



**Università degli Studi di Palermo**  
*Dipartimento di Ingegneria Informatica*



# **Elaborazione di Immagini e Suoni / Riconoscimento e Visioni Artificiali**

## **12 c.f.u.**

Anno Accademico 2008/2009

Docente: ing. Salvatore Sorce

## **Rappresentazione delle informazioni**

Testo, suoni, immagini

Facoltà di Lettere e Filosofia

# Rappresentazione di caratteri alfanumerici

➤ American Standard Code for Information Interchange (ASCII)

- Codice a 8 bit (usati solo i primi 7):
  - $B_8 B_7 B_6 B_5 B_4 B_3 B_2 B_1$
  - $B_8 = 0$

ASCII	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	
	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
	0	1	0	0	0	1	0	1	0	1	0	1	0	1	0	1
0000	N <sub>U</sub>	S <sub>H</sub>	S <sub>X</sub>	E <sub>X</sub>	E <sub>T</sub>	E <sub>O</sub>	A <sub>K</sub>	B <sub>L</sub>	B <sub>S</sub>	H <sub>T</sub>	L <sub>F</sub>	Y <sub>T</sub>	F <sub>F</sub>	C <sub>R</sub>	S <sub>O</sub>	S <sub>I</sub>
0001	D <sub>L</sub>	D <sub>1</sub>	D <sub>2</sub>	D <sub>3</sub>	D <sub>4</sub>	N <sub>K</sub>	S <sub>Y</sub>	E <sub>Σ</sub>	C <sub>N</sub>	E <sub>M</sub>	S <sub>B</sub>	E <sub>C</sub>	F <sub>S</sub>	G <sub>S</sub>	R <sub>S</sub>	U <sub>S</sub>
0010		!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
0011	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
0100	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
0101	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
0110	~	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
0111	p	q	r	s	t	u	v	w	x	y	z	{		}	~	P <sub>T</sub>

➤ Quanti bit per memorizzare la parola "Ciao"?



## Rappresentazione di caratteri alfanumerici

- American Standard Code for Information Interchange (ASCII)

- Codice a 8 bit (usati solo i primi 7):

- $B_8 B_7 B_6 B_5 B_4 B_3 B_2 B_1$
- $B_8 = 0$

ASCII	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	
	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
	0	1	0	0	0	1	0	1	0	1	0	1	0	1	0	1
0000	N <sub>U</sub>	S <sub>H</sub>	S <sub>X</sub>	E <sub>X</sub>	E <sub>T</sub>	E <sub>O</sub>	A <sub>K</sub>	B <sub>L</sub>	B <sub>S</sub>	H <sub>T</sub>	L <sub>F</sub>	Y <sub>T</sub>	F <sub>F</sub>	C <sub>R</sub>	S <sub>O</sub>	S <sub>I</sub>
0001	D <sub>L</sub>	D <sub>1</sub>	D <sub>2</sub>	D <sub>3</sub>	D <sub>4</sub>	N <sub>K</sub>	S <sub>Y</sub>	E <sub>Σ</sub>	C <sub>N</sub>	E <sub>M</sub>	S <sub>B</sub>	E <sub>C</sub>	F <sub>S</sub>	G <sub>S</sub>	R <sub>S</sub>	U <sub>S</sub>
0010		!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
0011	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
0100	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
0101	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
0110	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
0111	p	q	r	s	t	u	v	w	x	y	z	{		}	~	ᵀ

- Quanti bit per memorizzare la parola "Ciao"?

C -> 01000010

i -> 01101001

a -> 01100001

o -> 01101111

# Rappresentazione di caratteri alfanumerici

- Codificare la stringa "Ciao a tutti." in ASCII:
- 01000010 01101001  
01100001 01101111 ("Ciao")
- 00100000 (spazio/blank)
- 01100001 ("a")
- 00100000 (spazio/blank)
- 01110100 01110101  
01110100 01110100  
01101001 ("tutti")
- 00101110 (".")

