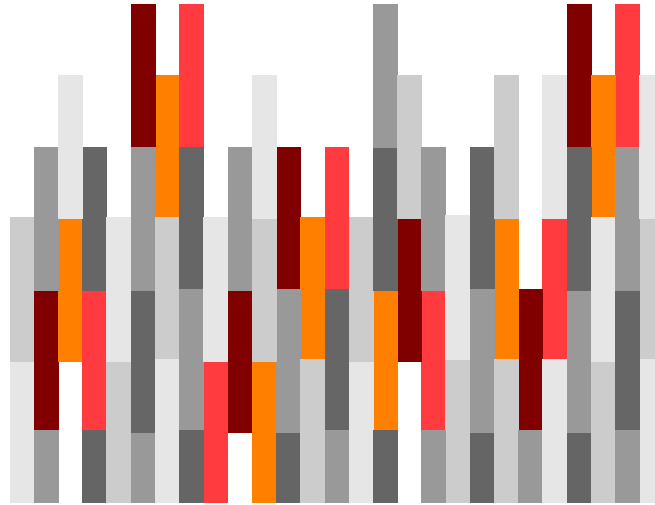


Fluency



Multimedia

Capitolo 8

Digitalizzare i colori

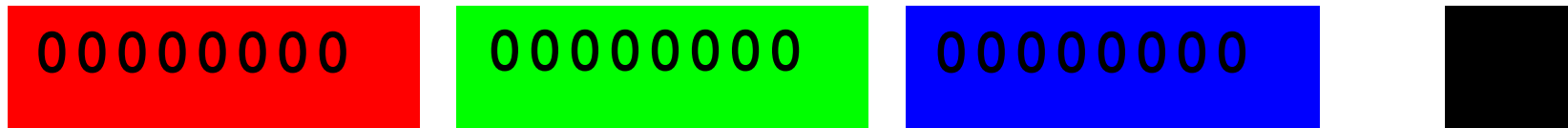
- Rappresentazione dei **colori RGB**
 - un colore è **specificato** da tre componenti: rosso, verde e blu
 - **l'intensità** di ogni colore è rappresentata da una quantità (da 0 a 255)

Intensità dei colori

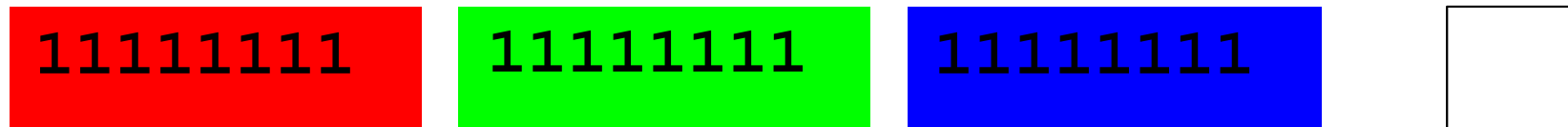
- Ogni intensità RGB è rappresentata da **un byte**
 - l'intensità più bassa è 0000 0000
 - la più alta è 1111 1111
- L'intervallo dei valori va da 0 a 255

Colori nero e bianco

Il **nero** corrisponde all'assenza di colore

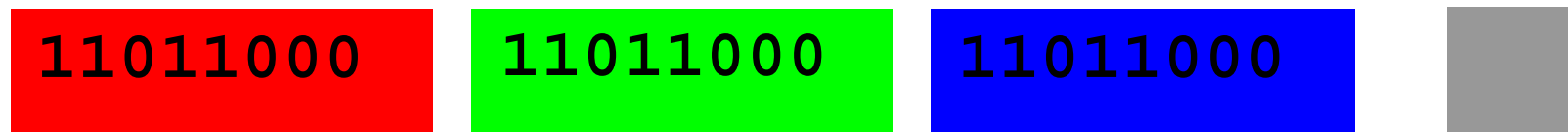


Il **bianco** corrisponde alla massima intensità di tutti i componenti



Grigi

Un **grigio** corrisponde alla stessa intensità di tutti i componenti



ogni byte corrisponde $(152)_{10}$

i colori con componenti RGB uguali sono il **nero**, il **bianco** e i **grigi**

Modificare un colore

- Aumentare l'intensità con un'addizione binaria
- P.e. per ottenere un grigio più chiaro,
 - incrementare i valori per avvicinarsi al bianco

