Bioimmagini – Esercitazione1 - 27 Marzo 2011

Introduzione all'IMAGE PROCESSING Toolbox di Matlab

Matlab:
piattaforma mediante la quale è possibile operare con un linguaggio che non deve essere compilato; sistema interattivo in cui il dato elementare è un vettore che non richiede di essere dimensionato a priori.

Image Processing Toolbox:
insieme di funzioni che estendono la capacità dell'ambiente di calcolo di Matlab, supportando un'ampia gamma di operazioni sull'elaborazione delle immagini.

Immagine = matrice. Ogni elemento della matrice corrisponde ad un *pixel* nell'immagine visualizzata.

Come si legge un file contenente un'immagine in Matlab?

- Immagine in formato array di Matlab (*.mat)

  load filename

- Immagine in formato grafico (*.bmp, *.jpg, *.tiff,…)

  immagine=imread('filename');
  info=imfinfo('filename');

- Immagine in formato DICOM (Digital Imaging and Communications in Medicine) (*.dcm)

  immagine=dicomread('filename');
  info=dicominfo('filename');

- Immagine in formato analyze7.5 (*.hdr, *.img)

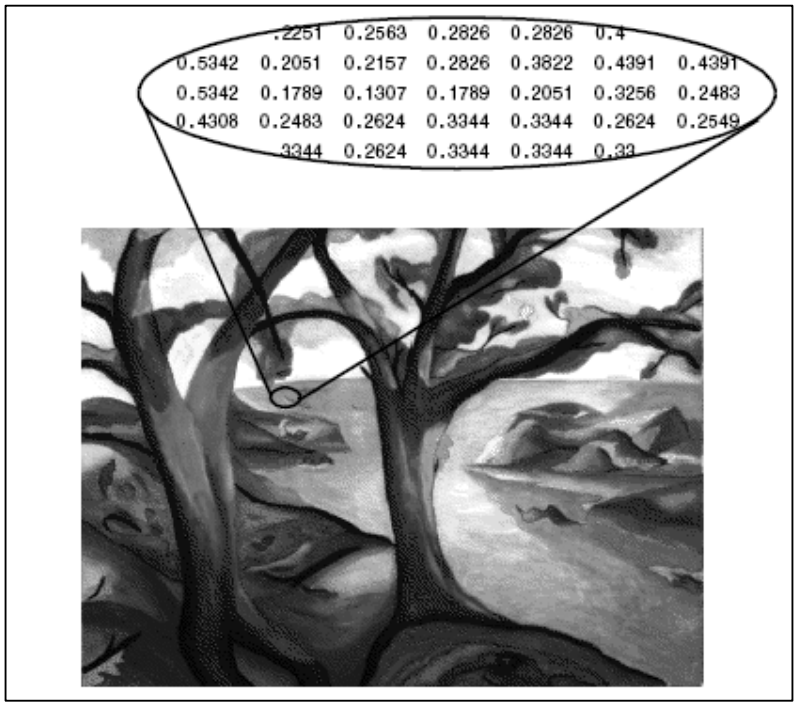
  immagine=analyze75read('filename');
  info=analyze75info('filename');

- Immagine in altro formato: occorre utilizzare le funzioni *fopen.m* e *fread.m.*

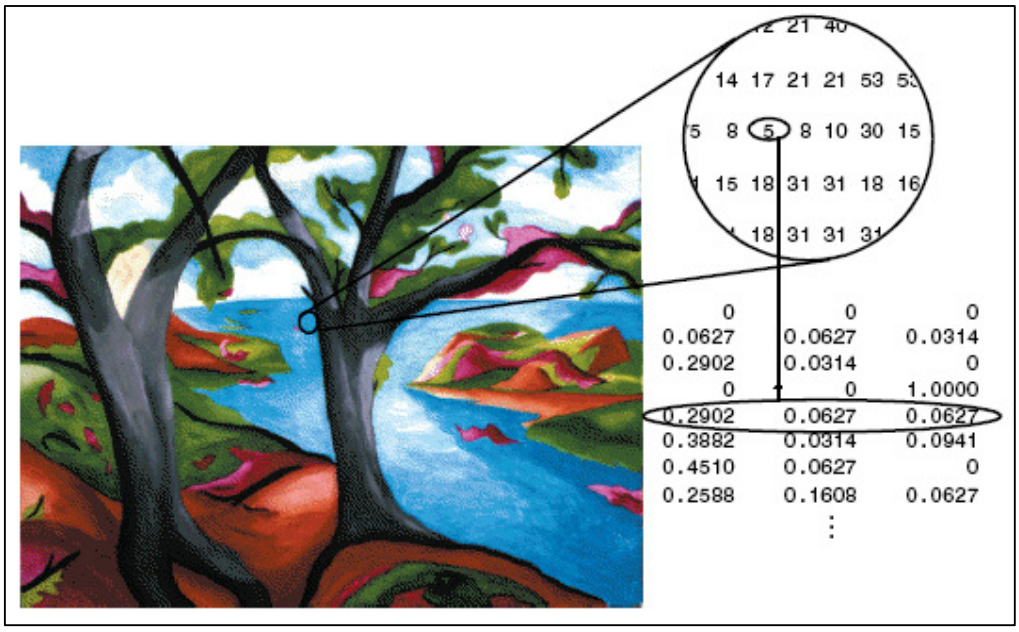
Quali tipi di immagini sono supportate dal Matlab?

