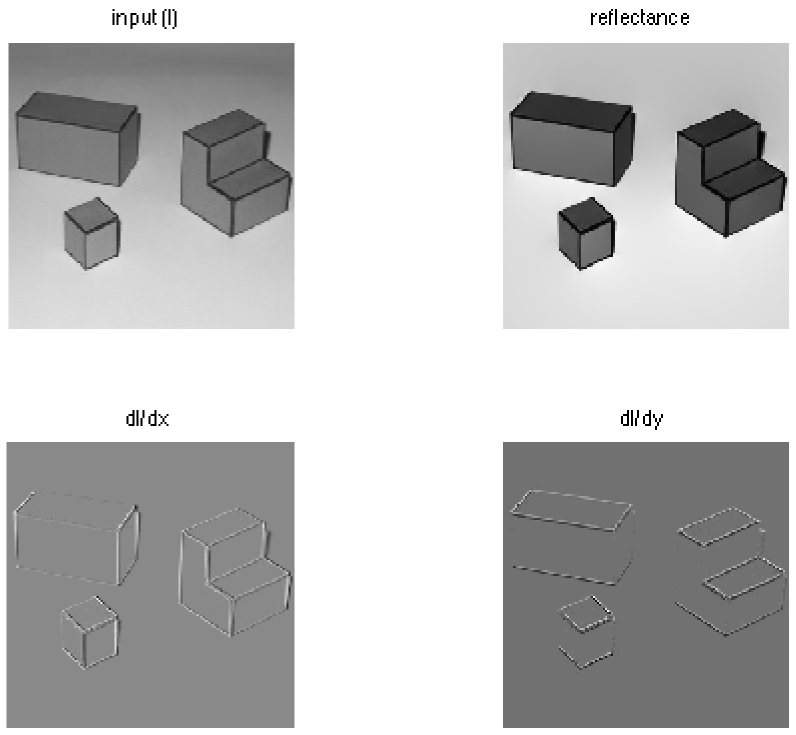
6.869 Advances in Computer Vision
Andrew Moran — Oct 2, 2014 (1:00pm)
Collaborators: None — Problem Set 4

Problem Set 4

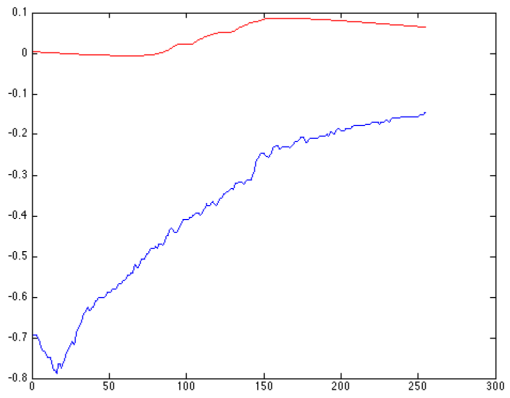
Resources with corresponding images and code are on Stellar under `andrewmo@mit.edu`. The files are in the `pset4.zip` folder. To reproduce the below figures, run `retinex.m` and `pset4main.m`.

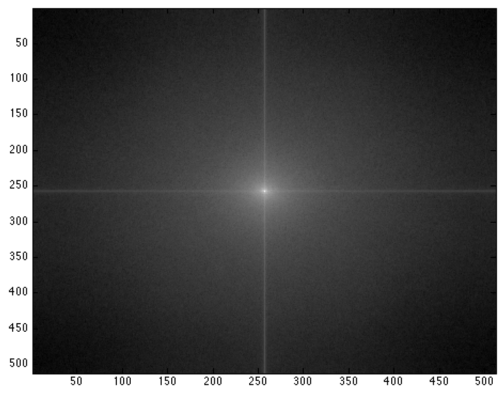
Problem 4.1

For the Retinex algorithm, we convolve the original image with the filters $[-1, 1]$ & $[-1, 1]^T$ to get the image derivatives. Then we apply a threshold in each direction to keep the higher values of the filtered images and zero out the lower values. I chose the threshold to be twice the avererage absolute value in the image. After deconvolving, we get a nice representation of the reflectance image shown below.



To better show the effects of the Retinex algorithm, below is the log of a 1D column (column 150) of the original and reflectance images. We see that the gradient due to the illumination disappears and the plot for the reflectance image is lower and smoother than the original.





