Politecnico di Milano
Master of Science program in Biomedical Engineering

Biomedical Image Processing Lab class

(5 credits)

March 18, 2012

Lesson 1 – Introduction to biomedical image processing

Enrico Caiani, PhD

General information

Schedule:

EG.8 classroom: Friday 11:30 – 17:15

(11.30-13.30; break; 14:30-16:45)

13 weeks: last class June 15

Tutors:
Dr. Francesco Maffessanti: francesco.maffessanti@mail.polimi.it
Ing. Chiara Carminati: maria.carminati@mail.polimi.it

Consultation hours:

Friday 10:00 – 11:15 (or by appointment)

Biomedical Eng. Dpt. – 3rd floor

Phone: 3390; E-mail: caiani@biomed.polimi.it

Class Resources

“Corsi on line” web-site
(http://www.polimi.it/didattica/corsi-on-line/) :
- weekly theory and practicum (.pdf)
- image examples
- homeworks

Other recommended readings:
-Digital Image Processing using Matlab 2nd edition (R.C. Gonzalez, R.E. Woods, S.L. Eddins)
- Fundamentals of Electronic Image Processing (A.R. Weeks)
- Image Processing with Matlab: Applications in Medicine and Biology (O. Demirkaya, M.H. Asyali, P.K. Sahoo)

Software:
Matlab ver. 7.0 or higher

Evaluation process

Every 2-3 weeks, homework assignment:

March 30 April 20 May 4 May 25 June 15

You have 10 days to hand back your solution (delays will count as penalties, if not justified).

• At the end of the semester, average of the 5 homeworks grades:

• if <=24: you can proceed to register it

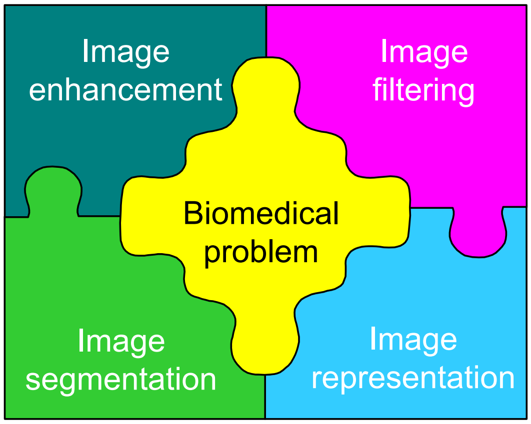
• if >24: 1 short-oral exam (mandatory, unless 24)

• starting grade for conventional oral exam (+/- 3 points)

N.B.: Discussion of the problems with other students is possible, but have to be explicitely declared. In any case, it is expected that **each student will produce an independent and original solution**. Otherwise, penalties will be considered in the grade.

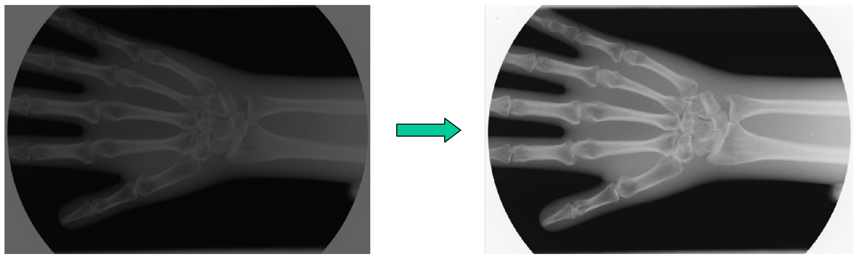
Class Objectives

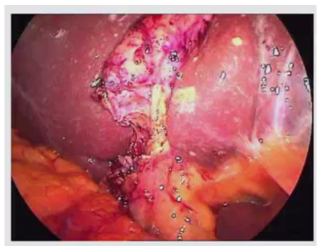
To provide the student with the theoretical and practical knowledge on the main topics relevant to biomedical image processing, thus giving the basis to tackle real problems in the biomedical field.



