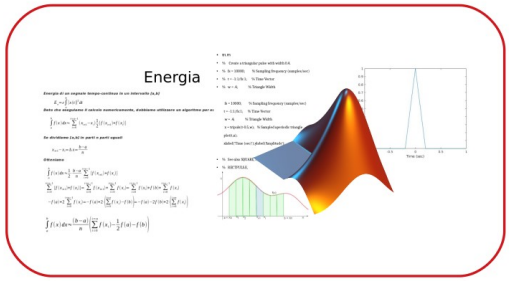
UNIVERSITÀ DEGLI STUDI DI TRIESTE

Programmazione Informatica

Ingegneria Civile e Ambientale
Ingegneria Navale

Lezione Teoria 2. Rappresentazione dell'informazione

Antonio Fiumara
antonio.fiumara@dia.units.it



A. A. 2025-26

UNIVERSITÀ DEGLI STUDI DI TRIESTE

Struttura del Corso

Gli argomenti trattati nel corso saranno:

- Lezione Teoria 1 – Introduzione, presentazione del corso, codifica dell'informazione
- **Lezione Teoria 2 – Rappresentazione dell'informazione in un computer**
- Lezione Teoria 3 – Algebra booleana, introduzione e applicazioni nella programmazione
- Lezione Teoria 4 – Architettura dei calcolatori. HW, SW, Sistemi Operativi
- Lezione Teoria 5 – Algoritmi e strutture dati
- Lezione Teoria 6 – Implementazione di algoritmi notevoli (ordinamento , ricerca, ecc)
- Lezione Programmazione - Matlab 1 – Introduzione, installazione, comandi di base, matrici
- Lezione Programmazione - Matlab 2 – Input/Output (I/O), strutture di controllo e funzioni
- Lezione Programmazione - Matlab 3 – Sistemi Lineari e Algebra lineare
- Lezione Programmazione - Matlab 4 – Analisi matematica
- Lezione Programmazione - Matlab 5 – Funzioni nel dominio del tempo e della frequenza
- Lezione Programmazione - Matlab 6 – Analisi ed elaborazione dei segnali
- Lezione Programmazione - Matlab 7 – Ripasso e approfondimenti
- Esonero esame Programmazione Matlab (3/12/25)

Introduzione

Rappresentazione dei dati

UNIVERSITÀ DEGLI STUDI DI TRIESTE

Conversione da decimale a base diversa da 10

- Conversione da base 10 a base $b \neq 10$: conviene utilizzare lo schema iterativo dei resti delle divisioni intere per b, eseguendole in base 10:

$$(d_k \dots d_0)_{10} = \boxed{q_{10}} \cdot b_{10} + \underline{r_{10}}$$

  - da base 10 a base 2:

$$\begin{aligned}
(13)_{10} &= \boxed{6} \cdot 2 + \underline{1} \quad \Rightarrow \quad a_0 = 1 \\
(6)_{10} &= \boxed{3} \cdot 2 + \underline{0} \quad \Rightarrow \quad a_1 = 0 \\
(3)_{10} &= \boxed{1} \cdot 2 + \underline{1} \quad \Rightarrow \quad a_2 = 1 \\
(1)_{10} &= \boxed{0} \cdot 2 + \underline{1} \quad \Rightarrow \quad a_3 = 1
\end{aligned}$$

$$(13)_{10} = (\underbrace{a_3\ a_2\ a_1\ a_0}_{})_2 = (1101)_2$$

cifre a_i dalla più significativa (a_3) alla meno significativa (a_0)

UNIVERSITÀ DEGLI STUDI DI TRIESTE

Conversione da decimale a base diversa da 10

- Conversione da base 10 a base $b \neq 10$:
  - da base 10 a base 8:

$$(87)_{10} = \boxed{10} \cdot 8 + \underline{7} \quad \Rightarrow \quad a_0 = 7$$
$$(10)_{10} = \boxed{1} \cdot 8 + \underline{2} \quad \Rightarrow \quad a_1 = 2$$
$$(1)_{10} = \boxed{0} \cdot 8 + \underline{1} \quad \Rightarrow \quad a_2 = 1$$

$$(87)_{10} = (a_2\ a_1\ a_0)_8 = (127)_8$$

  - da base 10 a base 16:

