



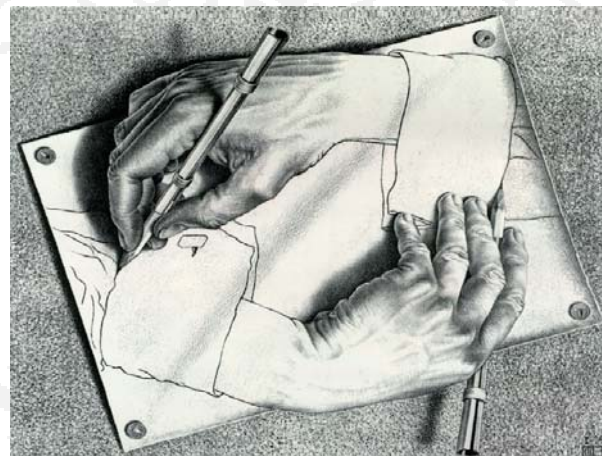
UNIVERSITÀ DEGLI STUDI DI
POLI FEDERICO II

Progetto Lauree Scientifiche

Algebra Lineare Numerica ed Elaborazione delle Immagini

ovvero:

una applicazione della Matematica Numerica.



Salvatore Cuomo
salvatore.cuomo@unina.it

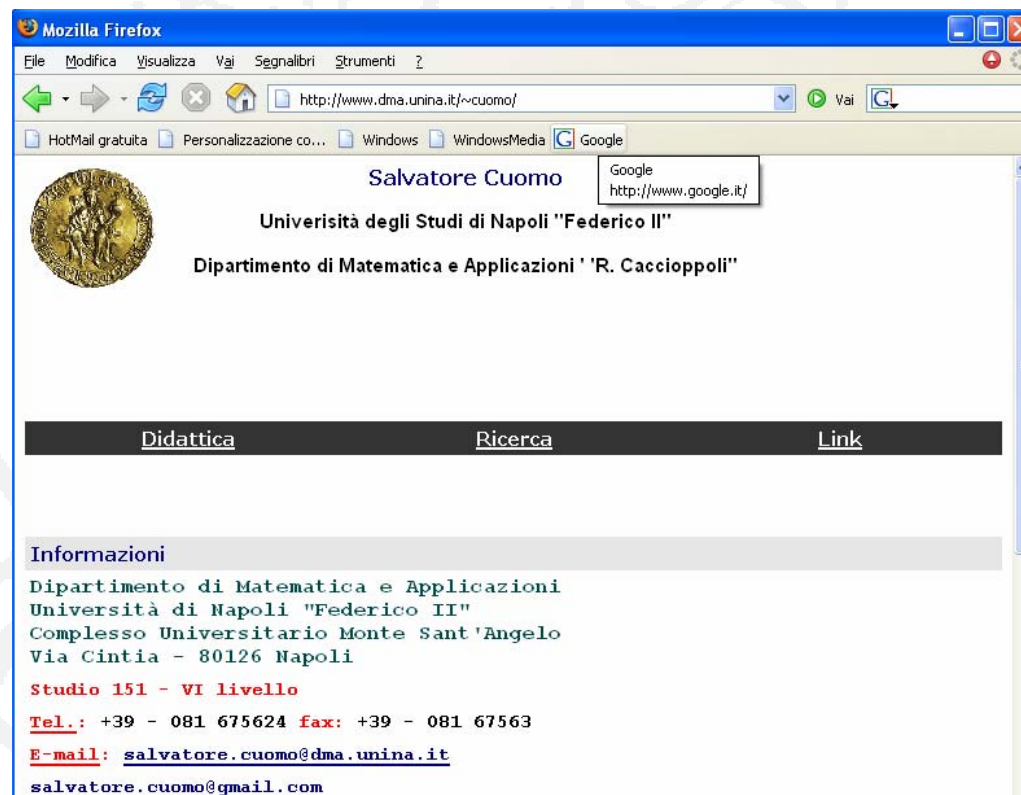
Obiettivo del seminario

- ❑ Introdurre concetto di immagine digitale dal punto di vista di un oggetto che “vive” e viene elaborato attraverso modelli più o meno complicati della Matematica.
- ❑ Il discente è “immerso” in quella che viene generalmente definita come “era del digitale”, in cui tutta la tecnologia adoperata quotidianamente (immagini al calcolatore, filmanti e video digitali, tv digitale terrestre, file mp3, ...) é il risultato della codifica e dell’elaborazione numerica dell’informazione (stringe binarie di informazione).
- ❑ In tale ottica non è dunque possibile trascurare l’aspetto modellistico matematico e numerico delle applicazioni tecnologiche del vasto campo dell’elaborazione delle immagini.

Alcune informazioni sul relatore

Web: <http://www.dma.unina.it/~cuomo/>

E-mail: salvatore.cuomo@unina.it



Contenuti del Seminario

- Nozioni introduttive sull'elaborazione delle immagini
- Principali operazioni di manipolazione
- Principali metodologie di elaborazione
-

Al fine di: Individuare i principali collegamenti tra l'Algebra
Lineare Numerica e l'Elaborazione delle Immagini
gli argomenti del seminario
saranno affrontati mediante l'ambiente di
risoluzione di problemi (PSE) MATLAB
(ed una panoramica su PSE "liberi" OCTAVE, SCILAB,...)

Partimamo da un Problema Reale...

Creare di uno modello (matematico)
per un'immagine
sfocata e perturbata



Un'immagine u
acquisita da un generico dispositivo ottico
subisce una variazione di luminosità



Immagine originale

u

Tale variazione
si manifesta
sull'immagine
con un effetto
di *sfocamento*

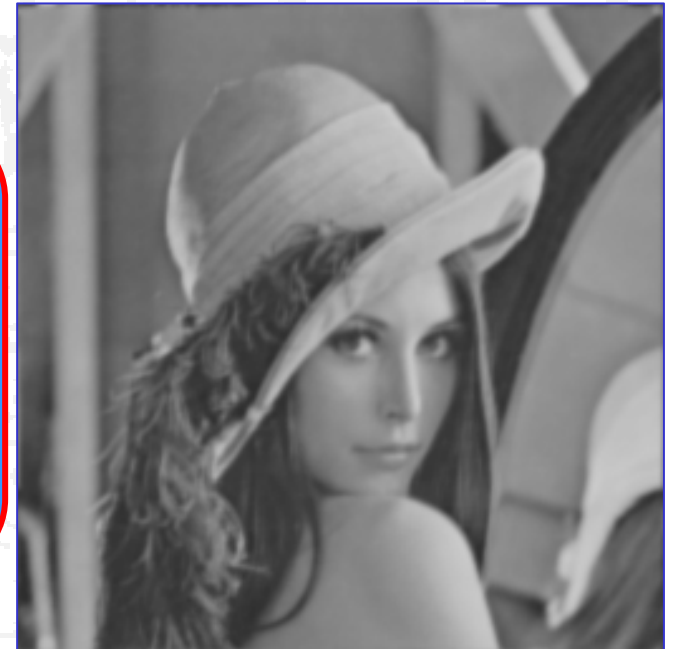
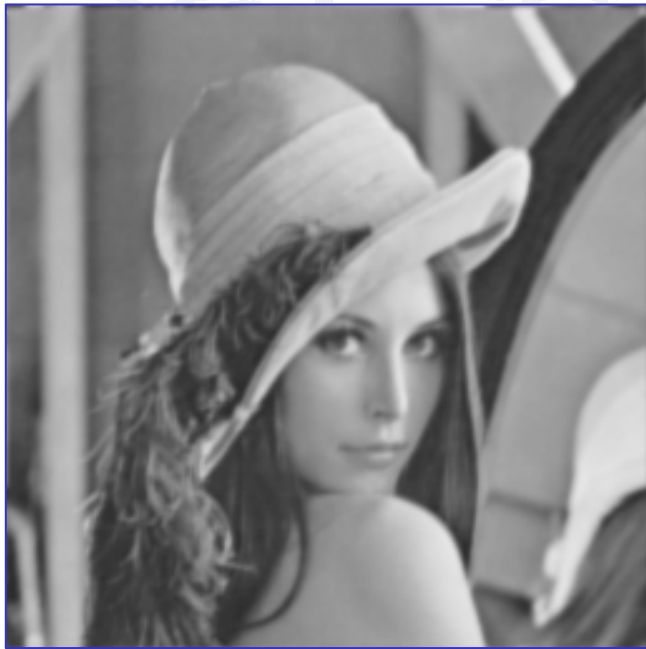


Immagine sfocata

$z = Ku$

A tale effetto di sfocamento (inevitabile),
si aggiungono **disturbi** di natura varia,
legati comunque al processo di
formazione e trasmissione dell'immagine



Tali disturbi
si manifestano
sull'immagine
perturbandola
ulteriormente,
e sono indicati
comunemente
col nome di
rumore



Immagine sfocata

$$z = Ku$$

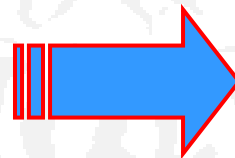


Immagine rumorosa

$$z_1 = Ku + \eta$$

Il processo di acquisizione e
formazione dell'immagine u
È descritto dal seguente

Modello matematico:

$$z_1 = Ku + \eta$$

K descrive il
dispositivo ottico che acquisisce l'immagine
(ad es. una telecamera,...)
e può essere descritto da un integrale del tipo:

$$Ku(x, y) = \iint_{\mathbb{R}^2} k(x, y; x', y') u(x', y') \, dx' dy'$$

$\eta(x, y)$ è una funzione che descrive
il rumore che perturba l'immagine.
La sua espressione matematica non è
nota a priori.



Immagine rumorosa

$$z_1 = Ku + \eta$$

A partire
dall'immagine
rumorosa
si vuole
ricostruire
l'immagine
originale

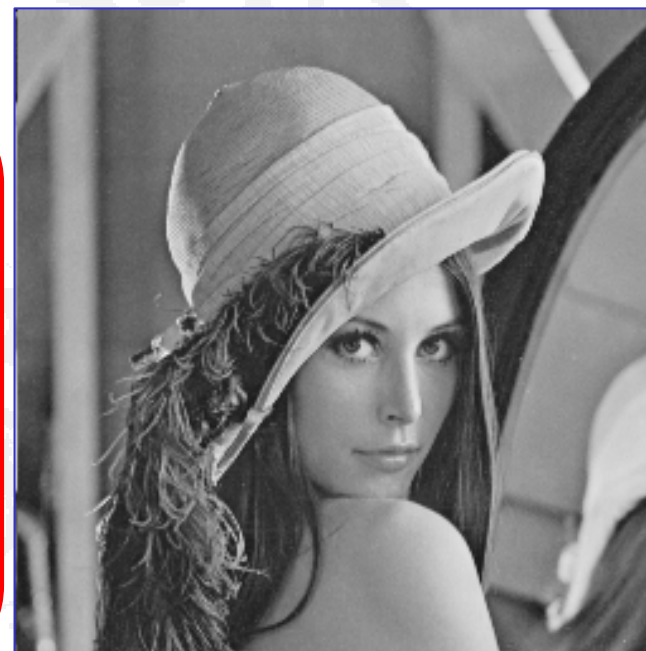


Immagine originale

u

?

Come ottenere u da z_1 ?

Partiamo da una domanda...

Cosa e' Una Immagine digitale ?

ovvero

*Come rappresentare mediante strumenti
della matematica le immagini ?*

(e più in generale filmati, suoni,...)

