Robot Vision [06-25024]
Summative Assignment 1

March 11, 2022

Submission deadline:	**12:00pm (BST), Friday, 18 March 2022**
Instructor:	Dr. Hyung Jin Chang
	Prof. Hamid Deghahni
	Dr. Jianbo Jiao
Total marks:	100
Contribution to overall module mark:	25%
Submission Method:	This assignment must be submitted through Canvas.

Important note on plagiarism

Plagiarism can have very serious potential consequences. Please see `https://canvas.bham.ac.uk/courses/47172/pages/computer-science-programme-information?module_item_id=1648181` for further information. Plagiarism includes failure to provide proper attribution for ideas originating from external sources and copying from other students or external sources. Changing the wording of the copied text is still considered plagiarism. It is acceptable and even encouraged, to discuss course content and assignments with your fellow students. However, you must write your answers to all assignments individually and you must not share those answers with other students.

Instructions

Your answer must be submitted to Canvas before the deadline in the form of **a single zip archive file** containing:

1. Your answers to the questions in prose and diagrams. This should take the form of a single PDF document with the answers to each question. You may choose any editor to create the PDF document, but it is highly recommended to use the provided LaTeX template.

2. Your code and any accompanying files necessary to execute the code for any programming questions as specified in the LaTeX template.

and **a separate PDF document** with the answers (two files in total; one zip file and one PDF file). Some or all of the text of each question is emphasised using *italics*. This emphasis indicates a question that must be explicitly answered or a task that must be completed.

Part 1

Question 1.1 **[8 marks]** *List the similarities and differences between passive stereo 3D coordinate measurement and active 3D coordinate measurement. Give two situations in which passive methods perform better, and two situations in which active methods perform better, and explain why.*

Question 1.2 **[8 marks]**

Consider the following image matrix and kernel (Figure 1). Produce the resulting convolved image matrix for the central dashed region. What affect does this kernel have? (Consider both this image and larger example images)

Kernel

-1	-1	-1
-1	8	-1
-1	-1	-1

*

Image

121	109	125	115	103
155	78	118	112	178
11	6	18	13	16
7	7	22	16	26
3	7	17	18	17

Figure 1: A Kernel convolved with an Image

Part 2

It is required to submit your code as username_assignment1_part2.m. In the absence of the source code this part will not be graded. The image files that will be used in this part are `Blue_Marble.jpg`, `Chicago_Downtown_Aerial_View.jpg`, and `Malards_in_Golden_Gate_Park.jpg`.

Question 2.1 **[8 marks]** *Implement a function to apply any difference of Gaussian filter to an image. It should be possible to give the size of the filter in pixels, and the parameters of the two Gaussian, as inputs to the function. You may use existing MATLAB functions for filtering but must write your own code to generate difference of Gaussian kernels. Implement a default set of parameters for the two Gaussian that approximates the Laplacian of the Gaussian in the case where the only input given to the function is the size of the filter in pixels. Apply a 15 × 15 difference of Gaussian filter to the accompanying three images (see above) using your function and show your results.*

Question 2.2 **[8 marks]** *Implement a function to apply difference of Gaussian filters to an image at different scales to produce a set of filtered images forming a scale-space pyramid. Apply your function to the accompanying three images (see above) and show the different scales produced by your function.*

Question 2.3 **[9 marks]** *Implement a function to search a scale-space pyramid and identify strong filter responses. Assume that each of the strong filter responses corresponds to a feature of interest in the image. The function should return the properties of each of the features, which should consist of the position and scale that maximises the filter response. Apply your function to the accompanying three images (see above) and show your results by overlaying the feature positions and sizes on the original images.*

Question 2.4 **[9 marks]** *Implement a function to examine the image neighbourhood of a feature and estimate the orientation of the feature. The scale of the feature should be taken into account when estimating the orientation. Apply your function to the accompanying three images (see above) and show your results by overlaying the feature positions, sizes, and orientations on the original images.*

Part 3

It is required to submit your code as username_assignment1_part3.m. In the absence of the source code this part will not be graded.

