COMP 558 Assignment 1

(Maximum Score: 60 points)

Prepared by Prof. Kaleem Siddiqi

Posted: Thursday, Sept 19, 2019

Due: Sunday, October 6, 2019 (by midnight, 11:59pm)

Introduction

This assignment covers the material up until Lecture 5. The questions concern basic image processing using convolution, and the core ideas of the edge detection method of Marr and Hildreth.

In order to work on this assignment, you need to know how to use Matlab and be used to indexing conventions for matrices and plots and the various image processing commands. If you do not yet know Matlab, then learning these tools will take some time and will be part of the work that you do in the assignment. You can always look at Matlab's documentation for help. The Mathworks website provides webinars and tutorials for various computer vision/image processing tasks.

Please use the mycourses discussion board for any assignment related questions. This will help us reach the entire class when we respond. However, follow appropriate protocol, e.g., do not reveal the answer to a particular question or ask if your proposed solution is correct. You are also free to discuss the questions with each other. *However, the solutions that you submit must reflect your own work and must be written by you.* You are not permitted to copy code or text used in your answers from one another or from any third party sources including internet sites.

Instructions

Submit a single zipped file **A1.zip** to your mycourses Assignment 1 folder. The zip file must contain:

- A PDF file with figures and text for each question. We suggest that you do not spend time with fancy typesetting. Just make sure that your explanations are clear and that the figures and their captions are easy to interpret and that you answer each part of each question.
- The Matlab code that you wrote so that we can run it if necessary.
- The images that you used.

In order to receive full points, your solution code must be properly commented, and you must provide a *clear and concise* description of your computed results in the PDF. Note that the TAs have limited time and will spend at most 20-30 minutes grading each submission.

Late assignment policy: Late assignments will be accepted up to only 3 days late and will be penalized by 10% of the total marks, per day late. For example, an assignment that is submitted 2 days late would receive a maximum of 80%.

Question 1: Theory (15 points)

a) [**Convolution Theorem**] (**5 points**) Consider a linear time invariant system with input $I(x)$ and impulse response $h(x)$.Recall that the impulse response is defined as the output of the system when the input is an impulse $\delta(x)$. Prove the convolution theorem, which states that the output of this system is then given by $h(x) * I(x) = \sum h(x-u)I(u)$.

Hint: Consider that $I(x)$ can be represented via an expression that involves the function $\delta(x)$.

b) [**Laplacian Operator**] (**5 points**) Consider a 2D image $I(x,y)$ and its Laplacian, given by $\Delta I = I_{xx} + I_{yy}$

Here the second partial derivatives are taken with respect to the directions of the variables x, y associated with the image grid for convenience. Show that the Laplacian is in fact rotation invariant. In other words, show that $\Delta I = I_{rr} + I_{r'r'}$, where r and r' are *any* two orthogonal directions.

Hint: Start by using polar coordinates to describe a chosen location (x,y). Then use the chain rule.

c) [**Condition of Linear Variation**] (**5 points**) Marr and Hildreth argue that to detect edges the zero-crossings of a second derivative taken in the direction perpendicular to the local direction of the edge should be used. They then use a non-oriented operator, the Laplacian, to detect these zero-crossings, arguing that under a certain condition this is mathematically sound. What is that condition? State it and write down what it means mathematically. Prove that their argument is mathematically sound, that is, under the condition they assume, the Laplacian picks up the direction perpendicular to the local direction of the edge.

Hint: Refer to the Marr-Hildreth paper, but fill in the necessary mathematical detail: http://rspb.royalsocietypublishing.org/content/207/1167/187

Question 2: Implementation (20 points)

a) [**2D Gaussian**] (**5 points**) Implement a Matlab function that returns a 2D Gaussian matrix which can be used for filtering an image. Here is the template for this function which you must use:

```
function g = make2DGaussian(N, sigma)

%  N is assumed to be odd, and so the origin (0,0) is
positioned at indices
%  (M+1,M+1) where N = 2*M + 1.
end
```

Attempting this question will help you understand what a 2D Gaussian is, and how matrix indices are defined in Matlab. In particular, the definition of convolution in Matlab is slightly different from the one given in class. https://www.mathworks.com/help/matlab/ref/conv.html The reason for the difference has to do with Matlab indices starting at 1 rather than 0.

