CS 484 Final Report

Hanzhen Yang, Yue Zhu, Miao He

1. Introduction

1.1. What is High-dynamic range imaging

High-dynamic range imaging(HDRI) is a method of reproducing greater dynamic range of luminosity than what is possible with standard digital imaging. Because the natural world contains a range of radiance values that is far greater than can be captured with any photographic sensor or film, any single exposure of a picture would contains saturated (overexposed) and dark (underexposed) regions. The aim of HDRI is to help create well-exposed photographs under challenging conditions such as extremely bright scenes. The common way of producing HDR image is to use pictures that was taken under a set of bracketed exposures, applying some transformation algorithm such as Debevec-Malik method to get the result HDR-image.

1.2 Brief intro to Debevec-Malik method

The Debevec-Malik method is one of the commonly used algorithm to implement HDR image. Following is a brief introduction to Debevec-Malik method.

Assume that we have three set of images at different exposures

Step1: *Estimate the radiometric response function from the aligned images:*

Measured pixel value z_{ij} , exposure time t_j for the j th image,

Irradiance; values E_i ,radiometric response function f

$$z_{ij} = f(E_i t_j) \qquad (2.1.1)$$

$$f^{-1}(z_{ij}) = E_i t_j \qquad (2.1.2)$$

Take logarithms of both sides we have

$$g(z_{ij}) = log f^{-1}(z_{ij}) = logE_i + logt_j \qquad (2.1.3)$$

Step 2: *Estimate a radiance map by selecting r blending pixels from different exposures*

Corresponding pixel values are plotted as function of log exposures(irradiance), shift the curves for each pixel's unknown Radiance until they all line up into a single smooth curve.

Second-order smoothness constraint by Debevec and Malik(1997):

$$\lambda \Sigma_k g''(k)^2 = \lambda \Sigma [g(k-1) - 2g(k) + g(k+1)^2] \qquad (2.1.4)$$

Putting all the terms together, obtain a least squares problem in the unknowns $\{g_k\}$ and $\{E_i\}$

$$E = \Sigma_i \Sigma_j w(z_{i,j})[g(z_{i,j}) - logE_i - log\ t_j]^2 + \lambda \Sigma_k w(k)\, g''(k)^2 \qquad (2.1.5)$$

Step 3: *Tone map the result HDR image back into a displayable gamut*

2. Result

2.1 Test 1

Figure 1: test 1 input

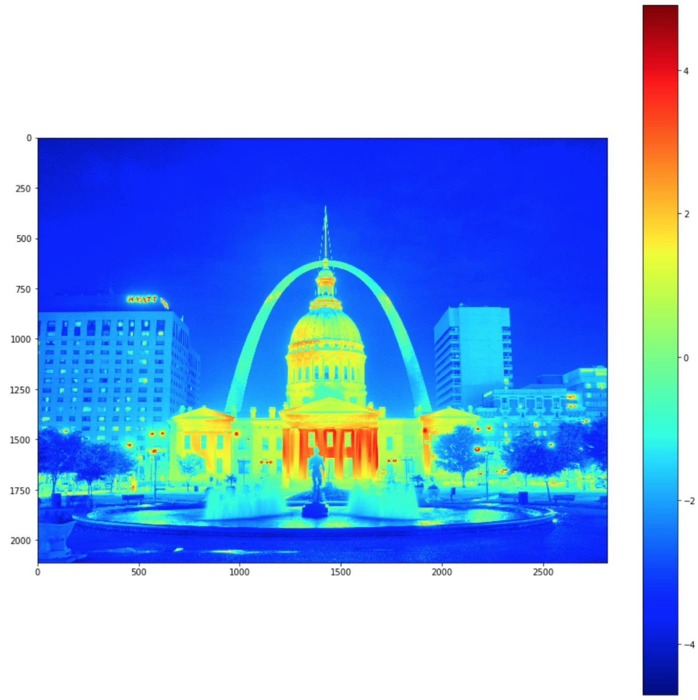

Figure 2: Test 1 Radiance map

Figure 3: Test 1 Result (Sigmoid tone mapping)

2.2 Test 2

Figure 4: Test 2 Input

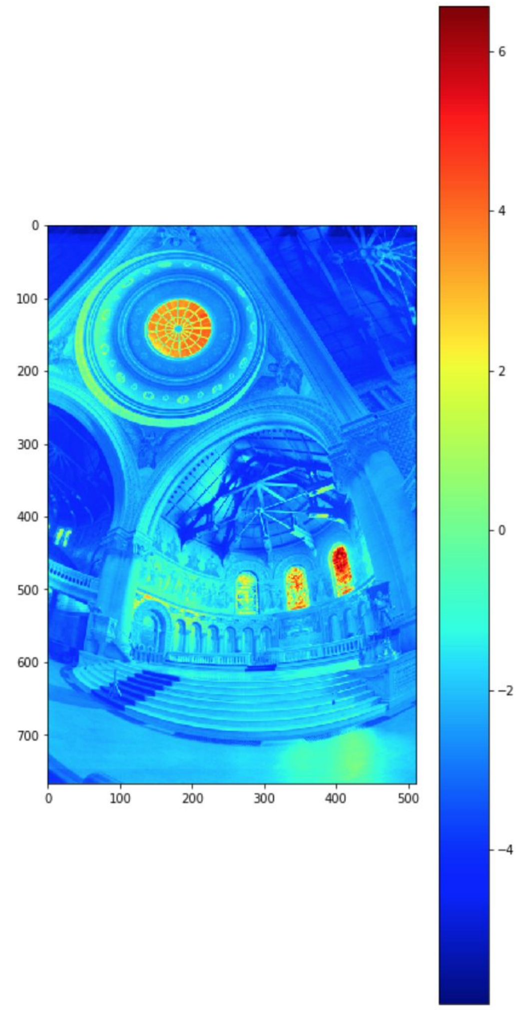

Figure 5: Test 2 Radiance map

Figure 6: Test 2 Result (Sigmoid)

