UN OCCHIO MATEMATICO SUL QUOTIDIANO

Lucia Della Croce – Giulia Maggi
Ada Pulvirenti
Dipartimento di Matematica -
Università di Pavia

Piano Lauree Scientifiche
Pavia – 12/13/14 Giugno 2012

Le immagini nella realtà

Tutti i giorni abbiamo a che fare con le immagini …

Acquisizione delle immagini

- L’acquisizione digitale di immagini è la fase di traduzione di una scena o di un oggetto fisico in formato digitale

- Semplici esempi di sistemi di acquisizione di immagini sono fotocamere, videocamere digitali, e scanner.

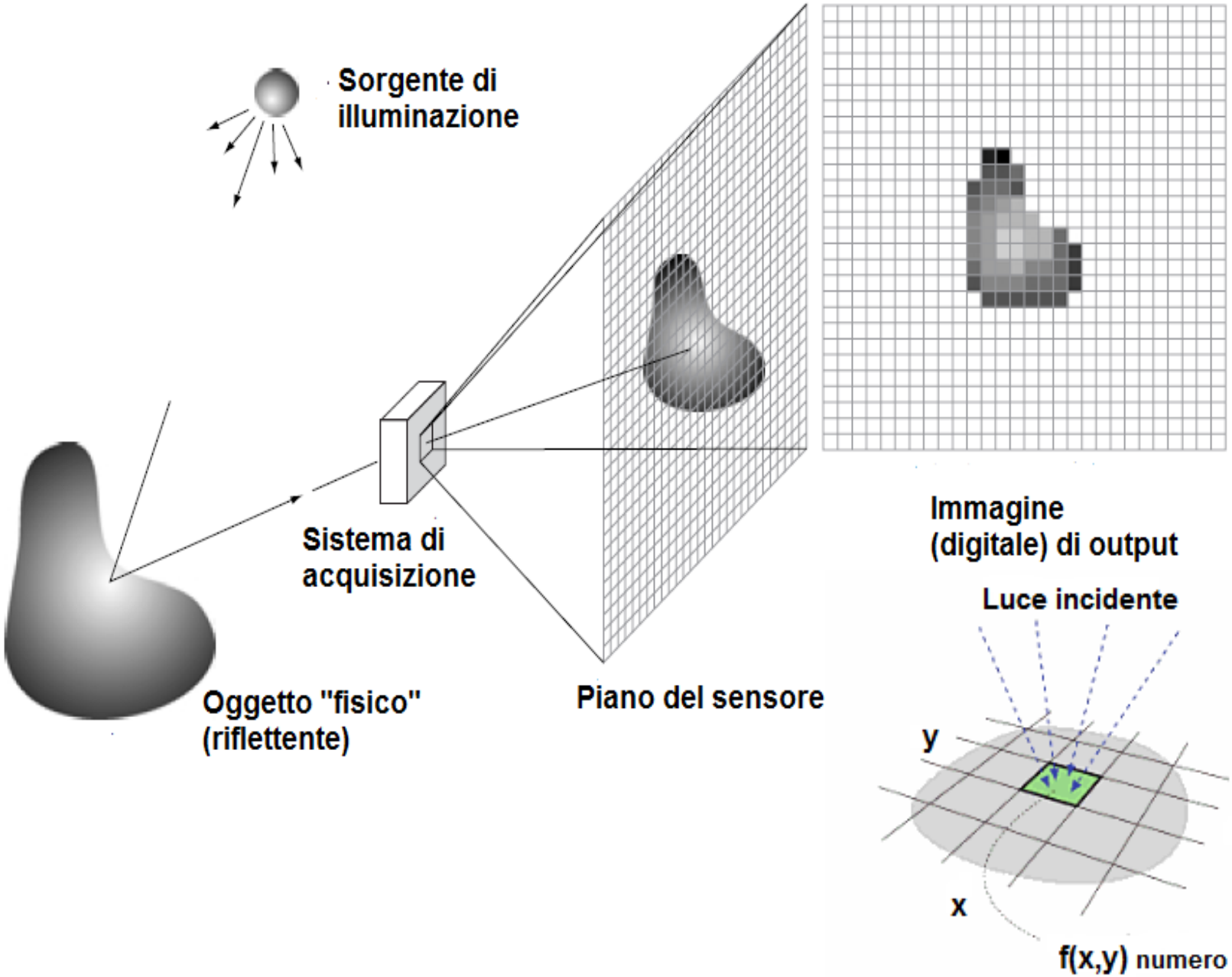
Acquisizione delle immagini nelle fotocamere digitali

- L’oggetto fisico è illuminato da una sorgente luminosa la cui energia emessa viene in parte assorbita e in parte riflessa dall’oggetto stesso.

- Parte della luce riflessa viene raccolta da uno o più ***sensori*** che registrano il valore della luce incidente.

- L’oggetto fisico è dunque convertito in una distribuzione spaziale continua di energia.

