CMSC498D Project 2 Report

Jacob Lin and William Lin

Introduction

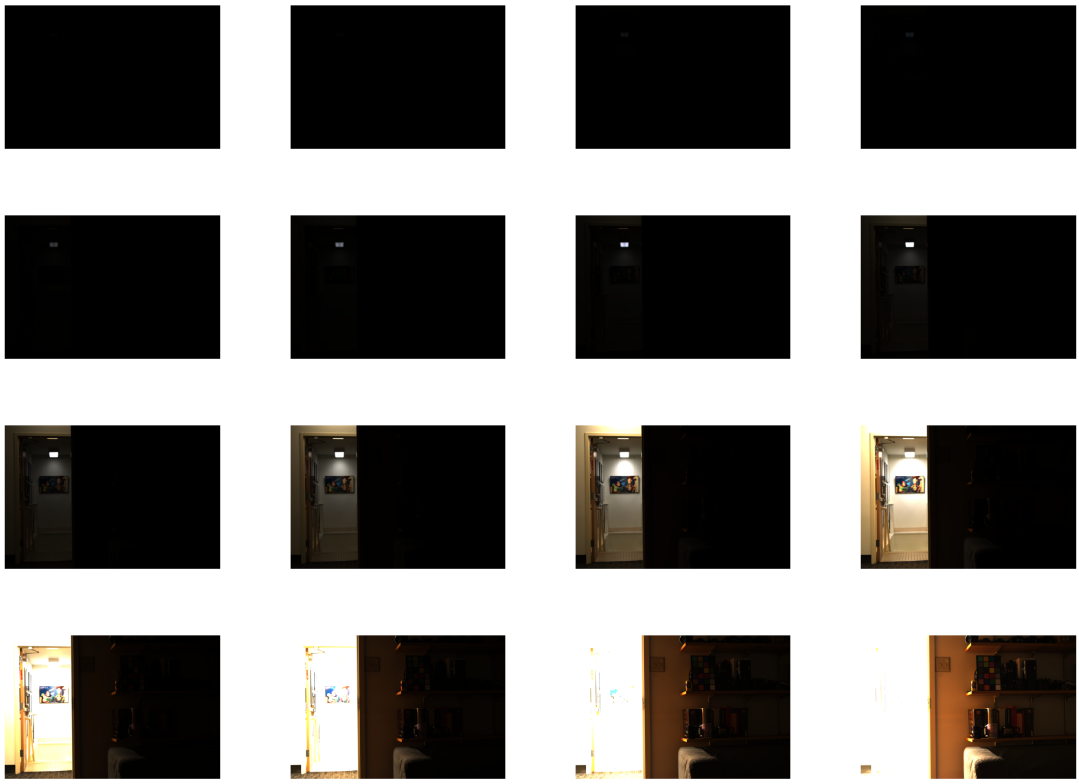
In this project our goal is to combine low-dynamic-range of images captured with different exposure and combine them into a single high-dynamic-range (HDR) image with some tone-mapping, gamma correction, lossy compression as well.

Processing the Raw Images

The first step is processing our raw images by deomosaicing and white balancing each raw image. After consulting the dcraw documentations, we noticed that using the `-i -v` options prints the metadata for a *.nef* file and helped us locate the camera multipliers used to convert *.nef* files into *.tiff* files. We used the following dcraw commands to achieve this:

```
> dcraw -i -v ./exposure13.nef
Filename: ./exposure13.nef
Timestamp: Wed Oct 11 19:33:04 2017
Camera: Nikon D3300
ISO speed: 100
Shutter: 2.0 sec
Aperture: f/4.8
Focal length: 40.0 mm
Embedded ICC profile: no
Number of raw images: 1
Thumb size:  6000 x 4000
Full size:   6016 x 4016
Image size:  6016 x 4016
Output size: 6016 x 4016
Raw colors: 3
Filter pattern: RG/GB
Daylight multipliers: 2.231936 0.932648 1.141543
Camera multipliers: 1.585938 1.000000 2.054688 0.000000
> dcraw -T -4 -r 1.585938 1.0 1.0 2.054688 ./*.nef
```

After opening the exposure files and using exifread the extract tags we will be using, we ended up with the 13 photos below. Notice that longer exposure times lead to brighter images.

