

bruno@magicbusmultimedia.it

<https://www.facebook.com/bruno.migliaretti>

<http://www.sisteminterattivi.org>

LEZIONE 01

RAPPRESENTAZIONE DIGITALE DELLE INFORMAZIONI

INTRODUZIONE

L'informatica l'ha inventata
Gutenberg

PRIMA DI **GUTENBERG**

A MANUENSIS



COME

- Ogni pagina (testo e figure) viene copiata a mano

PRO

- Assoluta libertà creativa: ogni copia è unica

CONTRO

- Alti costi: il lavoro di ogni amanuense produce una sola copia
- Possibilità di errori umani

PRIMA DI **GUTENBERG**

XILOGRAFIA



COME

- Ogni pagina viene ancora copiata a mano, ma incidendo una matrice di legno inversa.

PRO

- Una volta realizzata la matrice il processo di copia è automatico.

CONTRO

- Ogni matrice ha la vita di una pagina. Terminata la stampa non si può riutilizzare nulla.

DOPO GUTENBERG

CARATTERI COMPONIBILI



COME

- La matrice della pagina viene realizzata affiancando caratteri di piombo preesistenti.

PRO

- I caratteri di piombo possono riutilizzati molte volte.

CONTRO

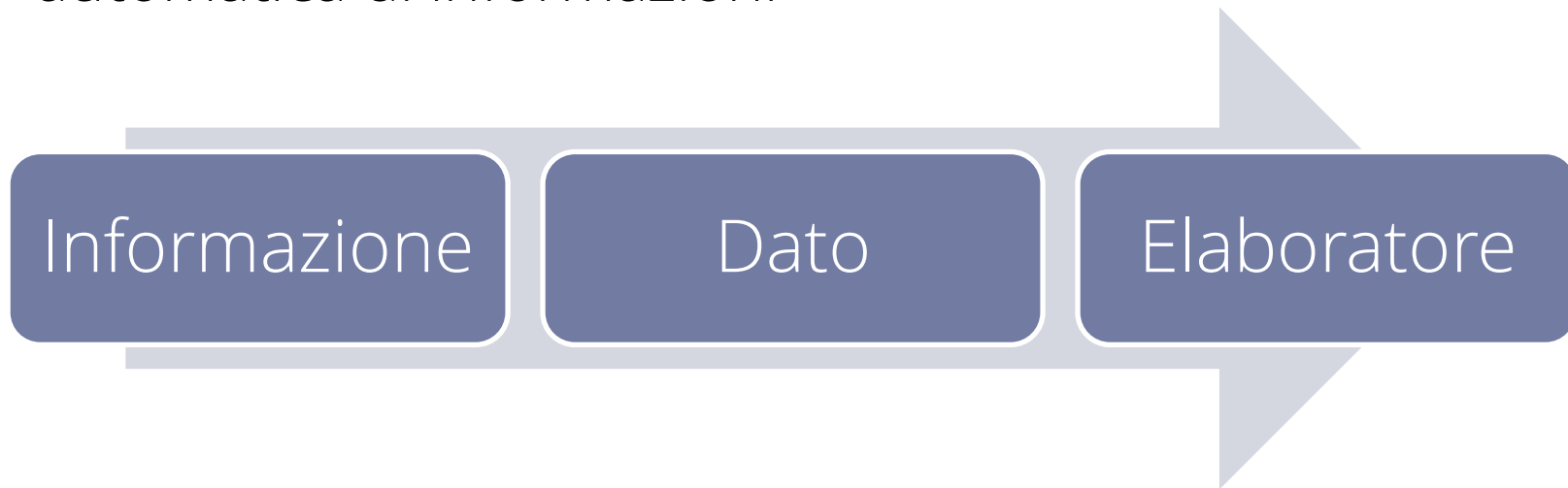
- Viene persa una parte di espressività artistica..

IL PROCESSO

- Ogni pagina viene scomposta in singole unità informative (i caratteri)
- Ad ogni carattere viene sostituita la versione che la macchina di stampa è in grado di elaborare (il carattere componibile di piombo)
- La pagina può essere data in pasto alla macchina che può stamparne migliaia di copie.

DEFINIZIONE DI **INFORMATICA**

- Informatica è la disciplina che studia l'elaborazione automatica di informazioni



DEFINIZIONE DI **INFORMATICA**

- Informatica è la disciplina che studia l'elaborazione automatica di informazioni



Pagina

Caratteri
componibili

Torchio da
stampa

INFORMATICA, ELABORATORE, **AUTOMAZIONE**

- *Informatica* è la disciplina che studia l'elaborazione automatica di informazioni
- *Elaboratore* è sistema per l'elaborazione automatica delle informazioni.
- Per *automazione* si intende tutto ciò che compie un compito prestabilito senza l'intervento umano.

AUTOMAZIONE

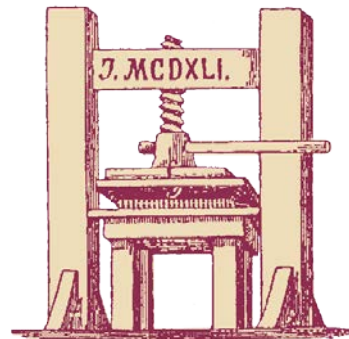
- Un processo viene automatizzato quando il numero di volte che esso deve essere eseguito è sufficientemente grande da rendere conveniente la progettazione e la costruzione di un sistema automatico che lo risolva.

