**Image Processing – Logbook**

**Chapter 1 (21/01/2020):**

These commands can be used to open and **read an image file** (and its header):

fid=fopen('head.128','r'); % Opens file for reading (assuming you are already in the correct directory)

[x,npels]=fread(fid,[128,128],'uchar'); % Reads data values

x=x';

fclose(fid); % Close the file handle

* Do not forget to transpose the ‘x’ matrix, as it is read differently than in MATLAB because the commands were written in C

image(x); % Displays imaged coded by matrix ‘x’

colorbar; %Presents a colorbar relating color to intensities

colormap(gray(64)) % Simply chooses the colormap from hardware to be used to display the image chosen in the image function. It returns an Nx3 matrix with the corresponding mapped out colors according to the pixel number. N is the number of grayscale levels. (in this case 64)

Colormaps can be created simply as Nx3 matrix of your desired N, which will vary the quantisation and intensity of the graphs displayed (For grayscale images)

For colour images, there are 8 bytes per pixel (and for each element of the colormap)

The following command displays a montage of images side by side on MATLAB:

montage(X, map); % Creates a montage of array X with colormap ‘map’

To display a moving sequence of images, use the following commands:

M=immovie(X,map); % Creates a movie out of the stored images and plays it

movie(M,20,10); %Third term determines the frame rate and second term determines the duration of the animation

**Chapter 2 (28/01/2020):**

* ‘rgb2gray(X)’ converts an image from RGB to grayscale
* ‘hist(<image\_array>,<number of bins>)’ displays a histogram with the input properties
* ‘bwlabel(gbin,4)’ creates a labelled image map for different sections connected by horizontal/vertical points
* ‘cat(n, s.Field)’ takes the ‘Field’ all elements of struct s and creates ‘n’ vectors with them