Example of typical problems

Image enhancement



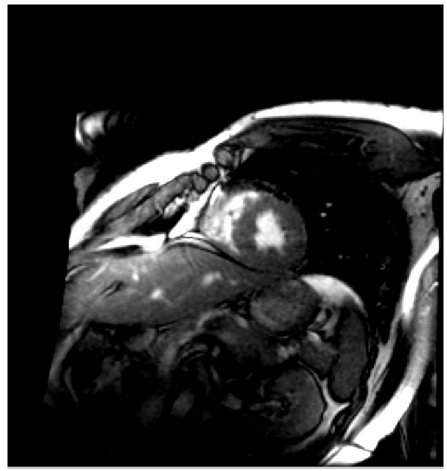


Illumination correction

Example of typical problems

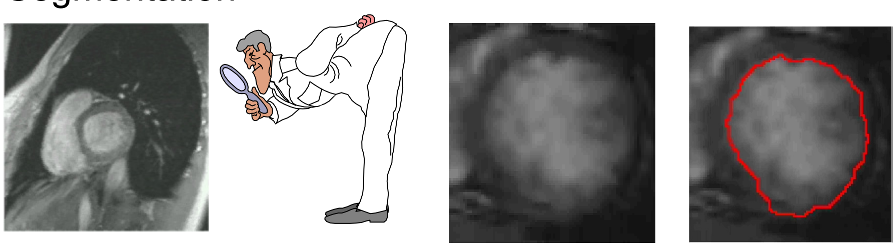
Noise characterization

Noise filtering

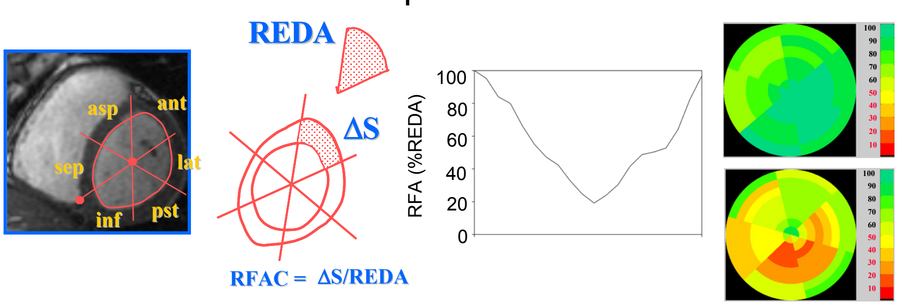


Example of typical problems

Segmentation

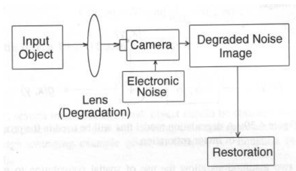


Quantification of clinical parameters



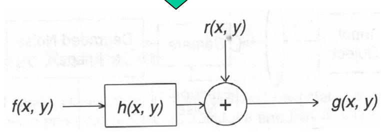
Example of typical problems

Restoration

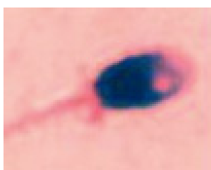


Blurring degradation

Motion degradation

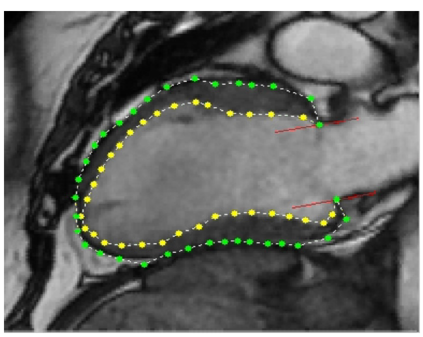


Color image processing

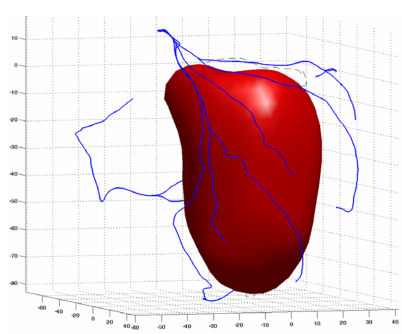


Example of typical problems

Point/feature tracking



Data fusion



Today's topics

Introduction to Matlab

Programming with Matlab

Introduction to biomedical images