ALCUNI **ESEMPI**

PROBLEMA SEMPLICE (non ripetitivo)	PROBLEMA SISTEMATICO (ripetitivo)
<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>
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 con la stessa quantità.
<i>Rubinetto manuale</i>	<i>Rubinetto a galleggiante</i>

CODIFICA

- *Codifica* è l'operazione con cui rendiamo disponibili le informazioni per l'elaborazione automatica. Ogni informazione viene tradotta in un sistema simbolico su cui l'elaboratore può operare.



CODIFICA

- A secondo del tipo di informazione da trattare di solito ho regole di codifica diverse.
- I caratteri componibili sono ottimi per stampare i testi ma non mi servono a nulla per stampare immagini o disegni.
- Per inserire disegni in una pagina stampata con il metodo di Gutenberg posso creare delle incisioni da affiancare ai caratteri componibili.

PROGRAMMABILITÀ

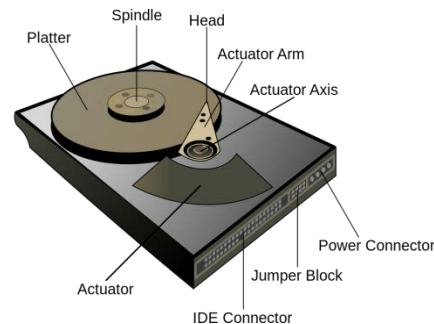
- Un torchio da stampa è una macchina estremamente semplice e specializzata in grado di svolgere un unico compito.
- Ci sono poi macchine che anche se dedicate ad un compito specifico, lo possono svolgere in maniera diversa a seconda dei programmi che eseguono (la macchine industriali a controllo numerico, ma anche le macchine da cucire da famiglia).
- Infine ci sono i computer che, nati come macchine per eseguire calcoli, hanno (soprattutto nelle varie versioni personal) perso ogni specificità e sono in grado, eseguendo i programmi adatti, di elaborare qualsiasi tipo di informazione.

CODIFICA E **COMPUTER**

- Il computer è l'elaboratore programmabile più complesso e completo oggi esistente.
- Come succede per tutti gli elaboratori, le informazioni devono essere correttamente codificate perché il computer possa elaborarle.
- Il computer è una macchina elettronica e quindi deve usare dei dispositivi elettrici per elaborare le informazioni codificate.
- Come base del sistema di codifica si è deciso di usare dei micordispositivi in grado di avere due stati elettrici definiti che nell'elaborazione rappresentano le cifre **0** e **1**

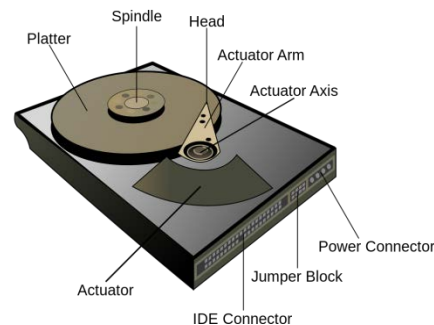
CODIFICA E COMPUTER

- Nelle schedine che vedete qui a destra sono collocate centinaia di miliardi di microdispositivi che possono avere due stati elettrici definiti (potete pensare ad acceso/spento o a carico/scarico)
- Sulla superficie magnetica di un hard-disk abbiamo decina di migliaia di miliardi di particelle in grado di avere due stati magnetici definiti.
- Nel primo caso abbiamo la cosiddetta **RAM**, memoria temporanea che il computer usa per elaborare le informazioni. Nel secondo caso tipo di **memoria di massa** dove i dati vengono memorizzati in maniera permanente.
- In entrambi i casi i due stati rappresentano le cifre **0** e **1**.



BIT BINARY DIGIT

- Letteralmente la parola bit significa cifra binaria
- In generale un bit e' una unità che può assumere un valore tra due possibili (normalmente si parla di 0 e 1)
- La rappresentazione fisica di un bit richiede un qualsiasi dispositivo in grado di trovarsi in uno di due possibili stati
- Interruttore (accesso/spento)
- Un condensatore (carico/scarico)
- Una bandiera (alzata/abbassata)
- Una particella magnetica (Nord/Sud)



SISTEMA **BINARIO**

- Il sistema numerico binario è un sistema numerico posizionale in base 2. Esso utilizza solo due simboli, di solito, indicati con 0 e 1, invece delle dieci cifre utilizzate dal sistema numerico decimale. I numeri espressi nel sistema numerico binario sono chiamati numeri binari.
- Un numero binario è una sequenza di cifre binarie (dette bit). Il valore della cifra nella posizione **n** (contando da destra verso sinistra iniziando da 0) si ottiene moltiplicando la cifra per **2^n** , anziché per **10^n** , come avviene nella numerazione decimale.

