# Computational Photography Assignment 2

Sagnik Ghosh

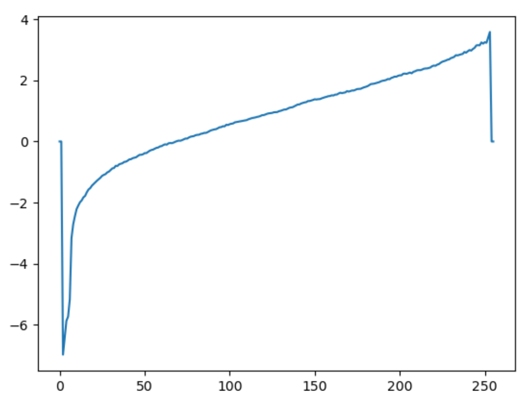
September 2024

# 1 HDR Imaging

## 1.1 Develop RAW images

Here's the command I used: dcraw -v -4 -w -o 1 -T im1.nef

## 1.2 Linearize rendered images




The function below $Z_{min}$ and greater than $Z_{max}$ is equal to 0 because there was no information in that region. It does not matter what values the function takes in those regions because we do not use them while merging anyway. I used a regularization factor of 1.

## 1.3 Merge exposure stack into HDR image

For RAW images, I observed that using values $Z_{min} = 0.01$ and $Z_{max} = 0.99$ resulted in the best HDR images. The same values for JPG were 0.05 and 0.95 respectively.

Please note that all the images displayed for this part are **post** color correction and tonemapping.

**Comparing 16 images.** Among JPG and RAW images, RAW images were clearly better for two reasons. First, the darker regions (couch) were yellowish in the JPG HDR images, which was not the case with the RAW images. Also, the bright regions in the JPG images look blueish. This could be because the estimated non-linearity is imperfect, making the linearized JPG stack non-ideal for HDR. Second, the RAW HDR images seemed to have a higher contrast. Here's an example:

Figure 1: JPG Gaussian Weighted Log Merged

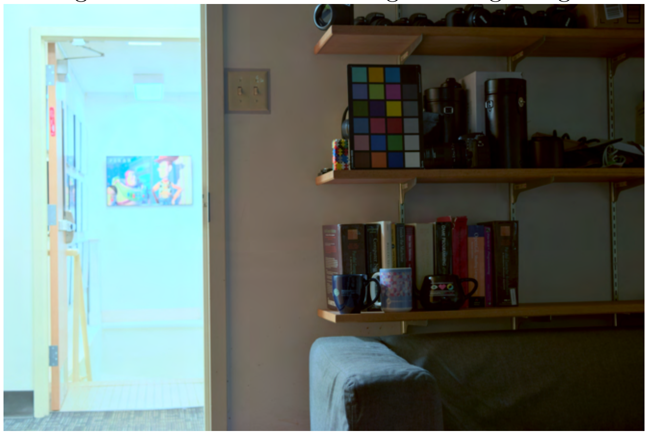
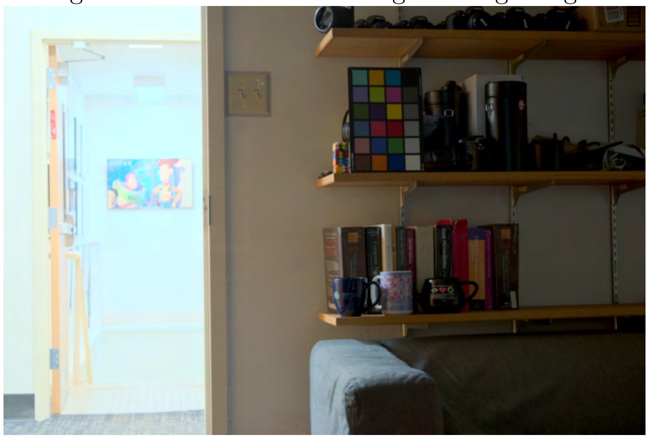
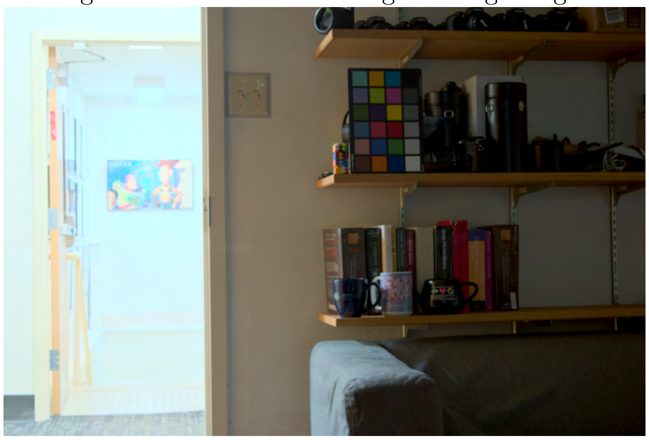