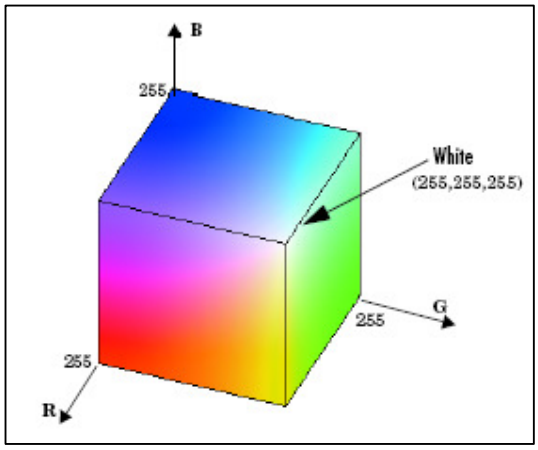
- Immagini di tipo intensità (gray levels): singola matrice, dove ogni elemento corrisponde al valore di grigio del pixel.
  Classi numeriche:
  - double: numeri reali in doppia precisione, 64 bits, range valori [0 1];
  - uint8: numeri interi senza segno, 8 bits, range [0 255];
  - uint16: numeri interi senza segno, 16 bits, range [0 65535];
  - single: numeri reali in singola precisione, 32 bits, range [0 1];

- int8, int16, int32, uint32 (interi con e senza segno, 8, 16 o 32 bits).

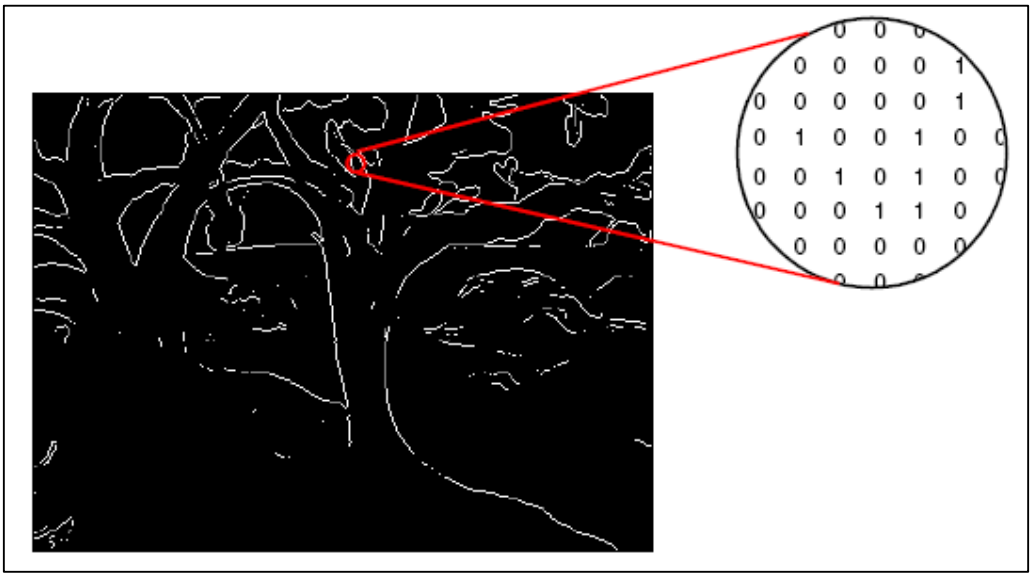


- Immagini di tipo indexed: due matrici, matrice dell'immagine e matrice della mappa dei colori (colormap).
  - Matrice immagine: il valore di ciascun pixel è l'indice della riga della colormap corrispondente al colore nel punto.
  - Colormap: ogni riga della mappa (classe double) definisce le componenti R G B di un singolo colore (numeri reali tra 0 e 1).