Binario	Esadecimale	Decimale
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7
1000	8	8
1001	9	9
1010	A	10
1011	B	11
1100	C	12
1101	D	13
1110	E	14
1111	F	15

SISTEMA **ESADECIMALE**

- Il sistema numerico esadecimale (spesso abbreviato come **esa** o **hex**) è un sistema numerico posizionale in base 16, cioè che utilizza 16 simboli invece dei 10 del sistema numerico decimale tradizionale. Per l'esadecimale si usano in genere simboli da **0** a **9** per le prime dieci cifre, e poi le lettere da **A** a **F** per le successive sei cifre, per un totale di 16 simboli.
- Il sistema esadecimale è molto usato in informatica, per la sua relazione diretta tra una cifra esadecimale e quattro cifre binarie.

Binario	Esadecimale	Decimale
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7
1000	8	8
1001	9	9
1010	A	10
1011	B	11
1100	C	12
1101	D	13
1110	E	14
1111	F	15

base
10

1125

Posizione	Calcolo	Valore decimale
0	$10^0 * 5$	5
1	$10^1 * 2$	20
2	$10^2 * 1$	100
3	$10^3 * 1$	1000
Totale		1125

base
2

1101

Posizione	Calcolo	Valore decimale
0	$2^0 * 1$	1
1	$2^1 * 0$	0
2	$2^2 * 1$	4
3	$2^3 * 1$	8
Totale		13

base
16

A1F8

Posizione	Calcolo	Valore decimale
0	$16^0 * 8$	8
1	$16^1 * 15$	240
2	$16^2 * 1$	256
3	$16^3 * 10$	40960
Totale		41464

CODIFICA

- La codifica e' l'operazione che consente trasformare le informazioni in dati numerici che calcolatori elettronici possono leggere ed elaborare.
- Un bit puo' assumere solo due valori (0 e 1)
- Per rappresentare insiemi costituiti da piu' di due stati/simboli si usano serie di bit.
- Una stringa di bit e' costituita da un certo numero di bit (normalmente 8 o multipli di 8) ed e' detta parola (*word*).

CODIFICA

- La codifica si articola in due fasi:
 - Divisione dell'informazioni in unità informative
 - Assegnazione ad ogni unità di un valore binario che la codifica
- La precisione con cui sarò in grado di rappresentare l'insieme delle informazioni che codifico dipenderà dal rapporto tra il numero delle unità informative diverse da codificare e numero dei codici di cui dispongo.
- Con n bit si possono rappresentare 2^n valori diversi e quindi si possono rappresentare 2^n informazioni diverse
- La lunghezza della parola, quindi, definisce quante informazioni possono essere codificate.

NUMERO DI COMBINAZIONI

parola di 8 cifre decimali

$10^8 = 100.000.000$ combinazioni
da 0 a **99.999.999**

parola di 8 cifre binarie

$2^8 = 256$ combinazioni
da 0 a **11111111**

CODIFICA DEL TESTO

IL TESTO

- Un testo è una sequenza di caratteri alfabetici, separatori e caratteri speciali
- Nella codifica ad ogni carattere viene assegnato un numero binario
- Più è lunga la parola che utilizzo per codificare il testo più saranno i caratteri che sarò in grado di rappresentare.
- Pensiamo (ad esempio) di dover rappresentare le sole lettere dell'alfabeto italiano (21 lettere + 10 numeri + 10 punteggiatura = 41 simboli).
- Possiamo usare una codifica a 6 bit. ($2^6 = 64$ combinazioni)

ESEMPIO CODIFICA TESTO A 6 BIT

- 000000 = a
- 000001 = b
- 000010 = c
- 000011 = d
- 000100 = e
- 000101 = f
- 000110 = g
- 000111 = h

Un testo è rappresentato dalla sequenza di byte associati ai caratteri che lo compongono, nell'ordine in cui essi compaiono

STANDARD ASCII

AMERICAN STANDARD CODE FOR INFORMATION INTERCHANGE

- La codifica ASCII prevede l'utilizzo di 128 caratteri diversi
- Ogni carattere è associato ad una diversa configurazione di 7 bit
- La codifica ASCII estesa prevede 256 simboli e 8 bit (*1 byte*) per ogni carattere
- Quindi un testo di 1000 caratteri richiede 1 Kbyte per essere rappresentato

STANDARD **UNICODE**

- Lo standard ISO10646/Unicode si basa su una codifica a 32 bit che consente oltre due miliardi di possibili caratteri
- UTF 8 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.

CODIFICA ESATTA E CODIFICA APPROSSIMATA

CODIFICA

- Una codifica esatta a n bit è possibile solo quando l'insieme delle informazioni da codificare è finito e di dimensione inferiore o uguale al massimo del valore che posso rappresentare con una parola di una determinata lunghezza
- I calcolatori sono oggetti finiti che elaborano e memorizzano un numero finito di bit.
- Se l'insieme da codificare ha una contiene un numero di informazioni maggiore di 2^n se ne puo' dare solo una rappresentazione approssimata o parziale. Questa limitazione avviene in due modi:
 - **Operazioni di limitazione**
 - **Operazioni di partizionamento**

NUMERI INTERI

- I numeri interi sono un insieme discreto illimitato.
- Per poter essere codificati devono essere limitati.
- Sottoinsieme simmetrico rispetto allo 0.
- Si usa 1 bit per rappresentare il segno e i restanti $a = n-1$ per rappresentare il modulo.
- Il massimo numero rappresentabile è (in modulo) 2^a-1 .
- Se il risultato di un'operazione eccede il modulo 2^a-1 non può essere codificato e il calcolatore restituisce un messaggio di overflow.

