

**MATLAB** mette a disposizione numerose funzioni, utili per lavorare con le immagini, molte di queste funzioni sono parte **dell'Image Toolbox**.

Una guida di questo Toolbox può essere reperita (gratuitamente) al link:

[http://www.mathworks.it/access/helpdesk/help/pdf\\_doc/images/images\\_tb.pdf](http://www.mathworks.it/access/helpdesk/help/pdf_doc/images/images_tb.pdf)

In Laboratorio considereremo solo le funzioni presenti nella versione base di MATLAB ma per vari concetti si può fare riferimento alla guida citata sopra.

Un ambiente di lavoro gratuito ed alternativo a MATLAB è **OCTAVE** (c'è una compatibilità molto alta tra sintassi e funzionalità dei due ambienti di Calcolo Scientifico)

**GNU Octave**



---

<http://www.gnu.org/software/octave/>

## Tipi di immagini

Esistono diverse modalità per memorizzare una immagine in MATLAB. Le principali sono le seguenti: **Binaria**, **Indicizzata**, **Scala di grigi**, **Truecolor**.

Image Type	Interpretation
Binary (Also known as a <i>bilevel</i> image)	Logical array containing only 0s and 1s, interpreted as black and white, respectively. See “Binary Images” on page 2-11 for more information.
Indexed (Also known as a <i>pseudocolor</i> image)	<p>Array of class <code>logical</code>, <code>uint8</code>, <code>uint16</code>, <code>single</code>, or <code>double</code> whose pixel values are direct indices into a colormap. The colormap is an <math>m</math>-by-3 array of class <code>double</code>.</p> <p>For <code>single</code> or <code>double</code> arrays, integer values range from <math>[1, p]</math>. For <code>logical</code>, <code>uint8</code>, or <code>uint16</code> arrays, values range from <math>[0, p-1]</math>. See “Indexed Images” on page 2-12 for more information.</p>

Image Type	Interpretation
Grayscale (Also known as an <i>intensity</i> , <i>gray scale</i> , or <i>gray level</i> image)	<p>Array of class <code>uint8</code>, <code>uint16</code>, <code>int16</code>, <code>single</code>, or <code>double</code> whose pixel values specify intensity values.</p> <p>For <code>single</code> or <code>double</code> arrays, values range from <math>[0, 1]</math>. For <code>uint8</code>, values range from <math>[0, 255]</math>. For <code>uint16</code>, values range from <math>[0, 65535]</math>. For <code>int16</code>, values range from <math>[-32768, 32767]</math>. See “Grayscale Images” on page 2-14 for more information.</p>
Truecolor (Also known as an <i>RGB</i> image )	<p><math>m</math>-by-<math>n</math>-by-3 array of class <code>uint8</code>, <code>uint16</code>, <code>single</code>, or <code>double</code> whose pixel values specify intensity values.</p> <p>For <code>single</code> or <code>double</code> arrays, values range from <math>[0, 1]</math>. For <code>uint8</code>, values range from <math>[0, 255]</math>. For <code>uint16</code>, values range from <math>[0, 65535]</math>. See “Truecolor Images” on page 2-15 for more information.</p>

Immagine binaria:

