Try the following:
>>Al=[1 0 1;0 1 0;1 0 1];
>>Bl=[0 1 0;1 0 1;0 1 0];
>>Cl=Al&Bl; Dl=Al|Bl; El=xor(Al,Bl); any(Al); all(Bl);

Searchin elements in the array:

Using ***find*** it is possible to search for those elements that satisfy the specified condition. The result will be 2 arrays with the row and column indices of the corresponding elements.

A=[1 2 3;4 5 6;7 8 9]

>>[i,j]=***find***(A>6);

i = 3	j = 1
3	2
3	3

By default, it search for nonzero elements.

Try these instructions. Change the condition and see the results.

Caiani 16/3/2012

"Sparse" matrix

sparse(A) transforms the matrix A in a "sparse" matrix, thus eliminating nul elements.

>>S=***sparse***(A);

full(S) performs the opposite operation, retransforming the "sparse" matrix into the original one, with zero elements (if any) :

>>B=***full***(S);

Define a 3x3 matrix with the central element (2,2) non zero, and the others zeros. Apply the sparse and full commands, observing their result into the workspace.

Caiani 16/3/2012

for: it is utilized to repeat for a predetermined number of times a series of commands.

>>**for** *counter=start:increment:end*,

commands

end

warning: never modify the counter value inside the cycle!!!

To iteratively create the array's elements of known dimensions, it is better to first initialize the array, to speed up the execution time.

Try to avoid cycles to speed up your code, using the array computaional power of Matlab

while: it repeats a series of commands for an indefinite number of times, until the condition stays verified:.

>>**while** *condition*,

commands

end

Caiani 16/3/2012

Write the code that :

For N from 1 to 100, computea 2^N, putting the results into a row vector.

Set T=200;
Until T>100, subtract 1 from T

Repeat the first exercise using vectors properties and not using ***for***

Caiani 16/3/2012

if else end CONDITION

It executes a series of commands if the relational test is satisfied:

```
>>if test
        commands
   end
```

```
>> if test
        command1
   else
        command2
   end
```

```
>> if test1
        command1
   elseif test2
        command2
   end
```

Caiani 16/3/2012

switch case CONSTRUCTION

When a series of commands have to be executed based on the value of a variable, this is more effective than multiple if statements:

```
>> switch variabile
        case {test1}
                commands
        case {test2}
                commands
        otherwise
                commands
    end
```

Caiani 16/3/2012

Operations with chars

A variable containing chars is a **string**: it is defined as char array.

>> stringa='This is a good example'

strfind searches for the correspondence between chars in the string (result=position in the string):

>>**strfind**(stringa,'a')

strrep substitues a string with another one:

>>**strrep**(stringa,'good','great')

strcmp comparetwo strings (1 if identical):

>>s='example'

>> **strcmp**(stringa,s)

eval executes the string, interpreting it as a command:

>>s='a=zeros(2)'

>> **eval**(s)

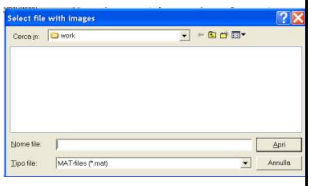
Try those examples.

Caiani 16/3/2012

User Interface Guide: Some commands activate input/output boxes:

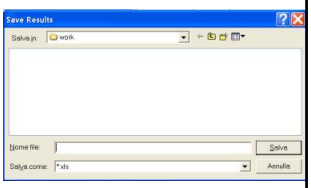
uigetfile activates a box to load a file:

>>[nomefile,pathfile]=**uigetfile**('*.mat','Select file with images');



uiputfile activates a box to save a file:

>>[FILENAME,PATHNAME]=**uiputfile**('*.mat','Save File')



load/save[PATHNAME '\' FILENAME]

questdlg activates a box for replying to a question:

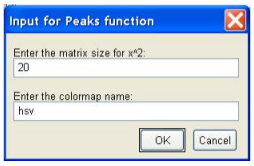
>>ButtonName=**questdlg**('Save results ?','SAVE','Yes');

Try those examples.

Caiani 16/3/2012

inputdlg activates a box for data entry:

```
>>prompt={'Enter the matrix size for x^2:','Enter the colormap name:'};
>>name='Input for Peaks function';
>>numlines=1;
>>defaultanswer={'20','hsv'};
>>answer=inputdlg(prompt,name,numlines,defaultanswer);
```



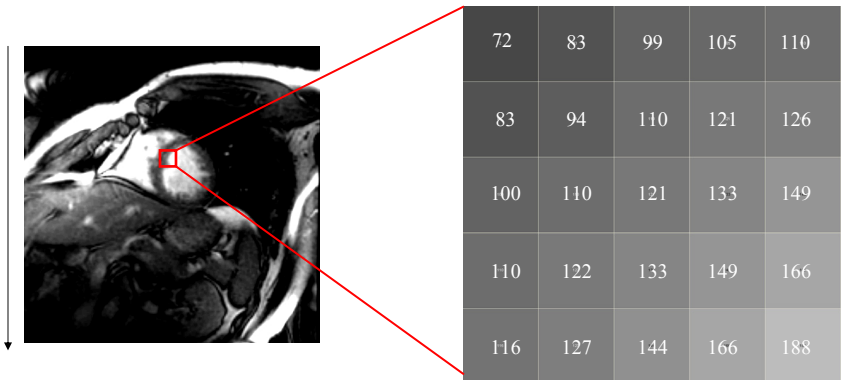
N.B.: the content of the inpudlg fields is considered as cell array. To use it as number, it is necessary to convert it first as a string, by the command ***char***, and then by converting the string to a number by ***str2num***:

```
>>ris=char(answer(1,1));
>>risultato=str2num(ris);
```

Try those examples and check the variables classes.

Caiani 16/3/2012

What is a digital biomedical image?



