Kevin Woodward (keawoodw@ucsc.edu), Jose Sepulveda (joasepul@ucsc.edu)
CMPE 264
November 14th, 2018

Project 1 Report

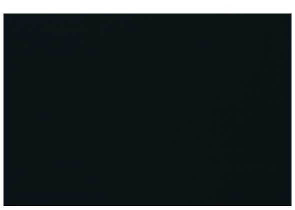
Camera specifications:
- Body: Canon EOS 20Da
- Lens: Tamron AF28-300mm A20
- Other: Promaster 67mm UV filter

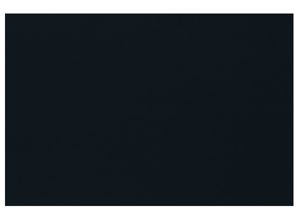
Image specifications common to all taken photos:
- ISO: 100
- F-number: 5.6
- Image dimensions: 3504 x 2336

Program details:
- Language: Python 2.7.14
- Packages:
  - os
  - math
  - pickle
  - numpy (1.14.0)
  - opencv-python (3.4.3.18)
  - matplotlib (2.1.2)
  - scipy (1.0.0)

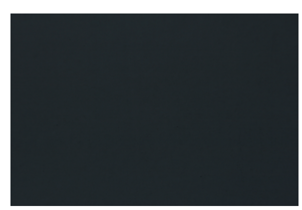
Part 1:

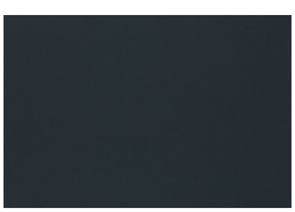
Calibration images were acquired on a white paper surface illuminated by sunlight. Full resolution images can be found in the WhiteImages folder. Here are the images displayed:


1/1000 sec


1/800 sec

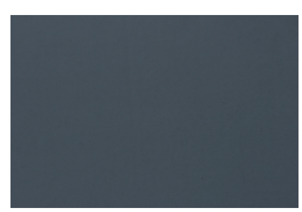

1/640 sec

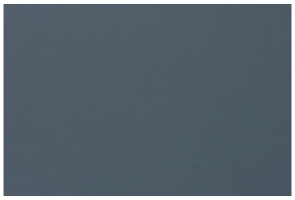

1/500 sec


1/400 sec


1/320 sec


1/250 sec


1/200 sec


1/160 sec


1/125 sec


1/100 sec


1/80 sec


1/60 sec


1/50 sec


1/40 sec


1/30 sec

1/25 sec

1/20 sec

1/15 sec

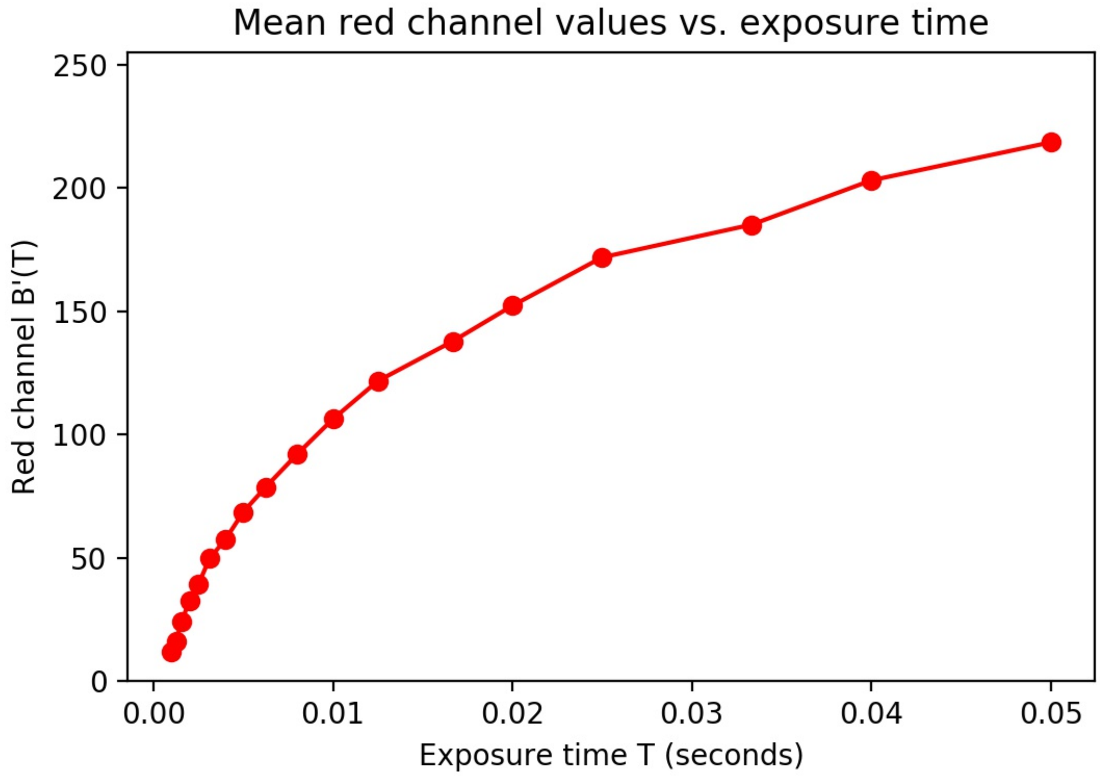
1/13 sec

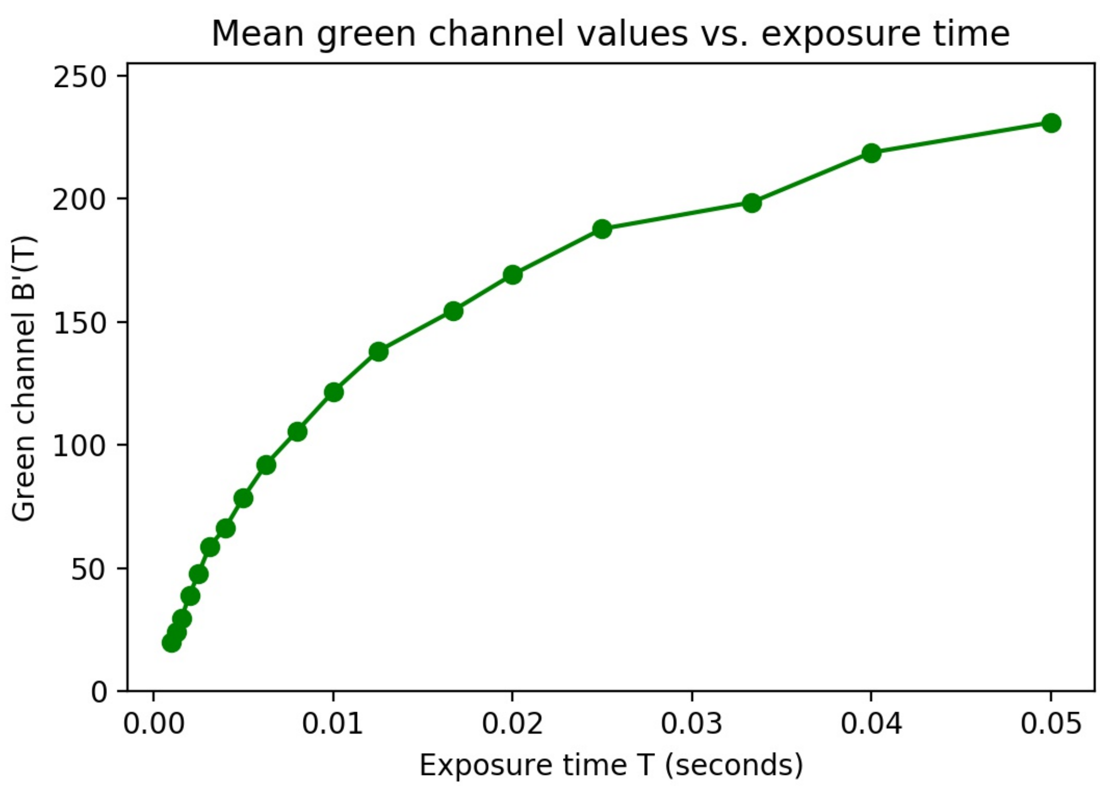
1/10 sec

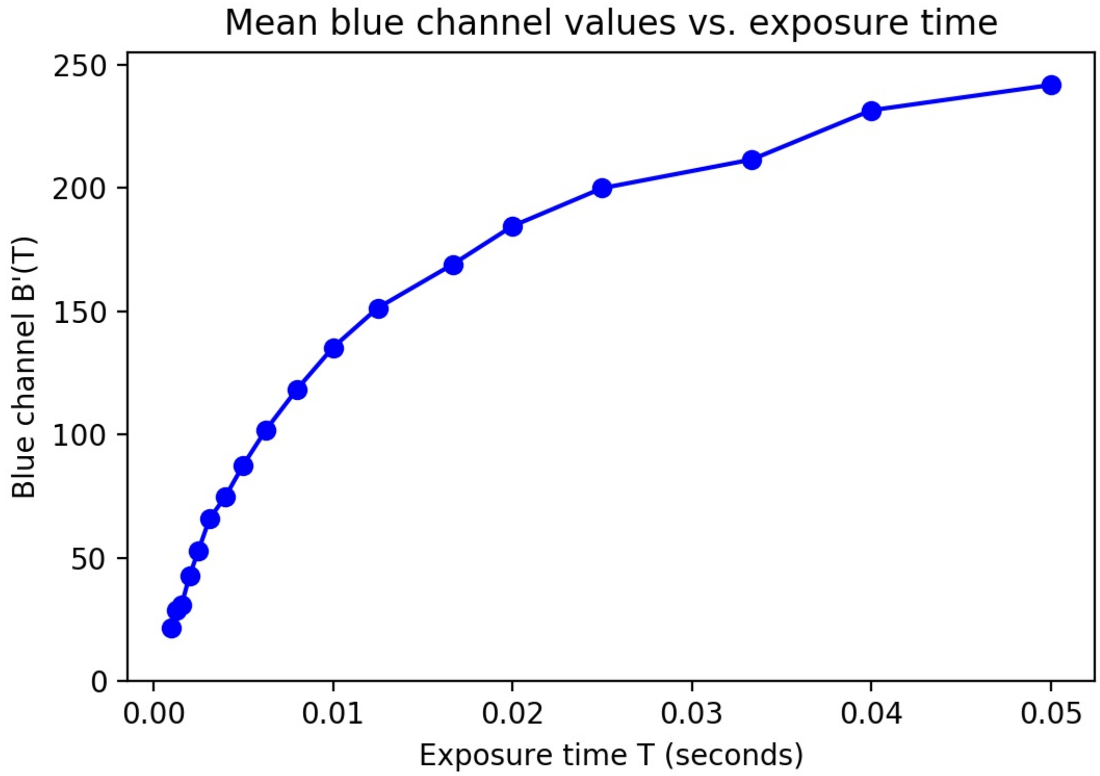
1/8 sec

The images for the values 1/15 sec, 1/13 sec, 1/10 sec, and 1/8 sec were overexposed and therefore not used in the following steps.

