HW4: High Dynamic Range Imaging and Tone-mapping

Yiming Dai, YDF9097

1. Write an Auto Exposure Bracketing (AEB) function for Tegra

Following TODO:hw4 tabs inside the bakcbone project, I got minimum and maximum exposures. Then I increased the exposure time by a factor of 2 and take a new picture, and I repeated this procedure until at least %20 of the pixels are saturated. Lastly, I set requester exposure time. Then I captured images and saved them in .jpg format.

2. Write a program to find the camera response curves for the shield tablet

After imread all 5 images, I chose a random subset of 1000 pixels of each image. Then I implemented the function in gsolve.m for each of the red, green and blue channel, to get the response curve g and the recovered log radiance lE for each of the pixels that I input to the algorithm. Afterwards, I ploted the recovered values for g versus the valid range of pixel values $Z\epsilon[0,255]$. In the same figure, I also included a plot of the log exposure for each of the pixels used as input to gsolve.m. I took the measured pixel value for each pixel and ploted it against the sum of the recovered log irradiance and log exposure time, $X[i,j] = lE[i] + \ln(B[j])$. The results are as follows (I chose $l = 5$):

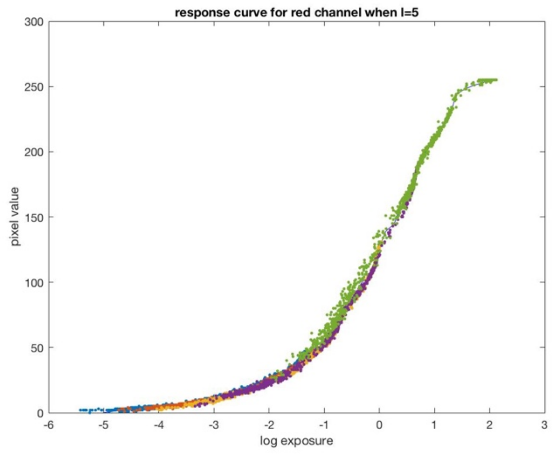

Figure 1

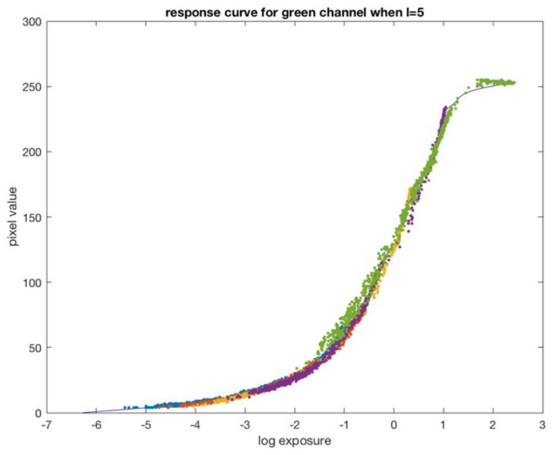

Figure 2

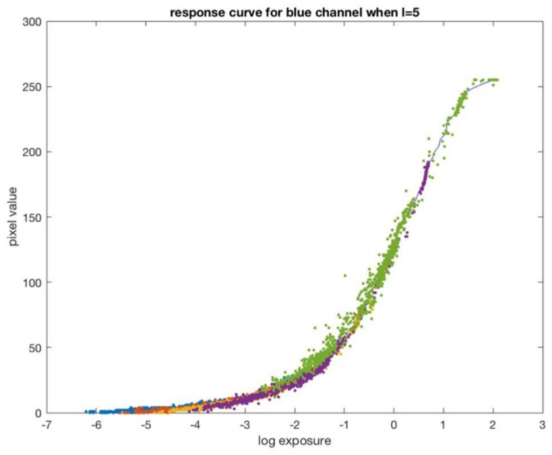

Figure 3

I tried a few different values in the range $l\epsilon[0.1,5]$ for red channel. Results are as follows.