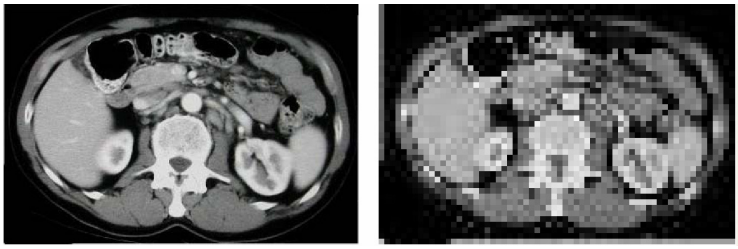
A digital image is a discrete function f(x,y) defined over a rectangular grid, representing the characteristics of the object being imaged, as a result of the sampling and quantization operations. Both x,y (spatial coordinates) and f(x,y) are all finite and discrete quantities. Each grid element (pixel) is defined by its location (x,y) and its value (intensity).

Caiani 16/3/2012

Image acquisition → Spatial resolution

The intensity value for each pixel is the result of the acquisition of the physical entity at the base of the generation of the image and its digitization, and it is equal to the mean value of the analog data in the space portion covered by each pixel in the sampling grid.

Denser sampling grid allow to obtain higher spatial resolution.



In general, spatial resolution is anisotropic:

$\Delta x \, \Delta y \, \Delta z$

Caiani 16/3/2012

Types of bioimedical images

A possible classification of the biomedical images is related to the physical quantities represented into the image, and the form of energy utilized to create the image.

A possible classification is the following:

Ionizing radiations, attenuation - X photons for **RX** and computerized tomography (CT), with energy around energia 20-100 keV (Hounsfield units (-1000 : 1000), with air = -1000 and distilled water = 0.

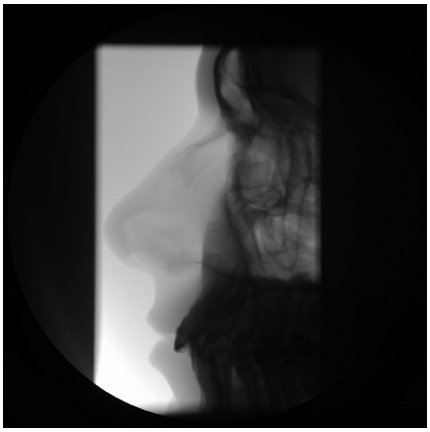
Ionizing radiations, emission - γ photons, emitted by radionuclides with large spectrum energy around 100 keV for **scintigraphy** and tomographic **SPECT** (single photon emission computerized tomography); couples of photons (512 keV) emitted in opposite directions follwoing positron emission in tomographic **PET** (positron emission tomography); radioactivity level measures the internal **concentration of tracing molecules** marked by radionuclides (Becquerel/millimeter (Bq/ml))

Caiani 16/3/2012

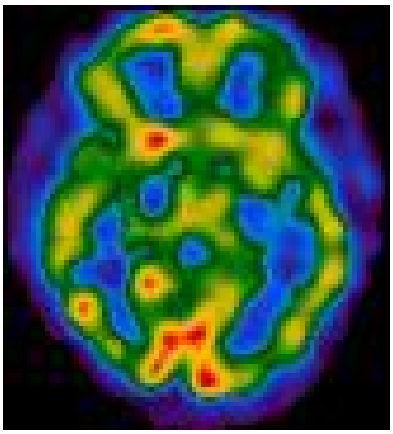
•Take into account the **Nyquist sampling theorem** (in space): If a function $x(t)$ contains no frequencies higher than B hertz, it is completely determined by giving its ordinates at a series of points spaced $1/(2B)$ seconds apart, or in other words, there has to be at least two pixels to represent the maximal spatial frequency contained in the image

Based on the intrinsic properties of the continuous image, the sampling frequency is determined: high resolution images (RX) require denser sampling grids (over 1000 x 1000 pixel); low resolution images (SPECT, PET) require less number of pixels for the same field of view.

Caiani 16/3/2012



High spatial resolution



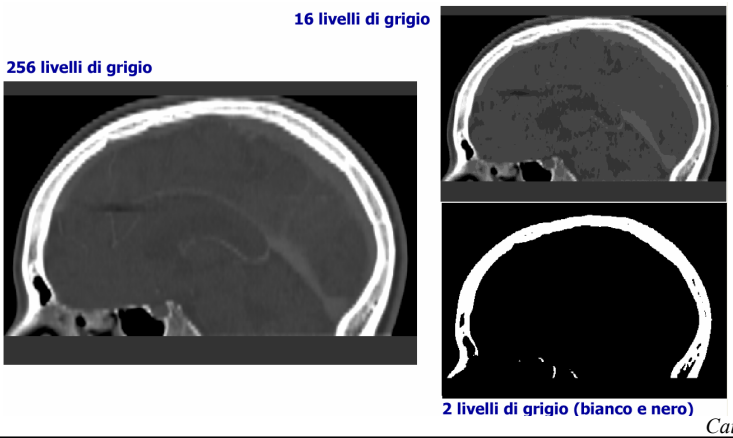
Low spatial resolution

Caiani 16/3/2012

Quantization

The pixel intensity level has to be digitalized into a finite range of values. The number of bits available for this operation is important in order to discriminate structures with similar values.

intensity levels= 2^n

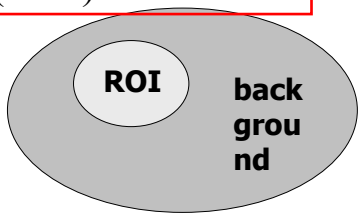


Caiani 16/3/2012

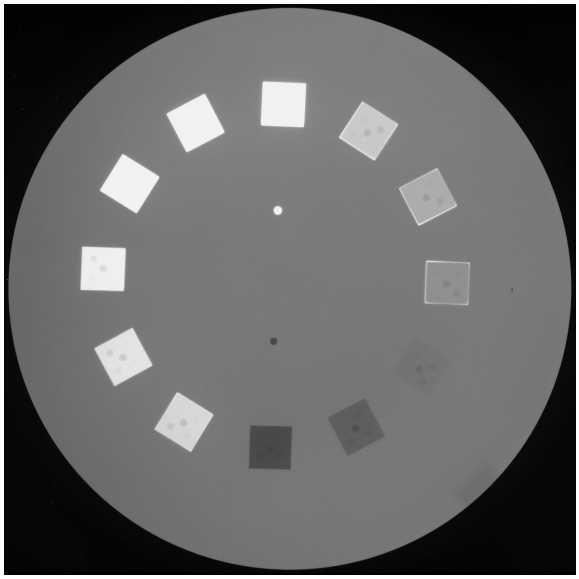
Image contrast

- The main desire in image processing is that the objects/regions of interest (ROI) present in the image are characterised by values of intensity that allow their differentiation from the other image portions (background).