The plots for each channel's mean value versus exposure time were then generated by loading the images, and averaging a 10px by 10px section from the center of each image for each exposure time. Full resolution plots can be found in the PlotsAndHists folder. Here are the plots displayed:


Mean red channel values vs. exposure time
Red channel B'(T)
Exposure time T (seconds)
0
50
100
150
200
250
0.00
0.01
0.02
0.03
0.04
0.05

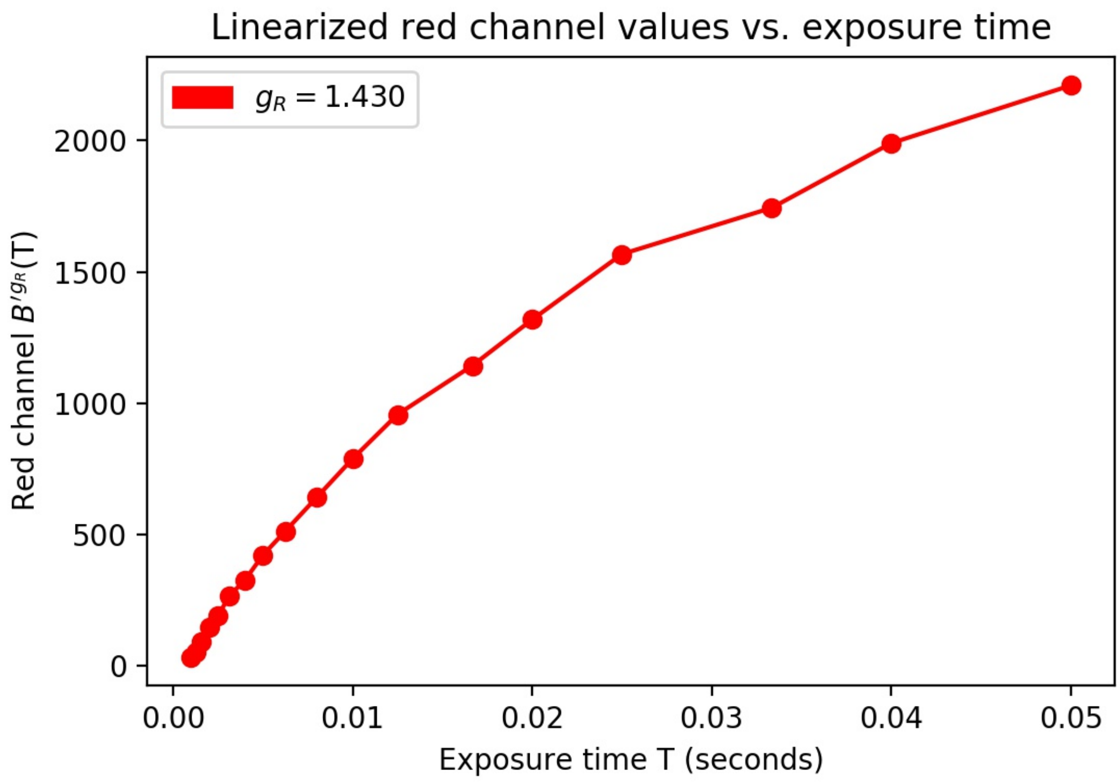

Mean green channel values vs. exposure time
Green channel B'(T)
Exposure time T (seconds)
0
50
100
150
200
250
0.00
0.01
0.02
0.03
0.04
0.05

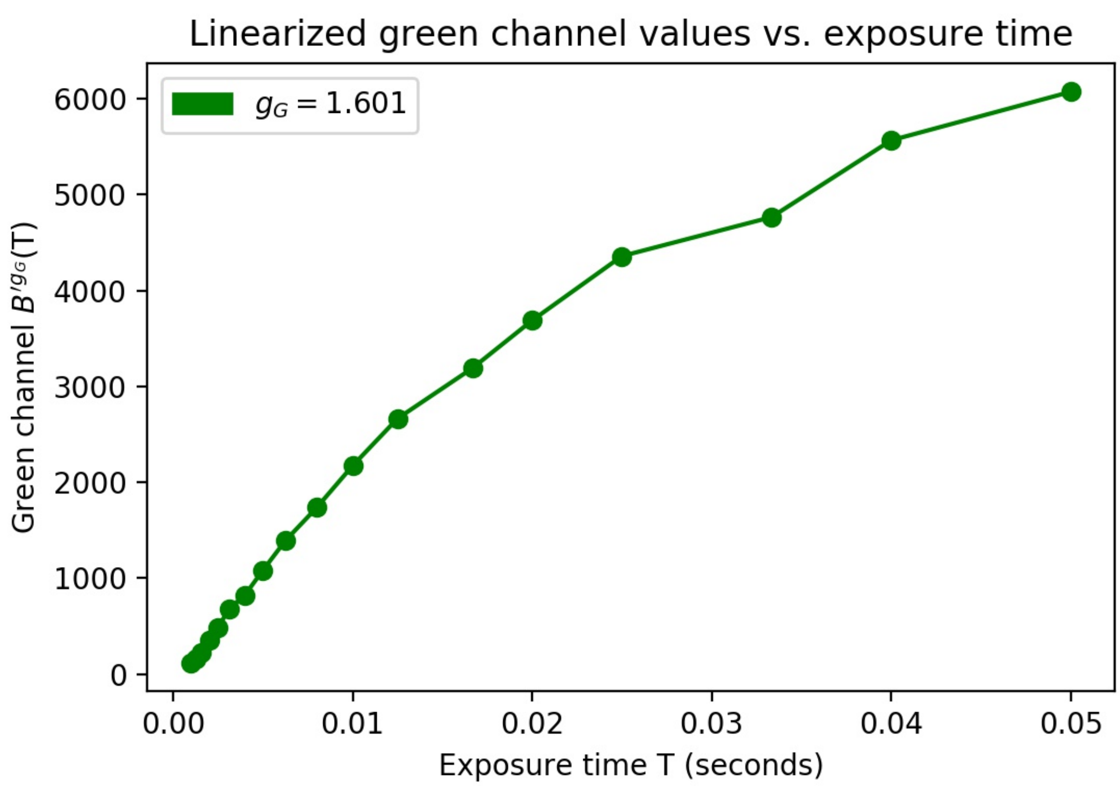


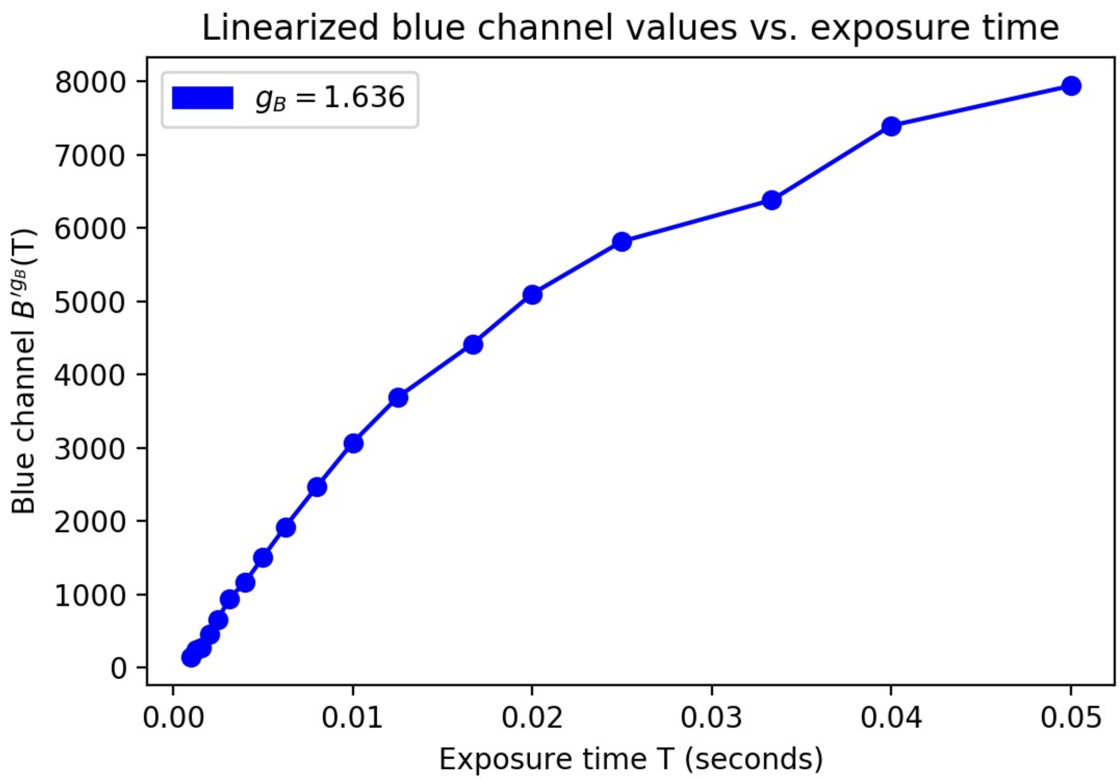
The g values for each channel were obtained by taking the natural log of both $B'(T)$ and T, and using the linregress function from the scipy Python package to obtain the slope of the best fit line to the resulting curve. The g values are simply the inverse of the slope, and are as follows (rounded to 3 significant digits):

- $g_{red} = 1.430$
- $g_{green} = 1.601$
- $g_{blue} = 1.636$

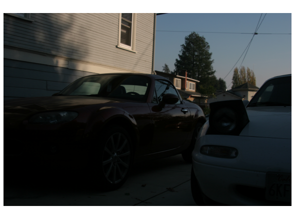
The $B'(T)$ values for each channel were then raised to the power of each respective g value. These $B'^{g}(T)$ Full resolution plots can be found in the PlotsAndHists folder. Here are the plots displayed:

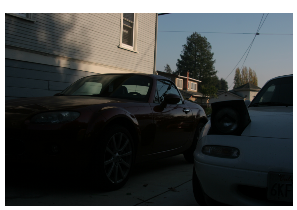

Linearized red channel values vs. exposure time
$g_R = 1.430$
2000
1500
1000
500
0
Red channel $B'^{g_R}(T)$
0.00
0.01
0.02
0.03
0.04
0.05
Exposure time T (seconds)

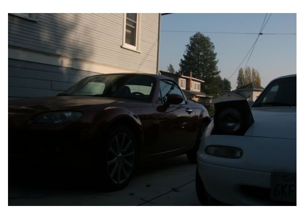

Linearized green channel values vs. exposure time
$g_G = 1.601$
6000
5000
4000
3000
2000
1000
0
Green channel $B'^{g_G}(T)$
0.00
0.01
0.02
0.03
0.04
0.05
Exposure time T (seconds)

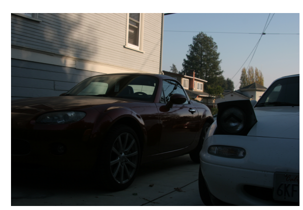


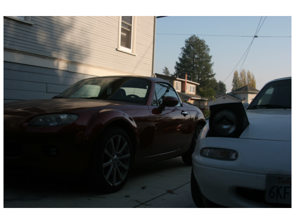
<u>Part 2:</u>
<u>Note</u>: From this point onward (until the last step), the images' data type is numpy float32.
The stack of images taken were taken in the same location and near the same time of day as the radiometric calibration images. The exposure times for these images ranged from 1/1000 sec to 1/15 sec in the same intervals as the radiometric calibration images. The 1/13 sec, 1/10 sec, and 1/8 sec images were not taken due to obvious overexposure already occurring at 1/15 sec. Full resolution images can be found in the (HDRStack folder. Here are the images displayed:

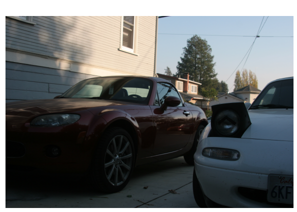

1/1000 sec

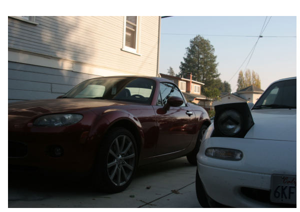

1/800 sec

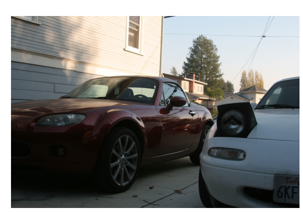

1/640 sec


1/500 sec


1/400 sec


1/320 sec


1/250 sec


1/200 sec


1/160 sec


1/125 sec


1/100 sec


1/80 sec


1/60 sec


1/50 sec


1/40 sec


1/30 sec

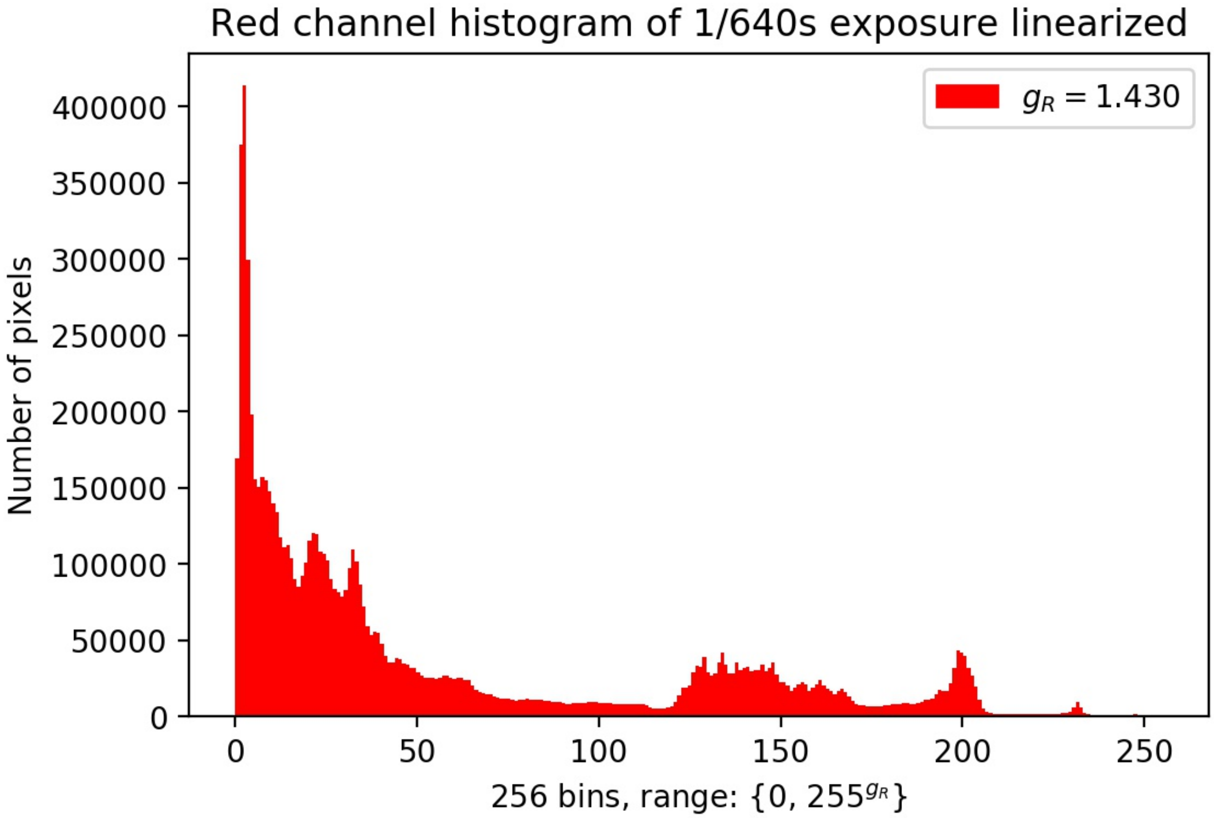

1/25 sec

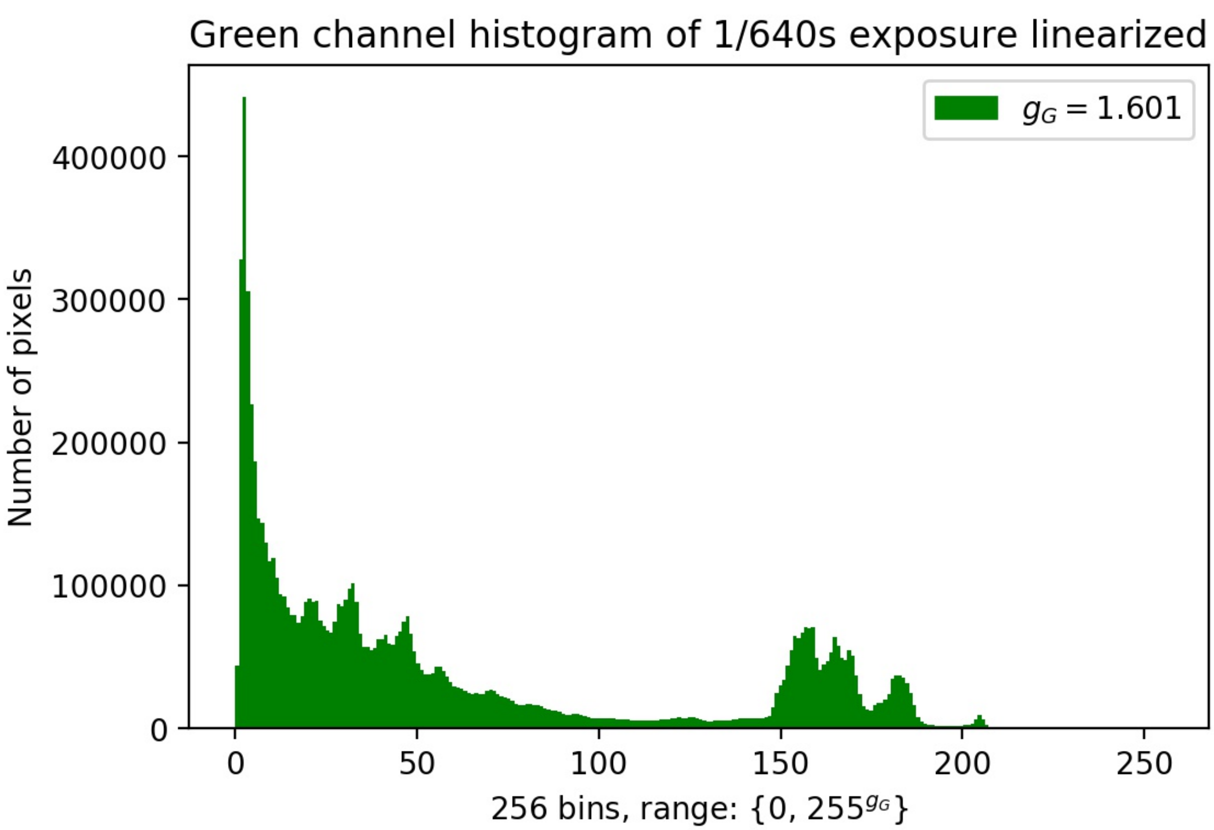

1/20 sec

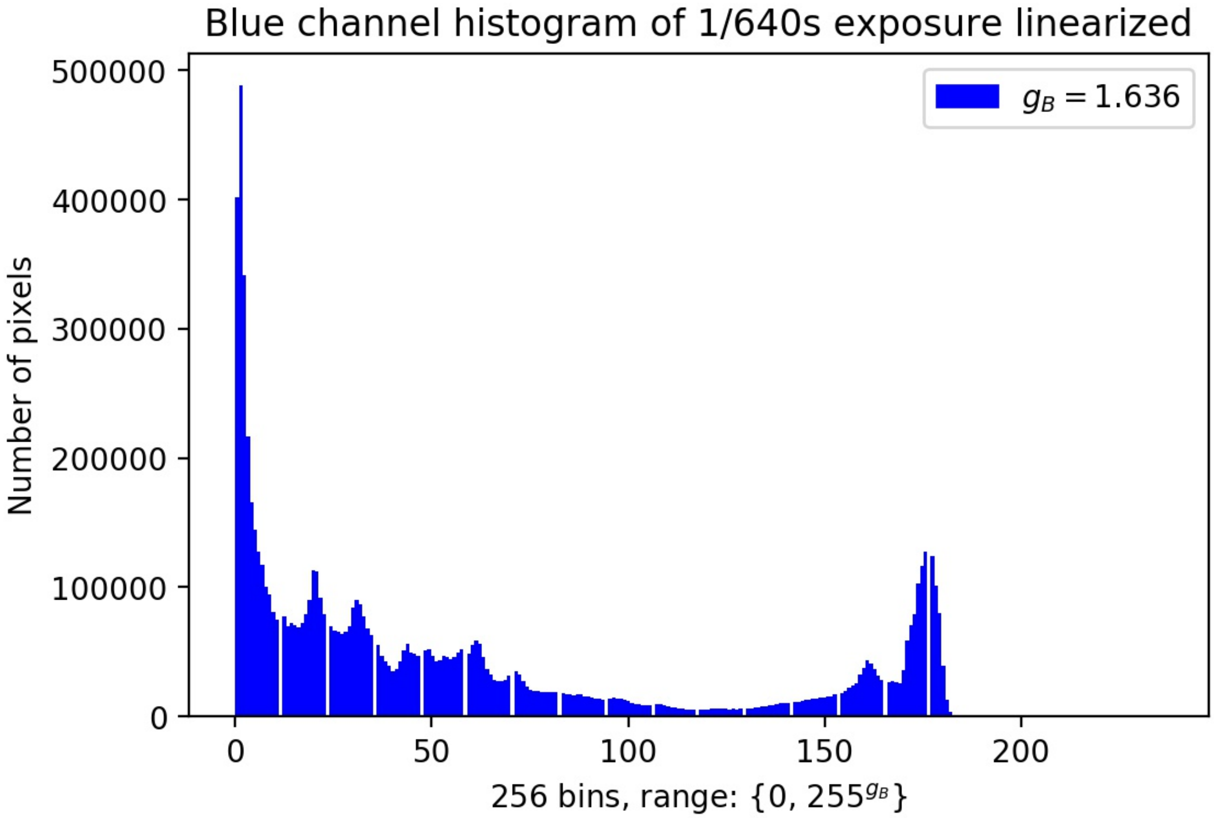

1/15 sec

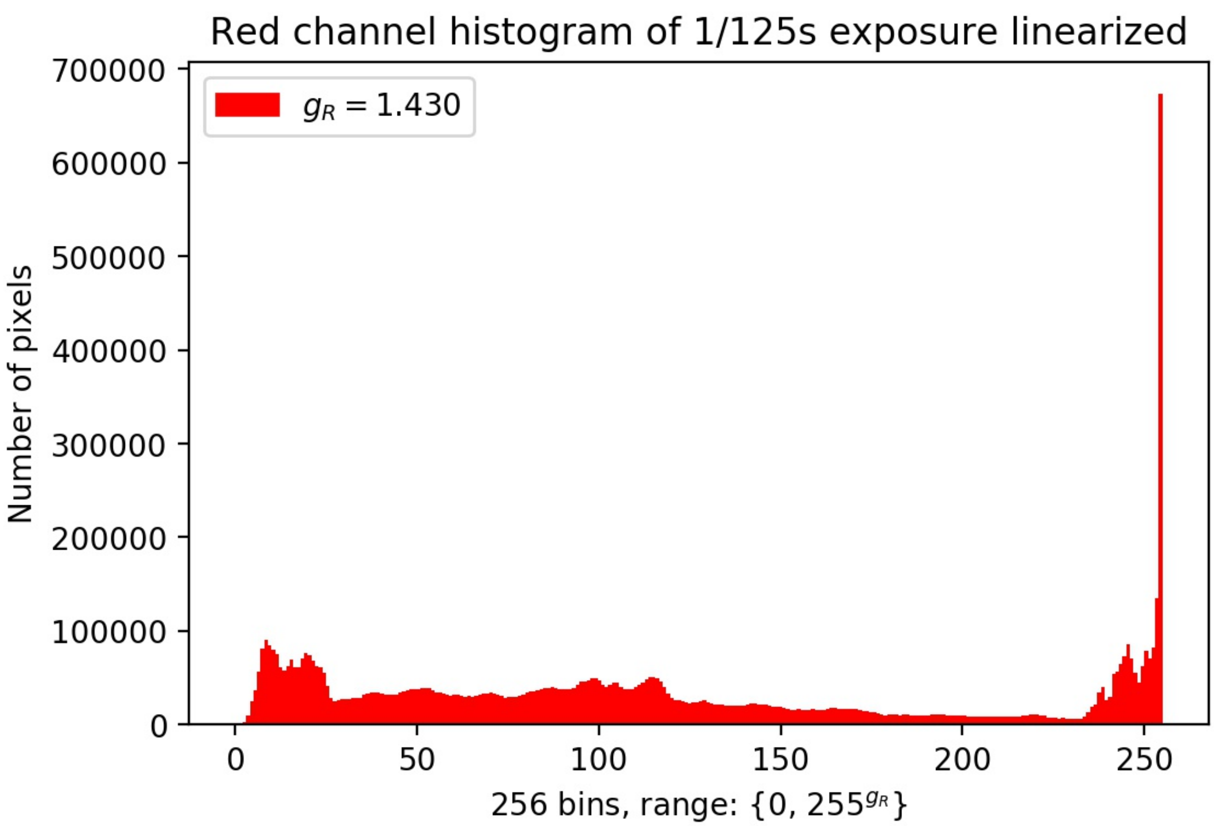
The three images chosen are at exposure times are at **1/640 sec**, **1/125 sec**, and **1/60 sec**. These are also marked with a bold caption in the above images.