Acquisizione delle immagini

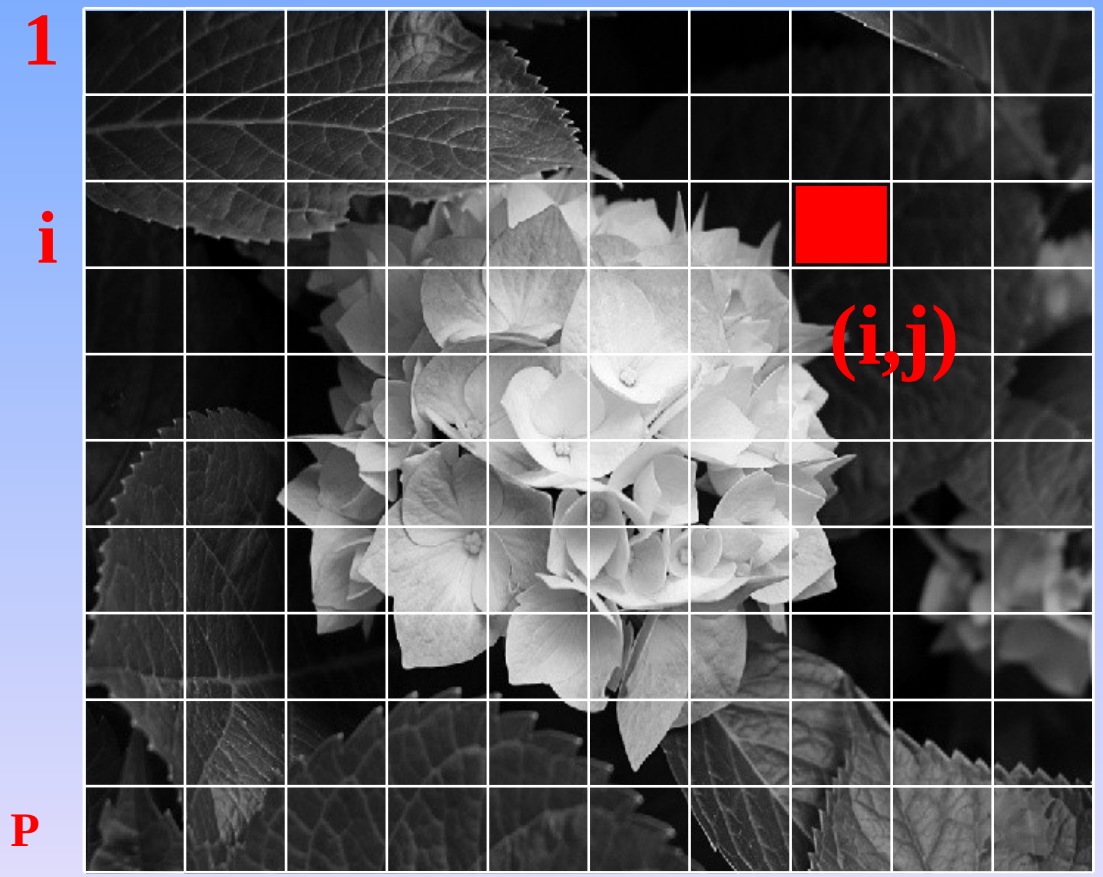


DIGITALIZZAZIONE DI IMMAGINI

Discretizzazione

- spaziale
- quantitativa

Discretizzazione spaziale

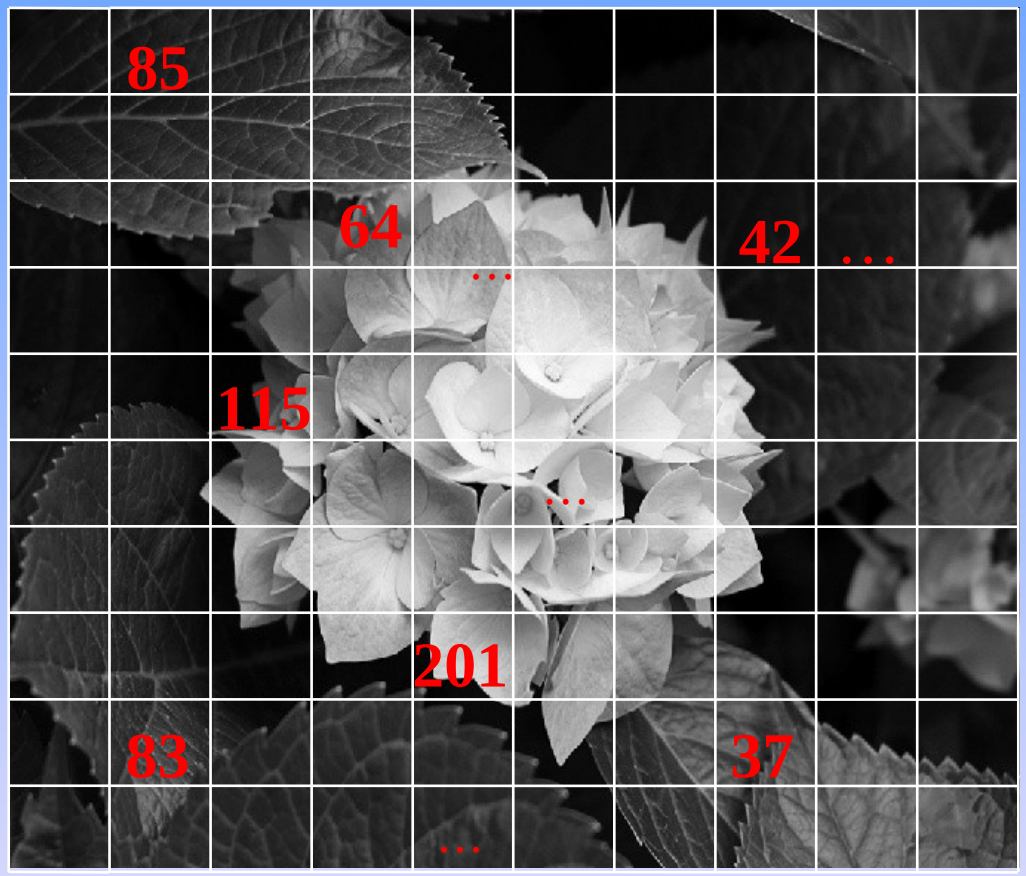


Matrice P x Q

$$F_{ij} = f(x_i, y_j)$$

Ogni elemento della matrice F è detto pixel

Discretizzazione quantitativa



I valori $f(x, y)$ vengono scelti in un insieme discreto di dimensione N (diadica, potenze del 2), che definisce il numero di livelli di grigi

DIGITALIZZAZIONE DI IMMAGINI

Passaggio dal continuo al discreto attraverso:

- **Campionamento**
  - Discretizzazione delle coordinate spaziali
- **Quantizzazione**
  - Discretizzazione dei valori di intensità
    - **Uniforme**
    - **Non - uniforme**

DIGITALIZZAZIONE DI IMMAGINI
Campionamento

- Campionare un'immagine continua significa leggerne i valori di luminosità in punti discreti.

- Il modo più semplice è la lettura dei valori secondo una griglia regolare con passi di discretizzazione detti intervalli di campionamento.

- Più la griglia di campionamento è fitta e migliore sarà il risultato (e maggiore sarà il numero di pixel generati)

Immagine 32×32

Immagine 64×64

Immagine 128×128

Immagine 256×256

CAMPIONAMENTO

Quantizzazione uniforme

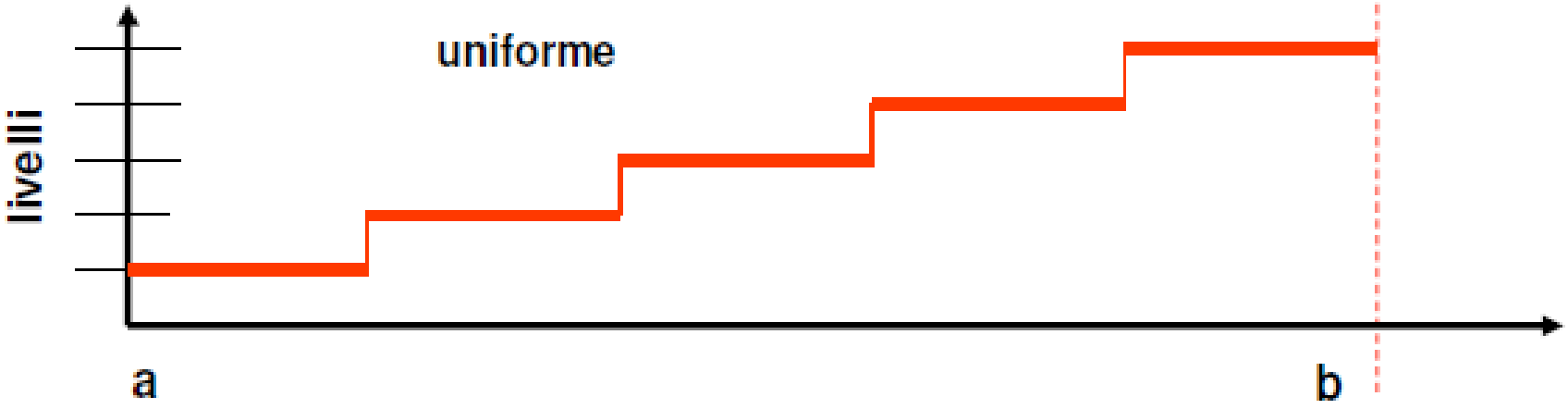


Immagine 256×256, 256 livelli di grigio

Immagine 256×256, 128 livelli di grigio

Immagine 256×256, 64 livelli di grigio

Immagine 256×256, 8 livelli di grigio

QUANTIZZAZIONE UNIFORME

Quantizzazione non uniforme

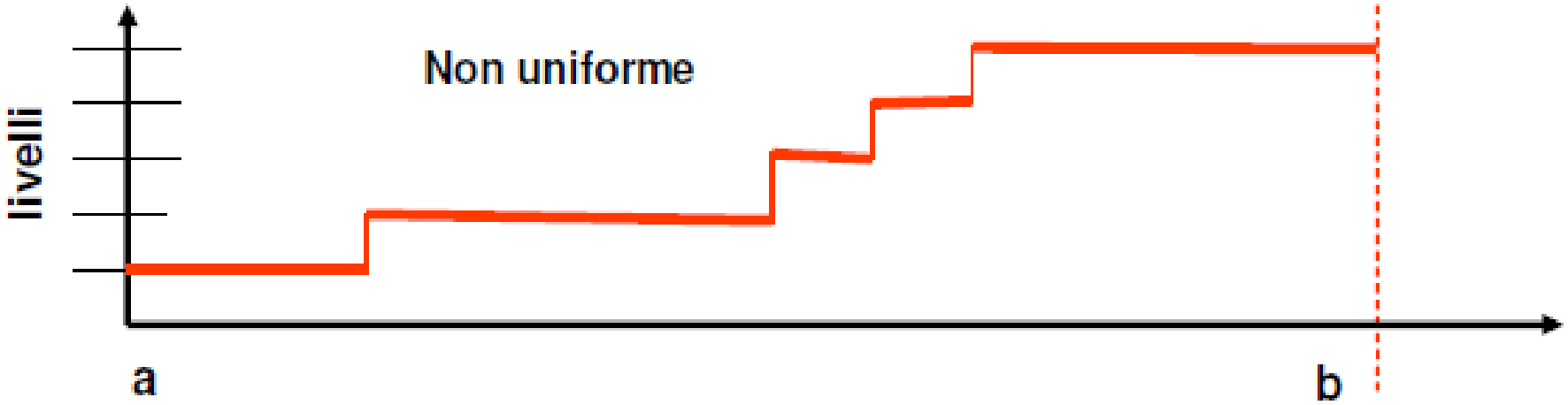


Immagine 256×256, 256 livelli di grigio

Immagine 256×256, 128 livelli di grigio

Immagine 256×256, 64 livelli di grigio

Immagine 256×256, 8 livelli di grigio

QUANTIZZAZIONE NON UNIFORME

TIPOLOGIE DI IMMAGINI

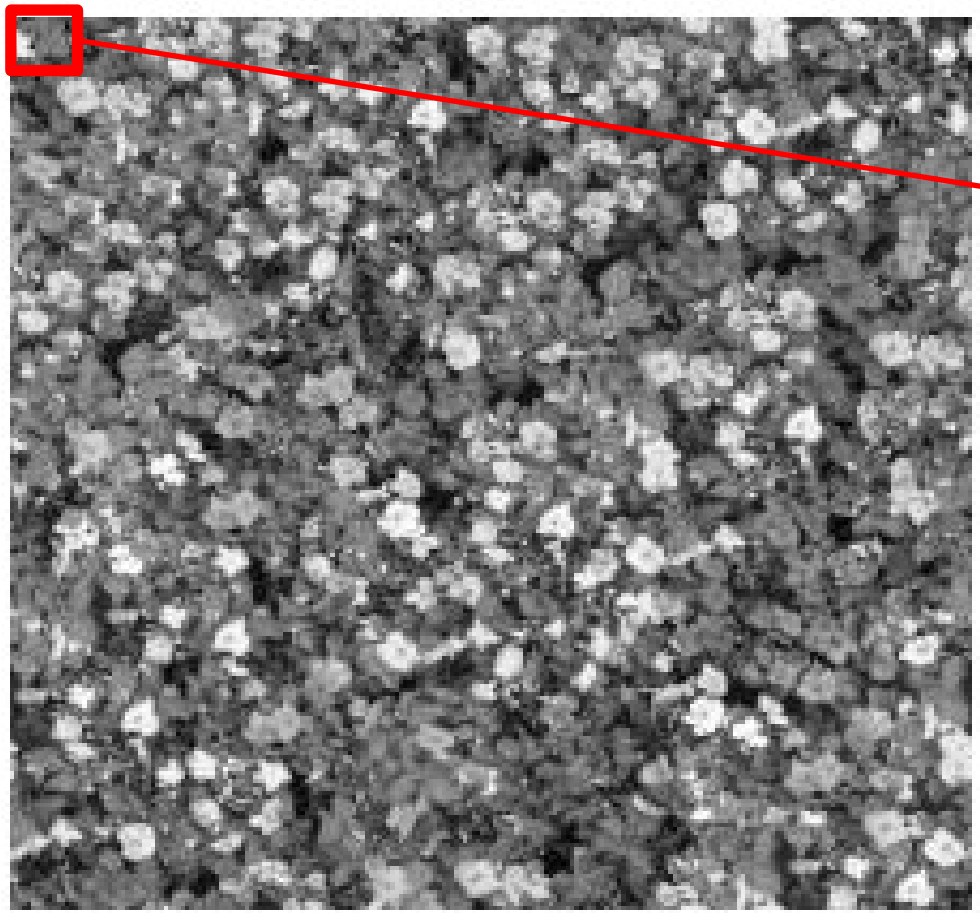
- Bianco/nero
- 1 bit per pixel
- Nella posizione (i,j) ci sarà o il valore 0 o il valore 1.