Addizione binaria

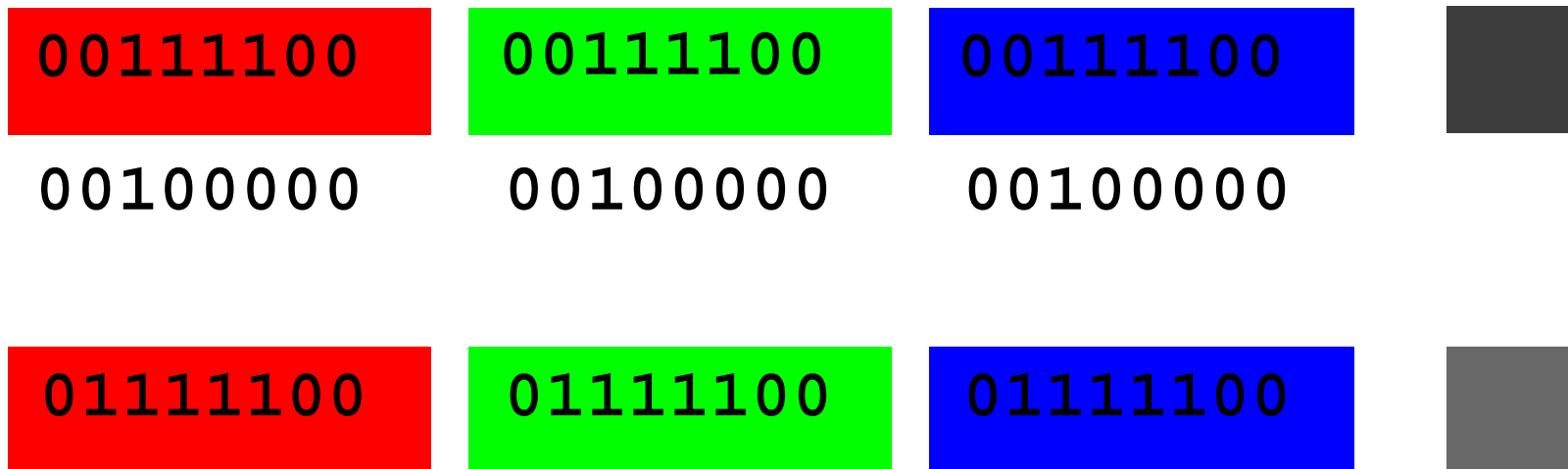
- È simile all'addizione decimale
 - da destra a sinistra sommando le cifre nelle colonne
- La somma di una colonna può
 - essere di una **sola cifra**: $1+0=1$
 - causare un **riporto** $1+1=10$, 0 con riporto di 1

Somma di 64

```
00111100
+01000000
-----
01111100
```

binario, che rappresenta il decimale
binario, che rappresenta il decimale
binario, che rappresenta il decimale

```
60
64
-----
124
```