NUMERI REALI

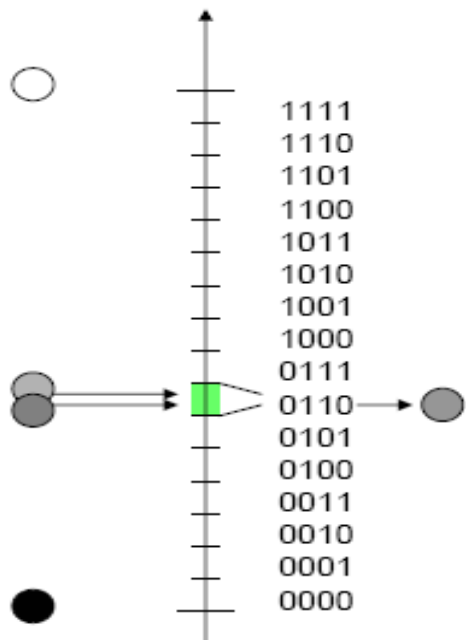
- I numeri reali sono un insieme continuo e illimitato.
- Per poterli rappresentare occorre limitarli (in modo simmetrico rispetto allo 0) e partizionarli.
- *Rappresentazione in virgola fissa*: Degli n bit della parola, 1 rappresenta il segno, a rappresentano le cifre prima della virgola e b le cifre dopo la virgola.
 - Il massimo numero rappresentabile è $(2^{n-1}-1)/2^b$
 - L'accuratezza assoluta è 2^{-b}
- *Rappresentazione in virgola mobile*: (Floating point).

CODIFICA DELLE IMMAGINI

RAPPRESENTAZIONE DELLE IMMAGINI

- Le immagini sono informazioni continue in tre dimensioni: due spaziali ed una colorimetrica.
- Per codificarle occorre operare tre discretizzazioni.
 - 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.

ESEMPIO: LIVELLI DI GRIGIO



- La codifica associa un unico codice ad un intervallo di livelli di grigio
- Tutti i livelli di grigio all'interno dell'intervallo vengono codificati allo stesso modo comportando una perdita di informazione
- Il livello di grigio originale non può essere ricostruito in maniera esatta dal codice binario

LIVELLI DI GRIGIO



8 bit



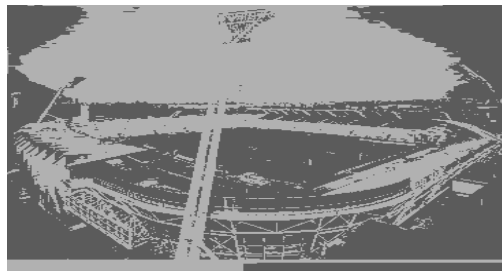
5 bit



4 bit



3 bit



1 bit

IMMAGINI A 256 COLORI

- La codifica è composta da due elementi distinti:
 - Una tabella di colori (**palette**) in cui vengono definiti fino ad un massimo di 256 colori
 - I punti (**pixel**) di cui è composta l'immagine il cui colore è definito da un byte (8 bit) che indica quale colore usare tra quelli definiti nella tabella.

Immagini a 256 colori

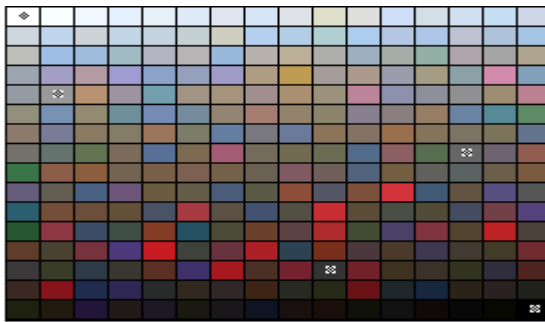


Tabella dei colori (palette) composta da 256 colori numerati da 0 a 255. Ogni colore viene definito per il suo contenuto di Rosso, Verde e Blu.

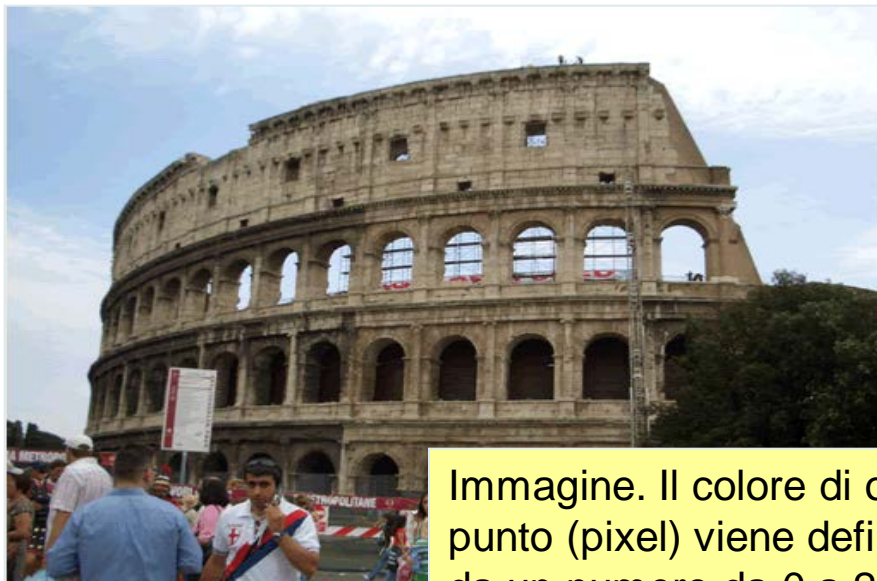
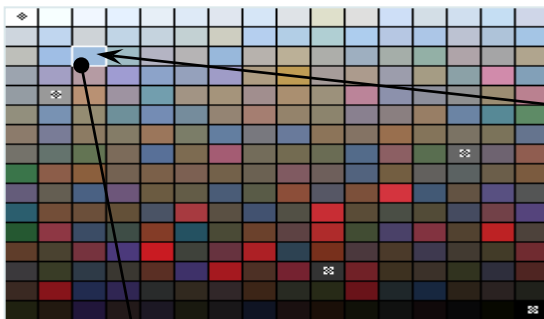