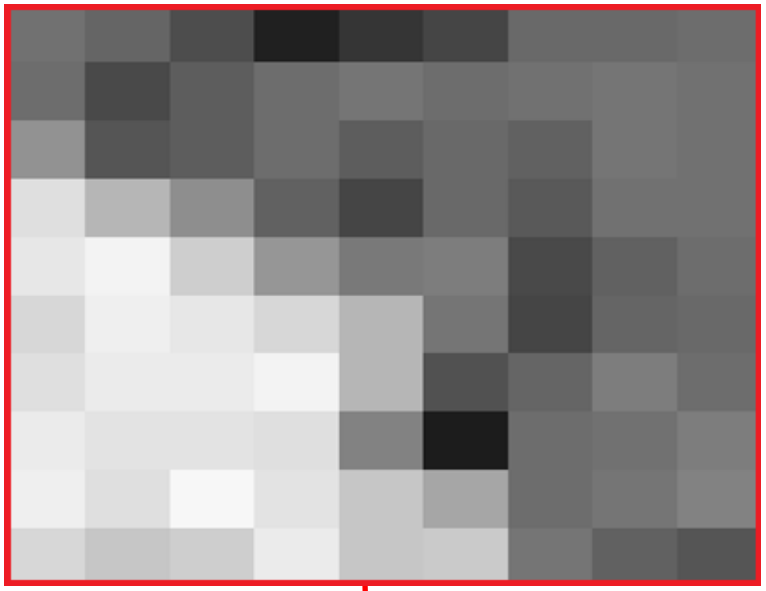
- A scala di grigi
- 8 bit per pixel
- Nella posizione (i,j) ci sarà un valore compreso tra 0 e 255.

- A colori (RGB)
- 8 bit per canale (24 bit)
- Nella posizione (i,j) ci sarà una terna (x,y,z) con x,y,z che assumono valori in [0,255].

Digitalizzazione di immagini

Immagine a scala di grigi

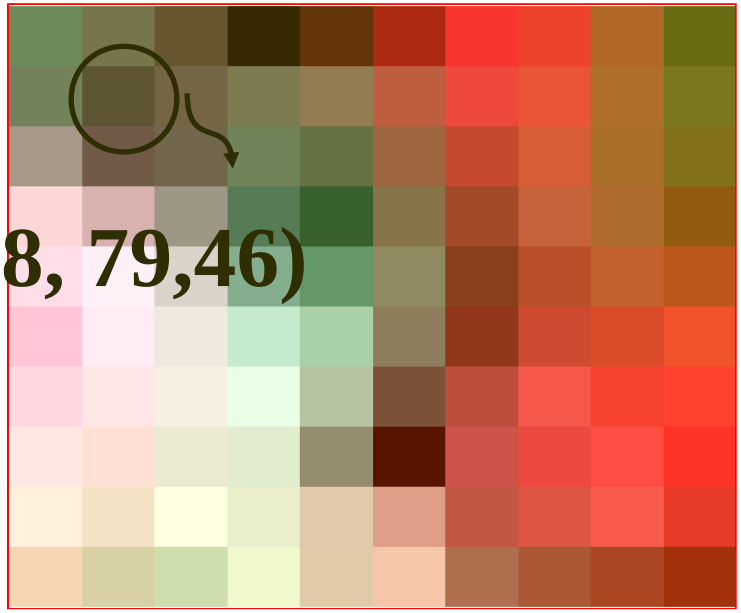




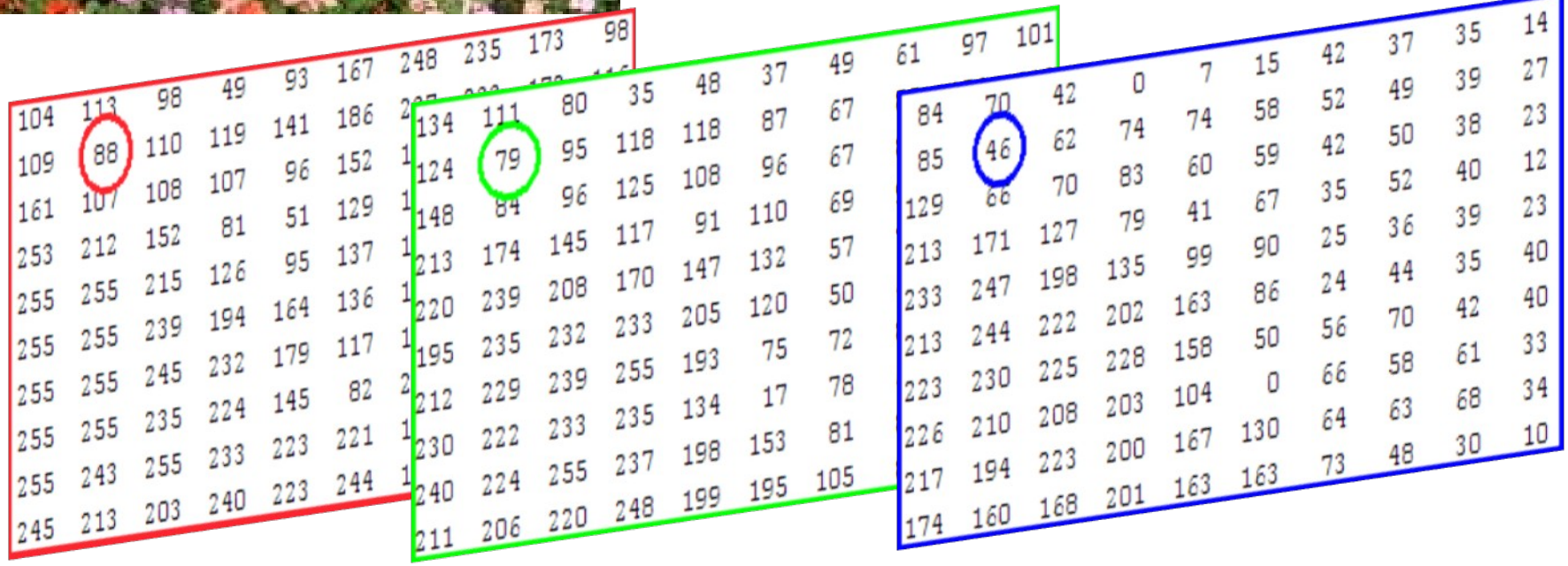
Valori dei pixel

119	107	81	35	57	73	108	110	113	90
115	78	96	113	120	113	116	122	117	105
150	89	97	115	99	109	102	121	116	103
225	185	145	102	73	111	92	119	116	94
233	247	209	153	126	129	76	102	115	106
217	244	233	218	188	121	74	106	109	122
228	237	239	246	185	85	104	130	113	118
239	231	231	228	134	34	114	116	128	107
244	226	251	232	202	171	112	120	132	103
217	203	209	240	202	206	121	103	90	74

