



# Sistemi di Elaborazione delle Informazioni

---

*Multimedia*

Corso di Laurea in Tecniche Sanitarie di Radiologia Medica

**Prof. Crescenzo Gallo**

*c.gallo@unifg.it*

# Informazioni multimediali



# Informazioni multimediali

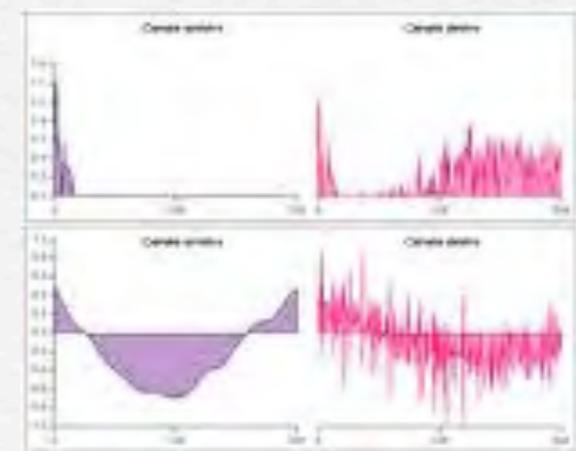
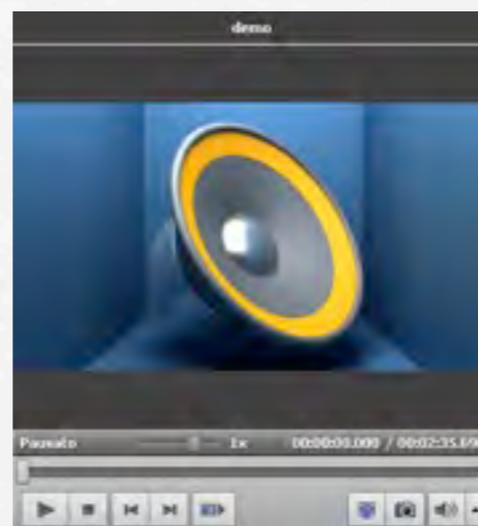
Lettere e numeri non costituiscono le uniche informazioni utilizzate dagli elaboratori, ma ci sono sempre più applicazioni che utilizzano ed elaborano anche altri tipi di informazione:

→ **diagrammi, immagini, suoni.**

In questi casi si parla di applicazioni di tipo multimediale.

Abbiamo bisogno di un sistema di **CODIFICA**.

In generale si parla di **DIGITALIZZAZIONE**.



# La digitalizzazione

- \* Il termine digitale deriva dall'inglese **digit** (cifra numerica).
- \* Per *digitalizzazione* si intende la rappresentazione (codifica) di un qualunque fenomeno o oggetto fisico attraverso una sequenza di numeri.
- \* La digitalizzazione consente di rappresentare un qualunque oggetto o fenomeno all'interno della memoria del computer.
- \* Consente ad un calcolatore di interagire, analizzare, modificare le rappresentazioni di tali oggetti.



# Esempi di digitalizzazione

## Immagine



Codifica

## Codifica jpeg

```
0101000101010111101
0101110101000000101
0101000100100010000
0011110101111100010
1010110101010100110
1101010100000110101
```

## Testo

Il termine *word* indica, invece, una serie di bit (in numero tale da essere potenza di 2) che hanno un particolare significato; si parla quindi di *word* di 4, 16, 32 o 64 bit. Normalmente, una *word* è la dimensione minima di bit su cui un calcolatore può eseguire operazioni elementari; i vecchi PC lavoravano con *word* di 8 o 16 bit, mentre gli attuali elaboratori hanno *word* di 64 bit.

In definitiva, se per noi è semplice eseguire calcoli nella forma  $12+15$  o ricercare la parola "ciao" in un testo, per un elaboratore elettronico è molto più facile sommare  $1100$  a  $1110$  (intesi come numeri binari) o cercare la sequenza di bit "01110101011101100101011011001101" (possibile codifica della parola "ciao") in una serie di un milione di cifre binarie.

### 4.1.2 Multipli utilizzati

La memoria utilizzata per codificare una pagina di testo è di qualche *migliaio* di byte, quella usata per una immagine può raggiungere il *milione* di byte, mentre un lungo filmato può richiedere *miliardi* di byte per essere memorizzato.

Si avverte la necessità, come nel *sistema metrico decimale*, di utilizzare dei simboli per rappresentare i multipli delle grandezze elementari; nella terminologia informatica sono stati quindi adottati gli stessi simboli del sistema decimale, ma visto che la misurazione della memoria ha come sua base principale il 2, il loro significato è *leggermente diverso*.

La seguente tabella riassume i simboli e i valori dei multipli più usati in campo informatico:

Codifica

## Codifica ASCII

```
0101000101010111101
0101110101000000101
0101000100100010000
0011110101111100010
1010110101010100110
1101010100000110101
```

# Conversione analogico-digitale



La conversione analogico-digitale trasforma un segnale analogico (valori continui in un tempo continuo) in segnale numerico (valori discreti in tempo discreto).

Questa operazione è costituita da tre fasi:

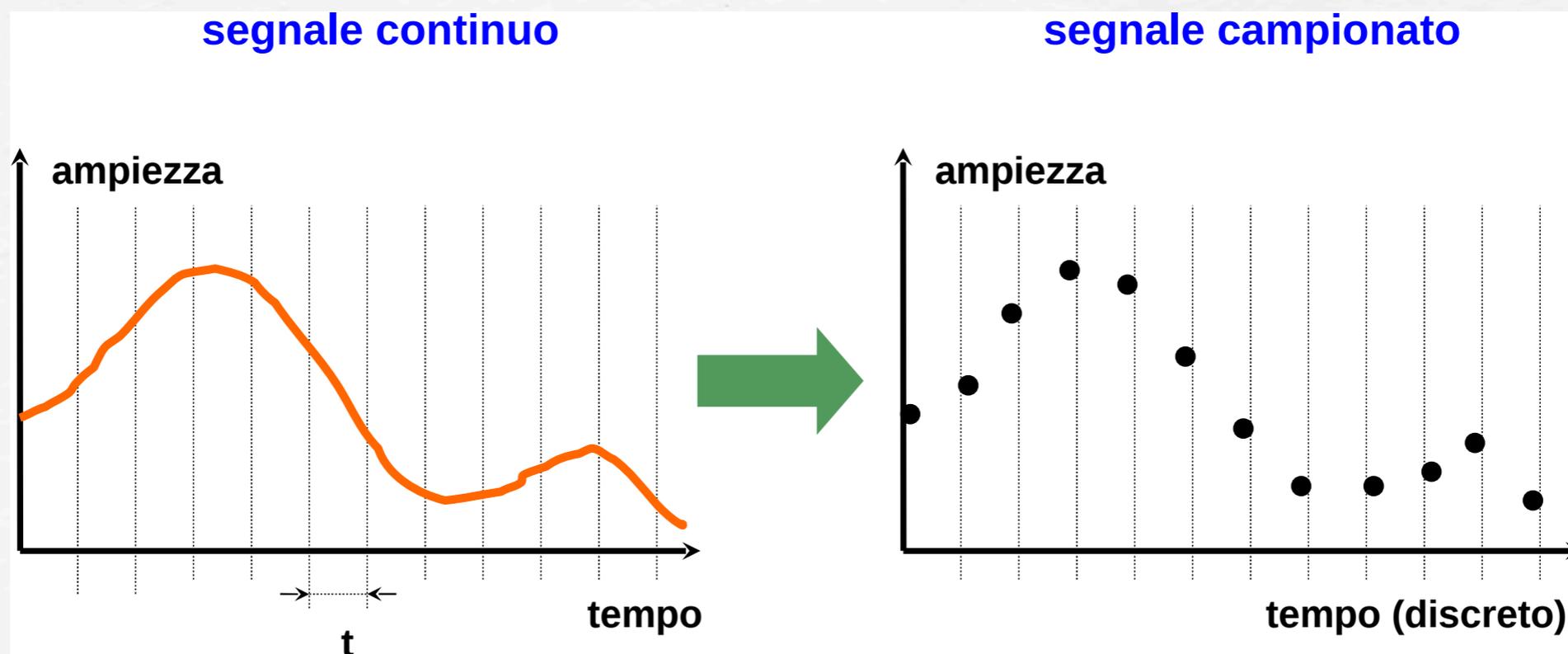
- ▶ **Campionamento**
- ▶ **Quantizzazione**
- ▶ **Codifica**

# Campionamento

Il **campionamento** è una tecnica di rappresentazione del segnale che consiste nella valutazione dell'ampiezza dello stesso ad intervalli di tempo regolari.

L'accuratezza di un campionamento dipende essenzialmente dalla **frequenza** di campionamento che è il numero di campioni rilevato nell'unità di tempo.

