“Sviluppo di Modelli Computazionali 3D”
(3D Models Generation)

Laboratory 02 – 10/03/2017

Warm-up:

1. Download the source code for this laboratory session at:
   http://www.banterle.com/francesco/courses/2017/be_3drec/lab/code_lab_2.zip
2. Extract the zip file in the folder MATLAB in Documents;
3. Add folders and sub-folders in the MATLAB path;
4. Read the file **slice_1016.png** in the folder **code_lab_2/data/mri** using **imread**.
5. Plot the 256 row.

Exercise 1:

1. Write a function with for reading a 3D volume (a n-m-1-d matrix, where n is height, m is width, and d is depth) stored as 2D images from a directory.
   MATLAB functions to be used:
   1. **dir**: a built-in function that lists the content of a directory;
   2. **imread**: a built-in function that reads images. For DICOM files, you have to use **dicomread**.

   This function needs to have this signature:

   function vol = ReadVolume(name_directory)

2. The values in the volume need to be normalized in [0,1] and stored in double precision.
3. Test this function using data in the folder **code_lab_2/data/mri**

Exercise 2:

1. Implement these functions:
   1. **ImContrastStretching**
   2. **ImGamma**

   in the folder **code_lab_2/MATLAB/contrast**
2. Play with this two functions using as input **CT-MONO2-16-ankle.dcm** image.
3. Play with **histeq** function using as input **CT-MONO2-16-ankle.dcm** image.
   NOTE: define a reasonable ROI to avoid peaks in the histogram (check this with function **hist**). Crop data if needed using the function **imcrop**.
4. Write a function that fits the result of a histogram equalization into a linear model (**ImContrastStretching**) and gamma model (**ImGamma**):
   MATLAB functions to be used:
   1. **dicomread**: a built-in function that reads DICOM files.
   2. **hist**: a built-in function that creates and visualizes the histogram of an image.
   3. **histeq**: a built-in function that equalizes an image histogram.
   4. **imshow:** a built-in function that visualizes images.
   5. **imcrop**: a built-in function that crops an image.

6. **fit**: a built-in function that fits data into functions;
7. **fminsearch**: a built-in function that minimizes a non-linear energy function, it can be used for fitting as well starting from an initial solution.
8.

This function needs to have this signature:

function fitting_parameters = contrastFit(img_original, img_hist_eq)

5. Test this function by reading the image **CT-MONO2-16-ankle.dcm**:
    1. info = dicominfo('CT-MONO2-16-ankle.dcm');
    2. img = dicomread(info);

Exercise 3:

1. Write a function that converts a volume encoded as axial images into a volume encoded as saggital images.

    This function needs to have this signature:

    function volOut = ConvertFromAxialToSaggital(volIn)

    MATLAB functions to be used:
    1. **reshape**: a built-in function that reshapes matrices.
    2. **imshow:** a built-in function that visualizes images.
2. Test this function using data in the folder **code_lab_2/data/mri**.
   NOTE: to understand if the result is correct either display images on screen or write them down into files whatever get the job down.

Exercise 4:

1. Read the image **mri_noisy.png** in the folder **code_lab_2/data/**.
2. Compute the SNR for the full image and in different regions.
3. Apply **bilateralFilterWrap** function to reduce the SNR in measured regions.
4. Write an iterative function that iteratively applies the bilateral filter until noise reaches a an input given threshold.

    This function needs to have this signature:

    function imgOut = IterBil(imgIn, sigma_s, sigma_r, thr)

    MATLAB functions to be used:
    1. **imrect**: a built-in function that creates rectangular ROI.
    2. **bilateralFilterWrap**: it is a wrapper to **bilateralFilter** function by Jiawen Chen
    3. **imshow:** a built-in function that visualizes images.