$$(2607)_{10} = \boxed{162} \cdot 16 + \underline{15} \quad \Rightarrow \quad a_0 = (15)_{10} = \mathrm{F}_{16}$$
$$(162)_{10} = \boxed{10} \cdot 16 + \underline{2} \quad \Rightarrow \quad a_1 = 2$$
$$(10)_{10} = \boxed{0} \cdot 16 + \underline{10} \quad \Rightarrow \quad a_0 = (10)_{10} = \mathrm{A}_{16}$$

$$(2607)_{10} = (a_2\ a_1\ a_0)_{16} = (\mathrm{A2F})_{16}$$

UNIVERSITÀ DEGLI STUDI DI TRIESTE

Conversione da esadecimale a binario e viceversa

La conversione da binario a esadecimale e viceversa è immediata,

- basta convertire ogni cifra con un gruppo di 4 bit (mediante la tabella accanto)

- $(3af)_{16}$=0011 1010 1111

  3 a f

- 1100 0001 0111 0110 = $(c176)_{16}$

  c 1 7 6

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

UNIVERSITÀ DEGLI STUDI DI TRIESTE

Rappresentazione di numeri interi
Codifica in binario puro

Supponiamo di voler utilizzare un byte, (il nostro sistema numerico ha un «parallelismo» di 8 bit), per rappresentare il nostro numero e di utilizzare una codifica in binario puro. In questo caso, tutto quello che dobbiamo fare è trasformare il nostro numero intero in un numero binario di 8 bit

La prima domanda che dobbiamo porci è quale sia il valore massimo che posso rappresentare. Dato che con 8 bit sono possibili 256 valori, con questa codifica posso rappresentare numeri interi da 0 a 255. Con 16 bit si possono rappresentare numeri interi positivi da 0 a 565’535 In generale, su N bit, con la codifica in binario puro posso rappresentare numeri interi da 0 a 2^N-1.

0 → 0000 0000

1 → 0000 0001

2 → 0000 0010

3 → 0000 0011

…

254 → 1111 1110

255 → 1111 1111

UNIVERSITÀ DEGLI STUDI DI TRIESTE

Operazioni su numeri interi
Codifica in binario puro

Consideriamo le 4 operazioni algebriche elementari, $A, B \in \mathbb{N} = \{0, 1, 2, 3, 4, 5, 6, ...\}$

Somma	A + B
Sottrazione	A - B
Moltiplicazione	A * B
Divisione	A / B

Spostamento a destra

Spostamento a sinistra

Vediamo come risolverle e quali insidie possono nascondersi dietro ad un semplice calcolo

UNIVERSITÀ DEGLI STUDI DI TRIESTE

Operazioni su numeri interi
Codifica in binario puro

$A, B \in \mathbb{N} = \{0, 1, 2, 3, 4, 5, 6, \ldots\ldots\}$, parallelismo 8 bit

Somma: S= A + B, si procede esattamente come con le somme tra numeri in base 10,

A= 1111 0110 + $(246)_{10}$

B= 0000 0101 = (5)_10

S= 1111 1011 (251)_10

N.B. Il valore massimo che può assumere A+B è 255. Cosa accade se A + B supera il valore massimo possibile? Il risultato sarà errato (**overflow**). Un sistema ben progettato deve rilevare l'overflow e comunicare all'utente il corrispondente codice di errore in modo che l'errore possa essere gestito

A= 1111 0110 + (246)_10

B= 0000 1101 = (13)_10

S= 0000 0011 (3)_10 **!!**

Operazioni su numeri interi
Codifica in binario puro

$A, B \in \mathbb{N} = \{0, 1, 2, 3, 4, 5, 6, \ldots\ldots\}$, parallelismo 8 bit

Differenza: D= A - B, si procede esattamente come con le somme tra numeri in base 10,

A= 1111 0110 - (246)_10

B= 0000 0101 = (5)_10

D= 1111 0001 (241)_10

N.B. Se A<B l'operazione non è possibile. Un sistema ben progettato deve rilevare l'errore e comunicare all'utente il corrispondente codice di errore in modo che l'errore possa essere gestito.

UNIVERSITÀ DEGLI STUDI DI TRIESTE

Operazioni su numeri interi
Codifica in binario puro

Prodotto: P= A * B $A, B \in \mathbb{N} = \{0, 1, 2, 3, 4, 5, 6, ...\}$

Un algoritmo per il calcolo del prodotto potrebbe essere il seguente:

Si esegue la somma di A con se stesso per un numero di volte pari a B

Esempio: 6*3 = 6+6+6=18

0000 0110+ $(6)_{10}$

0000 0110 + $(6)_{10}$

0000 0110 = $(6)_{10}$

0001 0010 $(18)_{10}$

Un altro metodo per eseguire il prodotto è la cosiddetta “operazione in colonna”, analogamente al caso della moltiplicazione in base 10

0000 0110 * $(6)_{10}$

0000 0011 = $(3)_{10}$

0000 0110 + $(6)_{10}$

0000 1100 = $(12)_{10}$

0001 0010 $(18)_{10}$

N.B. Anche nel caso della moltiplicazione può verificarsi l’errore di **overflow**.