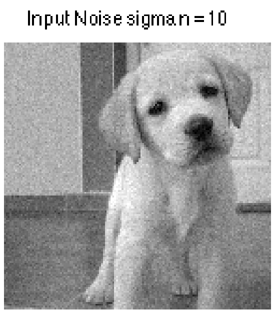
(a) Log of original and reflectance images on column 150 **(b)** FT of Mean Averaged Spectra.

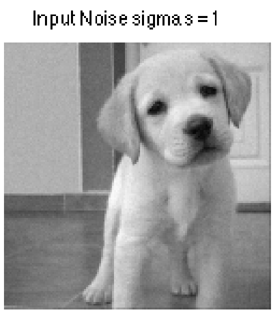
Problem 4.2

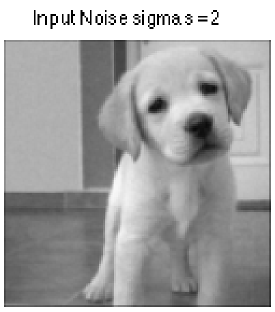
(A) In this problem, we compare Wiener filtering with Gaussian smoothing when denoising an image. With the 20 provided pet images, I took the Fourier transform and averaged together all the spectra.

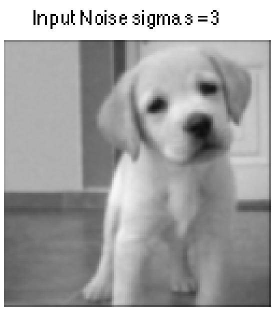
(B) The Mean Avergaed Spectra (M) is visualized above.

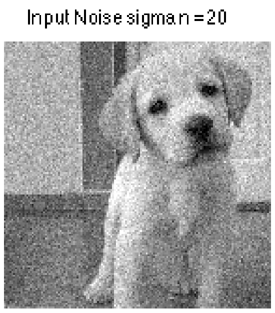
(C) When adding different amounts of Gaussian noise to the image, I used the `imnoise` function. Below are the results of adding in Gaussian noise with $\sigma_n = 10$ and $\sigma_n = 20$ for one of the images. Along with each noisy image, there are the results of applying different smoothed Gaussians with various σ_s. When choosing the best denoised image, you had to take into consideration both sharpness and noisiness. For the $\sigma_n = 10$ noisy image, the $\sigma_s = 1$ denoised image looked promising. For the $\sigma_n = 20$ noisy image, the $\sigma_s = 2$ denoised image looked better than the other denoised images without looking too blurry and with a satisfactory amount of noise gone.