ASCII	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0000	N <sub>U</sub>	S <sub>H</sub>	S <sub>X</sub>	E <sub>X</sub>	E <sub>T</sub>	E <sub>O</sub>	A <sub>K</sub>	B <sub>L</sub>	B <sub>S</sub>	H <sub>T</sub>	L <sub>F</sub>	Y <sub>T</sub>	F <sub>F</sub>	C <sub>R</sub>	S <sub>O</sub>	S <sub>I</sub>
0001	D <sub>L</sub>	D <sub>1</sub>	D <sub>2</sub>	D <sub>3</sub>	D <sub>4</sub>	N <sub>K</sub>	S <sub>Y</sub>	E <sub>Σ</sub>	C <sub>N</sub>	E <sub>M</sub>	S <sub>B</sub>	E <sub>C</sub>	F <sub>S</sub>	G <sub>S</sub>	R <sub>S</sub>	U <sub>S</sub>
0010		!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
0011	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
0100	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
0101	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
0110	~	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
0111	p	q	r	s	t	u	v	w	x	y	z	{		}	~	P <sub>T</sub>

# Rappresentazione di caratteri alfanumerici

Codici ASCII estesi per alfabeti nazionali a 8 bit (256 caratteri)

- Definiti dall'ISO (**I**nternational **S**tandard **O**rganization)
- Tanti codici per i diversi alfabeti
- Il nostro è il codice ISO Latin 1

Codice ISO UNICODE per la codifica di tutti i caratteri in una sola tabella

- Codice a 16 bit (65536 caratteri)
- Ancora poco usato
- E' il formato unico dei caratteri in Java

ASCII	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
	0	1	0	0	0	1	0	1	0	1	0	1	0	1	0	1
0000	N <sub>U</sub>	S <sub>H</sub>	S <sub>X</sub>	E <sub>X</sub>	E <sub>T</sub>	E <sub>O</sub>	A <sub>K</sub>	B <sub>L</sub>	B <sub>S</sub>	H <sub>T</sub>	L <sub>F</sub>	Y <sub>T</sub>	F <sub>F</sub>	C <sub>R</sub>	S <sub>O</sub>	S <sub>I</sub>
0001	D <sub>L</sub>	D <sub>1</sub>	D <sub>2</sub>	D <sub>3</sub>	D <sub>4</sub>	N <sub>K</sub>	S <sub>Y</sub>	E <sub>2</sub>	C <sub>N</sub>	E <sub>M</sub>	S <sub>B</sub>	E <sub>C</sub>	F <sub>S</sub>	G <sub>S</sub>	R <sub>S</sub>	U <sub>S</sub>
0010		!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
0011	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
0100	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
0101	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
0110	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
0111	p	q	r	s	t	u	v	w	x	y	z	{		}	~	D <sub>T</sub>
1000	S <sub>0</sub>	S <sub>1</sub>	S <sub>2</sub>	S <sub>3</sub>	I <sub>N</sub>	N <sub>L</sub>	S <sub>S</sub>	E <sub>S</sub>	H <sub>S</sub>	H <sub>J</sub>	Y <sub>S</sub>	P <sub>D</sub>	P <sub>V</sub>	R <sub>I</sub>	S <sub>2</sub>	S <sub>3</sub>
1001	D <sub>C</sub>	P <sub>1</sub>	P <sub>2</sub>	S <sub>E</sub>	C <sub>C</sub>	M <sub>M</sub>	S <sub>P</sub>	E <sub>P</sub>	O <sub>S</sub>	O <sub>O</sub>	O <sub>A</sub>	C <sub>S</sub>	S <sub>T</sub>	O <sub>S</sub>	P <sub>M</sub>	A <sub>P</sub>
1010	A <sub>O</sub>	i	ç	£		¥		§	¨	©	♀	«	¬	-	®	¯
1011	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
1100	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
1101	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
1110	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
1111	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ



## Rappresentazione di dati multimediali

- I dati multimediali (suoni, immagini, video) sono grandezze **continue**
  - Variano nel tempo e nello spazio senza soluzione di continuità
- Un calcolatore può rappresentare solo informazioni **discrete**
  - Non possono assumere valori infinitamente grandi o infinitamente piccoli
  - Non possono variare in maniera arbitraria tra valori vicini
- Esempio reale: 12647321455.572278990951886933278

Troppo grande

Troppe cifre dopo la virgola

**Non c'è spazio**



## Rappresentazione di dati multimediali

- I dati multimediali vengono acquisiti mediante i due processi di **campionamento** e **quantizzazione**
- Nel campionamento vengono prelevati dei campioni rappresentativi del dato multimediale **ad intervalli regolari di tempo (o spazio)**
- Nella quantizzazione si approssima ogni campione **con il valore digitale più vicino rappresentabile dal calcolatore**



