

EECS6323 – Assignment 5: White balancing

Assigned: Feb 6 (Tu)

Due: Feb 16 (Fri) by 11.59 pm

Percentage of total grade: 10%

Objective

1. To perform auto white balance (i.e., illumination estimation).
2. To perform interactive white balance.

Provided to you.

1. Four DNG files (all encoded using GBRB Bayer).
2. Video of the assignment

How the program should work.

1. Open up the DNG file.
2. Use a simple demosaic approach that down-samples the image by a factor of 2.
 - The resulting image is $\frac{1}{2}$ the Bayer image size in the x and y direction.
 - Use corresponding R, G, and B Bayer values (no need for interpolation).
 - In my case, I used $(G1+G2)/2$ as the G channel.
3. Perform AWB to compute the (R, G, B) illumination vector.
 - You can use any algorithm you like based on what we discussed in the lecture.
4. Apply the diagonal white-balance correction (“incorrect von Kries”) to the RAW image.
5. Change the first 100x100 pixels to the estimated value (R, G, B).
6. Change the last 100x100 pixels to be ‘red.’
7. Show the image normalized with 2.2 gamma applied (same as assign4).

Infinite Loop (Interactive)

8. If the user clicks on the image, change the illumination vector to RGB value in the original RAW image at the clicked location.

-> I used the following code to get clicks.

```
while True:
    click_event = plt.ginput(1, timeout=0) # Poll plot to check for click
    if not click_event:
        break                               # Returns to top of the loop
    . . .
```

8. If the user clicks in the top 100x100, reset to the AWB value.
 9. If the user clicks in the bottom 100x100 (red area), apply no WB correction.
- > This will allow you to see the RAW image with the color cast.

Example

(A) Starting window (result shown in first 100x100 image patch is AWB result).

Click on a pixel to get its RGB value



See next page . . .

(B) Result after clicking on the image (notice how 100x100 changed).



(C) Clicking the “red” region shows the original RAW with no white balance applied.



(D) If the user clicks in the top 100x100 image patch, show the AWB result again (same as the starting image).



What to submit via e-class:

Two files

1. Your Python code (zip).
2. Video of your program running w/o audio (mp4).

Comments:

1. The assignment must be done in Python.
2. You are welcome to use ChatGPT.

3. Useful Python packages/functions:

numpy

rawpy

matplotlib