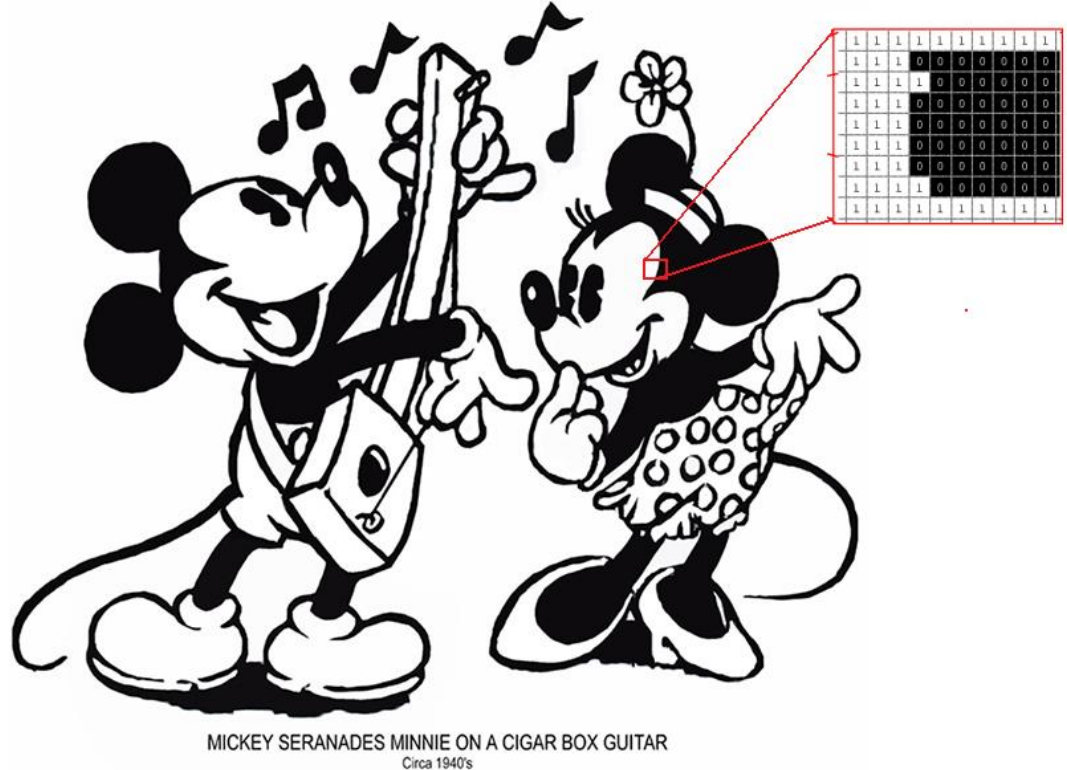
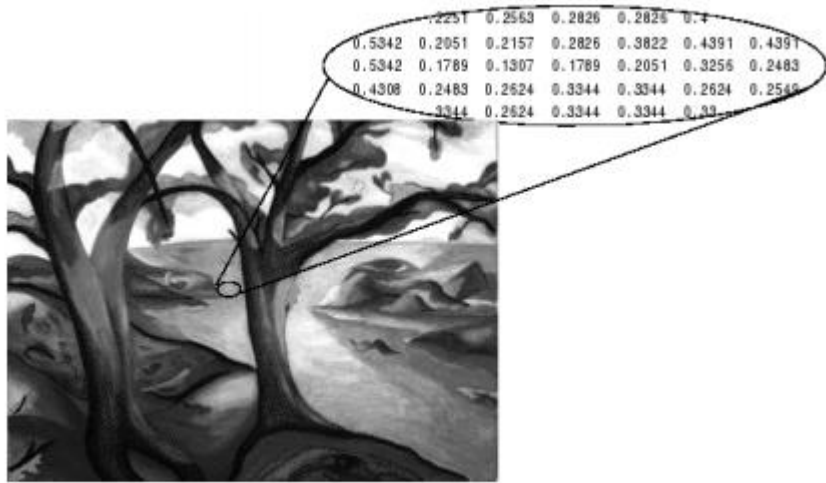


Immagine livelli di grigio:



Attenzione: tipo di dato uint8, unit16,... e double oppure int hanno codomini differenti

## Immagine colori:

(per ogni pixel tre valori)

		0.2235	0.1294	<b>Blue</b>	0.4190	
5804	0.2902	<b>0.0627</b>	0.2902	0.2902	0.4858	
5804	0.0627	0.0627	0.0627	0.2235	0.2588	
5176	0.1922	0.0627	<b>Green</b>	0.1922	0.2588	0.2588
5176	0.1294	<b>0.1608</b>	0.1294	0.1294	0.2588	0.2588
5176	0.1608	0.0627	0.1608	0.1922	0.2588	0.2588
5490	0.2235	0.5490	<b>Red</b>	0.7412	0.7765	0.7765
5490	0.3882	<b>0.5176</b>	0.5804	0.5804	0.7765	0.7765
5490	0.2588	0.2902	0.2588	0.2235	0.4824	0.2235
5490	0.2235	0.1608	0.2588	0.2588	0.1608	0.2588
5490	0.2588	0.1608	0.2588	0.2588	0.2588	0.2588



## Immagine Indexed:

Abbiamo due matrici:

- una matrice di valori per ogni pixel;
- una matrice colormap con la “mappa dei colori”

