ASSIGNMENT 2

15663 Computational Photograpy Fall 2023
DUE: September 29, 2023

1 HDR Imaging (60 points)

Develop RAW images (5 points) The `dcraw` flags are: `dcraw -w -q 3 -o 1 -T -4 $file`

- `-w`: Use camera white balance
- `-q 3`: Set the interpolation quality to the highest
- `-o 1`: Set the output colorspace to sRGB
- `-T`: Write TIFF instead of PPM
- `-4`: Write Linear 16-bit images

Linearize rendered images (25 points) Unlike the RAW images which are linear, the rendered images (`JPG` images) are non-linear. An intensity I_{ij}^k at pixel $\{i, j\}$ of image k is

$$I_{ij}^k = f(t^k L_{ij})$$

where L_{ij} is some unknown scene flux value and t^k is the know exposure time of image k.

We recover the function $g \equiv log(f^{-1}$ that maps I_{ij}^k to $g(I_{ij}^k) = log(L_{ij}) + log(t^k)$. As the domain of g is the set of discrete intensity values $\{0, ..., 255\}$, g is essentially just a 256-dimensional vector.

We can recover g by solving the following least-squares optimization problem:

$$\min_{g, L_{ij}} \sum_{i,j} \sum_k \{w(I_{ij}^k)/255[g(I_{ij}^2) - log(L_{ij}) - log(t^k)]\}^2 + \lambda \sum_{z=0}^{2} 55\{w(z/255)\nabla^2 g(z)\}^2$$

We used `numpy.linalg.lstsq` to solve the least-squares problem. The plots of g we recovered are shown below.

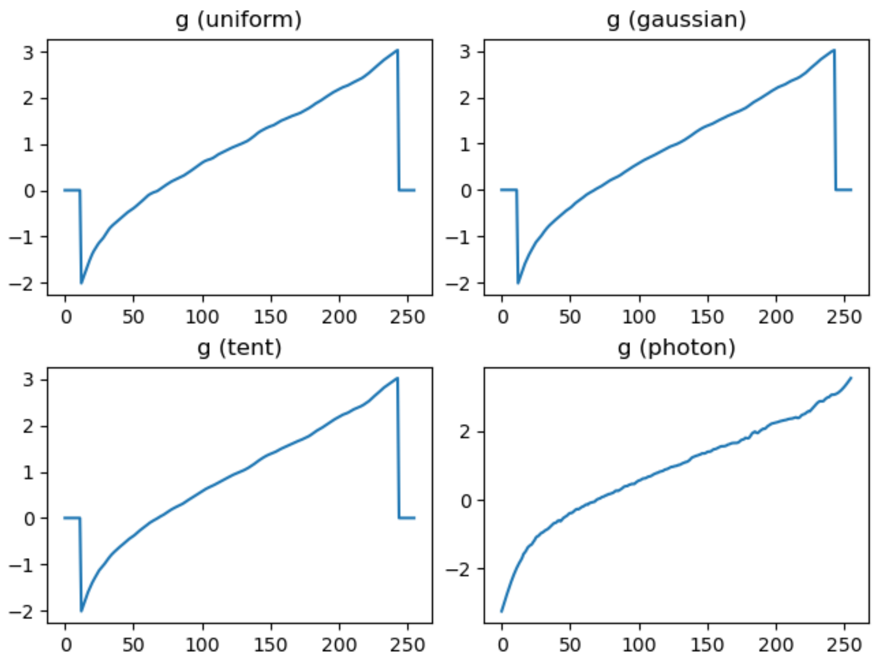

Figure 1: Plots of the function g

Merge exposure stack into HDR image (30 points) Given a set of k LDR linear images corresponding to different exposures t^k, we can merge them into an HDR image either in the linear or in the logarithmic domain.

When using linear merging, we form the HDR images as:

$$I_{ij,HDR} = \frac{\sum_k w\left(I^k_{ij,LDR}\right) I^k_{ij,lin}/t^k}{\sum_k w\left(I^k_{ij,LDR}\right)}$$

When using logarithmic merging, we form the HDR image as:

$$I_{ij,HDR} = exp\left(\frac{\sum_k w(I^k_{ij,LDR}\left(log(I^k_{ij,lin} + \epsilon\right) - log(t^k}{\sum_k w\left(I^k_{ij,LDR}\right)}\right)$$

The resulting HDR images:

Figure 2: $W_{Gaussian}$

Figure 3: W_{tent}

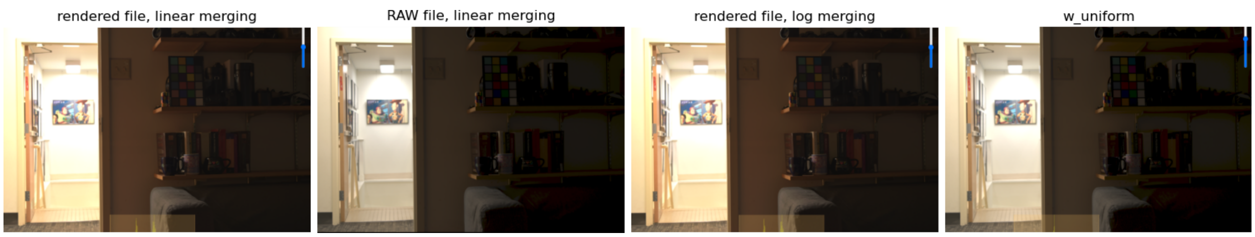

Figure 4: $W_{uniform}$

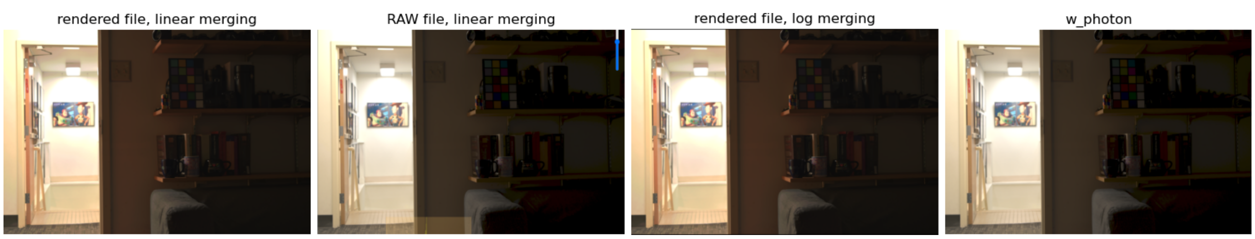

Figure 5: W_{photon}

We used $z_{min} = 0.05$ and $z_{max} = 0.95$ for all 16 images. We chose the image acquired through rendered image with linear merging and the weight function ($w_{uniform}$) for the subsequent steps.

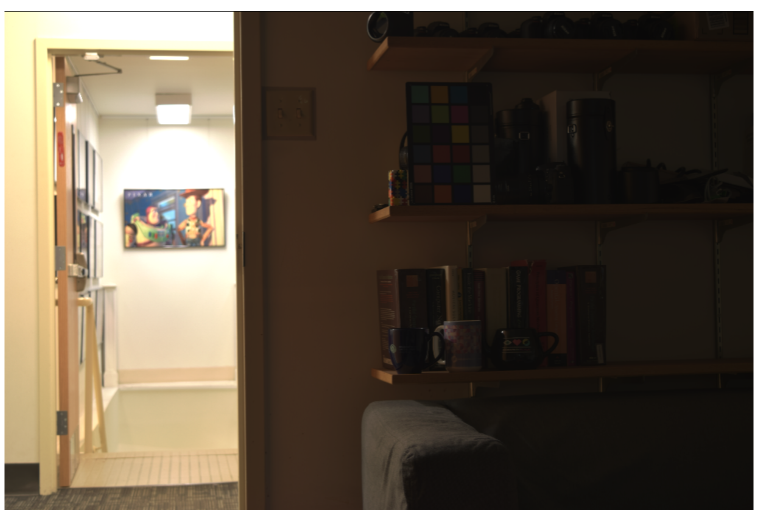

Figure 6: Merged HDR image

2 Color correction and white balancing (20 points)

The order of patches of the given color checker goes from the bottom left to the upper right column-wise. Below shows the original image and the color-corrected image. Personally, I prefer the original image. Despite not being the most accurate representation, the orange hue brings a very pleasing ambiance.