Digitalizzazione di immagini

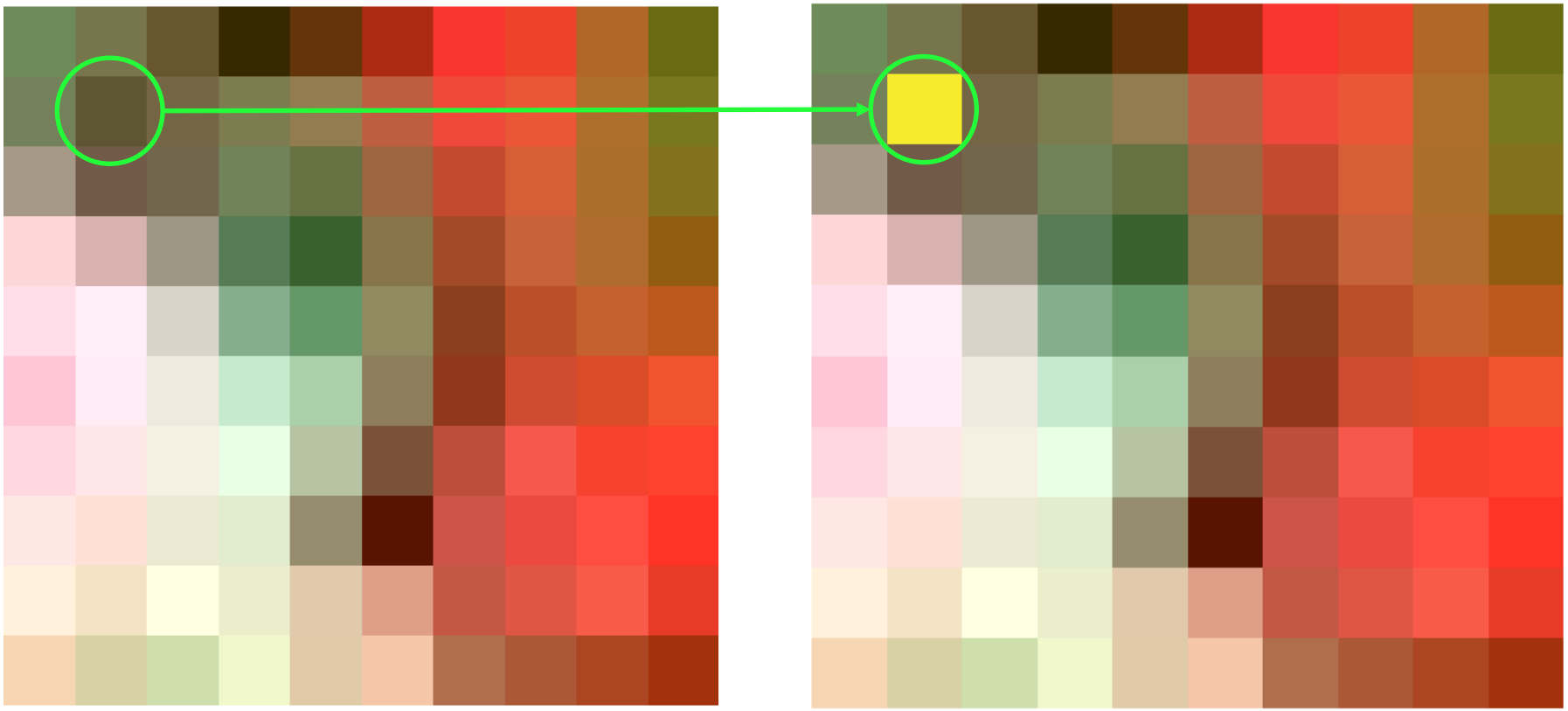


(88, 79,46)



Digitalizzazione di immagini

Modifica del valore del pixel in posizione (2,2) di ogni matrice



$$F_{2,2} = (88,79,46)$$

$$F'_{2,2} = (245,235,39)$$

Esempi di applicazioni: scurire e schiarire un’immagine

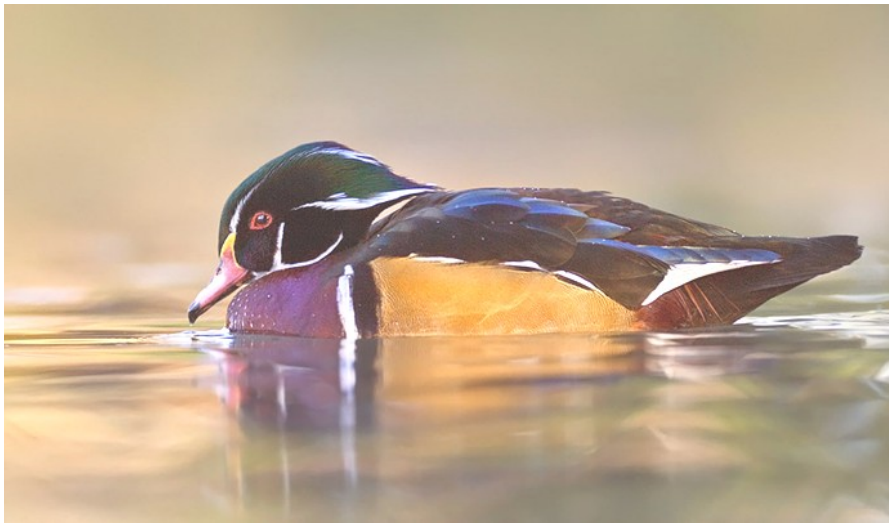
Sottrazione di una costante

Aggiunta di una costante

Scurire e schiarire un’immagine RGB

Sottrazione di una costante

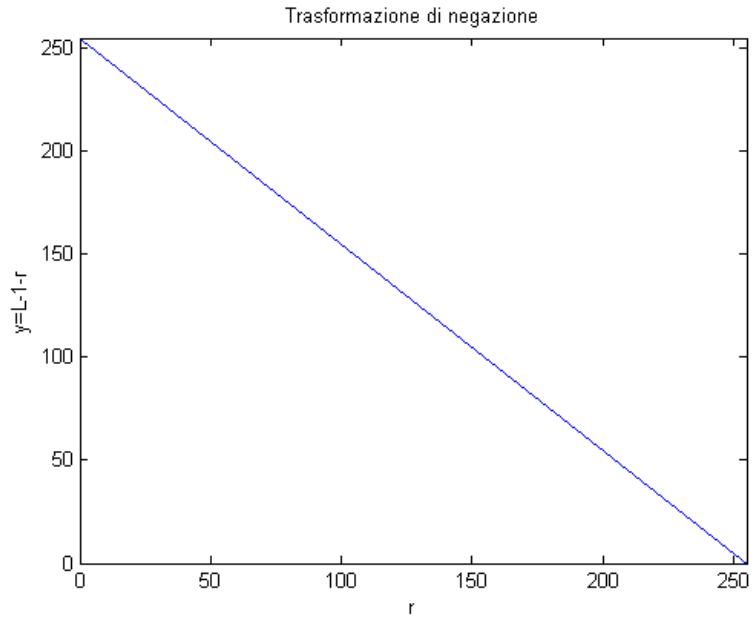
Aggiunta di una costante

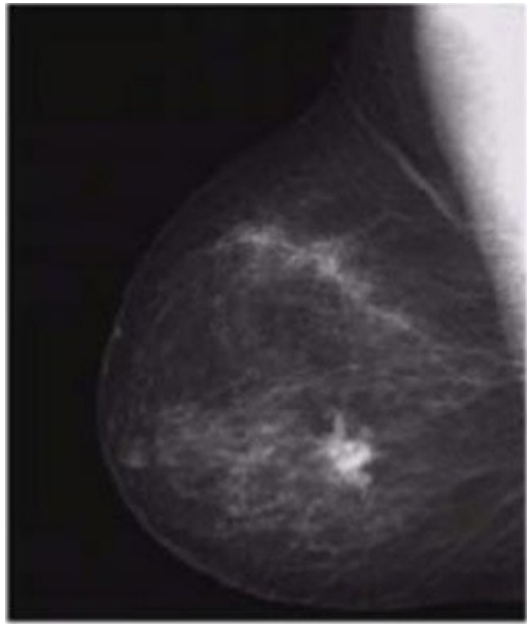


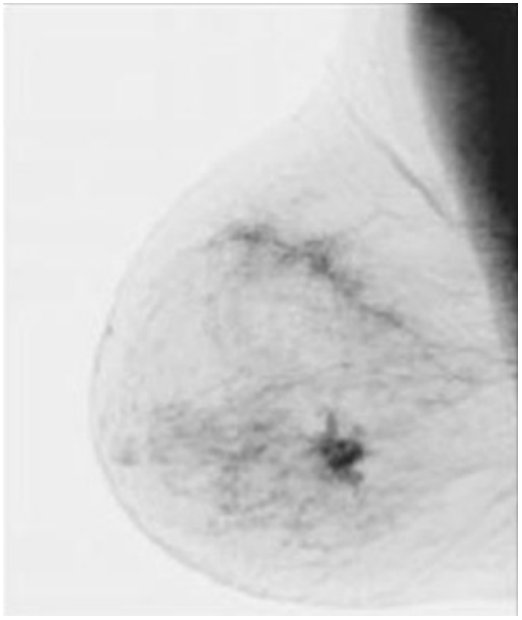
Complementare

$$L \rightarrow 255 - L$$

L livello di grigio





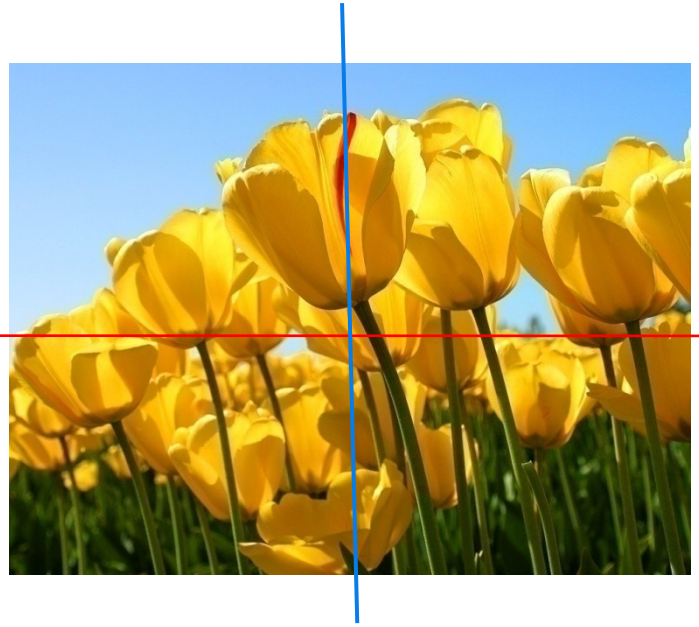


Simmetria assiale

$x_i \boxtimes x_i$

asse parallelo asse x

$y_i = Q - y_i$



Griglia

P x Q

asse parallelo

asse y

$x_i \boxtimes P - x_i$

$y_i = y_i$

Traslazione

$$\vec{v} = (h, k)$$

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} x + h \\ y + k \end{pmatrix}$$

Traslazione di vettore (50,50)