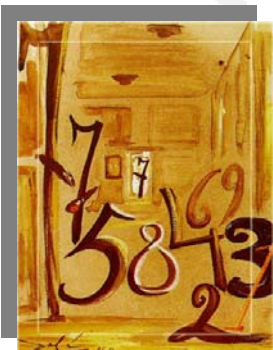




Immagine rumorosa

$$z_1 = Ku + \eta$$



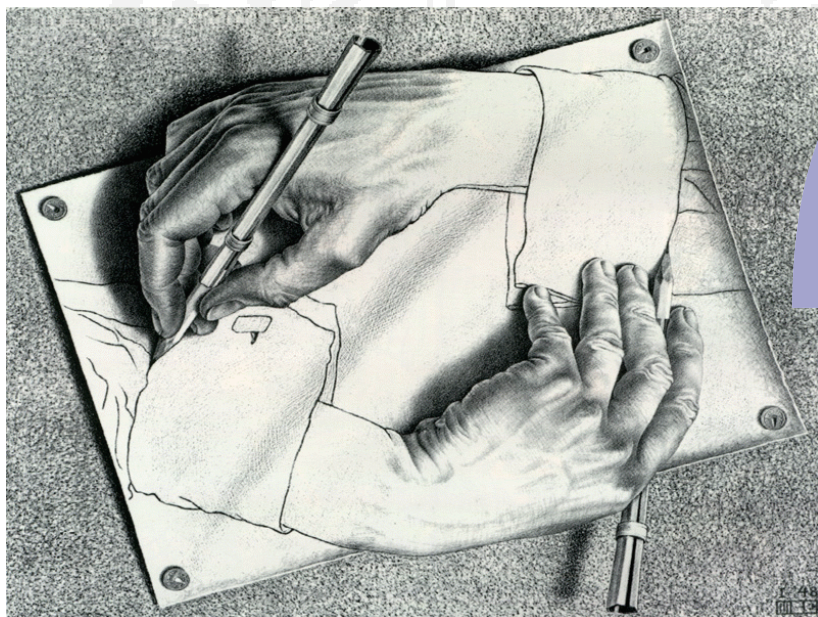
Immagine originale

u

Per risolvere problemi che riguardano l'elaborazione delle immagini dobbiamo generalmente “avere a che fare” con particolari oggetti matematici

LE MATRICI

Immagine digitale a scala di grigi

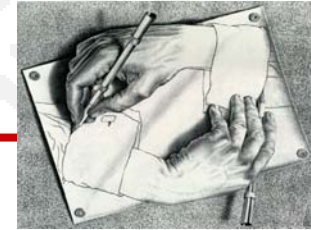


Per capirci:

Una Tabella **[Matrice]** di Numeri Interi i cui indici di riga e colonna **[Pixel]** definiscono la posizione sullo schermo ed il valore assunto il Livello di Luminosità **[Livello di Grigio]**

8	193	199	228	238	226	242	244	244	247	235	229	243	218	195	188	162
	196	203	220	221	204	235	235	225	238	244	230	222	218	198	195	175
6	192	195	199	186	202	211	208	199	207	224	228	220	214	199	202	185
	164	163	160	146	122	98	102	114	111	124	160	183	194	189	202	195
4	132	134	143	133	131	115	110	104	71	43	53	79	127	139	172	178
	134	147	173	167	165	178	164	141	121	82	51	56	79	104	151	165
2	155	178	212	200	199	172	144	134	135	123	103	93	68	99	151	165
	158	184	217	197	175	69	19	33	37	51	68	58	47	78	131	144
0	52	82	145	197	157	44	1	5	16	9	2	8	3	7	23	46
	58	66	47	17	46	0	0	0	0	0	0	5	8	1	0	0
-2	67	115	112	75	26	0	0	0	0	8	29	30	16	9	0	0
	3	51	117	156	182	174	177	153	120	79	44	5	0	0	2	3
-4	0	0	24	47	103	99	97	74	44	11	4	0	0	0	8	12
	35	28	15	6	27	19	8	3	10	9	32	33	23	31	32	25
-6	27	26	45	63	73	74	60	58	74	73	85	67	71	72	63	48
	68	73	83	80	91	119	122	115	123	118	133	113	104	106	99	85
-8																
	-8	-6	-4	-2	0	2	4	6	8							

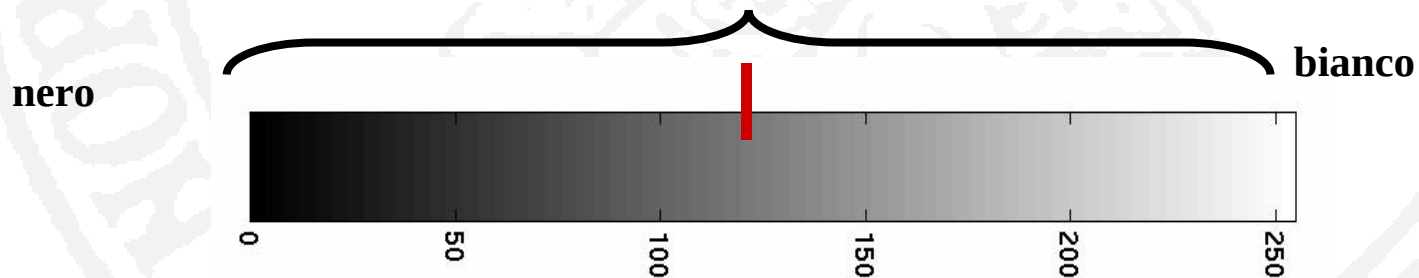
Formalmente (Modello continuo)



Una funzione di due variabili definita in un sottoinsieme D di $\mathbb{R}^2$ a valori interi:

$$f: (x,y) \in D \subset \mathbb{R}^2 \longrightarrow f(x,y) \in [0, I-1]$$

dove



f è anche detta funzione di “Intensità Luminosa”

E una Immagine digitale a Colori?



1

8	193	199	228	238	226	242	244	244	247	235	229	243	218	195	188	162
6	196	203	220	221	204	235	235	225	238	244	230	222	218	198	195	175
4	192	195	199	186	202	211	208	199	207	224	228	220	214	199	202	185
2	164	163	160	146	122	98	102	114	111	124	160	183	194	189	202	195
0	132	134	143	133	131	115	110	104	71	43	53	79	127	139	172	178
-2	134	147	173	167	165	178	164	141	121	82	51	56	79	104	151	165
-4	155	178	212	200	199	172	144	134	135	123	103	93	68	99	151	165
-6	158	184	217	197	175	69	19	33	37	51	68	58	47	78	131	144
-8	52	82	145	197	157	44	1	5	16	9	2	8	3	7	23	46
	58	66	47	17	46	0	0	0	0	0	0	0	5	8	1	0
	67	115	112	75	26	0	0	0	0	8	29	30	16	9	0	0
	3	51	117	156	182	174	177	153	120	79	44	5	0	0	2	3
	0	0	24	47	103	99	97	74	44	11	4	0	0	0	8	12
	35	28	15	6	27	19	8	3	10	9	32	33	23	31	32	25
	27	26	45	63	73	74	60	58	74	73	85	67	71	72	63	48
	68	73	83	80	91	119	122	115	123	118	133	113	104	106	99	85

2

8	193	199	228	238	226	242	244	244	247	235	229	243	218	195	188	162
6	196	203	220	221	204	235	235	225	238	244	230	222	218	198	195	175
4	192	195	199	186	202	211	208	199	207	224	228	220	214	199	202	185
2	164	163	160	146	122	98	102	114	111	124	160	183	194	189	202	195
0	132	134	143	133	131	115	110	104	71	43	53	79	127	139	172	178
-2	134	147	173	167	165	178	164	141	121	82	51	56	79	104	151	165
-4	155	178	212	200	199	172	144	134	135	123	103	93	68	99	151	165
-6	158	184	217	197	175	69	19	33	37	51	68	58	47	78	131	144
-8	52	82	145	197	157	44	1	5	16	9	2	8	3	7	23	46
	58	66	47	17	46	0	0	0	0	0	0	0	5	8	1	0
	67	115	112	75	26	0	0	0	0	8	29	30	16	9	0	0
	3	51	117	156	182	174	177	153	120	79	44	5	0	0	2	3
	0	0	24	47	103	99	97	74	44	11	4	0	0	0	8	12
	35	28	15	6	27	19	8	3	10	9	32	33	23	31	32	25
	27	26	45	63	73	74	60	58	74	73	85	67	71	72	63	48
	68	73	83	80	91	119	122	115	123	118	133	113	104	106	99	85

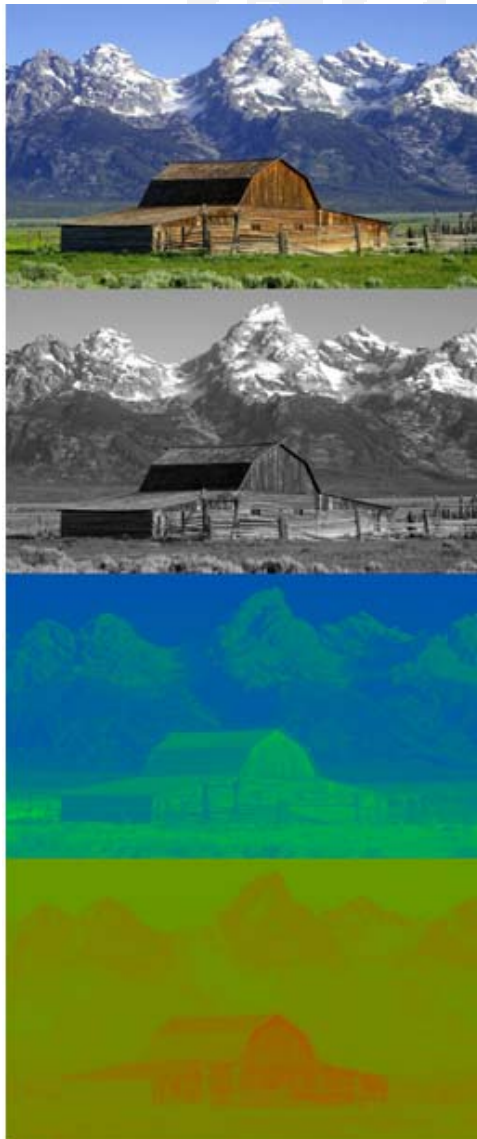
3

8	193	199	228	238	226	242	244	244	247	235	229	243	218	195	188	162
6	196	203	220	221	204	235	235	225	238	244	230	222	218	198	195	175
4	192	195	199	186	202	211	208	199	207	224	228	220	214	199	202	185
2	164	163	160	146	122	98	102	114	111	124	160	183	194	189	202	195
0	132	134	143	133	131	115	110	104	71	43	53	79	127	139	172	178
-2	134	147	173	167	165	178	164	141	121	82	51	56	79	104	151	165
-4	155	178	212	200	199	172	144	134	135	123	103	93	68	99	151	165
-6	158	184	217	197	175	69	19	33	37	51	68	58	47	78	131	144
-8	52	82	145	197	157	44	1	5	16	9	2	8	3	7	23	46
	58	66	47	17	46	0	0	0	0	0	0	0	5	8	1	0
	67	115	112	75	26	0	0	0	0	8	29	30	16	9	0	0
	3	51	117	156	182	174	177	153	120	79	44	5	0	0	2	3
	0	0	24	47	103	99	97	74	44	11	4	0	0	0	8	12
	35	28	15	6	27	19	8	3	10	9	32	33	23	31	32	25
	27	26	45	63	73	74	60	58	74	73	85	67	71	72	63	48
	68	73	83	80	91	119	122	115	123	118	133	113	104	106	99	85

Per capirci:

3 Tabella **[Matrici]** di Numeri Interi i cui
indici di riga e colonna **[Pixel]** definiscono
la posizione sullo schermo ed i valore
assunto il Livello di Luminosità **[Livello di
Rosso, Blu, Verde RGB]**

Immagine a colori(cont.)