Question 3.1 [**10 marks**] *Load the* `checkerboardPhoto.png` *and* `circlePhoto.png` *images. Convert them to grayscale. Write a function,* ***convolve****, which takes a filter, F, and an image, I, as input and calculates the 2D convolutions of I with F via the use of 'for' loops. Zero-pad the image within the function to ensure that the filtered image is the same size as the original image. You may assume that F always has an odd value for its height and width. DO NOT USE the existing conv, conv2, convn, filter, or filter2 functions. Apply a 15×15* ***Gaussian filter*** *with a standard deviation of 5 to both images using the* ***convolve*** *function. Show the results.*

Hint: The built-in function ***padarray*** *is useful for zero-padding.*

Question 3.2 [**7 marks**] *Apply a 15×15* ***box filter*** *to the loaded images using the* ***convolve*** *function. Show the results.*

Question 3.3 [**9 marks**] *Implement a nonlinear filter — a* ***median filter****. While the Gaussian filter works by computing locally-weighted values of the signal, the median filter first sorts the signal values within a window and then takes the middle value (i.e., the median).*

Implement a function, ***simpleMedian****, which takes as input an image, I, and the dimensions of the median filter, W and H, and returns a median filtered image. Zero-pad the image within the function to ensure that the filtered image is the same size as the original image.*

Apply a 15×15 ***median filter*** *to the loaded images. Show the results.*

Question 3.4 [**12 marks**] *Load the* `Malards_in_Golden_Gate_Park.jpg` *image from part 2. Convert it to grayscale. Corrupt it with:*

i. Salt and Pepper Noise (noise density = 0.075):

ii. Gaussian Noise (mean = 0.1, variance = 0.15):

Figure 2: Examples of Noise

Examples of the two types of noise are shown in Figure 2 Then examine the effects of your Gaussian, Box, and Median filters in turn on the two types of noise. Display the results in a 2 × 4 figure with subplots, with the noise type changing between the rows and the columns displaying the filter used (no filter, Gaussian, Box, and Median). The layout is shown in Figure 3.

Which filter performs the best on each noise and why? Explain what you see.

Hint: The built-in function ***imnoise*** *is useful for adding noise.*

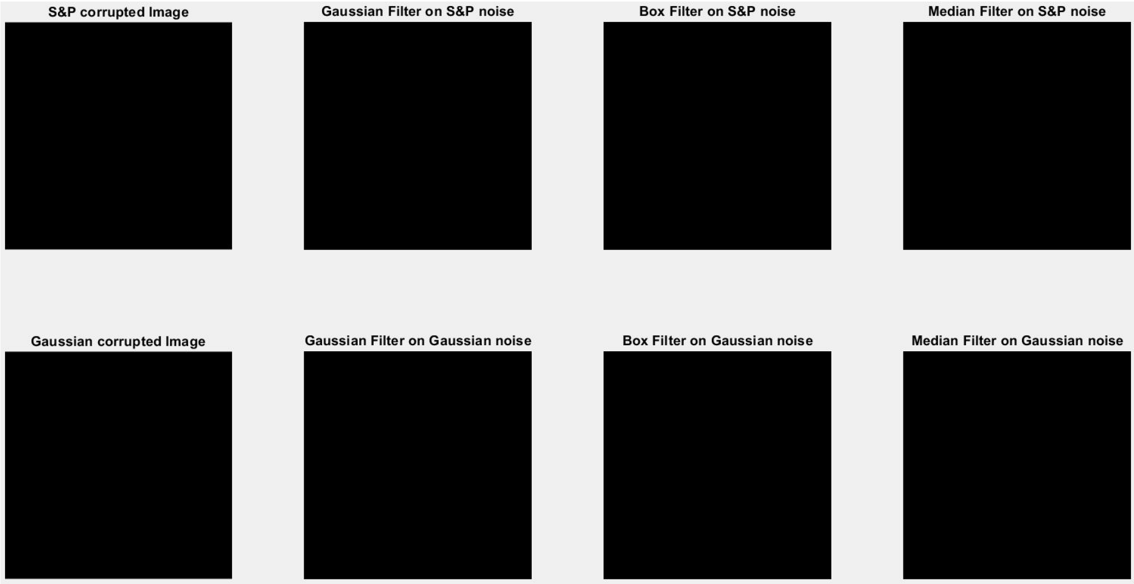

Figure 3: Example Layout for 3.4

Part 4

It is required to submit your code as username_assignment1_part4.m. In the absence of the source code this part will not be graded.

Question 4.1 [**12 marks**] *Use the Circular Hough transformation (**CircularHough_Grd.m**) for circle detection on the dice images. Your task is to implement a code that detects and counts the number of detected dots on the given dice images:* `dice1.jpg`, `dice2.jpg`, `dice3.jpg`, *and* `dice4.jpg`.
*Hint: Use **hold on** and **plot** functions to display the detected circles on the image.*

(a) Original image (b) Detected dots

Figure 4: Examples Results