Rotazione

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

Rotazione di 45°

RUMORE E FILTRAGGIO

Rumore: è un insieme di errori per esempio introdotti nella fase di acquisizione dell'immagine, nei valori dei pixel che differiscono dai valori ideali

Filtraggio: filtrare un'immagine significa eseguire alcune "operazioni" tramite i "filtri" in modo da esaltare o attenuare alcune sue caratteristiche

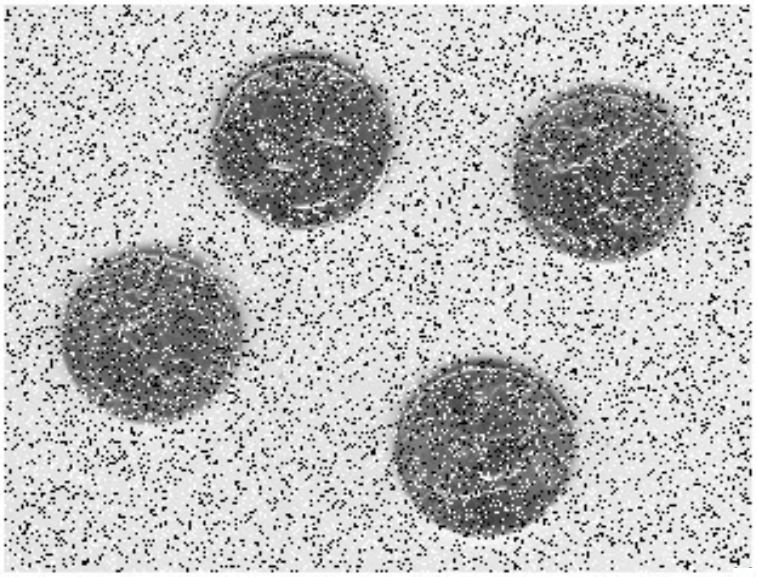
Cos’ è un filtro e come si applica

Il filtro rimpiazza il valore di un pixel con un nuovo valore di intensità

EFFETTI: la trasformazione mediante filtri dà origine ad una nuova immagine che può presentare miglioramenti o peggioramenti a seconda del filtro applicato.

Effetti del filtraggio

Miglioramento

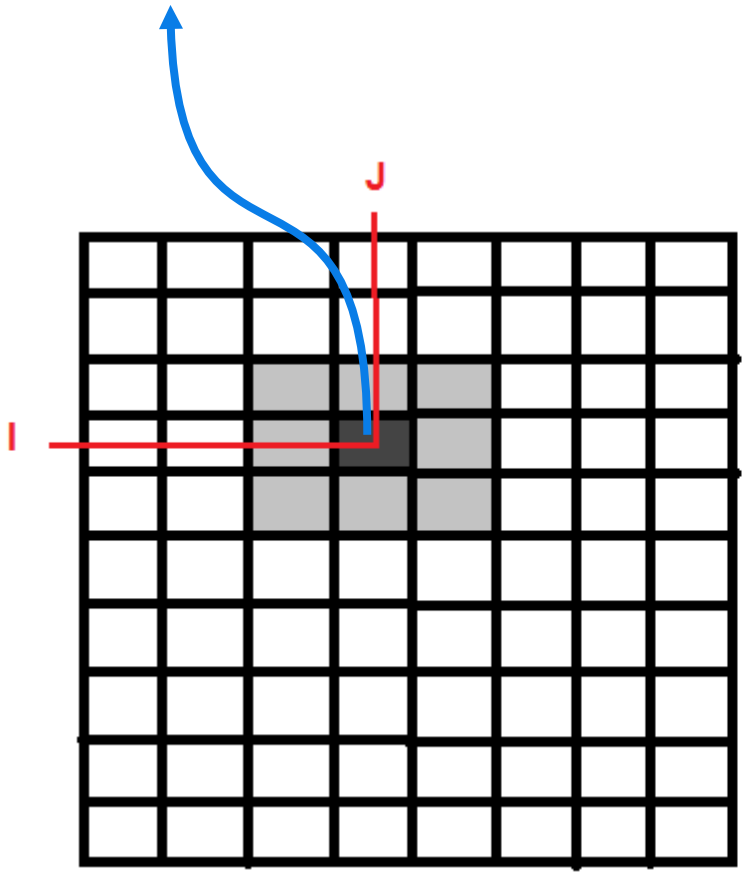


Peggioramento

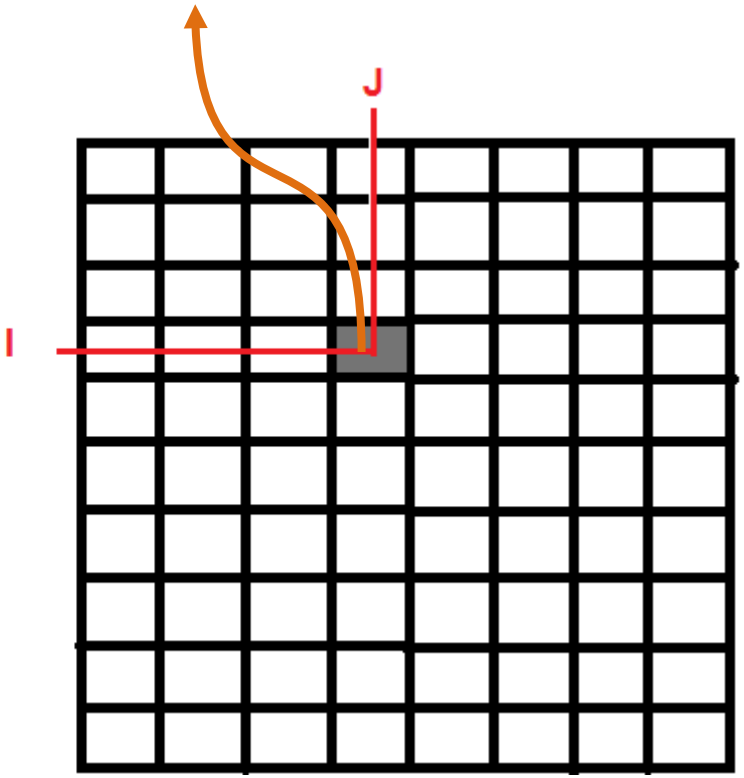
Un primo semplice esempio di filtro

Possiamo pensare di rimpiazzare il valore di un pixel con la media dei valori del pixel stesso e degli ***otto*** pixel vicini.

$$F(i, j) = p0$$



$$F'(i, j) = p0'$$



Esempio di applicazione “numerica” dei filtri

Consideriamo il filtro della “media” rappresentato dalla seguente matrice:

$$A_{avg} = \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

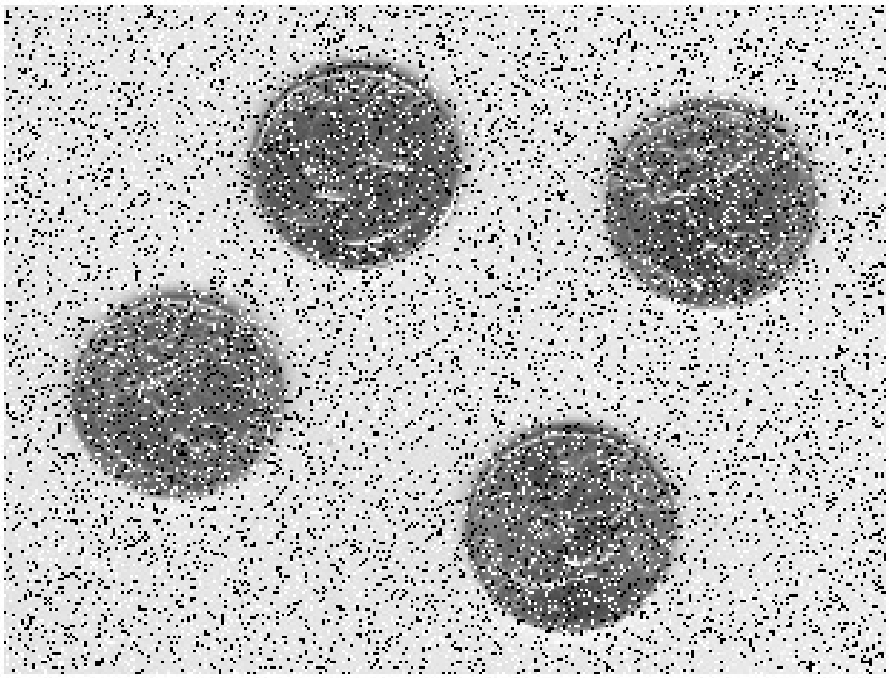
21	46	52	126	190	68
250	3	120	30	25	1
216	66	3	126	48	79
95	12	90	102	22	90
15	26	87	2	55	40
83	145	57	18	9	79

$$\frac{3+21+46+52+120+3+66+216+250}{9}$$

$$\frac{120+46+52+126+30+126+3+66+}{9}$$

86	64	80	77
95	61	63	58
68	57	59	51
68	60	49	46

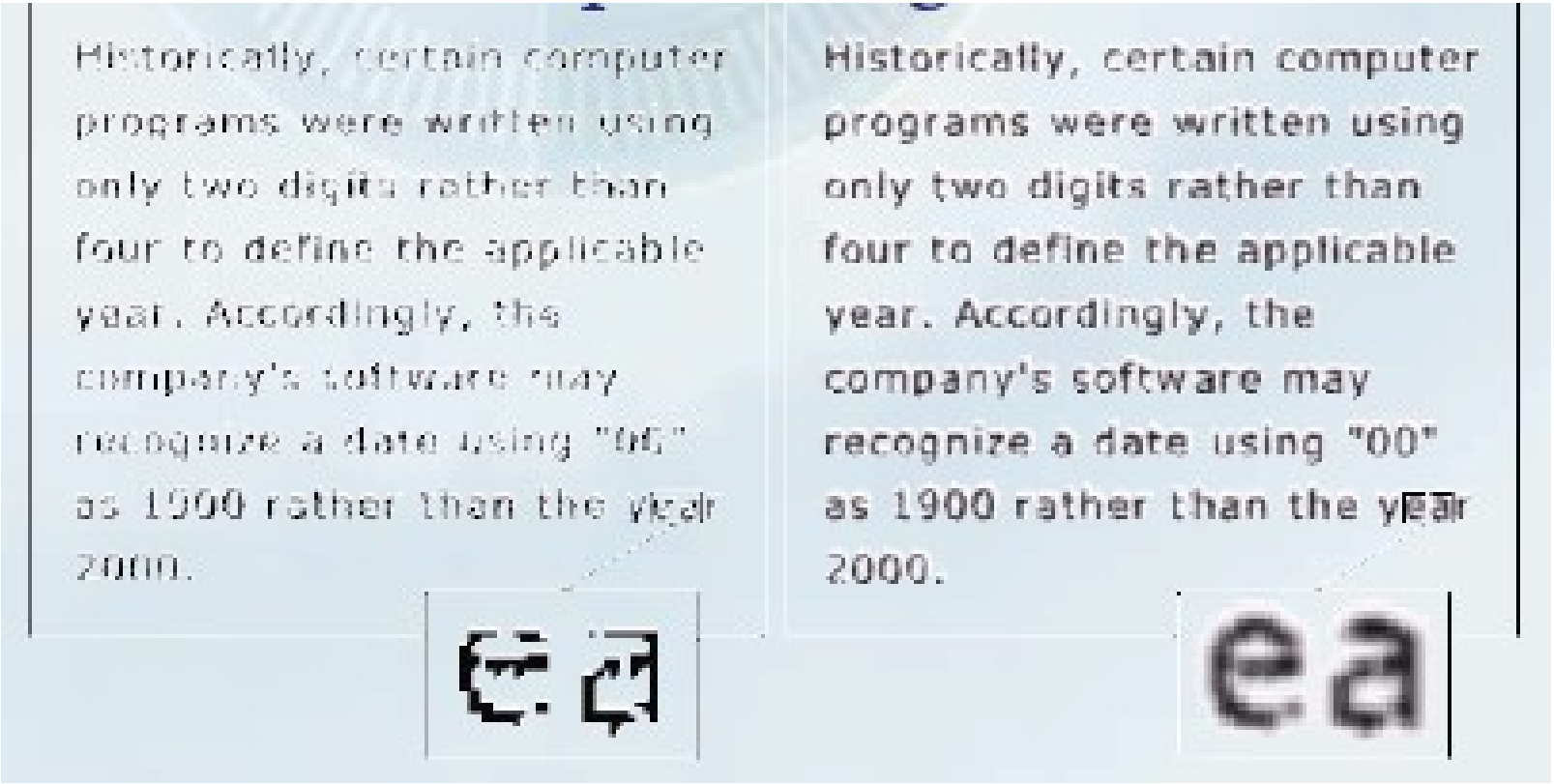
Esempio di applicazione del filtro “media”



Applicazioni di filtri

Applicazioni di filtri

Il trasferimento del testo via fax o altri processi generano un deterioramento caratterizzato dalla presenza di buchi, distorsioni della forma, ecc., ininfluenti per il riconoscimento umano, ma problematici per quello automatico.

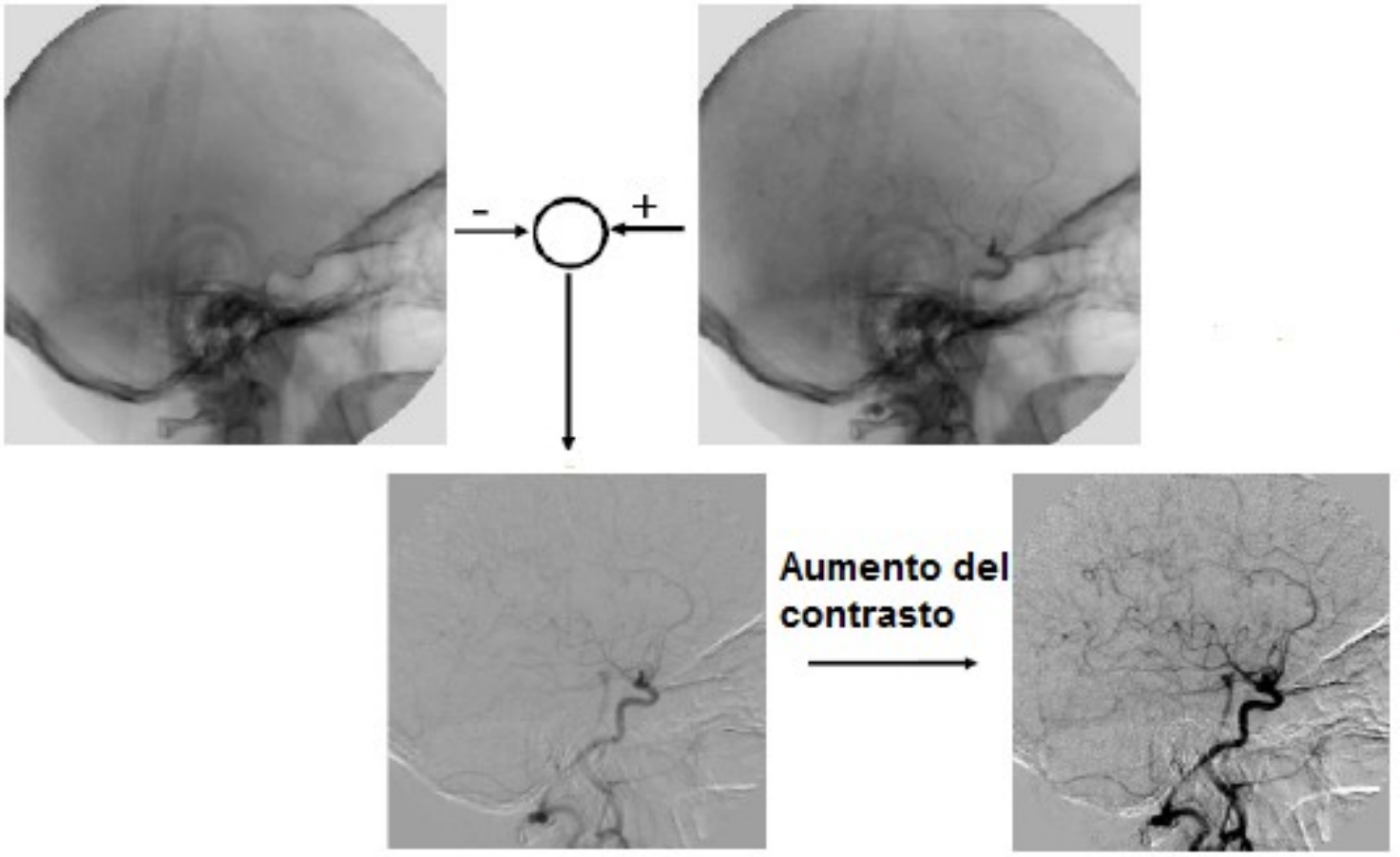


Applicazioni dell'elaborazione di immagini digitali

- Applicazione in ambito medico
- Estrazione dei contorni per:
  - riconoscimento dell'iride
  - riconoscimento di targhe automobilistiche

Applicazioni in ambito medico

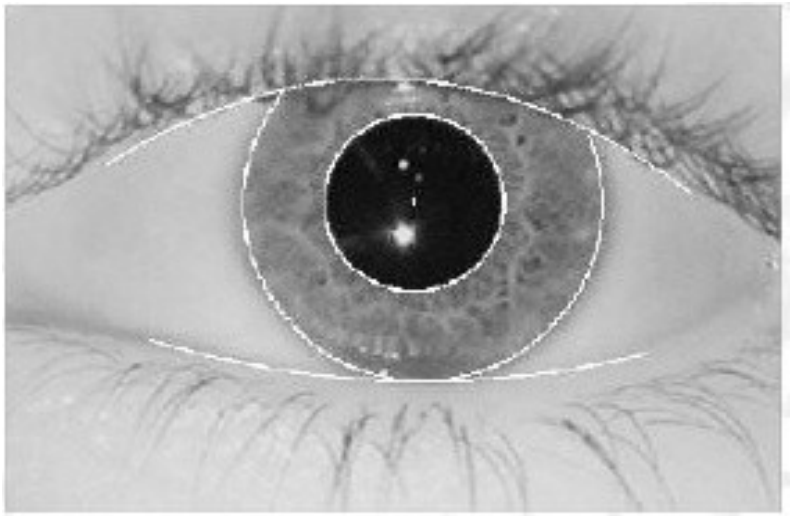
Angiografia a sottrazione digitale (DSA): è un'applicazione molto utilizzata in medicina per evidenziare le arterie da una serie di radiografie



Estrazione dei bordi: riconoscimento dell'iride

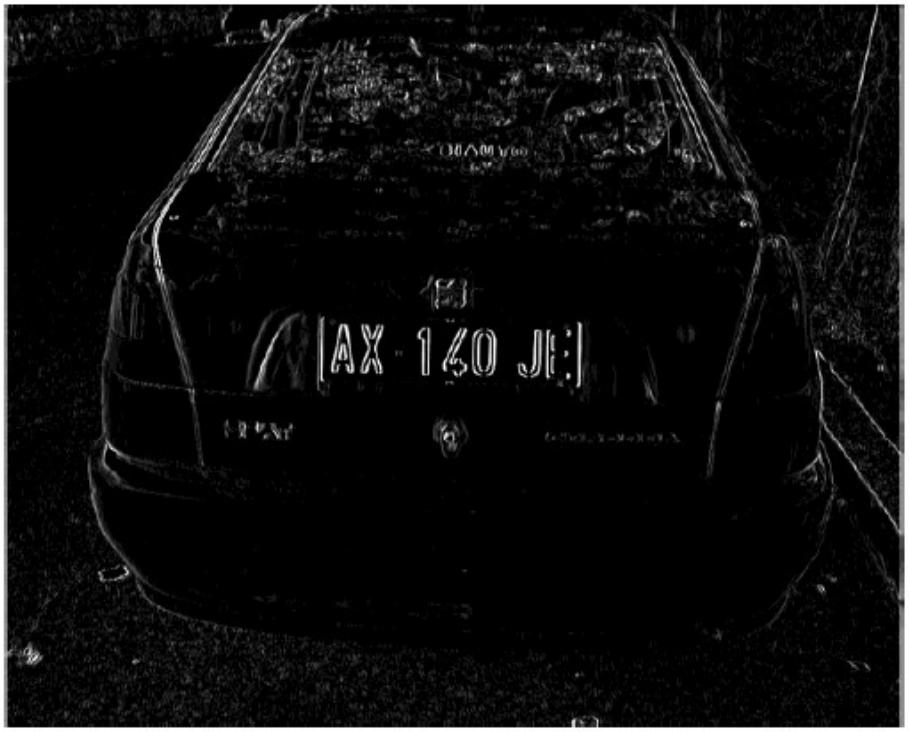
- I bordi nelle immagini sono zone dove l'intensità varia rapidamente; spesso corrispondono a contorni di oggetti
- Esistono algoritmi per l'estrazione dei contorni basati su filtri

$$X_{mask} = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \qquad Y_{mask} = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$



Estrazione dei bordi: riconoscimento di targhe

Applicazione del filtro di Sobel lungo l'asse x



Applicazione del filtro di Sobel lungo l'asse y

OCTAVE: ciclo “for”

Definiamo una matrice di ‘zeri’: **T = zeros(8,10)**

oppure di ‘uni’: **T = ones(8,10)**

oppure mediante un ‘loop’:

```
for i=1:8
    for j=1:10
        T(i,j) = i/j;
    end
end
```

Octave per le immagini

Per leggere un'immagine da file

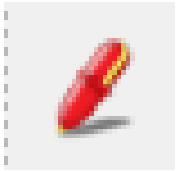
I = imread('FILENAME');

FILENAME comprende anche l'estensione del file

Per visualizzare l'immagine

```
figure(1)
imshow(I);
title('immagine');
```

Utilizzo dell'editor di QtOctave

Apriamo l'editor di QtOctave cliccando su 

Creiamo un file "New" con i seguenti comandi:

```
T=zeros(5,6);

for i=1:5
    for j=1:6
        T(i,j)=i*j;
    end
end
```

Salviamo il file in C:/qtoctave-0.6.8 come "prova" e richiamamolo da Terminale.

Esperimenti con QtOctave: caricamento di un'immagine

Carichiamo un'immagine da file. Per farlo occorre:

- Entrare nella cartella "public" seguendo il percorso:

Start -> My Computer -> public on 'Samba 3.4.7-0.50.fc11 (192.168.100.254)'

e aprire la cartella ***immagini***

Esperimenti con QtOctave: caricamento di un'immagine

Scegliere dall'elenco l'immagine che più piace

Per salvare l'immagine scelta: una volta aperta, si clicca sopra con il tasto destro.

L'immagine va salvata nella cartella:

C:/qtoctave-0.6.8

Esperimenti con QtOctave: caricamento e visualizzazione di un'immagine

Carichiamo un'immagine dalla cartella 'immagini':

I = imread('flower_grigio.png');

Mostriamola a video:

```
figure(1)
I = imshow(I);
title('Immagine originale')
```

Esperimenti con QtOctave

Di tale immagine si può considerare solo un angolo,
ad esempio quello in alto a sinistra di dimensioni 10 x 10.

Per far ciò possiamo definire una nuova matrice di zeri di dimensione 10 x 10:

J = zeros(10);

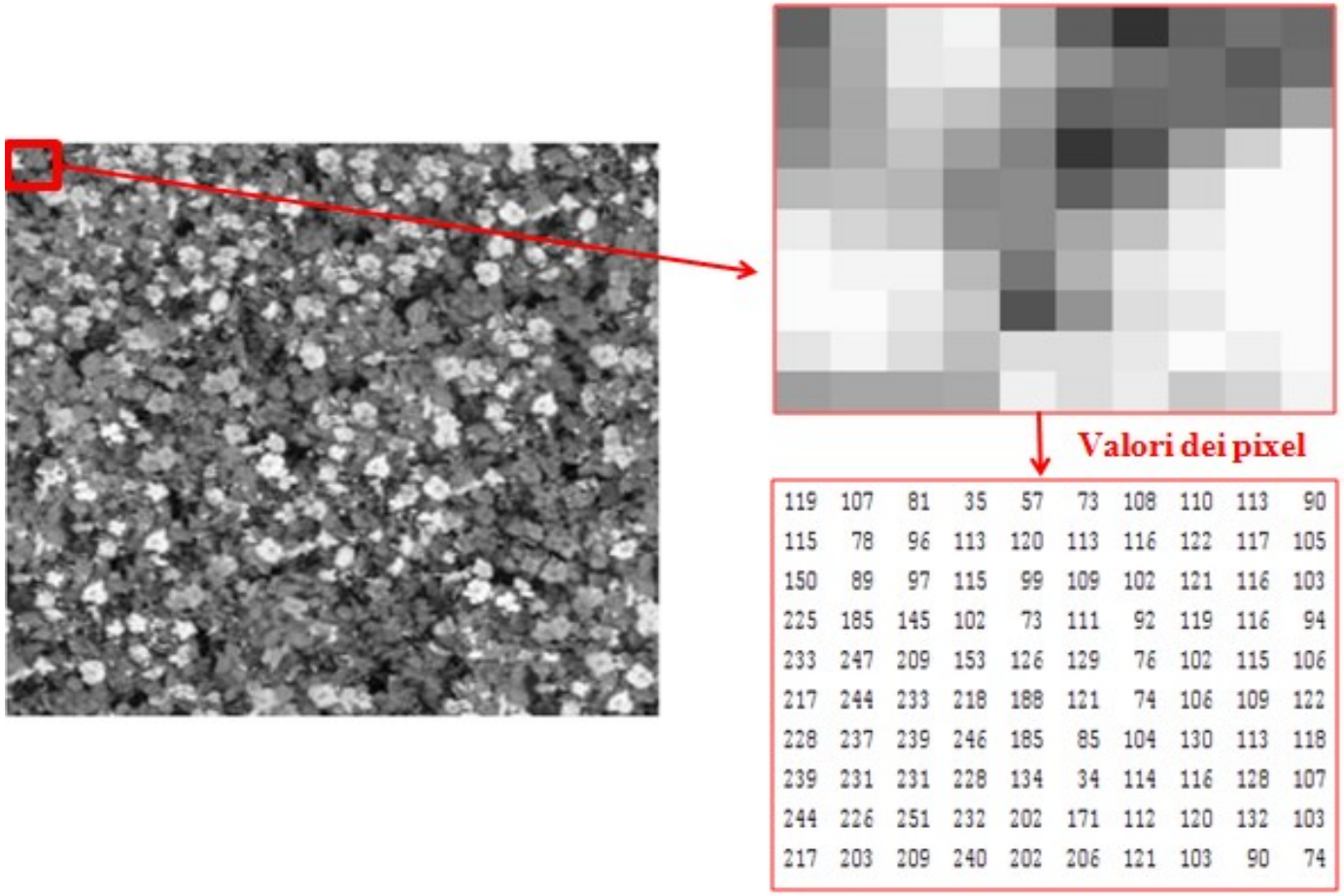
in cui copiamo l'angolo della matrice originale:

J = I(1:10,1:10);
figure(2)
imshow(J)
title('Zoom su immagine originale')

Esperimenti con QtOctave

E’ possibile anche stampare i valori della matrice J digitando da linea di comando:

J



Esperimenti con QtOctave

Supponiamo di voler cambiare il valore del pixel in posizione (2,2).