2.3 Test 3

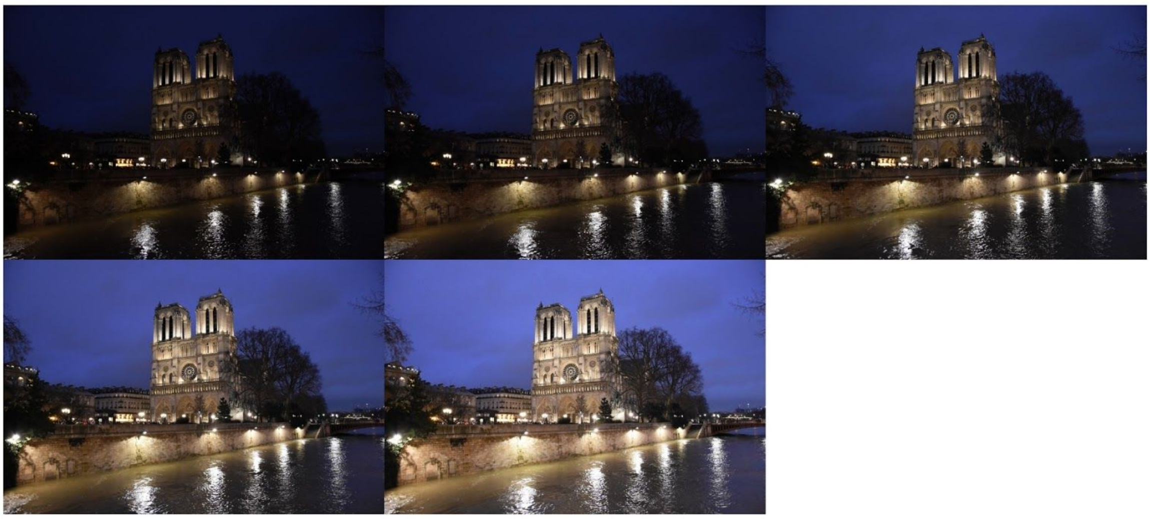

Figure 7: Input

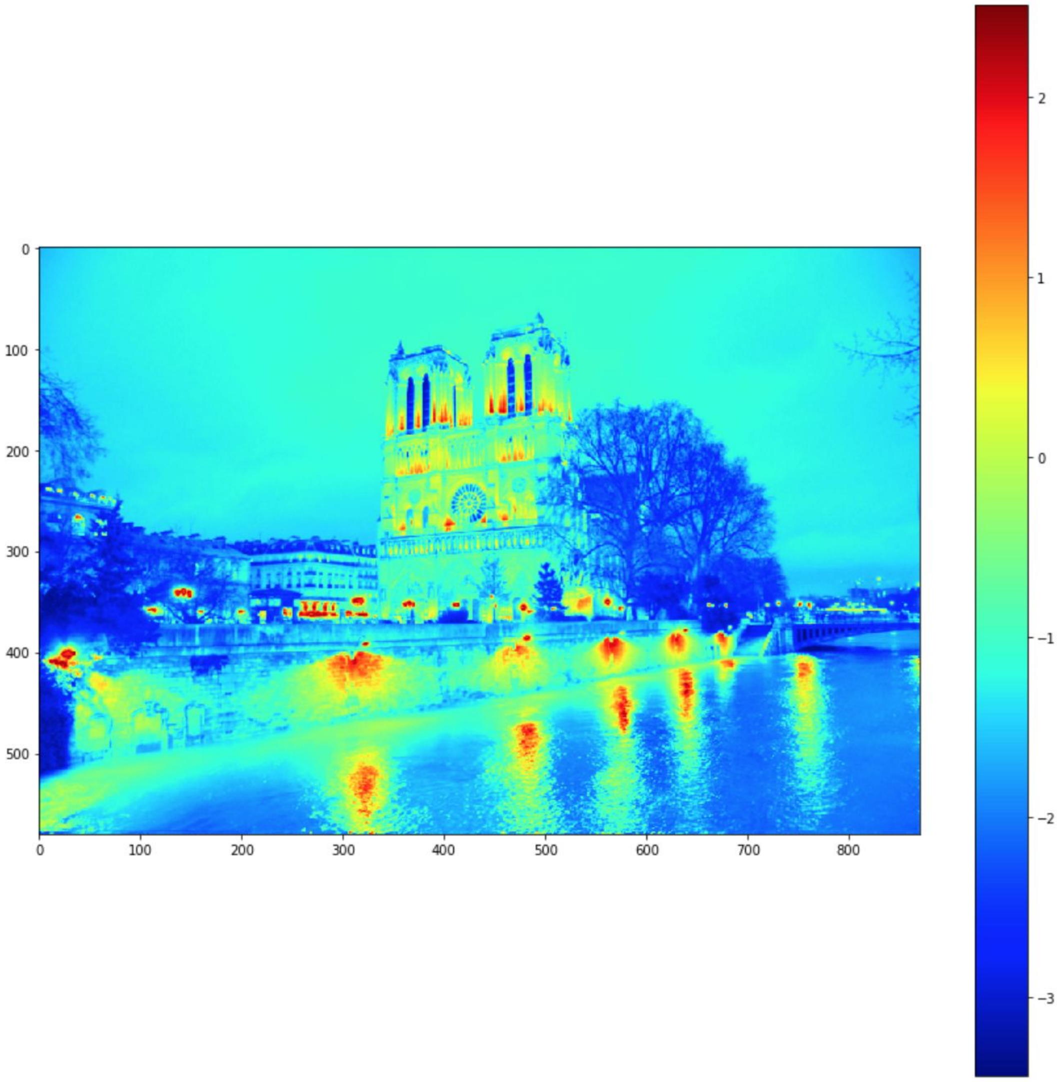

Figure 8: Test 3 Radiance map

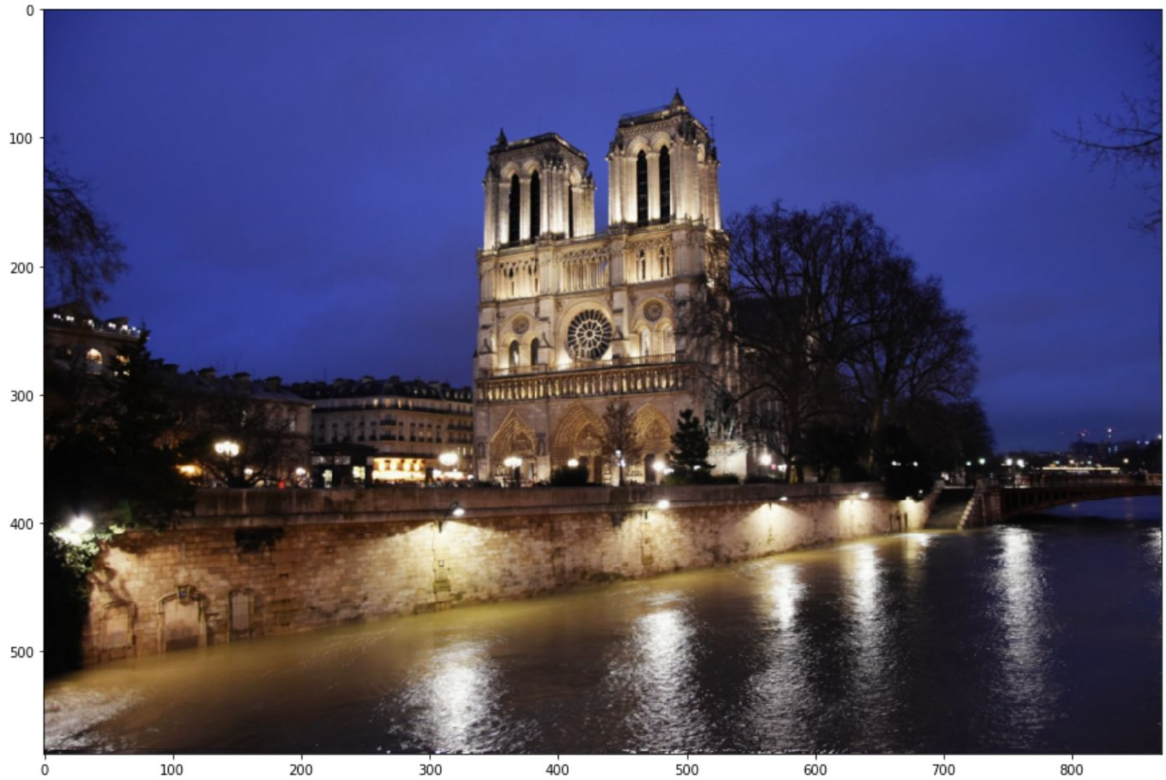

Figure 9: Test 3 Result (Sigmoid)