UNIVERSITÀ DEGLI STUDI DI TRIESTE

Operazioni su numeri interi
Codifica in binario puro

$B \neq 0 \; A, B \in \mathbb{N} = \{0, 1, 2, 3, 4, 5, 6, ...\}$

$A / B = \boldsymbol{Q} * B + \boldsymbol{R}$

Q è il quoziente

R è il resto

Possiamo procedere nel seguente modo

1) Pongo Q = 0 e R = 0

2) se A < B, pongo R=A - B e termino il calcolo

3) altrimenti {incremento(Q), A=A – B, vado al punto 2)}

N.B. Se B=0 l'operazione non è possibile. Un sistema ben progettato deve rilevare l'errore e comunicare all'utente il corrispondente codice di errore in modo che l'errore possa essere gestito.

UNIVERSITÀ DEGLI STUDI DI TRIESTE

Operazioni su numeri interi
Codifica in binario puro

Possiamo continuare ad utilizzare la codifica binaria pura riservando un bit al segno.

Ad esempio, possiamo utilizzare il bit in posizione 7, quello più a sinistra, per rappresentare il segno e i rimanenti 7 bit (posizione da 6 a 0) per codificare il valore assoluto del numero.

Codifichiamo il segno in questo modo:

Se il numero è positivo, il bit 7 vale 0

Se il numero è negativo, il bit 7 vale 1

Ad esempio il numero su 8 bit 0000 0011 è uguale a +3

Il numero 1000 0011 è uguale a -3.

Semplice ma c'è un problema. Il numero 0000 0000 rappresenta 0, e il numero 1000 0000 cosa rappresenta? Anch'esso rappresenta lo zero? Non abbiamo più una codifica univoca.

Possiamo escludere 1000 0000 dalla codifica. Può funzionare ma non è la soluzione più efficace.

Soluzione: codifica in "complemento a 2"

UNIVERSITÀ DEGLI STUDI DI TRIESTE

Codifica in complemento a 2

1) Rappresento i numeri positivi in binario puro + un bit (MSB) per il segno

2) Rappresento i numeri negativi mediante il complemento a 2 del corrispondente positivo

Esempio

In una codifica ad 8 bit (7+segno)

- il numero 9 è rappresentato da 0000 1001
- il numero -9 è rappresentato dal complemento a 2 di (0000 1001)

UNIVERSITÀ DEGLI STUDI DI TRIESTE

Codifica in complemento a 2

Come si calcola complemento a 2 di un numero di n bit?

Invertire tutti i bit del numero di partenza (operatore not),

sommare 1 al risultato,

Scartare eventuale riporto oltre l'n-esimo bit

Nel caso dell'esempio (numero 9) il complemento a 2 di 0000 1001

Not (0000 1001) → 1111 0110

1111 0110 +

0000 0001 =

1111 0111

Quindi -9 → 1111 0111

UNIVERSITÀ DEGLI STUDI DI TRIESTE

Codifica in complemento a 2

1) Il complemento a due del complemento a 2 di n è uguale a n

Complemento a 2 di (1111 0111) → (0000 1000)+

(0000 0001) =

0000 1001

2) Il valore minimo (negativo) rappresentabile su n bit è 2^(n-1)

Esempio: -128 se n=8, -32’768 se n=16

3) Il valore massimo (positivo) rappresentabile su n bit è 2^(n-1)-1

Esempio: +127 se n=8, +32’767 se n=16

4) E’ possibile eseguire la sottrazione tra due numeri attraverso una somma senza la necessità di esaminare il segno dei numeri

A - B = A + (-B) = A + complemento a 2 di B

UNIVERSITÀ DEGLI STUDI DI TRIESTE

Bit, Byte, File e multipli

Il bit è la più piccola quantità d'informazione (0/1),

Byte = insieme ordinato di 8 bit

File: sequenza di byte memorizzata su qualsiasi supporto, con un inizio ed una dimensione ben definiti. È un contenitore di informazioni. L'organizzazione delle informazioni all'interno del file dipende dal formato di file utilizzato.