**Esempio:**



# Quantizzazione/Codifica

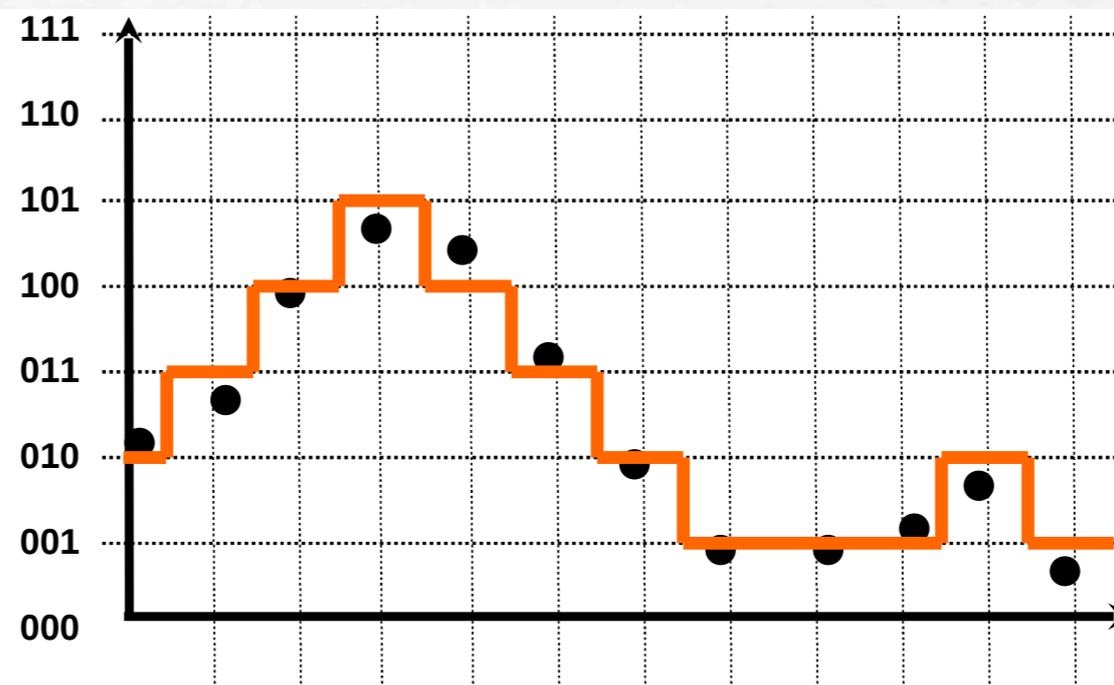
- ▶ In natura, la maggior parte delle grandezze possono assumere un insieme infinito e continuo di valori.
- ▶ Affinché una grandezza sia trasmissibile e codificabile con un numero finito di bit, è però necessario far sì che possa assumere solo un numero finito di valori discreti; ciò avviene tramite la **quantizzazione**.
- ▶ Un **quantizzatore** associa ad ognuno dei valori continui in ingresso il più prossimo tra quelli definiti.
- ▶ L'**errore di quantizzazione** è definito come la differenza tra due valori numerici successivi.

## Esempio:

Ampiezza  
(discreta e codificata)

Codifica a 3 bit

*L'operazione di **codifica** trasforma i valori numerici forniti dal quantizzatore in cifre binarie.*



segnale quantizzato

Tempo (discreto)

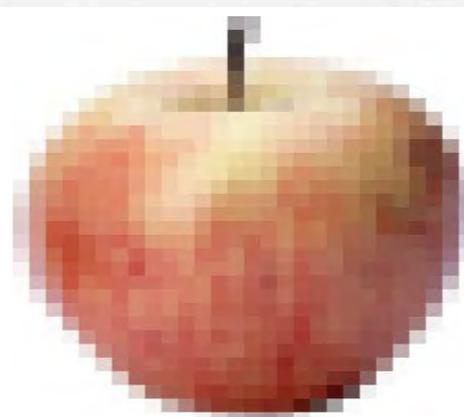
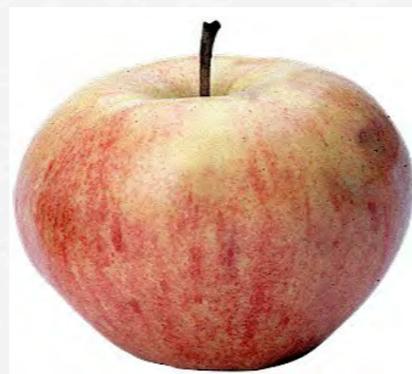
# Rappresentazione delle immagini

# Rappresentazione delle immagini

Le forme e i colori presenti nelle immagini sono grandezze di tipo continuo. E' necessario rappresentare tale grandezze in maniera discreta.

Nel caso delle immagini non è presente la dimensione temporale (come nel suono).

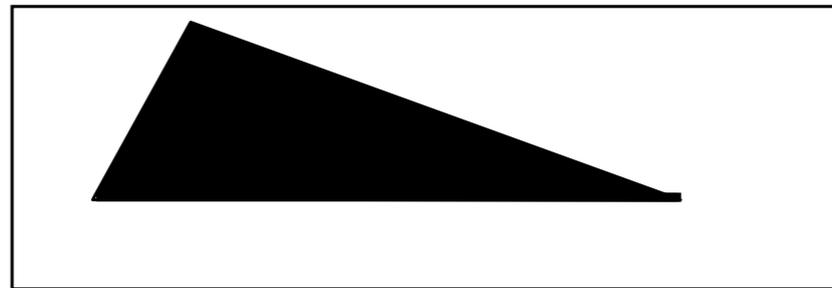
- ▶ Una maniera immediata per rappresentare un immagine è quella di suddividerla in un insieme di piccoli quadratini (**pixel**) e di memorizzare l'informazione relativa al colore presente nel quadratino.
- ▶ Ogni quadratino rappresenta appunto un pezzettino dell'immagine.



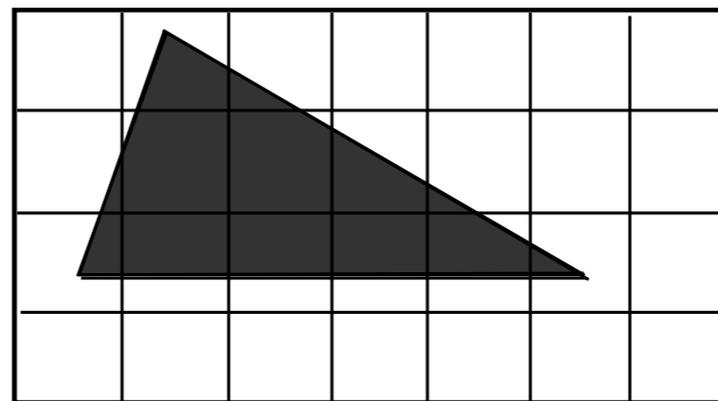
- ▶ Questo tipo di codifica delle immagini viene detto **bitmap** o **raster** (dal latino *rastrum*, rastrello) ad indicare il campionamento sistematico dell'immagine per mezzo dell'insieme di pixel che la rappresentano.

# Un esempio di codifica

Caso più semplice: immagini in bianco e nero senza livelli di grigio



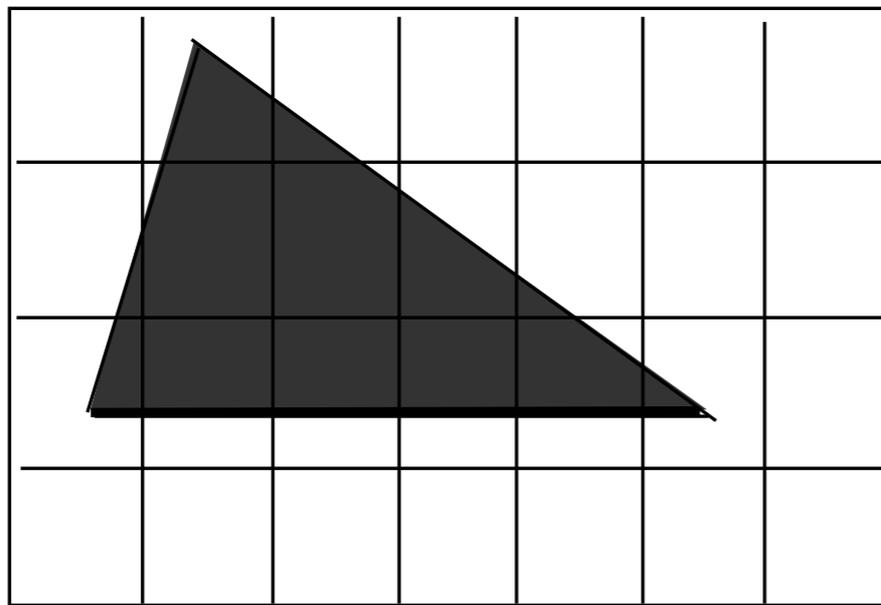
Suddividiamo l'immagine mediante un insieme di quadratini di dimensioni costante (campionamento dell'immagine)