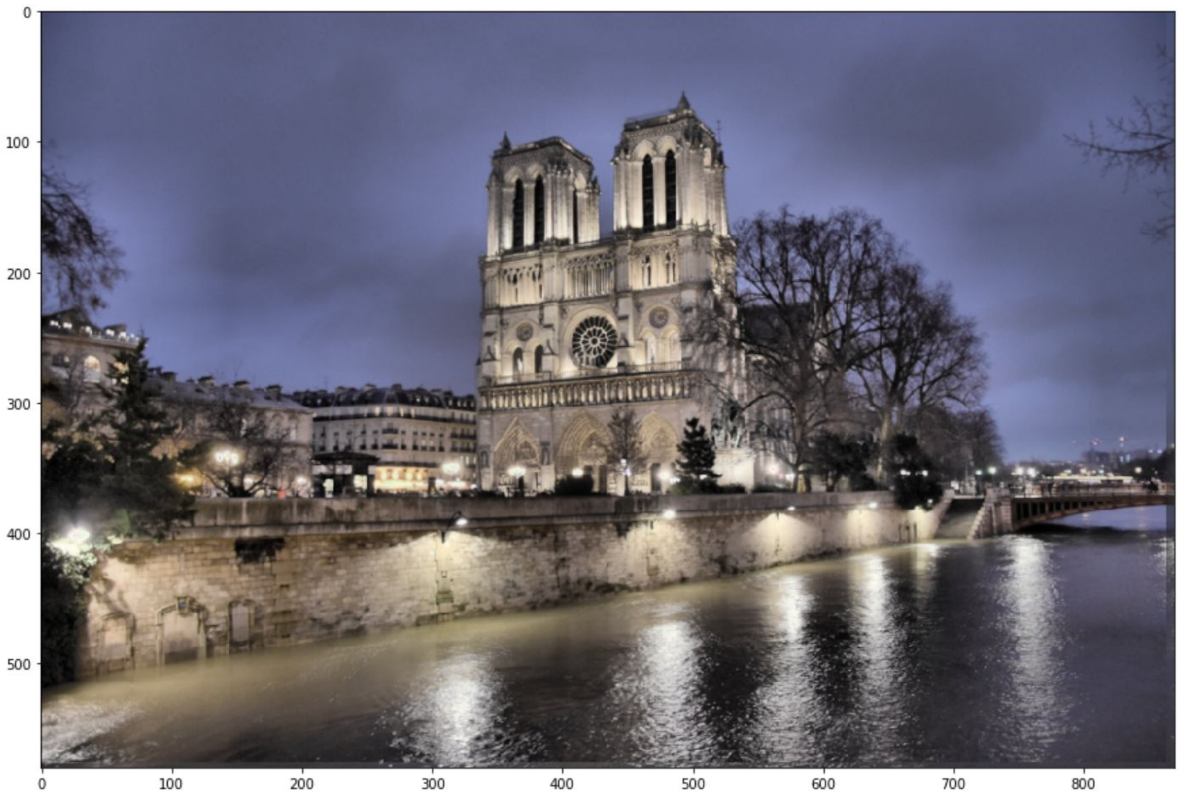

Figure 10: Test 3 Result (Equalized-adapthist)

