



ELEMENTI DI INFORMATICA

DOCENTE: FRANCESCO MARRA

INGEGNERIA CHIMICA

INGEGNERIA ELETTRICA

SCIENZE ED INGEGNERIA DEI MATERIALI

INGEGNERIA GESTIONALE DELLA LOGISTICA E DELLA PRODUZIONE

INGEGNERIA NAVALE


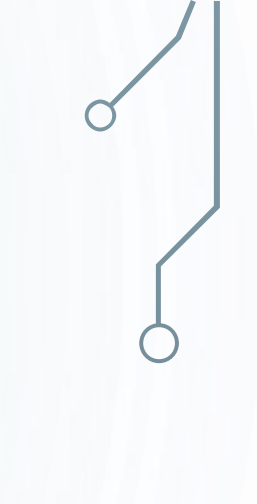
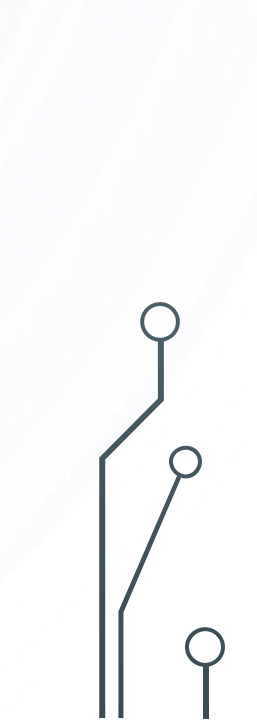
UNIVERSITÀ DEGLI STUDI DI NAPOLI FEDERICO II
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE

RAPPRESENTAZIONE DEL TESTO E DEI DATI MULTIMEDIALI





AGENDA

- La rappresentazione delle informazioni
 - testo
 - suoni
 - immagini
 - video
- 
- 
- 

CODIFICA DEL TESTO

- Cosa occorre per codificare un testo scritto, ad esempio in lingua anglosassone ?
 - un codice in grado di rappresentare l'insieme di simboli comunemente usati nell'alfabeto anglosassone, incluse le cifre numeriche, le lettere maiuscole e minuscole, i simboli di punteggiatura, le parentesi, gli operatori aritmetici, lo spazio bianco, e il simbolo di fine riga
 - un codice in grado di rappresentare almeno 128 simboli
- Codifica ASCII (**A**merican **S**tandard **C**ode for **I**nformation **I**nterchange) è il metodo più diffuso
 - codice a 8 bit (si interpretano solo i primi 7 meno significativi)
 - Il bit più significativo è posto sempre a "0"

CODIFICA DEL TESTO

- Il codice ASCII

- 8 bit
- $b_7 = 0$
- $b_6b_5b_4b_3b_2b_1b_0$
- 2^7 simboli

- Quanti bit servono per codificare la parola "Ciao"?

- "C" → 01000011
- "i" → 01101001
- "a" → 01100001
- "o" → 01101111

→ "Ciao" si codifica in ASCII su 32 bit (4 byte)

	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	1	0	1	0	1	0	1	0	1	0	1	0	1
0000	N _U	S _H	S _X	E _X	E _T	E _Q	A _K	B _L	B _S	H _T	L _F	Y _T	F _F	C _R	S _O	S _I
0001	D _L	D ₁	D ₂	D ₃	D ₄	N _K	S _Y	E _Σ	C _N	E _M	S _B	E _C	F _S	G _S	R _S	U _S
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 _T

CODIFICA DEL TESTO

- ASCII di «Ciao a tutti.»

- “C” → 01000011
- “i” → 01101001
- “a” → 01100001
- “o” → 01101111
- “ ” → 00100000
- “a” → 01100001
- “ ” → 00100000
- “t” → 01110100
- “u” → 01110101
- “t” → 01110100
- “t” → 01110100
- “i” → 01101001
- “.” → 00101110

	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	1	0	1	0	1	0	1	0	1	0	1	0	1
0000	N _U	S _H	S _X	E _X	E _T	E _Q	A _K	B _L	B _S	H _T	L _F	Y _T	F _F	C _R	S ₀	S _I
0001	D _L	D ₁	D ₂	D ₃	D ₄	N _K	S _Y	E _Σ	C _N	E _M	S _B	E _C	F _S	G _S	R _S	U _S
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 _T

“Ciao a tutti.” si codifica in ASCII su 13*8=104 bit (13 byte)

01000011 01101001 01100001 01101111 00100000 01100001 00100000
01110100 01110101 01110100 01110100 01101001 00101110

CODIFICA DEL TESTO

- Consideriamo il problema inverso
- Data una sequenza di bit codificata in ASCII, il testo che essa codifica può essere ottenuto nel modo seguente:
 - si divide la sequenza in gruppi di otto bit (un byte)
 - si determina il carattere corrispondente ad ogni byte

- Esempio

- 01101001 01101100 00100000 01010000 01101111 00101110
i l P O .