Assumiamo che un pixel sia codificato con un singolo bit che vale  
**0** se nel pixel il *bianco* è predominante  
**1** se nel pixel il *nero* è predominante

# Un esempio di codifica

Poiché una sequenza di bit è lineare, è necessario definire delle convenzioni per ordinare la griglia dei pixel in una sequenza. Assumiamo che i pixel siano ordinati dal basso verso l'alto e da sinistra verso destra



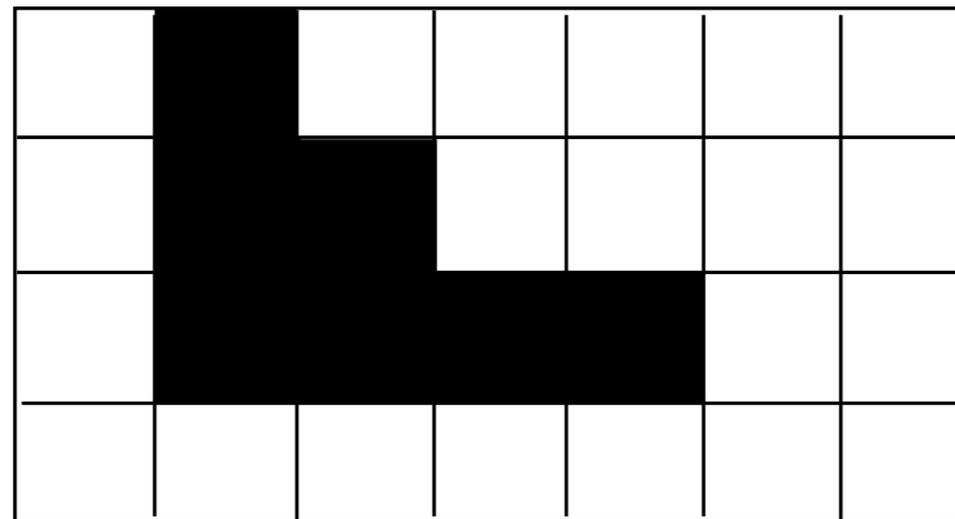
	0	1	0	0	0	0	0
22	23	24	25	26	27	28	
	0	1	1	0	0	0	0
15	16	17	18	19	20	21	
	0	1	1	1	1	0	0
8	9	10	11	12	13	14	
	0	0	0	0	0	0	0
	1	2	3	4	5	6	7

Con questa convenzione la rappresentazione della figura sarà data dalla stringa binaria

**000000 0111100 0110000 0100000**

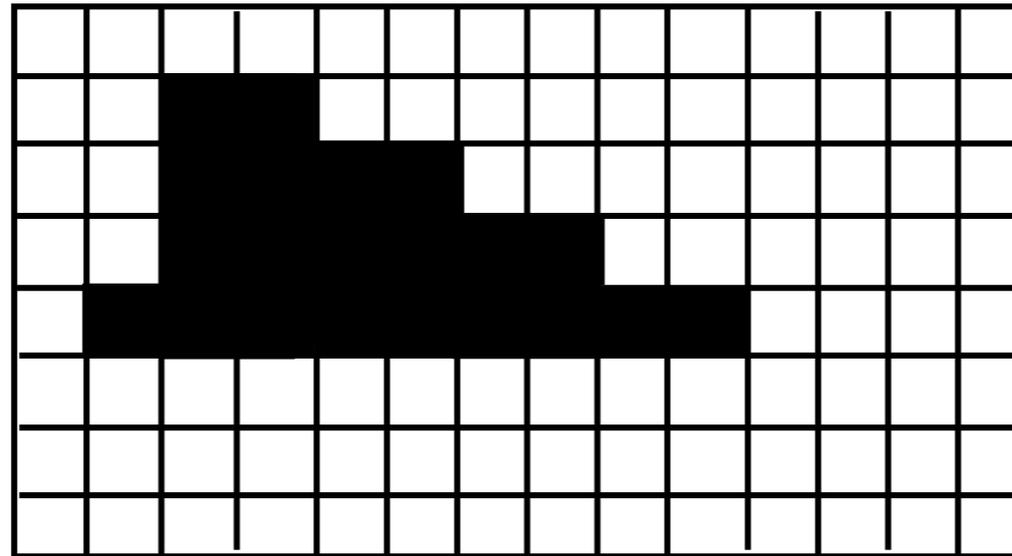
# Risoluzione

- Non sempre il contorno della figura coincide con le linee della griglia. Quella che si ottiene nella codifica è un'approssimazione della figura originaria
- Se riconvertiamo la stringa `0000000011110001100000100000` in immagine otteniamo



# Risoluzione

La rappresentazione sarà più fedele all'aumentare del numero di pixel, ossia al diminuire delle dimensioni dei quadratini della griglia in cui è suddivisa l'immagine:



Il numero di pixel in cui è suddivisa un immagine si chiama **risoluzione** e si esprime con una coppia di numeri ad es. 640 × 480 pixel (orizzontali × per verticali)

# Codifica dei livelli di grigio

Per ogni pixel viene misurato il livello medio di intensità luminosa (il **livello di grigio**): ogni pixel è codificato con un numero di bit  $> 1$ .

Ad esempio:

- se utilizziamo quattro bit possiamo rappresentare  $2^4=16$  livelli di grigio
- se utilizziamo otto bit ne possiamo distinguere  $2^8=256$ , etc.



Confronto tra una immagine a 32 bit ed una a 2 bit

# Codifica di immagini a colori

- ▶ Analogamente possiamo codificare le immagini a colori. In questo caso si tratta di individuare un certo numero di sfumature di colore differenti e di codificare ogni sfumatura mediante un'opportuna sequenza di bit.
- ▶ Ad esempio, i monitor utilizzano **risoluzioni** di 640×480 (VGA), 1024×768 (XGA), oppure 1280×1024 (SXGA) ed un numero di colori per pixel che va da 256 (8 bit) fino a sedici milioni (24 bit, *true color*).
- ▶ Il numero di bit usato per codificare i colori è detto **profondità** dell'immagine.



# Codifica di immagini a colori

La rappresentazione di un'immagine mediante la codifica dei pixel, viene chiamata **codifica bitmap**.

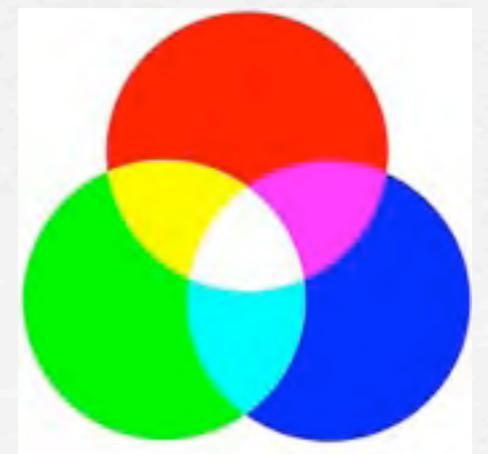
Il numero di byte richiesti dipende dalla risoluzione e dal numero di colori che ogni pixel può assumere.

Per distinguere 256 colori sono necessari otto bit per la codifica di ciascun pixel: la codifica di un'immagine formata da 640×480 pixel richiederà 2.457.600 bit (307.200 byte).



# La codifica RGB

- ▶ Come è possibile rappresentare l'infinità di colori presenti in natura?
- ▶ Un possibile modello di rappresentazione è noto con il nome di **RGB** (Red, Green, Blue), il quale usa questi tre colori primari per rappresentare tutti i possibili colori.
- ▶ Nella codifica RGB ogni pixel è rappresentato da una combinazione di tre numeri, ognuno rappresentante una diversa gradazione di uno dei colori primari.
- ▶ Con 8 bit per colore otteniamo:  $256 \times 256 \times 256 = 16.777.216$  colori diversi.
- ▶ Per ogni pixel sono quindi richiesti 3 byte (24 bit di *profondità*).



# Esempio di codifica RGB

La sfumatura di azzurro è determinata da una certa combinazione di RGB.