Figure 2: RAW Gaussian Weighted Log Merged



While all the weighting schemes produced very similar results, if you zoom into the constant dark regions, the photon weights seem to be less noisy. Apart from that, linear and logarithmic merging did not make a difference to my eye. While many of the images produced were almost equally good, if I had to choose one, I would choose the RAW stack photon-weighted log-merged HDR image for the reasons mentioned above. Here's the image:

Figure 3: RAW Photon Weighted Log Merged



## 2 Color correction and white balancing

The image without color correction looks unnaturally yellow and the darker regions are much darker than the image with color correction. Here's a comparison between the two:

Figure 4: Without color correction

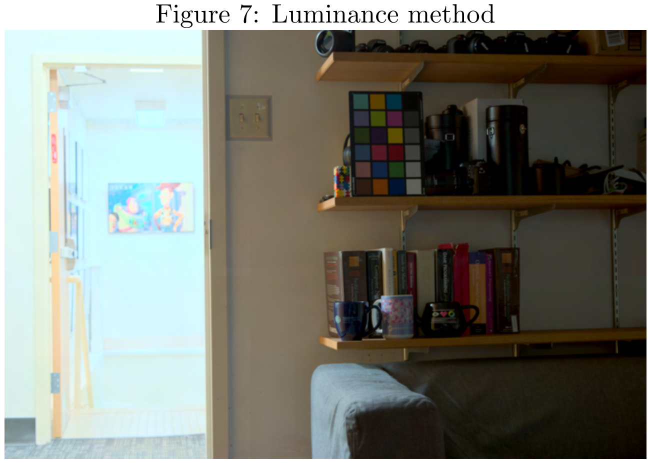
Figure 5: With color correction

# 3 Photographic tonemapping

I prefer the luminance method because in the darker regions, this method preserves the colors better. The RGB method saturates some of the colors in the darker regions. More specifically, if you look at the front face of the arm of the couch, you can notice some color artifacts on the image obtained by RGB tonemapping. Here are the images:

Figure 6: RGB method

Figure 7: Luminance method

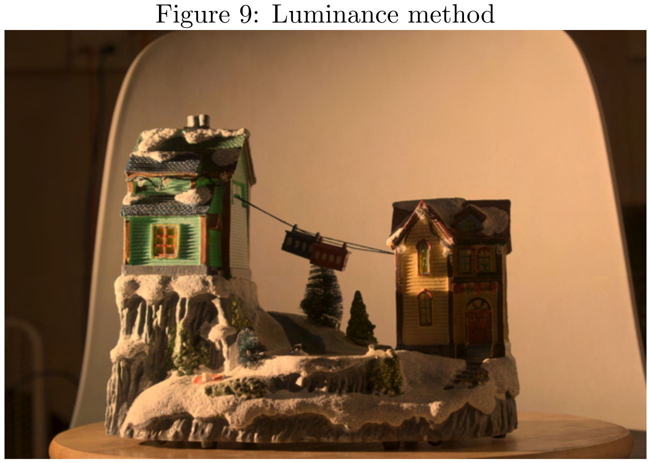


## 4 Create and tonemap your own HDR photo

I used the same exposure times as the data (door stack) provided. Comparing the luminance and RGB methods (on RAW, photon-weighted), the luminance method seems to better reproduce color. A good example of this is for the red and white button below the green house. The luminance method produces the red color as it's supposed to be whereas the RGB method produces a pale red. The luminance method also better reproduces the illumination color I used to light the scene. Here are the images:

Figure 8: RGB method

Figure 9: Luminance method



If we compare JPG to RAW, the tonemapped HDR image computed from the JPG stack is biased towards red. Hence, I prefer the RAW stack. Here's an example for photon-weighted, log-merged images:

Figure 10: JPG

Figure 11: RAW

A lower value of key darkens the dark regions and vice versa based on the images below. It's clear from the red car in the image. I would prefer a higher key (0.2).