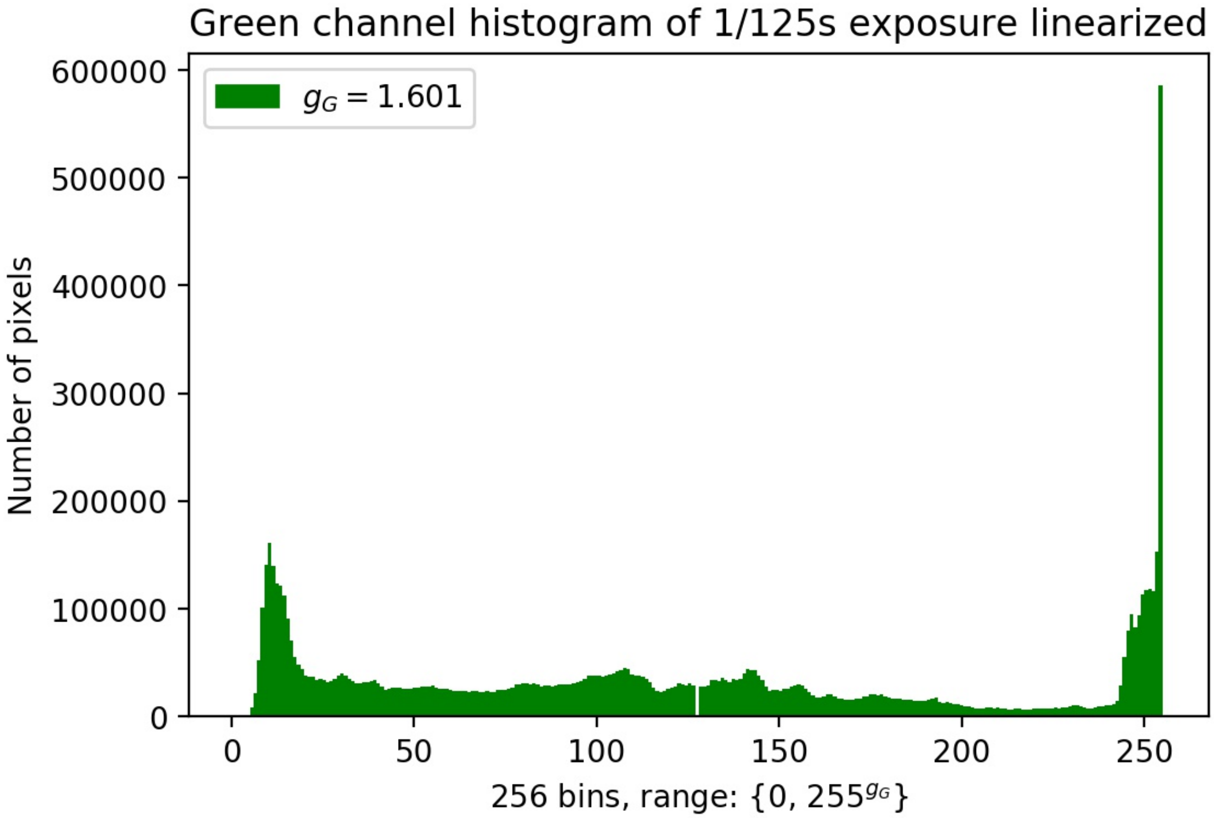
Histograms were then generated for each of the three selected images. The brightness values were linearized by raised to the respective g values for each channel to obtain $B'^{g}(T)$ for each channel. The histograms have 256 bins (typo in assignment shows 25 bins). Full resolution histograms can be found in the PlotsAndHists folder. Here are the histograms displayed:


Red channel histogram of 1/640s exposure linearized
$g_R = 1.430$
Number of pixels
400000
350000
300000
250000
200000
150000
100000
50000
0
0
50
100
150
200
250
256 bins, range: {0, 255^{g_R}}

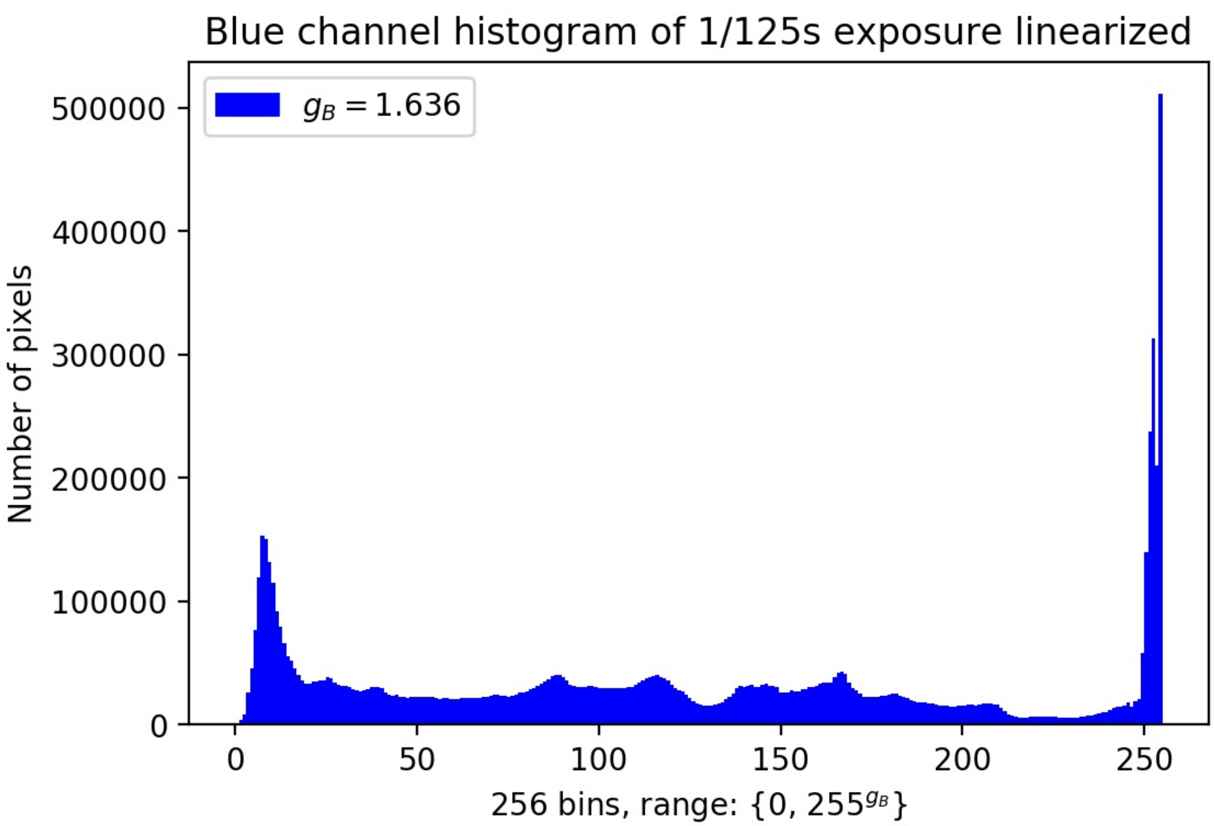

Green channel histogram of 1/640s exposure linearized
$g_G = 1.601$
Number of pixels
400000
300000
200000
100000
0
0
50
100
150
200
250
256 bins, range: {0, 255^{g_G}}


Blue channel histogram of 1/640s exposure linearized
$g_B = 1.636$
Number of pixels
0
100000
200000
300000
400000
500000
0
50
100
150
200
256 bins, range: {0, 255^{g_B}}

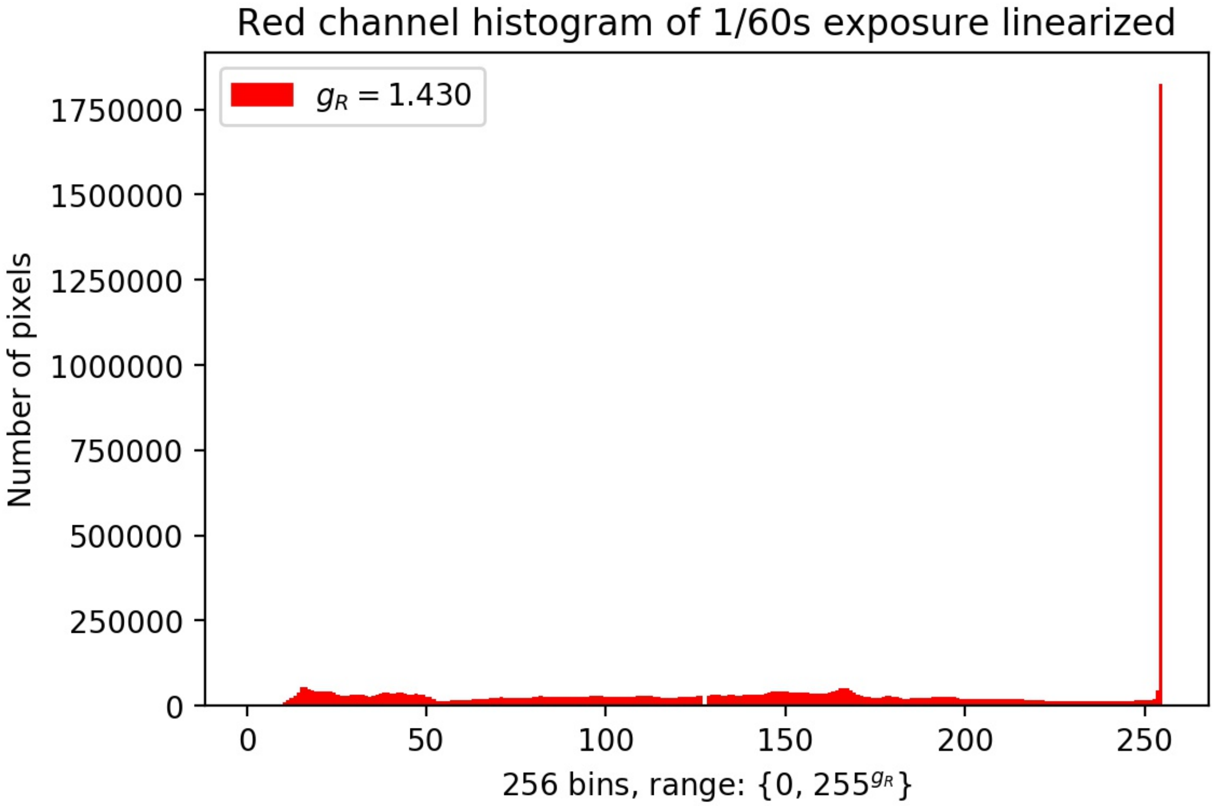

Red channel histogram of 1/125s exposure linearized
$g_R = 1.430$
Number of pixels
0
100000
200000
300000
400000
500000
600000
700000
0
50
100
150
200
250
256 bins, range: {0, 255^{g_R}}

