COL 783

Assignment 1

Rajbir Malik
2017CS10416

August 28, 2019

Tone Mapping HDR Images

Overview

In this assignment, we were asked to understand and try various tone mapping methods/algorithms on `HDR` images. We progressively managed to get a better map by first trying fixing the **linear scale**, then **logarithmic scale** and eventually an algorithm in general use. I chose to implement *Reinhard et al., "Photographic Tone Reproduction for Digital Images"*. The assignment was divided in 3 different sections, each of which are discussed separately, later.

Linear and Logarithmic Rescaling

Linear scaling was used to map the complete range (Luminance) of the HDR image between different (min - max) values. Following were the results produced with different scales,

- *Scaling, with large range* : Pixels with large values lose value in the image.
- *Scaling, with small range* : Pixels with small values lose value in the image.
- *Scaling, with range in between* : Somewhat decent, still this time many of lower and higher range pixels lose value in the image.

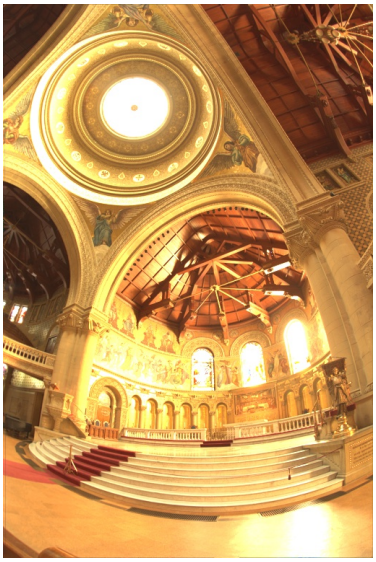


Figure 1: Scaling, with large range

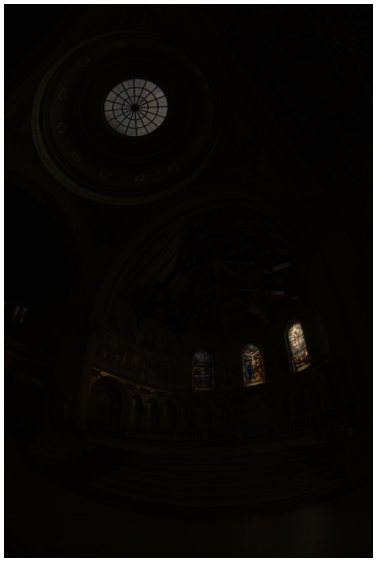


Figure 2: Scaling, with small range

Figure 3: Scaling, with range in between

Logarithmic scaling was implemented by processing the `log(luminance)`. It had much better results than the linear scaling as it lead to non-linear map which allowed accommodation of closer pixels. I tried in different bases, and the results are as followed,

- `Base:10` with scaling `0.1-2.1` (and thus linear scale is 1:100)
- `Base:3` with scaling `1-6` (and thus linear scale is 1:243)

