Medical Image Processing
Introduction to the Exercises

Medical image preprocessing techniques using MATLAB

Opening remarks:

1. Before start writing your code, please clear all existing variables, close all open figures and clean the command environment (MATLAB):
   I. clear all
   II. close all
   III. clc
2. Please write the comment (%.....)on each operational section (%%.....%%) of your code which increases the readability of your code.
3. Please use the proper names for your folders/subfolders, M-files, functions and processed files indicating the performed process and date (e.g., CT0001_Compressed.png, CT0001_HistEqual.png,…).
4. Please save the log activities of the processing chain.
5. Please store the intermediate processed results.
6. Please start a new project for each topic.
7. Please provide a UML model if you are running your code on an object oriented basis with full connectivity.
8. There are many available documents in Internet for those who are not familiar with MATLAB; please refer to them and for all technical problems please ask me.
9. Please specify the group names for semester project as soon as possible and inform me via Email (amir.tabatabaei@uni-siegen.de).
10. The presentation of the project will take place on **Friday 02/02/2018 8:00-10:00**. Each group is given ca. 12mins time slot.

Some useful basic commands in MATLAB image processing toolbox

- Reading/writing image files (any format)

- img = **imread**(’img1.tif’);
- BW = **rgb2gray**(img); (converting an RGB image to a grayscale image)
- **figure**; **imshow**(img);
- **figure**; **imshow**(BW,[]);
- **imwrite**(BW,’name.png’);
- **Imshowpair**(img, BW, ‘montage’); (displaying two images in one figure)
- dicimg = **dicomread**(‘dcmimg.dcm’); (reading a DICOM image)
- dicomInf = **dicominfo**(dicimg); (getting DICOM file information)
- **colormap**(map); (sets the colormap for the current figure to the colormap specified by map)

Reading/writing image files from/to a directory

```
clear;
srcFiles = dir('C:\users\Amir\Documents\MATLAB\PR-Project-1\seventh-image-set-2016-08\test\*.tif'); %Specifying the source files folder
mkdir('TestResults'); %creating a new folder called : TestResults for storing the read images.

for i=1:length(srcFiles)
    filename = strcat(srcFiles(i).name); %Reading fil names

    I = imread(filename); %Reading images
    BW = rgb2gray(I); %Converting to a grayscale image

%Showing the images
     figure; imshow(I);
     figure; imshow(BW);

       %Writing the images in specified folder TestResults
    baseFileName = sprintf(srcFiles(i).name);
    fullFileName = fullfile('TestResults', baseFileName);

    imwrite(I, fullFileName);
end
```

Implementing histogram equalization on grayscale images

```
%This function applies histogram equalization on the input image I and stores it in output image J

function  J = imHisEqual( I )

[row, col]=size(I);

freq = zeros(256, 1);

for i=1:row
    for j=1:col
        freq(I(i, j)+1)=freq(I(i, j)+1)+1;
    end
end

freq = freq/(row*col);

S=0;
newIntensity = zeros(256,1);

for i=1:256
    S = S + freq(i);
    newIntensity(i) = newIntensity(i)+S;
    newIntensity(i) = round(newIntensity(i)*255);
end

for i=1:row
    for j=1:col
        I(i, j) = newIntensity(I(i, j)+1);
    end
end

J=I;
end
```