$$contrast = \frac{mean(ROI) - mean(background)}{mean(ROI)}$$



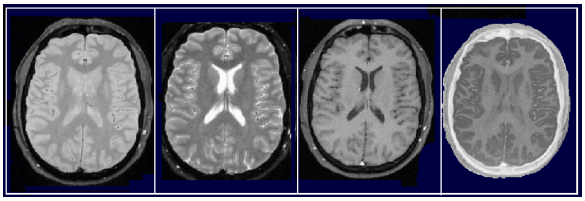
Caiani 16/3/2012



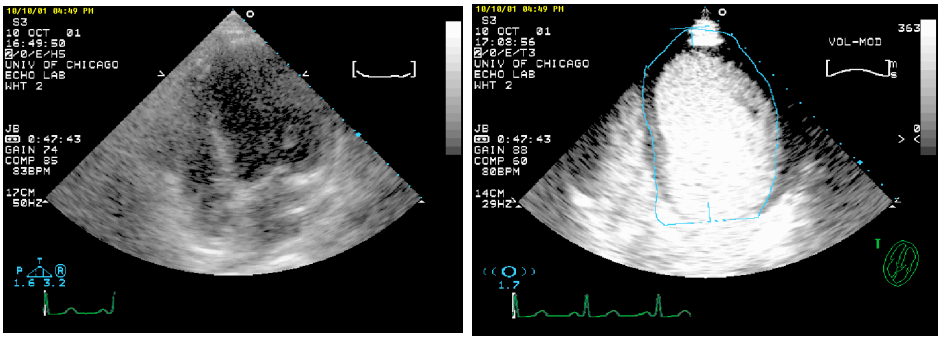
Caiani 16/3/2012

- In the image acquisition process, every effort is devoted to increase the contrast between pathological and normal structures.
- To this aim, different acquisition techniques based on different physical parameters, as well as on utilization of contrast media, can be utilized.

In MRI, different contrast is created based on the different T1 and T2 relaxation times;
Contrast media based on Barium or Iodium are used in RX, Gadolinium in MRI, microbubbles in US, complex tracers in nuclear medicine.



Caiani 16/3/2012



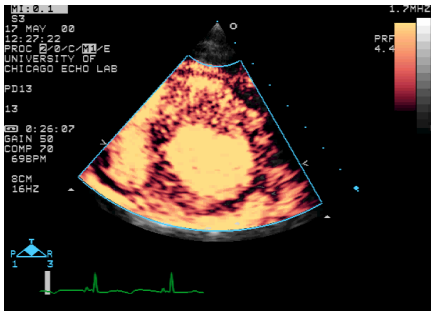
Caiani 16/3/2012

Acquisition time

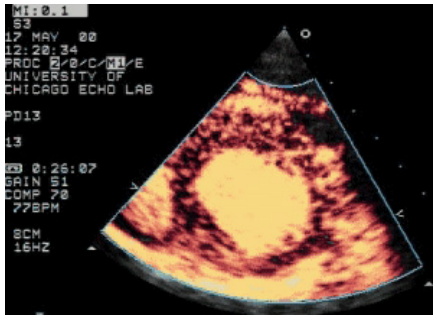
Shortening of the acquisition time is important for several reasons:

1) Reduction of movement artifacts, e.g. breathold for the time of acquisition;
2) Reduction of the total diagnostic time; usally, the exam results in the acquisition of multiple images (e.g. tomographic CT or MRI sessions) and the total time is multiple of the time needed for a single acquisition (CT requires about 1 sec, MRI requires 10-15 sec)
3) In dynamic acquisitions (same image over time), is important to increase the **temporal resolution** (e.g. US gives about 20-40 frame/sec; MRI about 30 frames/cardiac cycle)

Caiani 16/3/2012



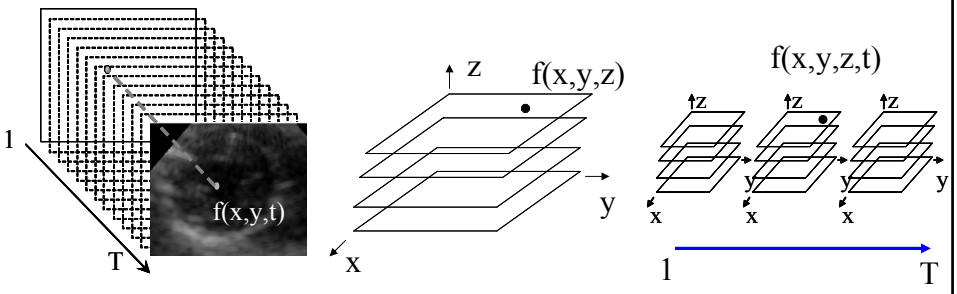
High temporal resolution



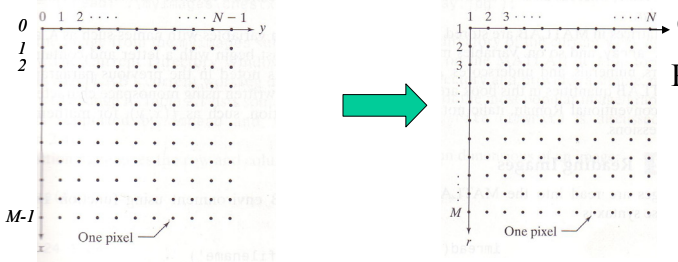
Low temporal resolution

Caiani 16/3/2012

2D+t **3D** **3D+t**



Digital image → Array of real numbers of size M rows x N columns



Pixel coordinates
Origin in (1,1)

