità audio superiore al formato MP3 con una codifica più compatta. Attualmente viene utilizzato principalmente da Apple nei suoi prodotti dedicati all'audio, difatti Apple usa una variante dell'AAC che gestisce i diritti d'autore per vendere musica attraverso il proprio negozio di musica on-line iTunes Music Store. Una compressione a 128Kbps, lo standard di iTunes Music Store corrisponde a circa 160 kbps di un mp3 a bitrate variabile.

Vari studi ed esperimenti, hanno mostrato come un AAC a 128Kbps, a differenza del formato mp3, sia pressoché identica a quella del cd originale.

Altri algoritmi di compressione lossy sono: Windows Media Audio (**WMA**), molto diffuso sui sistemi Windows; Dolby Digital (**AC3**) che può comprimere fino a 6 canali audio, di cui 5 a piena larghezza di banda ed uno per gli effetti a bassa frequenza (LFE), fino a 384 kbit/s. Viene utilizzato nei DVD e nel sistema americano ATSC DTV, MPC Musepack è un formato

opensource con una qualità maggiore dell'mp3 a parità di bitrate a scapito però delle dimensioni finali del file.

## Nota sul bitrate

I file multimediali sono per loro natura connessi al tempo che scorre. In altri termini ad ogni secondo è associato un certo contenuto informativo e quindi una certa sottosequenza di cifre binarie. Il numero di cifre binarie che compongono queste sottosequenze è detto bitrate. In altre parole il bitrate è il numero di cifre binarie impiegate per immagazzinare un secondo di informazione. Questo può essere costante per tutta la durata del file o variare all'interno di esso. Ad esempio i cd musicali vengono campionati (registrati) ad una frequenza pari a 44.100Hz. Da ciò si evince che ogni secondo si hanno 44.100 valori registrati dall'ipotetico microfono che vanno poi moltiplicati per i 2 canali del suono stereo che vanno a loro volta moltiplicati per 2 poiché la registrazione avviene a 16 bit (pari appunto a 2 byte). Quindi avremo:

$$44.100 \times 2 \times 2 \times 60 \text{ (secondi)} = \sim 10 \text{ MB ogni minuto}$$

La compressione, diminuendo la lunghezza globale del file, diminuirà di conseguenza la lunghezza media delle sottosequenze ossia diminuirà il bitrate medio. Il bitrate medio diventa dunque in questi casi l'indice dell'entità della compressione. Ad esempio se il file di origine possedesse un bitrate di 1411 Kbit/s e il file compresso possedesse un bitrate medio di 320 Kbit/s, allora avremmo ridotto di un fattore pari a circa 4.5.

## Compressione dei filmati

La codifica digitale e la compressione dei video è affidata ai cosiddetti codec video. Un **CODEC VIDEO** è un

programma o un dispositivo sviluppato per descrivere un flusso video sotto forma di dati numerici adatti ad essere memorizzati su un supporto digitale.

Usualmente i codec video effettuano anche una compressione dei dati in modo da ridurre l'elevata quantità di dati che compone un flusso video. La maggior parte dei codec video adottano tecniche di compressioni lossy (a perdita di informazioni) in modo da poter ridurre i dati necessari per trasmettere i flussi video anche di 20 volte o più, ma esistono anche dei codec utilizzati per applicazioni professionali che utilizzano compressioni lossless (senza perdita di informazione).

A seconda della diversa tecnica di codifica del flusso video, i codec video si dividono in due grandi famiglie: a codifica **intraframe** e a codifica **interframe**.

La codifica **intraframe** contraddistingue i codec che codificano e decodificano un flusso video descrivendo ogni singolo fotogramma che compone la sequenza video, rispettando quindi un approccio tradizionale alla quantizzazione video come sequenza di immagini statiche.

Nella codifica **interframe** invece i codec video si occupano di descrivere i cambiamenti che occorrono tra un fotogramma ed il successivo partendo da un fotogramma iniziale descritto con codifica **intraframe** e seguendo un approccio più innovativo alla quantizzazione video allo scopo di migliorarne l'efficienza sfruttando la capacità dei sistemi di riproduzione moderni in grado di elaborare l'informazione per poi mostrarne il risultato.

La diversa dinamica dei due approcci fa sì che la codifica intraframe è più adatta alla riproduzione di sequenze video particolarmente movimentate, descrivendo ogni singolo fotogramma infatti, un codec a codifica intraframe potrà degradare la qualità delle singole immagini all'aumentare del rapporto di compressione, ma tenderà comunque a lasciare inalterata la dinamica del movimento. I codec a codifica interframe risultano invece meno adatti alla codifica di sequenze movimentate per le quali necessitano di descrivere grossi cambiamenti tra i fotogrammi. Al contrario, in sequenze video statiche, ovvero con pochi elementi che cambiano nella scena, la codifica interframe risulta di notevole efficienza.

Ecco una breve panoramica dei principali standard:

- Il **DivX®** è un formato di compressione video sviluppato da DivX Inc. Attraverso l'apposito codec è possibile riprodurre e creare file video di questo formato. La particolarità del DivX sta nella sua versatilità nel produrre file di dimensioni ridotte di filmati di lunga durata, lasciando pressoché inalterata la qualità dell'immagine. In pratica, con le opportune impostazioni, è possibile convertire un film DVD di 6-8 Gigabyte in un file DivX di 700Mb (la dimensione di un cd rom) con una qualità video e audio più che discreta. Per questo motivo, è stato al centro di controversie per il suo utilizzo nella duplicazione e distribuzione di DVD protetti. Le prime versioni di DivX (fino alla 3) erano illegali in quanto n quanto prodotto da una copia rubata del Mpeg-4 di proprietà Microsoft. Dalla versione 4 DivX è diventato un codec indipendente dal suo

predecessore ed iniziata la sua vita anche commerciale. Oggi DivX è arrivato alla versione 6.

- **H.264** Questo codec video è stato sviluppato per video ad alta qualità anche a frequenze di trasmissione dei dati inferiori rispetto alle soluzioni attuali, ed è utilizzata per qualunque tipo di periferica: dai televisori ad alta definizione HDTV e DVD, ai telefoni cellulari 3G. I servizi di broadcast basati sullo standard H.264 occupano una banda inferiore rispetto al diffuso schema di codifica MPEG-2, a una frequenza di trasmissione dei bit decisamente inferiore. Gli operatori di broadcasting possono quindi trasmettere in modo economico un numero maggiore di programmi ad alta definizione. L'efficienza della compressione è migliorata di oltre il 50% rispetto al precedente MPEG-2. Attualmente i dispositivi con maggior diffusione ad utilizzare questo sistema di codifica sono l'iPod video e la console Sony PSP.
- **MPEG-1** è uno standard introdotto nel 1991 da MPEG (Moving Pictures Experts Group). Originariamente è stato ottimizzato per le applicazioni video a basso *bitrate* con una risoluzione video di 352x240 pixel con 30 fotogrammi al secondo per lo standard tv NTSC oppure di 352x288 pixel con 25 fotogrammi al secondo per lo standard tv PAL. MPEG-1 non è strutturalmente limitato a questi formati in quanto può raggiungere ad esempio i 4095x4095 pixel con 60 fotogrammi al secondo, ma di fatto il sistema è stato ottimizzato per un bit rate di 1,5 Mbit/s. I Video CD utilizzano il formato MPEG-1.



La qualità dell'output ai *bitrate* tipici di un Video CD è quella di un VCR.

- **MPEG-2** è un sistema di codifica digitale di immagini in movimento, che permette di comprimere i dati mantenendo una buona qualità. MPEG-2 è stato destinato al broadcast televisivo, fin dalla sua introduzione nel 1994. Una efficiente codifica per il video interlacciato e la scalabilità sono state le caratteristiche che hanno permesso di digitalizzare efficacemente i segnali televisivi. Grazie all'MPEG-2 si ottengono immagini televisive di buona qualità con bitrate compresi tra 4 e 9 Mbit/s. **MPEG-2** sta anche alla base dello standard **DVD**.
- **MPEG-4**, presentato nel 1998, è il nome dato a un'insieme di standard per la codifica dell'audio e del video digitale sviluppati dall'ISO/IEC Moving Picture Experts Group (MPEG). L'**MPEG-4** è uno standard utilizzato principalmente per applicazioni come la videotelefonia e la televisione digitale, per la trasmissione di filmati via Web, e per la memorizzazione su supporti CD-ROM. MPEG-4 supporta tutte le caratteristiche degli standard MPEG-1 e MPEG-2 oltre a tutta una serie di nuove caratteristiche come la gestione tridimensionale degli oggetti (tramite un'estensione del VRML). I flussi audio e video vengono trattati dello standard MPEG-4 come oggetti che possono essere manipolati e modificati in tempo reale. lo standard supporta caratteristiche specificate da terze parte come una particolare gestione dei Digital Rights

Management o una gestione interattiva dei contenuti

- **REAL VIDEO** è un video codec proprietario, sviluppato dalla RealNetworks. La sua prima release risale al 1997. **REAL VIDEO** è stato usato inizialmente per servire video streaming attraverso le reti internet ad un basso bit rate verso personal computer. Lo sviluppo delle connessioni verso la banda larga ha permesso in tempi recenti di offrire filmati con una maggior qualità. Inoltre, lo streaming consente di scaricare e vedere filmati di qualsiasi tipo su apparecchi cellulari. Real Video differisce dalle codifiche normali in quanto è un formato proprietario ottimizzato esclusivamente per lo streaming attraverso il protocollo (proprietario) PNA oppure tramite il Real Time Streaming Protocol. Può essere usato per scaricare e vedere video o per uno streaming live.
- **Windows Media Video (WMV)** è il nome generico per una serie di tecnologie proprietarie sviluppate da Microsoft per lo streaming di file video. Fa parte della piattaforma Windows Media. A partire dalla versione 7 (WMV1), Microsoft ha usato una sua versione modificata dello standard MPEG-4.
- **XviD** è un codec video open source aderente allo standard MPEG-4 (profilo ASP) originariamente basato su OpenDivX. Il progetto XviD è partito nel Luglio 2001, con la chiusura del progetto OpenDivX da parte di DivXNetworks Inc., società creatrice del popolare codec DivX. Il codec XviD ha avuto una considerevole diffusione soprattutto

nell'ambito del file sharing, dove viene utilizzato principalmente per comprimere la parte video dei film in modo che possano occupare poco spazio ed essere trasferiti velocemente nelle reti, appunto, di file sharing. La diffusione sul p2p di questo codec e il fatto che aderisse allo standard MPEG-4 ASP hanno fatto sì che molti produttori di lettori DVD che già erano in grado di leggere flussi video compressi in DivX (questi particolari lettori DVD spesso vengono chiamati *lettori Stand Alone*) aggiungessero il supporto ad XviD. La diffusione di XviD, oggi, è paragonabile a quella del codec "rivale" DivX.