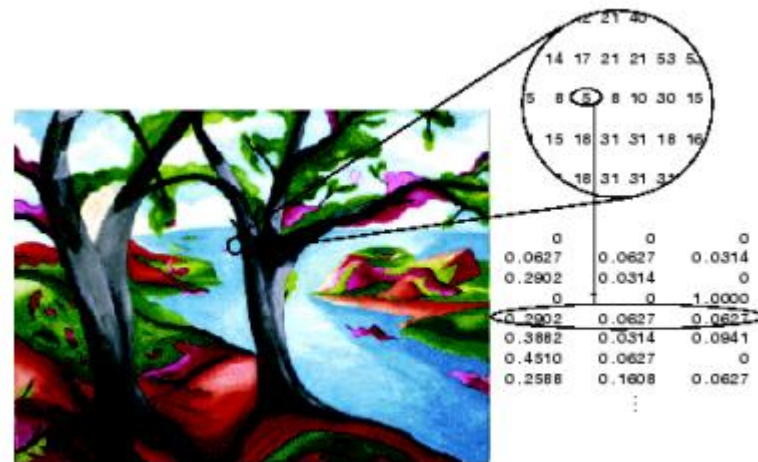


Image Courtesy of Susan Cohen

Immagine indexed:

11	23	5	7	19	....	13
7	12	14	1	120	....	125
100	87	81	10	12	....	91
....	....	....	....	....	....	....
....	....	....	....	....	....	....
14	101	19	23	10	....	100

Matrice pixel immagine

	....	
	....	
	....	
0.26	0.87	0.12
	....	
	....	
	....	
	....	

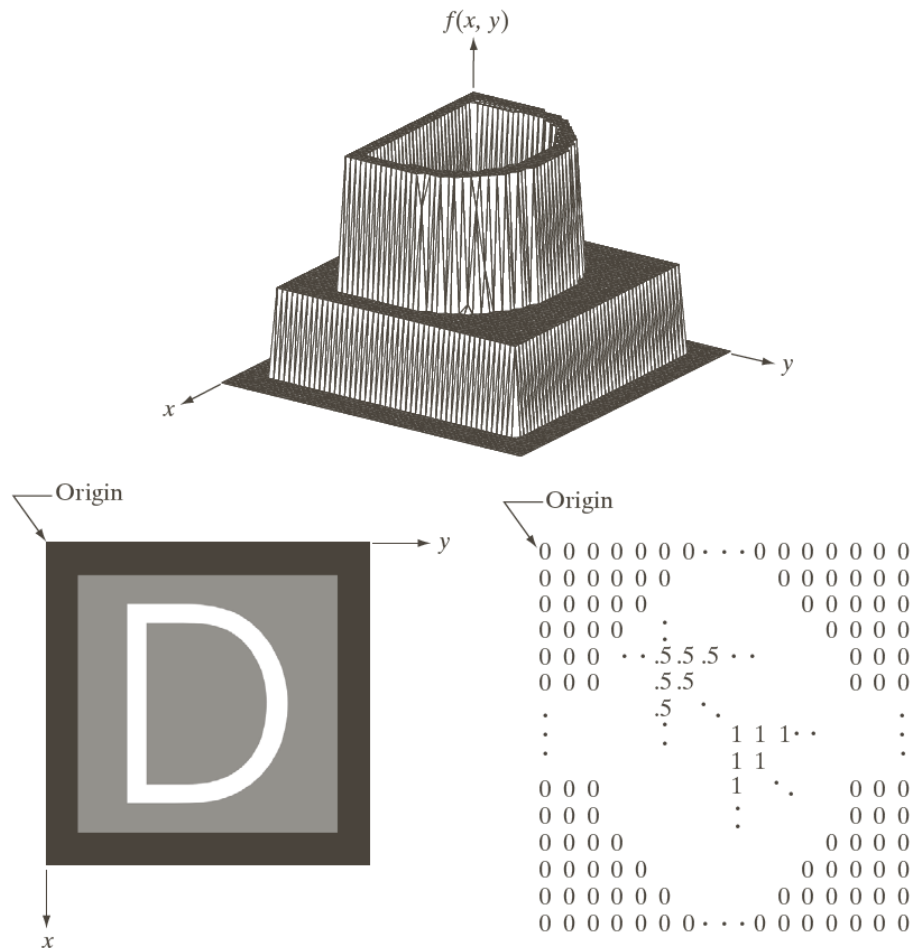
mapa dei colori

....	....	....	....
....	....	....	....
			....