Caiani 16/3/2012

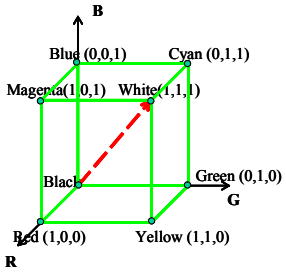
Image intensity → unsigned integer (*uint*) at 8 or 16 bits, *double* (64 bits), *vector* [a b c]

According to how intensity is represented, images can be of 4 types:

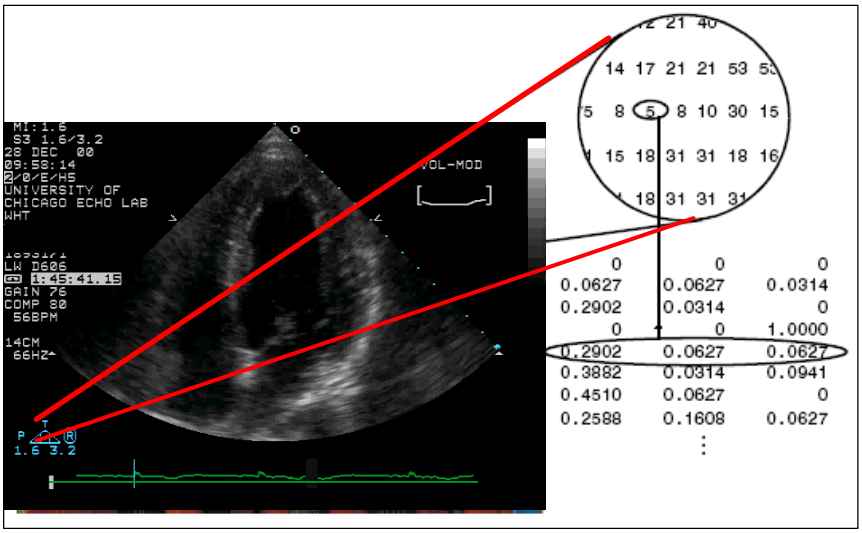
• INDEXED IMAGE

Constituted by **2 arrays**: the image array and the colormap (w x 3 columns)

In the image array, each value (integer) represents the pointer to the respective line of the colormap, where the intensity is defined. In the colormap, the three columns define the R,G,B component of the pixel intensity.



Caiani 16/3/2012



Caiani 16/3/2012

Image format http://www.fileinfo.com/filetypes/raster_image

Most images you see on your computer screen are **raster graphics**. They are made up of grid of pixels, commonly referred to as a bitmap. The larger the image, the more disk space the image file will take up. For example, a 640 x 480 image requires information to be stored for 307,200 pixels.
Since raster graphics need to store so much information, large bitmaps require large file sizes. Fortunately, there are several image compression algorithms that have been developed to help reduce these file sizes. JPEG and GIF are the most common compressed image formats on the Web, but several other types of image compression are available.

Raster graphics can typically be scaled down with no loss of quality, but enlarging a bitmap image causes it to look blocky and "pixelated." For this reason, **vector graphics** are often used for certain images which need to be scaled to different sizes.

File extensions: .BMP, .JPG, .RAW, .TIF, .GIF,

Caiani 16/3/2012

Compression algorithms

"Lossless"	"Loss"
- Original and compressed are = - Redundancy elimitated - compression ratio 3:1 - run-length encoding, Huffman encoding	- High compression rate, but irreversible alterations in the original images - JPEG2000

Caiani 16/3/2012

BMP (BIT-MAP)

Uncompressed raster images containing a file header (bitmap identifier, file size, width, height, color options, and bitmap data starting point) and bitmap pixels, each with a different color. BMP files may contain different levels of color depths per pixel (RGB), depending on the number of bits per pixel specified in the file header. They may also be stored using a grayscale color scheme, but do not support the Alpha channel (transparency).
Windows uses for BMP files an unchangeble colormap.

RAW

The RAW format is the most flexible to transfer data between different platforms.
It is constituted by a sequence of bytes describing the color info for each pixel position.

Caiani 16/3/2012

JPG

Compressed image format standardized by the Joint Photographic Experts Group (JPEG); commonly used for storing digital photos since the format supports up to 24-bit color; therefore, most digital cameras save images as JPG files by default.

JPEG is also a common format for publishing Web graphics since the JPEG compression algorithm significantly reduces the file size of images. However, the lossy compression used by JPEG may noticeably reduce the image quality if high amounts of compression are used.

TIFF (Tagged-Image file Format)

Graphics container that can store both raster and vector images; may contain high-quality graphics that support color depths from 1 to 24-bit; supports both lossy and lossless (Lempel-Ziv-Welch algorithm) compression; also supports multiple layers and pages.
It supports the Alpha channel, and binary, gray, and color images.

Caiani 16/3/2012

GIF (Graphics Interchange Format)

Image file in lossless (LZW) format that may contain up to 256 indexed colors; color palette may be a predefined set of colors or may be adapted to the colors in the image. GIFs are common format for Web graphics, especially small images and images that contain text (interleaved data). However, JPEG images are better for showing photos because they are not limited in the number of colors they can display. GIF images can also be animated and saved as "animated GIFs," which are often used to display basic animations on websites. Pixels in a GIF image must be either fully transparent or fully opaque (no transparency).