Visualizziamo prima il pixel:

disp('Valore originale del pixel in posizione (2,2):')

J(2,2)

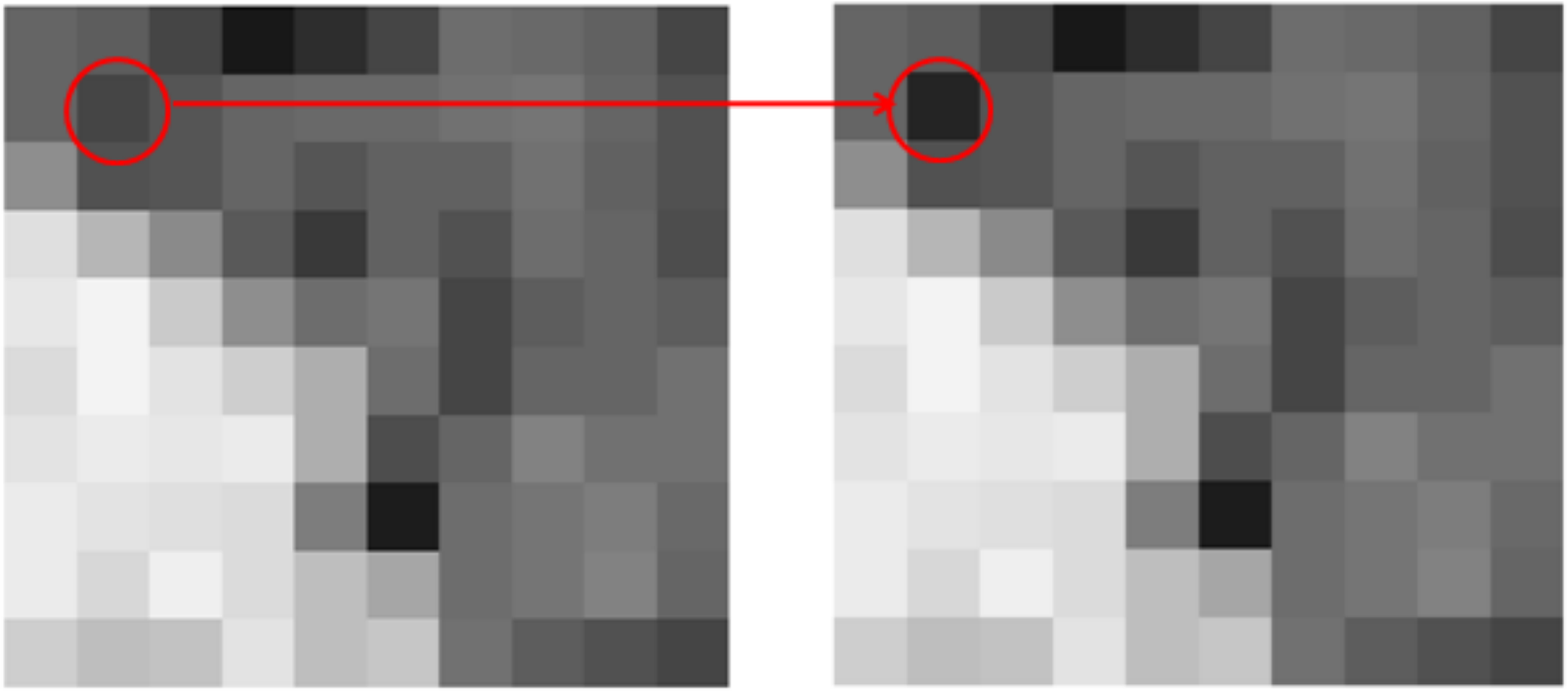
Esperimenti con QtOctave

Per modificarlo e visualizzare il risultato della modifica, utilizziamo i comandi:

```
J(2,2) = 40;
disp('Valore modificato del pixel in posizione (2,2):')
J(2,2)
figure(3)
imshow(J)
title(‘Angolo con pixel in posizione (2,2) modificato’)
```

Esperimenti con QtOctave

Il risultato sarà il seguente:



$J_{2,2} = 78$

$J'_{2,2} = 40$

Esperimenti con QtOctave: immagini RGB

Eseguiamo: **clear all**

clf

Carichiamo e visualizziamo a video un'immagine RGB a scelta.

Analogamente a prima consideriamo l'angolo in alto a sinistra e definiamo una nuova matrice composta da 3 matrici:

G = zeros(10,10,3);

Esperimenti con QtOctave

Per copiare al suo interno il contenuto della matrice originale e visualizzare il risultato:

G = I(1:10,1:10,:)
figure(1)
imshow(G)

Per modificare il colore all'interno del pixel, ad esempio in posizione (2,2), si devono ora modificare i tre valori delle tre componenti primarie.

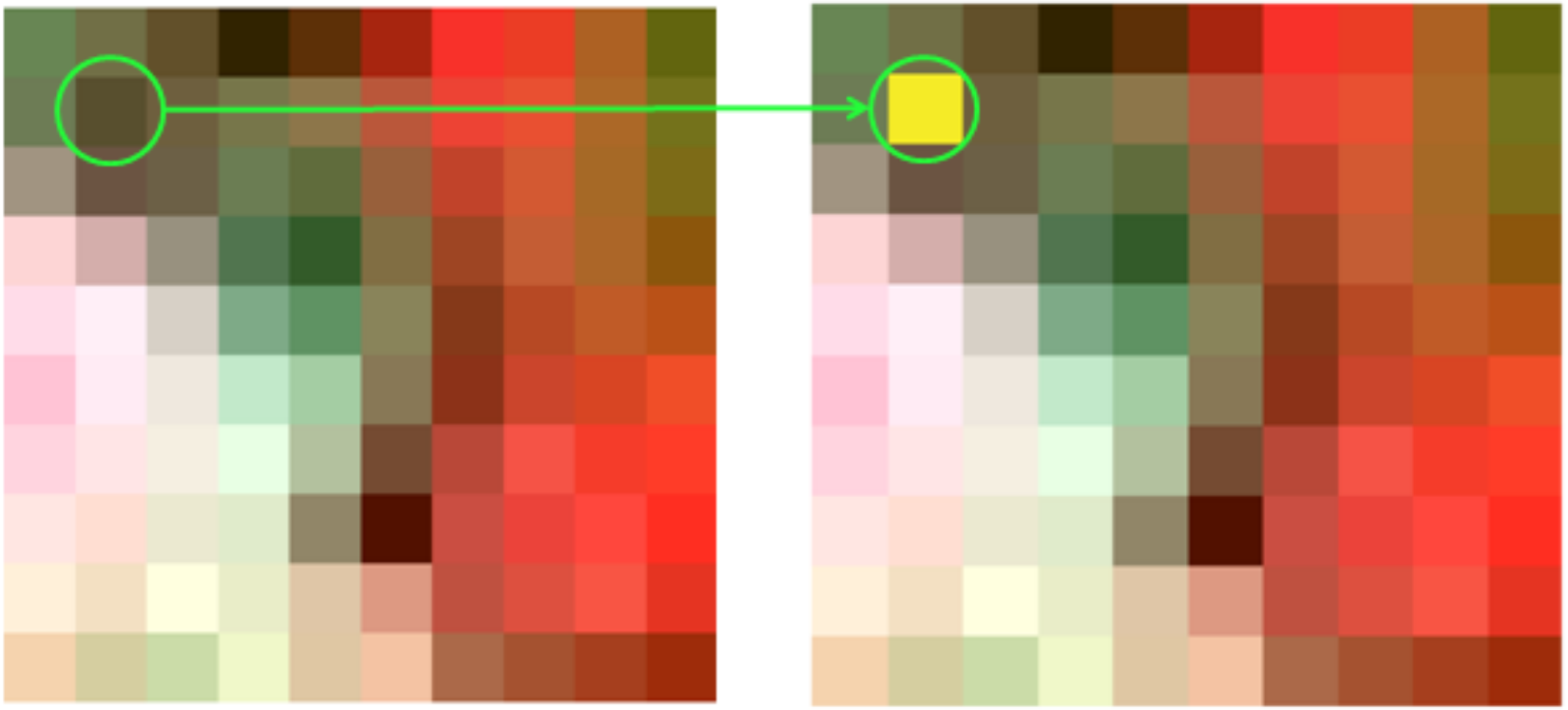
Esperimenti con QtOctave

Visualizziamo inizialmente il pixel e modifichiamone il valore assegnando una nuova terna:

```
disp('Valore originale del pixel in posizione (2,2):')
G(2,2,:)
G(2,2,:) = [245  235  39];
disp('Valore modificato del pixel in posizione (2,2):')
G(2,2,:)
imshow(G)
```

Esperimenti con QtOctave

Il risultato sarà il seguente:



$G_{2,2} = (88,79,46)$

$G'_{2,2} = (245,235,39)$

Esperimenti con QtOctave: RGB2GRAY

Possiamo trasformare l'immagine a colori nella relativa a scala di grigi con il comando:

H = rgb2gray(G);

combina le tre componenti primarie per ottenere il livello di grigio corrispondente con la seguente regola:

0.299*R + 0.587*G + 0.114*B;

Esperimenti con QtOctave: RGB2GRAY

```
I = imread('anatra_rgb.png');
figure(1)
imshow(I)

R = I(:,:,1);
G = I(:,:,2);
B = I(:,:,3);

S = 0.299*R+ 0.587*G + 0.114*B;
figure(2)
imshow(S)
```

Esperimenti con QtOctave: aggiunta di costante

```
I = imread(C:/immagini/pinguini_grigi.png');
figure(1)
imshow(I)
title('Immagine originale')
J = uint8(zeros(size(I)));
C = 50;
for i=1:size(J,1)
     for j = 1:size(J,2)
          J(i,j) = I(i,j)+C;
          if J(i,j)>255
             J(i,j)=255;
          end
      end
end

figure(2)
imshow(J)
title(['Aggiunta della costante C= ', num2str(C)])
```

Esperimenti con QtOctave: sottrazione di costante

```
K = uint8(zeros(size(I)));
D = 60;

for i=1:size(K,1)
       for j = 1:size(K,2)
            K(i,j) = I(i,j)-D;
            if K(i,j)<0
               K(i,j)=0;
            end
      end
end

figure(3)
imshow(K)
title(['Sottrazione della costante D= ', num2str(D)])
```

Grazie per l'attenzione

e

Buone Vacanze!!