2.4 Test 4

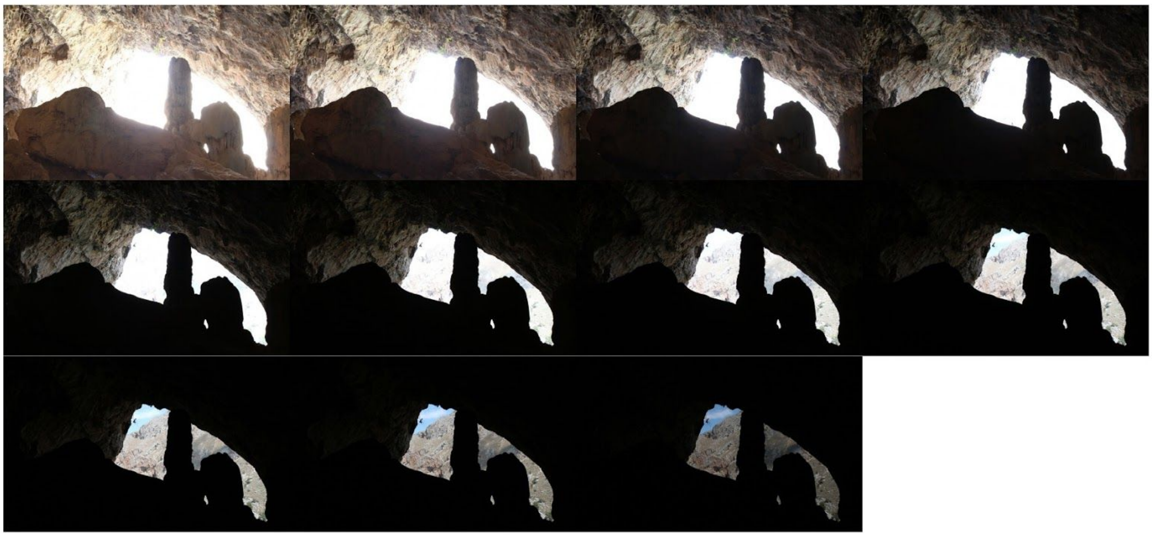

Figure 11: Input

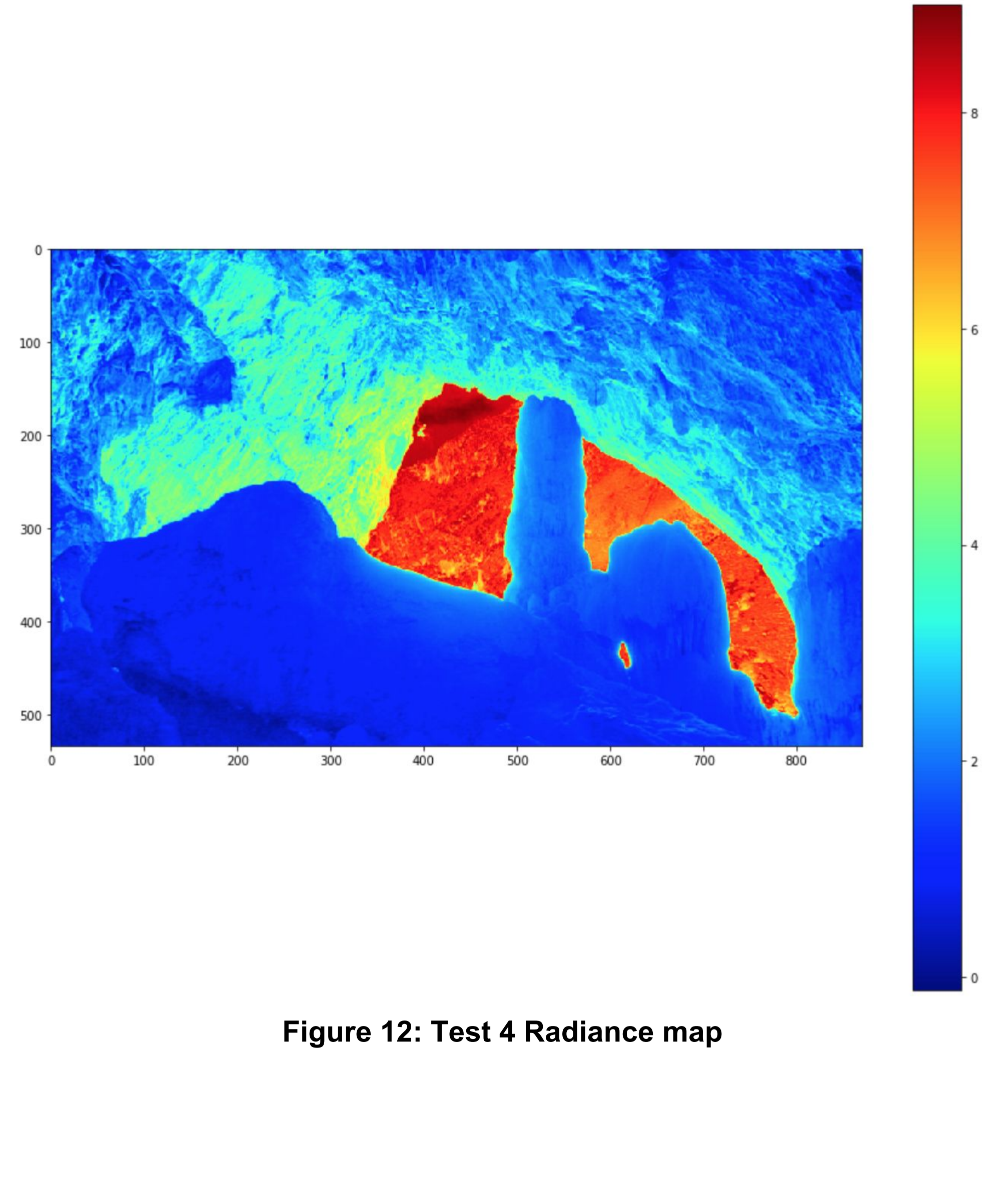

Figure 12: Test 4 Radiance map

Figure 13: Test 4 Result (Equalized-adapthist)

3. Discussion

3.1 With or without tripod: Image alignment

At first, we assume that photo that was taken with tripod might not need a step of alignment, comparing to the photo that was taken without tripod. However, after doing a little bit of research, we realized that , with or without tripod, there shall both be a step of image alignment. Because there is no way to guarantee a vibration-free environment such that the image can be perfectly aligned.

As indicated by the red circle in Figure 14 and Figure 15, if there is no image alignment, even the images were taken with tripod, the produced grayscale image (Figure 14) and radiance map (Figure 15) will present severe ghost artifacts.

In our implementation, we used homography-based registration of the sequence; similar to what we did in assignment 1 , to align our images. The produced result can be seen in Figure 16 and Figure 17. Notice that the area that was red circled in Figure 14 and Figure 15 now is sharpened and there is no sign of ghost artifacts.

Figure 14: Gray-scale HDR without image alignment

Figure 15: Radiance map without image alignment

In our Section 2: Result, test 3 is using images that were taken handheld, turns out our homographic image alignment can be extend to images taken without tripod.

Figure 16: Gray-scale HDR after image alignment

Figure 17: Radiance map after image alignment

3.2 Drop the assumption of Exposure Times

According to Cerman and Hlav´aˇc(2006), the camera reported exposure time or in the EXIF data are usually not the exact exposure time. The camera tend to round the exposure time to conform with photographic tradition in marking exposure times. Some of the camera may have significant variance between the reported exposure time and the exact exposure time. Therefore, knowing the real exposure time can positively affect the radiometric quality of the HDR image, but the knowledge is not essentially necessary.