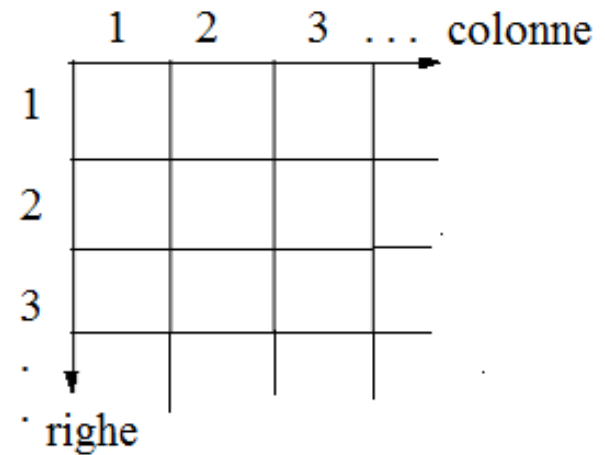
immagine

**Attenzione: indici unit8,... oppure double partono rispettivamente da 0 o da 1; se esco dall'intervallo degli indici permessi si prende il primo o l'ultimo colore.**

Nota: per ora considereremo le immagini digitali ricavate come campionamento di una immagine sottostante, stiamo quindi pensando alle immagini come funzioni:



Matrice dei pixel, l'origine è il vertice in alto a sinistra, variando le colonne mi muovo lungo l'asse delle x, variando le righe mi muovo lungo l'asse delle y.



Per leggere un file contenente una immagine in scala di grigi o a colori, MATLAB mette a disposizione il comando **imread**

```
>> I = imread(filename, fmt);  
>> [I,map] = imread(...);  
>> [I,map] = imread(filename);  
>> [I,map] = imread(URL,...);
```

Viene letta un'immagine di formato 'fmt' e di nome (e percorso) specificato da 'filename' o caricata dal web dall'indirizzo specificato da 'URL' e viene restituita una matrice I di dimensioni M x N se a scala di grigio (gli M x N pixels delle M righe e N colonne), oppure di dimensioni M x N x 3 se a colori rappresentati con il modello RGB. Gli elementi di questa matrice sono usualmente della classe uint8, cioè possono assumere valori da 0 a 255. Legge anche una immagine indicizzata, restituendo la relativa mappa di colore nella variabile map.

E' possibile ottenere le informazioni su un file di immagine in una struttura info, tramite il comando **imfinfo**

```
>> info = imfinfo(filename,fmt);  
>> info = imfinfo(filename);  
>> info = imfinfo(URL,...);
```

Per scrivere un file contenete una immagine in scala di grigi o a colori, MATLAB mette a disposizione il comando **imwrite**

```
>> imwrite(I,filename,fmt);  
>> imwrite(I,map,filename,fmt);
```

Tale comando salva sul disco un file di nome 'filename' e di formato specificato da 'fmt' contenente l'immagine rappresentata dalla matrice I. La matrice I può avere dimensioni M x N per immagini in scala di grigi, oppure dimensioni M x N x 3 per immagini a colori, rappresentata dal modello RGB. La seconda modalità salva una immagine indicizzata con la relativa mappa di colore, specificata dalla variabile map.



## MATLAB: visualizzare una immagine

Un modo per visualizzare una immagine in scala di grigi o a colori contenuta in una matrice A (quindi trattata come un'immagine), è quello di utilizzare il comando **image**

```
>> image(A);  
>> image(x,y,A);  
>> handle = image(...);
```

Questo comando visualizza il contenuto della matrice A come un'immagine. Posso anche specificare in due vettori x e y gli intervalli degli assi da visualizzare, la funzione image può restituire l'handle della figura.

```
>> [I,map]=imread('forest.tif');  
>> image(I);  
>> colormap(map);  
>> axis off  
>> axis image
```



## MATLAB: visualizzare una immagine

Un ulteriore modo alternativo per visualizzare una immagine in scala di grigi o a colori contenuta in una matrice *A* (quindi trattata come un'immagine), è quello di utilizzare il comando **imagesc**

```
>> imagesc(A);  
>> imagesc(x,y,A);  
>> imagesc(...,clims);  
>> handle = imagesc(...);
```

E' equivalente all'utilizzo di `image`, tranne il fatto che l'immagine ottenuta è visualizzata utilizzando tutta la mappa di colore. Questo comando visualizza il contenuto della matrice *A* come un'immagine. Posso anche specificare in due vettori *x* e *y* gli intervalli degli assi da visualizzare, ma l'immagine è la stessa della chiamata precedente. La funzione `image` restituisce l'handle della figura. Posso utilizzare anche una parte della mappa di colore compresa nell'intervallo `clims = [low high]`: i pixel con valore minore di `low` vengono visualizzati con il primo elemento della mappa di colore, quelli con valore maggiore di `high` con l'ultimo elemento della mappa di colore.

```
>> clims = [10 60];
```

```
>> I=imread('cameraman.tif');
```

```
>> subplot(1,2,1); imagesc(I);colormap gray
```

```
>> subplot(1,2,2); imagesc(I,clims); colormap gray
```