PNG (Portable Network Graphic)

Like GIF, it contains a bitmap of indexed colors (also 24 or 48 bit colors) under a lossless compression, but without copyright limitations of LZW; commonly used to store graphics for Web images. It compresses up to 30% more than TIFF LZW at 24 bit, and from 10% to 30% more than GIF at 8 bits. Additionally, while GIF images only support fully opaque or fully transparent pixels, PNG images may include an 8-bit transparency channel, which allows the image colors to fade from opaque to transparent. PNG images cannot be animated like GIF images. However, the related .MNG format can be animated.

Caiani 16/3/2012

Unlike JPEGs, GIFs, and BMP images, **vector graphics** are not made up of a grid of pixels. Instead, vector graphics are comprised of paths, which are defined by a start and end point, along with other points, curves, and angles along the way. A path can be a line, a square, a triangle, or a curvy shape. These paths can be used to create simple drawings or complex diagrams. Paths are even used to define the characters of specific typeface (font).

Because vector-based images are not made up of a specific number of dots, they can be scaled to a larger size and not lose any image quality. When you blow up a vector graphic, the edges of each object within the graphic stay smooth and clean. This makes vector graphics ideal for logos, which can be small enough to appear on a business card, but can also be scaled to fill a billboard. Common types of vector graphics include Adobe Illustrator, Macromedia Freehand, and EPS files. Many Flash animations also use vector graphics, since they scale better and typically take up less space than bitmap images.

File extensions: .AI, .EPS, .SVG, .DRW

Caiani 16/3/2012

LOADING AN IMAGE

If the image is in .mat format:

*>>**load** path\name*

If the image is in a different format:

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.

*>>[nomeimmagine,mappa]=**imread**('path','extension');*

If it is not an indexed image, *mappa* will be empty.

>>INFO = **imfinfo**(FILENAME,FMT) returns a structure with information about an image in a graphics file.

Check the info and load the echographic image:
>>load c:\ .... \ecoindex
You have X and MAP in the workspace; take a look to the class of these variables.

Caiani 16/3/2012

VISUALIZING AN IMAGE

Use:

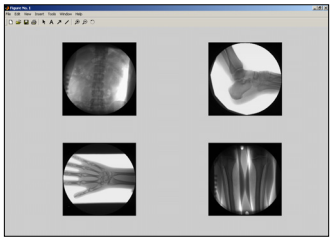
imshow(X,liv) ***imshow***(X,[low high]) ***imshow***(X,[ ])

liv: videointensity levels used in the visualization (default=256)

To visualize more images on the same figure, use ***imshow*** and ***subplot***: it divides the figure in *mxn* regions, making active the selected region:

subplot(1,2,1),***imshow***(I);

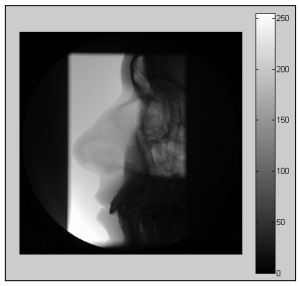
subplot(1,2,2),***imshow***(J)



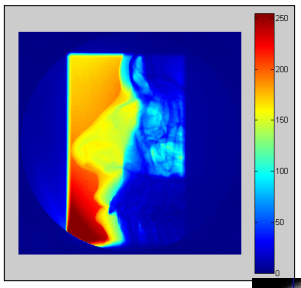
Visualize the previously loaded image, and different frames into the same figure.

Caiani 16/3/2012

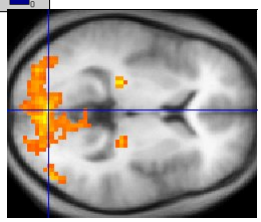
This image representation can be sometimes useful to represent the original image into user-defined pseudocolors, to better evidence possible details



colormap(gray)

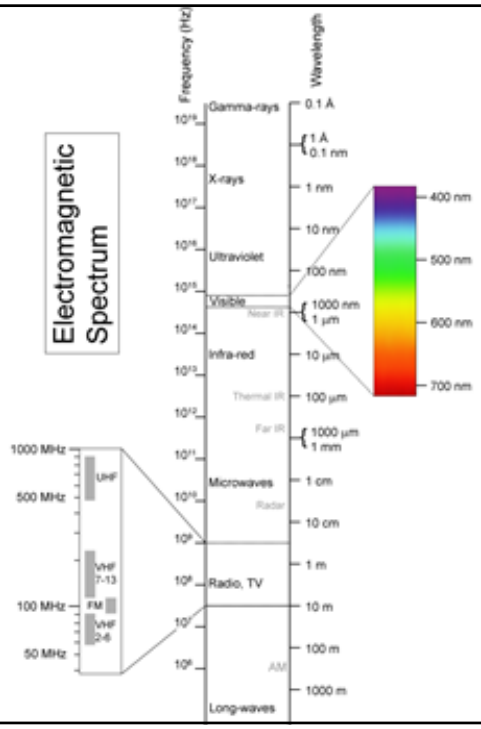


colormap(jet)



or to represent parametric images obtained by post-processing (i.e., fMRI).

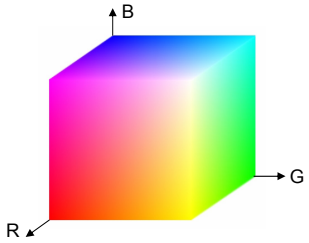
Caiani 16/3/2012

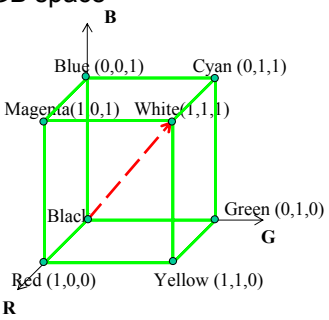


Caiani 16/3/2012

Colorspace - RGB

- Matlab represents colors into RGB space





R *G* *B*

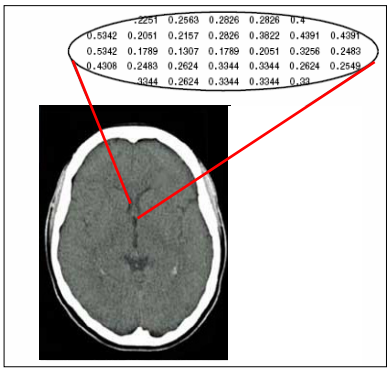
Caiani 16/3/2012

Define an image as the union of three arrays, each of 128 lines and 64 columns.
Assign a label to each of the elements of each array (1,2,3) and create a colormap with the colors (green, white, red).
Visualize the image using the created colormap.