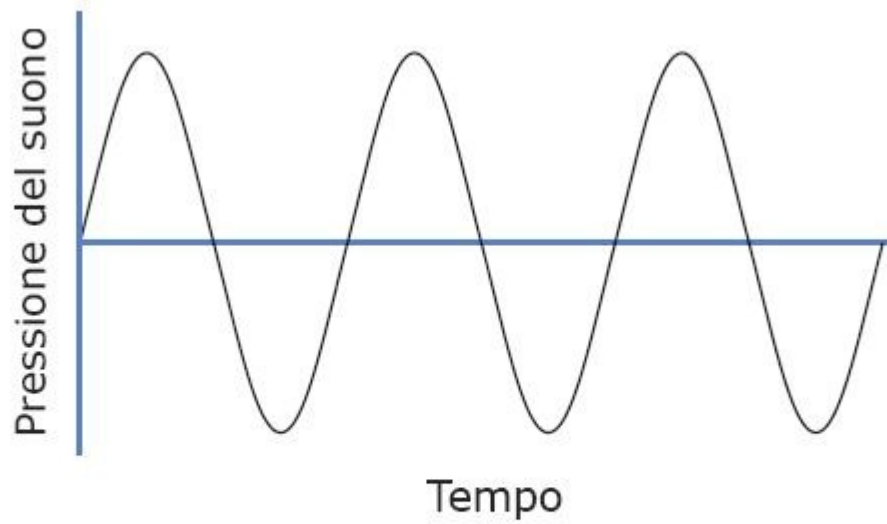
## Digitalizzare il suono

- Un oggetto produce suono vibrando all'interno di un mezzo come l'aria
  - Le vibrazioni si trasmettono nell'aria
  - Le onde di pressione sono emanate dall'oggetto e fanno vibrare i nostri timpani
  - La *forza* o intensità della pressione determina il volume
  - La *frequenza* (numero di oscillazioni al secondo) è l'altezza (tonalità)





# Digitalizzare il suono





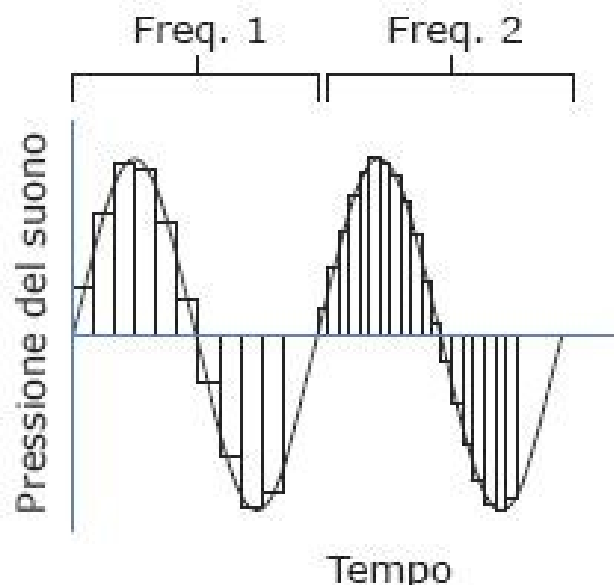
## Digitalizzare il suono

- Per digitalizzare informazioni continue bisogna convertirle in bit
- È possibile esprimere con un numero binario la distanza dell'onda dall'asse (la quantità di pressione positiva o negativa)
- Quando dobbiamo eseguire le misure? Non possiamo registrare ogni punto dell'onda