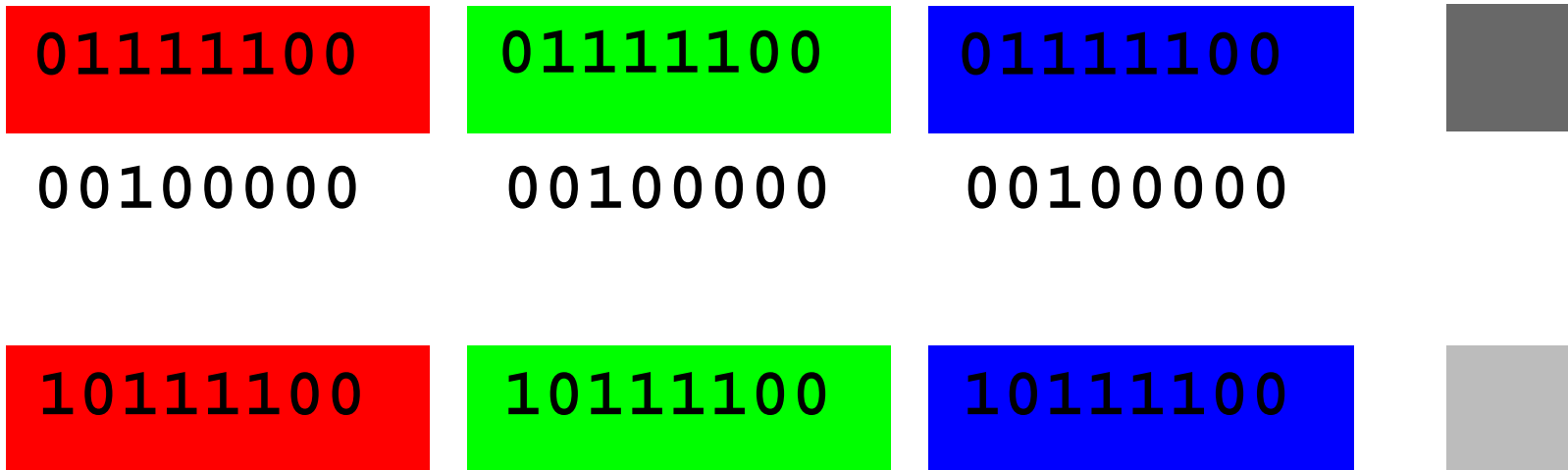
Ancora somma di 64

riporto

$$\begin{array}{r} 01111100 \\ +01000000 \\ \hline 10111100 \end{array}$$

binario, che rappresenta il decimale
binario, che rappresenta il decimale
binario, che rappresenta il decimale

$$\begin{array}{r} 124 \\ 64 \\ \hline 188 \end{array}$$



Overflow

- I computer utilizzano sequenze di bit di dimensioni prefissate
- Cosa succede quando c'è il **riporto** del bit più a sinistra?

Overflow

- Errori di overflow
 - il computer segnala quando il calcolo in esecuzione genera un *overflow error*;
- Sta al programmatore
 - prevederlo e
 - gestirlo

Colorare la luna

- Immaginiamo di avere una foto digitale, in bianco e nero, della luna

- I pixel della foto sono rappresentati da un byte RGB e se volessimo farne una versione colorata?

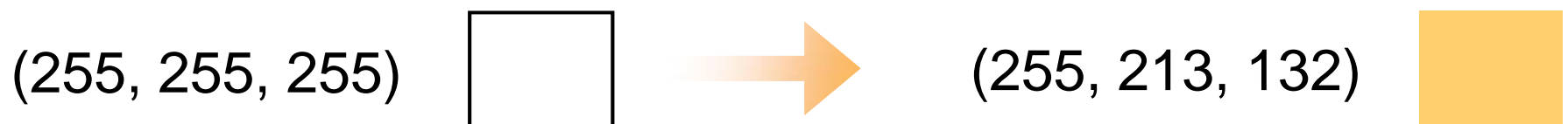
- Che valori hanno i pixel:
 - solo nero, bianco e grigi

Luna in bianco e nero



Colore del bianco

- Da bianco a (255, 213, 132)
- Non modifica i pixel grigi



Da bianco ad arancio

- Il byte rosso: rimane immutato



(255, 255, 255)

- Il byte verde: lo riduciamo di poco

- sottraiamo 42 (101010 binario)



(255, 213, 255)

- Il byte blu: lo riduciamo parecchio

- sottraiamo 123 (1111011 binario)



(255, 213, 132)

Colore dei grigi chiari

- Il byte rosso: rimane immutato



(218, 218, 218)

- Il byte verde: lo riduciamo di poco

- sottraiamo 42 (101010 binario)



(218, 176, 218)

- Il byte blu: lo riduciamo parecchio

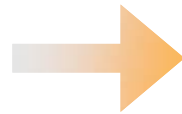
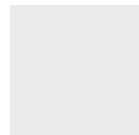
- sottraiamo 123 (1111011 binario)



(218, 176, 95)

Colore dei grigi chiari

(234, 234, 234)

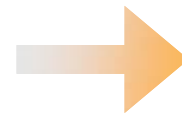
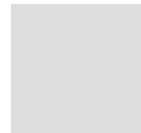


(234, 192, 111)



arancione
grigiastro

(228, 228, 228)



(228, 186, 105)



Luna arancio



Luna “spenta”

- Le parti grigie della luna devono essere *più luminose*
- Occorre *intensificare il rosso*

Intensificare il rosso

- Metodo troppo contrastato
 - assegnare 255 al componente rosso dei pixel arancioni è *troppo*
- Metodo bilanciato
 - assegnare al componente rosso dei pixel arancioni
 - metà della differenza con il rosso 255

Intensificare il rosso

(218, 176, 95)



(237, 176, 95)



(228, 186, 105)



(244, 186, 105)