Caiani 16/3/2012

• INTENSITY IMAGE

Constituted by the image array only: the pixel intensity describes the corresponding gray level (*double*: [0÷1], *uint8:* [0 ÷ 255], *uint16:* [0 65535])



MRI, CT, Rx

To transform a uint8 into double, or viceversa:

A=**im2uint8**(B);

B=**im2double**(A);

C=**im2uint16**(B);

They automatically perform the needed arrangements.

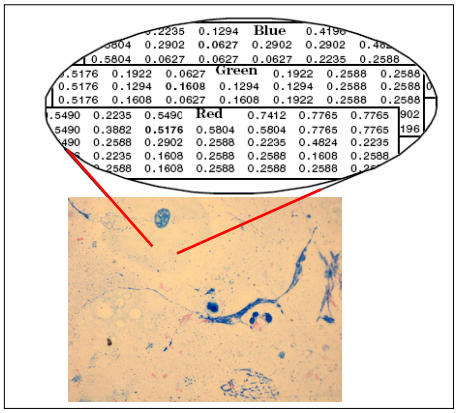
Load and visualize 'volto.jpg', change the format and see the effect on the image content and class

Caiani 16/3/2012

• RGB IMAGE

For each pixel position (x,y), the intensity is defined by an vector of 3 elements, defining the RGB components.

Thus, the image array is of dimensions M x N x 3.



Load and visualize 'spect.jpg', and check the corresponding variables in the workspace.

Caiani 16/3/2012

• BINARY IMAGE

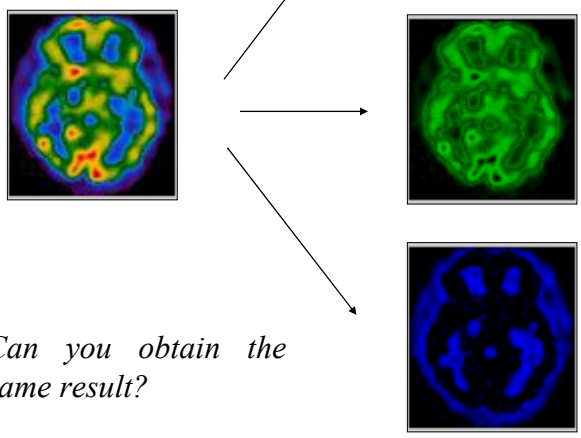
Constituted by the image array only, where each elemente can be only 0 (off) or 1 (on).

Every function resulting in a binary image, defines it as array *uint8*

Load and visualize 'mask.mat', that contain the variabile MASKOR representing a LV cavity from an apical echocardiographic view. Check its content and class. Visualize the negative image, using ***imshow****(~name), o change the binary colors:* ***imshow****(name,[1 0 0;0 0 1])*

Caiani 16/3/2012

Example



Can you obtain the same result?

Caiani 16/3/2012

Image type conversion

indexed ⟺ RGB

[X,MAP]=***rgb2ind***(RGB);

RGB=***ind2rgb(***X,MAP);

indexed ⟺ intensity ⟹ binary

I=***ind2gray***(X,MAP);

[X,MAP]=***gray2ind***(I);

BW=***im2bw***(X,MAP,threshold);

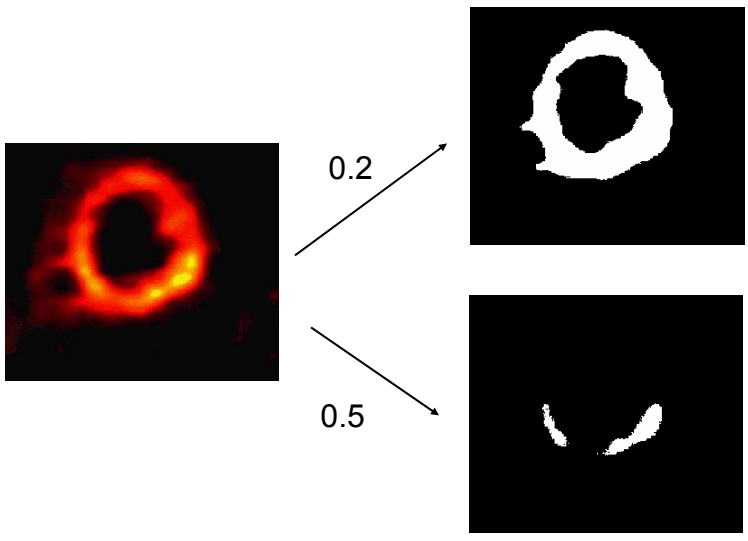
RGB ⟹ intensity ⟹ binary

I=***rgb2gray***(RGB);

BW=***im2bw***(RGB,threshold);

*I=0.2989 * R + 0.5870 * G + 0.1140 * B*

Caiani 16/3/2012



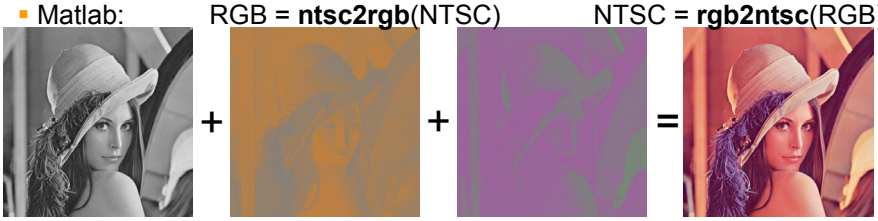
Caiani 16/3/2012

YIQ - colorspace

- NTSC (National Television System Committee)
- The pixel intensity (luminance *Y*) is separated by the color information, which is described by the crominance I (in-phase) and Q (quadrature) matrices. The YIQ system is intended to take advantage of human color-response characteristics. The eye is more sensitive to changes in the orange-blue (I) range than in the purple-green range (Q) — therefore less bandwidth transmission is required for Q than for I.