## Digitalizzare il suono: Campionamento

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



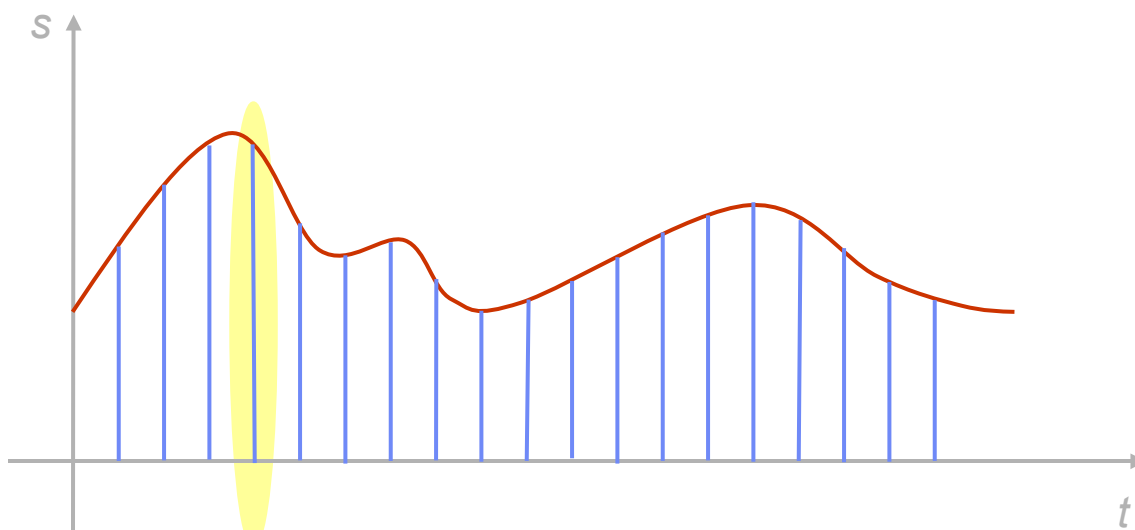


## Digitalizzare il suono: Campionamento

- La frequenza di campionamento dovrebbe essere legata alla frequenza dell'onda
  - una frequenza troppo bassa potrebbe perdere dettagli che "si infilano" tra un campione e l'altro
  - ***regola di Nyquist: la frequenza di campionamento deve essere almeno il doppio di quella massima contenuta nel segnale audio da registrare***
    - ◆ dato che l'uomo può percepire suoni fino a 20.000 Hz, un campionamento di 40.000 Hz è sufficiente
    - ◆ la frequenza standard è 44.100 Hz (44,1 KHz)



# Digitalizzare il suono: Quantizzazione



$$s(t_i) = 64.7478132412561726$$

$$S_i = 64.75 = 0100000011000011$$

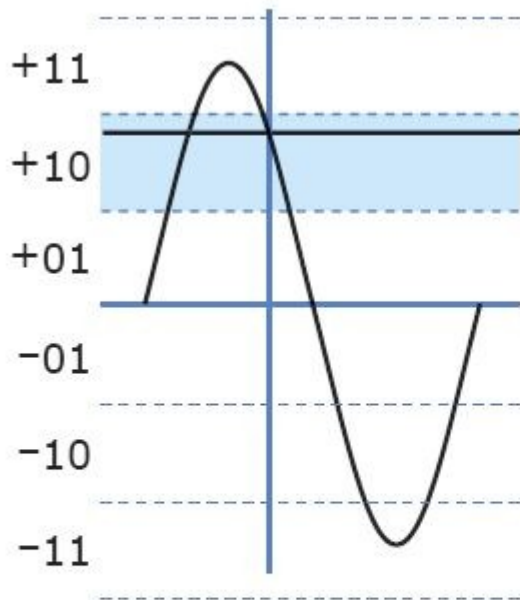


## Digitalizzare il suono: Quantizzazione

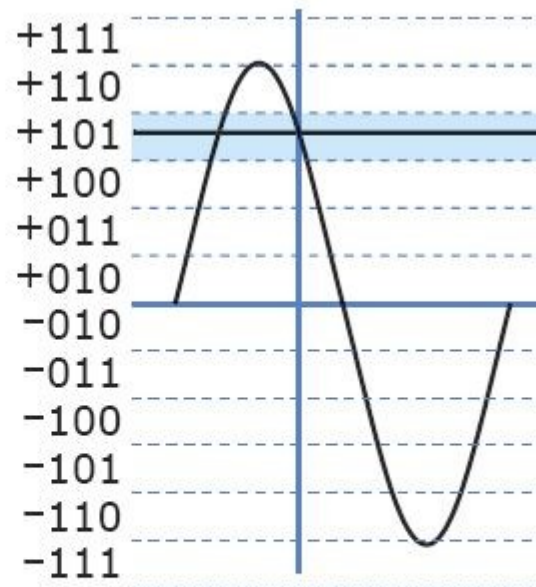
- Quanto deve essere accurato un campione?
  - i bit devono rappresentare i valori sia positivi che negativi
  - più bit ci sono, più è accurato il campione
  - la rappresentazione digitale dei CD audio utilizza 16 bit (registra 65.536 livelli, la metà per i valori positivi e altrettanti per quelli negativi)