Visualize this 2D Gaussian for two choices of `sigma`, choosing `N` appropriately for each case, using a suitable surface plotting function in Matlab.

b) [**2D Laplacian of Gaussian**] (**5 points**) Implement a Matlab function that returns a 2D Laplacian of Gaussian matrix. In class we reviewed the expression for this function which is the basis for Marr-Hildreth edge detection. Here is the template for this function ,which you must use:

```
function g = make2DLOG(N, sigma)

%  N is assumed to be odd, and so the origin (0,0) is
positioned at indices
%  (M+1,M+1) where N = 2*M + 1.
end
```

As with part a) visualize this 2D Laplacian of Gaussian for two choices of `sigma`, choosing `N` appropriately for each case, using a suitable surface plotting function in Matlab.

c) [**2D Gabor Filters**] (**10 points**) Implement a Matlab function that returns a 2D Gabor filter. Whereas we did not explicitly discuss this function in class, there is good Matlab documentation to help you. Here is the template for this function , which you must use:

```
function [even, odd] = make2DGabor(N, lambda, angle)

%  N is assumed to be odd, and so the origin (0,0) is
positioned at indices
%  (M+1,M+1) where N = 2*M + 1.
```

```
% lambda : wavelength of the Gabor filter
% angle : orientation of the Gabor filter
% Set sigma of Gaussian part of filter = wavelength lambda
end
```

Visualize both even and odd Gabor filters for 3 different wavelengths and 3 different orientations of your choice.

Question 3: Edge Detection (25 points)

For this part you will use both the `Paolina.jpg` image we have provided and one additional image of your choice for experimentation, e.g., one taken by your phone. For the latter make sure you convert an RGB image to just a single intensity channel, e.g, by using the heuristic `rgb2gray` method in Matlab. If you read the documentation you'll see that this method uses a formula very similar to the one we discussed in the lecture on color:

```
I = 0.2989 * R + 0.5870 * G + 0.1140 * B
```

a) **(5 points)** Convolve the 2D Laplacian of Gaussian filters from Question 2 b) with the input images, using the Matlab function `conv2`. Remember that convolution is commutative. However, when one filters an image, one needs to choose how to handle pixels at the image boundary and to decide how big the resulting filtered image should be. For your implementation, we ask you to treat the original image as being zero padded outside its domain and that the output be the same size as the input image (the first argument to `conv2`).

b) **(5 points)** Next find a way to detect zero-crossings of the results of these convolutions and display these results as images. In principle you will have edges at different blurring scales. Examine the locations of these edges both by showing them as line drawings and by overlaying them in a chosen color (e.g. yellow) on the original images. [You might want to verify your results by also trying Matlab's `fspecial` function, which allows you to create predefined filters of a variety of types (see the documentation).]

c) **(5 points)** Discuss your multi-scale edge detection results and in particular identify situations in which they appear to be successful versus those where they fail. For this you might want to consider the assumptions made in Question 1 c) and you might want to crop and zoom-in on specific regions of an image to show detail

d) **(10 points)** Repeat the analysis from parts 3 a) through 3 c) but now using odd Gabor filters of different wavelengths and orientation = 0, 45 and 90 degrees, instead of the Laplacian of Gaussian filters. For this part of the question a pixel should be detected as lying on an edge if it is a zero crossing in **any** of the orientations of the filter response. How do the edge detection results compare to results from the results in 3 b) and c)? Comment on the relation between the wavelength parameter of the Gabor filter and the `sigma` in the Laplacian of Gaussian filter.

Get started early! Good luck, and have fun!