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} .299 & .587 & .114 \\ .596 & -.274 & -.321 \\ .211 & -.523 & .311 \end{bmatrix} \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

- Matlab: RGB = **ntsc2rgb**(NTSC) NTSC = **rgb2ntsc**(RGB)

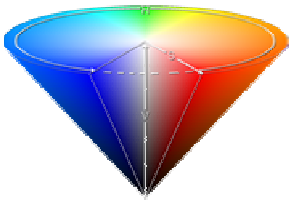


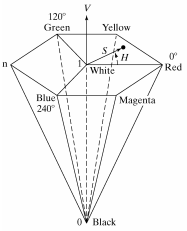
Y *I* *Q*

Caiani 16/3/2012

HSV colorspace

- Used in computer graphic to select colors from color picker
- Based on a cylindric coordinate system, where the 3 matrices represent the hue (H), saturation (S, from pure hue to white saturated) and value (V, from black to white).





- Matlab: HSV = **rgb2hsv**(RGB) RGB = **hsv2rgb**(HSV)

H *S* *V*

Convert a RGB image into the NTSC and HSV colorspaces, and see the effects.

Caiani 16/3/2012

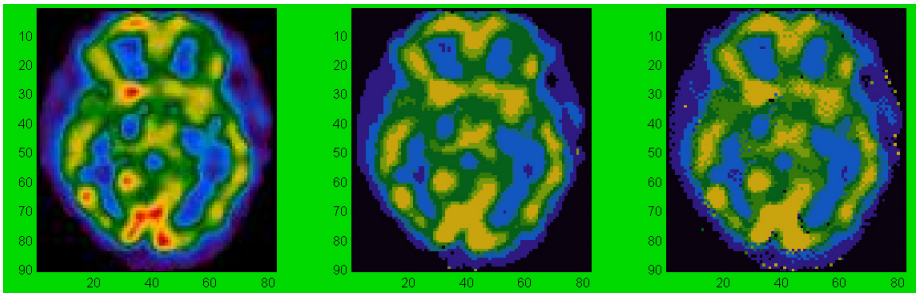
Using **whitebg** it is possible to change the background default color:

>>**whitebg**('g')

Long name	Short name	RGB values
Black	k	[0 0 0]
Blue	b	[0 0 1]
Green	g	[0 1 0]
Cyan	c	[0 1 1]
Red	r	[1 0 0]
Magenta	m	[1 0 1]
Yellow	y	[1 1 0]
White	w	[1 1 1]

Using **dither** it is possible to reduce the colors in an image, in conjunction with **rgb2ind**, by maintaining artificiously a better image quantization effect:

originale I [X1,map1]=**rgb2ind**(I,8,**'nodither'**) [X2,map2]=**rgb2ind**(I,8,**'dither'**)



Caiani 16/3/2012

Images with dithering present configurations of granular pixels. The human eye, for a certain distance, sees these configurations as an intermediate color (or videointensity) in respect of those of the pixels that compose them.

Dithering algorithm of **Floyd-Steinberg,** based on error dispersion:

a) look pixel (i,j)

b) change its intensity with the value at minimal distance in respect of the available map

c) compute the difference between original and new value

d) distribute this error on the neighbour pixels (see table) by summing it to their

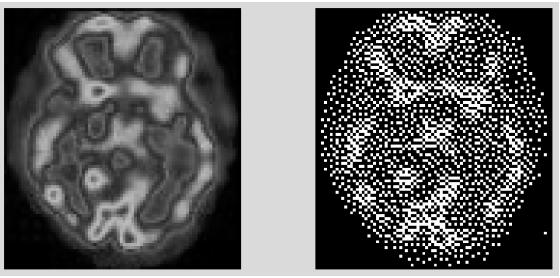
e) Repeat for the next pixel

0	Pixel attuale	7/16
3/16	5/16	1/16

Caiani 16/3/2012

If applied to an intensity image **dither** will convert it into binary, but providing a particular optical effect.

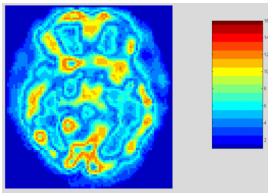
L=**rgb2gray(**I) L1=**dither**(L)



With **grayslice, a** thresholding of the intensity image is performed, with n thresholds equal to 1/n, 2/n, …, (n-1)/n:

>>X=**grayslice**(L,16);

>>**imshow**(X,jet(16)),**colorbar**



It can be useful to create a pseudocolor representation.

Try these commands.

Caiani 16/3/2012

Biomedical image dimensions

	Static image (Mb)	Dynamic (Mb*f/sec)	V stat (Mb*sez)	Vdin (Mb*sez*f/sec)
RX	2048x2048x12=50			
TAC	512x512x16=4.2	x16=67	4.2x32=134	x16=2150
RMN	256x256x12=0.8	x16=12.5	0.8x32=25.6	x16=410
SPECT	128x128x8=0.15		0.15x32=4.8	
PET	128x128x8=0.15	x16=2.4	0.15x32=4.8	x16=77
ECO	512x512x8=2	x25=50	2x60=120	x25=3000

Image dimensions have to be considered, both in the processing phase as well as in image archiving and transmission

Caiani 16/3/2012

General structure of medical imaging data

Three types of information are generally present in these files:
- image data, which may be unmodified or compressed,
- patient identification and demographics,
- technique information about the exam, series, and slice/image.

Extracting the image information alone is usually straightforward.

Dealing with the descriptive information is more difficult and requires deeper understanding of how the files are constructed.

Data formats:
- fixed format, where layout is identical in each file;
- block format, where the header contains pointers to information;
- tag based format, where each item contains its own length.

