MIT CSAIL

6.869 Advances in Computer Vision

Spring 2011

Problem Set 3: Statistical models of images

Posted: Wednesday, February 16, 2011 **Due:** Wednesday, February 23, 2011

You should submit a hard copy of your work in class, and upload your code (and all files needed to run it, images, etc) to stellar.
Your report should include images and plots showing your results, as well as pieces of your code that you find relevant.

Problem 3.1 *Retinex: improving the input for the simple visual system*

One of the main issues that the simple visual system from the first lecture had is that it was very sensitive to the illumination conditions. For instance, the background/foreground segmentation assumed a constant intensity value for the background. Also, images were corrupted with noise if the illumination intensity was not strong enough. The goal of this problem is to implement the Retinex algorithm as described in the class slides and to apply it to images of the simple world.

In the attached code, your task is to complete the script *retinex.m*. The script computes the image derivatives with filters $[-1, 1]$ and its transpose, using the trick described in [1].

Attach the code with your changes and comments, and examples of pairs of input and output images. Use *simpleworld.jpg* and some other image (or several) that have strong illumination effects that you want to remove.

To better show the effects of the Retinex algorithm make a plot of a 1D line (or column) from the input and output images. If you plot both lines in the same graph, it should be clear that the gradient due to illumination disappears.

Problem 3.2 [Szeliski 3.16] *Wiener filtering*

In this problem you will compare Wiener filtering with Gaussian smoothing for noise removal. Estimate the frequency spectrum of your personal photo collection and use it to perform Wiener filtering on a few images with varying degrees of noise.

1. Collect a few hundred of your images (as many as you can) by re-scaling them to fit within a 512×512 window and cropping them.
2. Take their Fourier transforms, throw away the phase information, and average together all of the spectra.

3. Pick two of your favorite images and add varying amounts of Gaussian noise, $\sigma_n \in \{1, 2, 5, 10, 20\}$ gray levels.
4. For each combination of image and noise, determine by eye which width of a Gaussian blurring filter σ_s gives the best denoised result. You will have to make a subjective decision between sharpness and noise.
5. Compute the Wiener filtered version of all the noised images and compare them against your hand-tuned Gaussian-smoothed images.

Submit your code, the two images you selected in part 3, and the outputs of the Gaussian and Wiener filters. Do not submit your photo collection...

Problem 3.3 [Szeliski 3.6] *Histogram equalization*

Compute the gray level (luminance) histogram for an image and equalize it so that the tones look better (and the image is less sensitive to exposure settings).

1. Convert the color image to luminance.
2. Compute the histogram, the cumulative distribution, and the compensation transfer function (See Szelisiki 3.4 for details).
3. Compensate the luminance channel through the lookup table and re-generate the color image using color ratios (Szeliski, Equation 2.116).
4. Color values that are clipped in the original image, i.e., have one or more saturated color channels, may appear unnatural when remapped to a non-clipped value. Extend your algorithm to handle this case in some useful way.

References

[1] Yair Weiss. Deriving intrinsic images from image sequences. pages 68–75, 2001. `http://www.cs.huji.ac.il/~yweiss/iccv01.pdf`.