In ambito informatico sono ampiamente utilizzati multipli (potenze di 2) del byte.

Tuttavia, è entrato nell'uso comune, soprattutto in ambito commerciale, l'impiego delle potenze di 10 del Sistema Internazionale (SI) per definire i multipli del byte:

UNIVERSITÀ DEGLI STUDI DI TRIESTE

Bit, Byte, File e multipli

I multipli SI del byte non coincidono con quelli binari ma ne costituiscono un'approssimazione per difetto

$2^{10} = 1024 \approx 10^3$

Prefissi SI			Prefissi binari		
kilobyte	kB	10^3	*kibibyte*	KiB	2^{10}
megabyte	MB	10^6	*mebibyte*	MiB	2^{20}
gigabyte	GB	10^9	*gibibyte*	GiB	2^{30}
terabyte	TB	10^{12}	*tebibyte*	TiB	2^{40}
petabyte	PB	10^{15}	*pebibyte*	PiB	2^{50}

UNIVERSITÀ DEGLI STUDI DI TRIESTE

Codifica dei testi

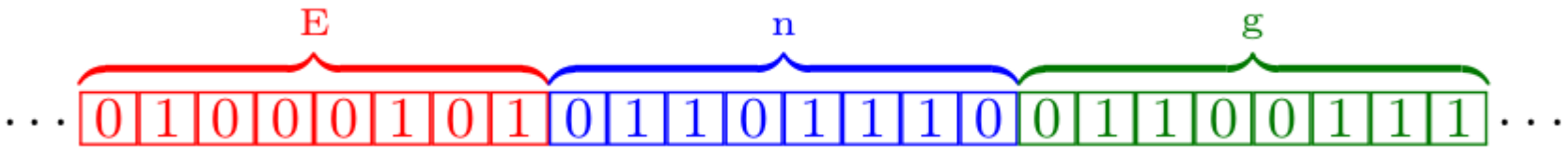
In che modo posso rappresentare un testo in una macchina numerica?

Un testo è una sequenza di caratteri.

Bisogna quindi codificare l'insieme di caratteri che si vuole utilizzare, cioè bisogna associare un numero ad ogni carattere.

Più precisamente, bisogna creare una corrispondenza biunivoca tra l'insieme dei caratteri che si vuole rappresentare (ad esempio alfabeto italiano, caratteri di punteggiatura e spaziatura) e un insieme di numeri interi.

Fissata la codifica, un testo può essere rappresentato in un calcolatore come un file contenente la sequenza delle codifiche (cioè dei bit/byte) dei caratteri del testo da rappresentare. Esempio con 1 byte/carattere:



A seconda del formato di file impiegato, possono essere incluse altre informazioni relative ad impaginazione, colore e font dei caratteri, ecc.

UNIVERSITÀ DEGLI STUDI DI TRIESTE

Codifica dei testi

Nel 1961 è stato introdotto il sistema US-ASCII (o più semplicemente ASCII), divenuto successivamente uno standard (ISO/IEC 646) ed è tutt'ora utilizzato .

ASCII - American Standard Code for Information Interchange

- Codifica a 7 bit
- Codifica di lettere maiuscole, minuscole e cifre segue l'ordine naturale
- Codifica punteggiature, alcuni simboli, spazio
- Codifica caratteri speciali: andare a capo, tabulazione
- Codifica caratteri di controllo "non stampabili"
- Non sono codificate lettere accentate
- Esistono programmi (editor di testo) che permettono di scrivere testo e salvare il corrispondente codice ASCII (in una sequenza di byte) su file:
- Esempio: notepad, gedit, emacs e molti altri

UNIVERSITÀ DEGLI STUDI DI TRIESTE

Codifica dei testi

ASCII TABLE

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	|
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

UNIVERSITÀ DEGLI STUDI DI TRIESTE

Codifica dei testi - esempio

Nel file esempio.txt è memorizzato il testo

<< Corso di Abilita' Informatica >>

Il comando hexdump (Linux) seguito dal nome del file mostra il contenuto del file in formato esadecimale

```
antonio@debian:~$ hexdump -C esempio.txt
00000000  43 6f 72 73 6f 20 64 69  20 41 62 69 6c 69 74 61  |Corso di Abilita|
00000010  27 20 49 6e 66 6f 72 6d  61 74 69 63 61 0a        |' Informatica.|
0000001e
```

UNIVERSITÀ DEGLI STUDI DI TRIESTE

Codifica dei testi

Per codificare caratteri non compresi nella tabella ASCII, come ad esempio lettere accentate, diversi tipi di alfabeto(cirillico, greco, cinese) è necessario estendere la codifica ASCII.