CODIFICA DEL TESTO

- Detto n il numero di caratteri del testo da codificare
 - per la codifica in ASCII occorrono $(8 * n)$ bit = n byte
- Esempio
 - «Quanti bit occorrono per codificare questo testo in ASCII?»

CODIFICA DEL TESTO

- Detto n il numero di caratteri del testo da codificare
 - per la codifica in ASCII occorrono $(8 * n)$ bit $= n$ byte
- Esempio
 - «Quanti bit occorrono per codificare questo testo in ASCII?»
 - $n=58 \rightarrow 464$ bit $\rightarrow 58$ byte

CODIFICA DEL TESTO

- Tabella ASCII
 - 8 bit – 128 caratteri
- Tabelle ASCII estese
 - 8 bit – 256 caratteri
 - ogni variante prevede ulteriori 128 caratteri
- Universal Encoding (UNICODE)
 - 16 bit – più di 65000 caratteri
- Unicode Transformation Format (UTF)
 - UTF-8 (8 bit)
 - UTF-16 (16 bit)
 - UTF-32 (32 bit)

CODIFICA DEL TESTO

- HyperText Markup Language (HTML)
 - Internet
 - ` testo rosso `
- DOC
 - Microsoft Word
- Portable Document Format (PDF)
- LaTeX

RAPPRESENTAZIONI DI DATI MULTIMEDIALI

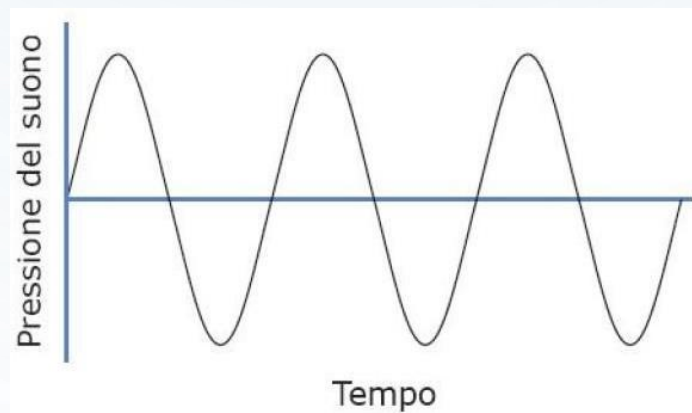
- I dati multimediali (suoni, immagini, video) sono grandezze **continue**
 - variano nel tempo e nello spazio senza soluzione di continuità
- Un calcolatore può rappresentare solo informazioni **discrete**
 - non possono assumere valori infinitamente grandi o infinitamente piccoli
 - non possono variare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

RAPPRESENTAZIONI DI DATI MULTIMEDIALI

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

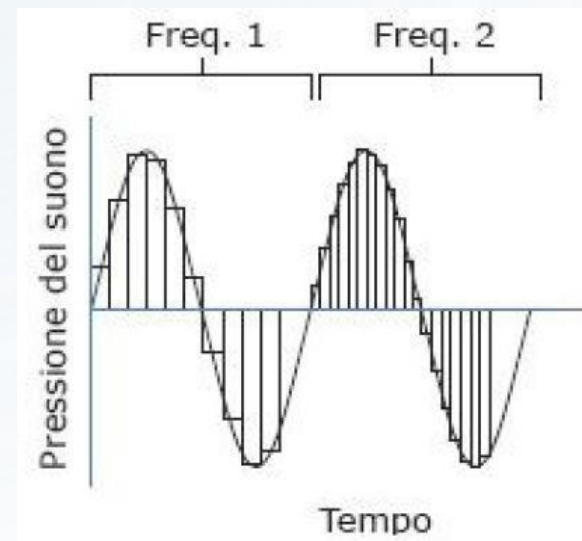
CODIFICA DEL SUONO

- 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) determina la tonalità



CODIFICA DEL SUONO: CAPIONAMENTO

- 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)
- 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

