COMP 546 Assignment 1

Prepared by Michael Langer

Posted: **Tues. Jan. 22, 2019**
Due: **Tues. Feb. 5 at 23:59**

General Instructions

For clarification questions, please use the mycourses discussion boards. You can send emails to the professor or TAs, 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.

TA's Office Hours and Location will be posted under Announcements on mycourses. TA's are Sandy Wong (sandy.wong@mail.mcgill.ca). and Arna Ghosh (arna.ghosh@mail.mcgill.ca).

Submit a single zip directory **A1.zip** to the myCourses Assignment 1 folder. The directory should contain all of your files, namely a PDF file with your answers, including figures, images, and explanations, as well as the Matlab files in the posted code. The Matlab files should be labelled sensibly e.g. **Q1.m, Q2.m, Q3.m**.

The TAs will grade the PDF file. For any figure that you include in your answers in the PDF, the code you used to generate that figure should be part of your submission. The TAs will not have time to go through your code, but if an issue arises afterwards where you wish to discuss your grade, then you need to be able to demonstrate how you obtained the figure.

We suggest that you use simple word processing software (Word or google docs) rather than latex. Latex is great for publishing, but one can waste a lot of time with latex figures. You will be graded more on the content than on the presentation style (although if the presentation is a mess to the point that TAs having trouble understanding, then you will be penalized).

You may use a language other than Matlab, if you wish. However, we do not guarantee that the TAs can help you if you have issues with that language, and if there are problems with your solutions then the TAs may not look at your code when grading.

Late assignment policy: Late assignments will be accepted only up to 3 days late and will receive a late penalty. The default will be 10 percent of your grade per day. For example, if your grade is 70/100 and you submit two days late, then you will receive a grade of 56/100. Note: If your assignment is 1 minute late, then myCourses will mark it as late, and we reserve the right to consider it to be 1 day late. So we strongly advise you to submit well before the deadline.

Introduction

In this assignment, you will work with RGB images and you will simulate some of the image filtering operations that are performed in the retina. This will give you 'hands on' experience with these operations, and also familiarize you with some key concepts, such as color and contrast.

The lectures discussed continuous light spectra and LMS cone responses, but we will keep it simple and just deal with RGB. Moreover, we will also use only simple synthetic images, rather than real images captured with a camera. [1]

Question 1: RGB images in Matlab (30 points)

In this question, you will familiarize yourselves with the basics of making color images in Matlab. You will make and display an RGB image in Matlab using a 3D matrix. Note that in Matlab, matrices are indexed by (row, column) which is what you learned in your linear algebra course. Note this is the opposite to what is used in other places in mathematics e.g. $I(x, y)$ generally means that (x, y) refers to horizontal and vertical, i.e. (column, row). Sometimes in Matlab the latter is used too; so be aware of this issue.

a) Create an RGB images that consists of three horizontal stripes (top, middle, bottom) which have the following properties:

   - the top stripe goes from bright red at the left image border to black at the right image border
   - the middle stripe goes from bright red at the left image border to middle grey at the right image border.
   - the lower stripe goes from bright red at the left image border to bright green at the right border.

   Within each stripe, the intensities in each R, G, B channel must depend only on x position, in particular, the intensities in each stripe must be of the form $I(x, y) = ax + b$, where a and b are constant (possibly 0).

b) Create your own anaglyph image which consists of two halves. The upper half has a white background and should have several small squares whose disparities correspond to depths that are closer than the background. When viewed through anaglyph glasses, the squares

---

[1] There is a good reason for this. When you capture an image with a digital camera, the RGB values typically have been processed already. They are *not* proportional to the filtered light intensities measured by the camera sensor, but rather they have been non-linearly transformed by the camera's image processing hardware. Details are beyond the scope of this course.

should appear to lie *closer* than the background, the depth of each square should be constant, and the depths should *decrease* (getting closer to observer) from left to right. You will need to choose appropriate colors for the squares.

The lower half consists of a black background. It should have several small squares with disparities corresponding to depths that are *further* than the background, and the depth of the squares should *increase* from left to right. You will need to choose the color of your squares in the upper and lower half, so that they are visible, given their background.

