Digital Visual Effects – HDR Imaging

Group 16: B11902078 張宸瑋, B10902058 胡桓碩

Summary

Using Python, we carried out **Paul Debevec's method** to assemble our HDR images. For bonus, we implemented **MTB Algorithm** for alignment, as for tonemapping, we implemented and tested with **Gamma Mapping**, **Photographic Global Mapping**, and **Tonemapping with Bilateral Filtering** as suggested. To refine our approaches and parameters for image processing, we captured 10 different sets of pictures. From these sets, we selected several sets of photos that we deemed satisfactory during our testing process. These sets were included in the homework file for evaluation and reference purposes.
Further detail and elaborations are provided below.

Photography and Exposure Handling

We utilize Sony's XQ-CT72's convenient exposure-adjusting feature for photography. This device automatically assigns values such as shutter speeds and aperture sizes, and then calculates these properties into the Exposure Value (EV). As a result, we can effortlessly record the exposure value of each original photo as a power of 2 for further operations. For instance, if a photo has an EV of +2, it implies that its exposure value can be set as $2^2 = 4$.

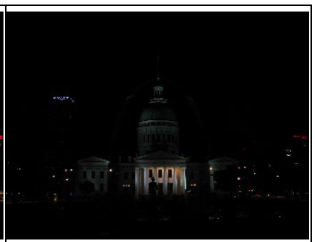
```
# LDR filenames       exposure
20240401_185439.JPG   3.482202
20240401_185443.JPG   2.828427
20240401_185447.JPG   2.462288
20240401_185452.JPG   1.741101
20240401_185455.JPG   1.414213
20240401_185459.JPG   1.148698
20240401_185503.JPG   0.812252
20240401_185508.JPG   0.659753
20240401_185511.JPG   0.574349
20240401_185515.JPG   0.406126
20240401_185518.JPG   0.378929
20240401_185521.JPG   0.307786
20240401_185523.JPG   0.25
```

This is some text in "data/library3/origin/image_list.txt," which documents the operation of our code on the original photos. We sorted the pictures by their exposures and recorded the corresponding values. Additional information regarding the execution of the code can be found in the "README.md."

Alignment

First, we use "align.py," which has the MTB algorithm within to align the original images.
We found that there will be different effects when choosing mean or median for threshold of bitmap conversion. Theoretically, it's better to choose median as the threshold. But the table below is an example we discovered, that shows, in the specific photo set, choosing mean for threshold performs better than choosing median.
This difference is likely because when using MTB in photo pairs with large exposure gaps, using median is not robust enough to generate fine result. Thus, to avoid the problem, we tend to take more photos in a set so that we can narrow the exposure gap. We also experimented on the two methods for best outcomes.
We also ignored the pixels with values close to threshold, which displayed as gray color in the bitmaps. And we found that the lower exposure the photo has, the larger the gray area in the generated bitmap will be.

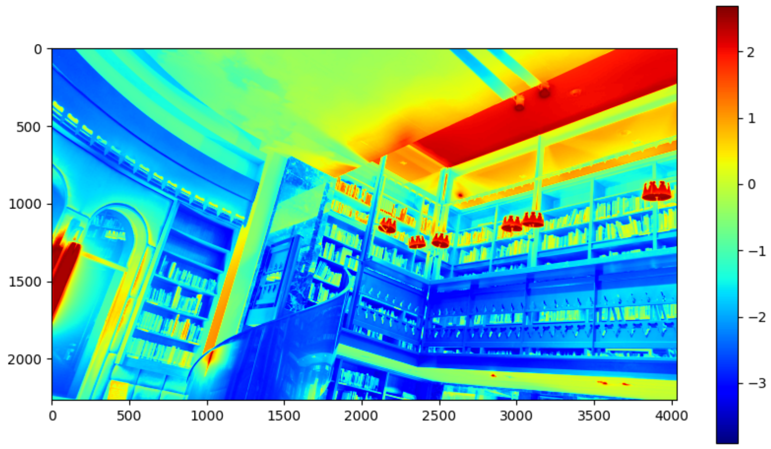
Bitmap Sets with Different Threshold

Original Photos				
Mean approach				
Median approach				

Generating HDR Image

Below is our chosen artifact from the photo set "data/library3." We have plotted its radiance map and its response curve.