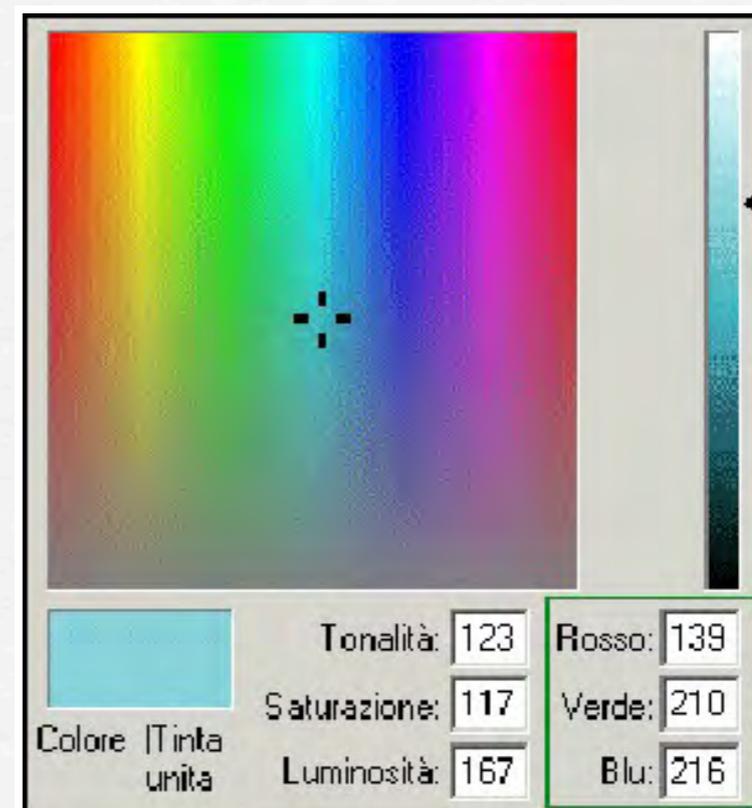
Le sequenze di bit sono espresse in base decimale.

Ogni sfumatura di colore primario è rappresentabile da 1 byte:

$$139_{10} \Rightarrow 10001011_2$$

$$210_{10} \Rightarrow 11010010_2$$

$$216_{10} \Rightarrow 10100111_2$$



# Altri esempi di codifica RGB



R	G	B
0	132	209



R	G	B
102	0	102



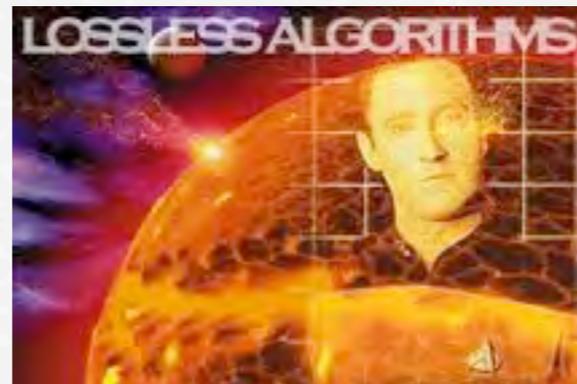
R	G	B
125	166	71

# Compressione delle immagini

# Tecniche di compressione

Esistono tecniche di compressione per ridurre lo spazio occupato dalle immagini. Queste tecniche sfruttano le regolarità delle immagini.

- ❖ **Compressione senza perdita di informazione** (sono dette **loss-less**): si memorizzano pixel vicini identici una volta sola e si ricorda quante volte occorrono nell'immagine.
- ❖ **Compressione con perdita di informazione** (sono dette **lossy**): non si memorizzano tutti i pixel, ma solo una frazione di essi. Si usano funzioni matematiche di interpolazione per ricostruire i pixel mancanti.



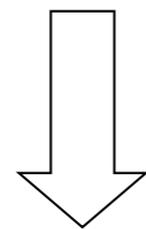


# La compressione loss-less

Immagine1.bmp (2.63 MB)



Il fattore di compressione  
è pari al 71%



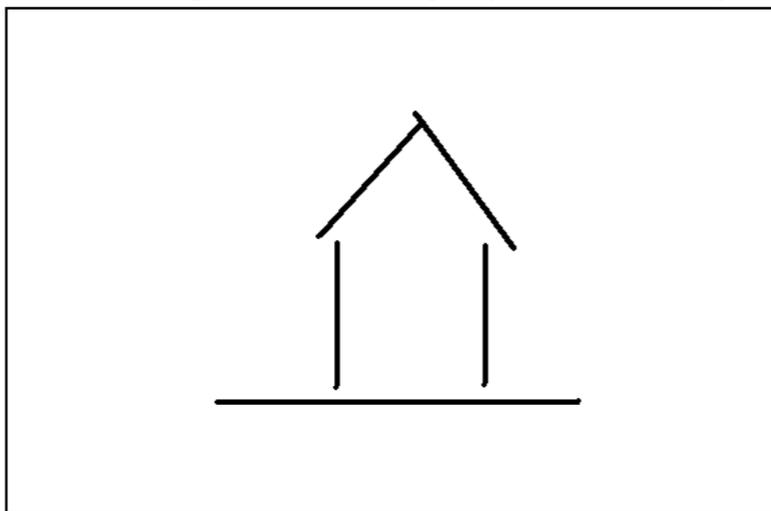
compressione

Immagine1.zip (1.876 MB)

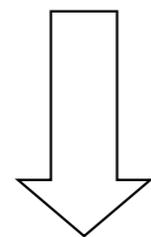


# La compressione loss-less

Immagine2.bmp (2.63 MB)



Il fattore di compressione  
è pari al 0.3%!



compressione

Immagine2.zip (5 KB)



# Formati standard

**GIF** (*Graphic Interchange Format*, brevettato da Unisys) utilizza 8 bit per pixel e quindi distingue 256 colori. Usa una tecnica di compressione senza perdita (algoritmo LZW).

**JPEG** (*Joint Photographic Expert Group*) utilizza 24 bit, quindi 16,8 milioni di colori. Usa una tecnica sofisticata di compressione con perdita.

Altri formati senza perdita sono **PNG** (*Portable Network Graphics*, algoritmo zlib) e **TIFF** (*Tagged Image File Format*).



# Formato JPEG

- ▶ Lo standard JPEG (Joint Photographic Expert Group) è aperto ed è stato sviluppato da un gruppo di esperti di fotografia e supporta 24 bit per pixel (bpp).
- ▶ Lo scopo dello standard è quello di comprimere immagini di tipo fotografico minimizzando la perdita di informazione al quale l'occhio umano è più sensibile.
- ▶ Quando un immagine BMP viene trasformata in una JPEG, il reticolo di pixel dell'immagine JPEG viene suddiviso in blocchi di dimensione 8x8.
- ▶ Ogni blocco 8x8 viene poi rappresentato con dei valori medi.
- ▶ Lo standard prevede diversi livelli di compressione.

*Maggiore è la compressione, minore sarà la qualità dell'immagine*

# Formato GIF

- ▶ Il formato **GIF** (**G**raphic **I**nterchange **F**ormat) riduce l'occupazione su disco di un'immagine limitando il numero di colori che compaiono in essa.
- ▶ Vengono scelti quelli più frequenti, alcune sfumature vengono perse e sostituite dalle sfumature più vicine fra quelle mantenute.
- ▶ Più si limita il numero di colori più l'immagine sarà piccola; il numero può andare da un minimo di 2 ad un massimo di 256 (8 bbb).
- ▶ L'insieme dei colori utilizzati viene salvato insieme all'immagine come palette di colori.
- ▶ Il formato GIF è adatto ad immagini geometriche, possibilmente con un numero di colori non elevato.

# La tavolozza dei colori

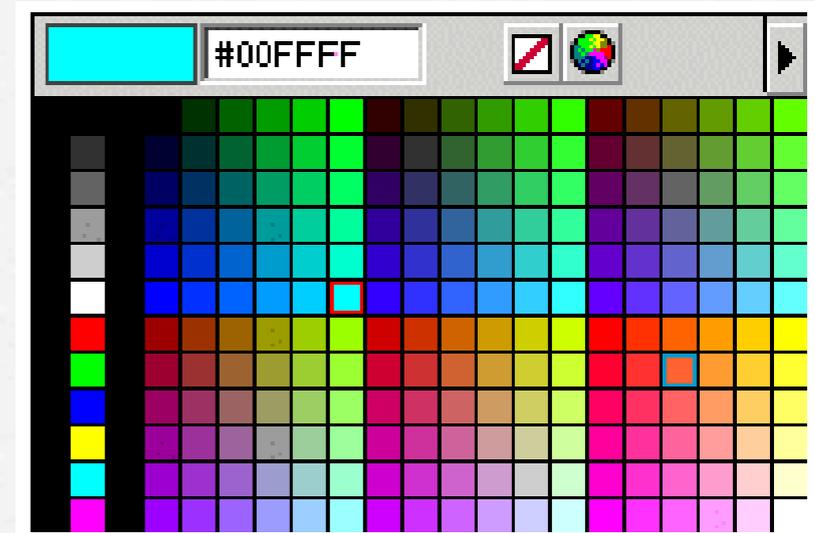
Nella maggior parte delle immagini sono presenti un numero ridotto di colori.

Questo fatto può essere sfruttato per costruire una tavolozza dei colori (*colour palette*).

Questa tavolozza non è altro che un elenco dei colori presenti nell'immagine.