Luna rossa



I passaggi fatti



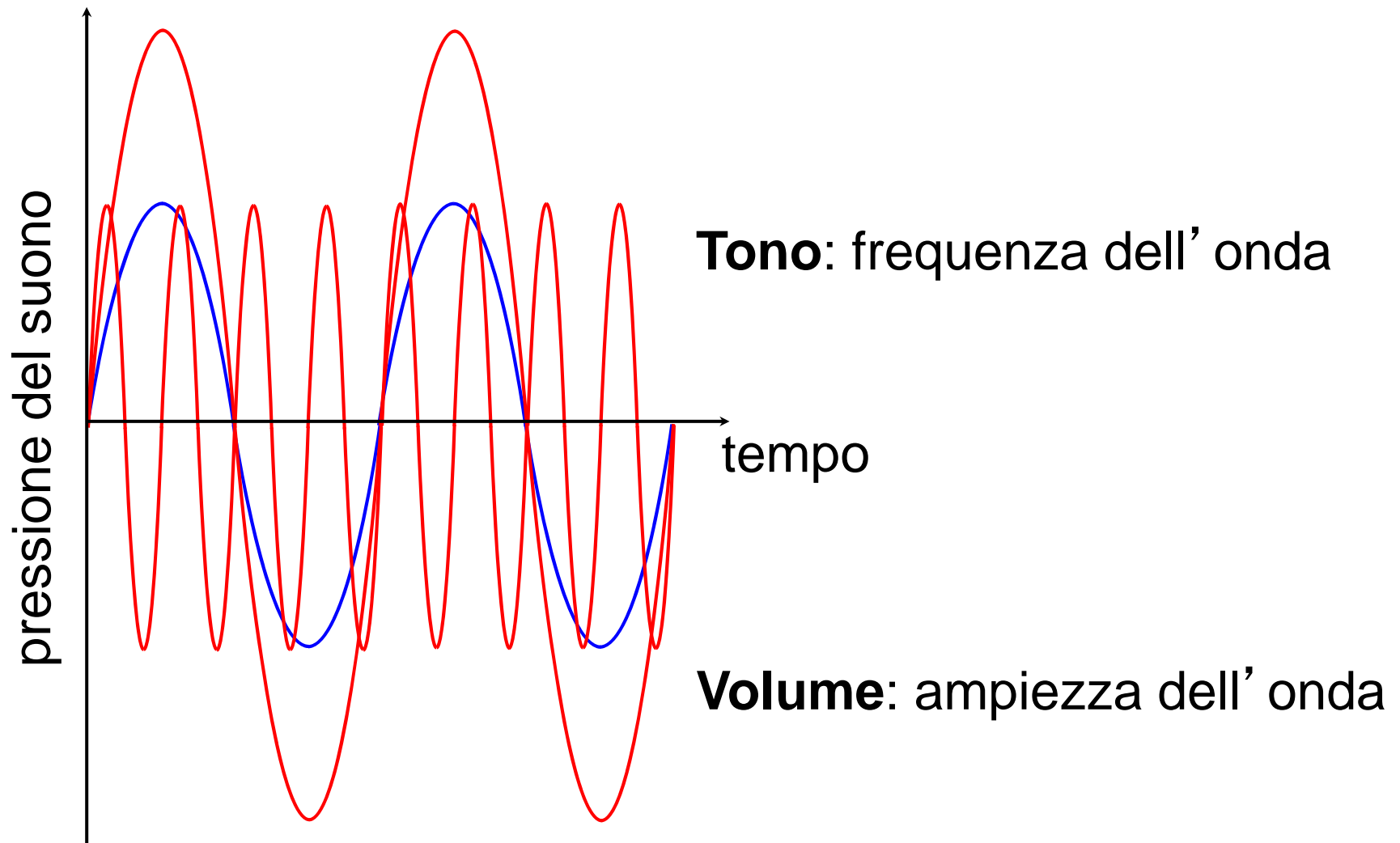
Emettere un suono

- Un oggetto produce suono vibrando all'interno di un mezzo come l'aria
 - onde di pressione sono emanate dall'oggetto
- Le vibrazioni si **trasmettono nell'aria**

Ascoltare un suono

- Le onde di pressione sonora fanno vibrare i nostri timpani
- La forza o intensità della pressione determina il **volume**
- La frequenza (numero di onde al secondo) è il **tono**

