**Assignment 9: High Dynamic Range Imaging**

**Introduction**

In this homework assignment, we will focus on the core algorithms behind computing HDR.

Before we get started, download the assignment files under Resources > Assignments > Assignment 9 that contains the files you will be working with. It also contains the assignment description if you wish to view it offline.

Once you have extracted the zip file above, you are ready to get started.

As with previous assignments, running **assignment9\_test.py**directly will apply a unit test to your code and print out helpful feedback. You can use this to debug your functions.

**Part 1: Building the Functionality.**

This is a quick highlight of the functions. assignment10.py has extensive function descriptions with step by step guidelines for each implementation.

1. **normalizeImage**

For an input image, normalize it from 0 to 255.

1. **linearWeight**

Function for outputting weights based on pixel value.

1. **getYXLocations**

Given an image and an intensity value, return the YX locations of where the intensity value appears.

1. **computeResponseCurve**

Given the images at different exposures for a single channel, get the response curve by performing a Singular Value Decomposition (SVD).

1. **readImages**

Reads in images from a folder (this is implemented for you).

1. **computeHDR**

Brings everything together (this is implemented for you).

**Part 2: Try it yourself!**

Note: We recommend resizing images to small versions (like provided) so the code takes ~1 minute to run. Larger images will degrade computation time significantly. We wrote a simple resizing boolean parameter that will resize your input by 1/4th its original size for you (if you set it = True).

**Obtaining a set of HDR images.**

Look online for a set of HDR images and their respective exposures. If you are provided with the original images, the exposure level may be in the metadata (be sure to check the image properties for it). Modify that in the provided sample code, and try running your code on those images to see if it works for other images. In the PDF provide thumbnails of the images and the output HDR image. Don't forget to modify the exposures to what your input is (feel free to add use less / more images to test it, but use at least 5 images at different exposures).

You are welcome to take your own if your camera supports taking multiple exposures. Dark indoor scenes with bright outdoors generally work great for this.

**Some notes:**

Originally we planned on doing tone-mapping. This is the final step in computing an HDR image. We opted not to due to some functionality which we did not obtain in the available libraries we use (numpy / scipy). Due to this the output images may not be perfect but you should notice more detail in over /under exposed areas as shown in the example output included for you.

This code is still experimental. If you think you find what you think is a bug by all means bring it up on Piazza so we can discuss it, we'd really appreciate that.

Lastly, this assignment is somewhat different than others in that it will also require simply reading code and understanding it. We provide this with the hope that it provides further insight as to how numpy can be used for matrix computations.

**The Writeup**

This is what we want you to do for the PDF.

1. Demonstrate the output you got on the input images we gave you. Since we give you the answer for this, you will be awarded zero points if you include this output but your code does not actually work.
2. Demonstrate thumbnails of the HDR images you chose to try.
3. Demonstrate the result for those HDR images.

**What to turn in:**

Please turn in the following files.**Keep the size limit of each file to 6MB**. If you need to compress your PDF before submitting, you can use <http://smallpdf.com/compress-pdf>:

* **assignment9.py** - Your code.
* **assignment9.pdf** - See above for the writeup, include the images in the PDF (only in the PDF)!