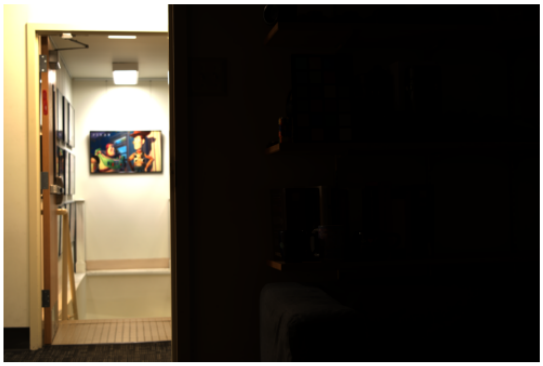


Merge the LDR images into an HDR image

Next, combined these LDR images using the provided equation. At first, we struggled to get the combination in an efficient manner. However, after consulting the class Piazza and using numpy, we were able to combine in a timely manner. At this point, the images are rather dark. And using `pyplot`, we are able to see what the images initially look like.

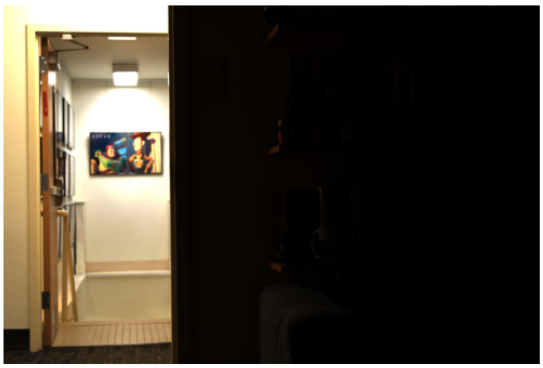
Using the uniform weight function we have the following:

$$w_{\text{uniform}}(z) = \begin{cases} 1 & \text{if } Z_{min} \leq z \leq Z_{max} \\ 0 & \text{otherwise} \end{cases}$$



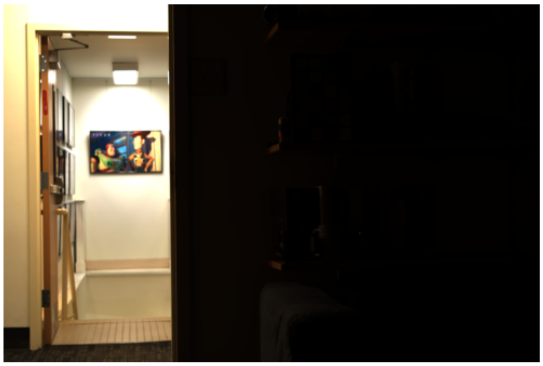
The next weight function is

$$w_{\text{tent}}(z) = \begin{cases} \min(z, 1-z) & \text{if } Z_{min} \leq z \leq Z_{max} \\ 0 & \text{otherwise} \end{cases}$$



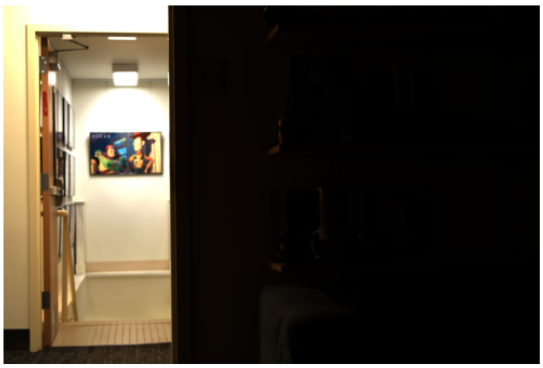
The next weight function is

$$w_{\text{gaussian}}(z) = \begin{cases} e^{-4\frac{(z-0.5)^2}{0.5^2}} & \text{if } Z_{min} \leq z \leq Z_{max} \\ 0 & \text{otherwise} \end{cases}$$