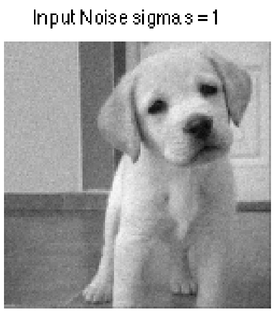


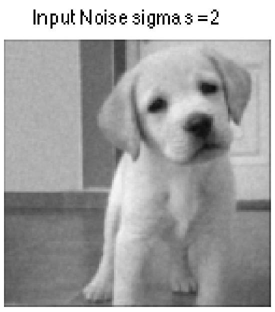


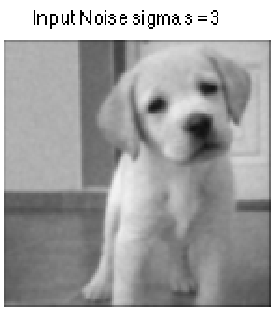




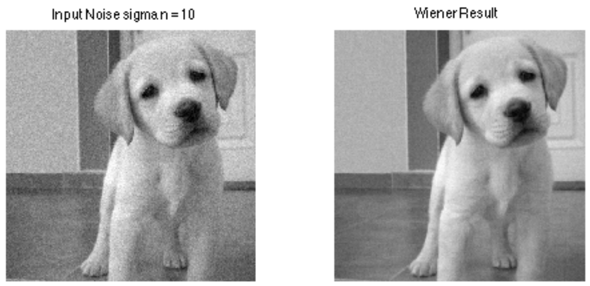


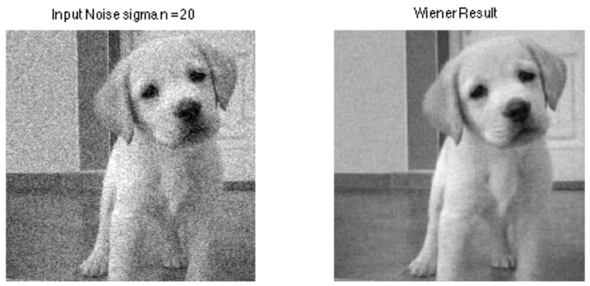






(D) To apply Wiener filtering, we apply Szeliski's formula from 3.4.3. With the Mean Averaged Spectra ($M(w_x, w_y)$), we can obtain the Power Spectrum by squaring M. Also, while working in the Fourier domain, we need to scale σ_n by w^2. As a result, the Wiener filter $W(w_x, w_y) = \frac{1}{1+w^2\sigma_n^2/M(w_x,w_y)^2}$. The results of applying the Wiener filter are good and reproduced below for both the noisy images.





(E) Measuring the reconstruction quality of the Wiener filter can be useful and obtained by taking the peak signal to noise ratio (PSNR). $PSNR(I_o, I_d) = 10 \cdot log_{10}(\frac{255^2 N}{||I_o - I_d||^2})$. Below are the results in dB when calculating PSNR for the orginal I_o and either the Wiener filtered denoised I_d, noisy I_n, and Gaussian smoothed I_s images (for both $\sigma_n = 10, 20$).

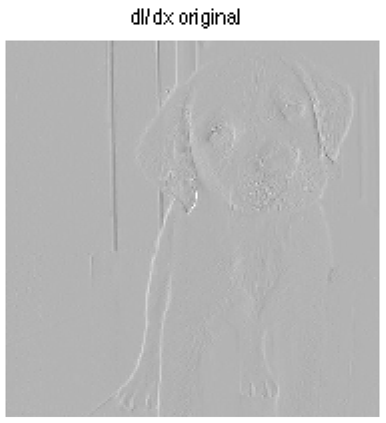
$PSNR(I_o, I_d)$	$PSNR(I_o, I_{\sigma_n=10})$	$PSNR(I_o, I_{\sigma_s=1})$	$PSNR(I_o, I_{\sigma_n=20})$	$PSNR(I_o, I_{\sigma_n=2})$
47.3732 dB	36.2181 dB	49.3056 dB	36.2128 dB	43.2118 dB

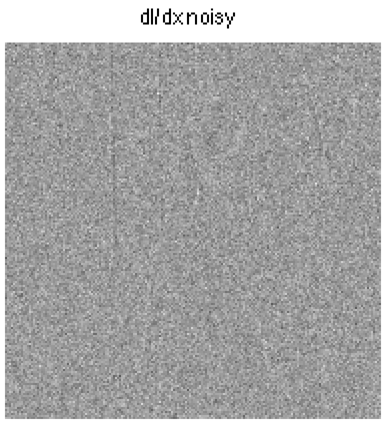
The higher in dB, the better the reconstruction. Wiener filtering generally has a high PNSR and is great for denoising, however, with a good σ_s value, the PNSR could be quite high for Gaussain smoothing as well.

(F) Optional!

