Computer Vision
Workshop 1
Introduction to MATLAB

Aims of the workshop

We have currently looked at some colour models and image representation during lectures. Part of this involved a live demo of MATLAB and use of MATLAB integrated applications. In this workshop you will be getting hands-on experience with MATLAB and becoming familiar with the interface and program as a whole.

MATLAB will be used for the ACW and within Lectures. The majority of lab-based workshops will make use of this. It is made available across University imaged machines. MATLAB can be installed by students on their own machine at home. We suggest using R2018b, but most versions will work. Guidelines on how to obtain your license can be found at the link on Canvas.

You may need to ensure that you install any additional packages you require (typically called toolboxes). These may include the Computer Vision packages. Further packages can be installed as required without requiring a full reinstall.

Please see ‘Useful Information’ below on how to lookup certain MATLAB functionality. The concept behind this workshop is about discovery, and experimentation surrounding topics covered so far.

Feel free to discuss the work with peers, or with any member of the teaching staff.

Useful Information

MATLAB Documentation

MATLAB functionality mentioned during lecture or workshops is fully-documented and available at: https://uk.mathworks.com/help/matlab/

You may find the “Getting Started” section of this documentation to be useful: https://uk.mathworks.com/help/matlab/getting-started-with-matlab.html

In addition, MATLAB Answers may be useful in finding common solutions to any issues you may encounter: https://uk.mathworks.com/matlabcentral/answers/help?s_tid=gn_mlc_ans_hlp

Reminder

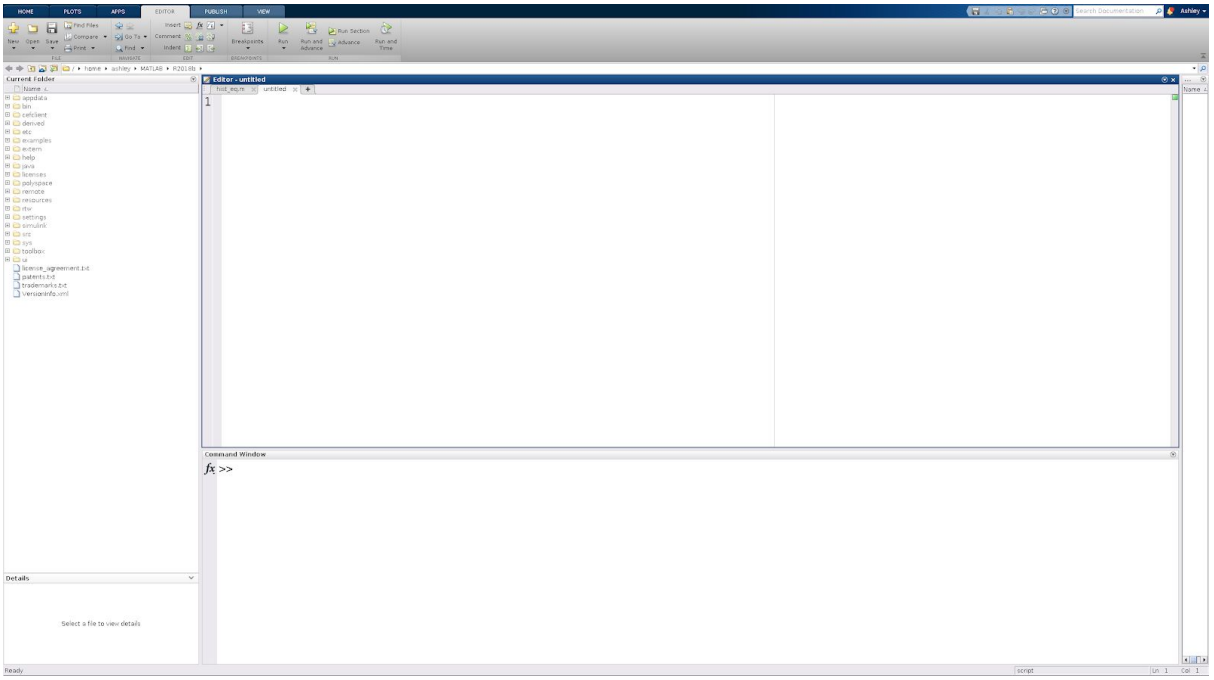
We encourage you to discuss the contents of the workshop with the delivery team, and any findings you gather from the session.

Workshops are not isolated, if you have questions from previous weeks, or lecture content, please come and talk to us.

First Steps

Exercise 1: Read the useful information section above.

Exercise 2: Start MATLAB from the main menu. The following image shows you what the initial MATLAB workshop looks like.



MATLAB scripts have the file extension .m; however, it also supports “Live Scripts” which enables the separation of code into ‘blocks’ where you can have text explaining certain blocks of code. An example of which can be seen below. More information is available: https://uk.mathworks.com/help/matlab/live-scripts-and-functions.html

Exercise 3: You can load images using the *imread* function. Matlab will automatically understand many image types.

```
% Note in Matlab inline comments follow the percent sign
% The following Matlab code loads and displays a Binary image
BinaryImage = imread('circles.png');
imshow(BinaryImage);
% The following Matlabcode loads and displays a grayscale image
GrayScaleImage = imread('coins.png');
imshow(GrayScaleImage);
GrayScaleImage2 = imread('cameraman.tif');
imshow(GrayScaleImage2);

% The following Matlabcode loads and displays a colour image
ColourImage = imread('lighthouse.png');
imshow(ColourImage);

% note each call to imshow overwrites the displayed image
```