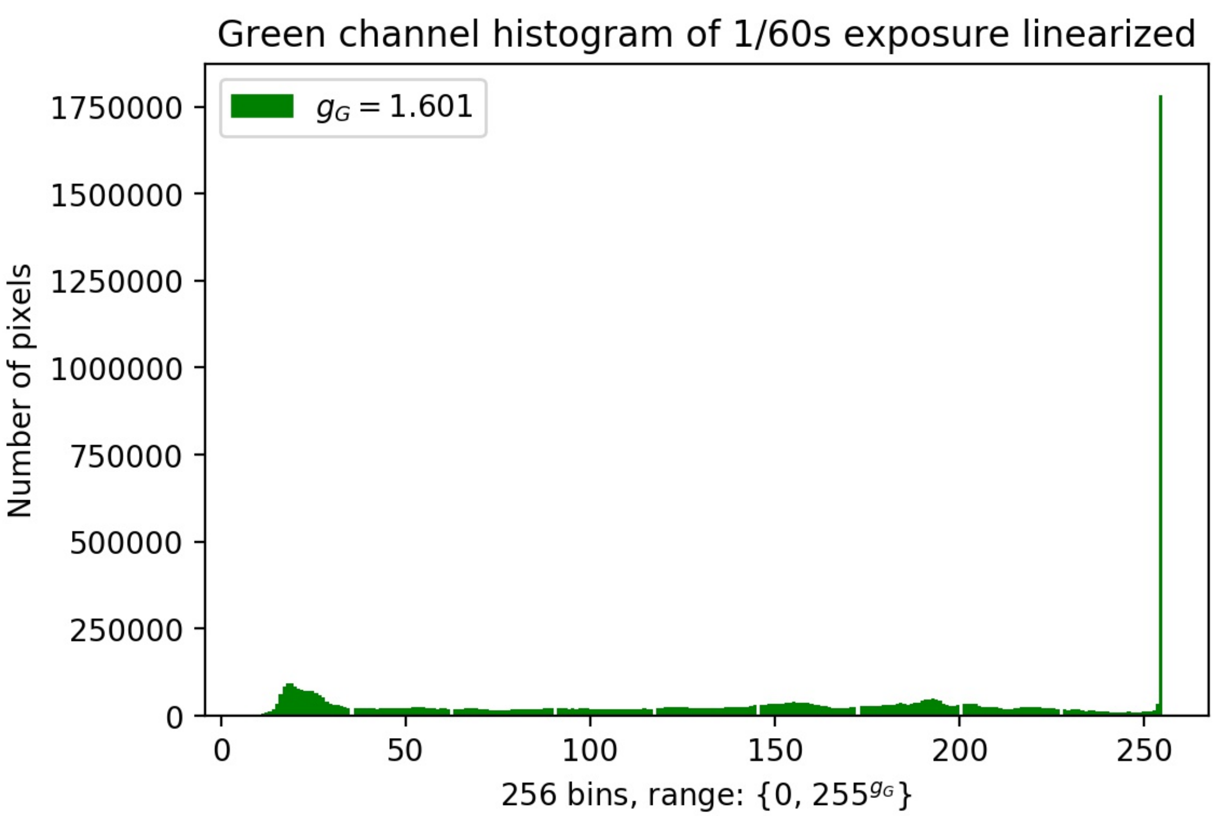


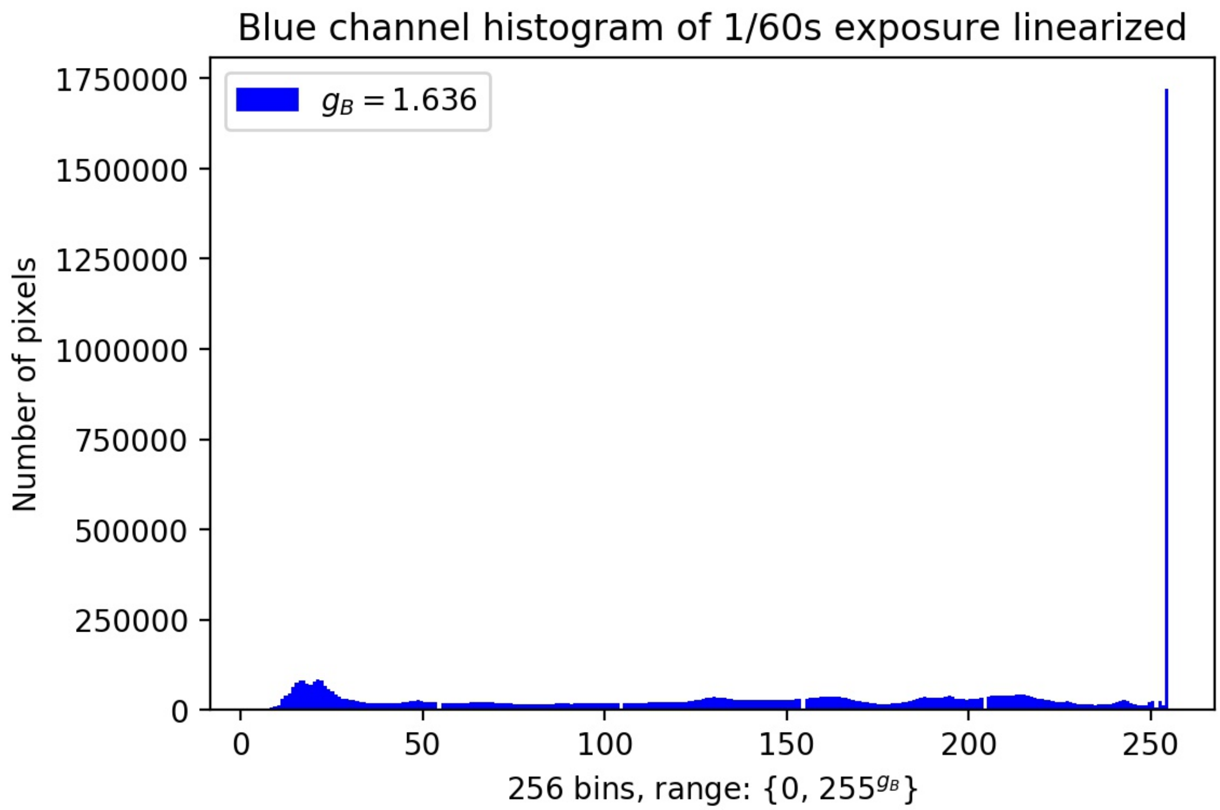
Green channel histogram of 1/125s exposure linearized
$g_G = 1.601$
Number of pixels
0
100000
200000
300000
400000
500000
600000
0
50
100
150
200
250
256 bins, range: {0, 255^{g_G}}



Blue channel histogram of 1/125s exposure linearized
$g_B = 1.636$
Number of pixels
0
100000
200000
300000
400000
500000
0
50
100
150
200
250
256 bins, range: {0, 255^{g_B}}


Red channel histogram of 1/60s exposure linearized
$g_R = 1.430$
Number of pixels
1750000
1500000
1250000
1000000
750000
500000
250000
0
0
50
100
150
200
250
256 bins, range: {0, 255^{g_R}}

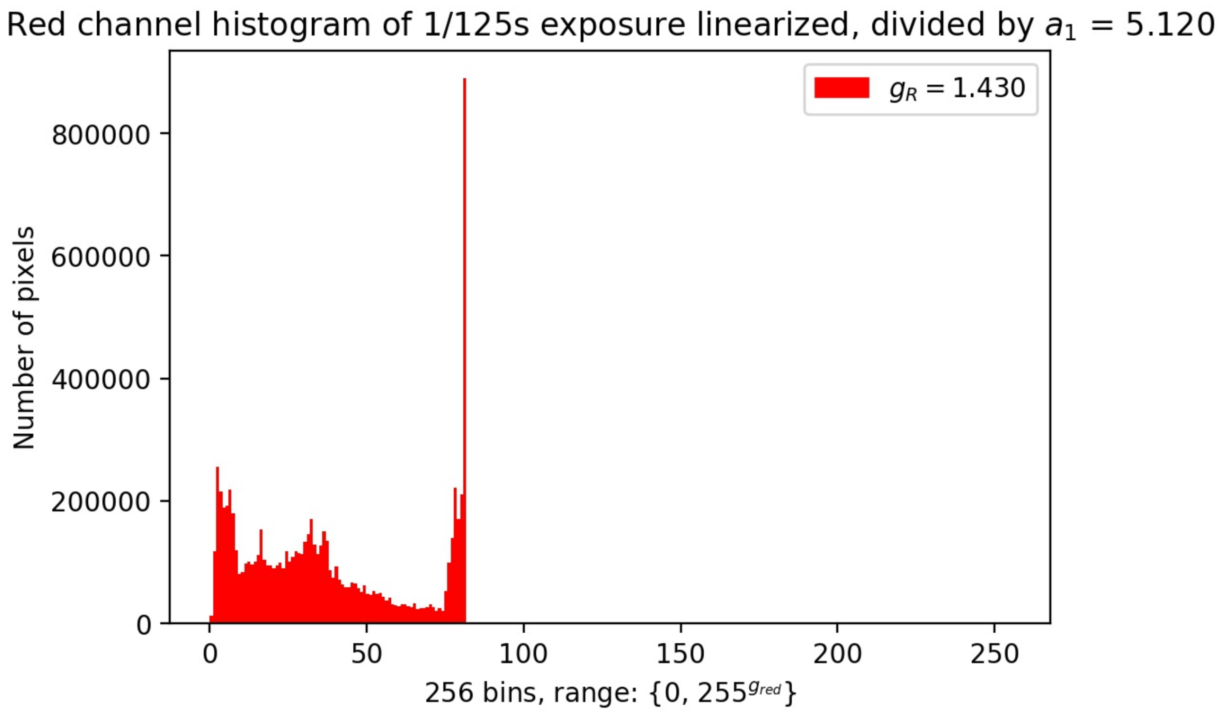

Green channel histogram of 1/60s exposure linearized
$g_G = 1.601$
Number of pixels
1750000
1500000
1250000
1000000
750000
500000
250000
0
0
50
100
150
200
250
256 bins, range: {0, 255^{g_G}}

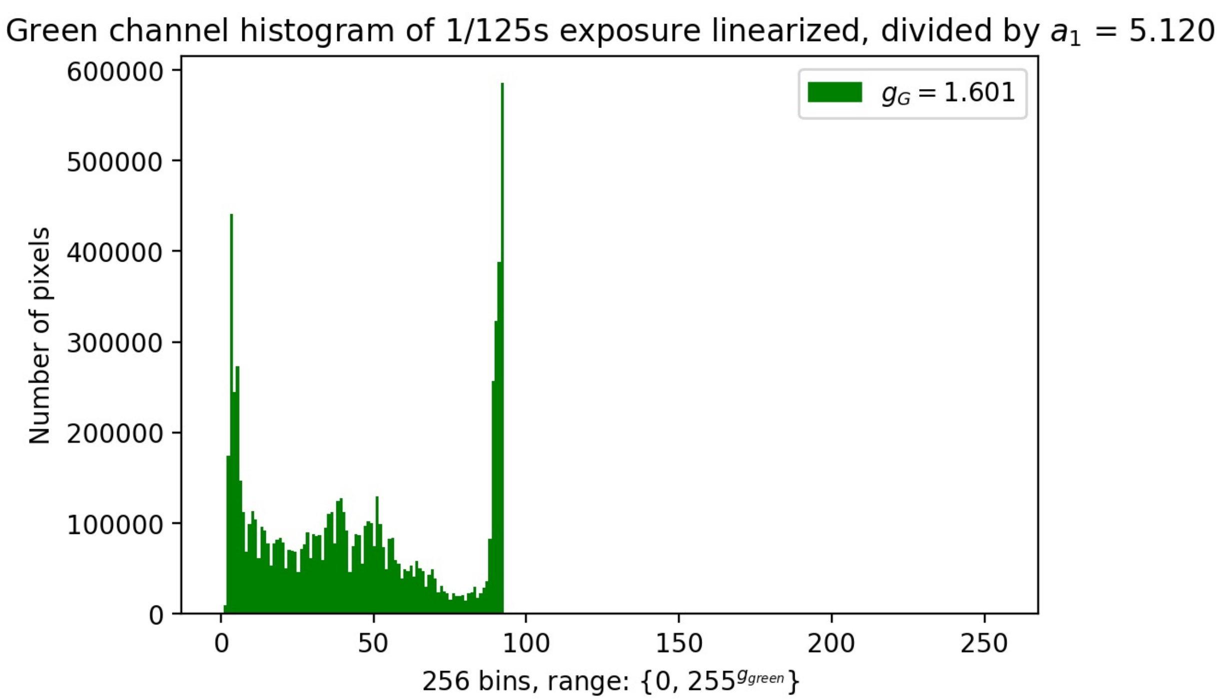


Histograms were also obtained for the second and third images' brightnesses divided by their a values, $\frac{B'^{g}(T)}{a_i}$. The a values were calculated as such:

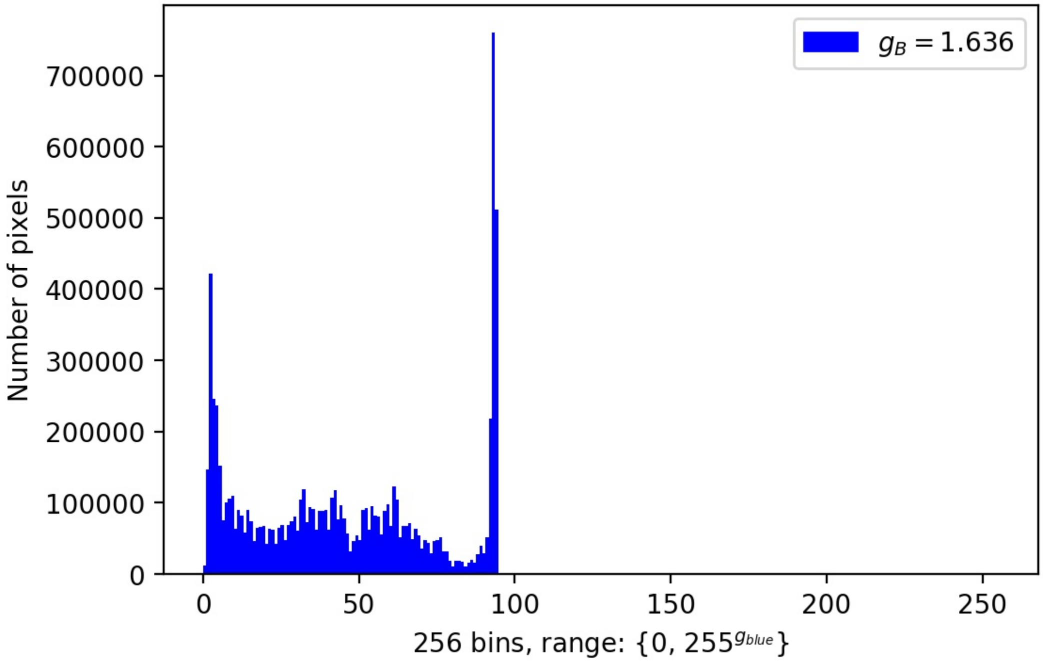
$$a_1 = \frac{T_1}{T_0} = \frac{\frac{1}{125}}{\frac{1}{640}} = 5.120,\ a_2 = \frac{T_2}{T_0} = \frac{\frac{1}{60}}{\frac{1}{640}} = 10.667$$

Full resolution histograms can be found in the PlotsAndHists folder. Here are the histograms displayed:

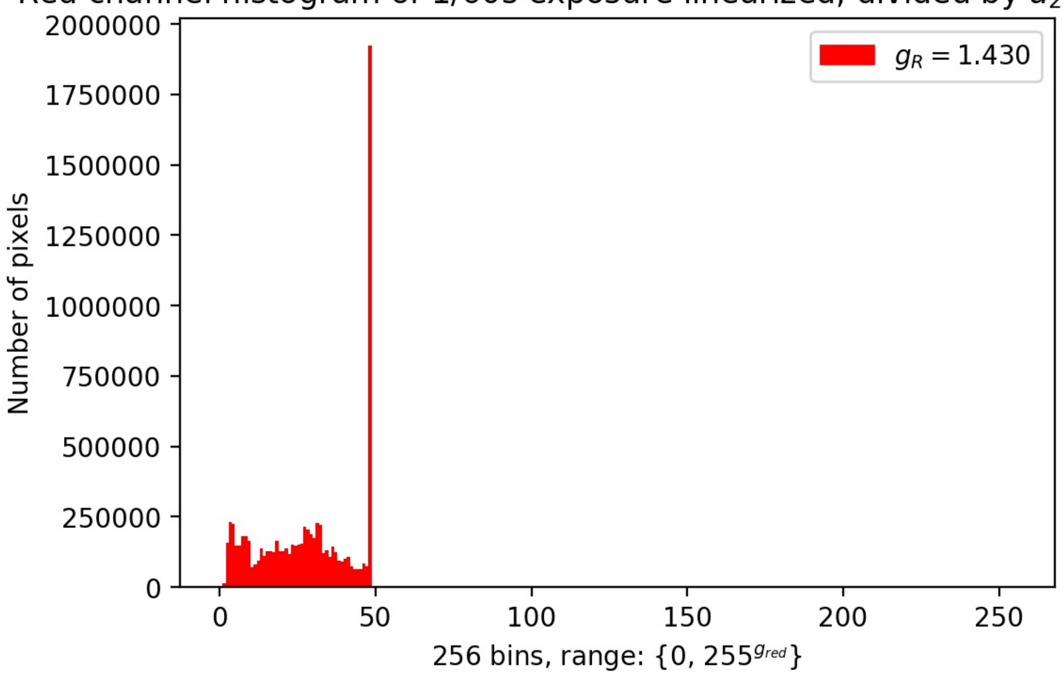

Red channel histogram of 1/125s exposure linearized, divided by a_1 = 5.120
$g_R = 1.430$
Number of pixels
800000
600000
400000
200000
0
0
50
100
150
200
250
256 bins, range: {0, $255^{g_{red}}$}

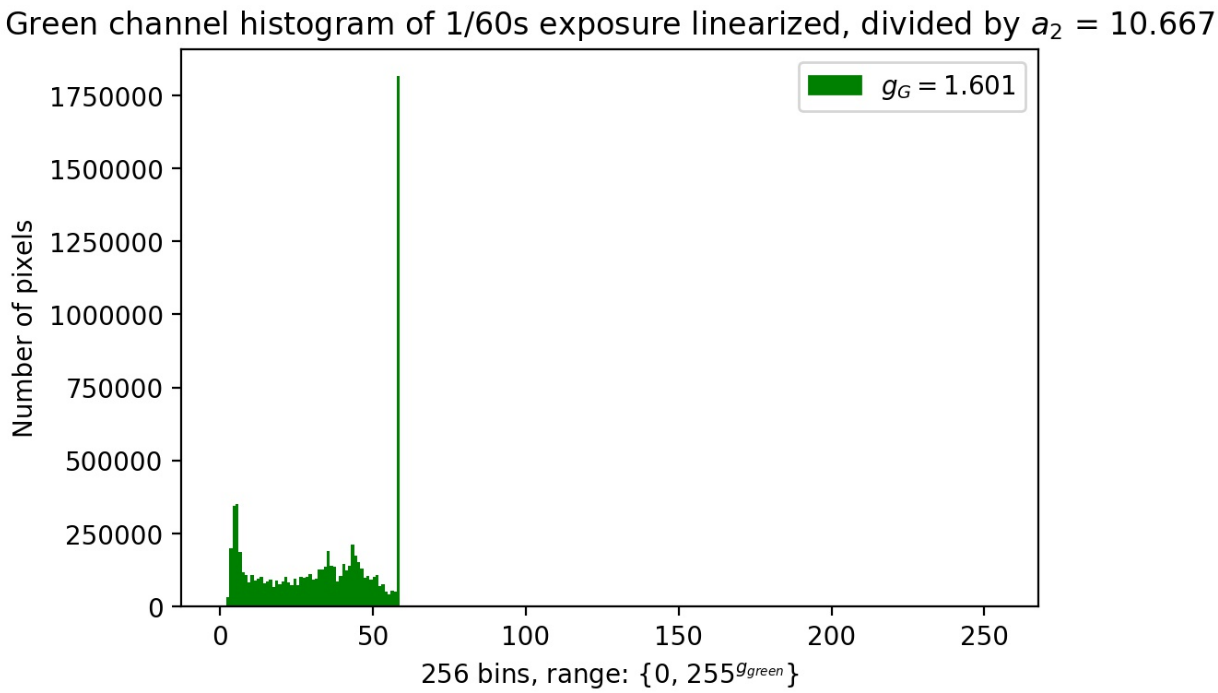

Green channel histogram of 1/125s exposure linearized, divided by a_1 = 5.120
$g_G = 1.601$
Number of pixels
600000
500000
400000
300000
200000
100000
0
0
50
100
150
200
250
256 bins, range: {0, $255^{g_{green}}$}

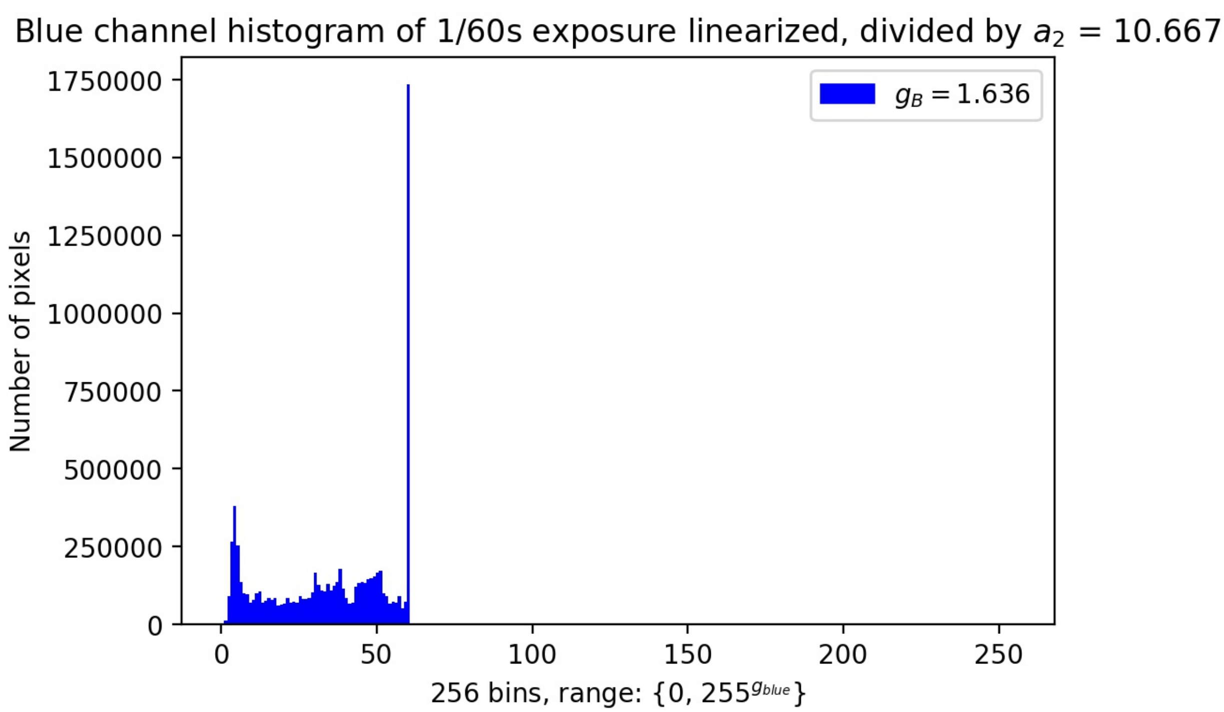
Blue channel histogram of 1/125s exposure linearized, divided by a_1 = 5.120