Caiani 16/3/2012

An introduction to the DICOM single-file format

The Digital Imaging and Communications in Medicine (DICOM) standard was created in 1985 by the National Electrical Manufacturers Association (NEMA) to aid the distribution and viewing of medical images.
Part 10 of the standard describes a file format for the distribution of images. Most people refer to image files which are compliant with Part 10 of the DICOM standard as DICOM format files.

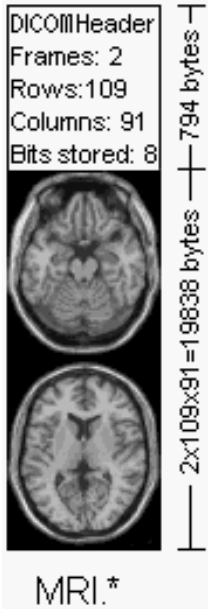
It should be noted that DICOM standards used by different vendors may have slight variations.

A single DICOM file contains both a header (which stores information about the patient's name, the type of scan, image dimensions, etc), as well as all the image data (which can contain information in three dimensions).

Each image modality has specific DICOM fields.

DICOM image data can be compressed (encapsulated) to reduce the image size. Files can be compressed using lossy or lossless variants of the JPEG format.

Caiani 16/3/2012



The DICOM header

In this example, the first 794 bytes are used for a DICOM format header, which describes the image dimensions and retains other text information about the scan.

The size of this header varies depending on how much header information is stored.

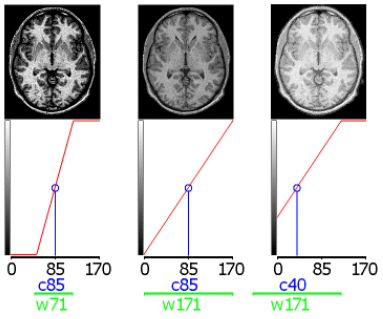
Here, the header defines an image which has the dimensions 109x91x2 voxels, with a data resolution of 1 byte per voxel (so the total image size will be 19838).

The image data follows the header information.

Caiani 16/3/2012

Window center and window width (brightness and contrast)

This is simply a way of describing the 'brightness' and 'contrast' of a medical image.
These values are particularly important for Xray/CT/PET scanners that tend to generate consistently calibrated intensities so you can use a specific C:W pair for every image you see (e.g. 400:2000 might be good for visualising bone, while 50:350 might be a better choice for soft tissue). Note that contrast in MRI scanners is relative, and so a C:W pair that works well for one protocol will probably be useless with a different protocol or on a different scanner.



Consider this image with intensities ranging from 0 to 170 (vertical axis: rendered brightness; horizontal axis: image intensity). A good starting estimate (middle panel) for this image might be a center of 85 (mean intensity) and width of 171 (range of values). Reducing the width to 71 would increase the contrast (left). On the other hand, keeping a width of 171 but reducing the center to 40 would make the whole image appear brighter (right). *Caiani 16/3/2012*

http://www.santesoft.com/dicom_viewer_free.html

DICOM files in Matlab

To read the infos in the file header :

>>info=**dicominfo**('CT-MONO2-16-ankle.dcm')

The variable info is a struct array: to access the content of each field, proceed as follows:

>>nome=info.PatientName

What's the difference in respect to cell array?

To read the DICOM data:

>>X=**dicomread**('CT-MONO2-16-ankle.dcm');

Read a DICOM header and explore its contents.

Caiani 16/3/2012

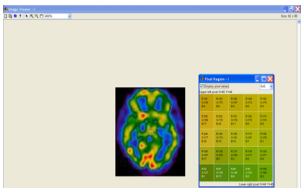
Visualization of 16 bits images

To visualize such images:

>>**imview**(X)

or trasform the image into 8 bits:

Y=**uint8**(X); **imshow(Y)**



In this case, the initial info on resolution of gray levels is lost.

Read a DICOM data, and visualize it with both options, observing the differences. Explore the tools of the imview window.

To save an image in DICOM format:

>>**dicomwrite**(X, MAP, filename,INFO, PARAM1, VALUE1, …);

Starting from the previously loaded image, change the patient name with yours. Save the image with the modified info in a new DICOM file. Clear the workspace, load the modified file and verify the modified information.

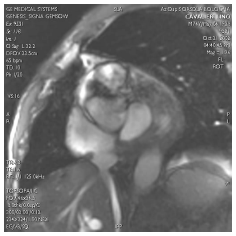
Caiani 16/3/2012

Data formats for image sequences

MPEG (Moving Picture Experts Group *.mpg) - LOSSY

It is the name of standards for audio and video compression, from 50:1 to 200:1.

There are 5 types of MPEG formats:
- MPEG1: standard for CD-I / Video CD media, up to 1.5Mbps
- MPEG2: standard for digital TV / Digital Video Disc, up to 4Mbps
- MPEG4: standard for video encoding
- MPEG7: description of multimedial contents
- MPEG21: platform for multimedia applications



Advantages
High resolution and high frame rate

Disadvantages
- Movie quality depends on the utilized system performance

Compared to other formats:
MPEG is the most compressed, not the most diffused; it can be converted into AVI and viceversa

Caiani 16/3/2012

AVI (Audio Video Interleaved *.avi)

Developed by Microsoft for Windows systems.
Movies reproduced at about 15 frame/sec.
It supports audio, but not captions. It stores video data that may be encoded in a variety of codecs;

Advantages
- Windows compatible, without need for additional drivers or players
- Highly diffused, good quality with adequate HW

Disadvantages
It typically uses less compression than similar formats such as .MPEG and .MOV, with reduced frame rate.

MOV (Quicktime)

Common multimedia format often used for saving movies and other video files developed by Apple for Machintosh. It may use one of many types of codecs to compress the file, including the H.264 video codec (from QuickTime 7.0), which uses a high-efficiency compression algorithm. Compatible with both Macintosh and Windows platforms.

Caiani 16/3/2012