Codifica estesa sfrutta l'ottavo bit:

- se uguale a 0 allora il carattere è ASCII
- se uguale a 1 allora altro tipo di codifica

Esistono varie codifiche estese che sfruttano l'ottavo bit per codificare altri caratteri.

Una molto comune è la UTF-8

- https://it.wikipedia.org/wiki/UTF-8
- http://www.utf8-chartab

0	c_7	c_6	c_5	c_4	c_3	c_2	c_1	c_0

N.B. La memoria dei calcolatori è organizzata in byte o multipli di byte, la codifica ASCII utilizza i 7 bit meno significativi di un byte. L'ottavo bit, il bit più significativo, è posto a 0.

UNIVERSITÀ DEGLI STUDI DI TRIESTE

Codifica di immagini

Per rappresentare un'immagine in una macchina numerica occorre poter tradurre l'immagine in una sequenza di numeri.

Esistono molti modi diversi per codificare un'immagine, ma in generale si può dire che

Un'immagine è bidimensionale,

E' possibile approssimare con una griglia sufficientemente fitta

Ogni elemento della griglia (picture element o pixel) può essere rappresentato da uno o più numeri che ne definiscono il colore (o il livello di grigio nel caso di immagine in bianco e nero)

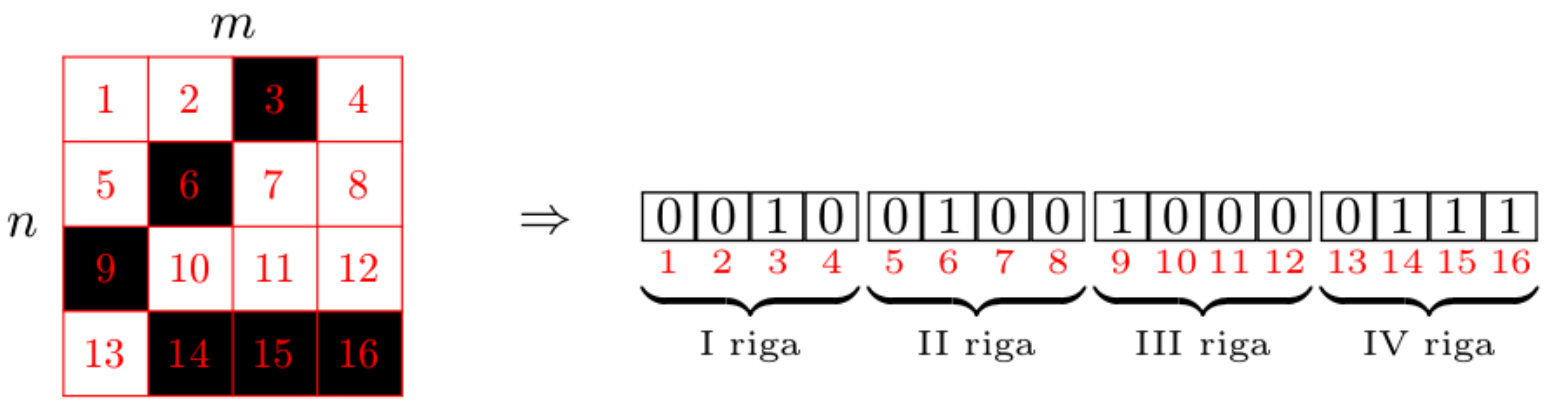
l'immagine è quindi codificata, ad esempio, con la sequenza dei colori dei pixel a partire dall'angolo in alto a sinistra

UNIVERSITÀ DEGLI STUDI DI TRIESTE

Codifica di immagini B/N

Consideriamo il caso più semplice di immagini in bianco e nero (B/N).