Y

U

V

Modello
del colore



R

G

B

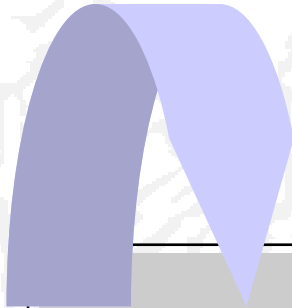
Formalmente (Modello continuo)



Una funzione di due variabili definita in un sottoinsieme D di $\mathbb{R}^2$ a valori vettoriali:

$$f: (x,y) \in D \subset \mathbb{R}^2 \longrightarrow (f_1(x,y), f_2(x,y), f_3(x,y)) \in [0, I_1-1] \times [0, I_2-1] \times [0, I_3-1]$$

f è anche detta funzione
di “Intensità Luminosa”



8

Riga
8

193	199	228	238	226	242	244	244	247	235	229	243	218	195	188	162
196	203	220	221	204	235	235	225	238	244	230	222	218	198	195	175
192	195	199	186	202	211	208	199	207	224	228	220	214	199	202	185
164	163	160	146	122	98	102	114	111	124	160	183	194	189	202	195
132	134	143	133	131	115	110	104	71	43	53	79	127	139	172	178
134	147	173	167	165	178	164	141	121	82	51	56	79	104	151	165
155	178	212	200	199	172	144	134	135	123	103	93	68	99	151	165
158	184	217	197	175	69	19	33	37	51	68	58	47	78	131	144
52	82	145	197	157	44	1	5	16	9	2	8	3	7	23	46
58	66	47	17	46	0	0	0	0	0	0	5	8	1	0	0
67	115	112	75	26	0	0	0	0	8	29	30	16	9	0	0
3	51	117	156	182	174	177	153	120	79	44	5	0	0	2	3
0	0	24	47	103	99	97	74	44	11	4	0	0	0	8	12
35	28	15	6	27	19	8	3	10	9	32	33	23	31	32	25
27	26	45	63	73	74	60	58	74	73	85	67	71	72	63	48
68	73	83	80	91	119	122	115	123	118	133	113	104	106	99	85

Formalmente (Modello Discreto)



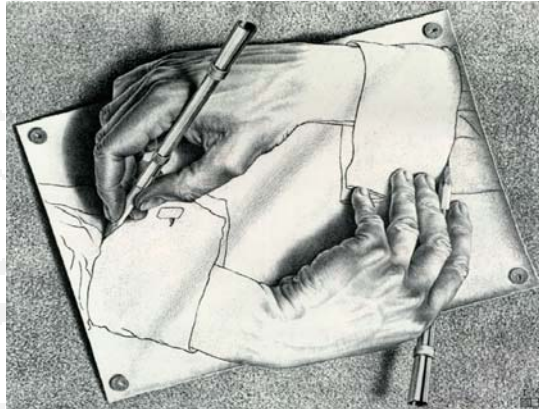
Una matrice rettangolare $M \times N$ costituita da opportuni campioni di valori di $f(x,y) \in [0, I-1]$

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \cdots & f(0,N-1) \\ f(1,0) & f(1,1) & \cdots & f(1,N-1) \\ \vdots & \vdots & \vdots & \vdots \\ f(M-1,0) & f(M-1,1) & \cdots & f(M-1,N-1) \end{bmatrix}$$

f è anche detta matrice di “Intensità Luminosa”

In particolare (Modello Discreto)

La matrice f che rappresenta una immagine digitale:



- è costituita da $M \times N$ pixel (picture elements), ciascuno dei quali ha un valore appartenente all'intervallo $[0, I-1]$
- Tipicamente $I = 2^k$, dove k è il numero di bit usati per codificare ciascun pixel

Esempio (Modello Discreto)

Sia f una matrice che rappresenta una immagine monocromatica



Se si utilizzano **8 bit** si rappresentano i numeri

0 0 0 0 0 0 0 0

0

...

1 1 1 1 1 1 1 1

255



dunque mediante un “modesto” si ha la possibilità di
rappresentare un numero di livelli **256** di grigio
(buon numero per l'occhio umano)

Esempio (Modello Discreto)

Se f è una immagine monocromatica tipica 512×512



Ricordando che ogni intero necessita di
8 bit per essere memorizzata



f occupa $(512 * 512 * 8)$

≈ 2 Mbit ≈ 256 kbyte di memoria

(1 bit sono circa 8 byte)

(informazioni sufficienti!)



Immagine rumorosa

$$z_1 = Ku + \eta$$



Immagine originale

u

Costruiamo adesso una
immagine sfocata e perturbata mediante
l'uso del modello discreto (battaglia
navale) proposto

LE MATRICI

(come linguaggio di descrizione useremo Matlab)

Esempio:



Immagine originale

u

I primi 15 x 15 elementi
di u

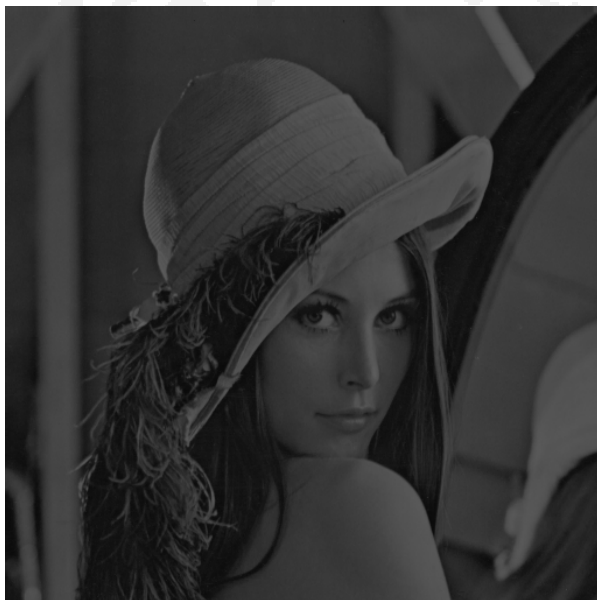
```
>> u=imread('lena512.jpg');  
>> size(u)  
      512 x 512
```

162	162	162	161	162	157	163	161	166	162	162	160	155	163	160
162	162	162	161	162	157	163	161	166	162	162	160	155	163	160
162	162	162	161	162	157	163	161	166	162	162	160	155	163	160
162	162	162	161	162	157	163	161	166	162	162	160	155	163	160
162	162	162	161	162	157	163	161	166	162	162	160	155	163	160
164	164	158	155	161	159	159	160	161	160	155	159	154	154	156
160	160	163	158	160	162	159	156	159	163	158	163	155	155	156
159	159	155	157	158	159	156	157	159	162	162	160	153	153	153
155	155	158	158	159	160	157	157	163	157	159	160	159	154	156
155	155	157	158	155	154	155	157	161	155	155	157	157	157	157
156	156	156	160	156	155	155	152	159	159	157	157	159	154	158
156	156	156	159	159	155	150	148	160	159	158	158	156	155	157
158	158	157	156	157	153	159	156	161	158	160	157	156	153	156
157	157	157	157	160	157	156	156	159	155	154	158	156	155	154
158	158	159	155	155	158	158	156	157	156	151	160	157	156	156

Esempio: Sfocamento

Supponiamo che $K=0.4$ e moltiplichiamo scalarmente:

$$z = K u$$



65	65	65	64	65	63	65	64	66	65	65	64	62	65	64
65	65	65	64	65	63	65	64	66	65	65	64	62	65	64
65	65	65	64	65	63	65	64	66	65	65	64	62	65	64
65	65	65	64	65	63	65	64	66	65	65	64	62	65	64
65	65	65	64	65	63	65	64	66	65	65	64	62	65	64
66	66	63	62	64	64	64	64	64	64	62	64	62	62	62
64	64	65	63	64	65	64	62	64	65	63	65	62	62	62
64	64	62	63	63	64	62	63	64	65	65	64	61	61	61
62	62	63	63	64	64	63	63	65	63	64	64	64	62	62
62	62	63	63	62	62	62	63	64	62	62	63	63	63	63
62	62	62	64	62	62	62	61	64	64	63	63	64	62	63
62	62	62	64	64	62	60	59	64	64	63	63	62	62	63
63	63	63	62	63	61	64	62	64	63	64	63	62	61	62
63	63	63	63	64	63	62	62	64	62	62	63	62	62	62
63	63	64	62	62	63	63	62	63	62	60	64	63	62	62

Moltiplicando per uno scalare abbiamo modificato l'intensità luminosa ovvero abbiamo **scurito l'immagine**.

Algebra Lineare Numerica

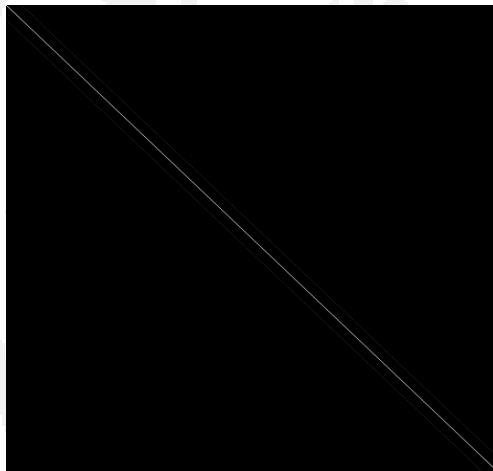
Moltiplicazione per uno scalare:

E' un'operazione che, data una matrice A ed un numero c (detto *scalare*), costruisce una nuova matrice cA , il cui elemento è ottenuto moltiplicando l'elemento corrispondente di A per c ; la matrice e lo scalare scelti devono appartenere allo stesso campo. Formalmente:

$$(cA)_{ij} := cA_{ij}.$$

$$2 \begin{bmatrix} 1 & 8 & -3 \\ 4 & -2 & 5 \end{bmatrix} = \begin{bmatrix} 2 \times 1 & 2 \times 8 & 2 \times -3 \\ 2 \times 4 & 2 \times -2 & 2 \times 5 \end{bmatrix} = \begin{bmatrix} 2 & 16 & -6 \\ 8 & -4 & 10 \end{bmatrix}.$$

Supponiamo che K sia una matrice (matrice di sfocamento)
e moltiplichiamola con u



K

$$\mathbf{z} = \mathbf{K} \mathbf{u}$$

```
>> K=blur(16,32);  
>> size(K)  
      512 x 512
```

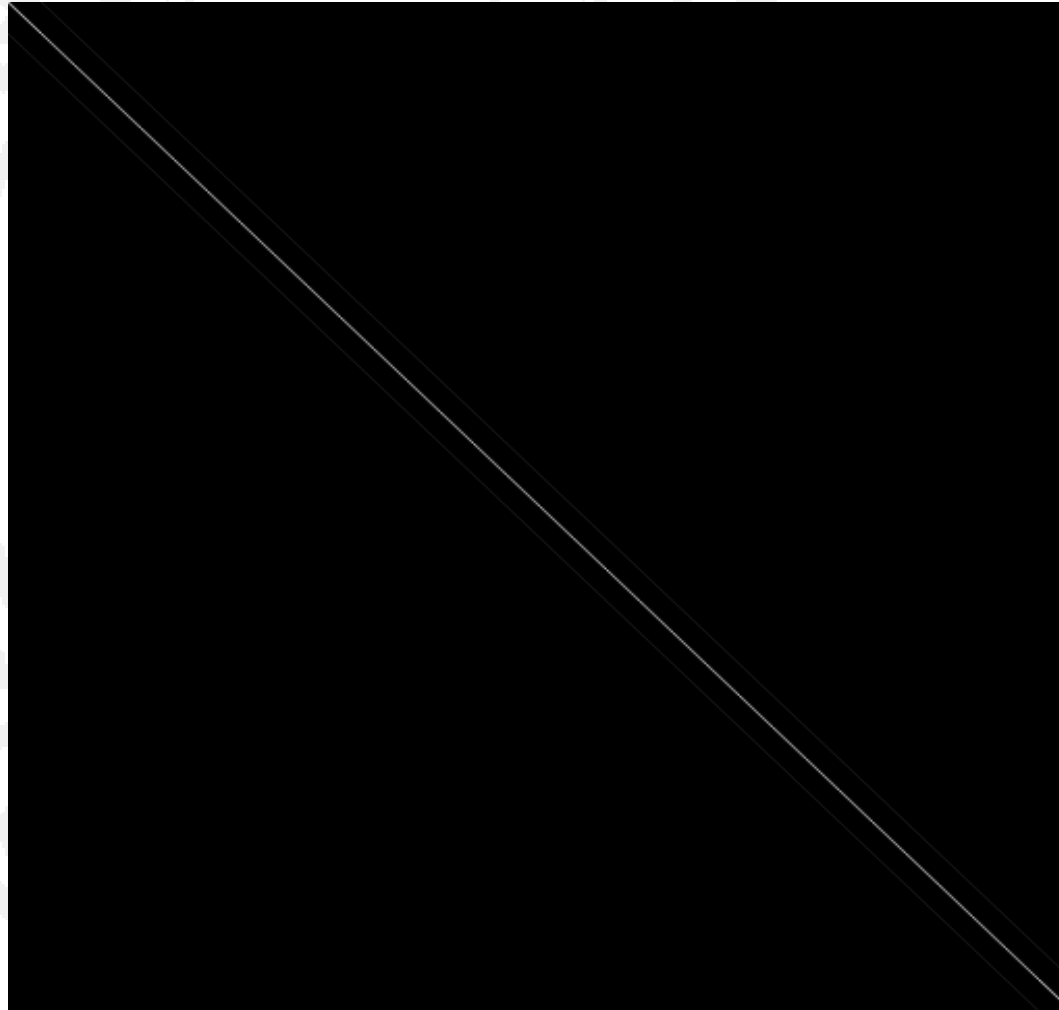
Matrice di sfocamento

I primi 10×10 elementi
di K

[illegible]