Image formats

Colorspaces

DICOM

MATLAB

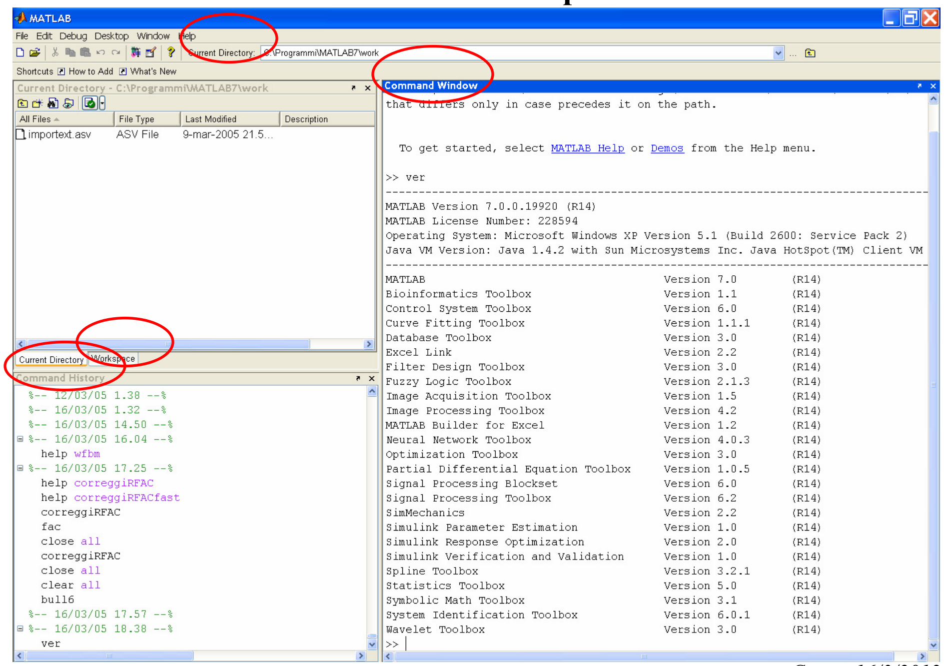
• it is a computational platform allowing operating with scripts that don't have to be compiled

• Interactive system having as elementary data the array, and not requiring to initialize (dimensions) it apriori

This allows to solve multiple computational problems, especially based on arrays and matrix, in a minimal time respect of what needed usin a non interactive programming language, like C or FORTRAN

Caiani 16/3/2012

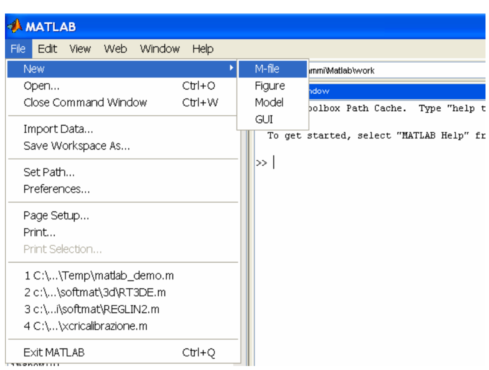
Matlab desktop

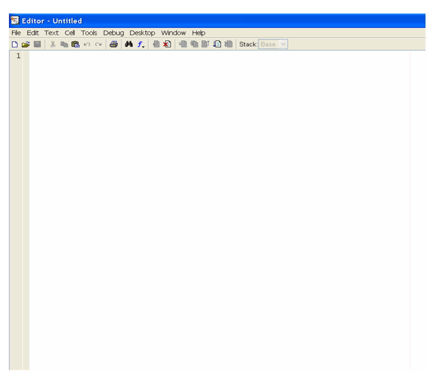


Caiani 16/3/2012

Matlab Editor

To write a Matlab program, select "New M-file" from the main menú; once the writing phase is finished, the script is saved as .m, and executed by calling it in the Command Window using the name utilized for saving.