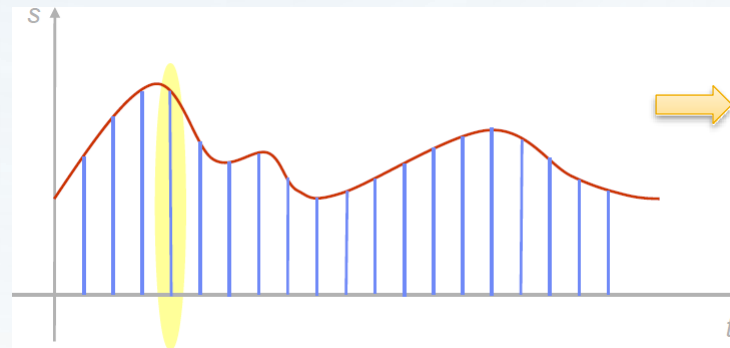


CODIFICA DEL SUONO: CAPIONAMENTO

- La frequenza di campionamento dovrebbe essere legata alla frequenza dell'onda
 - una frequenza troppo bassa potrebbe perdere variazioni significative dell'onda che avvengono 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
- L'uomo può percepire suoni fino a 20.000 Hz
 - un campionamento di 40.000 Hz è spesso sufficiente
 - la frequenza standard è 44.100 Hz (44,1 KHz)

CODIFICA DEL SUONO: QUANTIZZAZIONE

- Quanto deve essere accurato un campione?
 - i bit devono rappresentare sia i valori positivi che negativi del segnale
 - più bit ci sono, più è accurata la rappresentazione digitale del campione raccolto



$$s(t_i) = 64.7478132412561726$$

$$S_i = 64.75 = 0100000011000011$$

- Esempio: CD audio
 - campionamento a 44.1 kHz
 - quantizzazione a 16 bit per ogni campione(registrano 65.536 livelli)

CODIFICA DEL SUONO

- Esempio
 - quanti bit occorrono per memorizzare un brano musicale di 3 minuti,
 - campionato a 30 kHz, con quantizzazione a 16 bit?

CODIFICA DEL SUONO

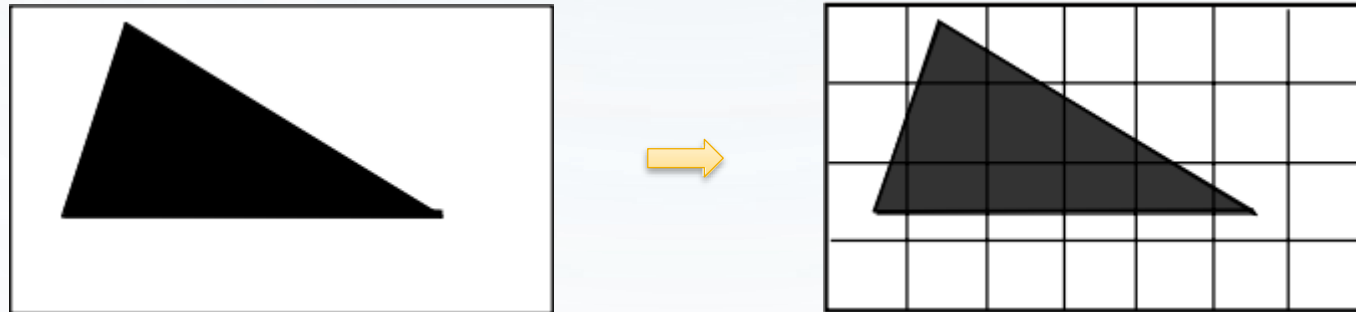
- Esempio
 - quanti bit occorrono per memorizzare un brano musicale di 3 minuti,
 - campionato a 30 kHz, con quantizzazione a 16 bit?
- Risposta
 - 3 minuti \rightarrow 180 secondi
 - 30 kHz \rightarrow 30000 campioni al secondo
 - 16 bit per campione \rightarrow 480000 bit per memorizzare ogni secondo
 - 86400000 bit per memorizzare tutto il brano musicale
 - 10800000 byte per memorizzare tutto il brano musicale
 - \simeq 10546,9 KB per memorizzare tutto il brano musicale
 - \simeq 10,3 MB per memorizzare tutto il brano musicale