Esempio: Sfocamento ZOOM

K

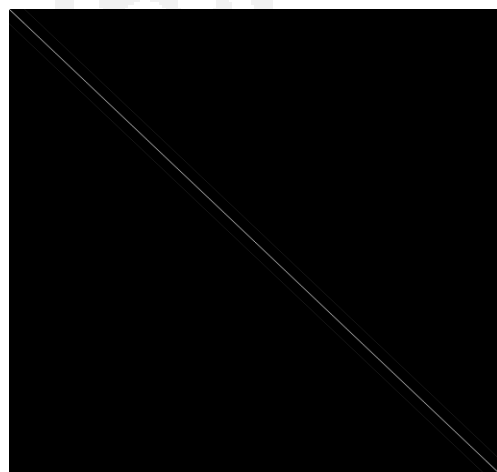


K “in gergo” una matrice saparsa (ovvero ha molti elementi nulli)

Esempio: Sfocamento

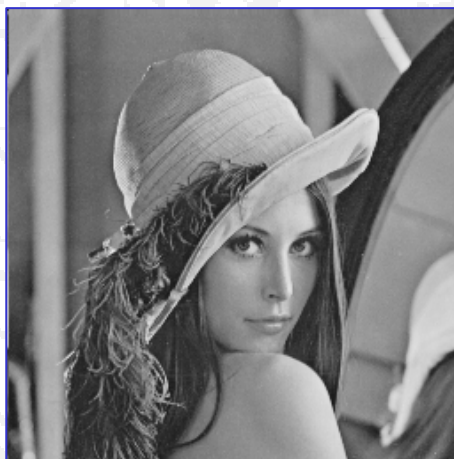
Supponiamo che K sia una matrice (matrice di sfocamento)
e moltiplichiamola con u

$$z = K u$$



K

*



u

=



z

Esempio: Sfocamento ZOOM

z



z è l'immagine sfocata

Algebra Lineare Numerica

Prodotto di matrici :

Sia A una matrice $m \times n$ e B una matrice $n \times k$, si definisce *prodotto tra le matrici A e B* la matrice $C = A B$ il cui generico elemento c_{ij} è la somma dei prodotti degli elementi della i-esima riga di A per i corrispondenti elementi della j-esima colonna di B, ovvero:

$$c_{ij} = \sum_{z=1}^n a_{iz} b_{zj}$$

$$A = \begin{bmatrix} 2 & 1 & -1 \\ -2 & 3 & 4 \end{bmatrix}$$

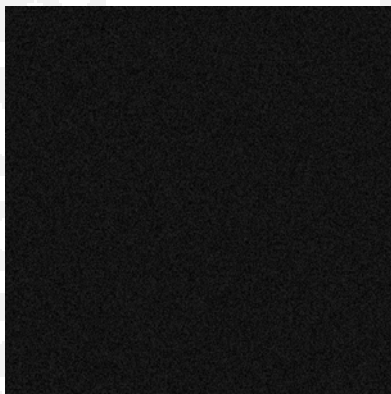
$$B = \begin{bmatrix} 1 & 1 & 2 & 2 \\ 2 & -3 & 1 & 3 \\ -1 & -1 & 5 & 2 \end{bmatrix}$$

$$C = A B = \begin{bmatrix} 5 & 0 & 0 & 5 \\ 0 & -15 & 19 & 13 \end{bmatrix}$$

$$c_{11} = [2 \ 1 \ -1] \begin{bmatrix} 1 \\ 2 \\ -1 \end{bmatrix} = (2 \cdot 1 + 1 \cdot 2 + (-1) \cdot (-1)) = 5$$

Esempio: Rumore

Supponiamo che η sia una matrice (matrice di del rumore) che contenga numeri “pseudocasuali” generati secondo distribuzione unif.



η

```
>> K=40*rand(512);  
>> size(K)  
512 x 512
```

Matrice del rumore

I primi 10 x 10 elementi
di η

32.5889	21.7554	7.2247	38.4623	9.9580	35.8564	16.0945	11.5059	27.5774	1.4009
36.2317	28.8419	1.8020	30.4966	19.0314	19.2892	35.3693	3.3851	2.0182	16.2190
5.0795	20.8998	28.9269	0.2939	15.9630	0.5637	28.0232	23.2873	7.3774	9.9868
36.5350	39.7482	13.8975	27.2015	23.9775	24.9152	9.6749	6.1227	1.8263	19.2359
25.2944	8.7471	26.4247	28.2380	32.0209	9.2438	30.3933	2.9238	35.4017	35.2336
3.9016	4.2319	15.3547	25.8052	4.2028	21.0974	11.6370	23.2226	33.5918	11.2274
11.1399	4.3879	25.0939	22.0924	32.8577	28.9997	11.0975	11.4806	4.7262	23.9657
21.8753	2.5437	0.8660	8.7243	33.6435	24.2966	0.2443	14.4768	16.4166	1.0490
38.3003	16.1832	36.4228	30.8946	14.1802	23.5347	14.9884	28.9930	4.8091	6.2079
38.5955	17.9349	32.0223	9.1211	17.2028	17.3374	17.4773	34.3325	22.8837	33.3564

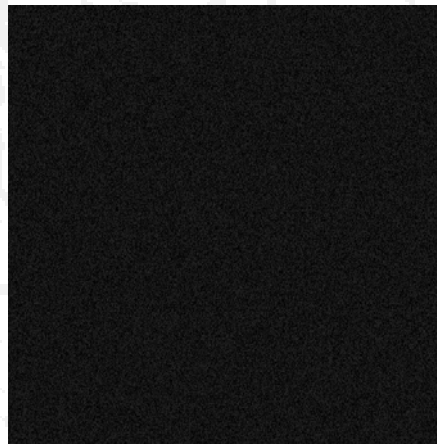
Esempio: Aggiungere rumore

Supponiamo che η sia una matrice del rumore e aggiungiamola (rumore di natura additiva) alla immagine di partenza

$$z_r = u + \eta$$



+



=



Esempio: Rumore ZOOM

z_r



Raddoppiando il livello di rumore

(Matlab: `imshow(uint8(double(u)+80*NOISE))`)

Algebra Lineare Numerica

La somma di matrici :

Due matrici A e B , entrambe di tipo, possono essere sommate. La loro somma $A + B$ è definita come la matrice i cui elementi sono ottenuti sommando i corrispettivi elementi di A e B . Formalmente:

$$(A + B)_{ij} := A_{ij} + B_{ij}$$

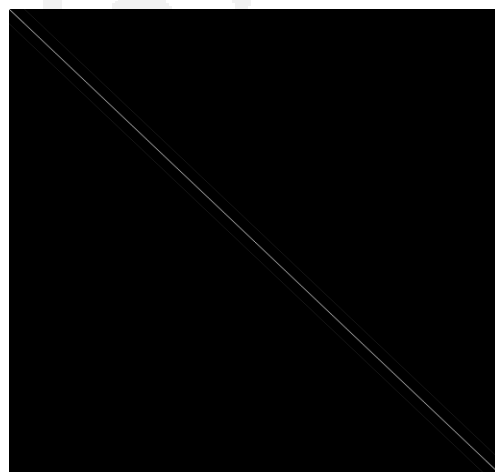
Per esempio

$$\begin{bmatrix} 1 & 3 & 2 \\ 1 & 0 & 0 \\ 1 & 2 & 2 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 5 \\ 7 & 5 & 0 \\ 2 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1+0 & 3+0 & 2+5 \\ 1+7 & 0+5 & 0+0 \\ 1+2 & 2+1 & 2+1 \end{bmatrix} = \begin{bmatrix} 1 & 3 & 7 \\ 8 & 5 & 0 \\ 3 & 3 & 3 \end{bmatrix}.$$

Esempio: Sfocamento + Rumore

Siamo adesso in grado di perturbare e sfocare una immagine

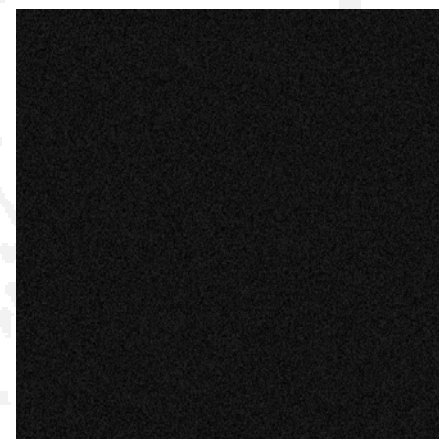
$$z_1 = K u + \eta$$



K



u



η

Quale risultato otteniamo ?

Esempio: Sfocamento + rumore

z_1



Immagine sfocata e perturbata

Altre operazioni

Supponiamo di volere scambiare le righe e le colonne di una Immagine (operazione di trasposizione) si ottiene...



Matrice originale



Matrice trasposta

```
>> u1=u';  
>> size(u1)  
      512 x 512
```

Algebra Lineare Numerica

Prodotto di matrici :

l'operatore di trasposizione, che si denota con un apice o con una T ad esponente, associa ad una matrice la sua relativa **trasposta**, ovvero la matrice il cui generico elemento con indici (i,j) è l'elemento con indici (j,i) della matrice originaria. In simboli:

$$(A^T)_{ij} = A_{ji}, \quad \forall A \in \mathbf{K}^{m,n}, 1 \leq i \leq m, 1 \leq j \leq n$$

Praticamente, la matrice trasposta si deve intendere come una matrice in cui le colonne diventano righe e le righe diventano colonne

$$A = \begin{pmatrix} 1 & 4 & 7 \\ 3 & 2 & 0 \\ 5 & 3 & 1 \\ 0 & 1 & 0 \end{pmatrix} \quad A^T = \begin{pmatrix} 1 & 3 & 5 & 0 \\ 4 & 2 & 3 & 1 \\ 7 & 0 & 1 & 0 \end{pmatrix}$$

Elaborazione digitale delle immagini

Un campo di ricerca “molto vasto” caratterizzato da numerosi algoritmi e software orientati alla risoluzione di numerosi problemi ...

... tra cui:

- Ricostruzione di immagini (image recontruction)
- Segmentazione di immagini (image segmentation)
- Compressione di immagini (image coding)
- Analisi di immagini (image analysis)
- Riconoscimento di forme (pattern recognition)
- ...



A partire
dall'immagine
rumorosa
si vuole
ricostruire
l'immagine
originale



Immagine rumorosa

$$z_1 = Ku + \eta$$

?

Immagine originale

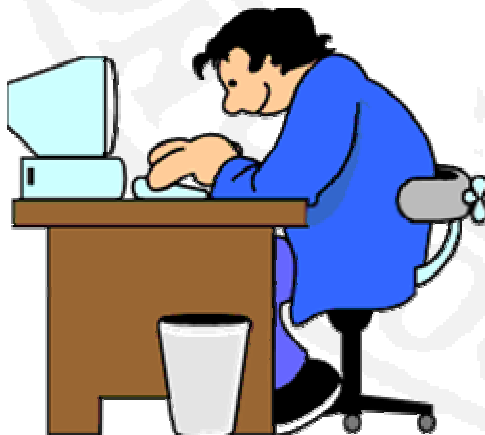
u

Come ottenere u da z_1 ?
(Problema Inverso)

Per risolvere il problema della ricostruzione di immagini sono richieste

Competenze matematico-numeriche:

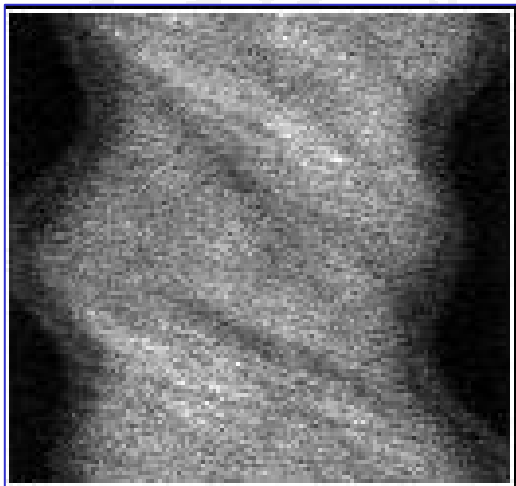
- Equazioni integrali;
- Equazioni a derivate parziali;
- Tecniche di approssimazione numerica;
- Metodi di risoluzione numerica;
- ...