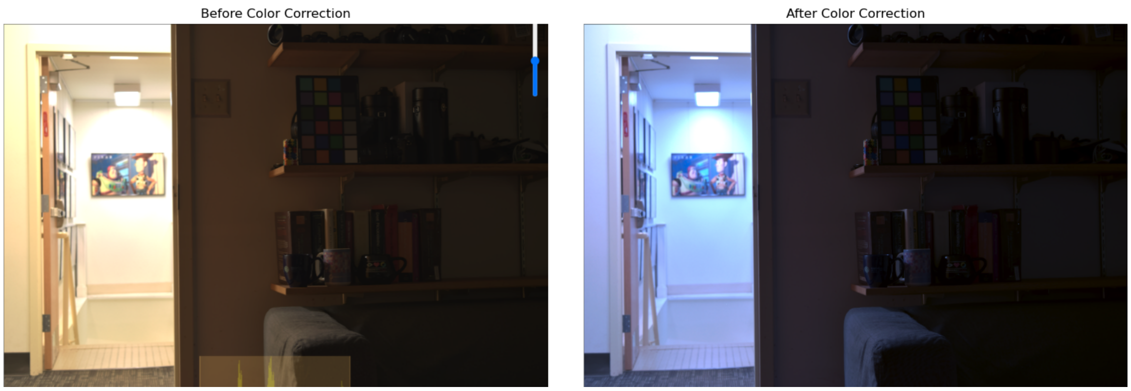

Figure 7: The original and color corrected images

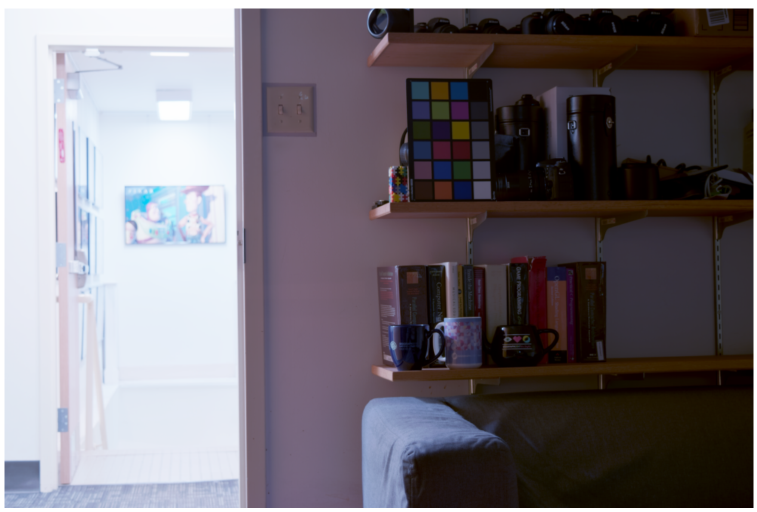

Figure 8: Tonemapped image with color correction

3 Photographic tonemapping (20 points)

The HDR images need to be tonemapped so that they can be displayed. We implemented both the RGB and the luminance methods, and the results are shown as below.

The representative tonemap for the RGB method:

Figure 9: RGB tonemapped image

The representative tonemap for the luminance method:

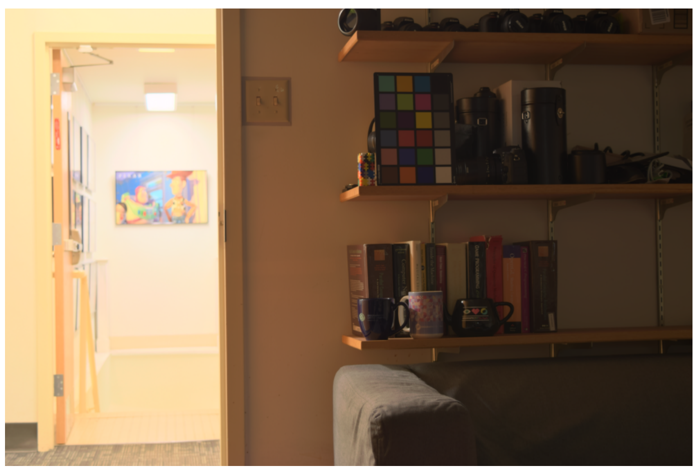

Figure 10: Luminance tonemapped image

We used $K = 0.15$ and $B = 0.95$ as the parameters for both methods. The image obtained through the luminance method exhibits more vivid and vibrant colors. I personally prefer the RGB tonemapped image as it more accurately captured the actual scene.

4 Create and tonemap your own HDR photo (50 points)

We used shutter speeds [1/4000, 1/2000, 1/1000, 1/500, 1/250, 1/125, 1/60, 1/30, 1/15, 1/8, 1/4] and captured 11 images in total. Some of the `.TIFF` images are shown here.