Red channel histogram of 1/60s exposure linearized, divided by a_2 = 10.667




Green channel histogram of 1/60s exposure linearized, divided by a_2 = 10.667
$g_G = 1.601$
Number of pixels
1750000
1500000
1250000
1000000
750000
500000
250000
0
0
50
100
150
200
250
256 bins, range: {0, $255^{g_{green}}$}


Blue channel histogram of 1/60s exposure linearized, divided by a_2 = 10.667
$g_B = 1.636$
Number of pixels
1750000
1500000
1250000
1000000
750000
500000
250000
0
0
50
100
150
200
250
256 bins, range: {0, $255^{g_{blue}}$}

Part 3:

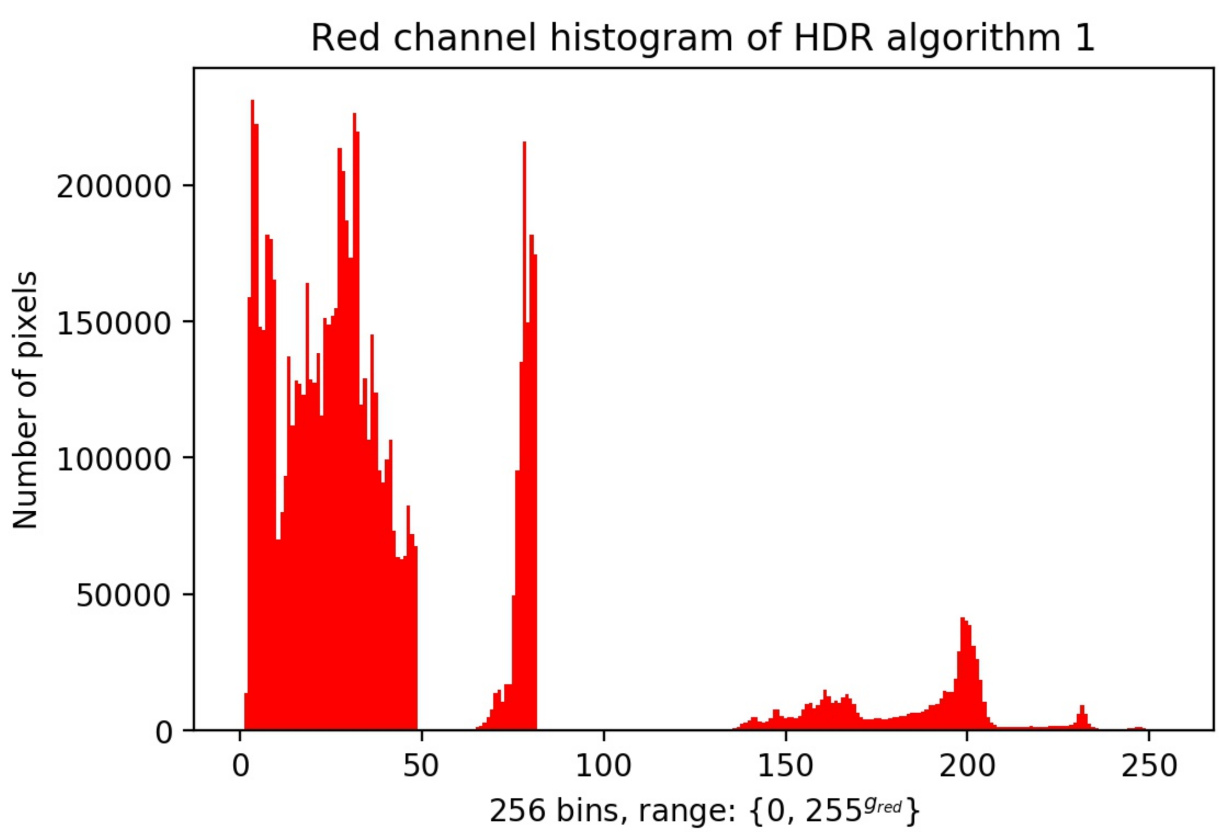
Notation:

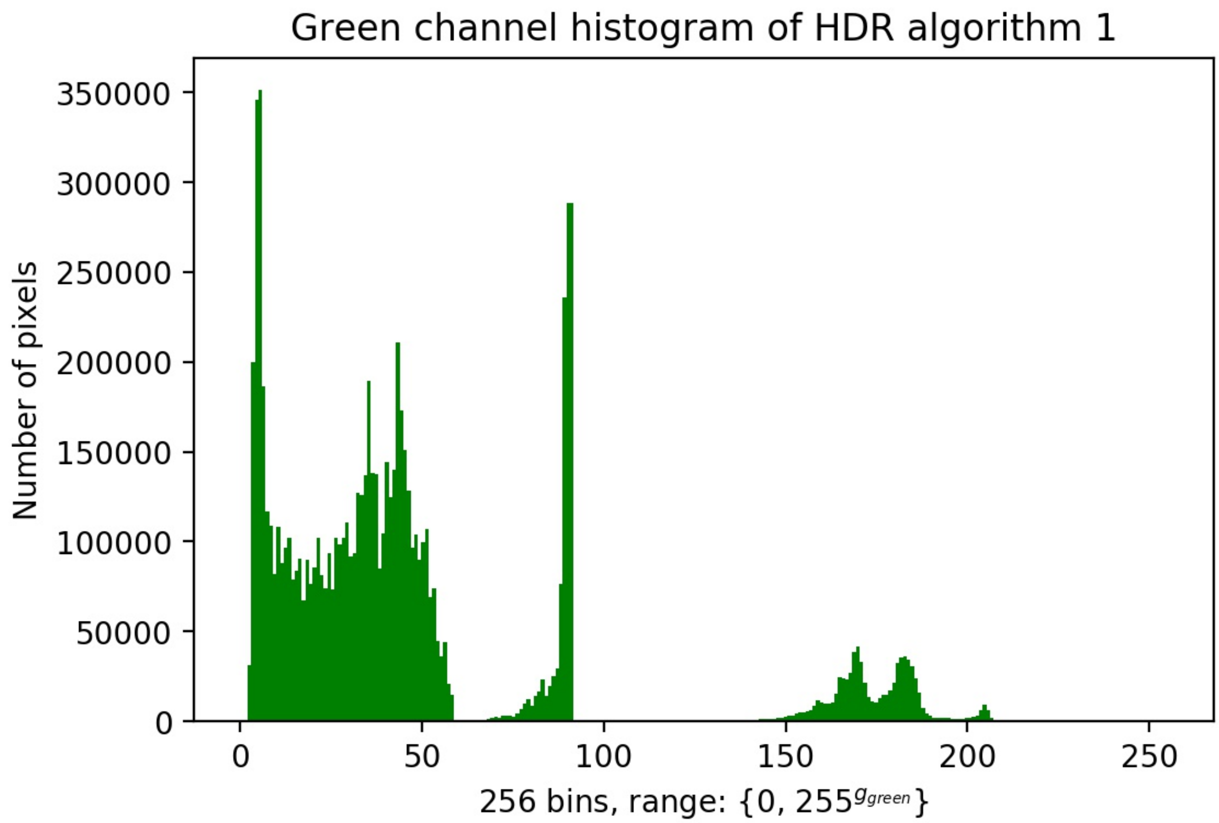
- Images I_i are the non-modified (non-linearized, not compensated for a_i) images with corresponding a_i values.
- Images I_i' are the modified linearized, compensated for a_i) images with corresponding a_i values.

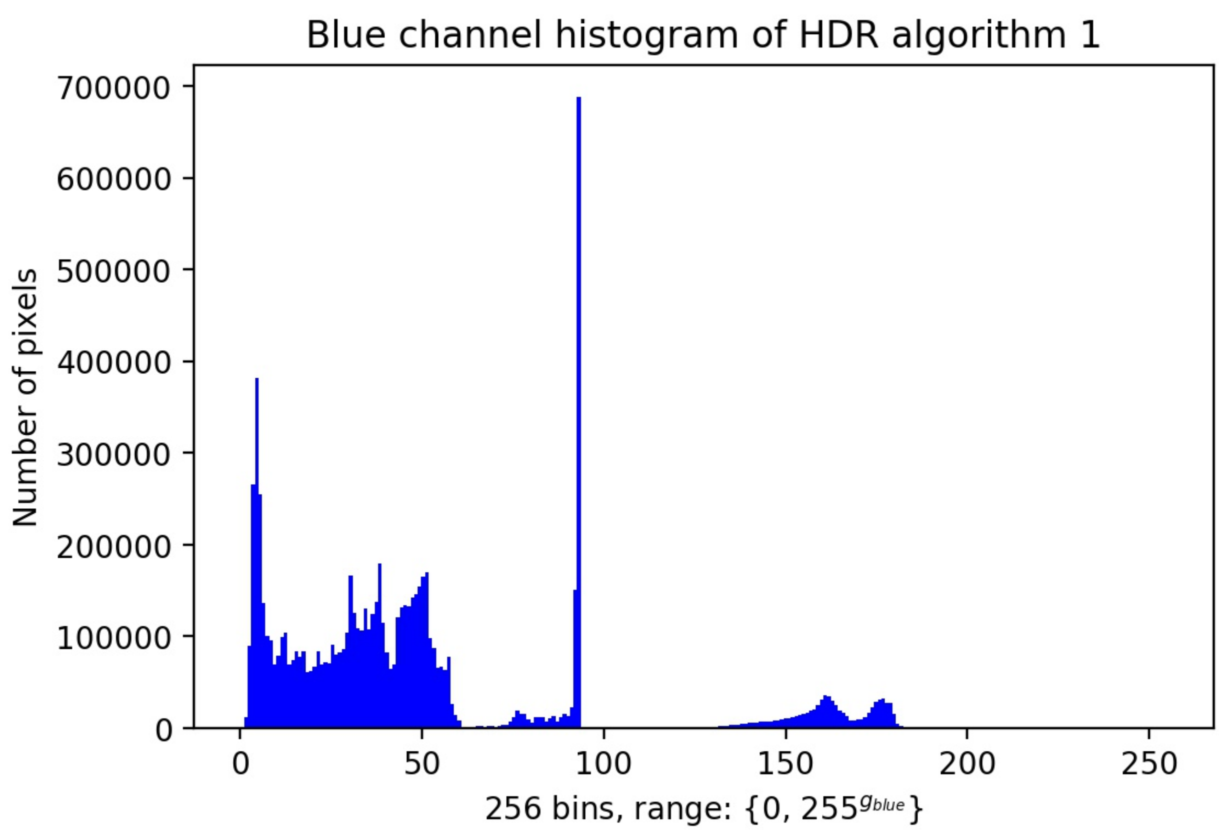
The first algorithm was done by iterating over the pixels of I_2. A copy of I_2' was made. For each pixel in I_2, if any channel in said pixel is saturated, replace the entire corresponding pixel in I_2' with the same pixel in I_1'.
This loop is then repeated with the composite image composed of $\{I_2', I_1'\}$, replacing pixels that still contain saturated channels with the entire corresponding pixel from I_0'.