## MATLAB: la mappa di colore

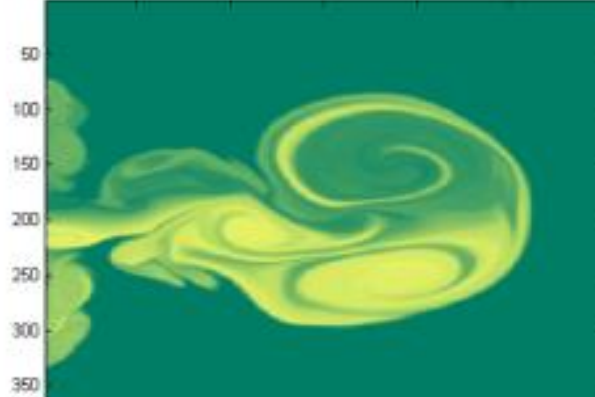
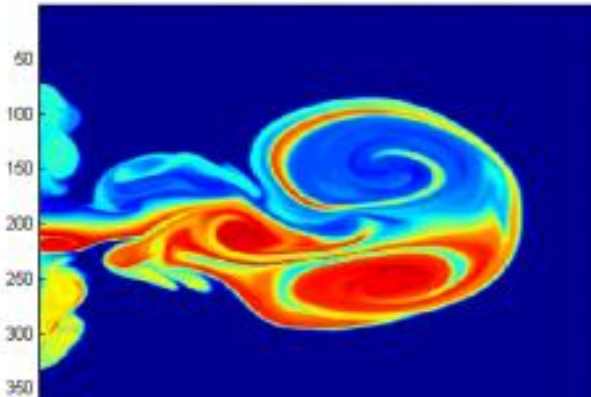
Una mappa di colore è una matrice di dimensione  $M \times 3$  di valori compresi nell'intervallo  $[0 \ 1]$ , che specifica i valori RGB degli  $M$  colori predeniti. In particolare, la riga  $k$ -esima della mappa di colore è la terna  $[r(k) \ g(k) \ b(k)]$  del  $k$ -esimo colore.

MATLAB mette a disposizione 17 mappe di colore predefinite,

```
>> colormap(map);  
>> colormap('default');  
>> cmap = colormap;
```

Il primo comando consente di utilizzare la mappa di colore specificata da `map`, il secondo comando consente di utilizzare la mappa corrente come default, mentre il terzo restituisce la mappa di colore nella matrice `cmap`.

```
>> load flujet; image(X), colormap(jet)  
>> image(X), colormap(summer)  
>> load spine; image(X), colormap(bone)
```



# Matlab: la *mappa di colore*

Le 17 **mappe di colore** disponibili in Matlab sono elencate qui sotto ed illustrate a fianco:

- `autumn;`
- `bone;`
- `colorcube;`
- `cool;`
- `copper;`
- `flag;`
- `gray;`
- `hot;`
- `hsv;`
- `jet;`
- `lines;`
- `pink;`
- `prism;`
- `spring;`
- `summer;`
- `white;`
- `winter.`



## Esercizio: cambiare luminosità

Scrivere un M-file che riceva una immagine e modifichi la luminosità alterando la mappa di colore che viene utilizzata. Ogni elemento della mappa di colore map viene elevato ad un esponente  $\gamma$  pari a

$$\gamma = \begin{cases} 1 - \beta & \text{se } \beta > 0 \\ \frac{1}{1 + \beta} & \text{se } \beta \leq 0 \end{cases}$$

Dove  $\beta$  è un parametro in ingresso. In particolare la mappa di colore è «schiarita» se  $0 < \beta < 1$  oppure «scurita» se  $-1 < \beta < 0$ . Cosa succede se applico una volta con  $\beta > 0$  e di seguito con  $-\beta$ ?

## Esercizio: equalizzazione dell'istogramma

Scrivere un M-file che realizzi l'equalizzazione dell'istogramma attraverso la seguente trasformata puntuale, con  $m$  parametro reale positivo,

$$Y(X) = 255 \frac{\sum_{i=0}^X [H(i)]^m}{\sum_{i=0}^{255} [H(i)]^m}$$

## Esercizio: selezionare aree differenti

Progettare e realizzare un metodo per l'estrazione di zone differenti di una Immagine utilizzando una opportuna immagine binaria come “maschera”  
Fare equalizzazione della parte selezionata.