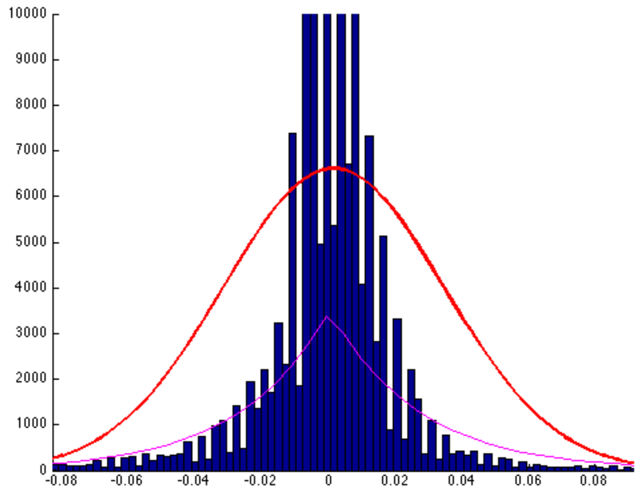
Problem 4.3

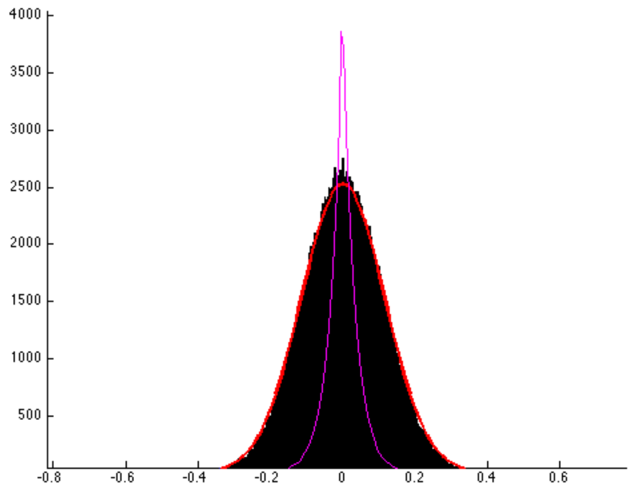
(A & B) This problem will based on Simoncelli & Adelson's wavelet-based denoising, however, we will be using gradient filters instead. Below is a representation of applying the [1,-1] filter with the original image and a Gassian noisy image of $\sigma_n = 20$ (image is the same picture as in the previous parts).





Next, I created a histogram of the filter responses and fitted a Gaussian (red) and Laplacian (purple) distributions to the histogram. For the original image, the Laplacian looked to be a better fit as it fits closer to the tails of the histogram. This is because the original image is sharper and less noisy, making the filter response retain that information with sharper edges and greater gradients. The Laplacian has a steeper distribution than the Gaussain. For the noisy image, the histogram is more distributed due to the general uniformness of the noise. Hence the Gaussian distribution better fitted this histogram. Results can be compared below.

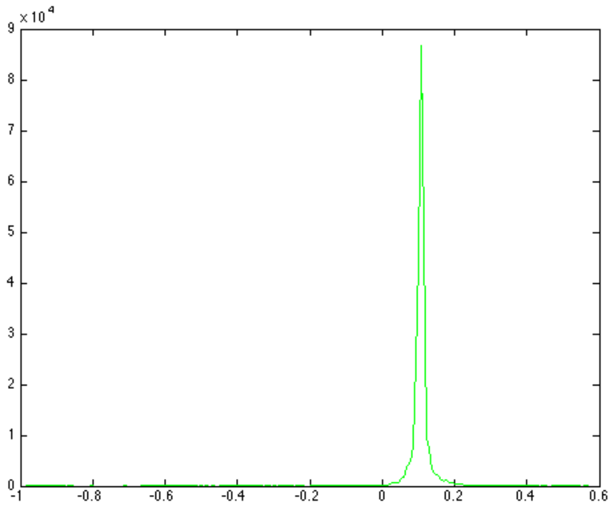




(C) For the observed (noisy) filter response y, we can estimate the true (uncorrupted) filter response x. The fitted Laplacian distribution in (a) will be the image prior. Ans the image has been corrupted with Gaussian noise. The estimate x given y using Bayes rule is

$$E[x|y] = \int_{-\infty}^{\infty} x p_{x|y}(x|y)\mathrm{d}x = \frac{\int_{-\infty}^{\infty} x p_{y|x}(y|x) p_x(x)\mathrm{d}x}{\int_{-\infty}^{\infty} p_{y|x}(y|x) p_x(x)\mathrm{d}x}.$$

The Laplacian prior is $p_x(x)$ and $p_{y|x}(y)$ is the normal distribution. Below is a plot of $E[x|y]$ as a function of y using the Laplacian distribution and $\sigma_n = 15$ in the liklihood. It is a mapping from corrupted filter observations to the least-squares estimate of the uncorrupted value.



(D) $E[x|y]$ can be used to denoise an image. The least squares estimate accounts for correlations bewteen the spectral bands. Least squares helps ensure the interband covariances are fully accounted for. When the Laplacian prior model is used as a better approximator on the marginal distributions of the gradient histogram, a finely tuned MSE can be evaluated and a better denoised image can be returned.

(E) Optional!