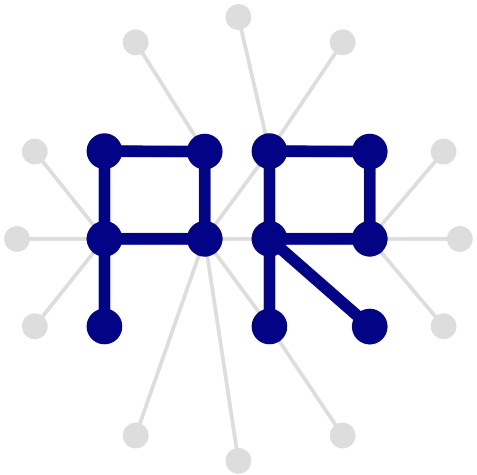
Medical Image Processing - Project

Image Processing in Matlab



Joanna Czajkowska, PhD

Research Group for Pattern Recognition
Institute for Vision and Graphics, University of Siegen

Please take a look here:

www.mathworks.com

Matlab

Basis

- Matrix/tabular oriented
- Many available libraries
- 2-D and 3-D graphics
- Symbolic analysis

Configuration of the Editor

Configuration

- Open new project
- Open existing project
- Change the paths
- Dock/Undock Editor
- Workspace
- Command Window
- Current Folder

Clean the workspace:

`clear all` - clear all existing variables
`close all` - close all the open figures
`clc` - clear command window

Run Section:

%% `...section...%%` + `Ctrl` + `Enter` - run section

Matrices

Matrix defining

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

Matrices

Matrix generation

```
» tab = []
» tab = 1:5
» tab = 1:-0.1:2
» X = rand(2)
» X = zeros(3)
» X = ones([1 3])
» N = nan([2 2])
```

Indexing

- ":" replaces all the elements
- the key word can be used instead of the last element

```
» A = zeros([3 4]);
» A(:) = 1:(size(A,1) * size(A,2));
» B = A(1:2, 2:3)
» A(:, 1:2:end)
» A(end:-2:1, 1:2:end)
```

Indexing

- **find**() - find non zero elements
- **find**(statement) - find the elements fulfilling certain statement
- **find**() - returns linear indices

```
» A = eye(3)
» f = find(A)
» f = find(A =1)
```

Indices

- referring the elements of a matrix with a single subscript `A(k)` - read it column by column
- it is possible to get the row-column equivalent **`int2sub`**`(`**`size`**`(A),ind)`
- to get the linear index from row-column equivalent use **`sub2ind`**`(`**`size`**`(A),rowind,colind)`

```
» A = [2 6 9;4 2 8; 3 5 1];
» linearind = sub2ind(size(A),3,2)
» [row col] = ind2sub(size(A),linearind)
```

Matrices

Cell array

```
» A = cell(2,1)
» A1,1 = zeros(3)
```

Structure

```
» B = struct
» B.name = 'example'
» B.data = A
» B = setfield(B,'label',1)
» getfield(B,'label')
```

Matrix operations

```
» A + B
» B - B
» A * B
» A.* B
» A'
» A/B
» inv(A) » sqrt(A)
```

Matrix operations

```
» B = eye(3)
» B(1,1) = 10
» B(:,2) = 10
» B = zeros(3); A = eye(3); B(A==1)=3 » whos
```

Data types

Types

» `x = 100`
» **`double`**`(x)` - double-precision
» **`single`**`(x)` - single-precision
» **`int8`**`(x)` - signed 8-bit precision integers
» **`uint16`**`(x)` - unsigned 16-bit precision integers

Data types

Conversions

```
» logical(1)
» num2str(10)
» str2double('5')
```

Logical connectives

Relations

- == - a=b
- >= - a>=b
- =< - a<=b
- ~= - a≠b
- > - a>b
- < - a<b

```
» 3>4
» true == 1
» [1 2 4] == [1 2 5]
» any([1 0 0])
» all([1 2 1])
```

Logical connectives

Connectives

- `~ a` - not a
- `a && a` - a and a
- `a || a` - a or a

```
» a=1; b=1;
» a && b
» a && ~ b
```

is*()

```
» A=ones(3)
» isempty(A)
» isequal(A,[])
» ~isempty(A)
» isnan(A)
```

Useful functions

Functions

```
A = ones(4); A(:)=1:16
» sum(A)
» sum(A(:))
» min(A)
» min(A(:))
» max(A,[],2)
» mean(A,2)
» median(A)
» unique(A)
```

Conditional statement

If... Then...

```
if expression 1
   statement 1
elseif expression 2
   statement 2
then
   statement 3
end
```

Example

```
» if a==1
»    disp('a is equal to 1');
» end
```

Loops

for...

```
for k=kinitval:kendval, statements, end
for k=kinitval:step:kendval, statements, end
```