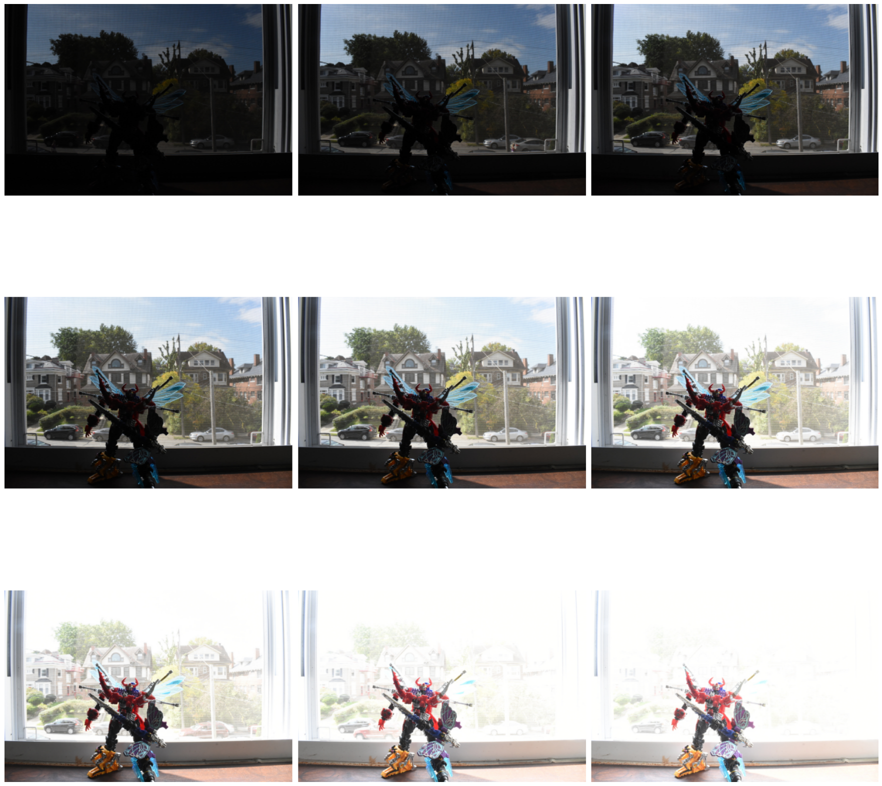

Figure 11: RAW images

With the RGB method, we used $K = 1.5$, $B = 0.95$ for rendered images, and $K = 100$, $B = 10.0$ for RAW images.

Figure 12: RGB tonemapped images

With the luminance method, we used $K = 0.8$, $B = 0.95$ for both rendered and RAW images.

Figure 13: Luminance tonemapped images

Personally, I prefer the image obtained through the luminance method with the RAW image. The colors appear more vibrant, and the overall brightness of the image is enhanced.

5 Noise calibration and optimal weights (50 points)

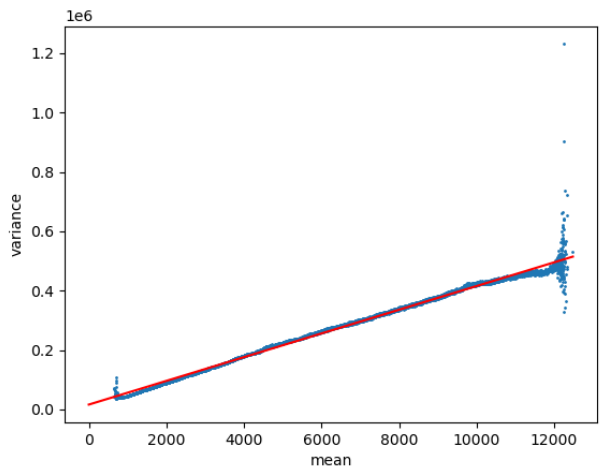

Figure 14: Mean-Variance plot and the fitted line

Noise Calibration The estimated gain $g = 39.94$, and the estimated variance $\sigma^2_{additive} = 16287.04$.

The shape of the histograms approximately obeys the Gaussian distribution.

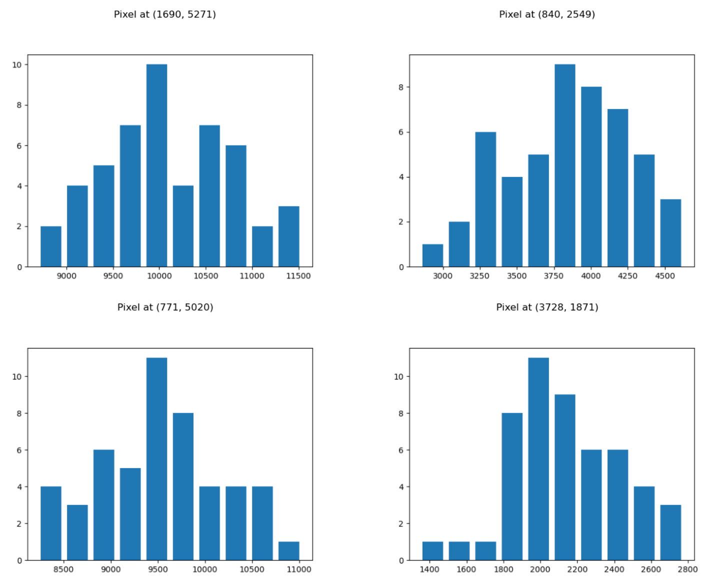

Figure 15: Caption

Merging with Optimal Weights