بسمه تعالی

مقدمه‌ای بر

پردازش تصاویر پزشکی در Matlab

By: Saeed Mohagheghi
Shahed University
Winter 1396 | 2018

فهرست مطالب

- مقدمه
  - فرمت‌های ذخیره تصاویر پزشکی
  - جهت‌های آناتومیکی در تصاویر پزشکی
- MATLAB
  - خواندن و نوشتن فرمت‌های مختلف تصاویر پزشکی (Meta، Analyze، Dicom)
  - نمایش اسلایس‌های تصویر (دوبعدی) و ماسک (سه‌بعدی)
  - ایجاد ROI با استفاده از ماسک / انتخاب دستی ROI در تصاویر
  - اعمال ROI یا ماسک برای جداسازی قسمتی از تصویر
  - آستانه گذاری / یکسان سازی هیستوگرام

2

مقدمه

Medical image formats / Anatomical Planes & Directions

3

فرمت‌های مرسوم ذخیره تصاویر پزشکی

Format	Extension	No. of files
Meta	.mhd / .raw (.zraw) .mha	2 (1 for header / 1 for image) 1 (1 for all)
Analyze	.hdr / .img	2 (1 for header / 1 for image)
Nifti	.hdr / .img .nii (.nii.gz)	2 (1 for header / 1 for image) 1 (1 for all)
Dicom	.dcm No extension!	? (1 for each slice + header)

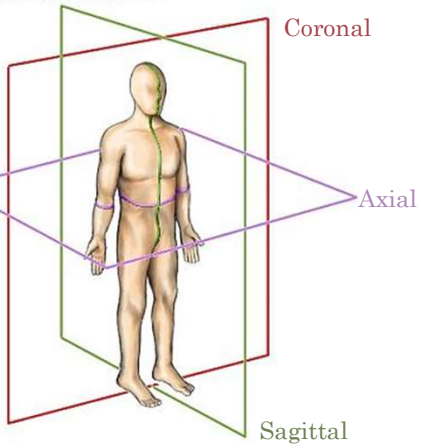
Nifti → Neuroimaging Informatics Technology Initiative
Dicom→ Digital Imaging and Communications in Medicine

4

صفحه‌های آناتومیکی

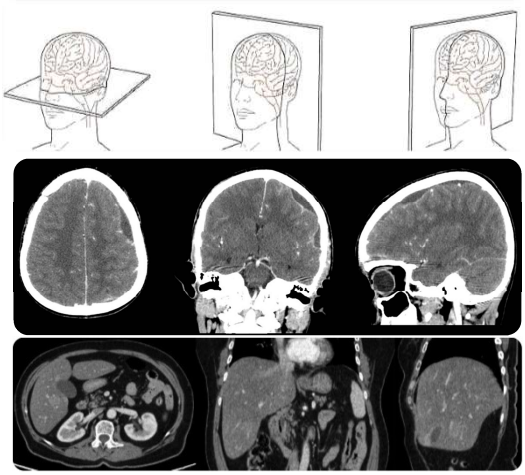
Anatomical Planes

- Anatomical position
- Anatomical planes
- Coronal (Frontal) Plane divides the body into front and back sections
- Sagittal Plane divides the body into left and right sections
  - Median - divides the body into equal left and right parts
- Axial (Horizontal or Transverse) Plane - divides the body into upper and lower segments

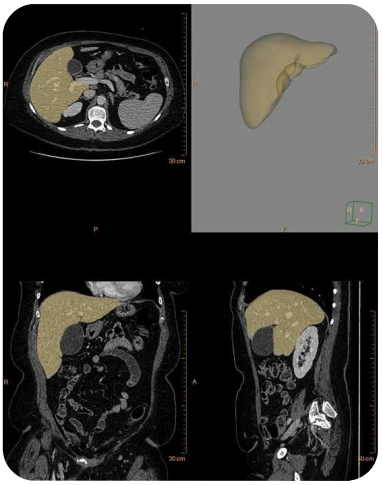


5

صفحه‌های آناتومیکی



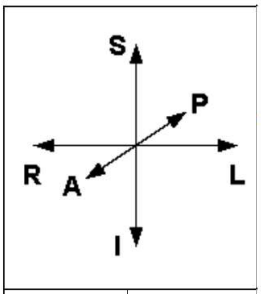
Axial Coronal Sagittal



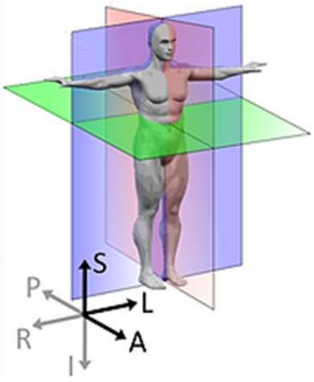
6

جهت‌های آناتومیکی

→ جهت‌ها نسبت به بیمار تعریف می‌شوند



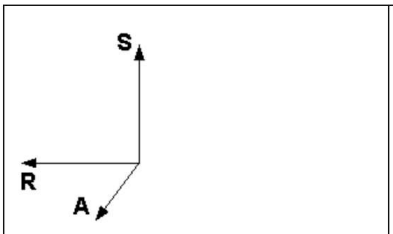
Right	Left
Anterior	Posterior
Inferior	Superior



