High Dynamic Range

CS 4475: Project 7

1 - Objective

The goal of this project is to explore a couple of different ways to tone map a high dynamic range image. In the process, you will write a routine to blur a floating-point image. We will need this blur function for the tone mapping algorithms. The photographic part of this assignment is to take three photos of the same scene with three different exposures. (You will not use these photos in the programming project, however.)

2 - Deadline

Your project should be submitted on T-Square by 11:55PM on Monday, July 7, 2014.

3 - Process

3.1 Download the base source

Download and unzip the provided zip file for this project. Try out the provided code using the example images that are already in the "data" directory.

3.2 Project description

The most important task for this assignment is to write a routine that blurs a floating-point image. We have provided skeleton code for this, an the template routine is called "blur()" near the end of the file floatImage.pde. The blur() routine has two input parameters: a floating-point image and a number of iterations. You should use the 1/16 * (1-4-6-4-1) pixel weighting template to blur horizontally and vertically to perform one step of blurring. Repeat this process the given number of iterations to create the correct amount of blur. When you perform 1-4-6-4-1 blurring across the rows or down a column of pixels, you will find that you need a place to put your temporary results. You should create another floating-point image in which to store these results. A floating-point image is just a 2-dimensional array of PVectors. Each pixel is represented by a single PVector, and the r,g,b colors are stored in the x,y,z values of the PVector.

You can use any method that you prefer to handle the pixels on the edges of the image. Some of your choices include using black, wrapping to the other side, or mirroring the values.

Even with just the template code, you can run the sample program. The image that is shown on the left is one of a collection of photos of a scene. There are six such images altogether, and each one was taken with a different shutter speed. You can look at each these images by pressing the left and right arrow keys. When your blur function is working properly, the right image will start as a blurred version of the current image on the left. You can press they "1" key to re-calculate this blurred image if you have switched to a different current image.

When the program is first run, it assembles the six input images into one high dynamic range image. When you have the blur function running properly, the keys "2" and "3" will produce two different tone mapped versions of this HDR image. Pressing "2" will create an image that uses the per-color channel version of high/low frequency separation and scaling (as described in class). With this method, the final colors are washed out. Pressing "3" will produce a tone mapped version of the HDR image that only operates on the luminance channel, and then re-introduces the colors at the end. This tone mapping routine does a better job of preserving the rich colors in the original images.

3.3 Expected Results

Below are the images that you should get once you correctly implement blur. Pressing the number keys will produce each of these images: (1) blurred current image, (2) per-color channel tone mapping, and (3) luminance-only tone mapping (best result).

3.4 Same Scene, Different Exposures

As part of this project, you will take three different photos of the same scene, with three different exposures. You will turn in these photos with the programming part of the assignment.

For DLSR cameras, you should be able to set the shutter speed manually, while keeping all of the other camera settings the same. Some cameras will allow you to shoot a sequence of "bracketed" photos, which means that the camera will shoot low, medium and long exposure images for you. Try to set the different shutter speeds to be a factor of four apart in duration.

If you have a point-and-shoot or a cell phone camera, you can still get the camera to shoot at different shutter speeds. On a cell phone such as the iPhone, you can touch the part of the screen that you with to focus on, and the camera will also set the exposure that is best for showing touched part of the image. If you touch a bright patch on the LCD screen before shooting, the camera will shoot a short exposure. If you touch a dim patch, it will take a longer exposure. Most point-and-shoot cameras will allow you to center the camera on one part of a scene, press the shutter button half-way down, and then change the camera direction to frame the final shot. After you have pointed in the correct direction, press the shutter button the rest of the way down. The camera should adjust the exposure to whatever you were pointing to when you half-pressed the button.

Be sure to select a scene that has a wide variety of radiance values. If you have the sky in your scene, for instance, also include a shaded area that is dark.

Note that we will **not** be asking you to using your images with the provided code as part of this assignment. It is too difficult to calibrate most images and to shoot them so that there is exact pixel correspondence. You are welcome to try your images with the code on your own, but be aware that the setup can be quite tricky. The provided code for assembling the HDR image is also not very robust.

3.5 Authorship Rules

The code that you turn in entirely your own. You are allowed to talk to other members of the class and to the Professor and the TA about general implementation of the assignment. It is also fine to seek the help of others for general Processing/Java programming questions. You may not, however, use code that anyone other than yourself has written. Using the provided code for this assignment is fine, and is expected. Code that is explicitly **not** allowed includes code taken from the Web, from books, from previous assignments or from any source other than yourself. You should not show your code to other students. Feel free to seek the help of the Professor and the TA's for suggestions about debugging your code.

3.6 Submission

In order to run the source code, it must be in a folder named after the main file. When submitting any assignment, leave it in this folder, compress it and submit via T-square. In addition, turn in three photographs of the same scene with different exposures. Name these images photo_short.jpg, photo_medium.jpg and photo_long.jpg, according to the length of the exposure.