## Digitalizzare il suono: Quantizzazione



(a)



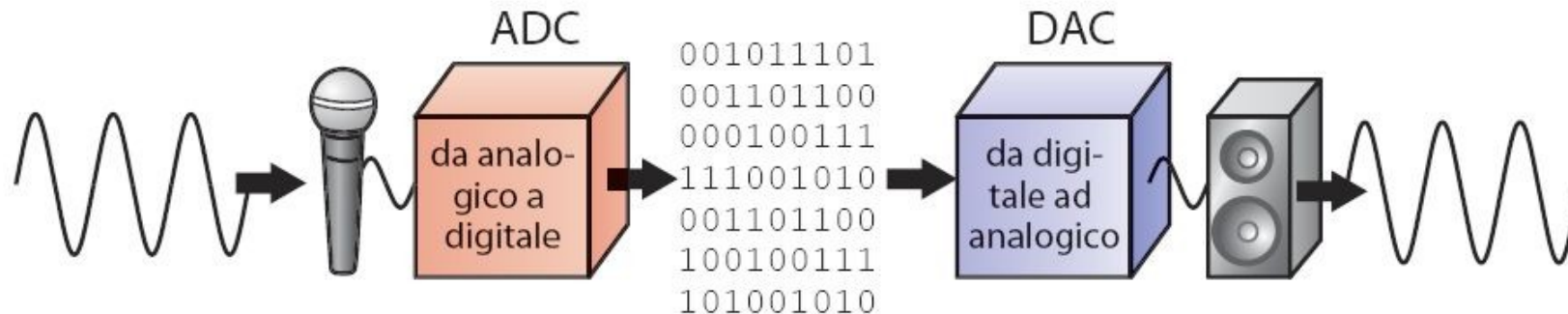
(b)

- (a) Usando campioni a tre bit il valore letto sarà approssimato come +10.  
(b) Aggiungere un altro bit raddoppia l'accuratezza del campione.



## Processo analogico-digitale-analogico

- Processo di digitalizzazione:
  - il suono è convertito dal microfono (*trasduttore*)
  - Il segnale entra in un *convertitore analogico-digitale* (ADC), che campiona l'onda a intervalli regolari e la passa alla memoria sotto forma di numeri binari
- Riproduzione del suono:
  - I numeri passano dalla memoria a un *convertitore digitale-analogico* (DAC), che ricrea l'onda elettrica più semplice che "passa" per tutti i punti rappresentati dal valore dei campioni
  - Il segnale entra poi in un altoparlante che lo converte in onda sonora





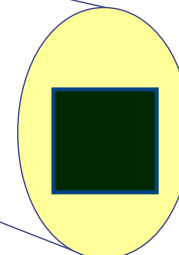
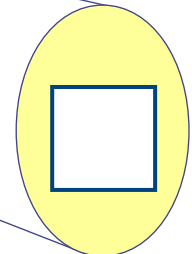
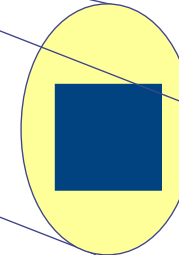


## Vantaggi del suono digitale

- Possiamo eseguire delle elaborazioni
- Compressione MP3
  - un'applicazione è la compressione dell'audio digitale (riduzione del numero di bit necessari alla rappresentazione)
  - le frequenze che l'orecchio umano non può udire sono rimosse
  - un file MP3 in generale arriva a un fattore di compressione di 10:1
- Riprodurre una registrazione
  - i bit possono essere copiati senza perdere informazioni
  - l'originale e la copia sono esattamente uguali



