Laboratorio di Elaborazione di Immagini

Esercitazione 5:

SPAZI COLORE

Silvia Obertino

19 aprile 2017

Comunicazioni di servizio

- Lezione del 26 aprile sospesa

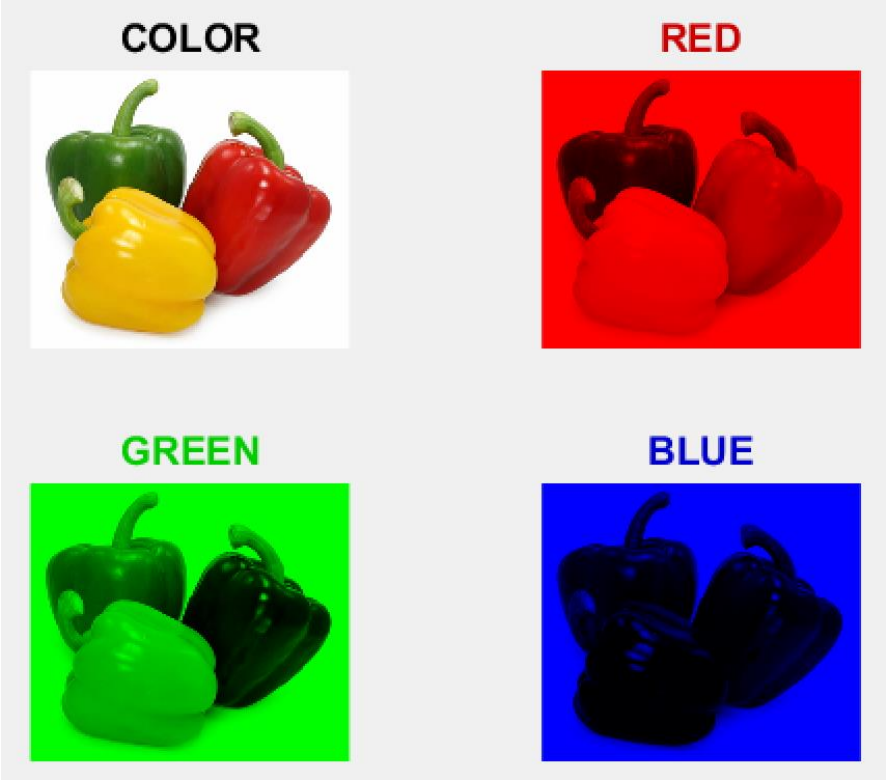
Colori RGB

- In MATLAB di default quando viene letta un immagine a colori viene caricata nello spazio RGB
  - Red
  - Green
  - Blue
- Questo spazio colore, derivante dalla fisica, potrebbe non essere il migliore per elaborare le immagini

esempio

```
>> img = imread('peppers.jpg');
```

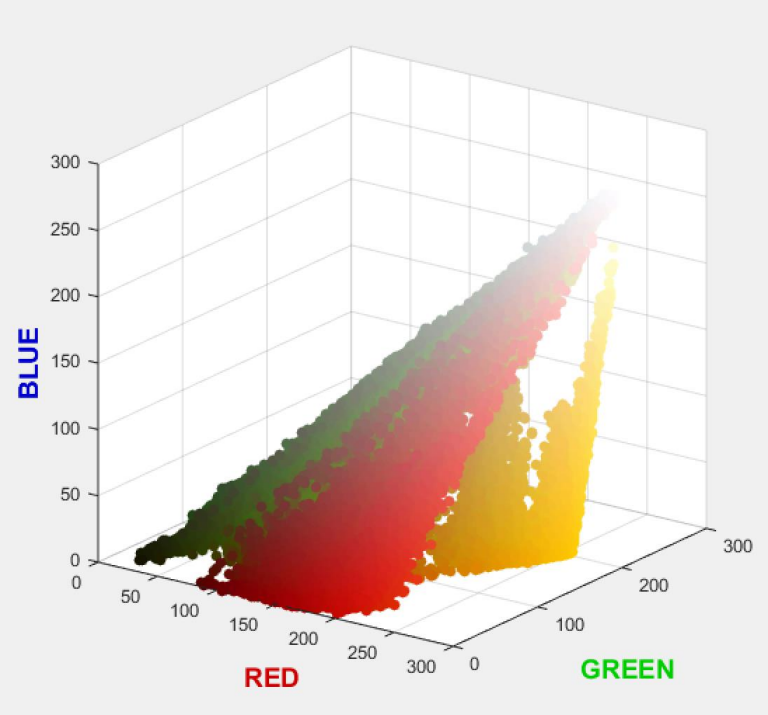
```
>> rosso = img(:,:,1);
>> verde = img(:,:,2);
>> blu = img(:,:,3);
```



esempio

```
>> [x,y,c] = size(img);
```

```
>> img_rgb = double(reshape(img,[x*y,c]));
>> figure;
```




```
>> scatter3(img_rgb(:,1), img_rgb(:,2), img_rgb(:,3), [] ,img_rgb/256.0, 'filled')
```