Immagine. Il colore di ogni punto (pixel) viene definito da un numero da 0 a 255 (8 bit). Viene utilizzato il colore definito nella palette all'indice corrispondente.

Immagini a 256 colori



Ogni colore viene definito nella palette specificando i tre colori fondamentali.

Indice	Rosso	Verde	Blu
00100010	10011110	10111101	11011110

Il colore del pixel è definito dal numero **00100010** (34 decimale) che rappresenta l'indice della palette.

Immagini RGB

Nelle immagini a 24 bit tre byte definiscono i livelli dei colori fondamentali.

Rosso	Verde	Blu
10011110	10111101	11011110

Nelle immagini a 16 bit si usano 5 bit per definire i livelli dei colori fondamentali.

Rosso	Verde	Blu
10011	10111	11011

Nelle immagini a 32 bit tre byte definiscono i livelli dei colori fondamentali il quarto il livello di trasparenza (alpha) del pixel.

Rosso	Verde	Blu	Alpha
10011110	10111101	11011110	11111111

Il colore del pixel è definito direttamente da 2 o più byte che ne specificano la composizione in termini di **R**ed, **G**reen, **B**lue.



IMMAGINI VETTORIALI

- La grafica vettoriale scompone in gruppi logici di componenti (linee, cerchi, rettangoli, ecc.)
- Le forme vengono memorizzate in termini di coordinate e colori dei vari elementi geometrici che le compongono
- Durante la visualizzazione, coordinate e colori vengono utilizzati per ricreare l'immagine
- La grafica vettoriale e' comunemente usata nei disegni, disegni animati e nella grafica lineare in generale

OGGETTO LINEARE

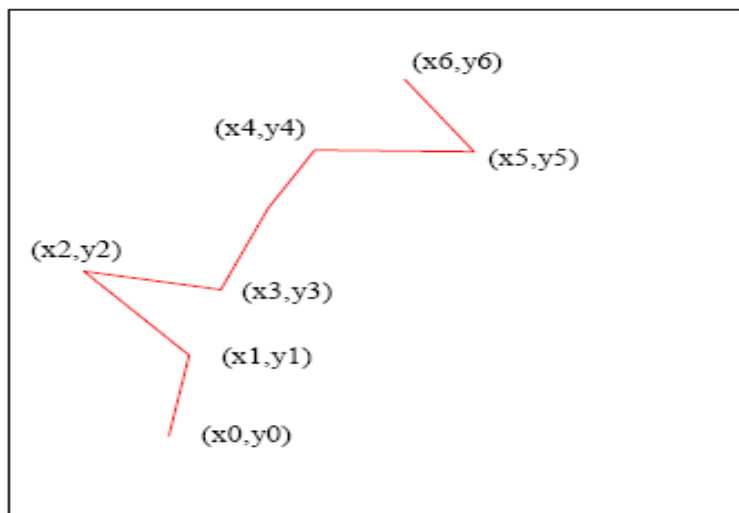
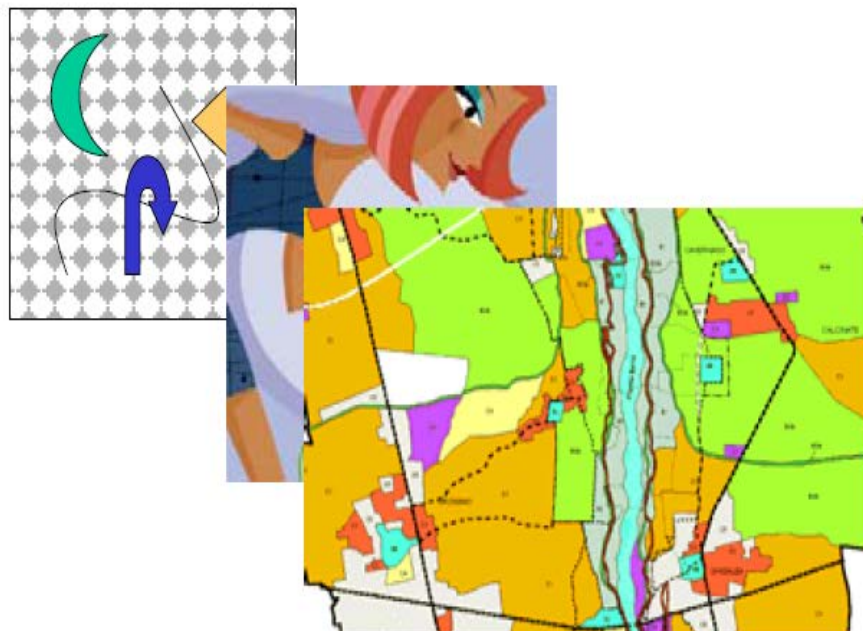