Onda sonora



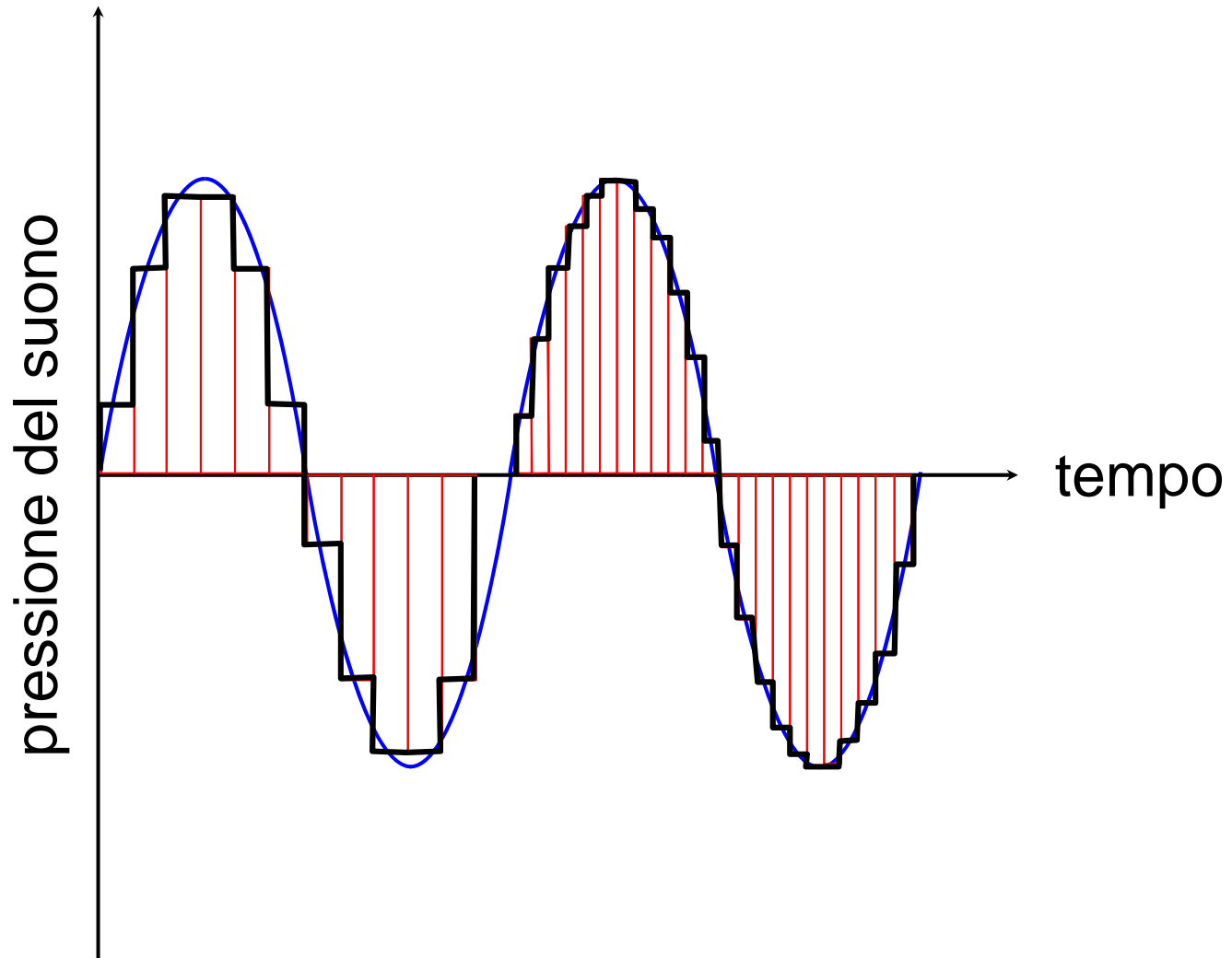
Da analogico a digitale

- Digitalizzare informazioni continue
 - bisogna convertirle in bit
- Esprimere con un numero l'ampiezza
- Quando eseguire le misure?
 - non possiamo registrare **ogni punto** dell'onda

Campionamento

- Si prendono le misure a intervalli regolari
- **Frequenza** di campionamento
 - il numero di misurazioni al secondo
 - maggiore è la frequenza, più accurata sarà la registrazione

