Australian National University
Research school of Engineering

ENGN4528/6528 - Computer Vision
Computer Lab 3: Image Features – Harris Corner, SIFT, Colour Segmentation

Objectives:

The goal of this computer laboratory is to help you practice various Matlab-based image filtering techniques on Harris corner detection, SIFT features and segmentation using K-means clustering.

Notes:

- There are 4 tasks, including a prelab task.
- The pre-lab task should be completed BEFORE students attending the laboratory. Students need to show them to your tutor to get marks during the first week's session. Any pre-lab work done during the sessions would not be marked.
- Your marks will be determined based on both your lab demo and your lab report.

Pre-lab Tasks (to be completed before the first lab):

(1) Explain the effect of the kernels listed below. What do they do on an image?

-1	0	1
-1	0	1
-1	0	1

(a)

-1	-1	-1
0	0	0
1	1	1

(b)

(2) For a second moment matrix (or autocorrelation matrix) H $= \begin{bmatrix} h_{11} & h_{12} \\ h_{12} & h_{22} \end{bmatrix}$, what is the Harris cornerness value R?

(3) Explain briefly how the Laplacian of Gaussians can be used in determining the relevant scale (that is the sigma value in a Gaussian kernel) around a corner feature (called a keypoint).

Task-1: Harris Corner Detector

1. Download and open the ***clab3_task1.m*** file. Read and understand the codes. You might need to use ***help*** command to understand the Matlab functions used.
2. **Comment** on every line of the code.
3. **Implement** two missing parts in the Matlab file, which are on

a) The Harris cornerness (R) calculation by utilising a Gaussian weighted (convoluted) 2nd moment matrix (H) (see Lecture 4A)

b) The non-maximum suppression by using a 11-by-11 mask to find the maximum value within the mask (non-maximum values will be suppressed to zero accordingly)

4. Test the function on the provided four test images (on Wattle)

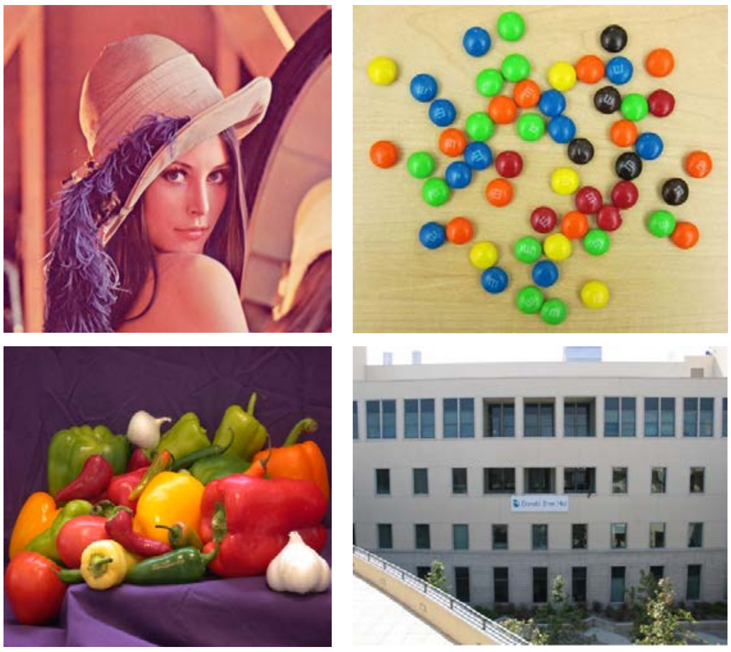

Figure 1. Four test photos for Harris corner detector (Lenna, M&M (top), Peppers, and Right (bottom))

5. Display your results by overlaying the detected corners onto the image.

6. Compare your result with Matlab's inbuilt ***corner()*** function.

Task-2: Image Matching using SIFT Descriptors

1. Read in one of your frontal face images, and generate three rescaled and rotated versions. For example, you may rotate your image by 3 degrees, 45 degrees, and 90 degrees, and rescale them by a scale-factor of 1.2, 1.4, and 0.8, respectively. Save all the three new images to your local directory.

2. Read, understand and test the provided ***SIFT.m*** code on your three images. Use provided ***showkey.m*** to display the SIFT key-points detected on each image. (Note: you need to copy all the provided files to your local directory)

3. Display two images side-by-side on a 1x2 image panel as shown in Figure 2 below. Compute their SIFT feature points, find feature matches between the two images, and draw lines connecting the matched SIFT feature points. In the step of drawing lines, in order to avoid cluttered visualisation, you may only show the lines for the first 10-15 SIFT matches as in Figure 2.