Colori LAB

- **Idea**: serviva uno spazio colore “comune” per gestire le immagini che fosse indipendente dal dispositivo usato per la visualizzazione (monitor o stampante)

- È stato progettato per approssimare il **sistema visivo umano** con una componente luminosità e uno spazio colore uniforme

Colori LAB

- Lo **Spazio colore Lab** o **CIELAB** o **CIE 1976 (L*, a*, b*)** è uno spazio colore
  - **L** per la dimensione luminosità
  - **a** e **b** per le dimensioni del colore

- È basato sulle coordinate dello spazio colore non lineare compresso CIE XYZ

Colori LAB

- La **luminosità** è calcolata usando la radice cubica della luminanza relativa

- Lab include tutti i colori percepibili ed è **indipendente** dal dispositivo che li rappresenta

Colori LAB in MATLAB

- Da RGB a LAB:

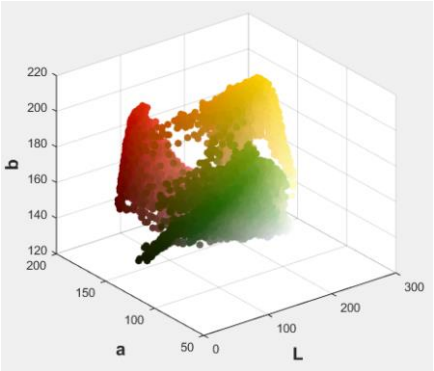
```
>> colorTransform = makecform('srgb2lab');
>> lab = applycform(img, colorTransform);
```

- Da LAB a RGB:

```
>> colorTransform = makecform(‘lab2srgb');
>> img = applycform(lab, colorTransform);
```

Colori LAB

- Data un'immagine provate a convertirla in LAB e visualizzare:
  - i tre canali dello spazio
  - lo spazio colore in un grafico 3D



Colori YUV

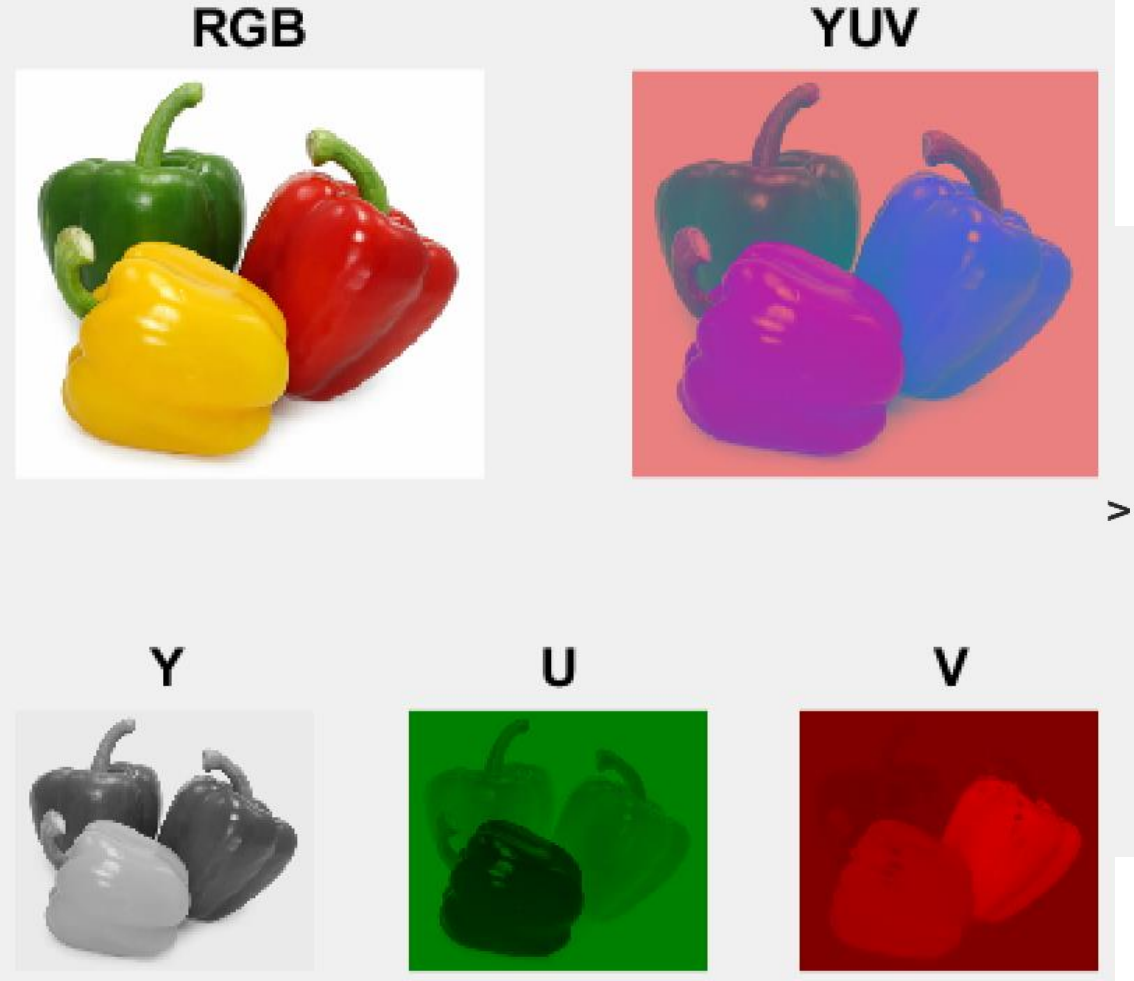
- Lo **Spazio colore YUV (o YCbCr)** è uno spazio colore
  - **Y** per la dimensione luminanza
  - **U** e **V** per le dimensioni della crominanza