I VANTAGGI IL SUONO DIGITALE

- Si possono eseguire delle elaborazioni sul segnale codificato
- Compressione MP3
 - un'elaborazione tipica è la compressione del segnale digitale allo scopo di ridurre il numero di bit necessari alla sua rappresentazione
 - le frequenze che l'orecchio umano non può percepire sono rimosse
 - un file MP3 può raggiungere fattori di compressione di 10:1
- Riproduzione
 - Il segnale digitale (i bit) può essere copiato senza perdere informazioni
 - l'originale e la copia sono esattamente uguali

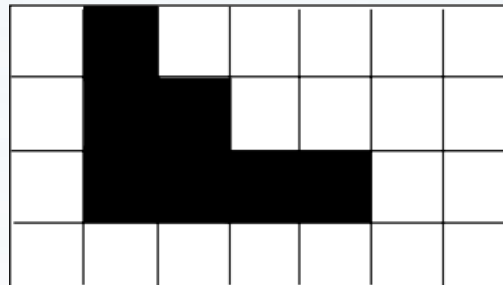
CODIFICA DELLE IMMAGINI

- Ogni quadratino derivante da tale suddivisione prende il nome di pixel (picture element) e può essere codificato in binario secondo la seguente convenzione
 - il simbolo “0” viene utilizzato per la codifica di un pixel corrispondente ad un quadratino dell’immagine in cui il bianco è predominante
 - il simbolo “1” viene utilizzato per la codifica di un pixel corrispondente ad un quadratino dell’immagine in cui il nero è predominante



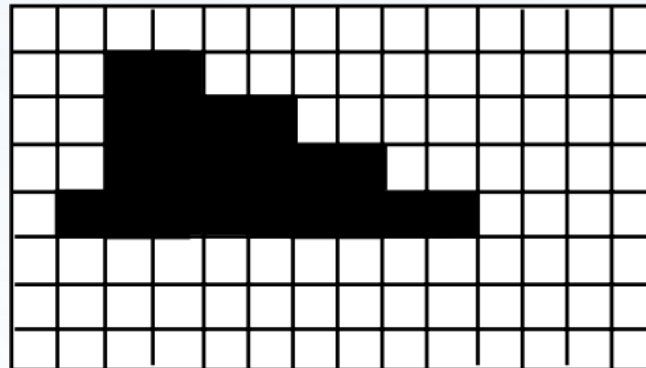
CODIFICA DELLE IMMAGINI

- Avendo digitalizzato un'immagine continua tramite informazioni discrete, avremo un'approssimazione della figura originaria
 - l'approssimazione sarà più fedele all'aumentare del numero di pixel, ossia al diminuire delle dimensioni dei quadratini della griglia in cui viene suddivisa l'immagine originale
- Ad esempio, proviamo a visualizzare l'immagine digitale associata alla stringa di bit 0000000011110001100000100000



CODIFICA DELLE IMMAGINI

- Avendo digitalizzato un'immagine continua tramite informazioni discrete, avremo un'approssimazione della figura originaria
 - l'approssimazione sarà più fedele all'aumentare del numero di pixel, ossia al diminuire delle dimensioni dei quadratini della griglia in cui viene suddivisa l'immagine originale
- Ad esempio, utilizzando 112 pixel invece di 28