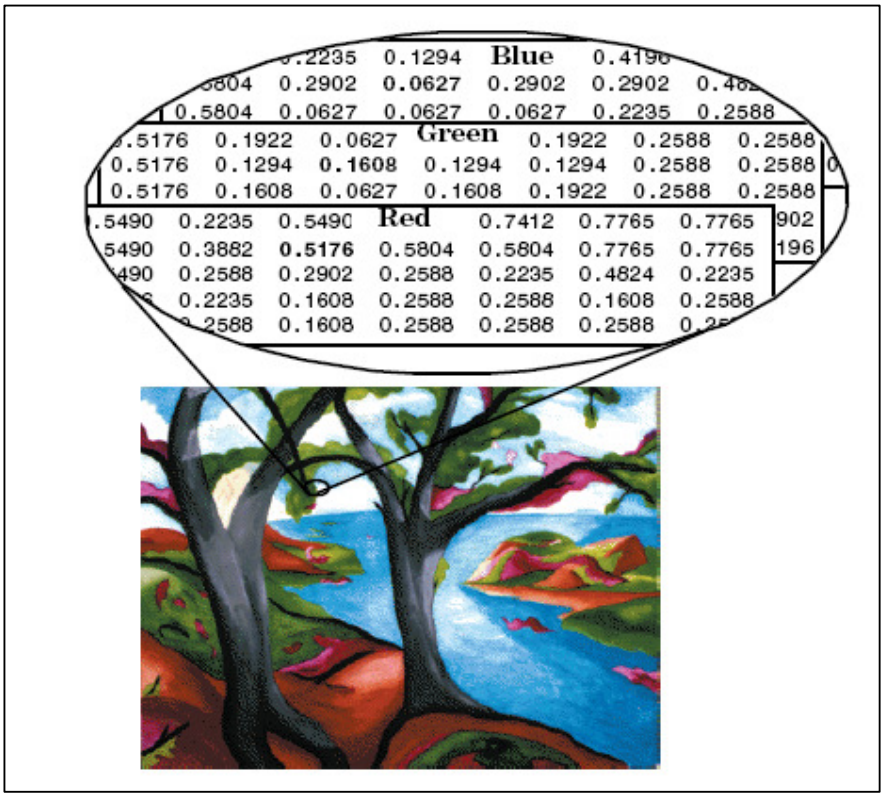




- Immagini binarie: una singola matrice in cui ogni elemento può assumere solo 2 valori discreti: 0 e 1. Classe numerica: logical.



- Immagini RGB: un array multidimensionale di dimensioni MxNx3 che definisce le componenti di colore R, G, B di ciascun pixel (classi double, uint8 o uint16).



Come si visualizzano le immagini in Matlab?

- Visualizzazione del contenuto della matrice come immagine in scala di grigi:
  imshow(im,[]) (default= *imshow(im,[intmin(class(im)) intmax(class(im))]*)

  Per le immagini di tipo intensità o indexed e possibile variare la mappa di colore mediante il comando *colormap* (help graph3d).
  Per visualizzare più immagini su una stessa figura occorre usare *imshow* insieme a *subplot*: tale comando divide la figura in mxn regioni visualizzabili rendendo attiva la regione selezionata. Esempio:
  subplot(121),imshow(im1,[]), subplot(122),imshow(im2,[])

- Visualizzazione del contenuto della matrice lungo l'asse z nello spazio 3D:
  mesh(im) oppure *surf(im)*

  Con il comando *axis* posso variare l'impostazione di visualizzazione degli assi:
  axis on/off: li mostro o meno
  axis xy: sistema cartesiano
  axis ij :sistema matriciale
  axis image: uso la stessa scala per visualizzare i 3 assi
  axis square: rendo quadrato il box
  axis normal: rimuove gli effetti dei comandi axis square e image

- Visualizzazione delle superfici equipotenziali:
  imcontour(im)

ESEMPIO 1: caricare e visualizzare l'immagine contenuta nel file CT.dcm.

```
ct=dicomread('CT.dcm');
figure,imshow(ct,[])
```

Estrarre una sottomatrice che contenga la parte informative dell'immagine.

```
impixelinfo
ct_crop=ct(80:336,15:440);
figure,imshow(ct_crop,[])
```

NB. Nella ricerca di elementi all'interno di una matrice corrispondente ad un'immagine, il primo indice si riferisce alla riga, il secondo alla colonna: es A(1,3) estrae il valore del voxel che si trova su riga 1 e colonna 3. Per la funzione *impixelinfo* valgono le convenzioni opposte.