As introduced by Cerman and Hlav´aˇc(2006), for a normalized produced HDR image, there is no need to know the exact exposure time to have brightness values ranging from 0 to 1. As long as the relative exposure against the darkest image is estimated, which is accomplished by setting the exposure time of the darkest image to be the unit exposure time, and then compute the exposure ratio between the darkest image and the other images, as indicated in the following formula:

Assume darkest image exposure time $\Delta t_1 = 1$, the exposure ratio k_j can be computed using formula: $\Delta t_j = k_{j-1}\Delta t_{j-1}$

Assume The calculation is shown below:

Assume for same point in the picture, E_i keeps the same in different pictures, then we can assume that for the same point in the different picture, there is a linear relationship between exposure time and the pixel value:

$$t_j = Z_{ij}a_i + b_i$$

where t_j is the exposure time for j^{th} picture, and Z_{ij} is the pixel value for pixel at position i in the j^{th} picture.
Notation: Let M[i] be the i^{th} row of M and M_j be the j^{th} column of M
Consider construct the matrix M $\in R_{i,j+1}$ by the following rule:

$$M_{pq} = \begin{cases} t_q & 1 \leq q \leq j \\ -b_p & q = j+1 \end{cases}$$

Construct the matrix X $\in R_{j+1,j}$ by the rule:

$$X_{pq} = \begin{cases} 1 & p = j+1 \ or \ p = q \\ 0 & otherwise \end{cases}$$

Construct the diagonal matrix A $\in R_{i,i}$, where diag $= (a_1, a_2, ..., a_i)$
e.g. for i = 3, j = 2

$$M = \begin{bmatrix} t_1 & t_2 & -b_1 \\ t_1 & t_2 & -b_2 \\ t_1 & t_2 & -b3 \end{bmatrix}; X = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix}; A = \begin{bmatrix} a_1 & 0 & 0 \\ 0 & a_2 & 0 \\ 0 & 0 & a_3 \end{bmatrix}; Z = \begin{bmatrix} Z_{11} & Z_{12} \\ Z_{21} & Z_{22} \\ Z_{31} & Z_{32} \end{bmatrix}$$