Questo può portare ad un notevole risparmio di spazio.

Ad esempio: se la tavolozza contiene 256 colori, posso utilizzare un byte per codificare i colori, ottenendo così un fattore di compressione pari a 3.



# Formati standard: un esempio



Formato gif 93K



Formato jpeg 30K

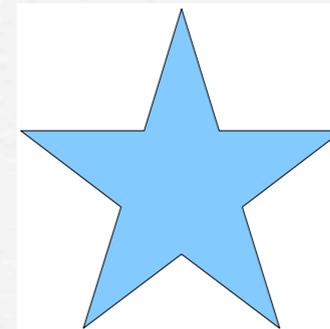
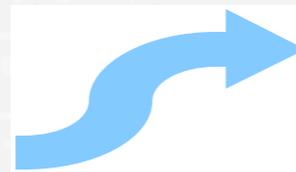
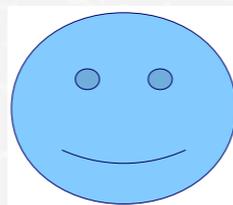
# Grafica vettoriale

# La grafica vettoriale

- ▶ Nella grafica vettoriale le immagini sono rappresentate per mezzo di **primitive geometriche**.
- ▶ Esempi di queste primitive sono: segmenti, curve, poligoni, cerchi, ellissi.
- ▶ Queste primitive sono a loro volta rappresentate per mezzo di equazioni matematiche.

## ***Esempio***

- Un oggetto di tipo circolare può essere rappresentato specificando le coordinate del centro e il suo raggio.
- Ogni oggetto possiede inoltre un insieme di attributi che ne specificano il colore, il riempimento, il bordo etc.



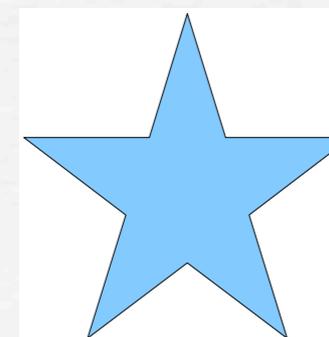
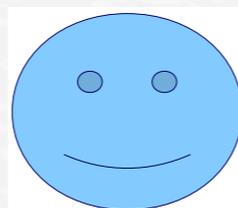
# La grafica vettoriale

## Vantaggi

- Gli oggetti presenti in un'immagine possono essere modificati (ruotati, allungati, etc.) in maniera immediata per mezzo di semplici operatori matematici.
- Un immagine può essere rappresentata in maniera compatta specificando solo le proprietà degli oggetti presenti.
- È indipendente dalla risoluzione.

## Svantaggi

- La visualizzazione delle immagini richiede l'uso di software complessi in grado di risolvere le equazioni associate alle primitive per tracciare i punti (pixel) che le soddisfano.
- Per immagini molto complesse può essere necessario l'uso di computer molto potenti.



# Formati vettoriali

- ▶ **CDR** (il formato di CorelDraw)
- ▶ **SWF** (ShockWave Flash, il formato di Macromedia Flash, utilizzato per la creazione di animazioni destinate al web. Richiede il plug-in Flash Player)
- ▶ **SVG** (Scalable Vector Graphics, formato standard del W3 Consortium, creato nel 1999, è visualizzabile dai browsers ma necessita del plug-in Adobe SVG Viewer. Come il formato di Flash, permette di creare delle animazioni. Si basa sul linguaggio XML.)
- ▶ **AI** (Adobe Illustrator)
- ▶ **FH** (Freehand)
- ▶ **DFX** (Drawing Exchange Format)

Formati ibridi (possono essere usati sia per le immagini vettoriali che raster):

- ▶ **EPS** (Encapsulated Postscript, utilizzato nel campo della stampa professionale)
- ▶ **PDF** (Portable Document Format. E' il formato visualizzabile con Adobe Acrobat Reader)
- ▶ **PSD** (il formato di Adobe Photoshop)

# Confronto vettoriale-raster

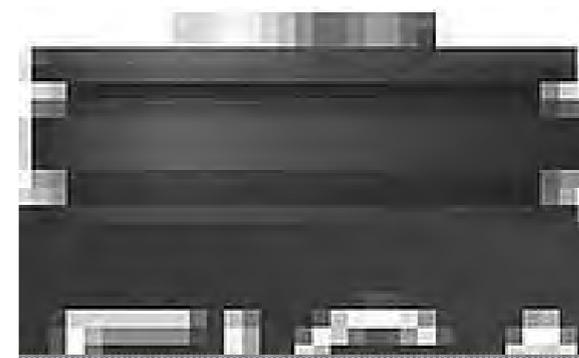
Immagine originale



Ingradimento  
vettoriale



Ingradimento  
raster

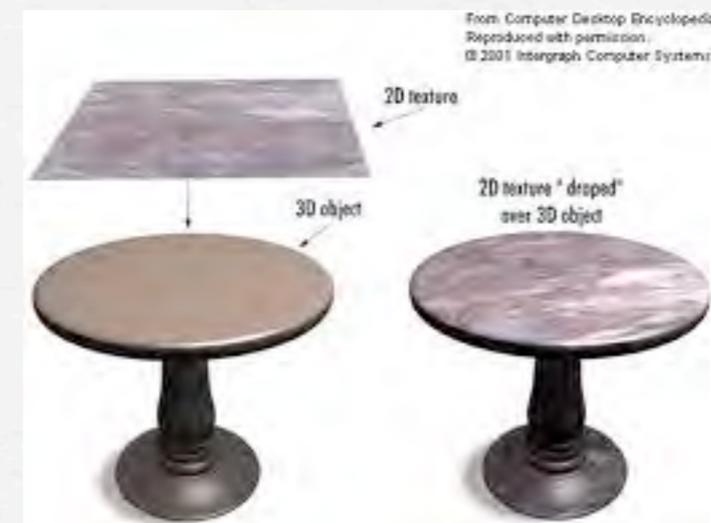


# Effetti



- ▶ Il **ray tracing** è una particolare tecnica di ombreggiatura.

- ▶ Il **texture-mapping** è un processo che consiste nell'applicare ad una superficie vettoriale un'immagine bitmap.



- ▶ L'**anti-aliasing** ammorbidisce i contorni di un oggetto.

# Font scalabili

## Tipi di carattere e font

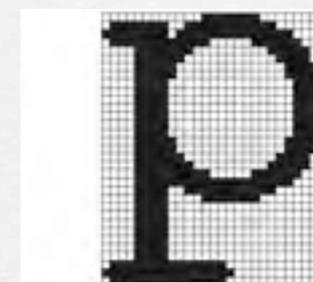
Un font è un insieme di caratteri e simboli creati in base a un particolare modello (tipo di carattere). La scelta dei tipi di carattere più consoni alle varie parti di un documento facilita la lettura di quest'ultimo.

## Font bitmap e font scalabili

Le stampanti utilizzano sia font scalabili che font bitmap.

I font bitmap vengono memorizzati come modelli di bit predefiniti che rappresentano un tipo di carattere con dimensione, stile e risoluzione specifici.

I font scalabili (chiamati anche font outline) sono gestiti da un programma detto "rasterizzatore" e sono memorizzati come programmi (serie di formule) che definiscono i profili dei caratteri di un font. Ogni volta che si stampano dei caratteri di un font scalabile, la stampante crea una bitmap dei caratteri con la dimensione in punti selezionata e la salva temporaneamente nella memoria.



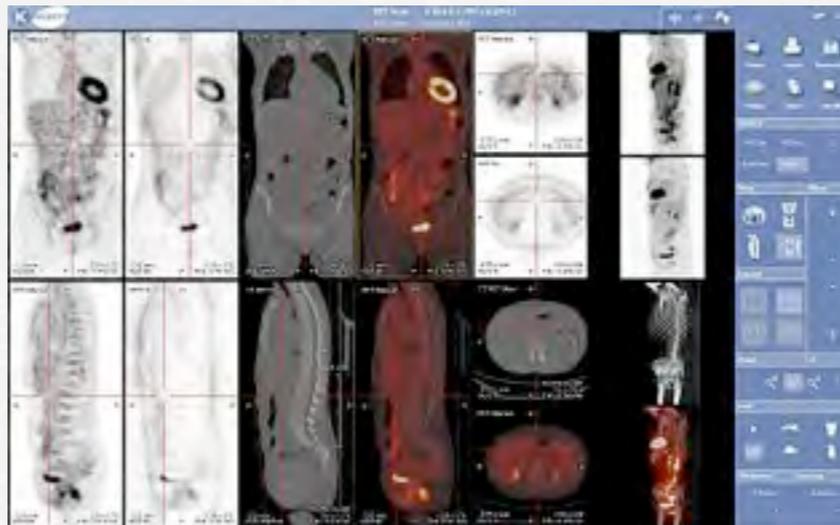
# Immagini radiologiche



# PACS

**PACS** è l'acronimo anglosassone di *Picture archiving and communication system* e consiste in un sistema hardware e software dedicato all'archiviazione, trasmissione e visualizzazione delle immagini diagnostiche digitali.