CODIFICA DELLE IMMAGINI

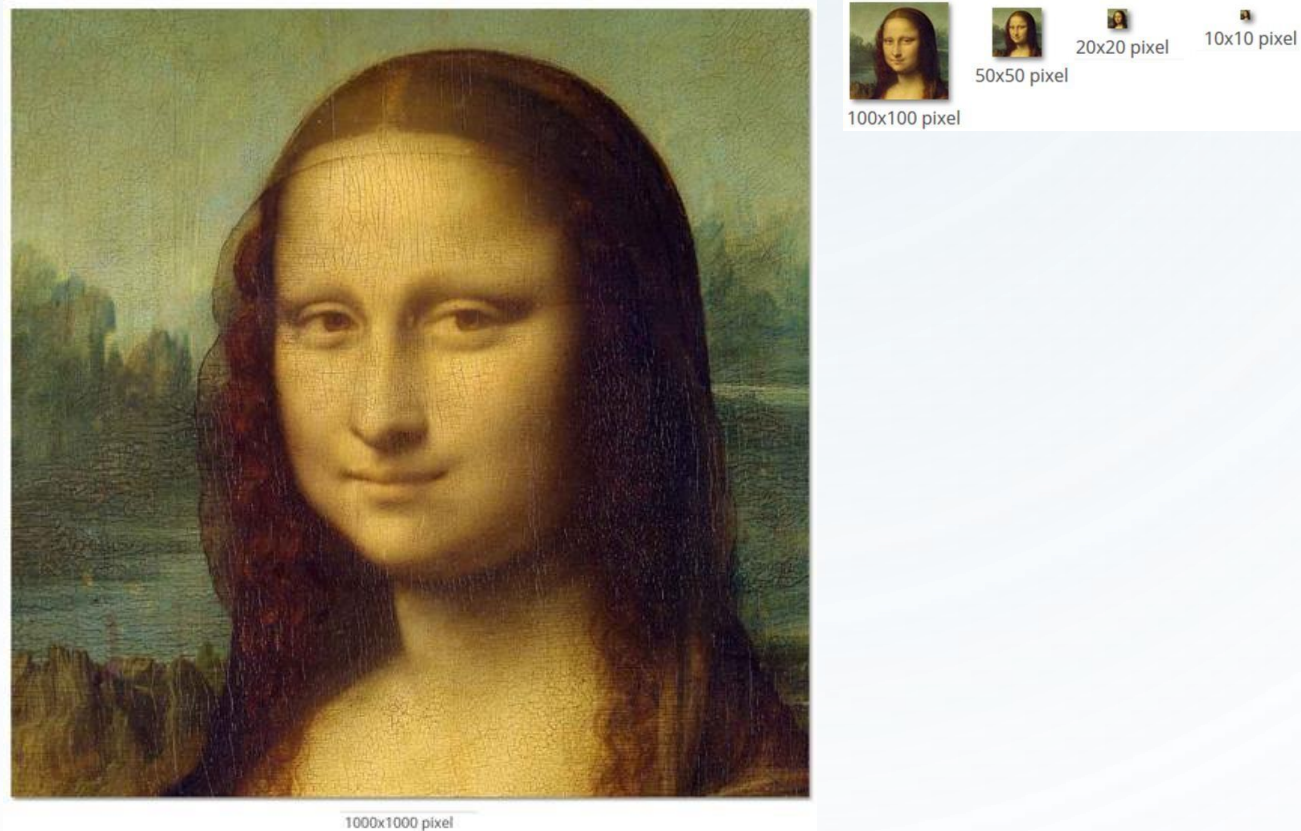
- Assegnando un bit ad ogni pixel è possibile codificare solo immagini in bianco e nero e senza livelli di chiaroscuro
 - sfumature date da diversi livelli di intensità di grigio
- Per codificare le immagini con diversi livelli di grigio si può una tecnica simile
 - Ogni pixel codifica il livello di grigio predominante in un particolare quadratino dell'immagine
 - il livello di grigio di ogni pixel non può essere codificato tramite un solo bit
 - ad esempio, utilizzando 4 bit si possono rappresentare 16 livelli di grigio, mentre 8 bit se ne possono rappresentare 256, ecc.

CODIFICA DELLE IMMAGINI

- Analogamente si possono 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 640X480, 1024X768, oppure 1280X1024 ed un numero di colori per pixel che va da 256 fino a sedici milioni di 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
 - l'approssimazione sarà più fedele all'aumentare del numero di pixel, e del numero di colori con cui è possibile rappresentare ogni pixel

