15663 Computational Photography Homework 2

1. **HDR Imaging**

   Develop RAW Images:

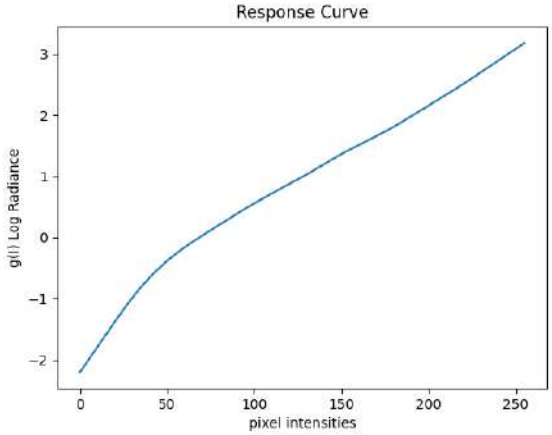
   Report the flags you used:

   dcraw -v -w -q 3 -o 1 -4 -T exposure*.nef

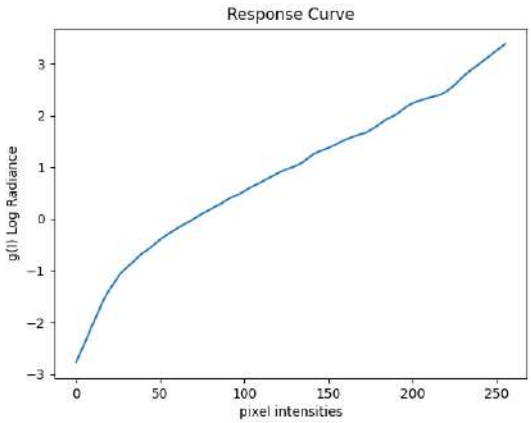
   -v: Enables verbose output, displaying processing details.
   -w: Uses the camera's white balance settings.
   -q 3: Sets the interpolation quality to the highest level (high-quality linear).
   -o 1: Converts to sRGB color space.
   -6: Outputs 16-bit linear data.
   -T: Saves the output as a TIFF file.
   exposure*.nef: Specifies the input files, with a wildcard to select multiple NEF (Nikon raw) files.

2. **Linearize Rendered Images:**

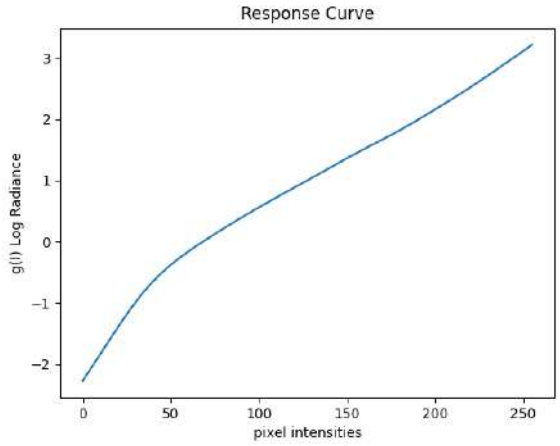
   Include a plot of function g you recovered:



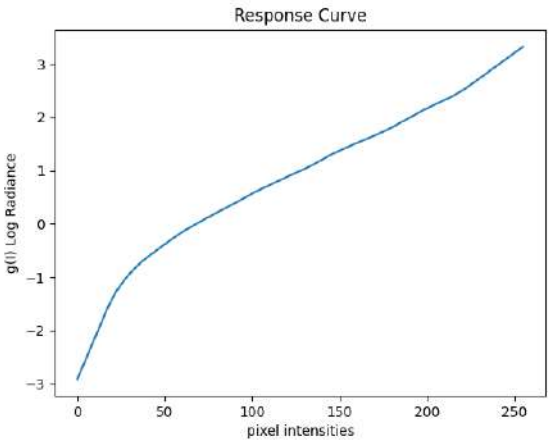
Gaussian Weighting



Photon Weighting



Tent Weighting



Uniform Weighting

The response curve g appears linear for pixel intensities above 50 across all the weight functions. However, for pixel intensities below 50, the curve becomes nonlinear, which, as we discussed in class, could be due to the noise characteristics in that range. I used a lambda value of 100 for all these weighting schemes.

