MIT CSAIL

6.869 Advances in Computer Vision

Fall 2012

Problem Set 2: Filtering Statistical models of images

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.

This problem uses pyramid image processing. Download and install the pyramid image processing toolbox by E. Simoncelli. When forming pyramid decompositions for these problems, you may always use the default decomposition filters.

Problem 2.1 *Image blending*

(a) Build a Laplacian pyramid of one image and show you can reconstruct back the original image. Code for the Laplacian pyramid is available in the pyramid image processing toolbox.

(b) Implement the function `PyrBlend(im1,im2,mask)` that takes as input two images and a binary mask (determining which part to use from each image) and produces the Laplacian pyramid blend of the two images. Use your function to blend two images of your favorite pets, friends or objects. Include in your report the original images, their Laplacian pyramids, the blending mask, and the resulting blended image.

Problem 2.2 *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.

References

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