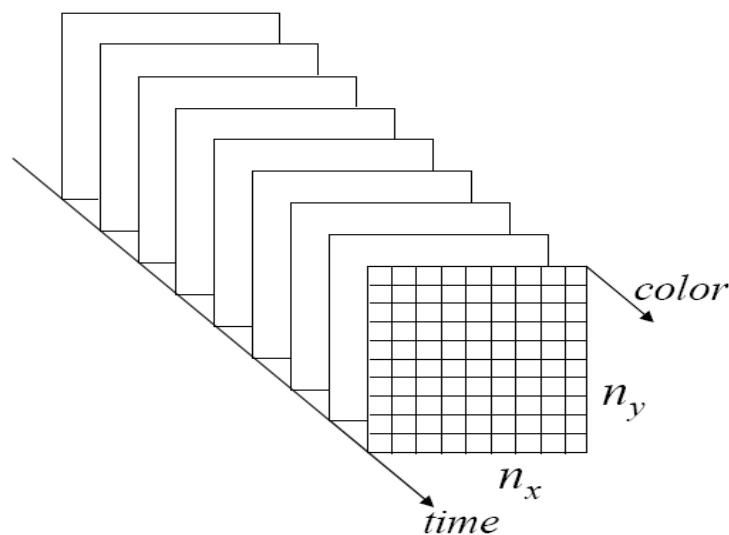
Immagine = $x_0, y_0, x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4, x_5, y_5, x_6, y_6$, rosso

IMMAGINI VETTORIALI



Codifica di un filmato video

$$size = s_{rate} \cdot T \cdot s_{size} = s_{rate} \cdot T \cdot \log_2 n_{col} \cdot n_x \cdot n_y$$



s_{rate} = frame rate

n_{col} = numero di colori

$n_x n_y$ = dimensione del frame

Filmato di 10 minuti a 25 frame al secondo, 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 dei segnali analogici

Codifica di segnali analogici

- *Segnale*: Quantita' fisica che varia nel tempo
 - *Analogico*: tempo-continuo, valore-continuo
 - *Digitale*: tempo-discreto, valore-discreto
- La codifica digitale di un segnale continuo comporta:
 - *Campionamento* (discretizzazione nel tempo)
 - *Quantizzazione* (discretizzazione nel valore)

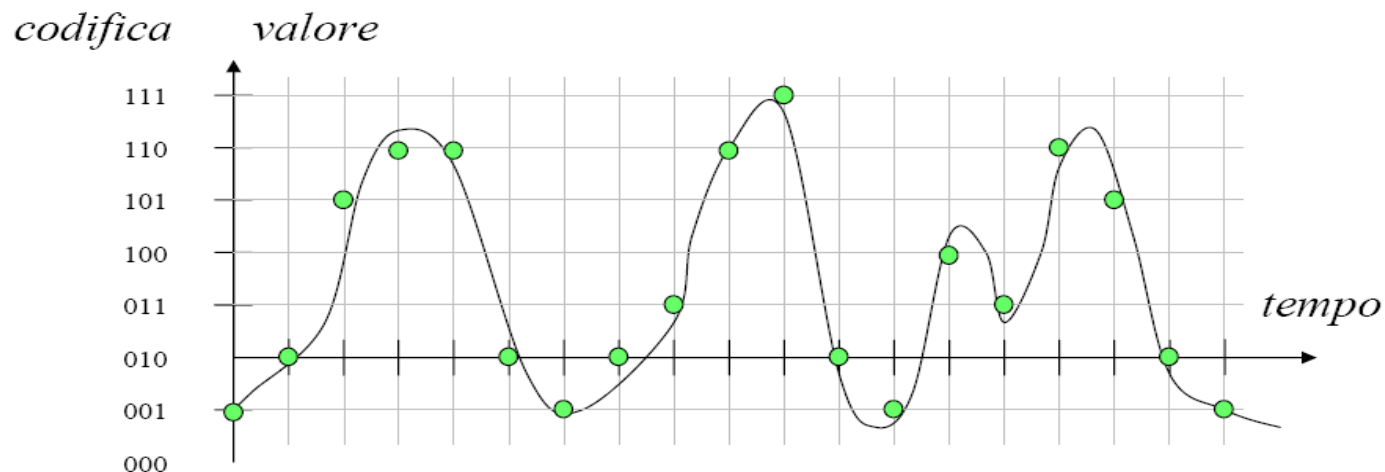
$$size = s_{rate} \cdot T \cdot s_{size}$$

