LAB 6: DUE 8 DECEMBER 2015

Task 1: Deconvolution Gaussian Prior (50 pts)

Implement deconvolution with Gaussian Prior in the frequency domain. You can ignore the artefacts at the boundaries. You have the paper here *http://groups.csail.mit.edu/graphics/CodedAperture/.*

You also have some details about implementation in here

http://groups.csail.mit.edu/graphics/CodedAperture/SparseDeconv-LevinEtAl07.pdf

You have to implement section 2. The formula 8.

If you have problems, or doubts, you can check their matlab implementation. Check the section 5 of the previous document.

http://groups.csail.mit.edu/graphics/CodedAperture/DeconvolutionCode.html

DATA: *http://groups.csail.mit.edu/graphics/CodedAperture/CodedAperture-Data-LevinEtAl.zip*
The filter files are *.mat. In python you can use **scipy.io**. Please read the README files. There are 7 filters per scale in the x axis. You can try with only with the one per scale for now. See how the image gets better or worse reconstruction depending on depth.

The clock image is probably the best to see the results.

IMPORTANT: When computing the deconvolution, the borders of the image don't have appropriate information, and the content of the image is displaced by half the size of the filter. Roll the images in both dimensions by half the size of the filter, so the are aligned independently of the size of the filter!

Task 2: Image Reconstruction (50)

Create an "all-in-focus image" using the best result from all scales. To select the best scale, you can use the following.

The structure of the algorithm is the following.

- Deblur the entire image for each of scaled kernels.
- Compute reconstruction error e_k for each of the scaled images.
- For small windows around every pixel, compute the error of that pixel,
- Select the minimum and create a depth map with the selected depths.

$$e_k = y - f_k \otimes x_k$$

$$E_k \sim \sum_{j \in W_i} e_k(j)^2$$

$$d(i) = \operatorname{argmin}_k E_k(y(i))$$

Simply take the values for x_k from the selected $k = d(i)$ *to create a final deblurred image.*

Task 3: Deconvolution Sparse Prior Spatial domain (70)

This is a more complicated to implement method and requires a gradient descent implementation. However you have all the details and code in files linked above. You are supposed to implement deconvSps.m, which involves to implement deconvL2_w. Follow the description and reimplement the code in python.

The results should be quite better, compare to task 1 and 2. Use the same method from task 2 to compute an all in focus image.

Task 4: Depth Estimation (30)

In task 2 you have selected the best filter/scale for each pixel, that also gives you the depth for that pixel. When displaying the depthmap (labeling of the pixels for each kernell) use some color coding that is not gray-scale, so you can actually see something.

Extra:

If you want to experiment with the regularisation part, you can find some graph-cut libraries for python here. http://cmp.felk.cvut.cz/~smidm/python-packages-for-graph-cuts-on-images.html