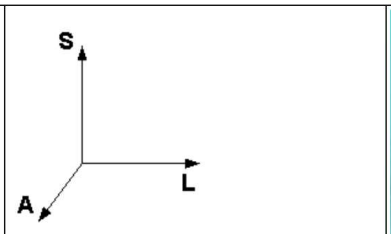
7

جهت‌های آناتومیکی

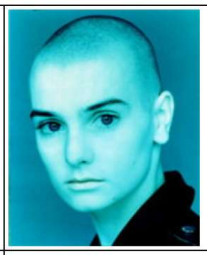
- دستگاه‌های مختصات مرسوم



"Neurological" convention for axes, aka "RAS"



"Radiological" convention for axes, aka "LAS"



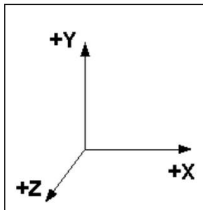
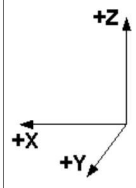
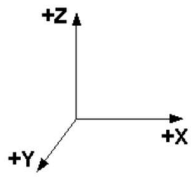
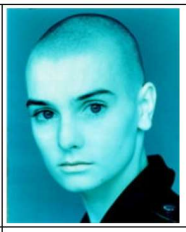
Head orientation reference

- Note that each convention specifies not just the positive direction of three axes, but also the order for listing them (eg: LAS, but not ALS).

8

جهت‌های آناتومیکی

- محورهای مختصات

			
1. Would be described as **LSA**: +X = L +Y = S +Z = A	2. Would be described as **RAS**: +X = R +Y = A +Z = S	3. Would be described as **LAS**: +X = L +Y = A +Z = S	Head orientation reference

9

جهت‌های آناتومیکی

- Image File Voxel Ordering (what comes in header file):

Storage order in file	Slice orientation	3-Letter "from" name	Increasing position
R-L → P-A → I-S	Axial	RPI	LAS
L-R → P-A → I-S	Axial	LPI	RAS
R-L → I-S → P-A	Coronal	RIP	LSA

R-L → P-A → I-S means:
Voxels ordered from right to left to store a row
Rows ordered from posterior to anterior to store a slice
Slices stored from inferior to superior to store a volume

10

تصاویر پزشکی در Matlab

Meta / Analyse / Dicom

11

خواندن تصاویر Meta

- (هنوز) تابع آماده برای خواندن این نوع تصاویر در Matlab وجود ندارد
- چند نمونه از تولباکس‌های موجود برای خواندن و نوشتن تصاویر پزشکی

- ReadData3D *
- Medical Image Processing **
- Slicer ***

* https://www.mathworks.com/matlabcentral/fileexchange/29344-read-medical-data-3d

** http://www.mathworks.com/matlabcentral/fileexchange/41594-medical-image-processing-toolbox

*** http://www.mathworks.com/matlabcentral/fileexchange/27983-slicer

12

اضافه کردن تولباکس‌ها و کدهای جدید به Matlab

- حالت 1: نصب برنامه با پسوند **.mlappinstall**
  - نصب از طریق تب APPS ← Install App
  - نصب از طریق تب HOME ← Open
  - ← **نتیجه**: اضافه شدن به برنامه‌های تب APPS

- حالت 2: نصب برنامه با پسوند **.mltbx**
  - نصب از طریق تب HOME ← Open
  - ← **نتیجه**: اضافه شدن به تولباکس‌ها در پنجره Add-Ons

13

اضافه کردن تولباکس‌ها و کدهای جدید

- حالت 3: اضافه کردن مسیر پوشه حاوی کدها به مسیرهای Matlab
  - از طریق گزینه Set Path در تب HOME
  - از طریق کدنویسی
  - ←نتیجه: اضافه شدن پوشه‌ها به مسیرهای شناخته شده در Matlab

- از طریق کدنویسی

- addpath(genpath('path'))
- savepath

14

خواندن تصاویر Meta

- ReadData3D
  1. info = mha_read_header(filename); % Read (filename = address of .mhd file)
  2. img = mha_read_volume(info); % (Also Reads compressed data)
  - [img info] = ReadData3D; % Load from GUI

- Medical Image Processing
  1. [img info] = read_mhd(filename); % Read (filename = address of .mhd file)
  2. write_mhd(filename, image); % Write (filename = address of .mhd file)

- Slicer
  1. Files → Open…
  2. Files → Export To Workspace…
  3. Files → Save Image…

15

تصاویر Meta

- Our Code

- Read
  1. info = metaImageInfo(filename); % Read (filename = address of .mhd file)
  2. img = metaImageRead(info);
  - [img info] = metaImageRead(filename);

- Write
  - metaImageWrite(img, filename); % Write (filename = address of .mhd file)
  - metaImageWrite(img, filename, info);

16

تصاویر Analyze

- توابع آماده فقط برای خواندن این نوع تصاویر در Matlab وجود دارد

- Read
  1. info = analyze75info(filename);
  2. img = analyze75read(info);
     img = analyze75read(filename);