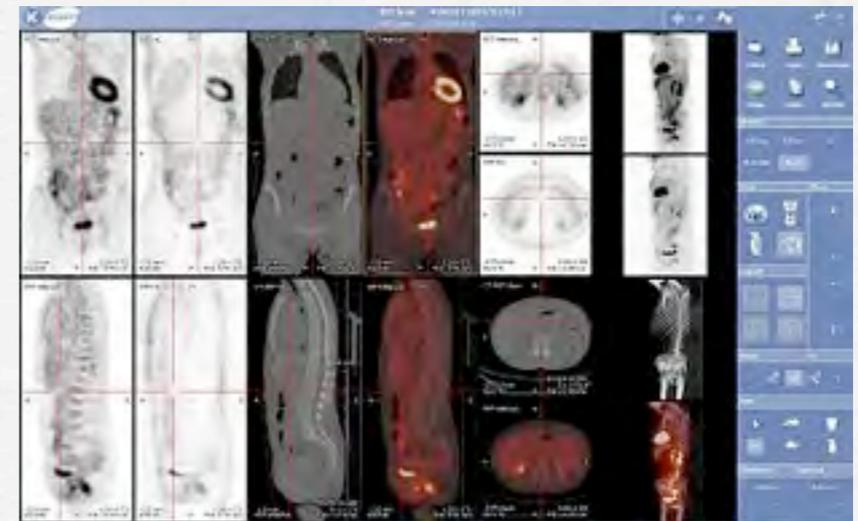
Un sistema PACS è normalmente composto da una parte di **archiviazione**, utilizzata per gestire dati e immagini e una di **visualizzazione**, che presenta l'immagine diagnostica su speciali monitor ad altissima risoluzione, sui quali è possibile effettuare la diagnosi; i sistemi PACS più evoluti permettono anche l'elaborazione dell'immagine, come per esempio le ricostruzioni 3D.



# PACS

Una parte fondamentale ma non visibile dall'utente finale si occupa del colloquio con gli altri attori del flusso radiologico, utilizzando di solito i relativi profili **IHE** (*Integrating the Healthcare Enterprise*) tramite lo standard **HL7** (*Health Level 7*).

In special modo, è fondamentale la sua integrazione con il **sistema informatico radiologico** o **RIS** (*Radiology Information System*), che rappresenta il software gestionale della Radiologia.



# PACS/DICOM



- ❖ Le immagini sono ricevute e trasmesse nel formato definito da **DICOM** (*Digital Imaging and Communications in Medicine*), che permette di inglobare e trattare anche testo (per esempio i referti) e documenti di vario genere, tra cui il formato PDF.
- ❖ I visualizzatori collegati sono in genere in grado di mostrare immagini e referti, ma anche di riconoscere i tipi di immagine e comportarsi di conseguenza: ad es. applicando determinati filtri di contrasto o mostrandole in un modo predefinito.

# PACS/DICOM

- ◆ I **sistemi PACS**, in origine creati per gestire le immagini generate dalle TAC (i primi sistemi diagnostici nati in digitale) oggi sono in grado di trattare **tutte le immagini radiologiche** digitali e, tramite speciali digitalizzatori, anche quelle create da modalità analogiche.
- ◆ Da notare che le immagini ricevute non devono essere modificate in alcun modo, per poter sempre risalire all'originale trasmesso dalla modalità; l'eventuale elaborazione viene registrata in aggiunta alle altre immagini.
- ◆ Di solito è ammessa una **compressione** senza perdita di dati (*lossless*) per diminuire lo spazio occupato su disco.
- ◆ Proprio per **garantire** che ogni immagine immagazzinata nel PACS sia effettivamente quella generata dalla modalità durante l'esame, spesso il PACS spedisce tutti gli oggetti DICOM ad un sistema di **archiviazione legale**.



# PACS/DICOM

- *Recentemente, con l'evoluzione della tecnologia delle reti, sempre più sistemi PACS stanno passando ad una **architettura di tipo web**, dove l'applicazione risiede su un server, permettendo un semplice accesso alle immagini con il solo utilizzo di un browser sul proprio computer, senza necessità di installazioni specifiche.*
- Per la semplice **distribuzione delle immagini** sia nei reparti che all'esterno dell'ospedale il computer può essere un normale desktop, mentre per la **diagnosi** la stazione di lavoro dovrà avere sufficiente RAM per contenere tutte le immagini sotto esame e una appropriata scheda grafica, in grado di pilotare i monitor diagnostici ad alta risoluzione (anche fino a 5 Megapixel, per gli esami mammografici), oltre ad un processore potente, per la veloce manipolazione di immagini che possono raggiungere i 20 MB l'una.



# PACS/DICOM

In origine, le immagini venivano archiviate immediatamente su memoria di massa locale ad accesso veloce (*on-line*) e lì tenute per un tempo variabile tra 3 e 6 mesi, e successivamente su DVD all'interno di un *juke-box*, da dove potevano essere richiamate in automatico in caso di necessità senza intervento umano (*near-line*), ma con tempi di risposta notevolmente superiori.

In seguito, i DVD venivano periodicamente tolti dal juke-box e, contrassegnati da un codice generato dal sistema, immagazzinati in armadi ignifughi (*off-line*): in caso di necessità, gli esami potevano essere immessi nuovamente nel sistema, ovviamente con intervento umano e tempi che non potevano essere minori di qualche ora.

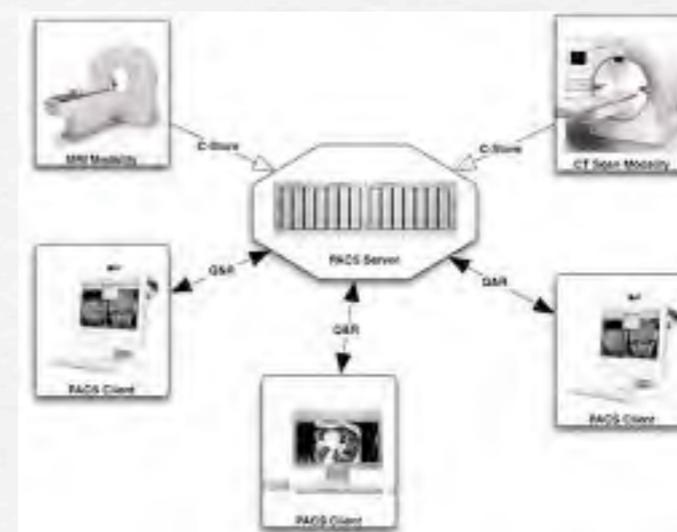
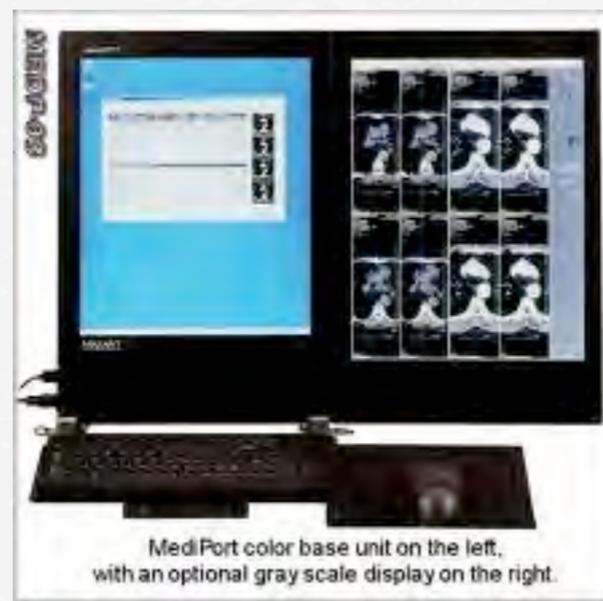
Con la diminuzione dei costi delle memorie di massa, è diventata prassi mantenere tutte le immagini nella memoria ad accesso immediato (*everything-on-line*) cioè su hard-disk; questo, assieme alle crescenti velocità delle reti, permette un tempo di accesso alle informazioni dell'ordine dei secondi per una singola immagine.



# PACS/DICOM

Come detto, un tipico sistema PACS è in grado di gestire solo oggetti DICOM; tali oggetti contengono al loro interno, oltre all'immagine, anche i **dati relativi al paziente e all'esame** a cui si riferiscono.

Il sistema PACS registra questi (meta)dati quando riceve le immagini e li utilizza quando gli viene richiesta una lista di esami o pazienti, invece di accedere ogni volta agli oggetti DICOM; in questo modo, tutte le **ricerche sono effettuate su un archivio testuale**, ricorrendo a quello DICOM solo quando è necessario visualizzare o comunque spostare le immagini.