Esempio di campionamento



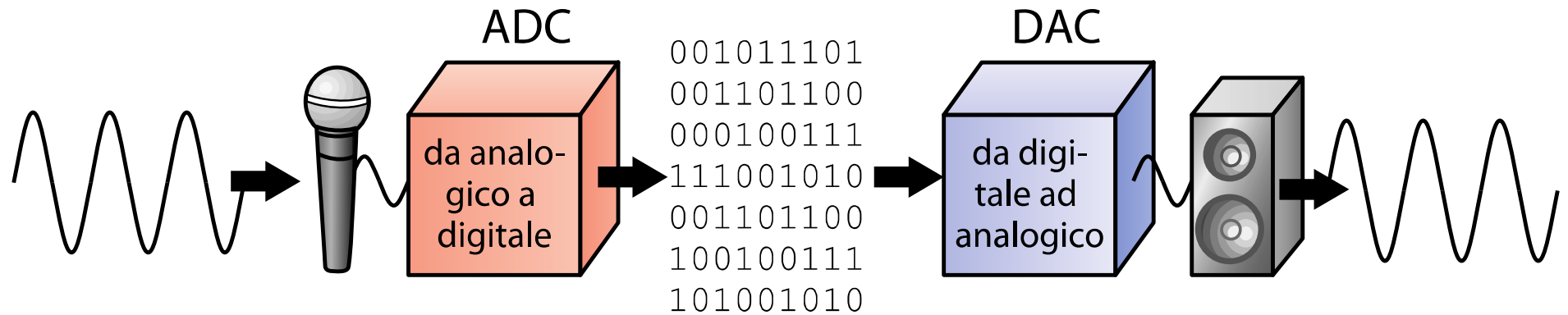
Quale frequenza?

- Frequenza di campionamento **legata a quella dell'onda**
- una frequenza *troppo bassa* potrebbe perdere dettagli che “si infilano” tra un campione e l'altro

Regola di Nyquist

- Frequenza di campionamento
 - **almeno il doppio** di quella dell'onda da registrare
 - l'uomo può percepire suoni fino a 20.000Hz, un campionamento di 40.000Hz è sufficiente
 - la frequenza CD standard è 44.100Hz

