EECS6323: Advance Topics in Computer Vision [Assignment3 – Computational Optics] Assigned: Feb 27, *2017 (*Monday*)*
Due: *March 10, 2017 (*Friday 5:00pm – email*)*

Goal of this assignment

1. To implement focal stacking
2. To implement blur detection and focus estimation

IMPLEMENT BOTH (SORRY, NO CHOICE THIS TIME ☹)

1. Extended Depth of Field using Focal Stacking

Use the data in the T1 directory to perform extended depth of field imaging. The images here should already be aligned (I haven't verified), so you probably don't need to worry about alignment. You should devise your own approach to determine image sharpness, noise cleanup/smoothing, and compositing. If you have access to Adobe Photoshop, you can try their focus stacking: http://www.youtube.com/watch?v=intzev1gsbI&feature=related.

There are two datasets: "Fly" and "Watch". The watch dataset has over 500 images, you can consider processing only every 10th image to speed up your computation.

For each dataset you should produce two outputs:

```
fly_composite.bmp          // the actual "all in focus" result.
fly_map.bmp                // map showing which image was used
                           // per pixel. The intensity can be the
                           // image id, e.g. I(x,y)=50 means the
                           // pixel at (x,y) is from image 50 in
                           // the stack
```

2. Defocus Blur Detection

Implement the blur detection algorithm used in by the Image Preconditioning Paper (Brown et al CVPR'06). The image used in the paper is provided with this assignment in directory T2.

In that paper, we first broke the image into 128x128 blocks around each projected feature and used an operator called the "Tenegrad Operator" to determine the sharpest feature. The Tenegrad operator is just the summation of the results of a Sobel Masks (horizontal and vertical) in the entire block. You can think of it as a response to the strength of the gradient in that region.

Selecting the sharpest feature as the exemplar, we then blurred it (the exemplar) with several blur kernels of increasing size. The Tenegrad operator is then performed on these synthetic blurred examples to find each of their responses. We then examine each 128x128 blurred feature in the original image to see which synthetic blurred feature it best matches (in terms of Tenegrad response). This allows us to build a PSF map and even generate a synthetic blurred image from the single sharp feature.

In this task, you are to implement this focus detection algorithm, as well as use this information to deblur the original image, i.e. use the estimated PSF to perform deblurring on each block (this last part was not done in the CVPR'06 paper). Your output should be the following:

infocus_feature	(128x128 image of the most in focus feature)
psf_map plot	(similar to figure 3 in the paper)

synthetic_image	(build a synthetic version of the original input using only the exemplar feature and estimated blurred kernel for each feature)
deblurred_image	(apply inverse filtering to each feature using its estimated blur kernel to construct a deblurred version of the original image)

SUBMITTING YOUR ASSIGNMENT

Programming Language

You may use any programming language you want. All tasks can be done in Matlab, but you are free to write you own code.

What to turn in?

1.) README File

For each task, provide a short description of your implementation followed by results from your output. Follow the directions of each task.

2.) Code and Images

Include your code and output images.
Make the following directory tree structure:

~\assignment4\Readme.{doc,tex,pdf}
~\assignment4\t1\yourfiles
~\assignment4\t2\yourfiles

Please double-check your zip file to make sure everything is in order.

Example, if your name is Zhang Bo, then name your submission will be:
zhangbo_assignment4.zip.

How to turn in?

Email me a link to your assignment. *Email your assignment by the due date. Late submission will have points deducted.*