Ogni pixel può assumere solo due valori, perciò basta l'informazione di un bit per identificarlo: 0 (bianco) e 1 (nero), per esempio ⇒ codifica B/N. Un'immagine B/N con risoluzione m colonne × n righe può essere quindi rappresentata in un calcolatore come una sequenza di m · n bit in un determinato ordine (per righe/colonne), oppure da una matrice di n righe e m colonne:



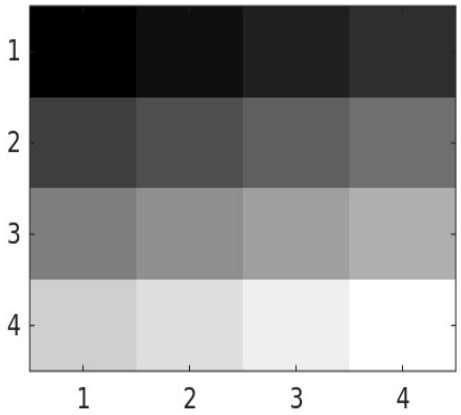
In fotografia, il termine B/N ri riferisce alle immagini in scala di grigi. Le fotografie su pellicola analogica e le relative stampe su carta sono in "B/N".

Codifica di immagini in scala di grigi

Un caso ancora semplice ma più evoluto rispetto al caso B/N è costituito dalle immagini in scala di grigi.

Ogni pixel può assumere M (più di 2) valori, quindi per ogni pixel occorrono $n = \log_2 M$ bit

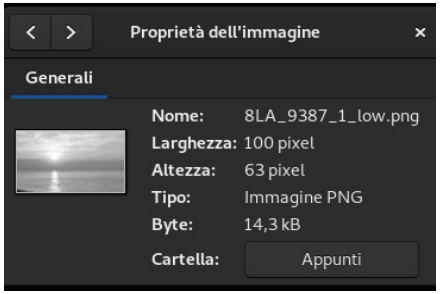
Un'immagine in scala di grigi con risoluzione x colonne × y righe può essere rappresentata come una matrice di $x \cdot y$ (gruppi di n bit)



0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

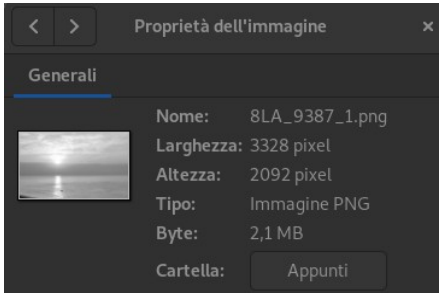
UNIVERSITÀ DEGLI STUDI DI TRIESTE

Codifica di immagini



UNIVERSITÀ DEGLI STUDI DI TRIESTE

Codifica di immagini



UNIVERSITÀ DEGLI STUDI DI TRIESTE

Energia

Energia di un segnale tempo-continuo in un intervallo [a,b]

$$E_x = \varepsilon \int_a^b |x(t)|^2 dt$$

Dato che eseguiamo il calcolo numericamente, dobbiamo utilizzare un algoritmo per e:

$$\int_a^b f(x)dx \approx \sum_{i=0}^{i=n-1} (x_{i+1}-x_i)\frac{1}{2}[f(x_{i+1})+f(x_i)]$$

Se dividiamo [a,b] in parti n parti uguali

$$x_{i+1}-x_i = \Delta x = \frac{b-a}{n}$$

Otteniamo

$$\int_a^b f(x)dx \approx \frac{1}{2}\cdot\frac{b-a}{n}\sum_{i=0}^{i=n-1}[f(x_{i+1})+f(x_i)]$$

$$\sum_{i=0}^{i=n-1}[f(x_{i+1})+f(x_i)] = \sum_{i=0}^{i=n-1} f(x_{i+1}) + \sum_{i=0}^{i=n-1} f(x_i) = \sum_{i=0}^{i=n-1} f(x_i) + f(b) + \sum_{i=0}^{i=n-1} f(x_i)$$

$$-f(a)+2\sum_{i=0}^{i=n-1} f(x_i) = -f(a)+2\left(\sum_{i=0}^{i=n} f(x_i) - f(b)\right) = -f(a)-2f(b)+2\left(\sum_{i=0}^{i=n} f(x_i)\right)$$

$$\int_a^b f(x)dx \approx \frac{(b-a)}{n}\left(\sum_{i=0}^{i=n} f(x_i) - \frac{1}{2}f(a) - f(b)\right)$$

- tri.m
- % Create a triangular pulse with width 0.4.
- % fs = 10000; % Sampling frequency (samples/sec)
- % t = -1:1/fs:1; % Time Vector
- % w = .4; % Triangle Width

```
fs = 10000;      % Sampling frequency (samples/sec)
t = -1:1/fs:1;   % Time Vector
w = .4;          % Triangle Width
x = tripuls(t-0.5,w);  % Sampled aperiodic triangle
plot(t,x);
xlabel('Time (sec)');ylabel('Amplitude')
```

- % See also SQUARE, S
- % RECTPULSE,