CODIFICA DELLE IMMAGINI

- La stessa immagine codificata con diverso numero di pixel



CODIFICA DELLE IMMAGINI

- Le immagini precedenti, viste alle stesse dimensioni



1000x1000 pixel



100x100 pixel



50x50 pixel



20x20 pixel



10x10 pixel

CODIFICA DELLE IMMAGINI

- La stessa immagine codificata con diverso numero di colori



256 colori



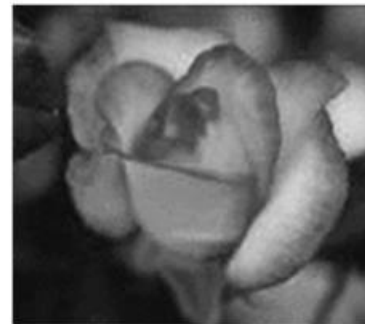
64 colori



16 colori



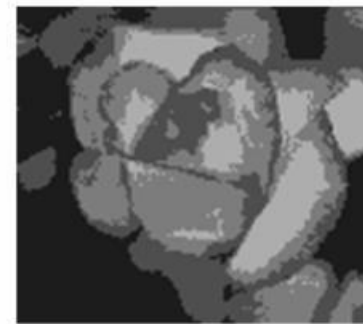
4 colori



256 livelli grigio



16 livelli grigio



4 livelli grigio



bianco/nero

CODIFICA DELLE IMMAGINI

- Codifica colori
 - RGB
 - il colore di ogni pixel visto come somma di tre colori fondamentali Rosso (Red), Verde (Green) e Blu (Blue)
- La profondità di colore indica il numero di bit usati per codificare il colore di ogni singolo pixel
 - 24 bit (8 per rosso, 8 per verde, 8 per blu) cioè 16.7 milioni di colori possibili per ogni pixel (colori veri)
 - 48 bit (alta definizione)

CODIFICA DELLE IMMAGINI

- Esempio
 - quanti bit occorrono per codificare un'immagine in bitmap usando una
 - risoluzione di 640X480 pixel a 256 colori ?

CODIFICA DELLE IMMAGINI

- Esempio
 - quanti bit occorrono per codificare un'immagine in bitmap usando una
 - risoluzione di 640X480 pixel a 256 colori ?
- Risposta
 - 640X480 pixel \rightarrow 307200 pixel
 - 1 pixel \rightarrow 256 possibili colori \rightarrow 8 bit per rappresentare un singolo pixel
 - 2457600 bit per codificare l'intera immagine
 - 307200 byte per codificare l'intera immagine
 - 300 KB per codificare l'intera immagine
 - \simeq 0,29 MB per codificare l'intera immagine

CODIFICA DELLE IMMAGINI

- Compressione
 - TIFF
 - JPEG
 - ...
- I nostri occhi non sono sensibili alle piccole variazioni di tonalità, ma sono sensibili alle piccole variazioni di luminosità
- JPEG
 - fornendo una descrizione meno accurata del colore, si riescono ad raggiungere compressioni fino ad un rapporto 20:1 senza variazioni percepibili ai nostri occhi

CODIFICA DEI VIDEO

- I video possono essere visti come immagini in movimento nel tempo
 - bisogna digitalizzare una sequenza continua nel tempo di immagini
- Campionamento
 - si cattura un'immagine (frame) per secondo (fps)
 - Esempi
 - Cinema: 24 fps
 - Televisione: 25-30 fps

CODIFICA DEI VIDEO

- Esempio
 - quanti bit occorrono per codificare un film in 4k (3840 x 2160), con alta definizione di profondità di colore (48 bit), a 24 fps, che dura 2 ore ?

CODIFICA DEI VIDEO

- Esempio
 - quanti bit occorrono per codificare un film in 4k (3840 x 2160), con alta definizione di profondità di colore (48 bit), a 24 fps, che dura 2 ore ?
- Risposta
 - 4k \rightarrow 8294400 pixel
 - 1 pixel \rightarrow 48 bit per rappresentare ogni singolo pixel
 - 398131200 bit per rappresentare un singolo frame
 - 9555148800 bit per rappresentare ogni secondo del film
 - 2 ore \rightarrow 7200 secondi
 - 68797071360000 bit per codificare il film
 - 8599633920000 byte per codificare il film
 - 8398080000 KB per codificare il film
 - 8201250 MB per codificare il film
 - $\simeq 8009$ GB $\simeq 7,8$ TB per codificare il film

CODIFICA DEI VIDEO

- Compressione
 - MPEG
 - VVC
 - ...
- MPEG
 - la stessa idea alla base del JPEG, ma applicata ai frame dei filmati
 - poiché in genere due immagini consecutive sono molto simili, la compressione MPEG memorizza solo “la differenza” tra fotogrammi consecutivi, ottenendo livelli di compressione molto elevati

DOMANDE, DUBBI, PERPLESSITÀ