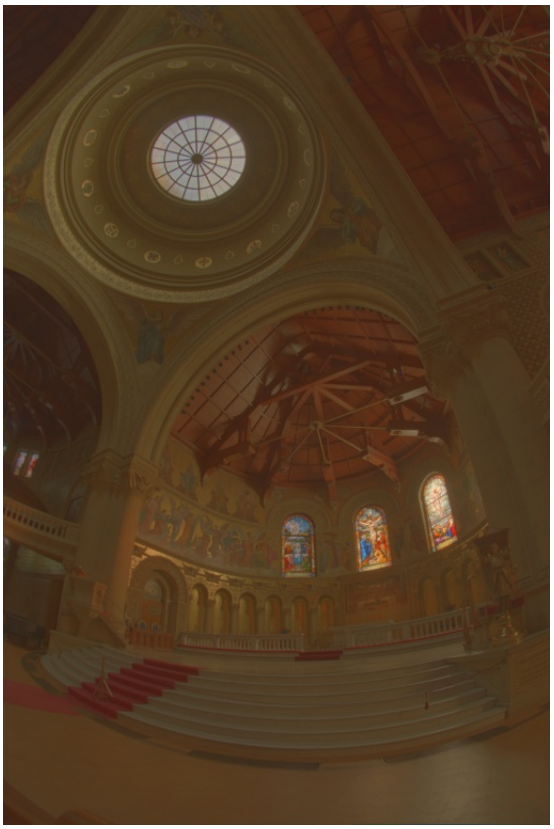


Figure 4: Base 10

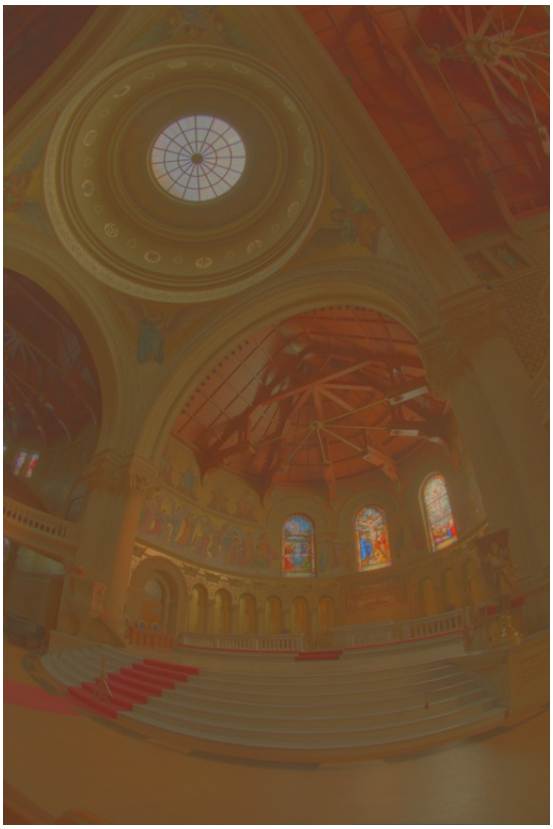


Figure 5: Base 3

Concluding, the images produced by `log-luminance` rescaling are much better and comprehensive. We shall now pursue improving these images in the next section.

Detail Enhancement

In this section we apply series of image enhancement techniques to improve the features of the image previously acquired. First the image was enhanced in the linear-luminance domain. Following steps were carried out,

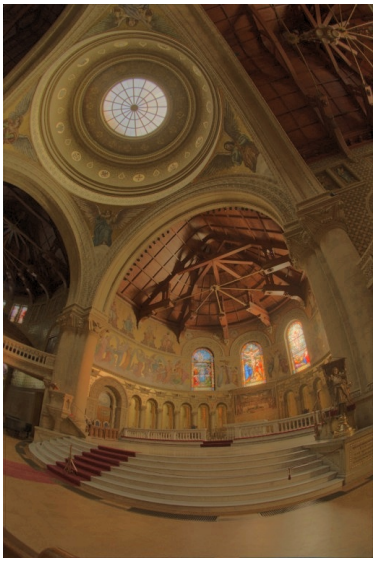


Figure 6: Histogram Eq.

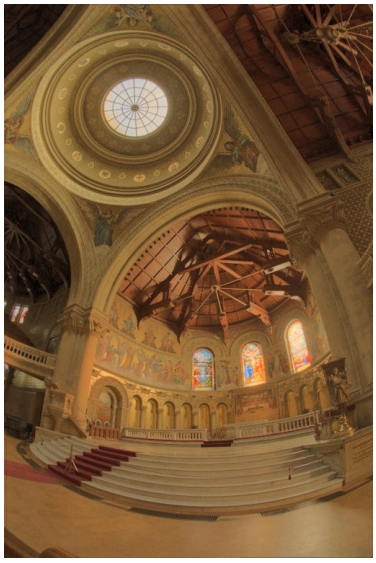


Figure 7: Gamma Correction

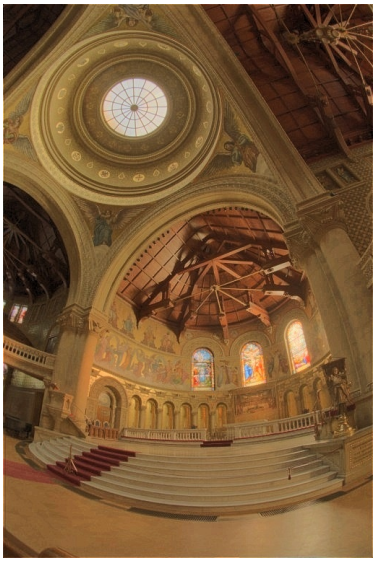


Figure 8: Unsharp Masking

Second the image was enhanced in the log-luminance domain. Following steps were carried out,

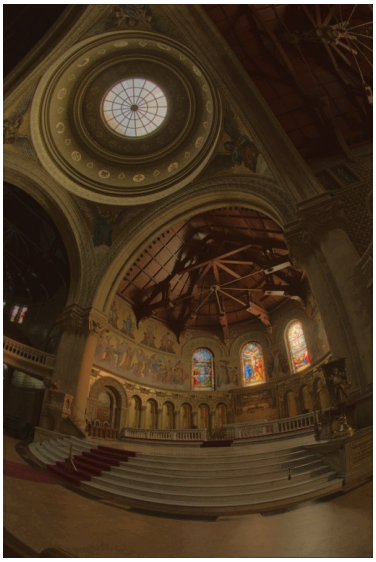


Figure 9: Histogram Eq.