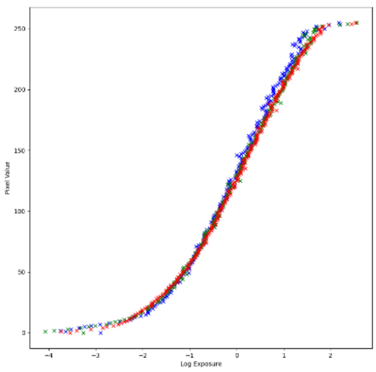
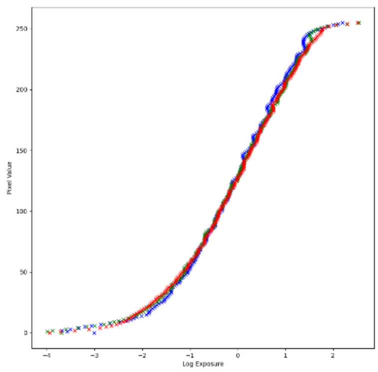
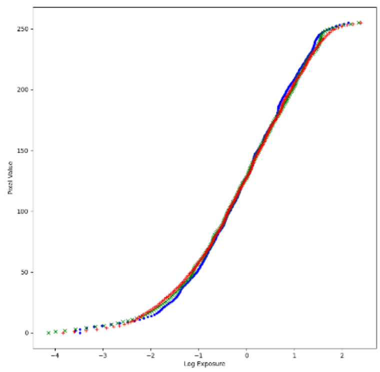
Radiance Map of the HDR File



Tonemapping Result of the HDR File

In addition, the λ value used in HDR generation also affects the color scheme of the image, though it's difficult to concisely describe its changes, by comparing the plotted response curve generated, we can see that the value affects the smoothness of the curves.

Response Curves with Different λ Values

$\lambda = 1$	$\lambda = 5$	$\lambda = 20$
		

Normalizing

After tonemapping, the level of pixels may still have values slightly larger than 1, so in order to transform the image into LDR format, we need to normalize the levels that they would all be smaller than or equals to 1. We use two ways to achieve the goal, the first way is directly normalizing the result image linearly, and the second way, which is used in Tonemapping with Bilateral Filtering, is making the result undergo another global mapping first, then normalize the result linearly.

Furthermore, we discovered that another two different ways of normalizing will have different effects. First is normalizing each of the color channels (namely, RGB) individually, and the second way is normalizing all channels simultaneously. When using the former, the output result will often become "green-er", we guess that this phenomenon may have a resemblance to the "Bayer's Pattern" in demosaicking mentioned in class, as said pattern is mostly composed of green pixels.

The Difference between the normalizing methods in Gamma Mapping

Normalize All Channels Simultaneously	Normalize Each Channel Individually
	

About Gamma Mapping

When implementing Gamma Mapping, we discovered two different algorithms, which leads to distinct effects.

The first approach, we called it “Gamma Intensity”, is applying Gamma correction $x \to x^{1/\gamma}$ solely to the luminance first, followed by the restore of color data. This approach exhibits high color saturation when the γ value is high. However, at lower γ values, the result might be high contrast and very dark.

Tonemapping Results with Different γ Values in “Gamma Intensity”

$\gamma = 1$	$\gamma = 1.5$	$\gamma = 2.2$
		

The second approach, we called it “Gamma Color”, directly applies Gamma correction to each color channel. As a result, the outcomes are about opposite to the first approach. At higher Gamma values, the overall image tends to have lower saturation, while at lower Gamma values, it has higher saturation.