- برای نوشتن تصاویر Analyze می‌توان از کد آماده زیر استفاده کرد

- Write (Our Code)
  - writeanalyze(img, filename, voxelsize)

17

تصاویر Dicom

- توابع آماده برای خواندن و نوشتن این نوع تصاویر در Matlab وجود دارد
- برای خواندن تصاویر Dicom باید تمام فایل‌ها (اسلایس‌ها) به ترتیب لود شوند

- Read
  1. info = dicominfo(filename);
  2. img = dicomread(info);
     img = dicomread(filename);

- برای نوشتن تصاویر Dicom باید تمام فایل‌ها (اسلایس‌ها) به ترتیب ذخیره شوند

- Write
  - dicomwrite(img, filename);
  - dicomwrite(img, filename, info);

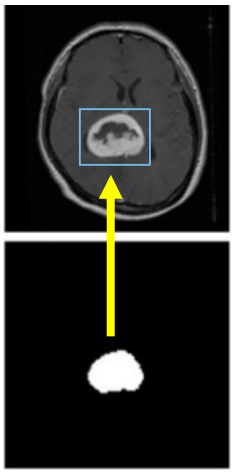
18

نمایش اسلایس‌ها

- 3rd dimension: axial plane
  - imshow(squeeze(img(:,:,80)), []); title('Axial slice')
- 2nd dimension: coronal plane
  - imshow(squeeze(img(:,80,:)), []); title('Coronal slice')
- 1st dimension: sagittal plane
  - imshow(squeeze(img(80,:,:)), []); title('Sagittal slice')
  - Attention: you may need to use some functions to have real view of the images such as fliplr, rot90, imrotate, …

19

استفاده از تصویر ماسک



- محاسبه ROI حول بافت مورد نظر با استفاده از تصویر ماسک
- [ROI, ROI_Coordinate] = GetBoundingBox(Image, Mask, pad)
- اعمال ماسک بر روی تصویر برای جداسازی بافت مورد نظر
- segmented_image = img .* mask
- بهتر است ماسک به صورت باینری باشد
- mask_bw = logical(mask)

20

نمایش سه بعدی ماسک

```
mask = smooth3(mask, 'box', 5);
mask = reducevolume(mask, [5,5,5]);
```
← Pre-Process volume (optional) ①

```
[m, n, p] = size(mask);
[x, y, z] = meshgrid(1:n,1:m,1:p);
```
← Create grids ②

```
fv = isosurface(x, y, z, mask);
```
← Create faces + vertices ③

```
patch(fv, 'FaceColor', [0.3,0.8,0.8], 'EdgeColor', [0.3,0.3,0.3]);
trisurf(fv.faces, fv.vertices(:,1), fv.vertices(:,2), fv.vertices(:,3));
trimesh(fv.faces, fv.vertices(:,1), fv.vertices(:,2), fv.vertices(:,3));
```
← Visualize ④

21

نمایش ماسک بر روی تصویر (Overlay)

```
maskSlice = edge(maskSlice);
```
← Mask edge (optional) ①

```
img_ov = imoverlay(imgSlice, maskSlice);
img_ov = imoverlay(imgSlice, maskSlice, color);
```
color examples: 'green' / 'g' / [0.3 0.6 0.6]

← Create overlay image ②

```
imshow(img_ov)
```
← Show overlay image ③

22

آستانه گذاری

```
imhist(img);
```
← Histogram (to check) ①

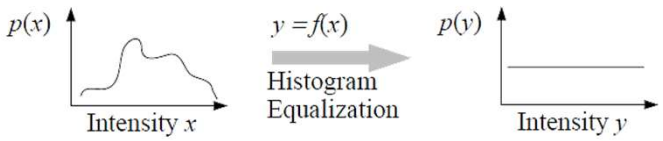
```
thresh = img > 0.5;
thresh = img > 0.1 & img < 0.6;
thresh = img > graythresh(img);
thresh_img = img .* thresh_mask;
```
← Thresholding ②

```
imshow(thresh_img)
```
← Show overlay image ③

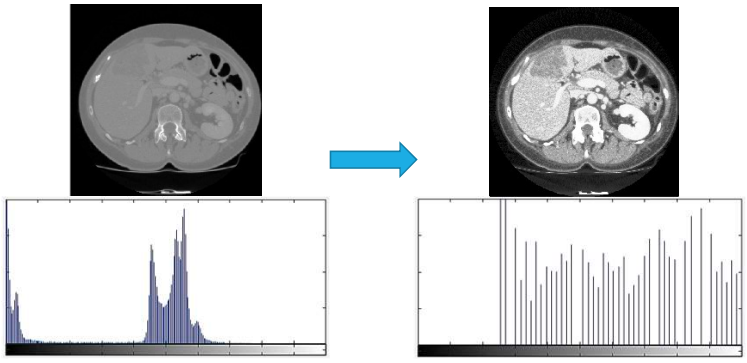
23

یکسان‌سازی هستوگرام

- Histogram Equalization



- img_eq = histeq(img);



24

25