frequenza di
campionamento

Durata

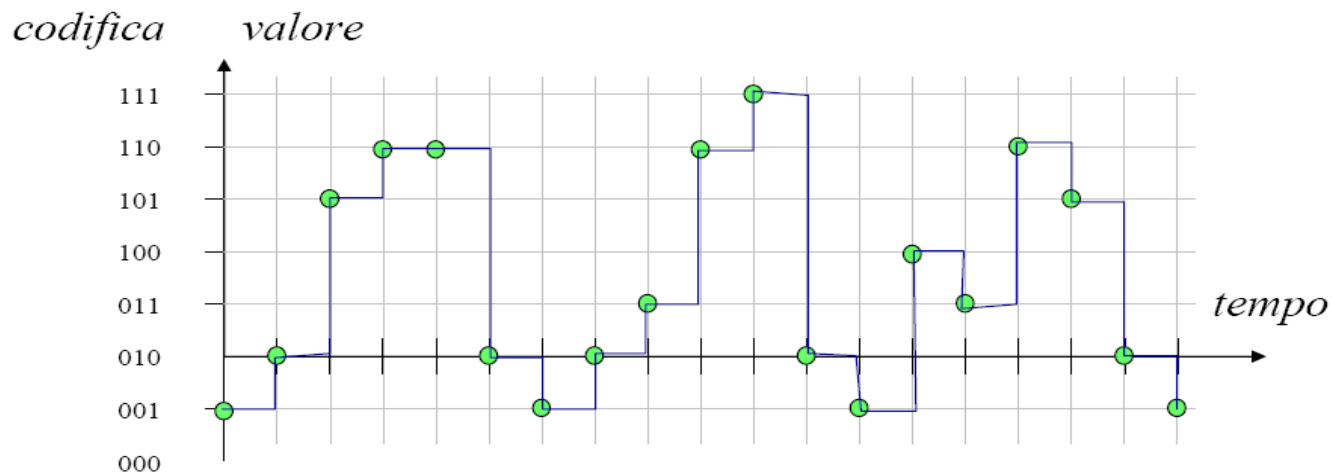
Dimensione del campione

Campionamento e quantizzazione



File finale = 001-010-101-110-110-010-001-010-011-110-111-010-001-100-011-110-101-010-001

Campionamento e quantizzazione



File finale = 001-010-101-110-110-010-001-010-011-110-111-010-001-100-011-110-101-010-001

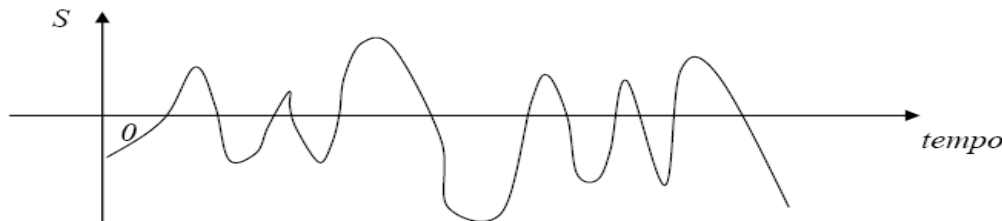
Codifica del suono

- Il suono è un segnale analogico tempo-continuo
- Un'onda sonora non è altro che una successione di rarefazioni e compressioni di piccole porzioni d'aria



Una descrizione del suono

- Il suono prodotto da un altoparlante e' 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.



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.

Lo standard telefonico prevede un campionamento a 8KHz ed una quantizzazione a 256 livelli (codificati con 8 bit). Quindi per ogni secondo di conversazione servono 64Kbit

Questo tipo di codifica e' comunemente utilizzato nel formato WAVE

Formato MIDI

(Musical Instrument Digital Interface)

- Il formato MIDI ha, tra i formati audio, lo stesso ruolo che, tra i formati grafici, ha il formato vettoriale.
- Un file MIDI non è una registrazione sonora
- Contiene comandi che, quando eseguiti, producono dei suoni (mediante un sintetizzatore hardware o software).
- I file MIDI, per questo motivo, sono molto piccoli rispetto ai file audio, ma riproducono un suono "sintetico", non registrato.
- Un file MIDI ha una forma del tipo:
 - Suona la nota x per un tempo t con lo strumento y

COMPRESSIONE DEI DATI

Compressione dei dati

- Tecniche di compressione vengono utilizzate ogni qualvolta si deve ridurre la dimensione dei dati su cui si sta lavorando
- Archiviazione (i supporti di memorizzazione sono limitati)
- Trasmissione (la velocità di comunicazione sulla rete o verso certe periferiche può rappresentare un fattore importante)

La ridondanza dei dati

- Una codifica e' detta ridondante se utilizza piu' bit del necessario. Es: Codice ASCII a 8 bit
- Se la codifica e' ridondante il valore di alcuni bit diventa prevedibile conoscendo gli altri
- Si dice che il contenuto informativo e' inferiore al 2^n per una codifica ridondante a n bit
- In altri termini, alcuni bit non aggiungono informazione a quanto gia' codificato dai precedenti

L'utilità della ridondanza

- L'uso di codifiche ridondanti può avere due motivazioni: la *flessibilità* e l'*affidabilità*.
- *La flessibilità*: possibilità di utilizzare la stessa codifica in situazioni diverse. Es: "Millennium Bug"
- *L'affidabilità*: facilmente comprensibile e non incline ad essere male interpretato anche nel caso in cui il messaggio venga trasmesso in modo solo parziale o corrotto.