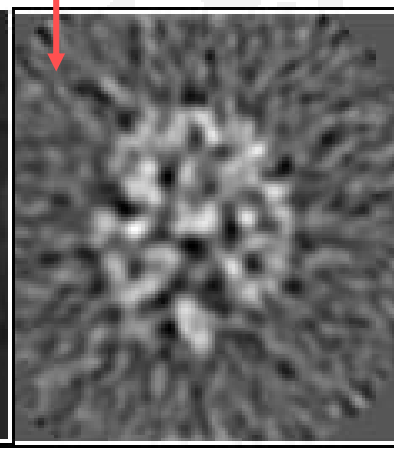
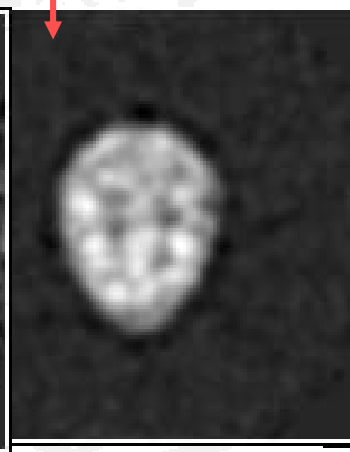
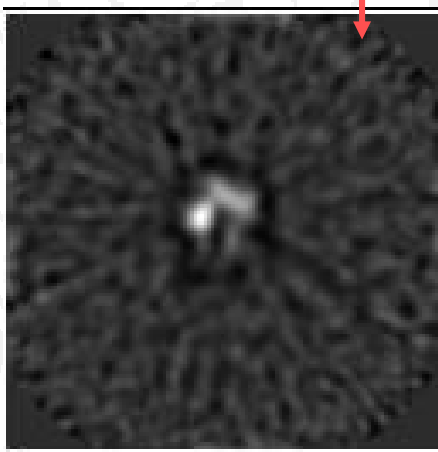
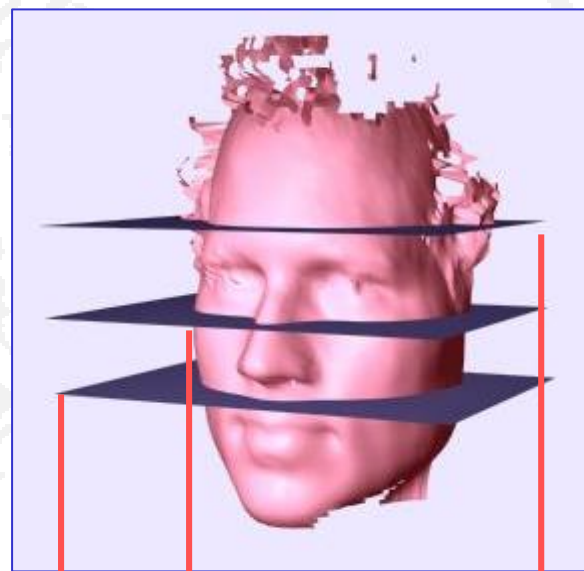
Competenze di informatica:

- Conoscenza di linguaggi di programmazione (**C**, **C++**, **Java**, ...);
- Sistemi operativi (**Unix**, **Windows**,...);
- Algoritmi e strutture dati
- ...

Un altro esempio di applicazione:
Ricostruzioni di immagini mediche
(SPECT Tomografie ad emissione di singolo fotone)



Dati acquisiti
dall'apparecchiatura
medica.



Sezioni ricostruite

**Formulazione del
Problema Reale**

P



**Formulazione del
Modello Matematico**

M(P)



**Approssimazione di M(P)
mediante Metodo Numerico**

M_h(P)



**Sviluppo di un
Algoritmo**

A(P)



**Implementazione
dell'algoritmo in un ambiente
di calcolo: Software**

S(P)



**Procedimento di
Risoluzione
Computazionale**

Conclusioni

“Some mathematics becomes more important because technology requires it.

Some mathematics becomes less important because technology replaces it.

Some mathematics becomes possible because technology allows it.”

“Una parte della matematica diventa più importante perché la tecnologia la richiede.

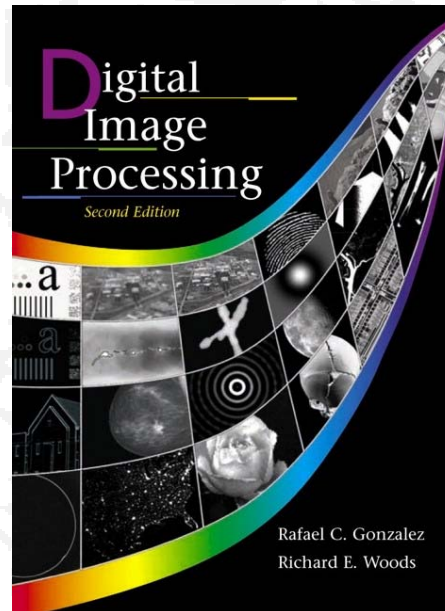
Una parte della matematica diventa meno importante perché la tecnologia la sostituisce.

Una parte della matematica diventa possibile perché la tecnologia lo consente. ”

(Bert Waits, co-fondatore Texas Instruments)

Libri di testo

Gonzalez and Woods, **Digital Image Processing Using Matlab**, [Prentice Hall](#) 2004.



A. Murli, **Matematica Numerica: Metodi, Algoritmi e Software**, Liguori ed.

C. Moler, **Numerical Computing with Matlab**, xxxxxx.

<http://www.mathworks.com/moler/>



UNIVERSITÀ DEGLI STUDI DI
POLI FEDERICO II

Seminario progetto Logim@t

Algebra Lineare Numerica
ed
Elaborazione delle Immagini

PARTE II
(Spunti per la didattica)



**Formulare problemi “nuovi”
al fine di attrarre l’attenzione degli studenti**

Di quanto spazio necessito?

Se f è una immagine monocromatica tipica 512×512



Ricordando che ogni intero necessita di
8 bit per essere memorizzata



f occupa $(512 * 512 * 8)$

$\approx 2 \text{ Mbit} \approx 256 \text{ kbyte}$ di memoria

(1 bit sono circa 8 byte)

(informazioni sufficienti!)



Scale di misura...

YB	Yottabyte	1000 ZB	tutti gli atomi di un essere umano ci vorrebbero 5900 yobibyte
ZB	Zettabyte	1000 EB	
EB	Exabyte	1000 Pb	Tutta la conoscenza umana è di circa 12 exbibyte
PB	Petabyte	1000 Tb	Capacità di enormi datacenter
TB	Terabyte	1000 Gb	200 film in DVD
GB	Gigabyte	1000 Mb	2 ore di film Divx
MB	Megabyte	1000 Kb	1 foto al cellulare
kB	Kilobyte	1000 byte	1 pagina corta di testo
-	Byte	8 bit in successione	1 carattere ASCII
-	Bit	può valere 1 o 0	



OGGI

Spunti per la didattica...

Problema 1:

Quanto occupa una immagine 64×64 utilizzata in applicazioni tipiche del web (icone)?

Problema 2:

Quanto occupa una immagine 640×480 a colori?

Problema 3:

Quale è la dimensione maggiore di una immagine non scalata che uno schermo a $19''$ può visualizzare?

Problema 4:

Un filmato contiene 25 immagini al secondo, si supponga che sia a scala di grigi ed ogni immagine sia 256×256 . Un video di 30 sec quanto occupa?

Spunti per la didattica...

Problema 5:

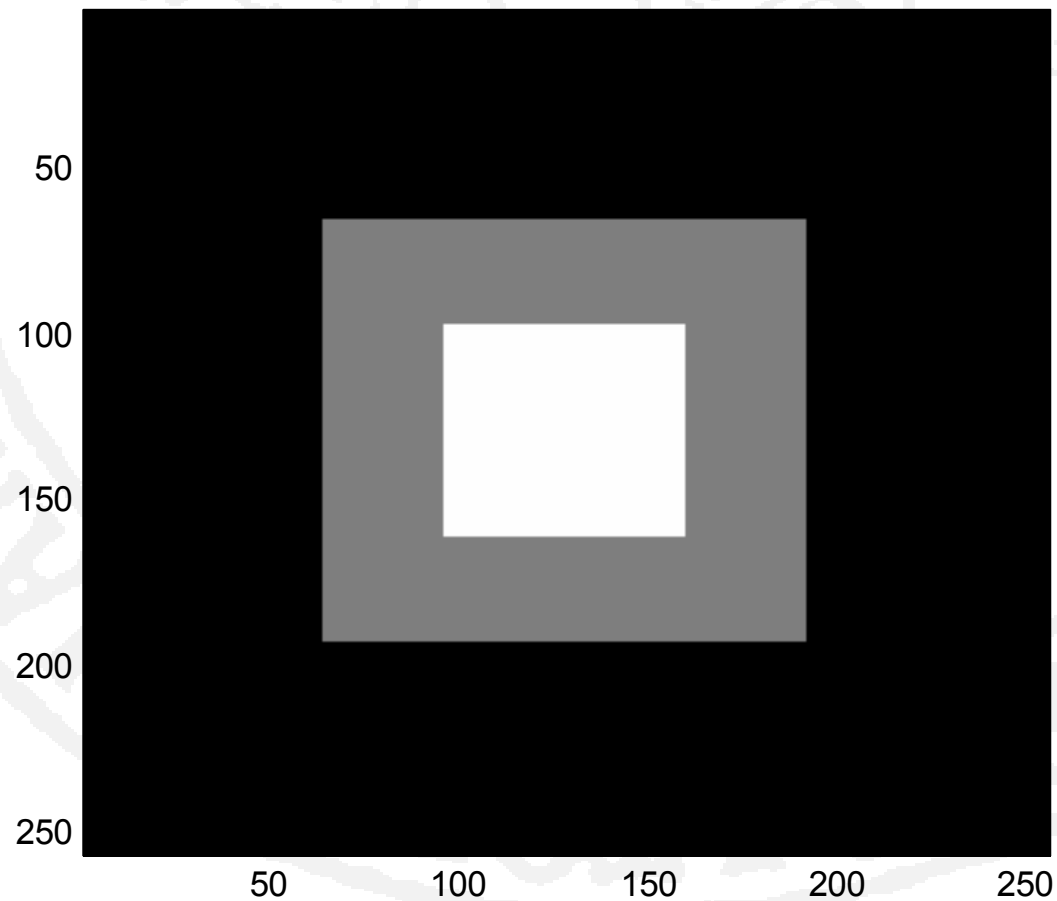
*Un video di **5000Mb** quanti secondi dura? In termini di GB quanto occupa?*

❑ **Contenuti:** Il concetto di numero, operazioni di aritmetiche, relazioni logico-matematiche dati soluzione, scale di misura...

❑ **Obiettivi specifici:** Individuare nei programmi ministeriali le nozioni di matematica utili alla risoluzione dei problemi posti.

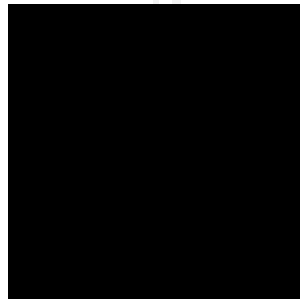
Costruzione di immagini sintetiche

Supponiamo di volere colorare delle zone “opportune” da colpire nel gioco della battaglia navale ... al fine di ottenere



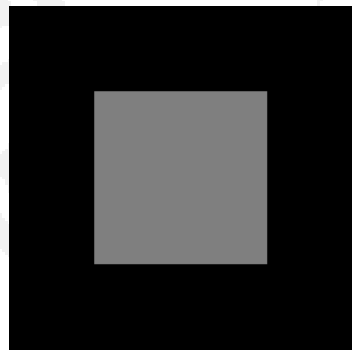
Costruzione come procedere...

- 1) Creare una matrice quadrata di 256x256 elementi con tutti 0



```
>> A=zeros(256);
```

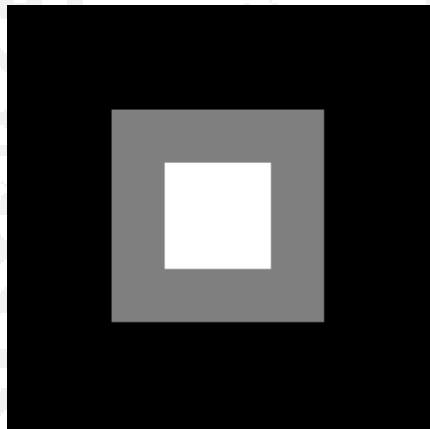
- 2) Creare una matrice quadrata di 128 x 128 elementi con tutti 127



```
>> B=127*ones(128);
```

Costruzione come procedere...