Merge Exposure Stack into HDR image:

I have used Zmin = 0.05 and Zmax = 0.95

HDR images generated by linear weighting of .jpg stack:

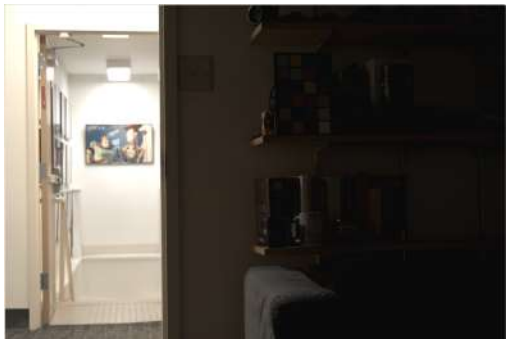
Gaussian Weighting

Photon Weighting

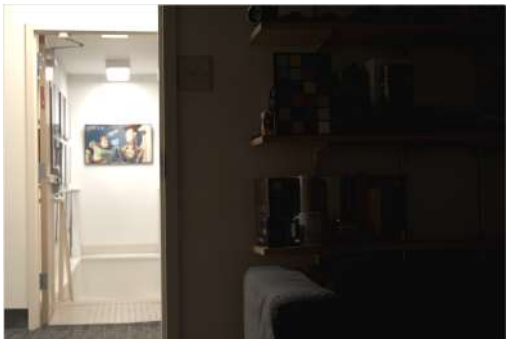
Tent Weighting

Uniform Weighting

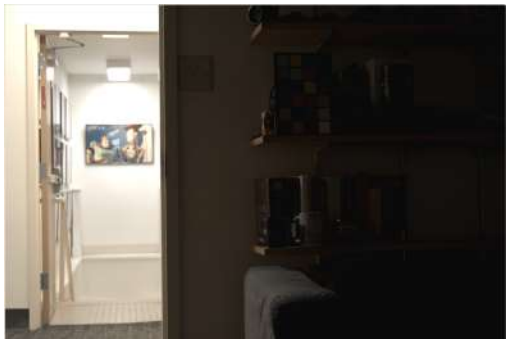
HDR images generated by linear weighting of .tiff stack:



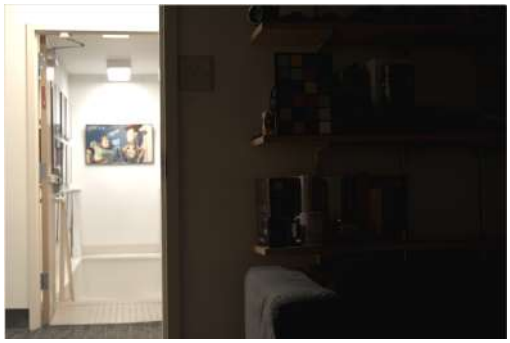
Gaussian Weighting



Photon Weighting



Tent Weighting



Uniform Weighting

HDR images generated by logarithmic weighting of .jpg stack:

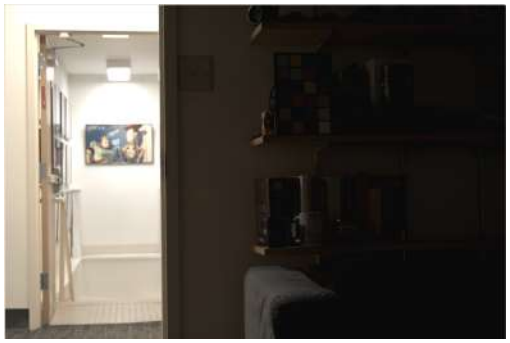
Gaussian Weighting

Photon Weighting

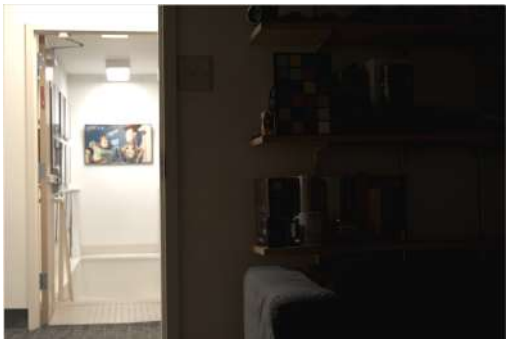
Tent Weighting

Uniform Weighting

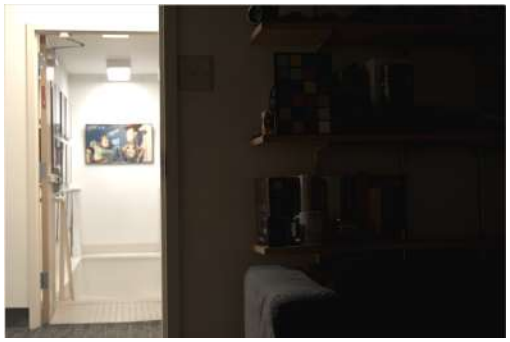
HDR images generated by logarithmic weighting of .tiff stack:



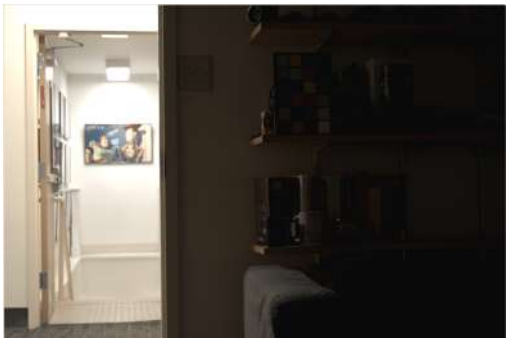
Gaussian Weighting



Photon Weighting



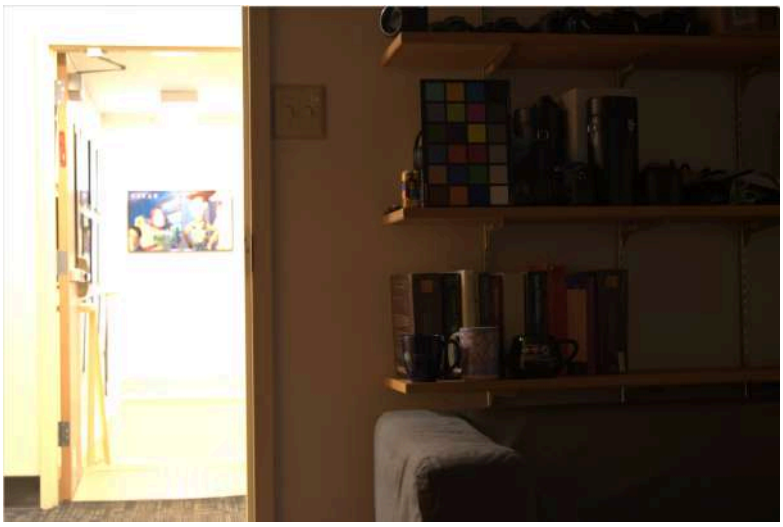
Tent Weighting



Uniform Weighting

Make sure to comment on why you selected the image you did.

All the images generated from the JPG stacks appear quite similar, and the same goes for those generated from the TIFF stacks. Also, when I scale the hdr generated by jpg stack by 0.09, the resulting image is close to the hdr image generated by tiff stack.

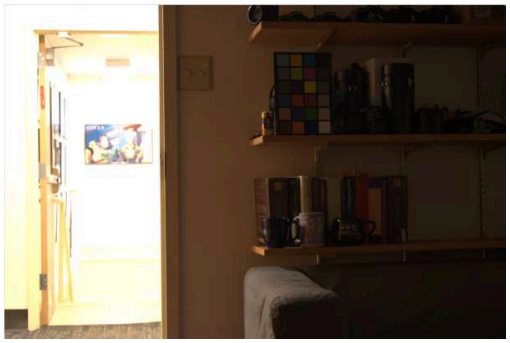


HDR generated by jpg stack, and scaled by 0.09

Given that the information is not lost in either cases, and both of them look alike, I would like to select the hdr image generated by tiff stack, as it has less color cast. But, this is too early to choose an image, given we haven't performed tone mapping and color correction yet.

2. Color correction and white balancing

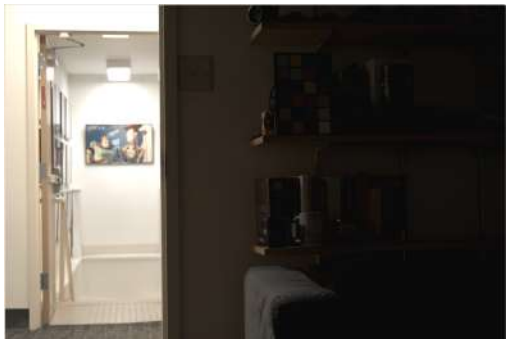
HDR image generated by .jpg stack:



Before

After

HDR image generated by .tiff stack:



Before

After

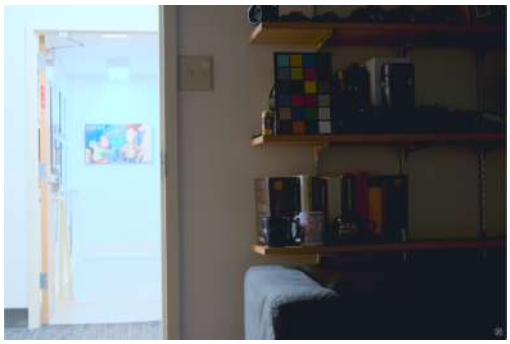
Compare the color-corrected image with the original, and discuss which one you like the best.

The color corrected and white balanced image looks better in both cases.

3. Photographic Tonemapping:

Plot representative tonemaps for both the RGB and luminance methods, and discuss your results.

Full pictures of tone mapped images below:



Luminance Tonemapping

RGB Tonemapping

Make sure to mention which tonemap you like the most.

I like the image generated by applying luminance tone mapping, because the colors look right, whereas in the other method, the colors look washed out.

4. Create and tonemap your own HDR photo

Luminance Tonemapping

RGB Tonemapping

Luminance Tonemapping

RGB Tonemapping

Experiment with different parameters, show a few representative tonemaps, discuss your results, and determine which result you like the most

I liked the resultant tonemapped images formed by using jpg stack. In case of images generated by tiff stack, the overall images look very dark, even though I scale it, probably because color correction has not been done on it.