Esempio di ridondanza

- Nella lingua italiana l'utilizzo della lettera "q" e' un chiaro esempio di ridondanza
- La lettera "q" e' sempre seguita dalla lettera "u"
- La parola "quadro" sarebbe comprensibile anche se scritta "q*adro" benché non corretta
- In generale: *tutto ciò che si riesce ad indovinare e' ridondante nel contesto in cui viene utilizzato*

Compressione e ridondanza

- Un primo esempio di compressione della lingua italiana e' quello di omettere il carattere "u" tutte le volte che questo e' preceduto dal carattere "q"
- Qualcosa del genere viene comunemente fatto quando si scrive "xche'" al posto di "perche'" o "ke" al posto di "che".
- In generale, tutte le codifiche ridondanti si prestano ad essere compresse

Tecniche generiche di compressione

- *Lossless*: tecnica senza perdita di informazione
 - *permettono di recuperare interamente l'informazione contenuta nel testo prima della sua compressione*
 - *Utilizzate nei casi in cui non e' possibile accettare neppure la minima perdita delle informazioni Es: Winzip*
- *Lossy*: tecnica distruttiva con perdita di informazione
 - *Non e' possibile ricostruire in maniera esatta i dati di partenza attraverso il processo di decompressione.*
 - *Utilizzate nella compressione delle immagini, dei filmati e dei suoni → riduzione della qualità*

Compressione Run Length Encoding (RLE)

- Algoritmo di compressione di tipo lossless
- Si basa sul fatto che nei dati da comprimere esistono sequenze, dette *run*, che si ripetono costantemente
- Una volta individuate le sequenze ripetute, vengono sostituite da un unico simbolo e dal numero delle ripetizioni presenti.

Esempio di compressione (*RLE*)

- una stringa di bit del tipo “ 011100001 ” verrebbe compressa codificandola in “03*14*01” che si legge “ 0, tre volte 1, quattro volte 0, 1 ” .
- Varianti di RLE si basano sulla diversa lunghezza minima da attribuire ad un *run*.
- Nell' esempio abbiamo usato *run* pari a uno
- Possiamo prendere in considerazione gruppi di simboli ($run > 1$) e trovare quante volte l'intero gruppo viene ripetuto all' interno della stringa.

Compressione a codifica differenziale

- Spesso le informazioni sono costituite da blocchi di dati, ognuno dei quali differisce leggermente dal precedente
 - Es: fotogrammi successivi di un filmato
- In questo caso la compressione differenziale memorizza non il blocco stesso ma le sue differenze rispetto al precedente

Compressione adattativa basata su dizionario

- Il termine dizionario si riferisce all' insieme di elementi di base sui quali viene ricostruito il messaggio compresso.
- I simboli del dizionario rappresentano particolari sequenze di bit e durante la compressione, ad ogni sequenza riconosciuta viene sostituito il simbolo corrispondente.
- Il dizionario viene creato dinamicamente durante il processo di compressione.
- L' algoritmo di compressione *Lempel-Ziv* si basa su questa tecnica (WinZip)

Compressione delle immagini

- Nella compressione delle immagini si usano sia tecniche di tipo *lossy* che tecniche di tipo *lossless*
- GIF, PNG → *lossless*
- JPEG → *lossy*
- La compressione di tipo *lossy* da ottimi risultati per quanto riguarda la dimensione del file prodotto apportando, in alcuni casi, perdite non visibili.
- La compressione di tipo *lossless* da buoni risultati per immagini a colori piatti (poco efficiente su immagini molto complesse e sfumate)

Formato GIF

(*Graphic Interchange Format*)

- Max 256 colori → profondità di colore 8 bit.
- Se l'immagine originale contiene un numero più elevato di colori è necessario effettuare una riduzione con conseguente perdita di qualità
- Integra una compressione di tipo LZW (ZIP)
- I colori sono memorizzati in una 'tavolozza', una tabella che associa un numero ad un certo valore di colore.
- Supporta il formato *interallacciato*
- consente anche di definire un colore come *trasparente*.

Formato JPEG/JPG

(Joint Photographic Expert Group)

- 24 bit = 16,8 milioni di colori.
- Tecnica di compressione basata di una codifica dell'immagine *percettiva* in cui viene distinta la luminosità dei pixel dal loro colore.
- Per ogni pixel viene codificata la componente della luminosità mentre il colore viene codificato a blocchi di 4 pixel; (codificando il colore medio dei quattro)
- Per 4 pixel → 6 valori (4 di luminosità e 2 di colore) anziché 12 valori come in 24 bit (RGB)
- E' possibile scegliere il grado di compressione/qualità'
- Il formato JPEG progressivo emula l'interlacciamento

Formato PNG

(Portable Network Graphic)

- 8, 16, 24 bit = 256, 65536, 16.8 milioni di colori.
- Non è in grado di raggiungere l'efficienza del formato JPEG per quanto riguarda il fattore di compressione
- Supporta sia l'effetto trasparenza che l'interlacciamento
- Può incorporare del testo all'interno dell'immagine (come stringa) utile per classificarla e per fare ricerche sui contenuti