Per estrarre una porzione di immagine si può in alternativa utilizzare la funzione *imcrop* che attiva il cursore con il quale è possibile selezionare la regione di interesse sulla figura attiva.

```
close all, figure,imshow(ct,[]),
ct_crop=imcrop;
```

Sull'immagine ct originaria, porre il valore dei pixel della parte esterna che non è di interesse pari al valore minimo dell'immagine.

```
minimo=min(ct(:))
cornice=ones(size(ct));
cornice(80:336,15:440)=0;
ct(cornice==1)=minimo;
```

NB: operatori relazionali < <= > >= == ~=

ESEMPIO 2: caricare e visualizzare l'immagine contenuta nel file fant_pet.mat.

```
load fant_pet
figure,imshow(fant_pet,[]),colormap(jet),colorbar
figure,imcontour(fant_pet)
```

Per poter utilizzare la funzione mesh, occorre convertire l'immagine in double.

```
im=double(fant_pet);
figure,mesh(im)
```

NB: - Esistono funzioni specifiche del Toolbox di Image Processing per convertire le immagini da una classe ad un'altra: im2double, im2uint8, …

```
max(fant_pet(:))
im1=im2double(fant_pet);        max(im1(:))
im2=double(fant_pet);           max(im2(:))
im3=uint8(im1);                 max(im3(:))
```

im4=im2uint8(im1); *max(im4(:))*

ESEMPIO 3: caricare e visualizzare l'immagine contenuta nel file SPECT.jpg.

```
spect=imread('SPECT.jpg');
figure,imshow(spect,[])
```

Visualizzare separatamente la componente R dell'immagine RGB.

```
R=zeros(size(spect),'uint8');
R(:,:,1)=spect(:,:,1);
figure,imshow(R)
```

Convertire l'immagine da RGB a tipo intensità.

```
im=rgb2gray(spect);
figure,imshow(im,[])
```

NB: è possibile trasformare un'immagine da un tipo ad un altro, perdendo però in alcuni casi dell'informazione così da non poter compiere i passi a ritroso.

RGB=ind2rgb(X,MAP); *[X,MAP]=rgb2ind(RGB);*
I=rgb2gray(RGB);
[X,MAP]=gray2ind(I); *I=ind2gray(X,MAP);*

```
BW=im2bw(RGB,threshold);
BW=im2bw(X,MAP,threshold);
```

ESEMPIO 4: caricare e visualizzare l'immagine contenuta nel file PET.jpg.

```
pet=imread('PET.jpg');
figure,imshow(pet,[])
```

Provare a convertire l'immagine RGB in binaria variando la soglia.

```
pet=pet(5:220,20:270,:);
im1=im2bw(pet,0.2);figure,imshow(im1)
im2=im2bw(pet,0.5); figure,imshow(im2)
```

ESEMPIO 5: caricare le immagini contenute nel file volume_coeff_atten.mat

```
load volume_coeff_atten
size(vol)
```

Visualizzare una sezione assiale, una sezione coronale e una sezione sagittale dell'array 3d.

```
figure,imshow(vol(:,:,40),[]),colormap(jet)
figure,imshow(rot90(squeeze(vol(75,:,:)),-1),[]);colormap(jet)
figure,imshow(rot90(squeeze(vol(:,75,:)),-1),[]);colormap(jet)
```

NB: esistono funzioni in grado di processare sequenze di immagini (array 3d), cioè lo stesso tipo di trasformazione 2D viene applicato a tutte le immagini della sequenza (es. *imfilter* per il filtraggio delle immagini).

ESEMPIO 6: caricare le immagini contenute nel file mri.mat.

```
load mri
figure,imshow(D(:,:,25),map)
```

Visualizzare l'intera sequenza di immagini mediante le funzioni *montage* e *immovie*.

```
figure,montage(D,map)
mov=immovie(D,map);
figure, movie(mov,3)
```

NB: le funzioni *montage* e *immovie* per la visualizzazione di sequenze di immagini, richiedono che le immagini (frames) siano concatenate lungo la quarta dimensione (multiframe image arrays). Ogni frame deve avere le stesse dimensioni e, se è di tipo indicizzato, deve utilizzare la stessa colormap.

ESEMPIO 7: scrivere una funzione che carichi il set di immagini dicom contenuto nella cartella testa_collo e lo metta in un array 3d.

```
function volume=leggi_cartella_dicom;
dicomfiles=dir('testa_collo\*.dcm');
Nfiles=length(dicomfiles);
for i=1:Nfiles
   Y = dicomread(['testa_collo\',dicomfiles(i).name]);
   if i==1
      volume=Y;
   else
      volume=cat(3, volume, Y);
   end;
end;
```

Salvare il volume ottenuto in un file testa_collo.mat.

```
save testa_collo volume
```

Tra matrici valgono le **operazioni algebriche**, con attenzione alle dimensioni delle matrici in gioco; inoltre gli operatori **.*** **./** **.^** comportano l'applicazione dell'operazione algebrica tra i singoli elementi corrispondenti nelle 2 matrici (necessariamente delle stesse dimensioni).