# Digitalizzazione delle immagini





## Digitalizzazione delle immagini

- Rappresentazione binaria dei colori RGB
  - un colore sul monitor è specificato da tre componenti: rosso, verde e blu
  - l'intensità di ogni colore è rappresentata da una quantità (da 0 a 255)
  - Ogni intensità RGB è rappresentata da un byte (8 bit)
  - l'intensità più bassa è 0000 0000, la più alta è 1111 1111
- Alcuni esempi:
  - **Rosso** -> R 1111 1111 G 0000 0000 B 0000 0000 (0xFF0000)
  - **Verde** -> R 0000 0000 G 1111 1111 B 0000 0000 (0x00FF00)
  - **Blu** -> R 0000 0000 G 0000 0000 B 1111 1111 (0x0000FF)
  - **Bianco** -> R 1111 1111 G 1111 1111 B 1111 1111 (0xFFFFFFFF)
  - **Nero** -> R 0000 0000 G 0000 0000 B 0000 0000 (0x000000)
  - **Giallo** -> R 1111 1111 G 1111 1111 B 0000 0000 (0xFFFF00)
  - **Magenta** -> R 1111 1111 G 0000 0000 B 1111 1111 (0xFF00FF)
  - **Ciano** -> R 0000 0000 G 1111 1111 B 1111 1111 (0x00FFFF)
  - **Grigio** -> R 0100 1010 G 0100 1010 B 0100 1010 (0x4A4A4A)



## Digitalizzazione delle immagini

- Quali colori rappresentano questi valori?  
1100 1000 1100 1000 1100 1000
- Ogni byte corrisponde al valore decimale 200. Il colore è rappresentato da RGB (200, 200, 200).
  - In HTML, scritto in esadecimale è #C8C8C8
  - ha una quantità uguale di rosso, verde e blu, è più vicino al bianco che al nero (grigio di media intensità)
  - tutti i colori con componenti RGB uguali sono il nero, il bianco e il grigio
- Aumentare l'intensità con un'addizione binaria
  - per ottenere un grigio più chiaro, occorre modificare i valori in modo che si avvicinino al bianco



## Elaborare una rappresentazione

- Cambiare i colori di una foto della luna
  - immaginiamo di aver digitalizzato una foto in bianco e nero della luna
  - nel computer, i pixel della foto formano una lunga sequenza di terne di byte RGB. Che valori hanno?
    - ◆ ci sono solo i neri, bianchi o grigi
  - e se volessimo fare una versione colorata?



## Colorare la luna di arancione

- Per colorare la luna di arancione dobbiamo cambiare i pixel bianchi:
  - scegliete una sfumatura di arancio, ad esempio (255, 213, 132)
  - cambiate il valore di tutti i pixel bianchi in quelli della sfumatura arancione prescelta
  - questo non modificherà il grigio dei crateri
  
- Da grigio chiaro ad arancione grigiastro:
  - il byte rosso: rimane immutato
  - il byte verde: lo riduciamo di poco (sottraiamo 42)
  - il byte blu: lo riduciamo parecchio (sottraiamo 123)



## Aumentare la luminosità

- Decidiamo di aumentare la luminosità della foto
- Una tecnica è quella di sommare lo stesso valore a tutte le tre componenti dei colori per tutti i pixel
- Per agire solo sui pixel arancioni, basta aggiungere lo stesso valore alle componenti G e B dei pixel che hanno la componente  $R=255$



(a)



(b)



(c)

Tre fotografie della luna: (a) l'originale in bianco e nero; (b) dopo la colorazione; (c) dopo aver messo in risalto le parti più illuminate.





## Immagini e video digitali

- Occorrerebbero 51 minuti per scaricare con un modem un'immagine a colori di 20 x 25 cm digitalizzata a 300 pixel per pollice (dpi) senza alcuna compressione
- Com'è possibile vedere immagini a tutto schermo in pochi secondi navigando sul Web?
- Un tipico schermo di computer ha meno di 100 pixel per pollice
  - un'immagine digitalizzata a 100 dpi richiede poco più di un decimo della memoria
    - ◆ richiede sempre 5 minuti e mezzo per essere spedita
- Soluzione: la compressione



## Compressione

- Cambiare la rappresentazione per usare un numero inferiore di bit per memorizzare o trasmettere informazioni
  - Esempio: i fax sono lunghe sequenze di 0 e 1 che codificano il contenuto di una pagina in bianco e nero. La codifica run length è utilizzata per specificare la lunghezza della prima sequenza di 0, seguita dalla sequenza di 1 ecc.
  - la compressione run length è *lossless*, cioè senza perdita – la rappresentazione originale può essere ricostruita perfettamente



# Compressione JPEG

- Utilizzata per le immagini fisse
- I nostri occhi non sono sensibili alle piccole variazioni di tonalità, ma sono sensibili alle piccole variazioni di luminosità
  - fornisce una descrizione meno accurata del colore
  - la compressione ottiene un rapporto 20:1 senza variazioni percepibili ai nostri occhi



Un dettaglio da un'immagine compressa mediante JPEG. (a) compressione 14:1; (b) compressione 140:1.



# Compressione MPEG

- La stessa idea alla base del JPEG, ma applicata ai filmati
- Come la compressione JPEG, è applicata a ogni frame
- Poiché in genere due immagini consecutive sono molto simili
  - la compressione MPEG memorizza solo “la differenza” tra fotogrammi consecutivi
  - ciò permette livelli di compressione elevatissimi



## Il riconoscimento ottico dei caratteri

- Leggere le targhe per applicare il giusto pedaggio
- Quali sono le difficoltà?
  - i computer devono catturare l'immagine della targa ma la telecamera potrebbe vedere altre immagini della strada
  - Il *frame grabber* riconosce il momento ideale per scattare l'immagine che poi viene spedita al computer
  - il computer deve capire dove si trova la targa nell'immagine
    - ◆ esamina gruppi di pixel cercando i bordi dove il colore cambia
    - ◆ cerca di identificare le *caratteristiche* peculiari
    - ◆ un *classificatore* le confronta con quelle delle lettere dell'alfabeto



## Tecnologia OCR

- Permette al computer di "leggere" caratteri stampati
  - applicazioni commerciali: le poste e le banche



## Ingannare i sensi: la realtà virtuale

- Creazione di un intero mondo alternativo
- Applica quest'idea a tutti i sensi e cerca di eliminare gli indizi che ci tengono legati alla realtà
- *Dispositivi aptici*
  - una tecnologia di input/output per il senso del tatto
  - un guanto aptico permette al computer di determinare la posizione delle nostre dita. Quando avviciniamo abbastanza le dita, il guanto le blocca dandoci la sensazione di aver afferrato un oggetto





## Il problema della latenza

- Uno dei problemi è che il sistema deve operare con velocità e precisione per sembrare naturale
- La *latenza* è il tempo impiegato dall'informazione per essere trasmessa
- Una latenza troppo lunga rovina l'illusione



## Il problema della banda passante

- La quantità di dati che è possibile trasmettere per unità di tempo
- Un aumento della banda passante significa una diminuzione della latenza
- La banda passante dipende:
  - Dal mezzo trasmissivo
  - Dalle tecniche usate per la trasmissione
- Si esprime in bit/sec



# I bit possono rappresentare tutto

- Principio del mezzo universale:
  - per mezzo dei bit si può rappresentare ogni sorta di informazione discreta; i bit non hanno un significato intrinseco.
- I bit sono un mezzo universale
  - tutte le cose che possono essere rappresentate possono anche essere manipolate
- I bit non hanno preferenze
  - il significato dei bit deriva interamente dall'*interpretazione* che ne dà il computer attraverso il programma
- I bit non rappresentano necessariamente numeri
  - i bit possono essere interpretati come numeri, ma anche no



# I bit possono rappresentare tutto

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



# I bit possono rappresentare tutto

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

Un colore RGB:

<b>R = 254</b>	<b>G = 210</b>	<b>B = 13</b>
----------------	----------------	---------------





# I bit possono rappresentare tutto

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

Un numero intero a 24 bit:

**16.699.917**



# I bit possono rappresentare tutto

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

Un numero esadecimale a 6 cifre:

<b>F</b>	<b>E</b>	<b>D</b>	<b>2</b>	<b>0</b>	<b>D</b>
----------	----------	----------	----------	----------	----------



# I bit possono rappresentare tutto

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

Due campioni di un segnale audio (L+R):

L = -19

R = +525





# I bit possono rappresentare tutto

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

Una sequenza di caratteri ASCII:

b	ò	CR (ritorno a capo = INVIO)
---	---	-----------------------------



# I bit possono rappresentare tutto

23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	1	1	1	1	1	0	1	1	0	1	0	0	1	0	0	0	0	0	1	1	0	1
$S_N$	Mantissa															$S_e$	Esponente						

$$-0.111111011010010 \times 2^{13} = -1111110110100.10_2 = -8116.5_{10}$$