

MIT OpenCourseWare  
<http://ocw.mit.edu>

9.35 Sensation And Perception  
Spring 2009

For information about citing these materials or our Terms of Use, visit: <http://ocw.mit.edu/terms>.

9.35 Spring 2009  
Problem set 4– 100 Points

For formatting clarity, descriptions of the problem are in plain text, conceptual questions are in bold, and Matlab instructions are in italics.

### **Problem 1 – 100 Points**

We have a remarkable ability to perceive white objects as white, gray objects as gray, and black objects as black regardless of how much light they reflect. But, illumination in the real world can span 10 orders of magnitude (from just a few photons to  $\sim 10^{10}$  photons per second in bright sunlight), meaning a piece of black paper under a bright light may reflect more light than white paper under a dim light! Nevertheless, for much of the range of possible illuminations, we can tell what shade any particular object in the world is. Typically we do not even notice changes in illumination across an object!

This may seem obvious to you, but the problem becomes apparent when we try to record what we see: i.e., photographs on film or with a digital CCD. These devices can capture a (small) range of light in a given scene. While a scene under varying illumination may look interpretable to our naked eye, without the benefit of our retina's adaptation and local gain control, photographs in such conditions often turn out to be awful (with some parts under-exposed, and others over-exposed). In this problem, we will explore how one can deal with variation of illumination in photographs, and how this relates to the way our own visual system may deal with the problem.

We will investigate this problem using an image with 16 bits of dynamic range (meaning the intensity of a given pixel can go from 0 to 65535 [ $2^{16}-1$ ]), which is much more than a typical 8-bit jpg (0 to 255). This is a higher dynamic range than most cameras will allow, so images like this must be composed from a series of lower-dynamic-range photographs. For our purposes, we will treat this image as accurately depicting the intensities of light present in the world before the range is compressed and processed by our eyes or a low-dynamic-range camera.

We have provided you with a simple function to display a limited dynamic range of the image, for clarity.

- 1.a) (5 Points) *Load the file 'mdr\_im.mat' into Matlab. Display the image (im) using our viewim function.*

**Which portions of the image are clear? Which are unclear? Why?**

*Display the image with different regions exposed properly. You can do this with the viewim function by providing an additional parameter q, which specifies which luminance quantiles to 'expose'. (Type help viewim for syntax details).*

Show an image that allows you to discern what is outside the window, and one that shows the TV clearly.

**Why is the luminance outside so much higher than the luminance inside? Is this a typical problem, and do you notice it?**

- 1.b) (5 Points) Recall that that luminance is the product of illumination and reflectance:

$$L_{x,y} = I_{x,y} * R_{x,y}$$

That is, the luminance of a pixel (x,y) is equal to the illumination times the reflectance of the object at that point. Light sensors (like photoreceptors and CCDs) capture the total luminosity of a particular point in space. However, that is not the most interesting variable to us: we care about the objects that are reflecting light. Solve this equation for reflectance. Now, let's assume that the visual system responds according to a log function. **Use this fact and provide a simple addition/subtraction equation that gives us log reflectance.**

- 1.c) (10 Points) We have luminance from the picture, and we want to solve for reflectance. To do this, we must devise a way to estimate the illumination. **Describe at least two methods that would allow us to do this (they don't have to be practical, but it would help). What assumptions about the world does each of these methods make? Describe how you would (mathematically) implement these methods (perhaps drawing on what you know from previous experience with convolutions and spatial frequency filtering), and why the algorithms might work.**

- 1.d) (10 Points) We have provided you with a function, filterim, which convolves a Gaussian filter of a given size with an image matrix you provide. **If we use this to estimate illumination, what assumptions are we making about the illumination and reflectance? Are these right? How might they go wrong?**

*Use this function to estimate log illumination, and reconstruct a log reflectance image given your solution to 1.b. Display the log reflectance image using viewim.*

***Note that you must decide whether you want to take the color of the illumination into account (that is, model light as either monochromatic and varying in intensity, or varying in both intensity and color across the image). If you do not want to take color into consideration, use mean(x, 3) to average (r,g,b) values in any matrix x. However, you should still treat the reflectance (and resulting images) as color matrices.***

**Is all of the illumination discounted? Are there any artifacts of this procedure?**

- 1.e) (10 Points) We can define a multiscale homomorphic<sup>1</sup> filtering algorithm like the following:

$$O = w_0 L + \sum_s w_s R_s$$

Here,  $O$  is the algorithm output,  $L$  is the log of the luminance,  $R_s$  is the log of reflectance calculated at the spatial scale  $s$ , and the  $w$ 's are linear weights. Basically, this algorithm will use the procedure from 1.d to estimate reflectance under assumptions of different spatial scales of lighting distribution, and then combine these estimates with one estimate of illumination.

*Implement this algorithm using the same procedure as 1.d. (This means, estimate illumination at different scales, extract reflectance at that scale, and then recombine these estimates with some linear weighting function).*

- 1.f) (10 Points) *Play around with different sets of spatial scales and weight combinations, and display the best resulting output. Also, sum up each calculated log illumination using the same set of weights and display this using viewim.*

*Find a long line in the image where the reflectance is the same, but the illumination varies greatly (i.e., along the ceiling or the floor). Plot the mean (r,g,b) intensity of pixels along this line. You can use a function like `improfile`, `interp2`, or just extract a row or column. Compare this with the original image.*

Note that 'best results' above is ambiguous. You can choose parameters to either 1) make the image as perceptually pleasing as possible, or 2) discount as much illumination as possible, as in making the line profile perfectly flat. **Let us know which optimization technique you used, and why you chose that one.**

(N.B.: Many cameras and image editors implement an algorithm like this. For instance, Google's Picasa calls this Fill Light, Nikon cameras call it D-Lighting, and HP cameras call it Digital Flash)

- 1.g) (10 Points) If done right, the results from 1.e should display properly in the small dynamic range shown by viewim. **Why and how does this work? Identify some regions in your image where the algorithm worked particularly well, and some regions with visible artifacts that do not look right. Why did the algorithm perform well/poorly in these regions?**

---

<sup>1</sup> This means: run a linear filter on a non-linear transformation of the original signal – try Wikipedia for more details.

- 1.h) (10 Points) *Insert two small uniform patches of gray into the original image in different regions with vastly different illumination (i.e., one by the TV, one outside). Rerun the algorithm in 1.e and 1.f (using the best set of scales and weights) on this modified image.*  
**Do the gray patches look the same? Are their values still the same? What illusion does this remind you of?**
- 1.i) (10 Points) *Repeat 1.f on an image of you (but don't do another line profile). You may want to change the scales and weights to optimize the output.* **Does the algorithm work as well on your picture? Why or why not?**
- 1.j) (10 Points) Which algorithms (and parameters) work 'best' at extracting reflectance depends on the assumptions the algorithms make about the world, and how close these assumptions are to the truth . **What are the assumptions of the classic 'Retinex' algorithm as described in class (differentiate, threshold, then integrate)? What are the assumptions of the algorithm we used above? What would "classic" Retinex have done better? Worse?**
- 1.k) (10 Points) **In your own words, what problem have we solved? How did we solve it? How is this similar to what happens in the brain?**

**Problem 2**

How many hours did you spend on this assignment?