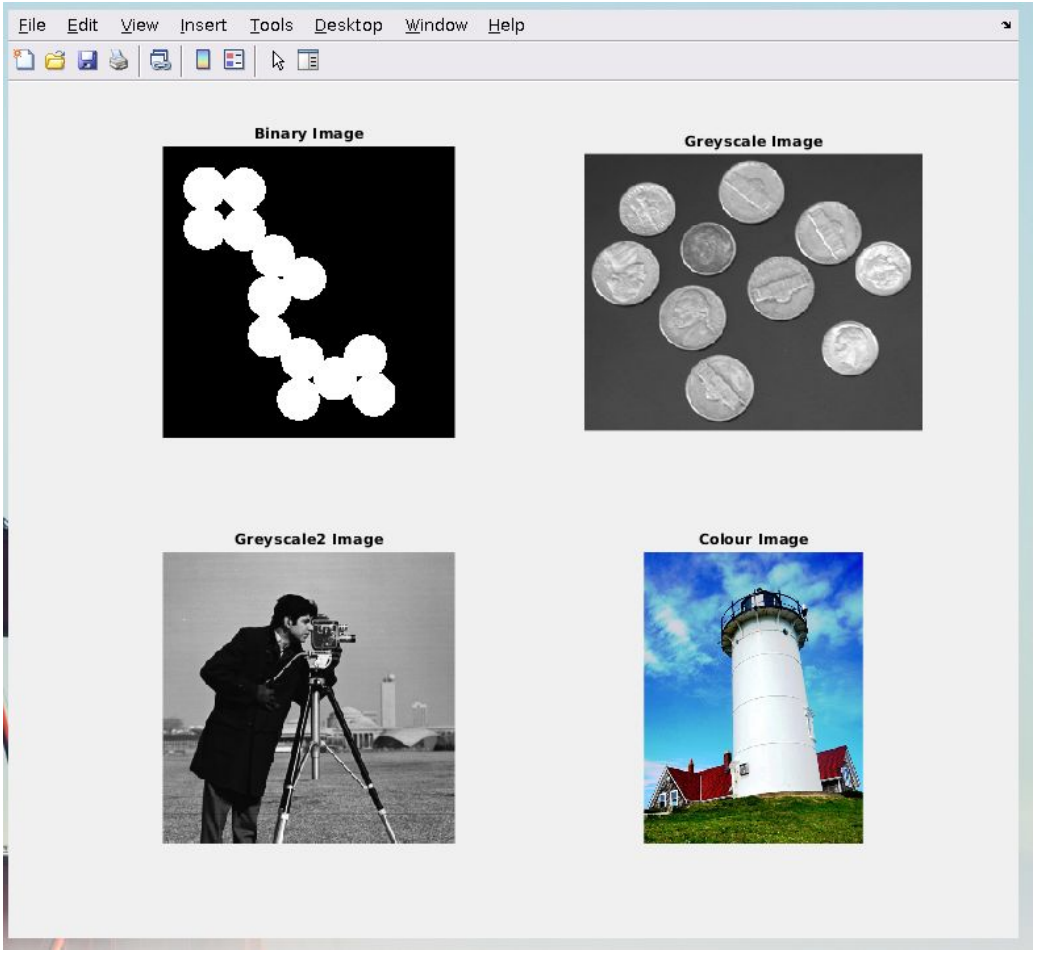
Exercise 4: Using *figure* and *subplot* we can arrange multiple images in a grid for displaying and comparing them

```
figure % new figure - note not followed by semi-colon
subplot(2, 2, 1); % define 2 by 2 space and specify 1st quadrant
imshow(BinaryImage);
subplot(2, 2, 2); % on 2 by 2 space specify 2nd quadrant
imshow(GrayScaleImage);
subplot(2, 2, 3); % on 2 by 2 space specify 3rd quadrant
imshow(GrayScaleImage2);
```

```
subplot(2, 2, 4); % on 2 by 2 space specify 4th quadrant
imshow(ColourImage);
```

Exercise 5: We can annotate the text by using *title* and *hold on / hold off* this allows us to keep previously rendered items and add to them. E.g our Title.

```
figure % new figure - note not followed by semi-colon
hold on; %
subplot(2, 2, 1); % define 2 by 2 space and specify 1st quadrant
imshow(BinaryImage);
title('Binary Image');
subplot(2, 2, 2); % on 2 by 2 space specify 2nd quadrant
imshow(GrayScaleImage);
title('Greyscale Image');
subplot(2, 2, 3); % on 2 by 2 space specify 3rd quadrant
imshow(GrayScaleImage2);
title('Greyscale2 Image');
subplot(2, 2, 4); % on 2 by 2 space specify 4th quadrant
imshow(ColourImage);
title('Colour Image');
hold off;
```



Exercise 6: You can load images from files by specifying the file-path. E.g “C:/…./ashley/Documents/RedCar.png”. Download an image from google, and load this into MATLAB.

MATLAB Apps

The next steps are to use the MATLAB Apps from within the Computer Vision Toolbox.

Click on the **Apps** tab from within Matlab. Scroll down to **“Image Processing and Computer Vision Toolbox”**. Select **“Image Viewer”**. Import / Load an image (either from the Workspace which you’ve previously loaded. Or directly).

Experiment with the options under Tools, in particular:

1. Image information to access generic information about the image as well as any metadata.
2. Measure distances within the image (I.e How many pixels between subjects in your image).
3. Pixel region - Create, resize, relocate Box to investigate different properties of regions within your chosen image.

Experiment with different applications under the “Image Processing and Computer Vision Toolbox”, and example we used in lectures is the Colour Thresholder. Can you filter the eyes out of the Redcar example (image on Canvas) to make it scary?