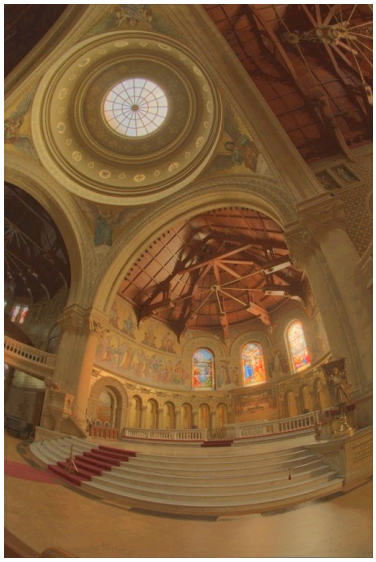


Figure 10: Gamma Correction

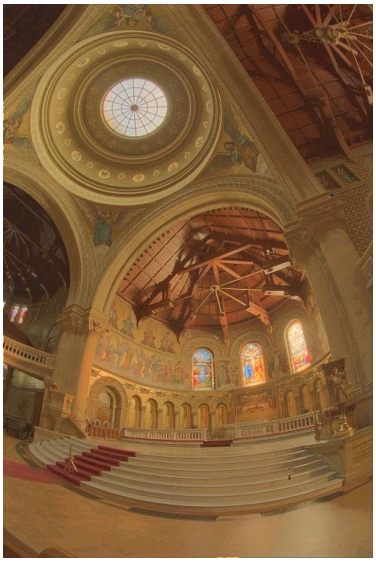
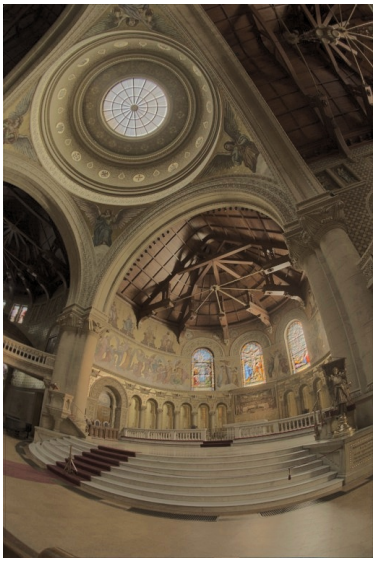


Figure 11: Unsharp Masking

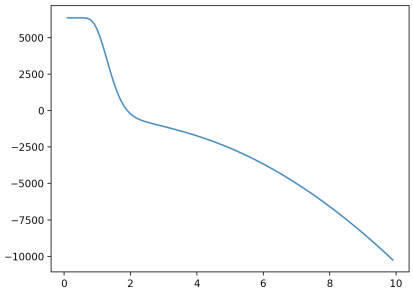
Concluding, there is much difference b/w the final images produced except for that log-domain maintains chromaticity better. Also, directly working with RGB values is not logical as there correlation to produce a color is non-zero, and thus any unsynced changes would lead to loss of the original color.

Tone Mapping Algorithm

In this section, I focused on implementing the **Reinhard Tone Map** and applied on a set of images. The original images produced, were processed to give promising mappings, shown below.



Also, following was the general variation in the deciding s parameter.



Also, following are how images vary if different parameters are set.

Figure 12: s = 0.1

Figure 13: s = 2.1 (Optimal)

Figure 14: s = 9.9

Figure 15: key = .36

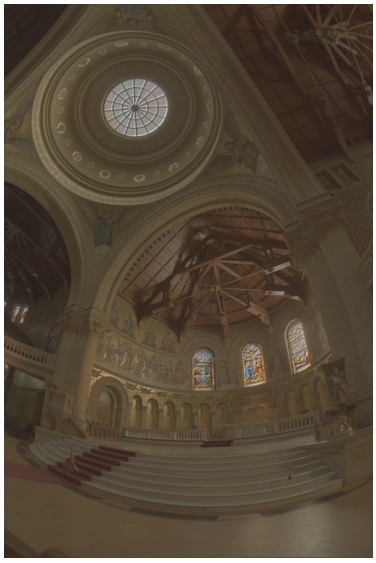


Figure 16: key = .45

Figure 17: key = .72

The results from the tonemapping algorithm are distinctively better than the methods implemented before. Seemingly intelligent, the tone-map provides gives a more realistic idea of how the original image may have looked.

Summary

All required by the assignment was implemented and a lot of new things were learned/understood. The codes are written in Python (for part 1 & 3) and in C++ (for part 2). All the functions were implemented on own (except for the FFT, which was allowed to be used).
I am also looking forward to implementing other options from the assignment (which unfortunately due to internship acquiring process I couldn't devote time to).
The assignment was intriguing and I learnt a lot of new ideas. Thanks and Regards.