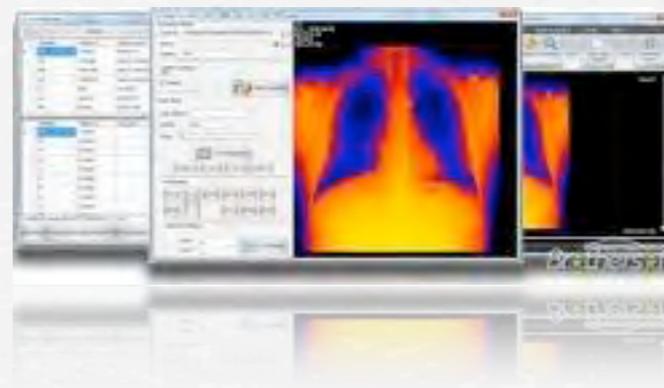


# PACS/DICOM



L'architettura della parte hardware viene progettata *ad hoc* per ogni situazione, in quanto può dipendere dal numero di ospedali coinvolti, dal loro carico di lavoro e dalle politiche di backup necessarie per mantenere la continuità del servizio.

L'archivio DICOM *on-line* è di solito registrato su memorie di massa su sistemi SAN o NAS, spesso configurati in RAID o con architettura ridondante. Ogni disco può essere sostituito in caso di problemi senza interrompere il funzionamento del sistema (*hot-swap*).



# I formati audio



# Digitalizzazione audio

- ▶ Un suono digitalizzato con qualità **CD-DA** viene campionato a 44,1 KHz su 16 bit (2 canali stereo).
- ▶ I formati audio si dividono in **lossy** (con perdita) e **no lossy** (o **lossless**, senza perdita di qualità).
- ▶ L'idea dei formati lossy (i più diffusi tra gli utenti comuni) nasce dall'idea che non tutti i suoni presenti nei 44.100 campioni al secondo contenuti in un file WAV standard vengono correttamente percepiti dall'orecchio umano.



# Digitalizzazione audio

- ▶ In questo modo (così come avviene per il JPEG delle immagini) si vanno a tagliare le alte frequenze, che si ritiene siano meno distinte dal nostro orecchio.
- ▶ Più si tagliano queste frequenze, più lo spazio occupato dalla nostra traccia diminuisce ma diminuisce anche la qualità del risultato.
- ▶ I formati senza perdita (**lossless**, come ad es. WAV) invece cercano di diminuire lo spazio occupato dalla traccia senza andare a toccare il suono; la percentuale di compressione sarà decisamente inferiore rispetto ai lossy, ma non ci sarà perdita di qualità.
- ▶ Se riconvertito da WAV (magari per essere elaborato), il suono sarà **identico** all'originale.



# Digitalizzazione audio

- Lo standard **MIDI** (Musical Instrument Digital Interface) viene utilizzato per uniformare le comunicazioni tra gli elaboratori e gli strumenti musicali elettronici.
- **WAV** (contrazione di WAVEform) è un formato audio sviluppato da Microsoft e IBM per personal computer IBM compatibili:
  - È una variante del formato **RIFF**<sup>(\*)</sup> di memorizzazione dei dati.
  - I dati vengono salvati in "chunk" (blocchi).
  - È simile anche al formato **AIFF**<sup>(+)</sup> utilizzato dai computer Apple Mac.



(\*) RIFF = Resource Image File Format, una specifica Microsoft per la memorizzazione di file multimediali.

(+) AIFF = Audio Interchange File Format

# Formato MP3

- Il formato **MP3** usa una compressione con perdita di informazioni.
- **MP3** (noto anche come *MPEG Audio Layer III*) è un algoritmo di compressione audio di tipo lossy, sviluppato dal gruppo MPEG (Motion Picture Expert Group), in grado di ridurre drasticamente la quantità di dati richiesti per memorizzare un suono, rimanendo comunque una riproduzione accettabilmente fedele del file originale non compresso.
- Il *bitrate* per un file in formato MP3 indica il numero medio di bit per un secondo di dati audio. Tipicamente minimo 128 Kbps per una qualità accettabile.



# Formato AAC

- 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 mantenendo la stessa dimensione di compressione.
- Attualmente viene utilizzato principalmente da **Apple** nei suoi prodotti dedicati all'audio (*iTunes*); difatti Apple usa sia una variante dell'AAC che gestisce i diritti d'autore DRM (AAC Protected), con compressione a 128 Kbps, sia una versione senza protezione (AAC Plus), con compressione a 256 Kbps.
- L'AAC è diventato recentemente il formato standard di audio per le console PlayStation 3, Nintendo DSi e Wii.



# Formati AC3, Real

**AC3.** E' il formato audio usato dai DVD. In genere lo troviamo a 384 kbps (e 6 canali), ma è possibile anche averlo in soli 2 canali e a bitrate inferiori. Uno dei programmi gratuiti che lo supporta (e che consente per esempio di ridurre il bitrate) è *BeLight*.

**Real Player** (.ra, .rm e altre estensioni simili). Codec molto usato in certi video e in certi audio. RealPlayer è un programma gratuito (con certi limiti...) che consente di eseguire tracce audio e video create con il suo codec.

Una valida alternativa, *davvero* gratuita, a Real Player è rappresentata dal programma *Real Alternative*.



# Immagini in movimento



# Immagine in movimento

- Memorizzazione mediante sequenze di **fotogrammi**.
- La qualità della memorizzazione dipende dal numero di **fotogrammi al secondo**.

## Esempio:

- le immagini televisive vengono trasmesse con 25 fotogrammi al secondo
- poniamo di avere una risoluzione di  $1024 \times 768$ , in formato JPEG
- se ogni immagine è 200 Kbyte, dieci minuti di filmato occupano 3 Gbyte



# Compressione immagini in movimento

Problema dell'occupazione di spazio: per ottimizzare lo spazio non si memorizzano tutti i fotogrammi.

I fotogrammi variano in modo continuo: si memorizza un primo fotogramma in modo completo, e per i successivi  $N$  solo le differenze con il primo (compressione **inter-fotogramma**).

Anche il singolo fotogramma completo viene compresso (come un'immagine) per ridurre l'occupazione (compressione **intra-fotogramma**).

I film su DVD usano la compressione MPEG-2.



# Codec video

Il processo di compressione consiste nell'applicare un algoritmo al video di origine allo scopo di creare un file compresso pronto per la trasmissione o la memorizzazione.

Al momento della riproduzione del file compresso, viene invece applicato un algoritmo inverso che genera un video contenente praticamente lo stesso contenuto del video originale.

Il tempo richiesto per comprimere, trasmettere, decomprimere e visualizzare un file rappresenta la cosiddetta **latenza**. Più avanzato è l'algoritmo di compressione, più alta è la latenza.

Una coppia di algoritmi utilizzata contemporaneamente rappresenta un **codec** (codificatore/decodificatore) **video**.

I codec video di standard diversi non sono generalmente compatibili tra loro, il che significa che il video compresso con uno standard non può essere normalmente decompresso usando un altro standard.



*Ad esempio, un decodificatore che supporta lo standard MPEG-4 non può essere utilizzato con un codificatore che supporta lo standard H.264, semplicemente perché uno dei due algoritmi non è in grado di decodificare l'output dell'altro algoritmo correttamente; tuttavia, è possibile implementare più algoritmi nello stesso software o hardware e consentire la coesistenza di più formati.*

# Confronto tra compressione delle immagini e compressione video

- ▶ I vari standard di compressione utilizzano metodi diversi per ridurre i dati e offrono, quindi, velocità di trasmissione in bit, qualità e latenze diverse.
- ▶ Gli algoritmi di compressione si suddividono in due tipi: compressione delle immagini e compressione video.
- ▶ La compressione delle immagini utilizza la tecnologia di codifica intra-fotogramma. I dati vengono ridotti all'interno di un fotogramma immagine semplicemente rimuovendo le informazioni non necessarie che potrebbero essere non visibili all'occhio umano.
- ▶ Motion JPEG (**MPEG**) è un tipico esempio di standard di compressione di questo tipo. In una sequenza Motion JPEG le immagini sono codificate o compresse come singole immagini JPEG.

*Nel formato Motion JPEG le tre immagini della sequenza mostrata vengono codificate e trasmesse come immagini univoche distinte (fotogrammi di tipo I) senza dipendenze tra loro.*



# Compressione inter-fotogramma

Gli algoritmi di compressione di video, quali MPEG-4 e H.264, utilizzano la predizione inter-fotogramma per ridurre i dati video tra una serie di fotogrammi.

*Ciò implica tecniche come ad esempio la codifica differenziale, dove ciascun fotogramma viene confrontato con quello di riferimento e vengono codificati solo i pixel modificati rispetto al fotogramma di riferimento.*

Pertanto, il numero dei valori dei pixel codificati e trasmessi risulta significativamente ridotto. Quando si visualizza una sequenza codificata di questo tipo, le immagini vengono riprodotte come nella sequenza video originale.

