COMP 546 Assignment 1

Prepared by Michael Langer

Posted: Friday, Jan. 21, 2017
Due: Friday, Feb. 3, 2017 (midnight)

General Instructions

For clarification questions, please use the mycourses discussion boards. You can send emails to the professor or TA, but if the question is of general interest then we may ask you to post it on mycourses so that everyone can benefit from the answer.

David will not hold fixed office hours. Instead, he will be available to meet by appointment. *This is intended to give you more flexibility in reaching him, so please use it.* To make an appointment to meet him, please send him email at david.bourque@mail.mcgill.ca. His office is ENGMC 336/337.

Submit a single zip directory **A1.zip** to the myCourses Assignment 1 folder. The directory should contain all of your files, including the Matlab files in the posted code. Your program(s) should run in that directory. It should also include a PDF file with your answers. *For any figure/image that you include in your answer, indicate how and where in your code this figure was computed so that David can quickly verify if he wants to.*

David will grade the assignments. If you have any issues that you wish him to be aware of, then include them as a comment with your submission.

**Late assignment policy:** Late assignments will be accepted up to only 3 days late and will be penalized by 20 points per day. The assignment is out of 100 points.
Introduction

In this assignment, you will simulate some of the image “filtering” operations that are performed in the retina. This will help you get a feel for what these operations achieve. Please note that although the lectures discussed spectra (emission, reflection, absorption) and the relationship between continuous spectra and discrete spectra (RGB, LMS), here we only deal with RGB. We will ignore continuous spectra and we will make the (erroneous) assumption that RGB and LMS are equivalent. This will help us keep the computations simple.

Let’s get started! The Matlab program A1.m reads in a color image and decomposes it into R, G, and B channels. Each RGB pixel then is transformed to a local contrast as follows: we define the intensity at each pixel to be (R+G+B)/3, then we subtract the local mean intensity from the pixel, and then we normalize the intensity by dividing by the local mean intensity. In the lectures, I did not discuss the normalization step, but it is there in the retina and believed to be important.

[One technical aside: subtracting the mean intensity from a pixel is formally equivalent to taking the inner product (i.e. sum of pointwise products) with a DOG, where the center Gaussian has standard deviation 0 and the surround Gaussian defines the local neighborhood for computing the mean. If we blurred the image slightly before subtracting the local mean, we would have a DOG similar to the one defined in the lecture.]

Questions (10 points each)

1) Write a matlab function makecheckerboard.m that returns an N x N RGB image of a checkerboard pattern, as follows. Each square of the checkerboard should be at least 10 pixels wide, and the grid of squares should be at least 10 squares x 10 squares. The RGB values should be constant within each square of the checkerboard (but R, G, B should be chosen independently as mentioned in the FAQ), and the values should vary randomly between squares.

Next, multiply the RGB values in the right half of the pattern by a small constant, say 0.2. Scaling down the intensities makes it more difficult to visually distinguish neighboring squares in the right half. The effect is similar to what occurs when a shadow is cast on a surface. It is as if the checkerboard pattern is a grid of surface patches, and the right side receives and hence reflects 20% as much illumination as the left side.

This question is not about vision. It is merely here to give you some practice working with matrices in Matlab. In particular, you can index a block of a matrix I using I(i1:i2, j1:j2). In Matlab, it is better to assign values as blocks when you can, and to avoid for loops which are notoriously slow in Matlab.

Hint: see makeImageSimultaneousContrast.m for an example of how to make an image.
2) Vary the sigma that is used for computing the local average, and examine how changing the sigma affects the local contrast of the shadowed checkerboard. Are there values of sigma for which the local contrast counteracts the effect of the shadow, for example, so that the local contrast allows one to better compare the values of shadowed squares?

Questions 3-6 below do not require RGB images. A single grey channel is sufficient.

3) Consider a simultaneous contrast display such as:

![Simultaneous Contrast Display](image)

The left grey square appears to be slightly brighter than the right one, although in fact the two grey squares have the same RGB values. Can local contrast explain this illusion, in the sense that the local contrast image is consistent with this illusion? Justify your answer by computing the local contrast and comparing values across the image.


It is traditionally argued that local contrast cannot explain this illusion. Explain why, by computing the local contrast on the given image Whites_illusion.jpg and examining the values.

5) Define a DOG using two Gaussians whose sigmas are very similar. Note that the values of the DOG will be near zero everywhere, so you may wish to rescale the DOG weights (although this has no important effect on your answer).

Use the `filter2` function to cross correlate your DOG with an image that consists of a sequence of alternating stripes (e.g. black, white, black, white, ….), all of the same width. How does the peak amplitude of the response of the DOG vary with the width of the stripes? (To answer this question, you will need to repeat for a wide range of stripe widths.) What is the width of the stripes that gives the largest (peak amplitude of) response, and how does this ‘optimal’ stripe size compare to the size of the sigma(s) used in defining the DOG?
Note that for this question (and the next one), we are not asking you to normalize the responses i.e. as we did to compute local contrast.

6) Using the same DOG as in Question 5, consider the response to a single square on a uniform background. Vary the size (width) of the square and plot the peak response amplitude of the DOG as a function of the square size. Which size square gives the largest peak response? How does this ‘optimal’ size compare to the size of the sigma(s) used in defining the DOG?

7) Define an opponency channel R-G that compares red and green. (We could define a blue-yellow opponency channel too, but for simplicity we’ll just deal with red-green.) For each RGB pixel, it computes the difference of the R and G values.

One can normalize these differences similar to what was done with local contrast. e.g. dividing by:
- the intensity (R+G+B)/3 at that pixel,
- the local average intensity in some Gaussian defined neighborhood (and in that case, how big a neighborhood should one choose?),
- the local average of R-G in some Gaussian defined neighborhood.

What might be the goal of normalizing the R-G values? Which of the normalization methods just mentioned would be better for achieving such a goal?

Hint: think about the hue, saturation, value properties of color. In particular, hue is the direction on the color wheel and is independent of value. See lecture 4 slides 18-23.

Give an example to illustrate your argument. For simplicity, use random checkerboard images for which B=0 everywhere.

8) Another way to define an R-G opponency channel is to use a DOG, namely a Gaussian in the R channel and a (possibly different size) Gaussian in the G channel, cross-correlate these R and G Gaussians with the R and G image channels respectively, then subtract one from the other, and then normalize. If the sigmas are the same for R and G, then the situation is similar to the previous question. Instead, take the case that the sigma for R is less than the sigma for G. Characterize the receptive field of such a DOG cell in terms of its spatial and color properties. For what image patterns would one expect to yield a large response?

9) Give an example of a real RGB image (e.g. JPG) that contains objects of various colors and sizes. Cross-correlate the DOG in Question 8 with the image, and show that one indeed obtains a large response at the locations you expected for your image.
10) In the previous two questions, the cross-correlation values (responses of DOG cells) could be either positive or negative, corresponding to one opponent color dominating over the other. Neurons cannot have negative responses, however. Positive only responses are often modelled using half wave rectification. Using such a model, show two images to indicate regions where R dominates over G, and G dominates over R. For example, for the R dominant image, the pixels where R dominates over G should be bright (white) and the remaining pixels should be dark (black).

Good Luck!