The last weight function is

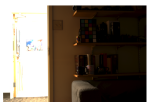
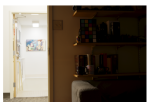
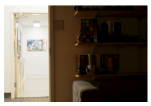
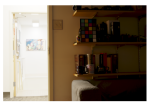
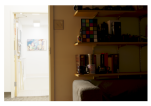
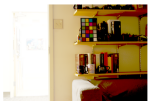
$$w_{\text{photon}}(z, t^k) = \begin{cases} t^k & \text{if } Z_{min} \leq z \leq Z_{max} \\ 0 & \text{otherwise} \end{cases}$$



Tone-map the HDR image into an LDR image

In this step, we had some troubles with writing the function and errors. However, it was actually fairly straightforward given that the provided formulas are followed. After a successful implementation, the following images using w_{gaussian} and its respective K and B values are provided below. We noticed that increasing generally made the whole images brighter and increasing B generally

preserved more highlight contrast. Decreasing K would in turn darken the image and decreasing B compresses the highlights.

$K \backslash B$	**1e-5**	**1e-2**	**0.1**	**0.95**	**3**
0.02					
0.075					
0.15					
0.5					
1.0					

We decided to go with $K = 0.02$ and $B = 3$ since the K provided a dark image that will later be brightened with the Gamma-correction and so we want to account for that beforehand. B didn't provide much of a difference for values greater than 0.95, so this choice wasn't motivated by much except that we can't go too low.

Gamma-Correct the LDR image for display

There wasn't much trouble to this part as we implemented a Gamma-Correction using the provided function.

$$C = \begin{cases} 12.92C_{\text{linear}} & C_{\text{linear}} \leq 0.0031308 \\ (1+0.055)C_{\text{linear}}^{1/2.55} - 0.055 & C_{\text{linear}} > 0.0031308 \end{cases}$$

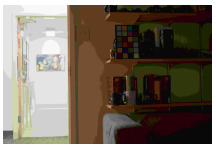
The resulting images are the following:

Uniform Weight **Tent Weight** **Gaussian Weight** **Photon Weight**

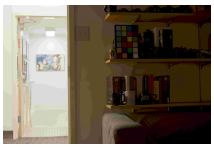
Lossy Compression

In this section, `cv2.imwrite` is used to compress the tone-mapped and gamma-corrected uniform ldr image. We looked at documentation for *OpenCV* to understand how to compress the image into *.jpg* files. The compressed images along with its quality and size are included below. We noticed that as the quality reached below about 40, the pixels on the walls started to distort. So a quality of 40 is the lowest compression quality setting in which the compressed image is indistinguishable from the original. Thus the compression ratio is

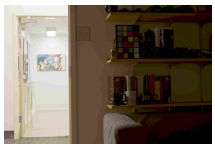
`original size / compressed size` $= 16290530/572013 = 28.47930029562265$



Q=5, Size=387KB



Q=10, Size=410KB

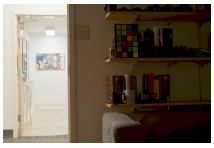


Q=15, Size=432KB

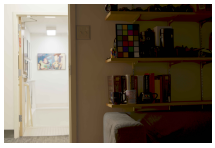
Q=20, Size=458KB

Q=25, Size=484KB

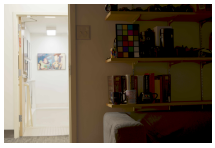
Q=30, Size=509KB



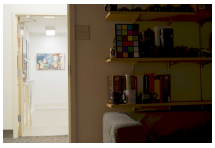
Q=35, Size=537KB



Q=40, Size=558KB



Q=45, Size=586KB



Q=50, Size=611KB

Q=55, Size=637KB

Q=60, Size=670KB

Q=65, Size=715KB

Q=70, Size=769KB

Q=75, Size=838KB

Q=80, Size=956KB

Q=85, Size=1130KB

Q=90, Size=1508KB

Q=95, Size=2626KB

Q=100, Size=8333KB