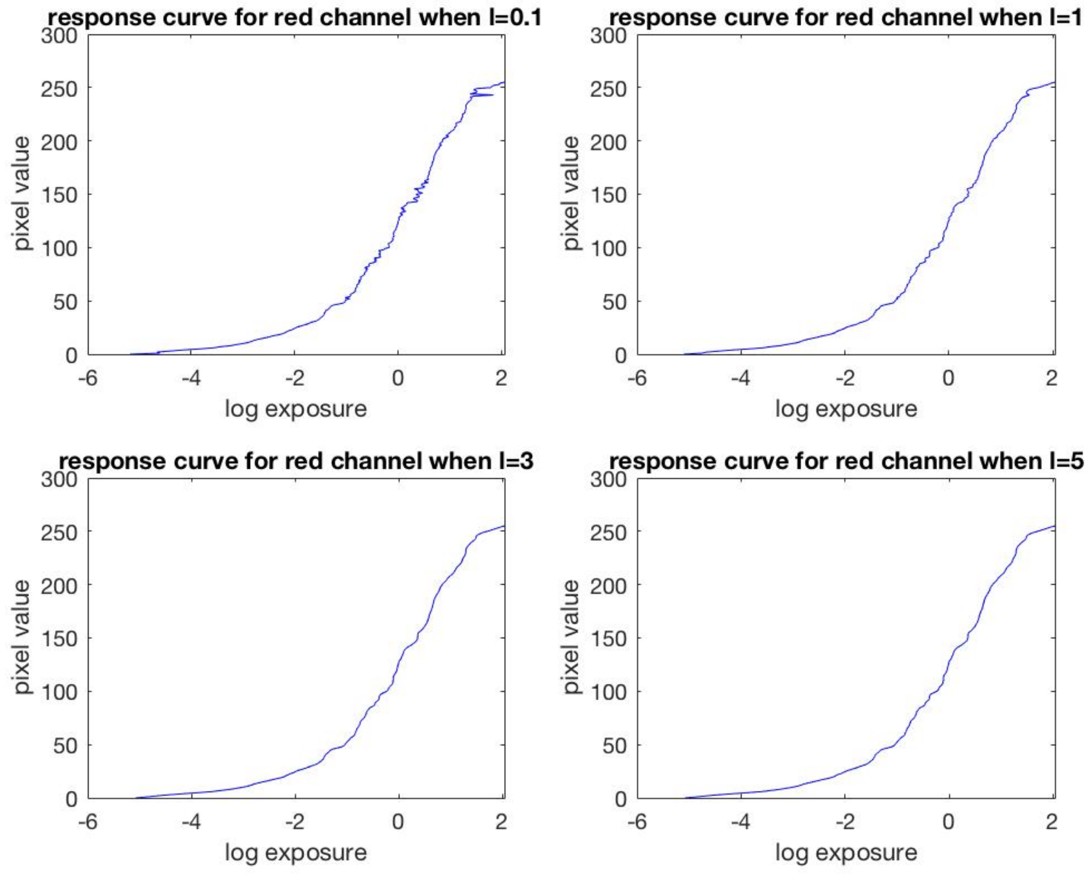

Figure 4

How does adjusting this parameter affect the result you get for the response curve?
According to results above, obviously that when l get larger, the response curve is smoother.

3. Recover the HDR radiance map of the scene

To recover the HDR radiance map of the scene, I iterated over all possible pixel values and used the Matlab 'find' function to get the indices of the pixels that correspond to a particular pixel value. Then I used the equation $\ln(E[i]) = \frac{1}{P}\sum_{j=1}^{P}(g(Z[i,j]) - \ln(B[j]))$ to find the irradiance for those pixels. I repeated the procedure for each possible pixel value from 0-255. After getting irradiance at each pixel’s location, I can recover the HDR radiance map.
I tested out my code by the given response curve and the memorial image sequence that I downloaded from Paul Debevec's website, the results are as follows:

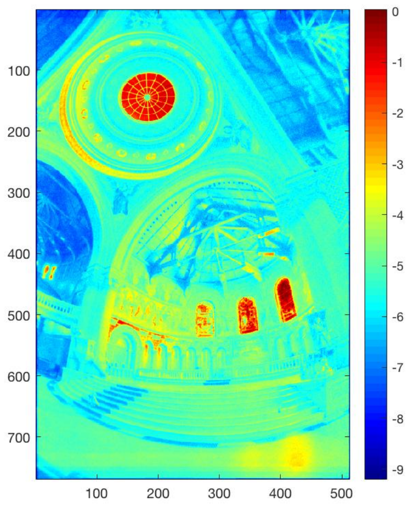

Figure 5 red channel

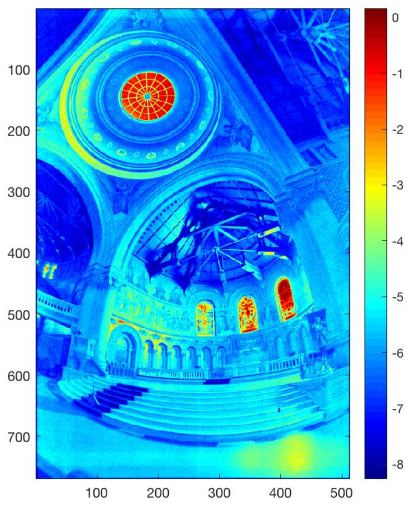

Figure 6 green channel

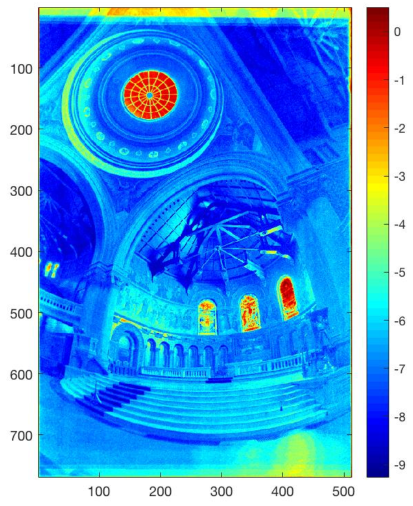

Figure 7 blue channel

4. Show a plot of the radiance image recovered from the AEB sequence I captured in part 1

I ran my code on images I captured, results are as follows:

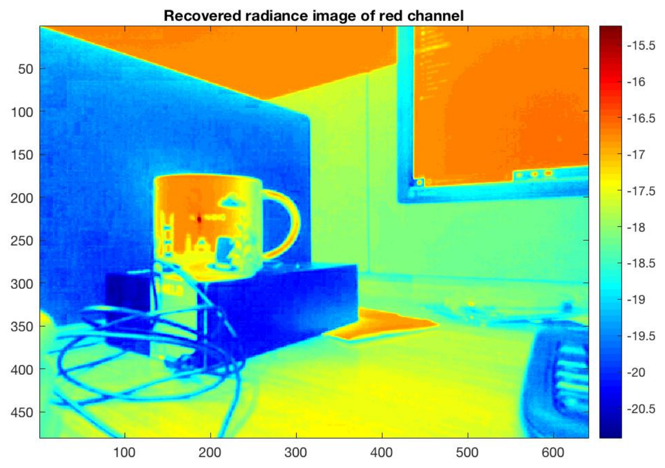

Figure 8

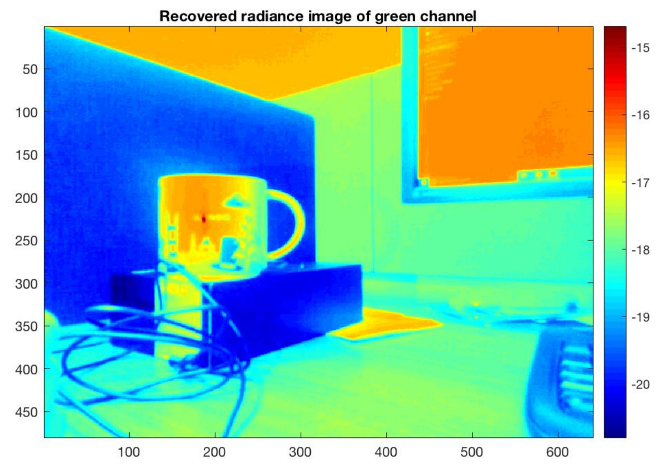

Figure 9

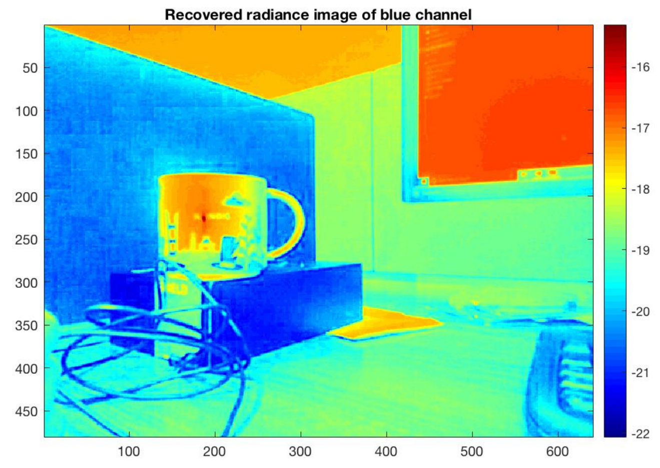

Figure 10

Question: What is the dynamic range of your scene?
Red channel: 10^7
Green channel: 10^6
Blue channel: 10^7

5. Implement a tone mapping algorithm to display my HDR image

The radiance map I recovered in part 4 likely has a much larger dynamic range. So, I need to apply a simple global tone-mapping algorithm to my radiance image.
(1) First, I scaled the brightness of each pixel uniformly by using the following equation so that all of the pixels fall in the range [0,1].

$$E_{norm}[i] = \frac{E[i] - E_{min}}{E_{max} - E_{min}}$$

where E_{max} and E_{min} are the maximum and minimum pixel values taken across all color channels. The result is as follows:

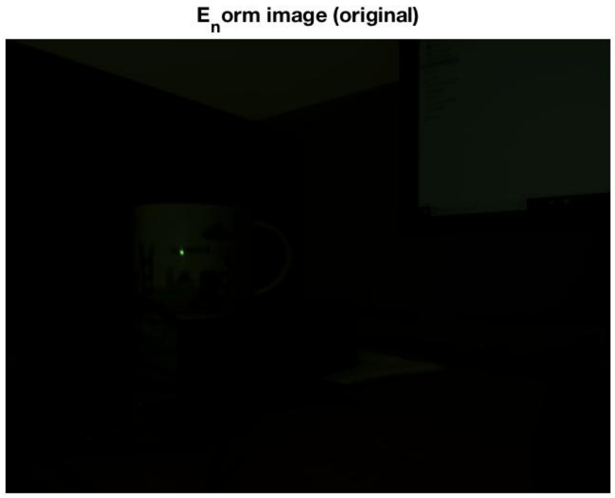

Figure 11

which is really dark.

(2) Then I applied a gamma curve to the image by using the following equation:

$$E_{gamma}[i] = E_{norm}^{\gamma}[i]$$

I used different values for γ, results are as follows:

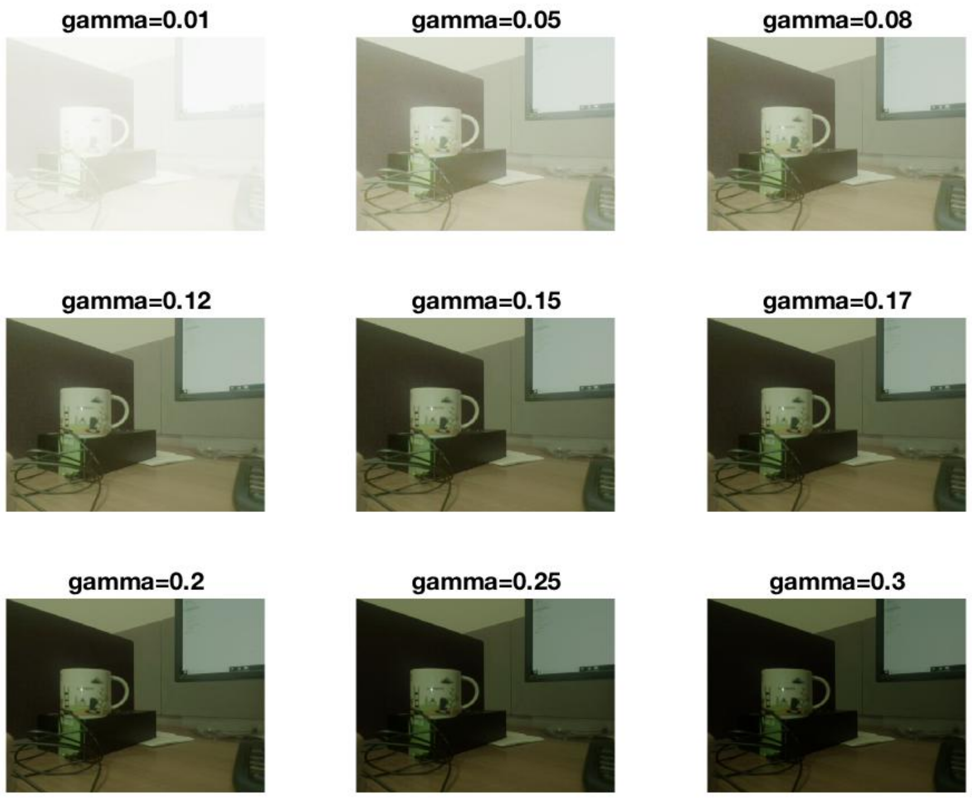

Figure 12

Question: Can you find a value that gives you visually pleasing results?
Yes. From results above, I found out that when $\gamma = 0.17$, the result is visually pleasing.

And the zoom-in image pairs are as follows:

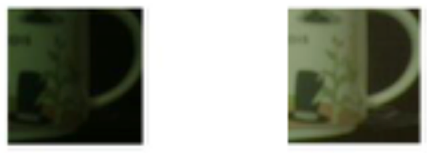

Figure 13 left is original zoom-in image, right is HDR zoom-in image

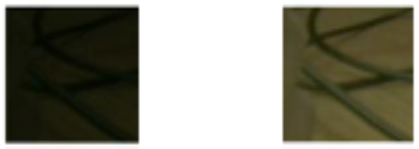
Figure 14 left is original zoom-in image, right is HDR zoom-in image

As we can see from results above, the HDR reconstruction is working.

(3) Lastly, I tried the global tone mapping operator from Reinhard '02 [2].
First, I converted the radiance image from color to grayscale by using the following equation

$$L[i] = rgb2gray(E_{norm}[i])$$

Then I used the following equations to implement the tone mapping. I first calculated the log average exposure

$$L_{avg} = exp\left(\frac{1}{N}\sum_i \ln{(L]})\right)$$

Next I scaled the image according to:

$$T[i] = \frac{a}{L_{avg}} L[i]$$

Next I applied the Reinhard tone-mapping operator:

$$L_{tone}[i] = \frac{T[i](1 + {}^{T[i]}\!/_{T_{max}^2}}{1 + T[i]}$$

Finally, define the scaling operator

$$M[i] = {}^{L_{tone}[i]}\!/_{L[i]}$$

and use this to scale each of the color channels in the radiance image.

$$E_{new}[i] = M[i] \times E[i]$$

I experimented with different values for $a = [0.09, 0.18, 0.36, 0.72]$. The results are as follows:

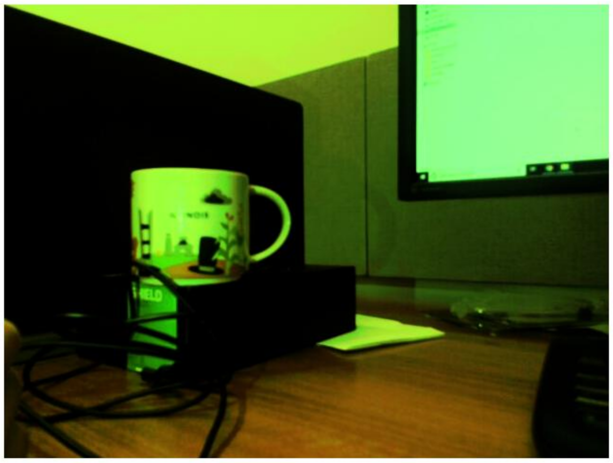

Figure 15 a=0.09

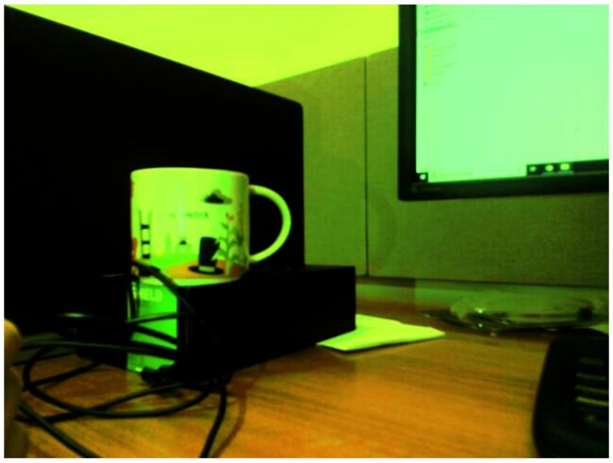

Figure 16 a=0.18

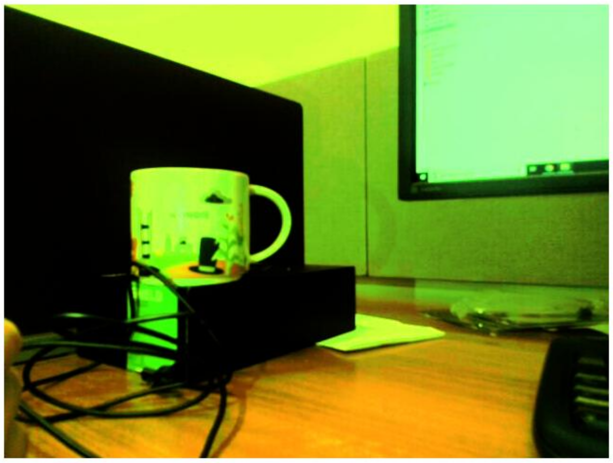

Figure 17 a=0.36

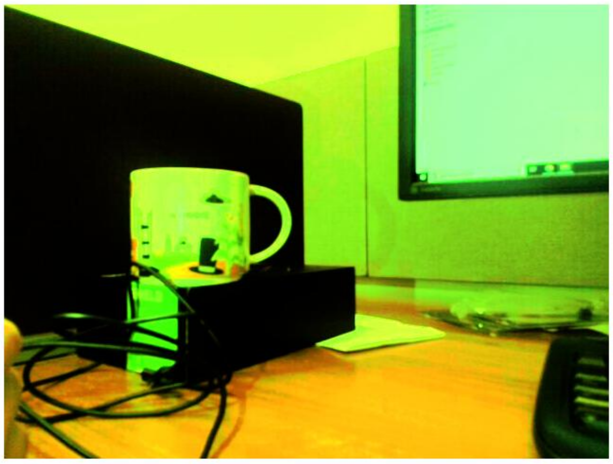

Figure 18 a=0.72

The images are green because the images we captured by tablet are different from real images, so in this situation, the HDR reconstruction results would not obey the normal rule.
From the results above, I found out that when $a = 0.36$, the result is visually pleasing.
And the zoom-in image pairs are as follows:

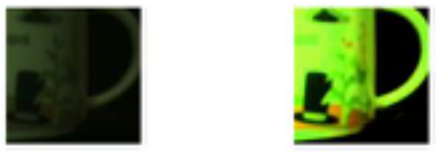

Figure 19 left is original zoom-in image, right is HDR zoom-in image

Figure 20 left is original zoom-in image, right is HDR zoom-in image

As we can see from results above, the HDR reconstruction is working.

What problems I encountered:

1. For part of recovering radiance image, at first I did not quite understand the equation's meaning, since I was confused by the lE I got from gsolve.m and lE calculated by the equation. Then I realize that I need to iterate over all possible pixel values to actually calculate lE to recover radiance image.
2. When going to the part of displaying HDR image, I did not always get the right results. Later, I found out that the E_norm I calculated is actually $log(E\_norm)$, I needed to implement $E\_norm\ =\ E\_norm.$ ^10. Then results were normal.