3) Creare una matrice quadrata di 64x64 elementi con tutti 255

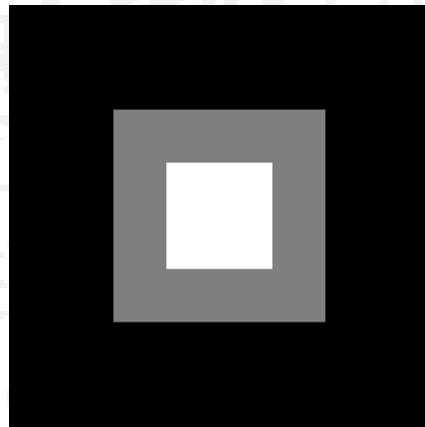


```
>> C=255*ones(64);
```

Soluzione: individuare le coordinate del centro e posizionare “opportunamente” i quadrati.

Soluzione con Matlab

```
>> A=zeros(256);  
>> A(64:191,64:191)=127*ones(128);  
>> A(96:159,96:159)=255*ones(64);  
>> imshow(A,[0 255])
```



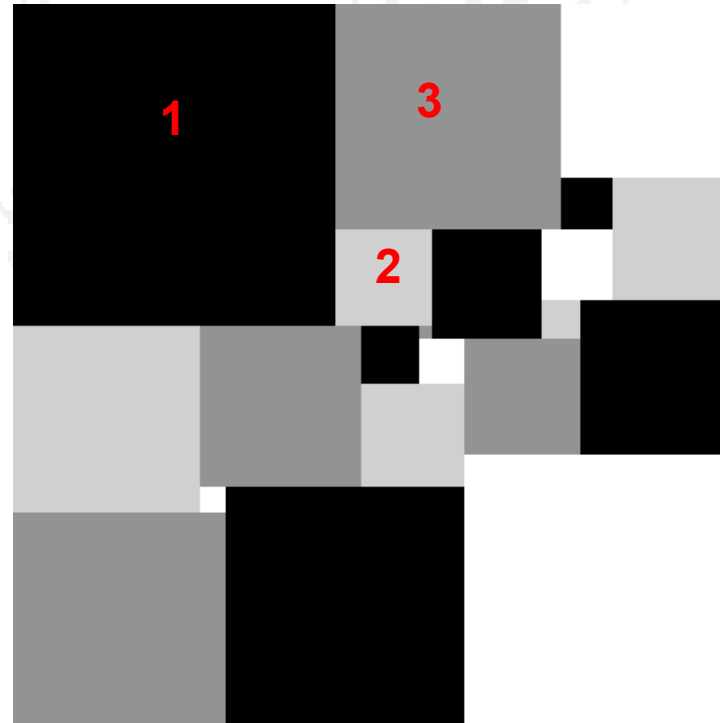
Da fare in classe

Utilizzare un foglio a quadretti per ripetere l'esperienza

- 1) Creare una matrice quadrata di 47x47 elementi con tutti 0 (Nero)
- 2) Creare una matrice quadrata di 27x27 elementi con tutti 127 (Grigio)
- 3) Creare una matrice quadrata di 15x15 elementi con tutti 255 (Bianco)



La battaglia Navale

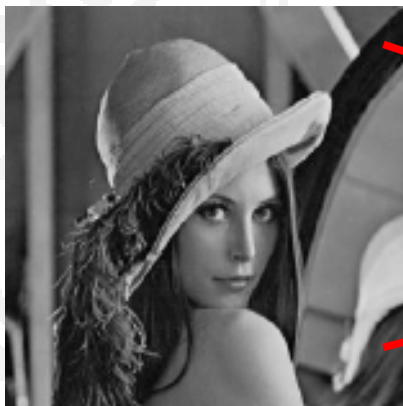


Problema:

Ricostruire l'immagine a partire dalle coordinate del centro di ogni quadrato.

- 1) Centro (50,50) lato 100
- 2) Centro (110,110) lato 20
- 3) Centro (130,130) lato 40

Problema



Come è possibile estrarre delle **regioni di Interesse (ROI)**
da una immagine

?

Approccio 1)



Scrivere algoritmi per

Estrarre gli indici di riga e colonna a cui appartiene

la zona interessata



```
for i=n1 to n2
  for j=m1 to m2
    estr(i,j)=im(i,j)
  end
end
```

Approccio 1) (cont)



Vantaggi:

Metodo intuitivo;

Manipolazione diretta dell'immagine

....

Svantaggi:

Algoritmi "ad hoc" per estrarre regioni non rettangolari;

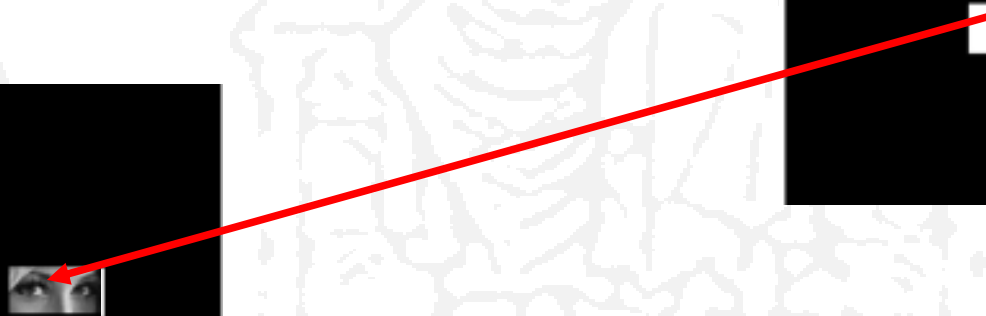
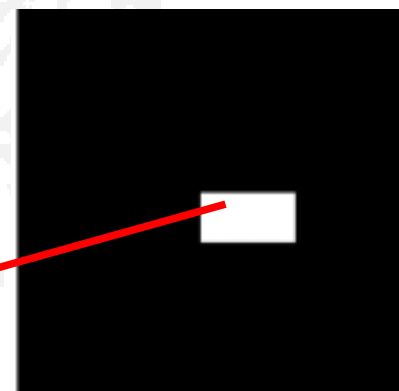
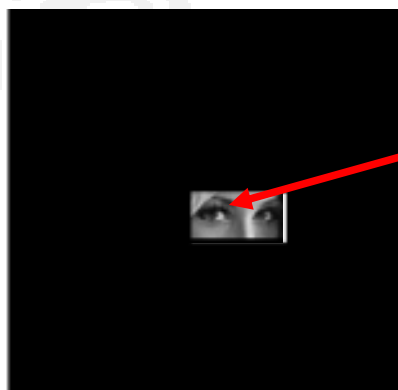
Elevata complessità computazionale nel caso di estrazioni multiple

o immagini di elevate dimensioni....

Approccio 2)



Predisporre una immagine (**Maschera**) che contiene l'informazione da estrarre ed imporre che nella zona di interesse ci sia cio' che vogliamo

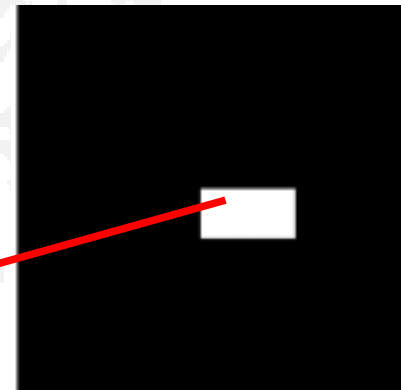
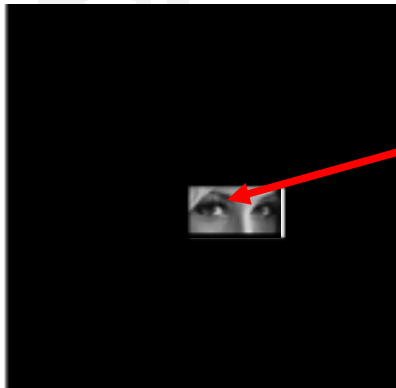


Approccio 2)



Come Realizzare questa idea

?



Utilizzare operazioni bit a bit



Infatti se un pixel vale 10010011 allora:

10010011 **AND** 11111111 = 10010011

10010011 **AND** 00000000 = 00000000

10010011 **OR** 11111111 = 11111111

10010011 **OR** 00000000 = 10010011

Pixel Estratto

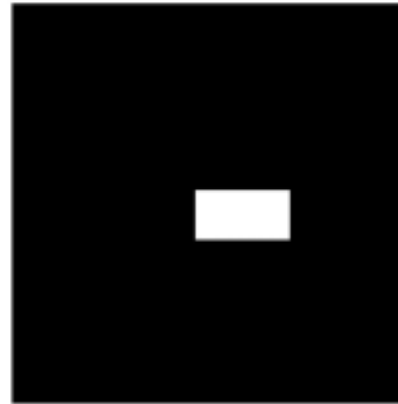
A	B	$A \wedge B$
F	F	F
F	V	F
V	F	F
V	V	V

A	B	$A \vee B$
F	F	F
F	V	V
V	F	V
V	V	V

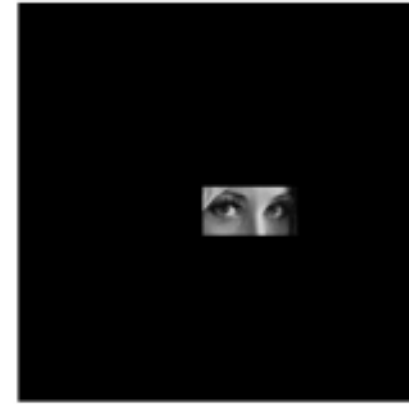
Utilizzare operazioni bit a bit



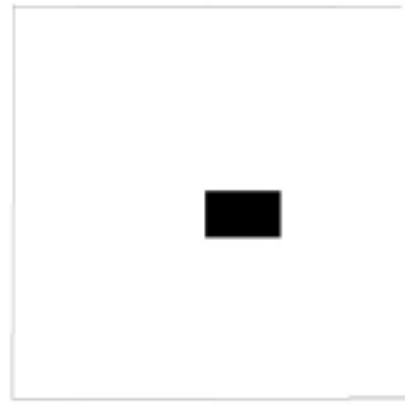
AND



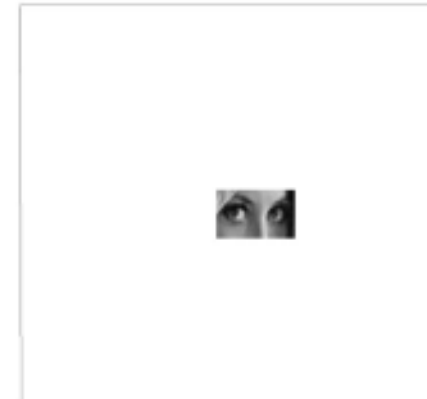
=



OR



=



Approccio 2)



Vantaggi:

Metodo intuitivo;

Elevata efficienza degli algoritmi basati su operazioni logiche

Utilizzo di linguaggi a basso livello

Estrazione di contorni non rettangolari

....

Svantaggi:

Creazione di una immagine sintetica ad hoc (la maschera)

Maggiore occupazione di memoria

Matrici Sparse (un esempio di Compressione)

i metodi iterativi non modificano la matrice dei coefficienti del sistema



utilizzo di **algoritmi opportuni per memorizzare matrici sparse**
(solo i coefficienti non nulli)



riduzione della complessità di spazio

ESEMPIO 13

$$\begin{pmatrix} 7 & 0 & -3 & 0 & 1 & 0 \\ 2 & 8 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ -3 & 0 & 0 & 5 & 0 & 0 \\ 0 & -1 & 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & -2 & 0 & 6 \end{pmatrix}$$



schema dei tre vettori

$$R = (7, -3, -1, 2, 8, 1, -3, 5, -1, 4, -2, 6);$$

elementi non nulli di A riga per riga

$$IC = (1, 3, 5, 1, 2, 3, 1, 4, 2, 5, 4, 6);$$

$IC(k)$ è l'indice di colonna in A di $R(k)$

$$J = (1, 4, 6, 7, 9, 11, 13);$$

$J(i)$ è la posizione in R del primo elemento non nullo della i -ma Riga di A (l'ultimo elemento di J è uguale al numero di elementi non nulli di A +1)

Complessità di spazio - $S(n)$

A matrice $n \times n$

m numero di elementi non nulli di A

1. array bidimensionale

$$S_1(n) = n^2$$

2. schema dei tre vettori

$$S_2(n) = m + m + (n+1)$$

[dim. di R + dim. di IC + dim. di J]

Esempio:

A matrice 100×100 , $m=400$

$$S_1(n)=10000, \quad S_2(n)=901$$

se A è sparsa e n è grande si ha
un notevole risparmio di memoria.