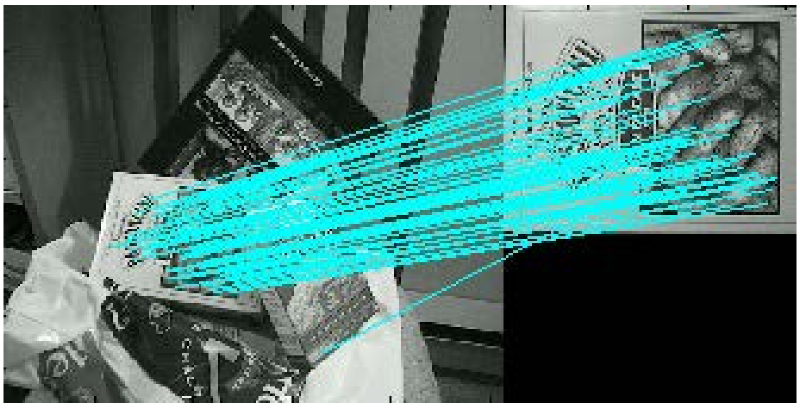

Figure 2. An example of SIFT matching results

4. Hint: In order to match two SIFT features, you need to exhaustively test for every SIFT feature in the first image, and try to find its best match in the second image. This is done by comparing the Euclidean distance between the two 128-dimensional SIFT feature descriptors. Two SIFT descriptors are found to be a "matching pair" if and only if their distance is less than a ratio times the distance to its second closest match (that is for the sake of non-maximum suppression and avoiding ambiguous matches). This ratio is usually set between 0.6 - 0.8.

Task-3: Colour Segmentation using K-means Clustering

1. In this task, you are asked to implement your own K-means clustering algorithm for colour image segmentation, and test it on the following two images in Figure 3. Please note that the PNG-type (portable network graphics) images are in 48-bit format, and you need to first convert them to 24-bit colour images.

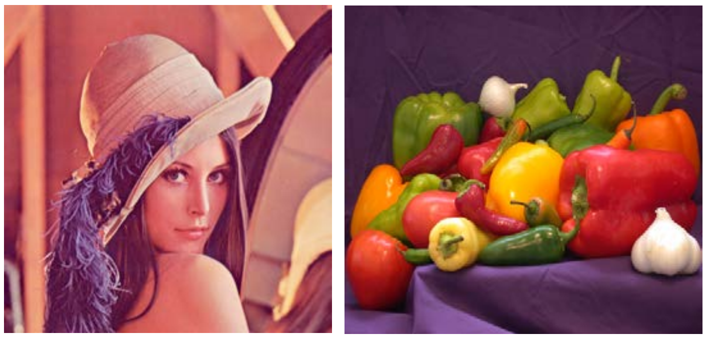

Figure 3. Two test images for colour segmentation (Lenna and Peppers).

2. The following example illustrates a typical k-means segmentation result, where the colour feature vectors used were [L^*, a^*, b^*, x, y] where

   - L^* represents the lightness of the colour,
   - a^* represents the colour position between the red and green,
   - b^* is the position between yellow and blue
   - x and y are the pixel coordinates.

   Your segmentation results should look similar.

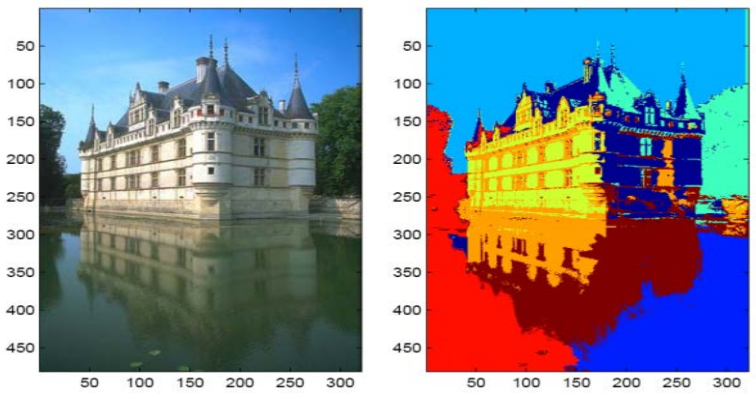

Figure 4. An example of colour segmentation

3. Download and open the ***clab3_task1.m*** file. Read and understand the codes. Your task is to complete the codes, and test the images. Note: your solution must be based on the provided supporting functions.

Assessment Requirement for CLAB3:

1. Demonstrate your programs to tutor, and answer tutor's questions if any.
2. Upload a single PDF-file by the due date. You must use the following file name:
   (a) CLAB3_Report_UID.pdf (replacing UID with your student ID)
3. Your PDF file must contain the following contents
   (a) A PDF report containing sample results with necessary comments and descriptions.
   (b) Provide your answers to the questions in the tasks, if any.
   (c) Implementation of Matlab codes with explanation.

============= End of CLab ===========