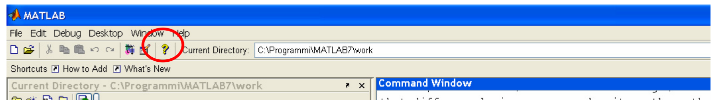




Caiani 16/3/2012

Help

It is possible to access helpful info in different ways:



>> ***help***

>> ***help*** COMMAND

How to save data contained in the workspace

>> ***save*** NOMEFILE   save all variables in NOMEFILE.mat

>> ***save*** NOMEFILE X   save only variable X

(some attention in portability between versions is needed)

>> ***load*** NOMEFILE   load the saved .mat file

>> ***edit*** NOMEFILE   open .mat file

Caiani 16/3/2012

MATLAB can be used as **calculator**:

- sum **+**   subtraction **-**   multiplication *****
- division **/**   power **^**

>>5+3

ans=

8

The ***ans*** variable contains the result of the last command.

Different **variables** can be defined, which names follow these rules:

- CAPS or non caps names are different
- max length 31 characters
- start the name with a letter, and it can containt letters, numbers or _

>>pippo=4;

>>pluto='abcd';

>>PLUTO=678;

Caiani 16/3/2012

To see the defined variables in the workspace, with their type and size, digit ***whos***.

To delete a variable :***clear*** *nomevar*

To clear all variables in the workspace: ***clear all***

ARRAY

They can be defined as follows:

• row array

directly listing te elements: >>x=[1 3 5 7 9]

by their relationship: >>y=(1:2:9)

using special commands:

>>w=***linspace***(1,9,5) (*start, end, n° el*)

>>k=***logspace***(0,.95,5) (*exp start, exp end, n°*)

combining other arrays: >>z=[x y]

Try to define the arrays described before, visualize their content, and see how they are classified into the workspace

Caiani 16/3/2012

• column array

use **;** as end of the line:

>>x=[1;3;5;7;9]

create a row vector and traspose it by **'**:

>>x=[1 3 5 7 9]**'**

• matrix

>>A=[1 1 1;2 2 2;3 3 3]

>>B=[x x x]

Zero array:

>> Z=**zeros**(N) >>ZZ=**zeros**(M,N) >>ZZZ=**zeros**(M,N,P)

Ones array:

>> U=**ones**(N) >>UU=**ones**(M,N) >>UUU=**ones**(M,N,P)

Try to define the arrays described before, visualize their content, and see how they are classified into the workspace

Caiani 16/3/2012

Array size

The command ***size(***A***)*** gives a row vector with the number of row and columns of the array A;

length(A***)*** gives the greatest number between the two dimensions.

numel(A***)*** gives the number of the element of A.

FIND ELEMENTS

To extract one element :		>>A(1,3)
	more elements:	>>A(2,1:3)
		>>A(3,3:end)
		>>A(1,1:2:end)
	a row:	>>A(2,:)
	a submatrix:	>>B=A(1:2,1:2)
	erase a column:	>>B(:,2)=[]

Try the commands and interpret the result.

Caiani 16/3/2012

MULTIDIMENSIONAL ARRAYS

It is possible to create an array with arbitrary number of dimensions. 3D arrays can be defined by joining 2D arrays using ***cat***, as for book's pages.

>>a=[1 0;0 1];b=[2 2;2 2];c=[0 3;3 0];

>>d=**cat**(3,a,b,c);

To get the desired page:

>>d(:,:,2)

To get the desired element:

>>d(1,2,2)

squeeze eliminates a unit dimension.

Try to apply these commands and see the results in the workspace

Caiani 16/3/2012

Between arrays, the **algebric operations** are stil valid, but pay attention to the size of the matrices:

```
>>  A = 1   2        B = 5   6            A*B =19   22
        3   4            7   8                 43   50
```

The operators **.*** **./** **.^** will apply the algebric operation between the single corresponding elements in the two arrays (that have to be of the same size, in this case)

```
>> A.*B =  5   12
          21   32
```


Try to apply these commands and see the results in the workspace

RELAZIONAL OPERATORS: < <= > >= == ~=

LOGICAL OPERATORS :

& AND

AND	0	1
0	**0**	**0**
1	**0**	**1**

~ NOT

NOT	
0	**1**
1	**0**

| OR

OR	0	1
0	**0**	**1**
1	**1**	**1**

xor(x,y)

xor(x,y)	0	1
0	**0**	**1**
1	**1**	**0**

any(A) 1 for each column of A with at least a non zero element

all(A) 1 for each column of A with all non zero elements