Anaglyph glasses are available for loan from the instructor.

Question 2: Local Contrast (20 points)

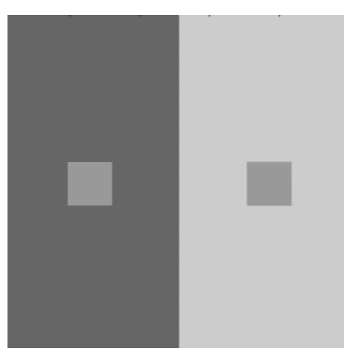
Given any original RGB image, let's define a few other images:

- an *intensity* image is obtained by taking the average of the RGB values at each pixel, that is, (R+G+B)/3. From here on in this question, we will deal with grey levels (intensities) only.

- the *local mean intensity* image is obtained by blurring the *intensity* image with a Gaussian. You may use the Matlab `imgaussfilt.m` function. You will need to choose a sigma of this Gaussian.

- the *local contrast* image is obtained by taking the intensity image, subtracting the *local mean intensity* from each pixel (which can result in either a positive or negative value), and then dividing by the *local mean intensity*. The division can be considered a normalization.

Consider the image shown below. The two halves are dark grey and light grey, and the two small squares within the two halves have identical grey values. However, the left grey square appears to be slightly brighter than the right one, even though the two grey squares have the same RGB values. (Note that for this original image, the RGB channels at each pixel are all of the same value, and so we only need to deal with intensity from the start.)



Some have argued that this illusion that the two grey squares have different values is due to local contrast. That is, we do not perceive absolute intensity at each point, but rather we perceive relative intensities, namely relative to what is nearby in the image.

Is *local contrast* (as defined above) consistent with the perceptual effect that the squares don't appear equally bright? Justify your answer by making an image such was shown above, and computing the local contrast and comparing values across the image. Note that the values of local contrast can be positive or negative, so be sure to use the `colorbar` command to indicate what the values are.

In your answer, compare the effects of choosing different sigma values for the Gaussian.

Question 3: DOG filtering of spatial patterns (50 points)

a) Define a 2D difference of Gaussian (DOG) function using two 2D Gaussians whose sigmas are very similar. The cell should be on-center off-surround. There are several ways you can do this, and feel free to use Matlab's built in libraries.

   Consider again the image which you made for Q2. Choose the DOG such that the on-center is roughly the same size as the grey squares. Convolve this image with your 2D DOG function and show the result. Briefly discuss whether your result is consistent with the illusion by comparing the computed values at the pixels corresponding to the left versus right squares.

   In your answer, be sure to indicate the size of the Gaussians you used to create the DOG and the size of the grey square and image overall.

b) A red-green *double-opponent* cell can be modelled by using a DOG in the R channel and a DOG in the G channel, such that the sizes of the two DOGs are the same but *the signs are opposite*. For this question, let's say the DOG in the R channel is on-center off-surround and the DOG in the G channel is off-center on-surround. Use the same DOG sizes as in (a).

   Verify that such a double opponent cell produces zero output when it is convolved with the image you made for Q2.

   Now create a new image which is similar to the lower stripe of the image you made in Q1. This image should go from bright red at the left image border to bright green at the right border. Add in grey squares to this image, one on the left side and one on the right side. The square sizes should be roughly the same size as the center of the DOG. Show this image, and show the output of the double opponent cell when convolved with this image. Note that each of the RGB channels of the image and DOG will need to be convolved separately and the results should be summed together.

Briefly describe any perceptual effects you observe when look at your image. (Do the grey squares appear grey?) Relate those perceptual effects to what your double opponent cell computes.

c) Here we go back to grey level only. Take the DOG that you used in (a) and convolve it with a grey level image that is a raised 2D sinusoid.

$$I(x, y) = \frac{1}{2}\left(1 + \sin(\frac{2\pi}{N}\ k\ x)\right).$$

Repeat this for many different spatial frequencies k, and for each k compute the root mean square (RMS) values of the filtered image. Plot these RMS values *as a function of spatial frequency* k. Which spatial frequency gives the largest RMS filter outputs? How does the wavelength of this 'optimal' spatial frequency compare to the size of the center and surround regions of DOG ?