Tonemapping Results with Different γ Values in “Gamma Color”

$\gamma = 1$	$\gamma = 1.5$	$\gamma = 2.2$
		

About Global Mapping

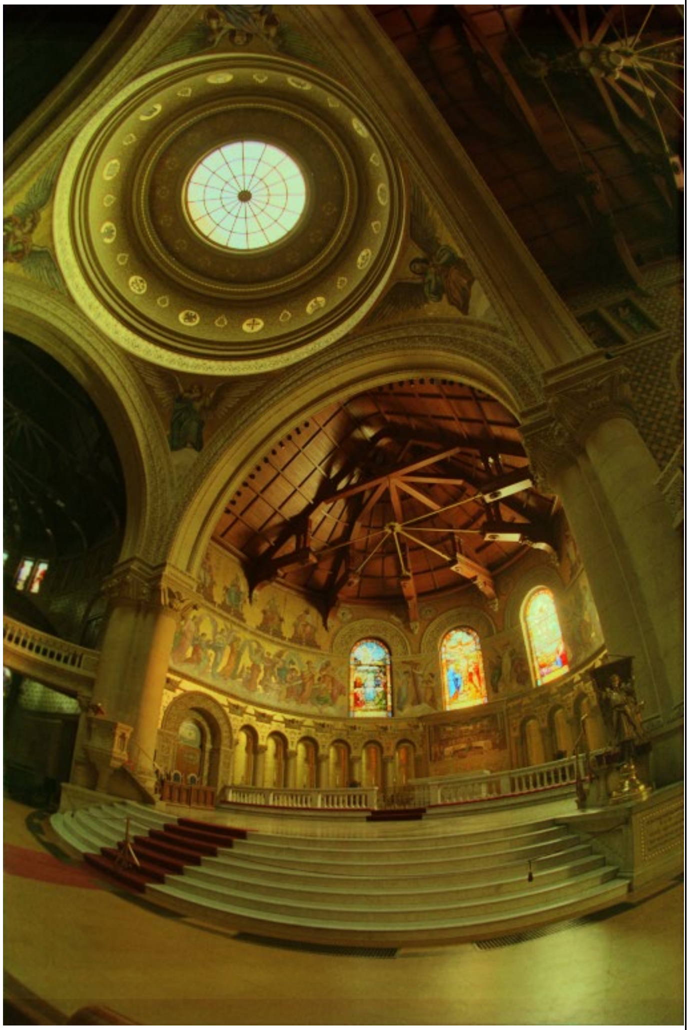
The Global Mapping Algorithm is used as a whole itself, and is also needed in the bilateral filtering as we mentioned above when talking about normalizing.

We discovered another point that worth noticing is the value of delta, which is the δ in the following equation (also mentioned in README.md),

$$\overline{L_w} = \exp\left(\frac{1}{N}\sum_{x,y} log(\delta + L_w(x,y))\right)$$

would affect the output images. We picked two examples to compare below.

Global Mapping Results with Different δ Values

$\delta = 1$	$\delta = 10^{-6}$
	

We worked on and experimented with some parameters in this project, their descriptions are noted in the "README.md."

About our Artifact

We finally chose the generated tonemapping result of the "library3" photo sets as the "result.png" artifact. Here's an overview of our process in creating this image.
Initially, we experimented with Gamma Mapping, Global Mapping, and Tonemapping with Bilateral Filtering as well as the high and low frequency images generated from the latter. We also compared the results with those generated by using Drago Tonemapping from OpenCV. Eventually, we found that the result of Bilateral Mapping looked the best for this particular photo set.
Subsequently, we began adjusting the parameters for Tonemapping with Bilateral Filtering.
At the End, the parameters for the finished artifact are (1, 3.5, 13, 40, 5), which can also be found within the "data/library3/hdr/tonemap.txt."

Appendix

The folders labelled "test" holds the pictures that we collected from the internet prior to our photo collecting, which have helped us with establishing the project's structure and how to set certain parameters for desired

results.
Last but not least, we had fun when collecting photos and ventured into places that we didn't know existed in NTU campus. And by experimenting with the pictures, we discovered the potency of image-processing that for example, by simply changing a picture's tone, can have a result whose characteristic is vastly different from its original. Overall, this is a challenging yet rewarding project, and thanks to the course's team that provided guidance during the project.