**Matlab
e
Elaborazione delle Immagini**

Leggere, visualizzare e salvare un immagine

Leggere

```
>> f=imread('escher.jpg');  
>> figure,imshow(f);  
>> size(f);  
>> whos
```

Scrivere

```
>> imwrite(g, 'escher1.jpg', 'quality', q);  
>> figure,imshow(g);  
>> size(g);
```

Salvare una immagine

Formati di immagini supportati dalle funzioni:

imread

imwrite

Format Name	Description	Recognized Extensions
TIFF	Tagged Image File Format	.tif, .tiff
JPEG	Joint Photographic Experts Group	.jpg, .jpeg
GIF	Graphics Interchange Format [†]	.gif
BMP	Windows Bitmap	.bmp
PNG	Portable Network Graphics	.png
XWD	X Window Dump	.xwd

[†] GIF is supported by `imread`, but not by `imwrite`.

Estrarre informazioni da una immagine

```
>> iminfo escher1.jpg
```

Questo comando restituisce informazioni relative a:

dimensione, formato dell'immagine, data di creazione,...

Calcoliamo il rapporto di compressione

```
>> K=iminfo('esher.jpg');  
>> im_byte=K.Width*K.Height*K.BitDepth/8;  
>> comp_byte=K.FileSize;  
>> rate_compr=im_byte/comp_byte
```


Tipo di dato di un immagine

```
>> f=imread('escher.jpg');  
>> figure,imshow(f);  
>> size(f);  
>> whos
```

L'immagine f e' una matrice di dati **uint8**, ovvero intero unsigned ad 8 bit.

Name	Description
double	Double-precision, floating-point numbers in the approximate range -10^{308} to 10^{308} (8 bytes per element).
uint8	Unsigned 8-bit integers in the range [0, 255] (1 byte per element).
uint16	Unsigned 16-bit integers in the range [0, 65535] (2 bytes per element).
uint32	Unsigned 32-bit integers in the range [0, 4294967295] (4 bytes per element).
int8	Signed 8-bit integers in the range [-128, 127] (1 byte per element).
int16	Signed 16-bit integers in the range [-32768, 32767] (2 bytes per element).
int32	Signed 32-bit integers in the range [-2147483648, 2147483647] (4 bytes per element).
single	Single-precision floating-point numbers with values in the approximate range -10^{38} to 10^{38} (4 bytes per element).
char	Characters (2 bytes per element).
logical	Values are 0 or 1 (1 byte per element).

Operazioni consentite su dati unit8

Operator	Name	MATLAB Function	Comments and Examples
+	Array and matrix addition	plus(A, B)	$a + b$, $A + B$, or $a + A$.
-	Array and matrix subtraction	minus(A, B)	$a - b$, $A - B$, $A - a$, or $a - A$.
.*	Array multiplication	times(A, B)	$C = A .* B$, $C(I, J) = A(I, J) * B(I, J)$.
*	Matrix multiplication	mtimes(A, B)	$A * B$, standard matrix multiplication, or $a * A$, multiplication of a scalar times all elements of A .
./	Array right division	rdivide(A, B)	$C = A ./ B$, $C(I, J) = A(I, J) / B(I, J)$.
.\	Array left division	ldivide(A, B)	$C = A .\ B$, $C(I, J) = B(I, J) / A(I, J)$.
/	Matrix right division	mrdivide(A, B)	A / B is roughly the same as $A * \text{inv}(B)$, depending on computational accuracy.
\	Matrix left division	mldivide(A, B)	$A \setminus B$ is roughly the same as $\text{inv}(A) * B$, depending on computational accuracy.
.^	Array power	power(A, B)	If $C = A.^B$, then $C(I, J) = A(I, J)^{B(I, J)}$.
^	Matrix power	mpower(A, B)	See online help for a discussion of this operator.
.'	Vector and matrix transpose	transpose(A)	$A.'$. Standard vector and matrix transpose.
'	Vector and matrix complex conjugate transpose	ctranspose(A)	A' . Standard vector and matrix conjugate transpose. When A is real $A.' = A'$.
+	Unary plus	uplus(A)	$+A$ is the same as $0 + A$.
-	Unary minus	uminus(A)	$-A$ is the same as $0 - A$ or $-1 * A$.
:	Colon		Discussed in Section 2.8.

Operazioni consentite su dati unit8 (cont)

Function	Description
imadd	Adds two images; or adds a constant to an image.
imsubtract	Subtracts two images; or subtracts a constant from an image.
immultiply	Multiplies two images, where the multiplication is carried out between pairs of corresponding image elements; or multiplies a constant times an image.
imdivide	Divides two images, where the division is carried out between pairs of corresponding image elements; or divides an image by a constant.
imabsdiff	Computes the absolute difference between two images.
imcomplement	Complements an image. See Section 3.2.1.
imlincomb	Computes a linear combination of two or more images. See Section 5.3.1 for an example.

Conversione di tipi

```
>> g=im2double(f);  
>> whos
```

E' possibile convertire una immagine **unit8** a double

Name	Converts Input to:	Valid Input Image Data Classes
im2uint8	uint8	logical, uint8, uint16, and double
im2uint16	uint16	logical, uint8, uint16, and double
mat2gray	double (in range [0, 1])	double
im2double	double	logical, uint8, uint16, and double
im2bw	logical	uint8, uint16, and double

Un piccolo script...

Realizziamo un piccolo script per estrarre una parte di una immagine **estrai.m**

```
clear % Libera la memoria.
% carica immagine
f=imread('esher.jpg');
imshow(f) % Visualizza f
I=im2double(f); % Converte a double.
whos % verifico l'occupazione di memoria
%estraggo parte dell'immagine
for i=1:200
    for j=1:200
        Ired(i,j)=I(i,j);
    end
end
whos % Verifico l'occupazione di memoria
imshow(Ired) % Visualizzo
```

Seminario progetto Logim@t

Algebra Lineare Numerica
ed
Elaborazione delle Immagini

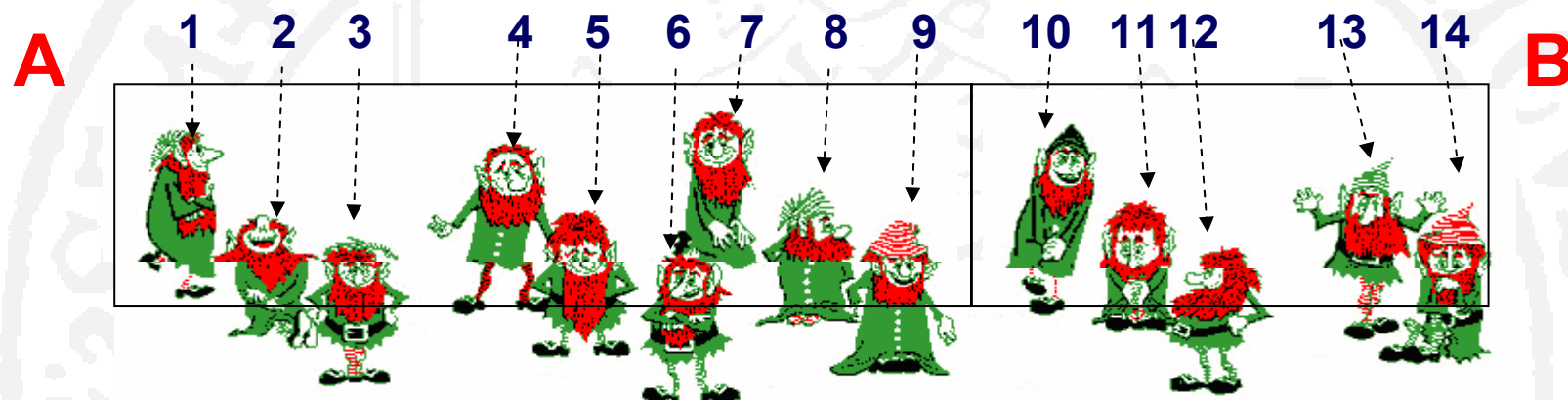
PARTE III
(Matematica Ricreativa)



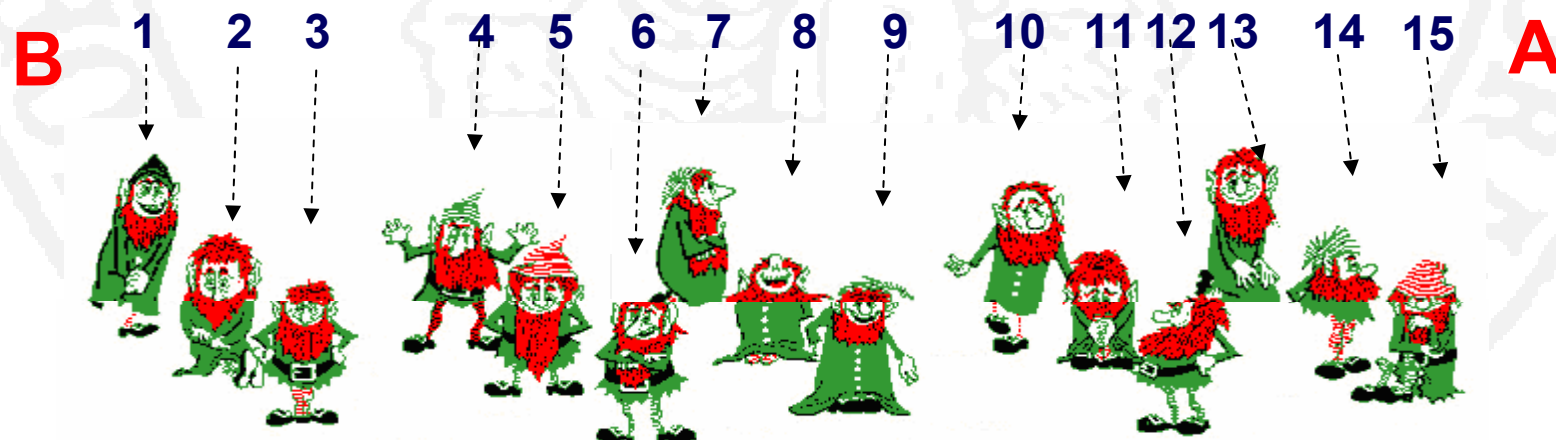
IL GIOCO DELLE SPARIZIONI



Gli gnomi dispettosi...