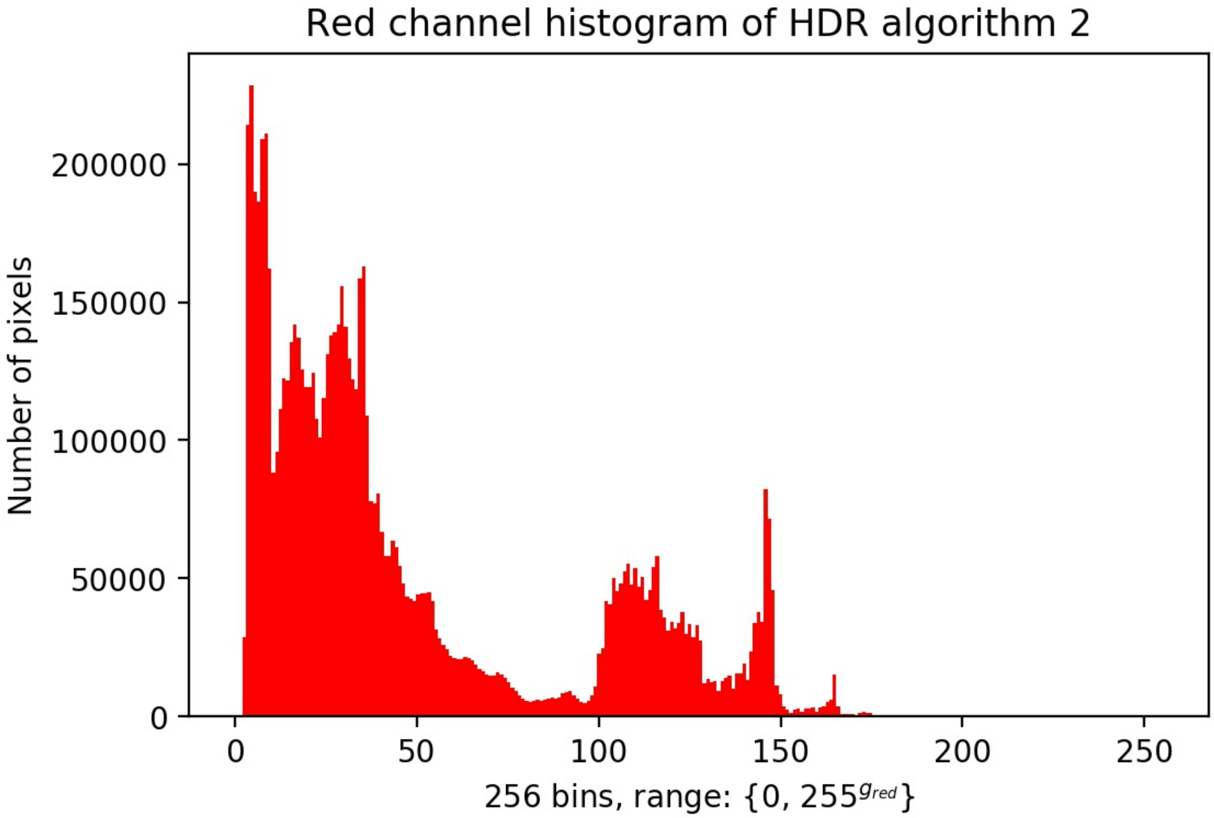
The second algorithm also iterates over the pixels of I_2, but only contains one loop. For each pixel, there is a set of nested if statements that compute the average of the channels for each pixel with no saturated channels.

Histograms for each color channel were generated from the resulting composite images for each algorithms. Full resolution histograms can be found in the PlotsAndHists folder. Here are the histograms displayed:

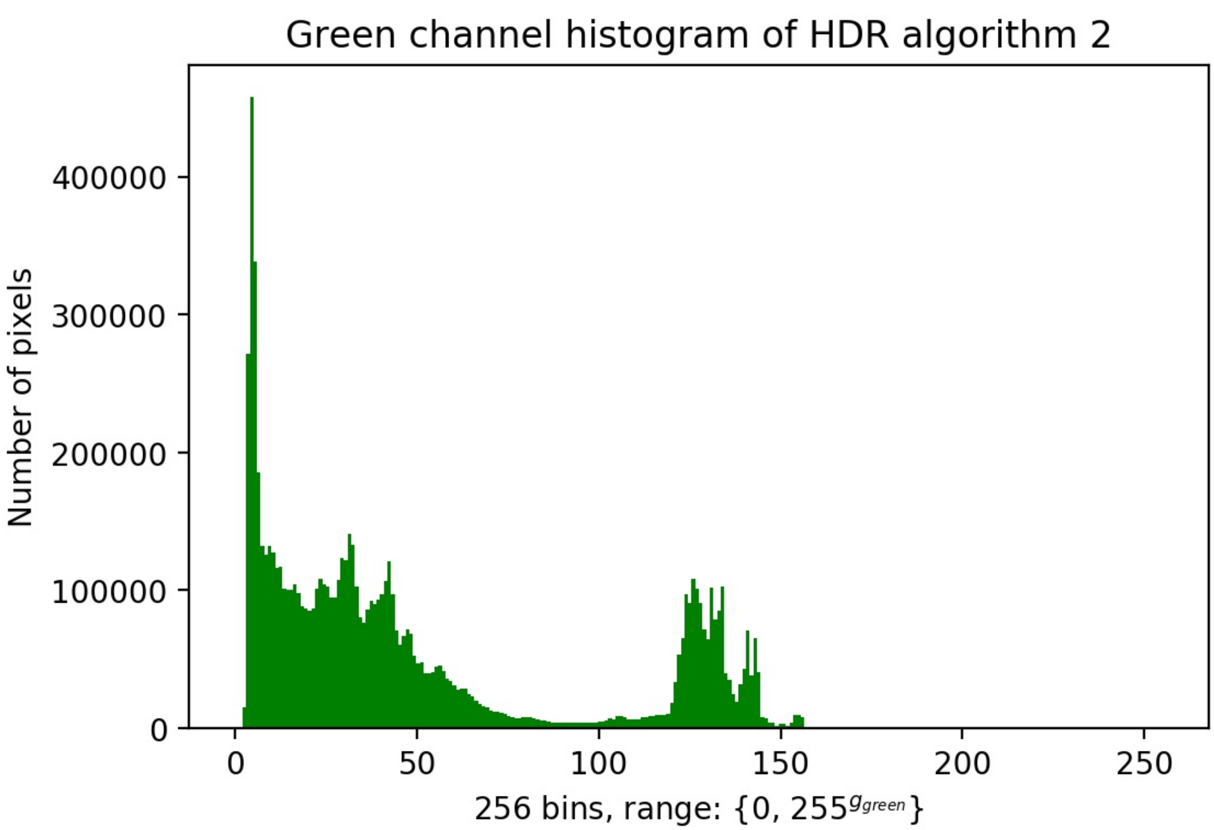



Green channel histogram of HDR algorithm 1
Number of pixels
0
50000
100000
150000
200000
250000
300000
350000
0
50
100
150
200
250
256 bins, range: {0, 255$^{g_{green}}$}

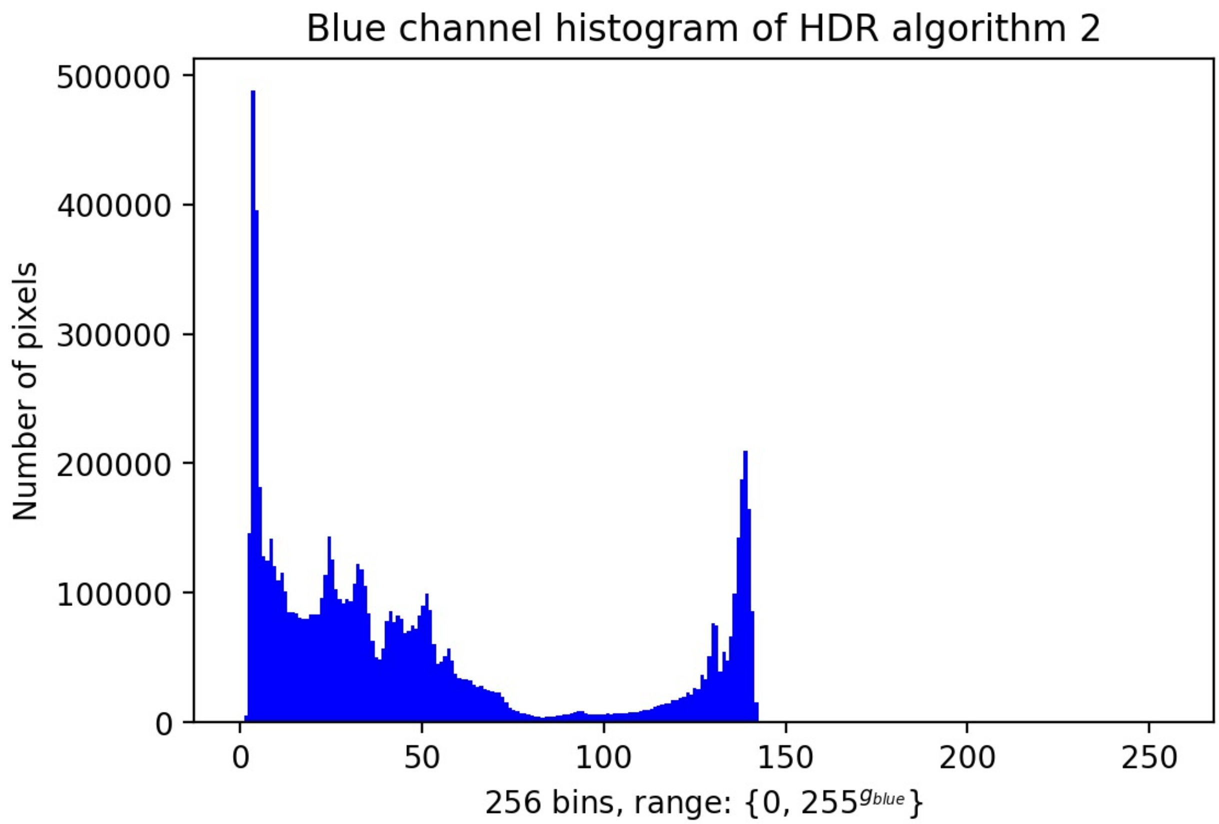

Blue channel histogram of HDR algorithm 1
Number of pixels
0
100000
200000
300000
400000
500000
600000
700000
0
50
100
150
200
250
256 bins, range: {0, 255$^{g_{blue}}$}



Red channel histogram of HDR algorithm 2
Number of pixels
200000
150000
100000
50000
0
0
50
100
150
200
250
256 bins, range: {0, 255^{g_{red}}}



Green channel histogram of HDR algorithm 2
Number of pixels
400000
300000
200000
100000
0
0
50
100
150
200
250
256 bins, range: {0, 255^{g_{green}}}



Part 4:

Tone mapping of the composite images was done using the Reinhard tonemap class in OpenCV, and then converted back to uint8 for writing to disk. The gamma, intensity, light adapt, and color adapt parameters of the Reinhard tone map function were chosen manually to produce the most visually accurate results in terms of brightness and color. Full resolution images can be found in the HDRStack folder. Here are the images displayed:

HDR algorithm 1, tone mapped

HDR algorithm 2, tone mapped

Conclusions:

The resulting tone mapped HDR images are a success. From the original three images, dark areas are visible and saturated areas are not saturated. The borders of the saturated patches for the $T \cdot a_1$, $T \cdot a_2$ images are very apparent when using the first algorithm. However, this is almost entirely mitigated by using the second algorithm, with the exception of a soft shadow edge along the side of the house in the upper left quadrant of the image.

There seems to be a slight blue tint in both of the resulting tone mapped HDR images, which was minimized when adjusting the Reinhard tone mapping parameters. The best assumption for why this would be the case is due to some oddities in the blue channel when performing radiometric calibration. This may have given us a less correct g_{blue} value for our first order approximation. The fact that only a first order approximation from a linear regression was used implies a non-perfect mapping anyways. Shown are the plots of $log(B'(T))$ versus $log(T)$. Notice the non-conformities in the blue channel that are not present in the red or green channels for low $log(T)$ values: