Computer Vision Project 1

Weiting Zhan(wzhan83@ucsc.edu) Yuan Yang (yyang175@ucsc.edu)

Sunday November 18 2018

1 Camera radiometric calibration

1.1 Introduction to Camera Radiometric Calibration

If the camera were linear, we should get the relationship of B (Brightness), E (current [electrons/s]), T(exposure time) and G(Gain) in the equation 1.

$$B = c \bullet E \bullet T \bullet G \tag{1}$$

where c is an proportionality factor.

We cannot measure B directly, instead, the camera gives $B^{'}$:

$$B^{'} = f(B) \tag{2}$$

where $f(\bullet)$ is an unknown non-linear function, which changes from camera to camera.

For the first part, we focus on reverse-engineer the function $f(\bullet)$ on each color channel. $f_R(\bullet)$, $f_G(\bullet)$, $f_B(\bullet)$.

1.2 Method

1.2.1 Collect Data

1.Our camera is Iphone7 with APP Camera+ . Take 10 pictures of a white paper with uniform radiance with different exposure time T and set G = 32 ISO.

$$T = [1/8772s, 1/6410s, 1/4167s, 1/3096s, 1/2037s, 1/1520s, 1/1008s, 1/754s, 1/501s, 1/351s]$$

The photos we used for calibration, as shown in Figure 1

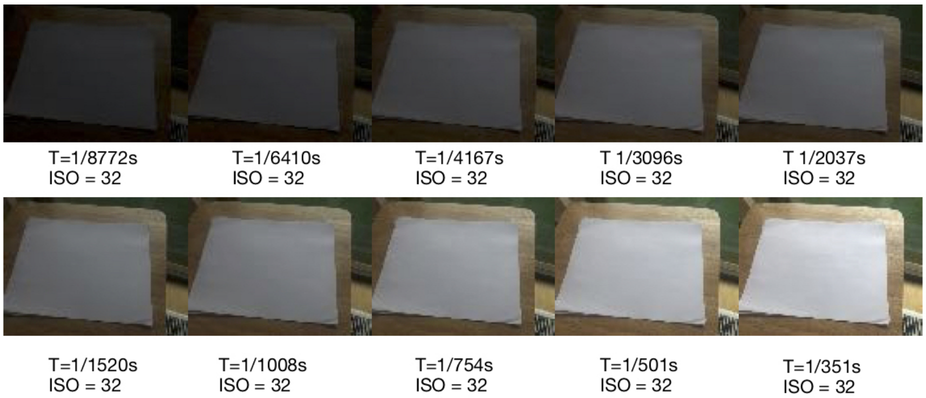

Figure 1: photos for calibration

Crop out a central portion (768×512) of each image, as shown in Figure 6

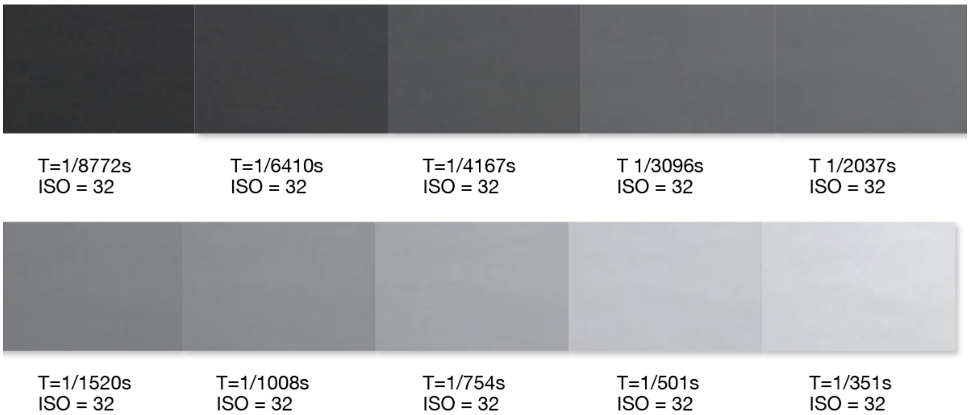

Figure 2: photos for calibration after cropping

1.2.2 Data preparation

a. break the original image into RGB channel.
b. For each channel, compute the average value in the central area. Make sure $B^{'}(T_1) < 40$ and $230 < B^{'}(T_n) < 255$

1.2.3 Estimate parameter g

We assume that $f(\bullet)$ has following format:

$$B^{'} = f(B) = B^{\frac{1}{g}} \tag{3}$$

which known as first-order approximation.

From equation 3, we can get the original brightness:

$$B = (B^{'})^{g} \tag{4}$$

We use the brightness of each channel and corresponding exposure time to estimate the unknown value g. we define K as:

$$K = (c \bullet E)^{\frac{1}{g}} \tag{5}$$

So that:

$$B'(T) = f(B(T)) = f(c \bullet E \bullet T \bullet G) = K \bullet T^{\frac{1}{g}} \tag{6}$$

As shown in Figure 3, is a plot of the measured B' as a function of T :

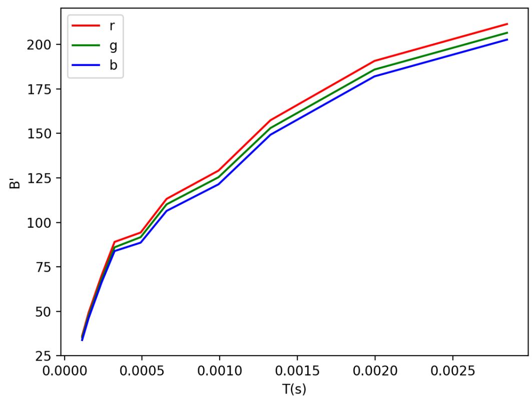

Figure 3: Relation between B'and T

Took logarithm of B':

$$\log(B'(T)) = \log(K) + \frac{1}{g} \bullet \log(T) \tag{7}$$

$\log K$ and $\frac{1}{g}$ are constants in equation 7.

Apply linear regression to B' and T, we get $\frac{1}{g}$, $\log(K)$ and g. for each channel.

channel	log K	$\frac{1}{g}$	g
Red	8.49237	0.520165	1.922
Green	8.47882	0.522226	1.923
Blue	8.47859	0.525988	1.887

Plot B'^g against T, we got Figure 4

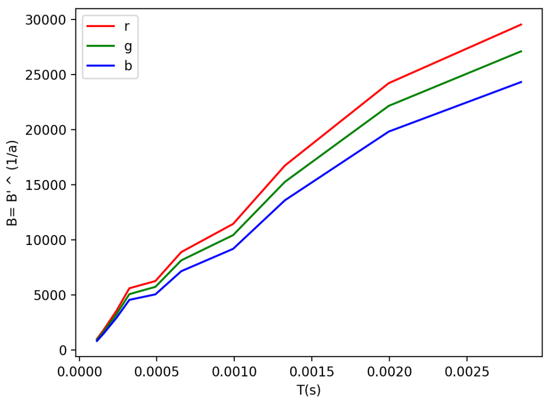

Figure 4: Relation between B'^g and T

2 Acquire a picture stack

2.1 full stack of images

We took 13 images in total at different T with fixed G.

Figure 5: Full image stack

2.2 3 images

Image 1 : Optimal exposure, chosen at the largest T that gives no saturated pixels. T = 1/2037 s.

Image 2: Chosen based on the image contrast. $T_1 = 1/351s, a_1 = 5.80342$

Image 3: Chosen based on the image contrast. $T_1 = 1/180s, a_2 = 11.3167$

(a) 1/2037s (b) 1/351s (c) 1/180s

Figure 6: Image with different exposure time

2.3 Histogram of $B'^g(T)$, $B'^g(a_1T)$ and $B'^g(a_2T)$

Repeat part 1, transform uint8 into 32bit float. Plot the Histogram of $B'^g(T)$, $B'^g(a_1T)$ and $B'^g(a_2T)$ as shown in Figure 7 , Figure8 , Figure 9 , respectively.

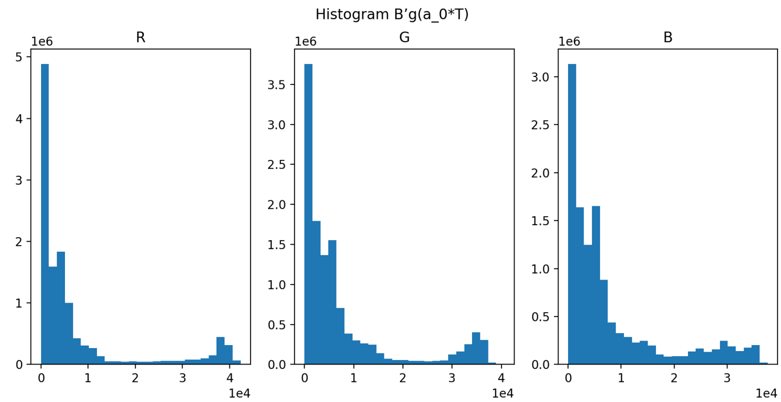

Figure 7: Histogram of $B'^g(T)$

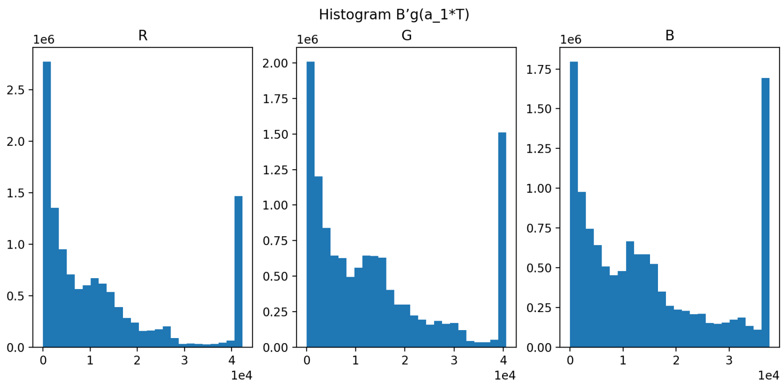

Figure 8: Histogram of $B'^g(a_1T)$

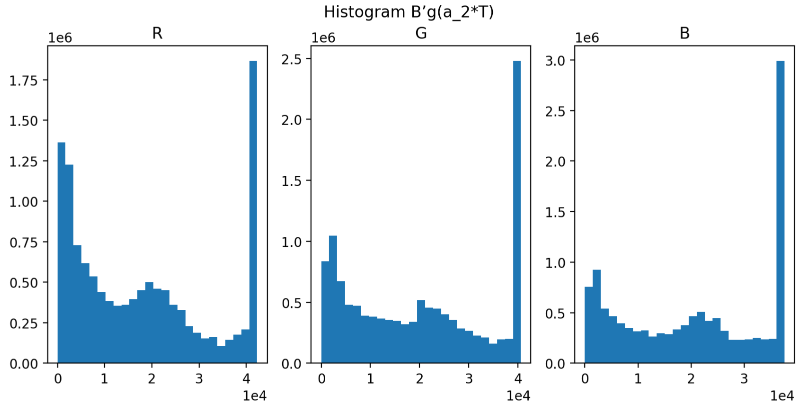

Figure 9: Histogram of $B'^g(a_2T)$

2.4 Histogram of $\frac{B'^g(a_1T)}{a_1}$ and $\frac{B'^g(a_2T)}{a_2}$

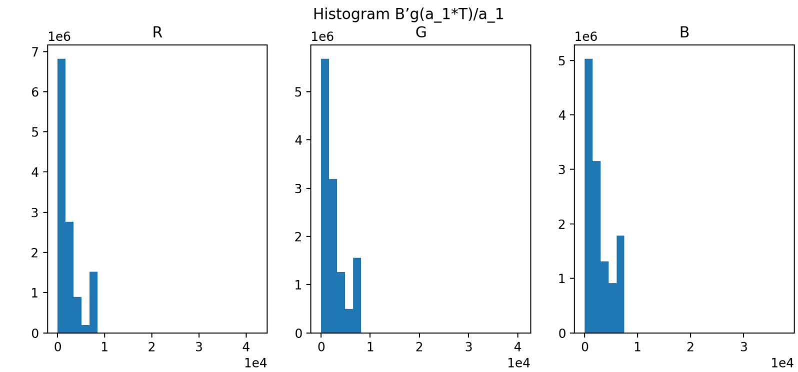

Figure 10: Histogram of $\frac{B'^g(a_1T)}{a_1}$

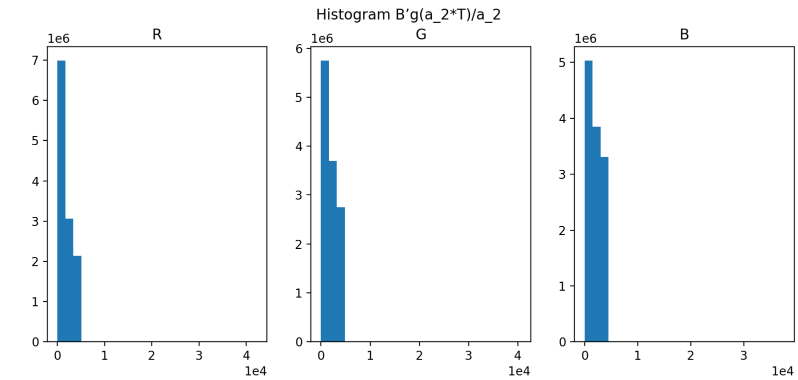

Figure 11: Histogram of $\frac{B'^g(a_2T)}{a_2}$

3 Create a composite image

3.1 Composition Algorithm 1

For each pixel, use the pixel value of the image with largest exposure time such that the pixel is not satuarted.

If the pixel is saturated in the second and third image, we use the pixel value in the first image, i.e. $B'^{g}(T)$

If the pixel is saturated in the third image, we use the pixel value in the second image divided by a_1. i.e. $\frac{B'^{g}(a_1T)}{a_1}$

If the pixel is not saturated in three images, we use the pixel value in the third image divided by a_2. i.e. $\frac{B'^{g}(a_2T)}{a_2}$

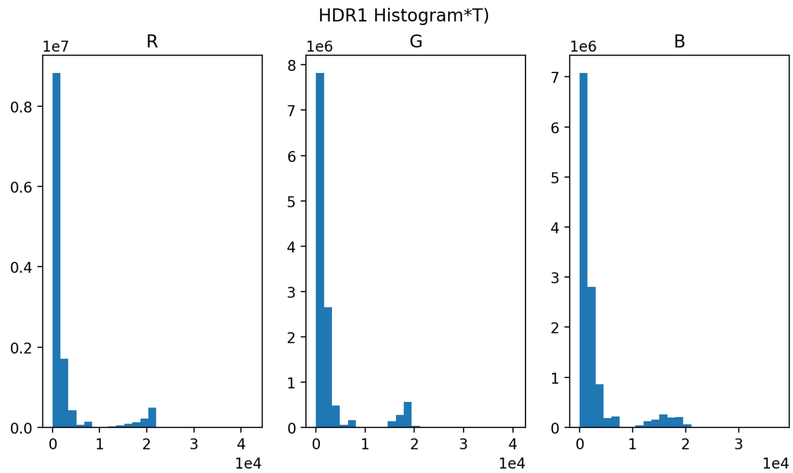

Figure 12: HDR1 algorithm

3.2 Composition Algorithm 2

For each pixel, use the average pixel values of the images that is not satuarted.
If the pixel is saturated in the second and third image, we use the pixel value in the first image, i.e. $B'^{g}(T)$.
If the pixel is saturated in the third image, we use the average pixel value for the first and second image divided by a_1. i.e. $\frac{B'^{g}(T)+B'^{g}(a_1T)/a_1}{2}$.
If the pixel is not saturated in three images, we use the average pixel values of the first image, second image divided by a_1 and the thrid image divided by a_2. i.e. $\frac{B'^{g}(T)+B'^{g}(a_1T)/a_1+B'^{g}(a_2T)/a_2)}{3}$.

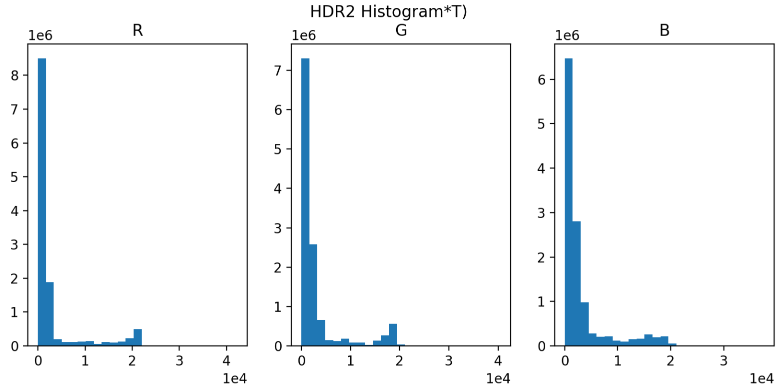

Figure 13: HDR2 algorithm

4 Reproduce composite image

The tone-mapped composite images form Algorithm 1 and Algorithm 2.

(a) HDR algorithm1

(b) HDR algorithm2

Figure 14: HDR image using Algorithm 1 and Algorithm 2

5 Code

Weiting Zhan implement the homework using Python in jupyter notebook.

Github:https://github.com/WaitingZhan/HDR

Yuan Yang implement the homework using C++ and using Python to plot.

Github:https://github.com/yuanyang7/HDR_System