*Nella codifica differenziale, viene codificata interamente solo la prima immagine (fotogramma di tipo I). Per le due immagini successive (fotogrammi di tipo P), vengono stabiliti dei collegamenti agli elementi statici della prima immagine, come la casa. Solo gli oggetti in movimento, come l'uomo che corre, vengono codificati usando vettori di movimento, in modo da ridurre la quantità di informazioni trasmesse e memorizzate.*

— Trasmesso

— Non trasmesso



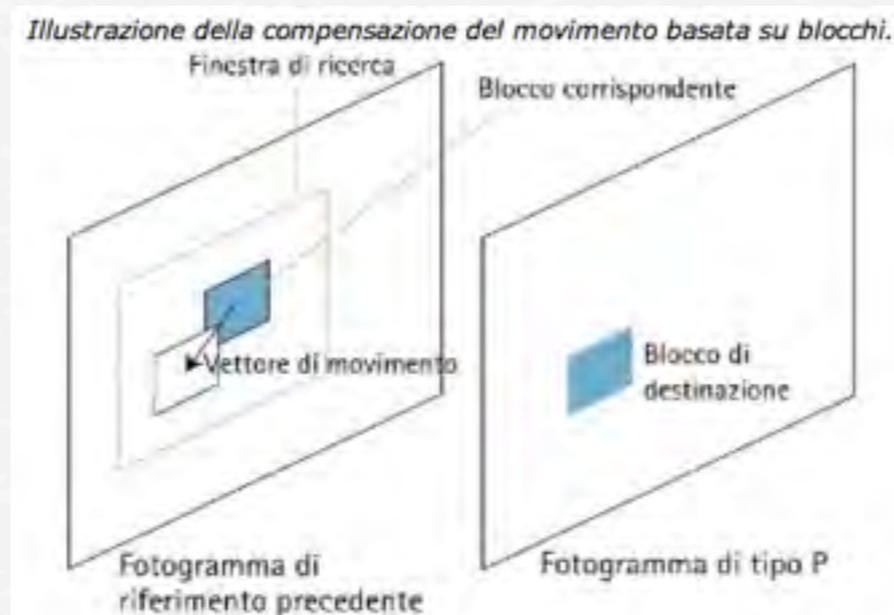
# Compensazione del movimento basata su blocchi

Per ridurre ulteriormente i dati, è possibile utilizzare altre tecniche come la **compensazione del movimento basata su blocchi**.

Questa tecnica individua la parte di un nuovo fotogramma di una sequenza video che corrisponde a quella di un fotogramma precedente, anche se riferito a una posizione diversa, e divide il fotogramma in una serie di macroblocchi (blocchi di pixel).

Ciò consente di creare o “prevedere”, blocco dopo blocco, un nuovo fotogramma, ricercando un blocco corrispondente in un fotogramma di riferimento.

Se si rileva una corrispondenza, il codificatore codifica la posizione in cui si trova il blocco corrispondente all'interno del fotogramma di riferimento. La cosiddetta codifica del vettore di movimento richiede una quantità inferiore di bit rispetto alla codifica del contenuto effettivo del blocco.

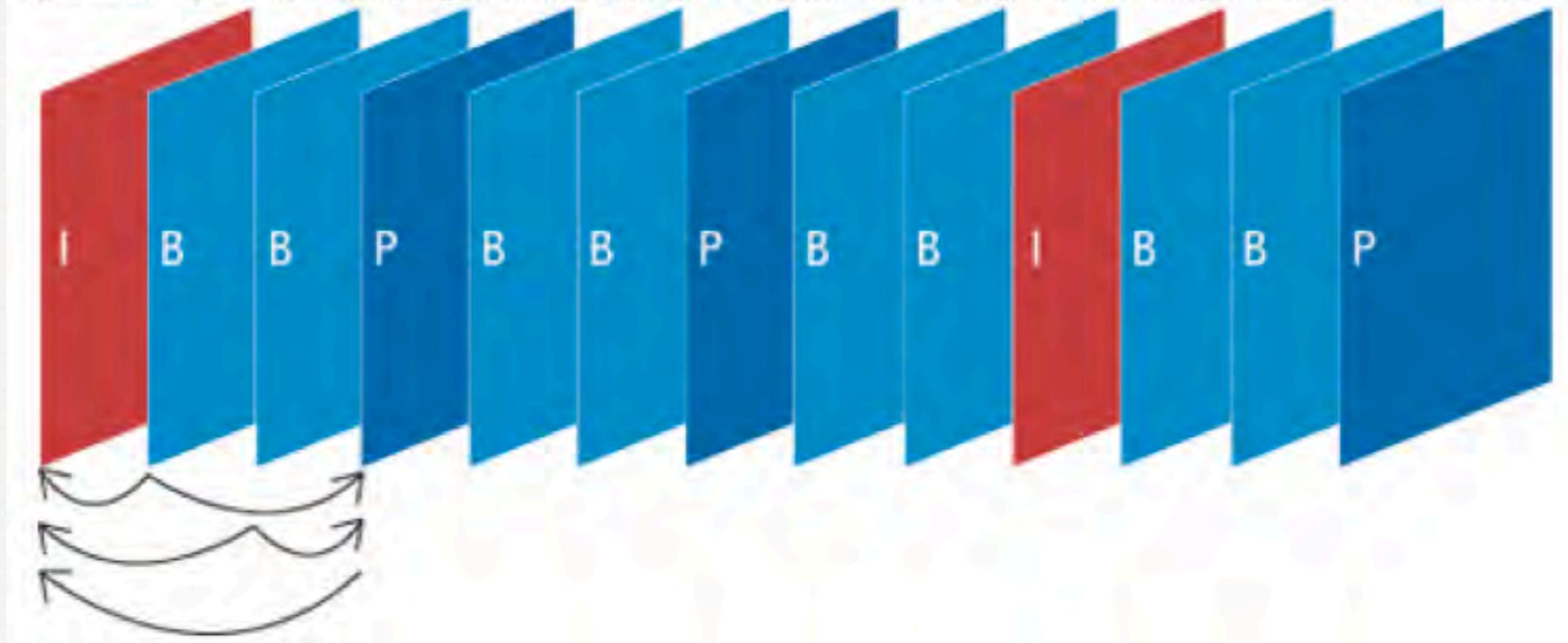


# Fotogrammi di tipo I, P o B

- Con la predizione inter-fotogramma, ciascun fotogramma di una sequenza di immagini viene classificato come un determinato tipo di fotogramma, ad esempio I, P o B.
- Un **fotogramma di tipo I** o intra-fotogramma è un fotogramma a sé stante che può essere decodificato in modo indipendente senza fare riferimento ad altre immagini. La prima immagine di una sequenza video è sempre rappresentata da un fotogramma di tipo I. Questo tipo di fotogrammi viene usato come punto iniziale per i nuovi visualizzatori o come punto di risincronizzazione nel caso in cui il flusso in bit trasmesso risulti danneggiato. I fotogrammi di tipo I possono essere usati anche per le funzioni di avanzamento veloce, il riavvolgimento e altre funzioni di accesso. I codificatori inseriscono automaticamente fotogrammi di tipo I a intervalli regolari o su richiesta, se il flusso video deve essere visualizzato su nuovi client. Lo svantaggio di questo tipo di fotogrammi è rappresentato dal fatto che richiedono una maggiore quantità di bit, ma non producono molti artefatti, causati da dati mancanti.
- I **fotogrammi di tipo P**, che sta per inter-fotogramma “predittivo”, fa riferimento al cambiamento a sezioni di fotogrammi I e/o P precedenti per codificare il fotogramma. Questo tipo di fotogramma richiede generalmente una quantità inferiore di bit rispetto ai fotogrammi di tipo I, ma in compenso è molto sensibile agli errori di trasmissione a causa della stretta dipendenza dai fotogrammi P e/o I precedenti.
- Un **fotogramma di tipo B**, o inter-fotogramma “bi-predittivo”, fa riferimento sia a un fotogramma precedente che a un fotogramma successivo. L’uso di fotogrammi di tipo B aumenta la latenza.

# Fotogrammi di tipo I, P o B

*Sequenza tipica con fotogrammi I, B e P. Un fotogramma P può fare riferimento solo a fotogrammi I o P precedenti, mentre un fotogramma B può fare riferimento a fotogrammi I o P precedenti e successivi.*



Quando un decodificatore video ripristina il video decodificando il flusso di bit un fotogramma alla volta, la **decodifica** deve sempre iniziare con un fotogramma di tipo I.

I fotogrammi di tipo P e B, se utilizzati, devono essere decodificati insieme ai fotogrammi di riferimento.