And we can get that MX = AZ and A^{-1} is also a diagonal matrix with diag$=(\frac{1}{a_1}, \frac{1}{a_2}, ..., \frac{1}{a_i})$
Let K = $A^{-1}M$ and then KX = Z, we can use least square to solve for K and K $\in R_{i,j+1}$ with

$$K_{pq} = \begin{cases} \frac{t_q}{a_p} & 1 \leq q \leq j \\ -\frac{b_p}{a_p} & q = j+1 \end{cases}$$

e.g. for the same example

$$K = \begin{bmatrix} \frac{t_1}{a_1} & \frac{t_2}{a_1} & -\frac{b_1}{a_1} \\ \frac{t_1}{a_2} & \frac{t_2}{a_2} & -\frac{b_2}{a_2} \\ \frac{t_1}{a_3} & \frac{t_2}{a_3} & -\frac{b_3}{a_3} \end{bmatrix}$$

Let $s_i \in R_j$ and $s_i[c] = K_{ic} = \frac{t_c}{a_i}$, then let $r_{i+1} = \frac{a_{i+1}}{a_1} \Rightarrow a_{i+1} = a_1 r_{i+1}$, by observe we can find that $s_1 = \frac{a_{i+1}}{a_1} s_{i+1} = r_{i+1}s_{i+1}$, use least square to solve for $r_1, r_2, .., r_i$, also by the K matrix we can get:

$$a_i = r_i a_1$$

$$t_j = a_1 K_{1j}$$

$$b_i = -a_i K_{i(j+1)} = -r_i a_1 K_{i(j+1)}$$

Then we can estimate exposure time by giving a a_1

Therefore, technically, the assumption of exposure time can be dropped if a_1 is given, usually we can assume it is 1.

3.3 Challenges during implementation

3.3.1 Image alignment

However, if the image has extremely low or high brightness, the process alignment will be really hard. The reason is that if the brightness is too low or too high, the RGB range is too narrow for the whole image; therefore, it will be hard to find the corresponding pixels between images, which will affect the alignment of the images.

The method that we used to solve this problem is to use a ‘for loop’ to find the corresponding points. If the number of corresponding points returned from matching feature is less than the threshold number (in our implementation threshold number = 15), we loop through the image again. The process will loop at most 20 times, if after 20 loops there is still not enough corresponding points, we will ditch the image and go to the next one.

The reason we chose to loop for 20 times is that for most of the image this is a sufficient number of loops for us to find a threshold number of corresponding points. We also have tried using while loop to find maximum number of corresponding points but there are a few images that is too dark or too light to find enough corresponding points so that it ends up to be an infinite loop.

The reason that we are ditching the picture that won’t return enough corresponding points is, if we include these pictures, too many out-liners will result in a wrong transformation projection matrix, the resulting HDR will be stretched, since some of the images were poorly aligned with little number of corresponding points.

3.3.2 Sample selecting

In our implementation, we used two methods to select the sample points. The first method is to resize or downsampling the image into 10*10 pixels, thus we have a 100 points sample. By this method, the computed camera response curve (Figure 18) seems a little bit off at the end.

Secondly we used K-means for sampling. K-means can compute a smoother response curve (Figure 19) using a k-cluster of around 20, the cluster number may varies according to the size of the input image. .The reason is that the selected sample points in k-means using RGB-XY have lower intensity variance, thus they are better representation of the image. The only weakness of this method is that k-means takes a lot longer than simple resize or down sampling. Although the response curve is better, it is not time-efficient since the resulting radiance map (Figure 20.2) is almost identical to radiance map using method 1 (Figure 20.1). Also, by using a bigger smoothness constant (in our implementation, function gsolve’s parameter L) in method 1 we can also result in a smoother response curve.

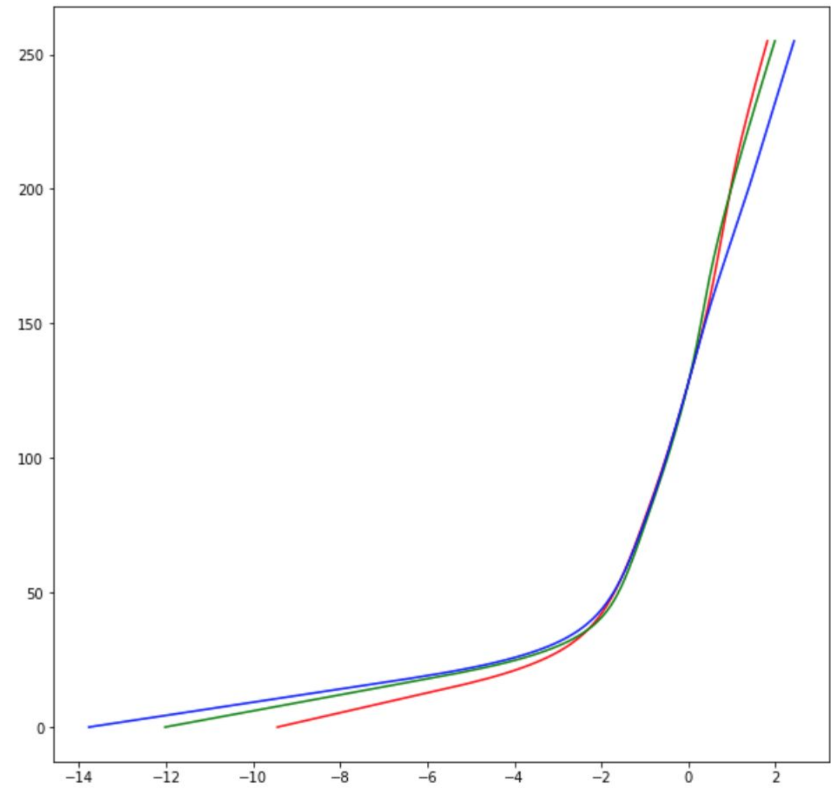

Figure 18: Camera response curve (Resize/downsampling)

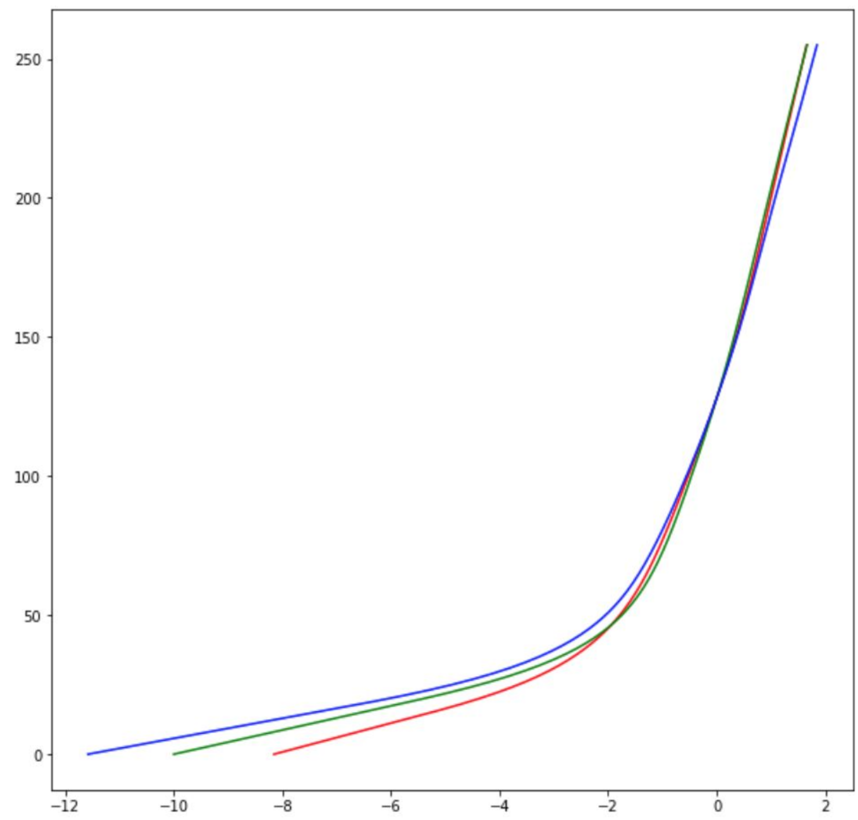

Figure 19: Camera response curve(K-means)

