

Prima di procedere oltre...

Bit, Byte, KiloByte, MegaByte,...

- Bit = '0' oppure '1'
- Byte = 8 bit = 2^3
- KiloByte (KB) = 2^{10} byte = 1.024 byte
- MegaByte (MB) = 2^{20} byte = 1.048.576 byte
- GigaByte (GB) = 2^{30} byte = 1.073.741.824 byte
- TeraByte (TB) = 2^{40} byte = ...
- PetaByte (PB) = 2^{50} byte = ...
- ExaByte (EB) = 2^{60} byte = ...

Tipologie di codici

Nel seguito vedremo tipologie di rappresentazioni diverse:

- Senza assumere limitazioni sul numero di bit a disposizione:
per numeri [notazione binaria, ovvero posizionale con base 2]
- Disponendo di un numero di bit limitato:
 - numeri naturali
 - interi relativi [valore assoluto e segno, complemento a due]
 - “reali” [virgola fissa e virgola mobile]
 - valori logici, caratteri alfabetici, testi
 - suoni, immagini e sequenze video
 - codici per la rilevazione e correzione di errori
- Codici di compressione (senza | con perdita)

Codifica binaria di valori logici

- Quanti oggetti?
DUE: V (vero) e F (falso)
- E' quindi sufficiente **un bit**, ad esempio:
 - **0: falso**
 - **1: vero**
- A volte sono disponibili più bit, ad esempio 8 (cfr. linguaggi di programmazione) e sono possibili diverse convenzioni, ad esempio:
 - Usare il bit meno significativo lasciando i rimanenti a 0
 - **0: falso; ogni altro valore: vero**

Codifica binaria dei caratteri

- Quanti oggetti?
 - 10 cifre
 - 26 lettere minuscole + 26 lettere maiuscole = 52
 - ~30 segni di interpunzione
 - ~30 caratteri di controllo (LF, CR, EOF, ...)

$\sim 120 \text{ oggetti} \Rightarrow k = \lceil \log_2 120 \rceil = 7$
- Il **codice ASCII** utilizza **7 bit** e quindi può rappresentare al massimo $n = 2^7 = 128$ caratteri
- Codice **ASCII esteso**: utilizza **8 bit** e quindi codifica 256 caratteri
- Codice **UNICODE**: utilizza **16 bit** e quindi codifica 65536 caratteri (anche quelli delle lingue orientali... non tutti! Sono circa 200 000! ...e aumentano! Estensione a 21 bit...)

Rappresentazione del testo

- In generale interessa rappresentare non solo i caratteri che compongono un testo ma anche altre caratteristiche di *formattazione* quali:
 - lo stile di scrittura (**grassetto**, *corsivo*, sottolineato,...)
 - la dimensione e il tipo o font (times new roman, arial, courier,...) del carattere
 - l'ampiezza dei margini della pagina
 - ...
- Per rappresentare i testi al calcolatore sono disponibili quindi diversi codici (comunemente detti formati) che consentono di rappresentare gradi diversi di caratteristiche del documento:
 - *Testo semplice* (text o txt): corrisponde a una sequenza di caratteri ASCII senza caratteristiche di formattazione
 - *Testo arricchito* (rich text format o rtf): in grado di rappresentare un ristretto insieme di formattazioni (stile, dimensione e colore dei caratteri)
 - *Testo Word* (document o doc): in grado di rappresentare un insieme ampio di formattazioni

Formati dei documenti per stampa e visualizzazione

- Formati standardizzati che sono orientati alla produzione di documenti destinati a *stampa* e *visualizzazione*:
 - Documenti in formato **PDF** (*Portable Document Format*, formato di documento portabile)
 - Documenti in formato **PS** (*Postscript*) - l'applicazione **GSview** (Ghostview) legge il formato postscript
- Non limitati alla rappresentazione di testo ma possono includere anche immagini e disegni

Tipologie di codici

Nel seguito vedremo tipologie di rappresentazioni diverse:

- Senza assumere limitazioni sul numero di bit a disposizione:
per numeri [notazione binaria, ovvero posizionale con base 2]
- Disponendo di un numero di bit limitato:
 - numeri naturali
 - interi relativi [valore assoluto e segno, complemento a due]
 - “reali” [virgola fissa e virgola mobile]
 - valori logici, caratteri alfabetici, testi
 - suoni, immagini e sequenze video
 - codici per la rilevazione e correzione di errori
- Codici di compressione (senza | con perdita)

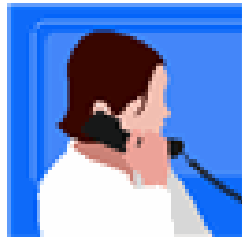
Rappresentazione di informazioni multimediali

- Con appositi metodi e codifiche si possono rappresentare informazioni complesse con sequenze di bit:
 - Suoni
 - Immagini
 - Filmati
- I problemi collegati alla gestione della rappresentazione, memorizzazione e elaborazione di queste informazioni sono:
 - la richiesta di elevate capacità di elaborazione
 - occupazione elevata di memoria: per questo si utilizzano tecniche di compressione dei dati che consentono di ridurre la richiesta di memoria

Digitalizzazione

rappresentazione anche approssimata
tramite sequenze di bit

- Voce



- Suoni



- Immagini

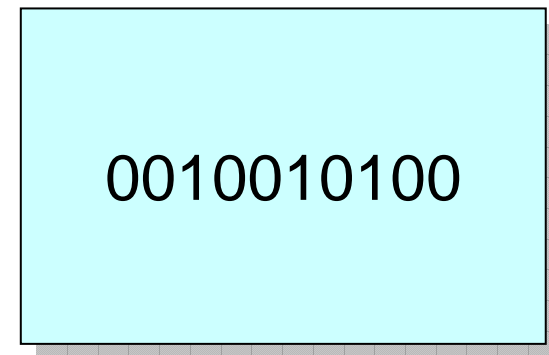


- Filmati



Informazione
“continua”

Codifica digitale



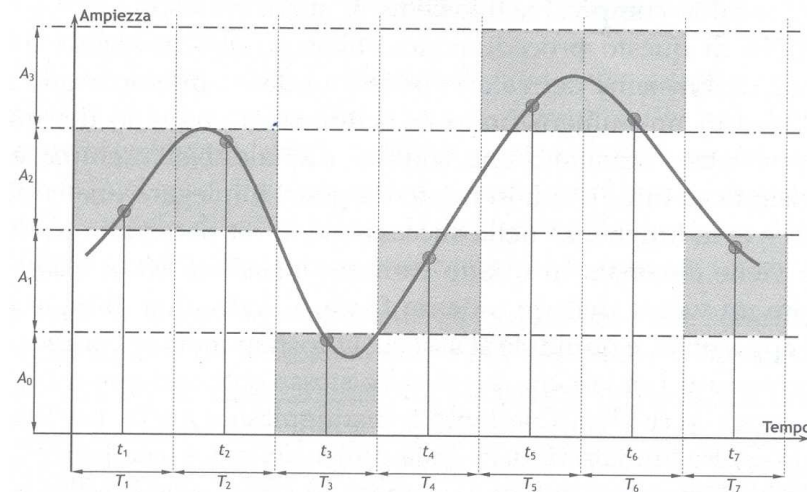
Nel calcolatore

La digitalizzazione

- **Problema:** codificare in modo digitale una grandezza fisica (es. volume di un suono) i cui valori si assumono variabili in un intervallo continuo
- Il processo di **digitalizzazione**, o **conversione analogico-digitale**, risolve tale problema
- Discretizzazione dei valori attraverso approssimazione con uno dei valori compresi fra quelli previsti nella codifica → **quantizzazione**
- Se la grandezza inoltre varia anche nel tempo (es. un suono) o nello spazio (es. colore di un'immagine) occorre anche un procedimento di **campionamento**

Codifica di un suono...

- ... in maniera digitale attraverso **campionamento** (nel tempo) e **quantizzazione** (in ampiezza)
- **Frequenza di campionamento:** numero di campioni acquisiti nell'unità di tempo (es. numero di campioni al secondo, misurato in Hertz) - Es. frequenza di 5 Hz significa acquisizione di 5 campioni al secondo
- Il **numero di bit** necessari per codificare ogni valore è il parametro che qualifica la quantizzazione - Es. con 8 bit posso codificare 256 valori diversi



Per ciascun intervallo di tempo viene scelto l'istante di campionamento t_i in cui viene rilevato il valore della grandezza

La quantizzazione è su 4 livelli, quindi il risultato del campionamento è $A_2 A_2 A_0 A_1 A_3 A_2 A_1$

La successione (codifica digitale) potrebbe essere
10 10 00 01 11 11 01

Rappresentazione dei suoni nei CD audio

- Segnali audio:
onde analogiche \rightarrow *campionamento* + *quantizzazione* \rightarrow *valori digitali*
- Frequenza di campionamento $f_c = 44.1$ kHz (44.100 campioni) per i 2 canali
- Quantizzazione = 65536 valori diversi (numero di livelli di quantizzazione) \rightarrow bit/campione = 16 bit (2 byte)
- Il numero di bit memorizzati è dunque:
$$N_{bit} = F(\text{durata}, f_c, \text{bit/campione})$$
- Nello standard CDDA (CD Digital Audio) 70 minuti registrazione richiedono $70 \times 60 \times 44.100 \times 2 \times 2$ byte ≈ 750 MByte
- Scheda audio e diffusori ricevono i valori digitali e ricostruiscono le onde analogiche

Alcuni formati audio

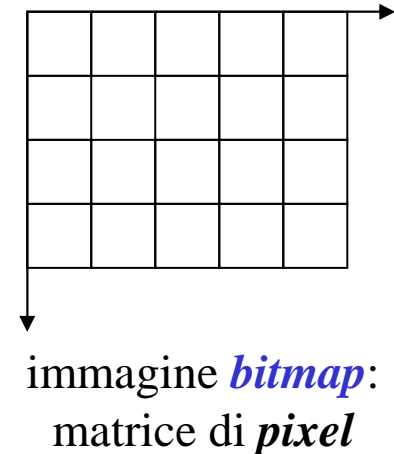
- **WAVE**: **.wav**, occupano molto spazio (~1MB per minuto)
- **MIDI** (*Musical Instrument Digital Interface*): **.mid**, i file memorizzano non suoni ma **comandi** (es. le note musicali di un particolare strumento) che vengono inviati ai dispositivi MIDI per riprodurre i suoni, occupano molto meno spazio dei **.wav**.

Es: tastiera musicale + calcolatore permettono di suonare un brano su tastiera che viene automaticamente trascritto in notazione musicale in un file midi che può poi essere anche modificato

- **MP3** (MPEG-1 Layer 3): grande diffusione su Internet. Utilizza tecniche di compressione (vedi poi). MPEG nasce da un gruppo di lavoro di standardizzazione

Codifica bitmap (raster) di immagini

- **Campionamento**: discretizzazione nello spazio: si divide l'immagine in quadrati, per ognuno dei quali si dovrà prelevare un campione che si considera rappresentativo di tutto il quadrato
- Questi quadrati sono trattati come elementi di base dell'immagine digitale: sono chiamati **pixel**
- Quanti più pixel vengono considerati per unità di superficie tanto più precisa sarà la riproduzione dell'immagine (**risoluzione**)
- **Quantizzazione**: codifica del colore associato a ogni pixel:
 - Immagini in bianco e nero: 1 bit per pixel (b/n)
 - Immagine a scala di grigi: es. un byte per pixel (256 livelli di grigio)
 - Immagine a colori: es. con modello RGB (colore ottenuto per sovrapposizione di Rosso Verde e Blu) 3 byte per pixel, uno per ogni colore primario

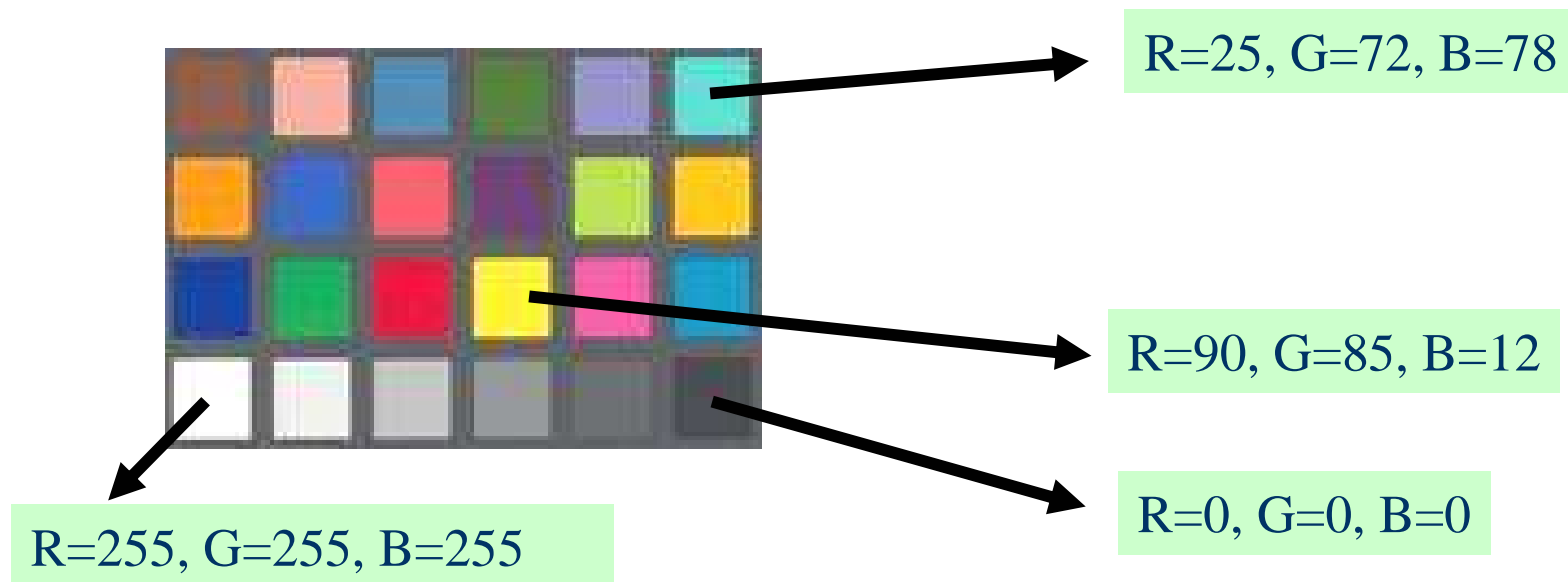


Numero di bit e capacità di rappresentazione

- 1 bit (per pixel) → **2** informazioni diverse (b/n)
- 4 bit (per pixel) → **16** colori o livelli di grigio
- 8 bit (per pixel) → **256** colori o livelli di grigio
- 16 bit (per pixel) → **64K** colori ($2^{16} = 2^{10} \times 2^6$)
- 24 bit (per pixel) → **16M** = 2^{24} colori (ogni pixel può assumere più di 16 milioni di colori)

Immagini bitmap RGB

- L'informazione sul colore di ogni pixel occupa 3 byte, uno per ogni colore primario (rosso, verde, blu)
- La quantità di colore primario è data da un valore tra 0 e 255 (o equivalentemente è rappresentato da una sequenza di bit a partire da 00000000 fino a 11111111)



Problemi con *immagini bitmap (o raster)*:

- ogni ingrandimento fa perdere qualità (l'immagine si “sgrana”):
vedi esempio nel prossimo lucido
 - per evitare questo problema: **codifica vettoriale** (vedi poi)
- occupano molta memoria. Possibili soluzioni:
 - uso di tavolozze (palette)
Es. immagine a 256 colori:
uso $256 \times 3 = 768$ byte per codificare la tavolozza
e solo un byte per pixel (seleziona uno dei 256 colori)
 - formati compressi (vedi poi)
 - codifica vettoriale

Immagini bitmap: esempio



Un'immagine RGB occupa:

3 byte x risoluzione verticale x risoluzione orizzontale

Esempio: un'immagine RGB con risoluzione 1024x768 pixel necessita di più di 2 MB

... e se la
ingrandisco si
sgrana...



I formati di file bitmap

- **BMP**: formato standard non compresso per MS Windows, 24 bit per pixel, **.bmp**
 - In grado di gestire palette a 2, 16, 256 colori + true color
- **TIFF** (*Tagged Image File Format*): alta qualità (32 bit per pixel), 16 milioni di colori (24 bit + ulteriori proprietà), dimensioni file molto grandi, **.tif**, molto adottato da scanner e macchine fotografiche

FORMATI COMPRESSI:

- **GIF** (*Graphic Interchange Format*): formato compresso, 8 bit per pixel (256 colori), **.gif**
- **JPEG** (*Joint Picture Experts Group*): 16 milioni di colori, formato compresso (più del gif), **.jpg**
- **JPEG 2000**: formato ancor più compresso, **.j2k** o **.jp2**

Codifica vettoriale di immagini

- Le immagini sono rappresentate tramite un insieme di elementi grafici (linee, rettangoli, ellissi, archi e curve)
- Memorizzazione come *coordinate numeriche* o *formule matematiche* che specificano forma e posizione: occupa poca memoria (es. un cerchio solo centro e raggio, un segmento le coordinate degli estremi)
- Necessaria *un'operazione di rendering (rasterizzazione)* che, a partire dalla descrizione matematica, produca l'immagine raster
- *Programmi di tipo draw* (programmi di grafica vettoriale): es. Corel Draw, programmi di CAD, programmi di grafica tridimensionale
- Vantaggi:
 - Controllo accurato di linee e colori
 - Ingrandimento, riduzione, rotazione senza perdita
 - Possibilità inserimento testo attorno agli oggetti
- Alcuni formati: **DXF, DWG, CDR, AI, WMF**

Immagini vettoriali: esempio



Codifica di sequenze video

- In teoria, una sequenza video è semplicemente una sequenza di fotogrammi (immagini):
 - campionamento nel tempo (successione di fotogrammi)
 - ciascun fotogramma rappresentato normalmente:
campionamento nello spazio e quantizzazione
- In pratica, le dimensioni risulterebbero troppo elevate. Facendo un conto “a spanne”, se un’immagine occupa 1 MB con 25 fotogrammi al secondo:
 - 25 MB al secondo
 - 25 x 60 MB al minuto = circa 1,5 GB al minuto!



Tecniche di compressione specifiche per le sequenze video