Conversioni



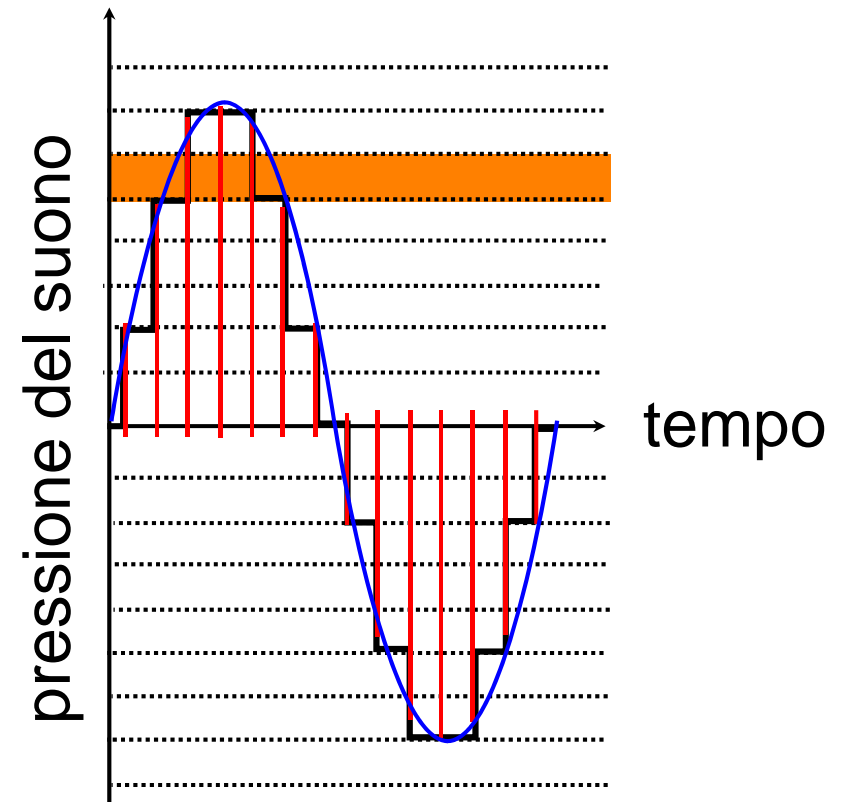
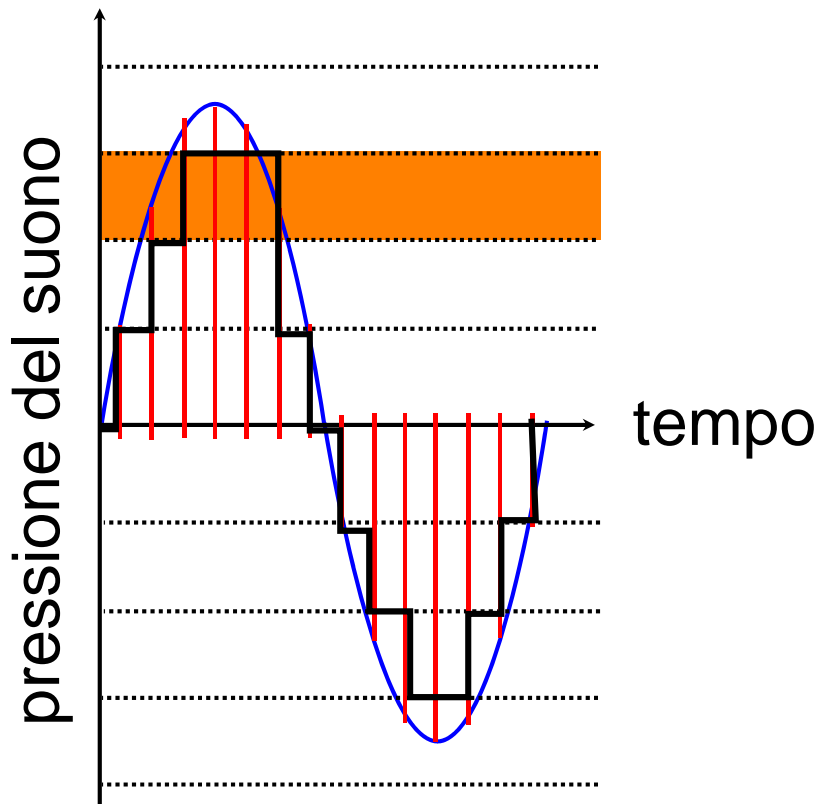
Quanti bit per campione?

- Quanto dev'essere **accurato** un campione?
 - i bit devono rappresentare i valori sia positivi che negativi
 - più bit ci sono, più è accurato il campione

Campione CD

- La rappresentazione digitale dei CD audio utilizza 16 bit
 - registra 65.536 livelli,
 - la metà per i valori positivi e altrettanti per quelli negativi

Accuratezza del campione



Compressione MP3

- Riduzione del **numero di bit** usati per la rappresentazione digitale
- Rimozione delle frequenze non udibili
- Un file MP3 in generale arriva a un fattore di compressione di 10:1
 - larghezza di banda
 - trasmissione via Internet

Duplicazione suono

- I bit possono essere copiati senza perdere informazioni
- L'originale e la copia sono esattamente uguali

Immagini e video digitali

- Immagine a colori di 20 x 25 cm a 300 (dpi)
 - circa **20 secondi** per scaricarla con una ADSL a 7Mb/sec
- Com'è possibile vedere immagini a tutto schermo, in pochi secondi, navigando sul Web?

Immagini e video digitali

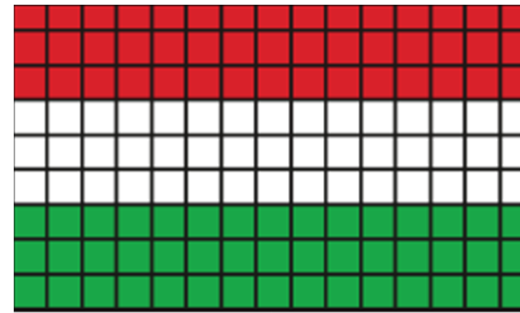
- Un tipico schermo di computer ha meno di 100 pixel per pollice
 - un'immagine digitalizzata a 100 dpi è poco più di un decimo di una a 300dpi
 - richiederebbe un paio di scondi
- Soluzione: la *compressione JPEG*

Compressione

- Cambiare la rappresentazione
 - usare un numero inferiore di bit
 - p.e.: i **fax** sono sequenze di bit che codificano il contenuto di una pagina in bianco e nero.