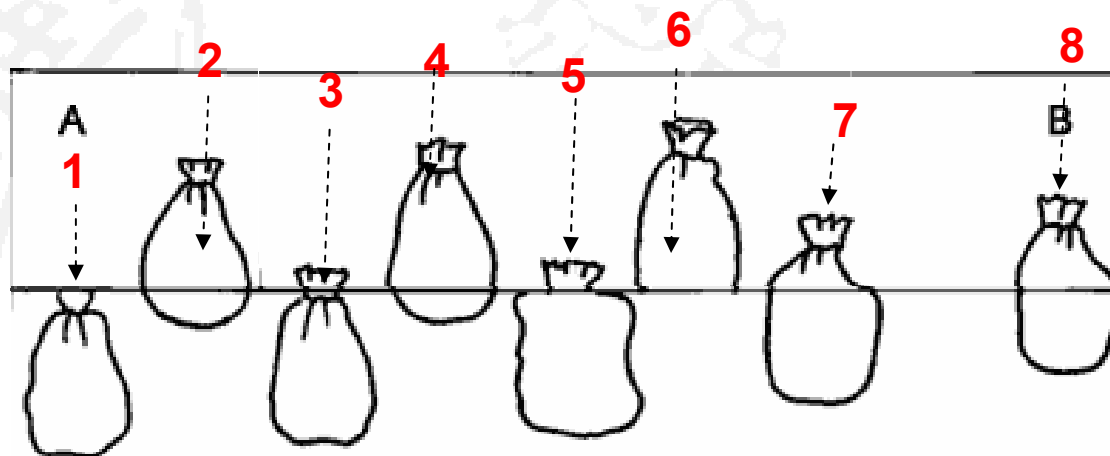


Dividiamo la figura in riquadri ed Invertiamo il riquadro A con quello B

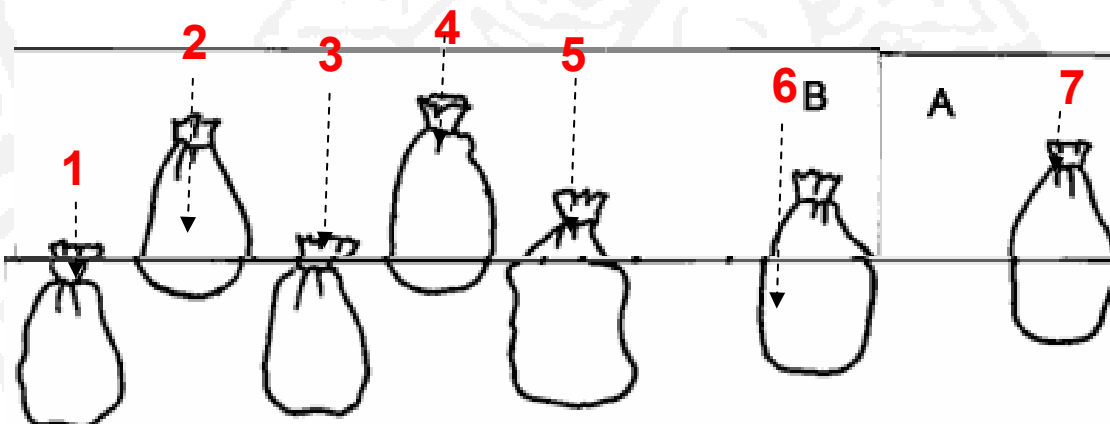


E' Comparso uno GNOMO!

Sacchi di monete...

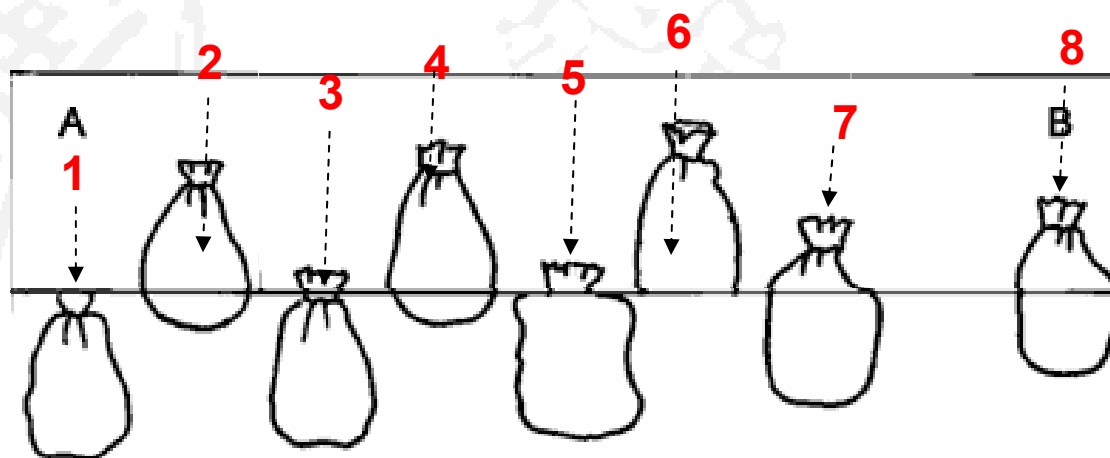


Dividiamo la figura in riquadri ed Invertiamo il riquadro A con quello B

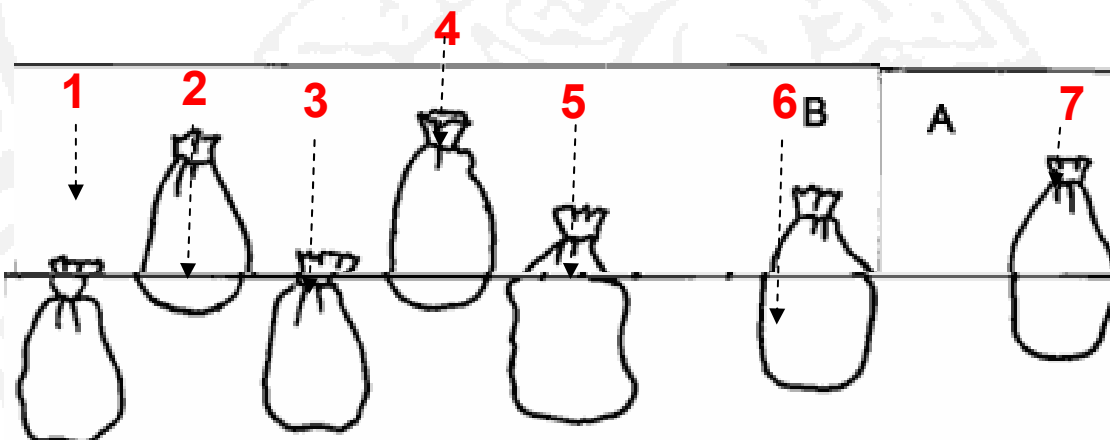


E' Scomparso un SACHETTO!

Sacchi di monete...



Se ogni sacchetto conteneva 100 euro nel primo caso ne abbiamo 800



Nel secondo 700 euro, ovvero sono spariti 100 EURO!



Come hanno risposto gli Studenti

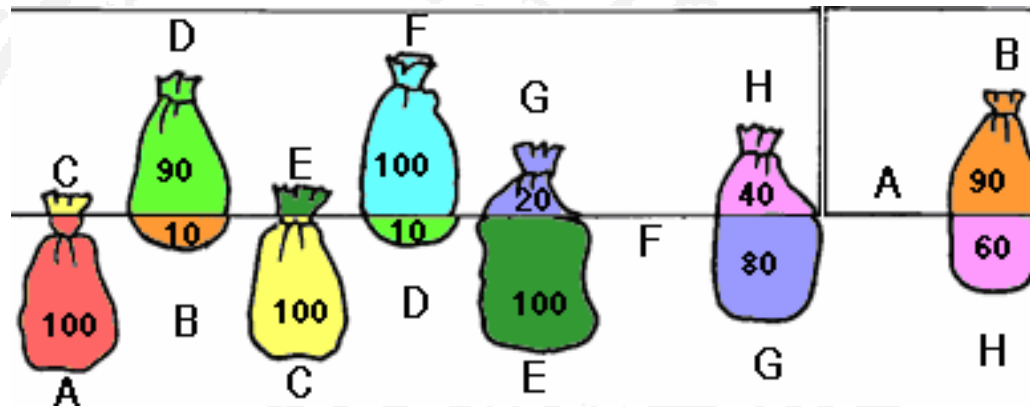
?

Ivan - Risposta: **Dipende dalla figura !**

GiannMarco - Risposta: **Alcuni sacchetti sono disegnati male !**

Rossella - Risposta: **E' un mistero !**

Percorso per arrivare alla soluzione...



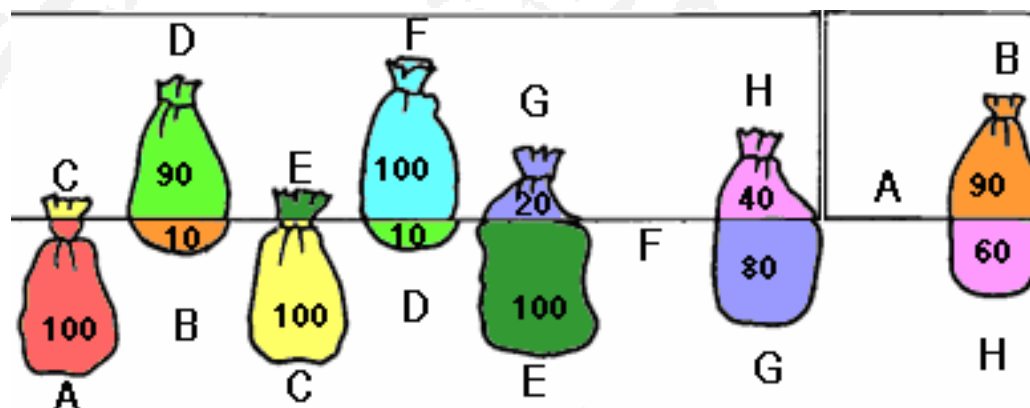
- 1) Facciamo **colorare** i **sacchetti** agli studenti
- 2) Facciamo **enumerare** con numeri e lettere i sacchetti
- 3) Inseriamo nei sacchetti **le monete**

Noteranno che nei sacchetti

Sono presenti sempre

800 EURO ma distribuiti diversamente !

Soluzione matematica al gioco



*Esistono trasformazioni piane che
non conservano la forma ma la quantità*

Ovvero

*Le alcune figure (nel caso un sacchetto) si distribuiscono
in maniera uniforme...*

Alcuni dei giochi proposti



Crittografia: codici a sostituzione

L'Alfabeto Carbonaro

Si scambiano le coppie A/O, B/P, C/G, D/T, E/I, F/V, L/R, M/N, S/Z, lasciando invariate le H, K, J, Q, U, W, Y, X.

Nell'alfabeto carbonaro le lettere cifrate assomigliano a quelle originali e quindi permettono di tenere il cifrario a memoria facilmente.

Cifrario:

Chiario	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
cifrato	O	P	G	T	I	V	C	H	E	J	K	R	N	M	A	B	Q	L	Z	D	U	F	W	X	Y	S

Crittografia: Esempio 4

L'alfabeto carbonaro

Giufà compra 10 asini

Ceuvo' ofifo ganbloda teige ozeme.

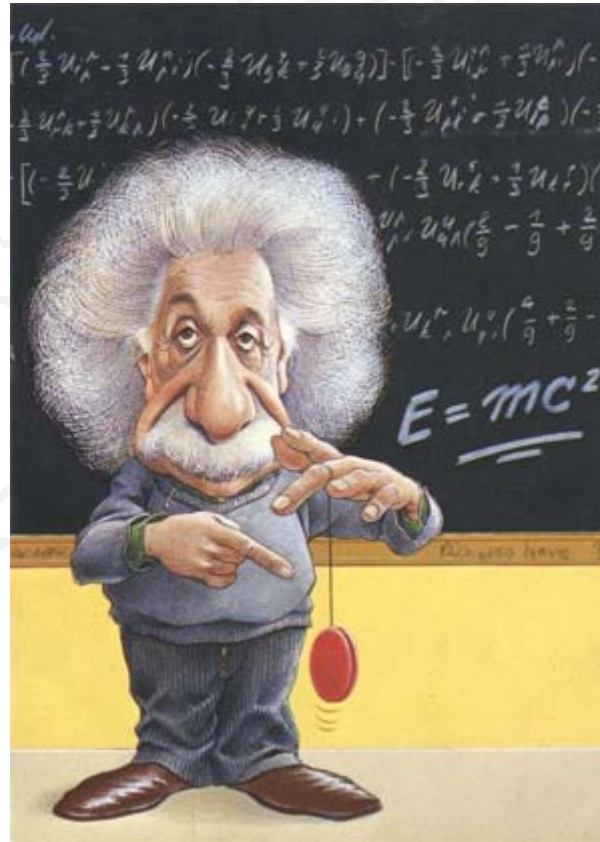
Namda' zablo o uma i ze offea' filza gozo dladdilirromtaem
gato or plomga.

Taba um ba' te dinba tigezi te gamdolre no izzimtaze
tenimdegoda tirro pizdeo zu gue namdofo, cre lezurdolama
zara mafi ozeme. Zgizi torr'ozema bliagguboda i leganemgea'
ro gamdo: iloma teige. Lezore' zurr'ozema, dalma'
o gamdolre: mafi.

Taba ofil lebiduda beù fardi quizdo abiloseami, ze tezzi:

"I' nicrea omtoli o beite i cuotocmoli um ozema, beuddazda ghi
biltilmi uma bil zdoli zituda!"

I gaze ze emgonnema' teidla oe zuae ozeme, ollefomta em boizi
zdomga nalda.



Molti altri giochi di Logica