Figure 12: key=0.1

Figure 13: key=0.2

As the scene does not have strong highlights, varying the burn parameter does not seem to have a noticeable effect. Here's a comparison:

Figure 14: burn=0.95

Figure 15: burn=0.85

In summary, I prefer RAW, log-merged, photon-weighted, luminace-tonemapped, and a key=0.2 and burn=0.95. Here's the final image:

Figure 16: Final image

## 5 Noise calibration and optimal weights

The histograms are proportional to the number of photons arriving during a given exposure time. We know that that is a Poisson distribution. Hence, the histograms below are approximately shaped like a Poisson distribution.

Figure 17: Histogram 1

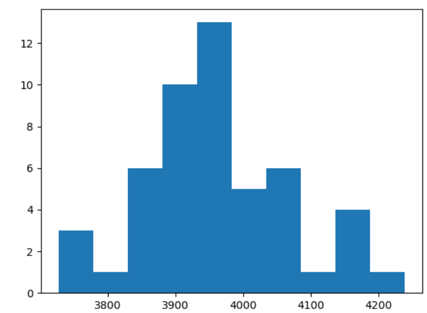


Figure 18: Histogram 2

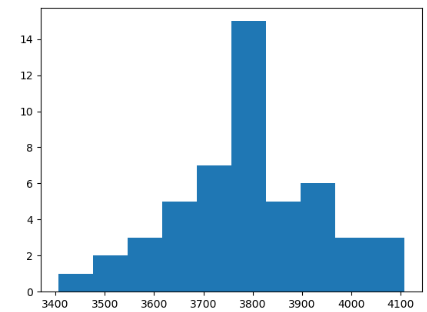
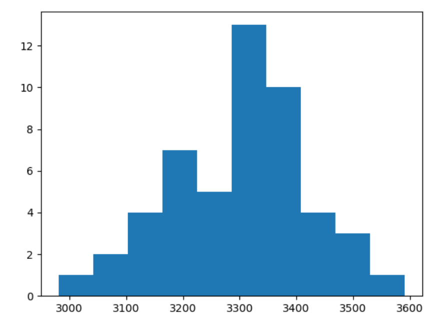



Figure 19: Histogram 3



Here are the mean-variance plots for the different color channels. For red, $g = 6.04, \sigma_{add}^2 = 998.57$. For green, $g = 1.88, \sigma_{add}^2 = 107.56$. For blue, $g = 4.51, \sigma_{add}^2 = 265.64$.

Figure 20: Red

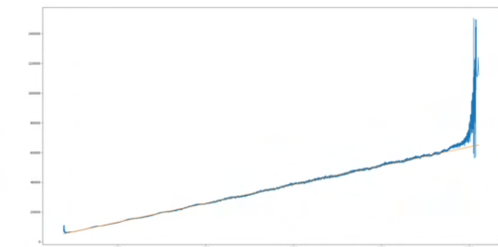


Figure 21: Green

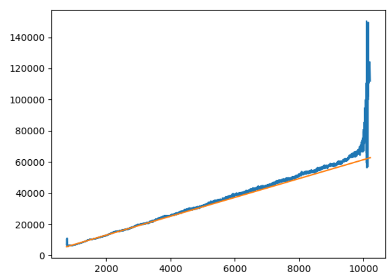
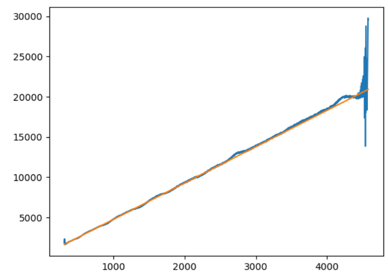


Figure 22: Blue



# 6 Merging with optimal weights

Here's a comparison between the best result from part 4 and merging with optimal weights.

Figure 23: Part 4

Figure 24: Optimal weights

Although they look very similar, if you look closely at the darker regions, merging with optimal weights produces higher contrast in those regions. A good example is the base of the house on the right. Another example is one of the feet of the whole scene.

# 7 Description of all included images

As GitHub does not allow ¿ 25MB uploads, please go to this link for all the deliverable data.

- part1.hdr, part2.hdr, part4.hdr, part5.hdr - HDR files for different parts
- exposure9.jpg, exposure9.tiff - Captured JPG and RAW images