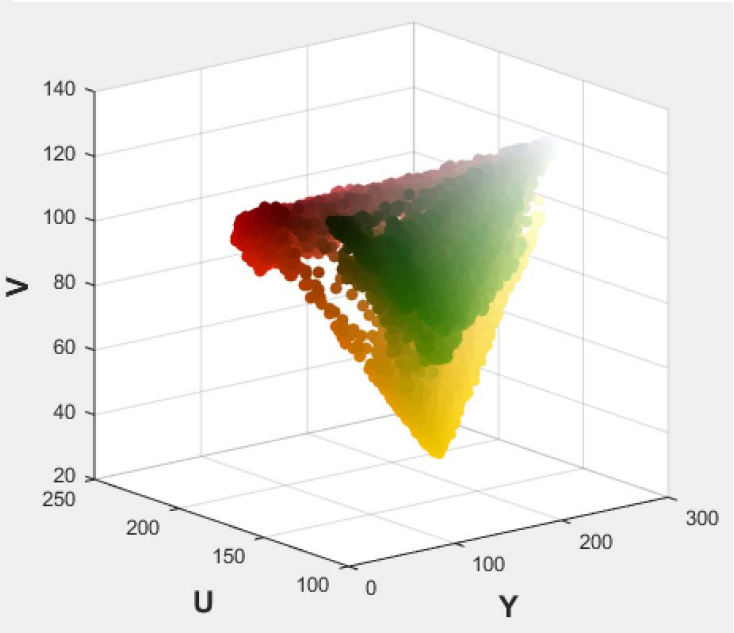
- Storicamente definito per le televisioni analogiche in bianco e nero

Colori YUV

- Data un'immagine provate a convertirla in YUV e visualizzare:
  - i tre canali dello spazio
  - lo spazio colore in un grafico 3D