Examples

```
» for i=1:3, disp(i), end
» for i=[1 2 3], disp(i), end
```

Loops

while...

```
while expression
   statement
end
```

Example

```
» i=0;
» while(i < 10)
»    disp(i);
»    i = i + 1;
» end
```

Reading Image Files

Read/write image file:

```
img = imread(filename, fmt);
```

$fmt = \{BMP, CUR, GIF, PNG, PBM, PCX, \ldots\}$

```
» img = imread('board.tif');
» BW = rgb2gray(img); % conversion
» whos
» figure; imshow(img);
» figure; imshow(BW,[]);
» imwrite(BW,'name.png');
```

Color mapping

Map

```
» imshow(BW);
» colormap(copper(256));
» colormap(hot(256));
» colormap(gray(256));
```

Paths

Find all the files with the set extension

```
» av_files = dir(fullfile(path,'*.bmp'));
» name = av_files(k,1).name;
```

Reading DICOM Files

Read DICOM file:

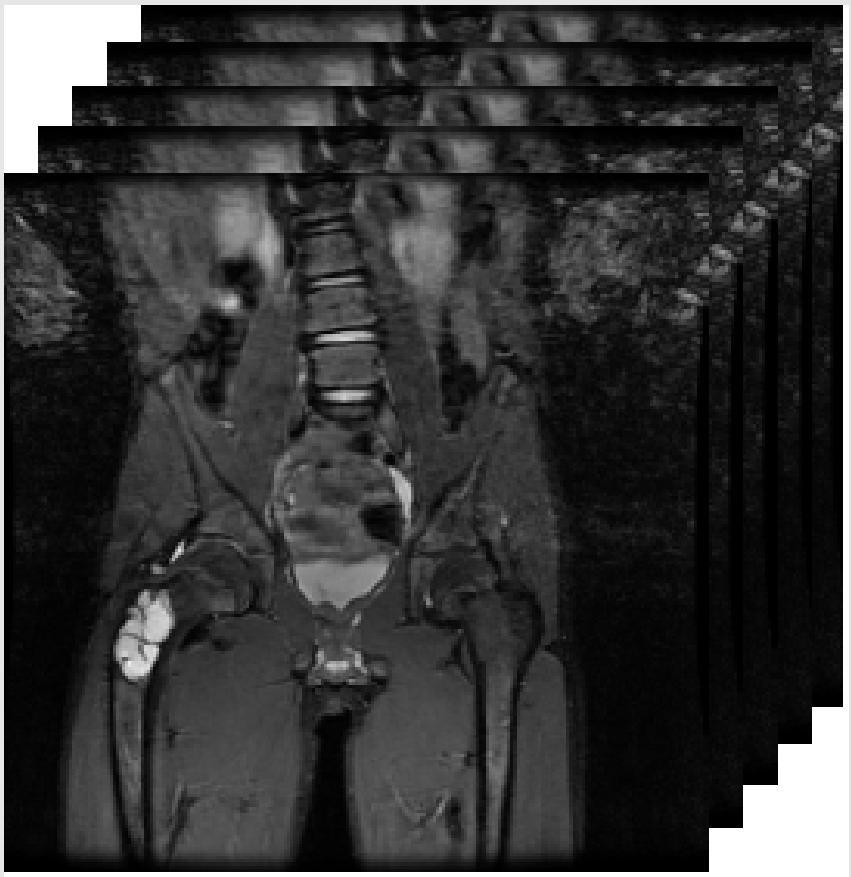
```
» X = dicomread(fullfile(path,name));
```

Read DICOM header:

```
» Y = dicominfo(fullfile(path,name));
```

Create 3-D matrix

```
» I = cat(3,img1,img2);
```



Exercise

Read the whole series into a matrix:

```
» av_files = dir(fullfile(path,'*.dcm'));
» img=[];
» for k=1:size(av_files,1)
»    name = av_files(k,1).name;
»    X = dicomread(fullfile(path,name));
»    img=cat(3,img,X);
» end
» Y =
dicominfo(fullfile(path,av_files(1,1).name))
```

Plot

Show the single slice in the series

```
» figure; imshow(img(:,:,k),[]);
```

Show two slices in the series

```
» figure; imshow([img(:,:,k)
img(:,:,k+1)],[]);
```

Subplot

```
» figure; subplot(2,2,1);
imshow(img(:,:,1),[]);
» subplot(2,2,2); imshow(img(:,:,2),[]);
» subplot(2,2,3); imshow(img(:,:,3),[]);
» subplot(2,2,4); imshow(img(:,:,4),[]);
```

Subplot

```
» figure; » for k=1:4
»    subplot(2,2,k); imshow(img(:,:,k),[]); »
end
```

Functions

Create a new function

```
function [outputs]=name(inputs)
```

- The declaration must be in the first executable line of the function.
- Save the function code in a text file with a .m extension.
- You can declare multiple local functions within the same file, or nest functions.
- If any function in a file contains a nested function, all functions in the file must use the end keyword to indicate the end of the function. Otherwise, the end keyword is optional.