Compressione

- La codifica **run length** dei fax
 - la lunghezza della prima sequenza di 0,
 - seguita dalla sequenza di 1 ecc
- La compressione è **lossless**, cioè senza perdita
 - l'originale può essere ricostruito perfettamente



[15 × 9] 45:1, 45:2, 45:3

colore	RGB	cod
	255,0,0	1
	255,255,255	2
	0,255,0	3



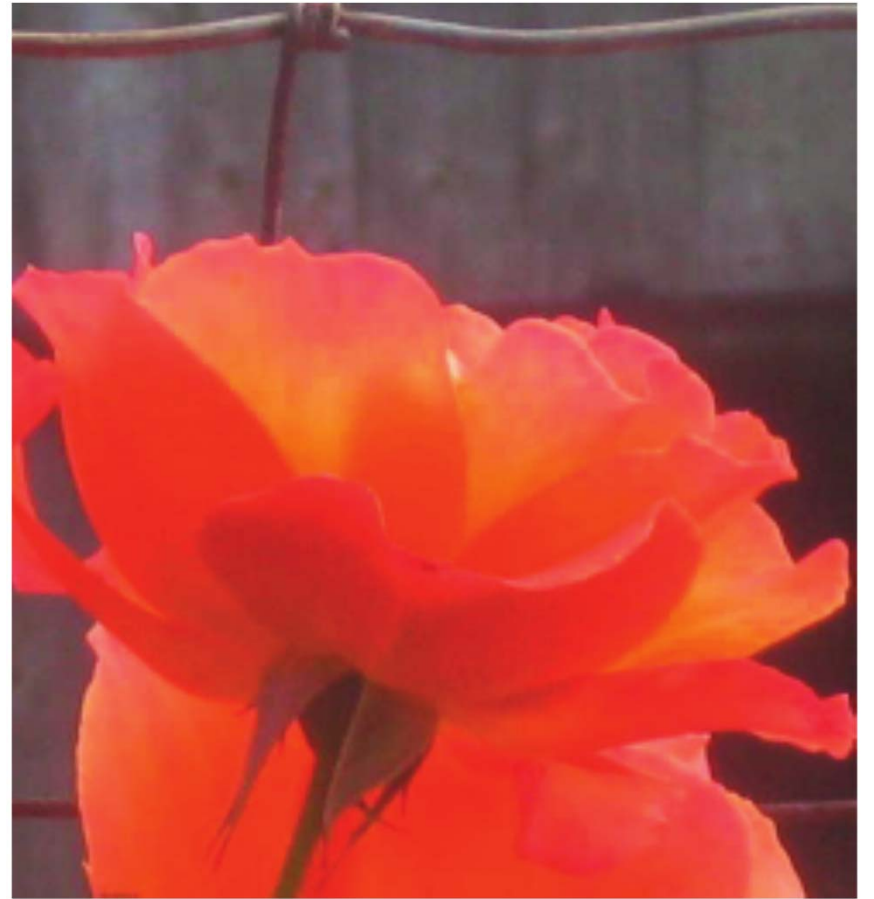
[15x9]5:3,5:2,5:1,5:3,5:2,5:1,5:3,5:2
 ,5:1,
 5:3,5:2,5:1,5:3,5:2,5:1,5:3,5:2,5:1,
 5:3,5:2,5:1,5:3,5:2,5:1,5:3,5:2,5:1

JPEG

- Utilizzata per le immagini fisse
- Fisiologia degli occhi
 - non sono sensibili alle piccole variazioni di *colore*,
 - sono sensibili alle piccole variazioni di *luminosità*
- Descrizione meno accurata del colore
- Compressione 20:1 senza variazioni percepibili



originale



compressione 45:1



originale



compressione 125:1

Compressione MPEG

- La stessa idea di JPEG, ma **applicata ai filmati**
 - la compressione JPEG, è applicata a “ogni frame”

Compressione MPEG

- In genere due immagini consecutive sono molto simili
 - memorizza solo “la differenza” tra due fotogrammi consecutivi
 - un fotogramma completo (JPEG) ogni 5

Mezzo universale

- I bit **non hanno** un significato intrinseco: sono dati
- I bit possono **rappresentare tutto**
 - ogni sorta di **informazione discreta**
- Una volta rappresentato può essere **manipolato**

Esempio

0000 0000 1111 0001

0000 1000 0010 0000 =