Colori YUV


RGB
YUV
Y
U
V


140
120
100
80
60
40
20
V
250
200
150
100
U
0
100
200
300
Y

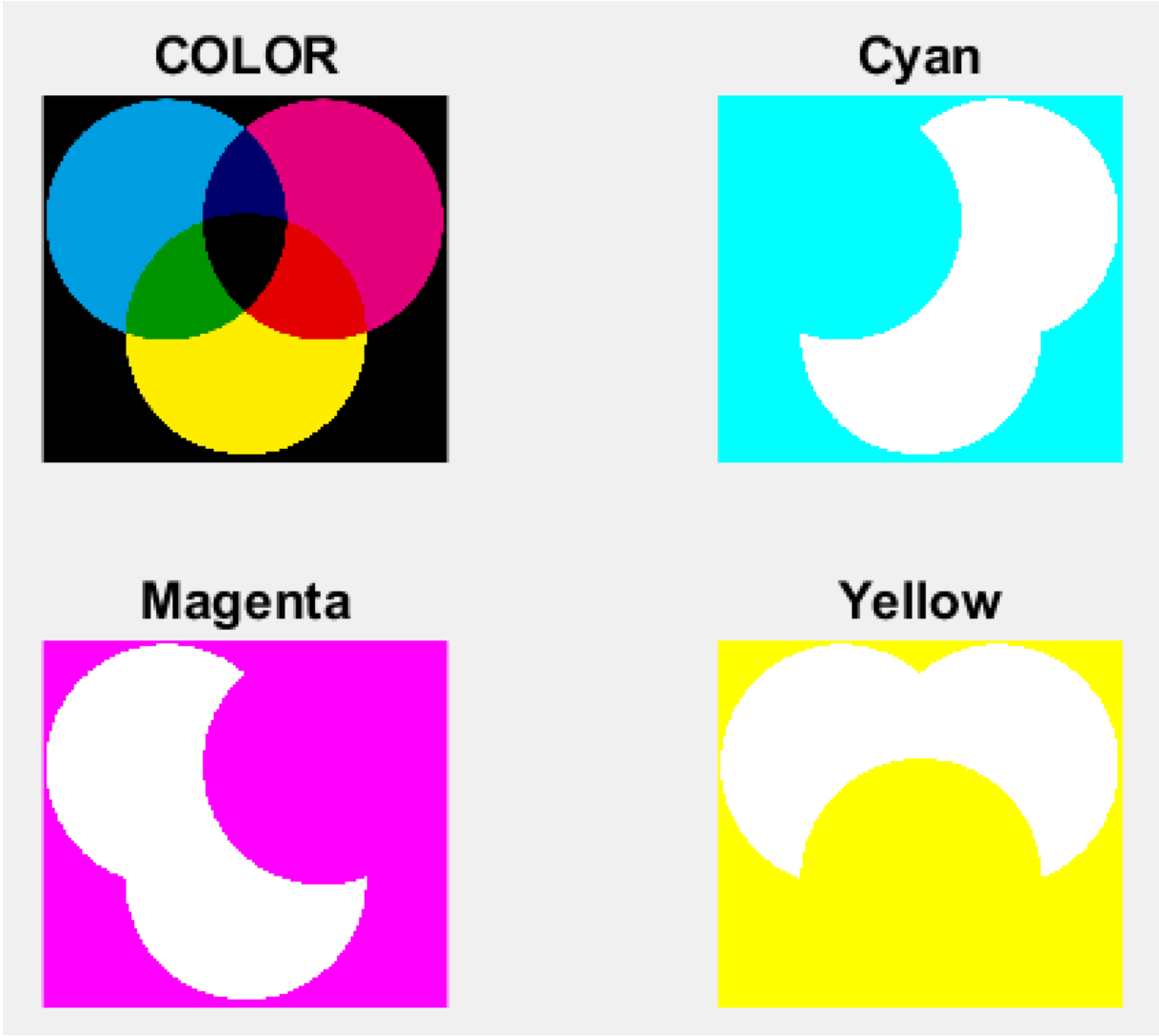
Colori CMY

- Lo **Spazio colore CMY** è uno spazio colore definito da
  - Cyan
  - Magenta
  - Yellow
- Usato in particolare per i sistemi di stampa

Colori CYM

- Data un'immagine provate a convertirla in CYM e visualizzare cosa la vostra stampante stampa:
  - i tre canali dello spazio

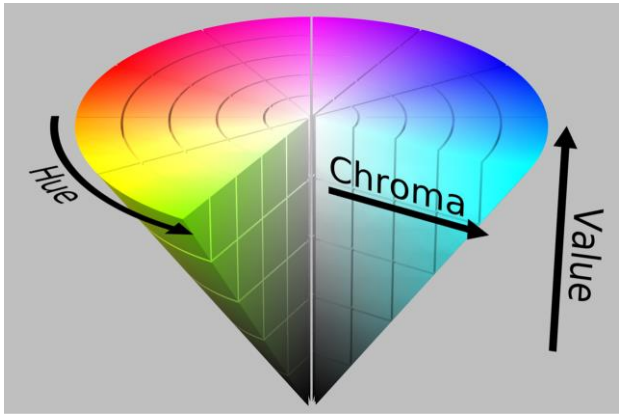
Colori CMY



Colori HSV

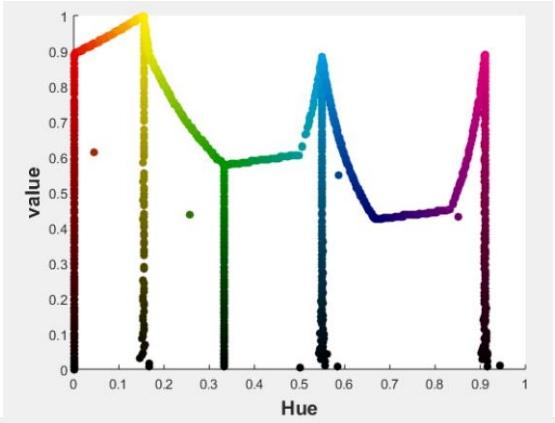
- Lo **Spazio colore HSV (o HSI o HSL)** è uno spazio colore
  - **H** per il colore (Hue)
  - **S** per la saturazione (Saturation)
  - **V (o I o L)** per l'intensità (Value o Intensity o Lightness)

- Usato maggiormente per l'enhancement delle immagini digitali

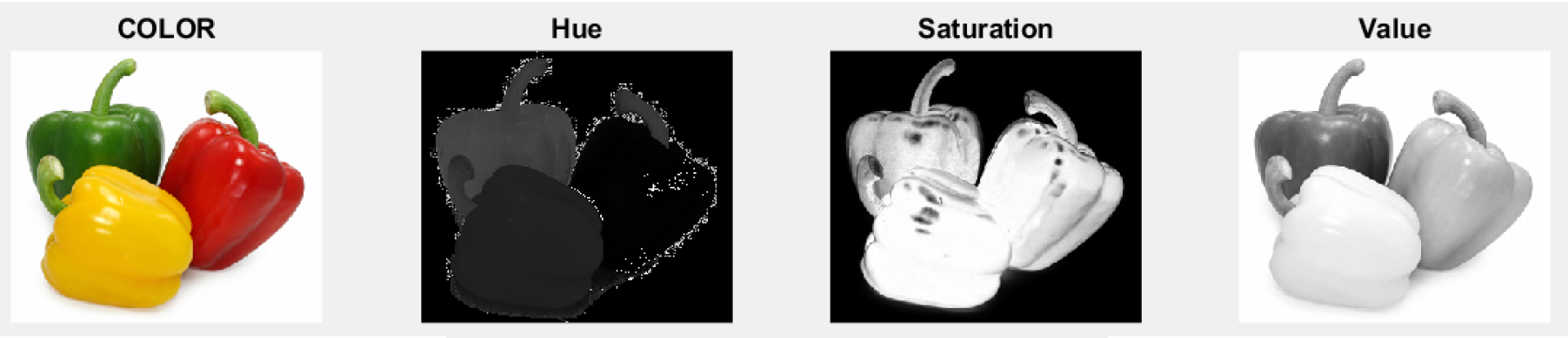


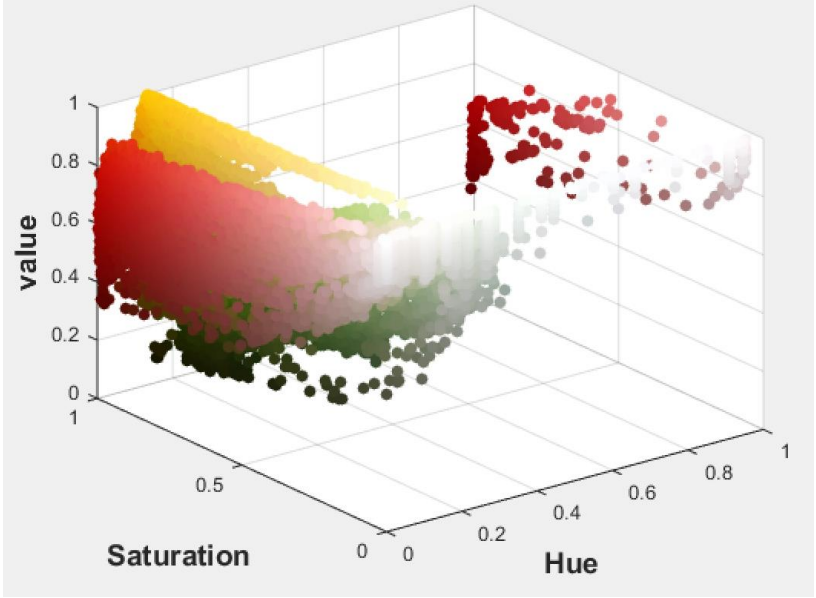
Colori HSV

- Data un'immagine provate a convertirla in HSV e visualizzare:
  - i tre canali dello spazio
  - lo spazio colore in un grafico 3D



Colori HSV


COLOR
Hue
Saturation
Value


value
1
0.8
0.6
0.4
0.2
0
Saturation
0
0.5
1
Hue
0
0.2
0.4
0.6
0.8
1

Laboratorio di Elaborazione di Immagini

Esercitazione 5:

MORFOLOGIA

Silvia Obertino

19 aprile 2017

Operazioni morfologiche

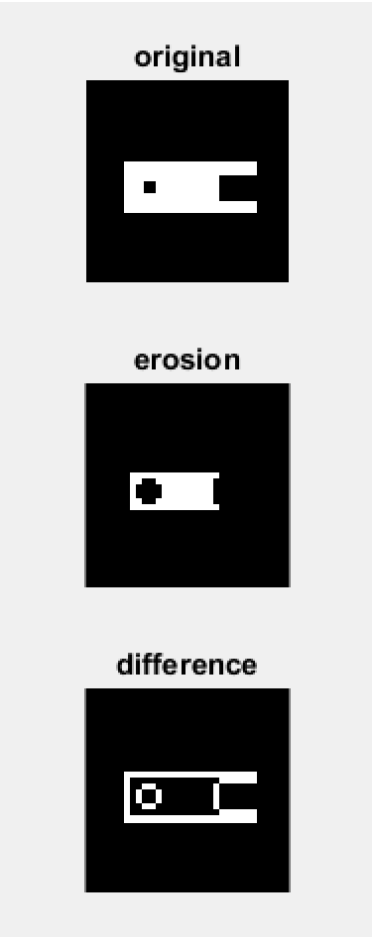
- Erosione
- Dilatazione
- Apertura
- Chiusura

Erosione

```
function img_erode = my_erode(img,se)

[x,y] = size(img);
img_erode = zeros(x,y);
[sx,sy] = size(se);

lx = floor(sx/2);
ly = floor(sy/2);

for i=(lx+1):(x-lx)
    for j=(ly+1):(y-ly)
        roi = img(i-lx:i+lx, j-ly:j+ly);
        bool_roi = and(roi,se);
        if isequal(se,bool_roi)
            img_erode(i,j) = 1;
        end
    end

end
```

Erosione

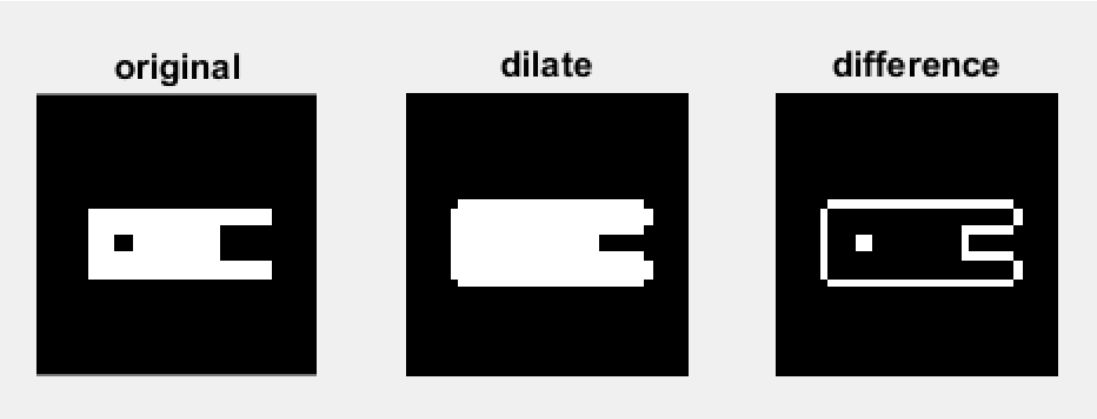
- Caricare l'immagine pixels_wb.tiff
- Fare una soglia a 100 per binarizzarla
  - img = img > 100;
- Creare un elemento strutturale (es: crocetta)
  - s = [0,1,0;1,1,1;0,1,0]
- Applicare l'operazione di erosione
  - img_erode = my_erode(img,s);
- Visualizzare il prima il dopo e le differenze



Dilatazione

- Provate a modificare il codice di “my_erode” per creare l’operazione di dilatazione

- Create una nuova function e salvatela come my_dilate.m



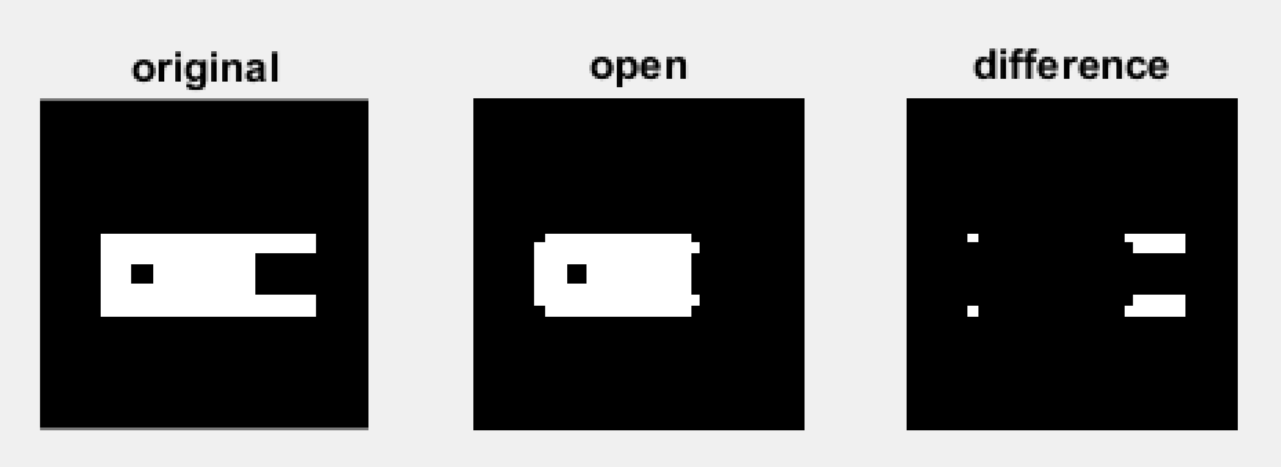
MATLAB: erosione e dilatazione

- MATLAB possiede già delle implementazioni
  - strel
  - imerode
  - imdilate
- Confrontate il risultato di my_dilate con quello di imdilate per controllare che abbiate fatto giusto
  - s = strel(‘diamond’,2);
  - isequal(img_dilate, img_dilate2)