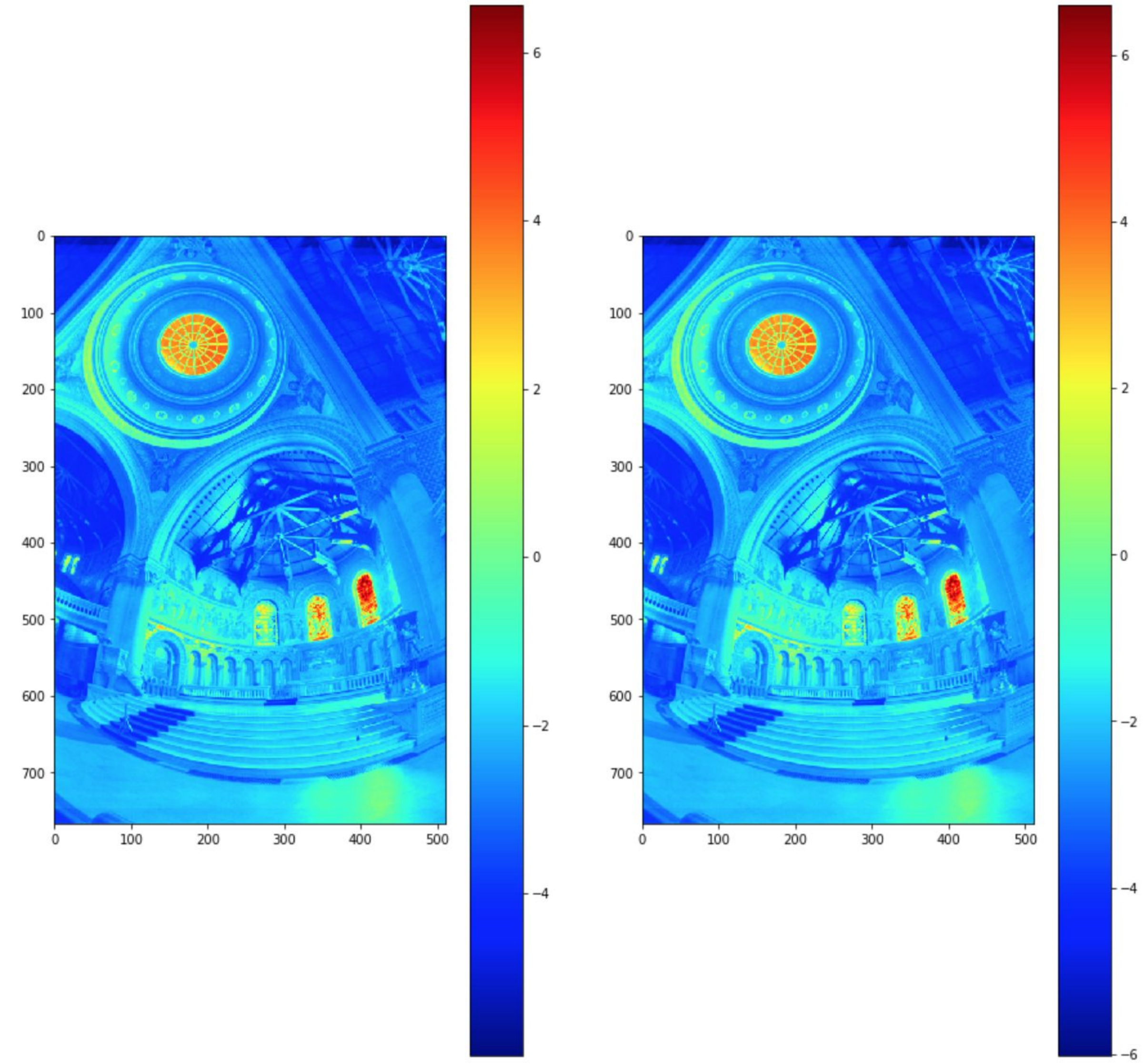

Figure 20.1, 20.2: Radiance map (left: resize; right: K-means)

3.3.3 Tone mapping

Tone mapping is a method used in image processing to map one set of colors to another to approximate the appearance of HDR in a medium that has a limited dynamic range. It is invented to mimic the coloring result of printers, CRT or LCD monitors which have limited dynamic range that is inadequate to reproduce the full range of light intensities represent in natural scenes.

In our implementation we only used globalized tone mapping, which will generally apply to the whole image, instead of localized tone mapping, which can be applied to a segment of the image.

Sigmoid Tone mapping

Sigmoid Tone mapping is one of the most commonly used tone mappings.

It is a S shaped function to adjust the smoothness of the detail of image. It will smooth the dark spot to 0, and smooth the light spot to 1.

Equalize-adapthist

Equalize-adapthist is a tone mapping that mainly used to adjust the contrast level.

3.3.4 Exposure time information not given

For some of the images in the sample site, the exposure time information was not given. The only given information is the $\pm EV$. Therefore a preprocess of exposure time estimation is required.

In our case we hand processed the exposure time, and the process is explained below.

For the $0\,EV$ image, which is taken under normal exposure condition, we set its exposure time to 1. Then, according to the EV value, the exposure time can be computed as 1×2^{EV} . If the EV value is positive, then the exposure time will be longer and the image would be over exposed. If the EV value is negative, the image would be under exposed.

3.3.5 Runtime issue

The runtime of our implementation have two big segments: the K-means sampling and the computation of radiance map. The K-means takes around 2 mins for small to medium size image and even longer for large images. The radiance map can take around 2-3 mins for large images because it is loop through the image pixel by pixel. Unfortunately, so far we haven’t find a way to optimize this implementation.

4. Conclusion

4.1 Accomplishment

In our implementation, we manage to align a set of pictures that are under different exposures. After selecting representative samples using either resize/downsampling or k-means, a radiance map is computed. Using the radiance map, by normalizing the pixels to range [0,255] we can construct the result HDR. The result HDR are images with great color contrast, reasonable exposures and great detail. Three methods of tone mapping were used to reconstruct the HDR. 4 tests were demonstrated using different methods of sampling and different tone mappings.

4.2 Future Improvement

One of the improvement that we should consider is to customize individual tone mapping for different segments of the image. For example, result image sample from the internet (Figure 21) comparing to our HDR image Figure 9, has more realistic and contrast color. The reason is that we are using the same tone mapping for the whole picture which is also called global tone mapping, but the sample result image on the internet (Figure 21) was using localized tone mapping for the sky and the river waves separately. Also, since the waves are moving while the image was taken, they are quite different in each picture, simple image alignment does not work well for this set of image, severe ghost artifact can be observed in the result HDR image. Therefore, a different approach should be considered to process this kind of image, such as manually remove the ghost artifact using some software like photoshop and etc.

Figure 21: Sample result image on internet

5. References

Cerman, L., & Hlavac, V. (2006, February). Exposure time estimation for high dynamic range imaging with hand held camera. In *Proc. of Computer Vision Winter Workshop, Czech Republic*.

Debevec, P. E., & Malik, J. (2008, August). Recovering high dynamic range radiance maps from photographs. In *ACM SIGGRAPH 2008 classes* (p. 31). ACM.

Mallick, S. (2018). High Dynamic Range (HDR) Imaging using OpenCV (C++/Python) | Learn OpenCV. Retrieved from https://www.learnopencv.com/high-dynamic-range-hdr-imaging-using-opencv-cpp-python/

Okonek, B. (2018). *Sample HDR photos created with easyHDR*. [online] Easyhdr.com. Available at: https://www.easyhdr.com/examples/ [Accessed 9 Dec. 2018].