Apertura

- L’apertura è un erosione seguita da dilatazione

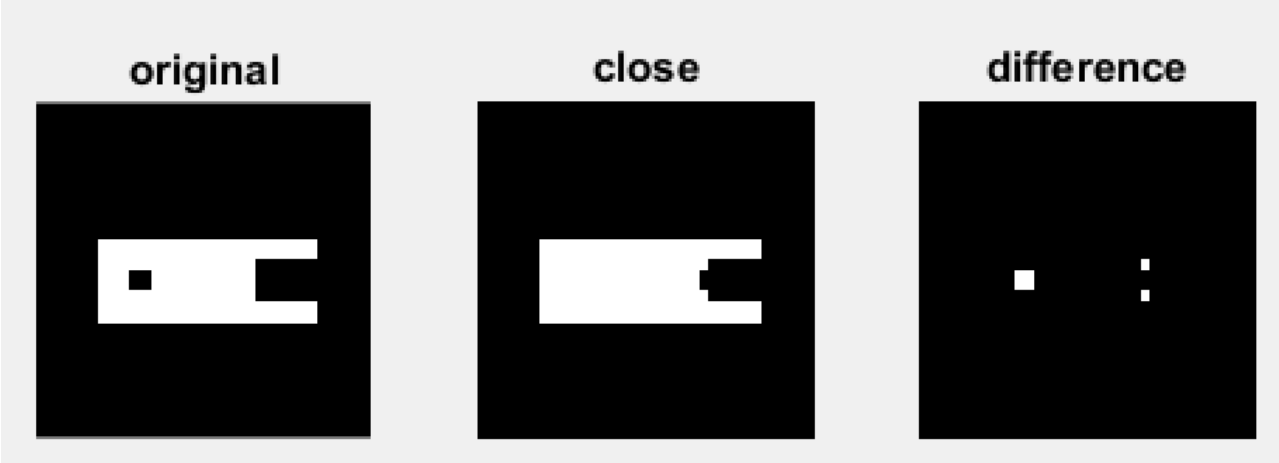
- Provate!



Chiusura

- La chiusura è una dilatazione seguita da una erosione

- Provate!



MATLAB: apertura e chiusura

- MATLAB possiede già delle implementazioni
  - imopen
  - imclose

Applicazione: immagini grayscale

- Caricate l'immagine "lena.jpg" (o la vostra preferita)
- Convertitela in scala di grigi
- Testate gli operatori morfologici

- Provate a cambiare l'elemento strutturale… Cosa cambia?

Applicazione: immagini grayscale

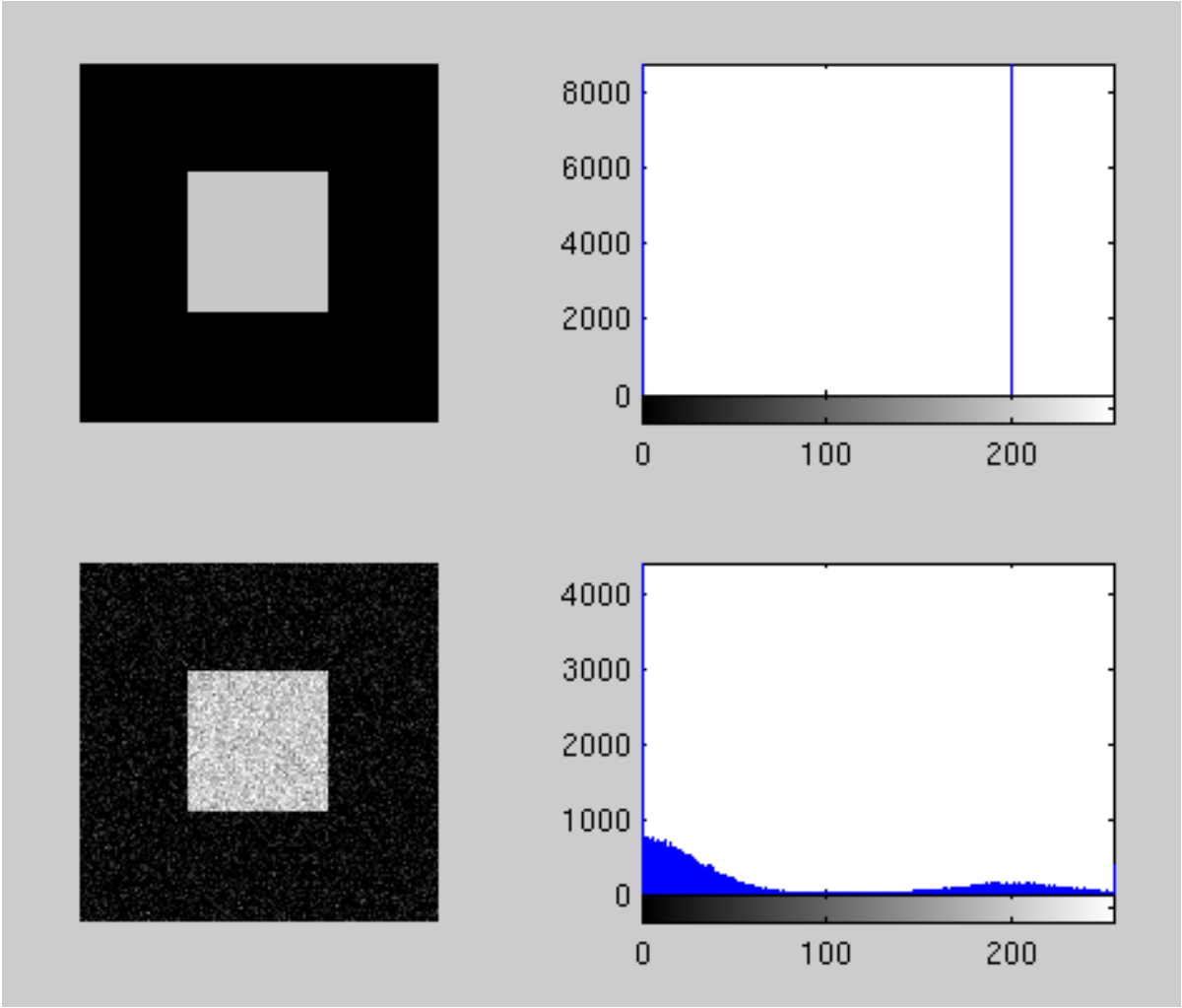
Applicazione: NOISE REDUCTION

- Create un immagine 256x256 con dentro un quadrato bianco

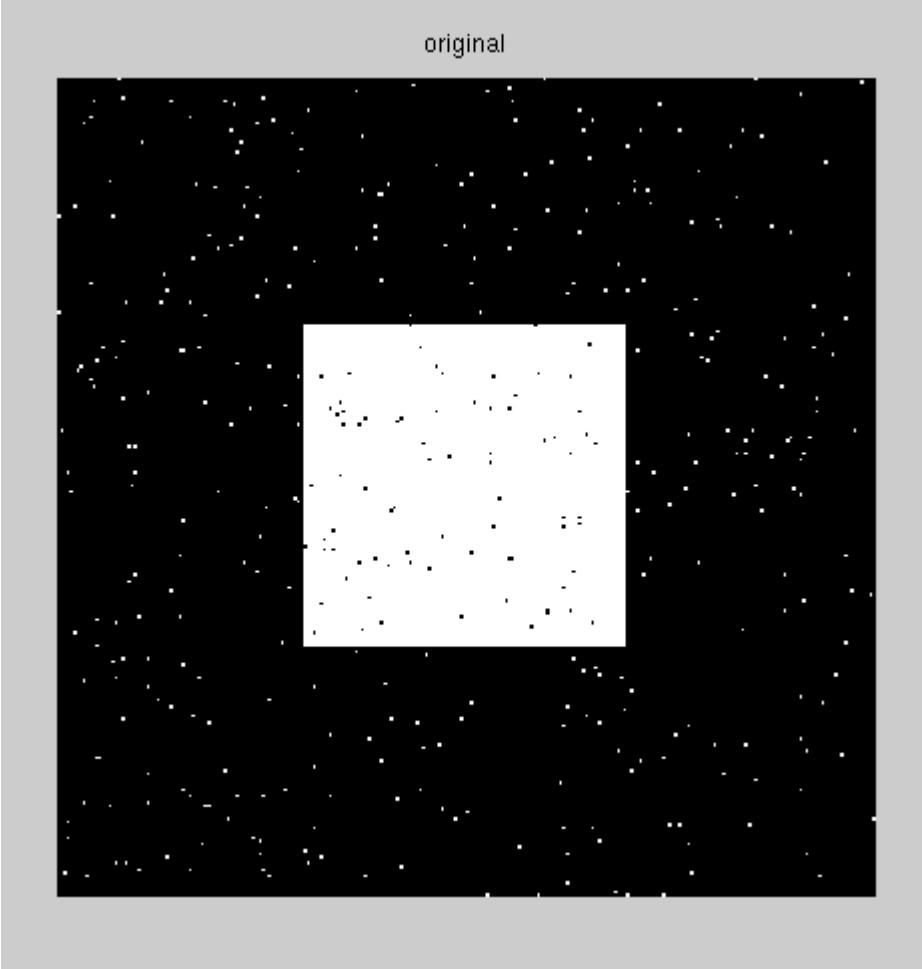
- Aggiungete rumore

- Provate a sogliare l'immagine (aiutandovi con l'istogramma) al fine di ritrovare il quadrato

Applicazione: NOISE REDUCTION



Applicazione: NOISE REDUCTION



Applicazione: NOISE REDUCTION

- Trovate la sequenza di operatori morfologici che vi permetterà di “ripulire” il quadrato