VFX HW1

Outline

We implemented the following features. The bonus part is highlighted in bold.

- **Image Alignment**
- HDR Reconstruction
  - Paul Debevec's method
  - **Mitsuaga and Nayar's method**
- Tone Mapping
  - OpenCV
  - **Reinhard**
  - **Bilateral Filter**
- **Ghost Removal**

Image Alignment

We implement Multiscale technique for Image Alignment. We align every other images with the fourth image, since it take in the middle of total time span, and we expect the total need if shift is minimized.

First, we need to obtain MTB of each image. We use csv.cvtColor to transfer image into gray scale, and use mean of the whole photo as threshold. The result is as following.

We also apply mask on pixels that are in range of $threshold \pm 4$, the result after applying mask is as folowing.

MTB result	MTB reslut after applying mask
	

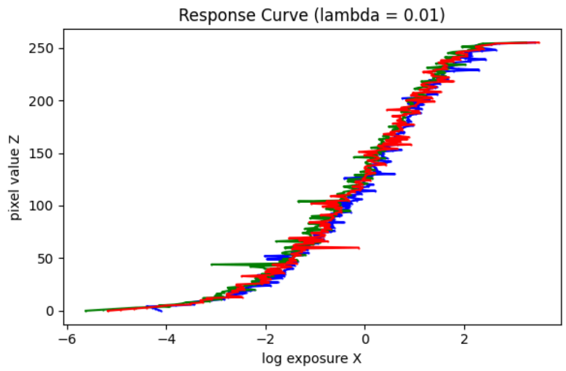
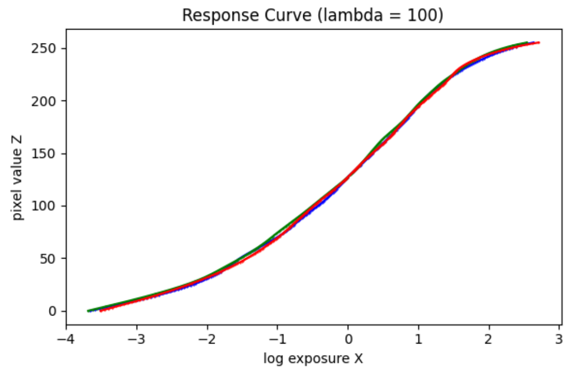
After gererating MTB, we halved the width and height for 7 level. For every level, we take the result from previous level, double it, and try if additional shift within one pixel will make current shift a better match.

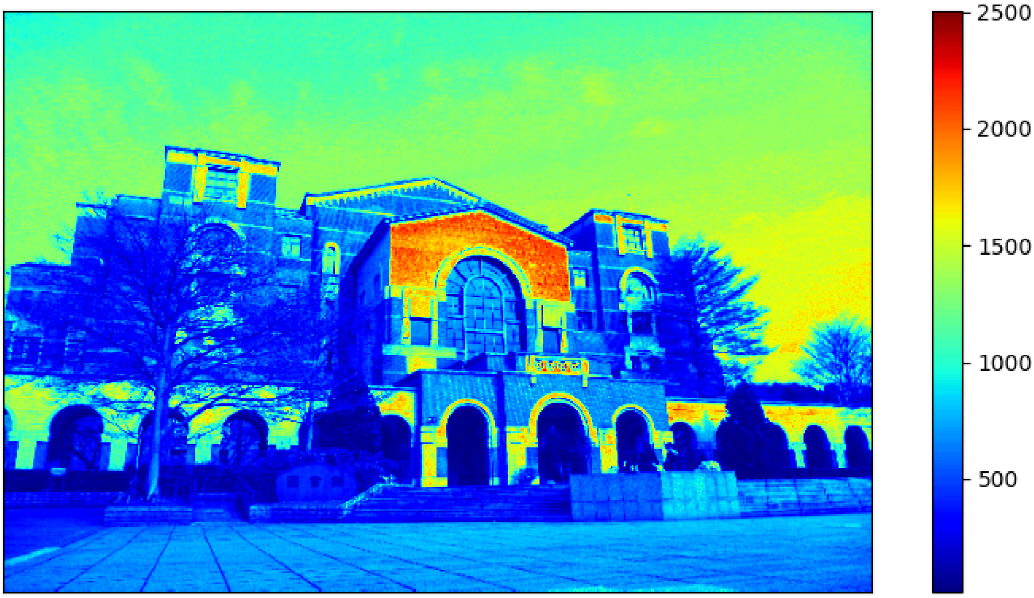
After finding best shift of each photo. We want to add same padding to each image so the affect of shifting could be ignored. However, if pure color padding is added, it may cause problem if HDR reconstruction algorithm pick the pure color pixel as sample, we might get the wrong response curve. To solved the problem, we chop the image and apply cv2.copyMakeBorder with cv2.BORDER_REFLECT as padding.

HDR Reconstruction

1. Paul Debevec's method

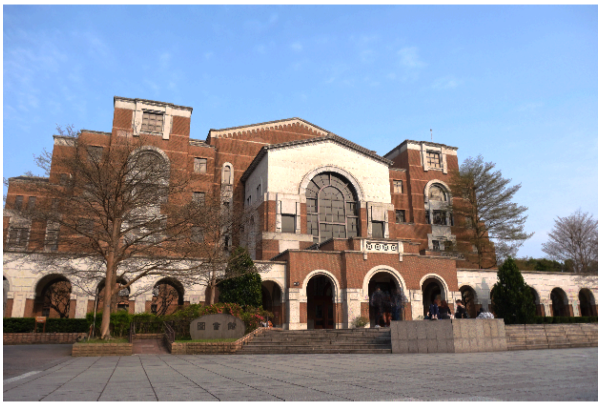
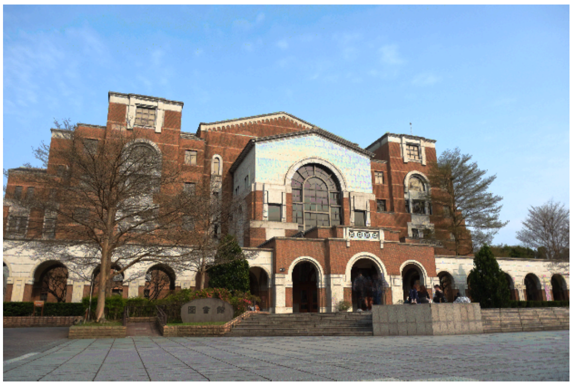
First, we recovered the response curve g with images under different exposure time(shutter speed). We set $lambda = 100$ to smooth the curve. Afterward, we applied the images to the curve g to reconstruct the HDR radiance map.

Lambda = 0.01	**Lambda = 100**
	



What's worth noting is, instead of randomly choosing 50 points to recover the response curve, we downsampled the image to size $20 * 30$ and chose the middle $10 * 15$ points to recover the response curve. The reason is randomly sampling points may lead to large bias(imagining sampling 50 points with value close to 255), and usually the main object lays in the middle of the image.

In the comparison below, you can find that randomly choosing 50 points causes some artifacts on the facade of the library. (The HDR images here are tone-mapped using Reinhard global mapping)

Downsample	Randomly Choose 50 points
	

2. Mitsuaga and Nayar's method

In this method, we assume the response function is a polynomial function with degree $M = 5$.

We use formula as following to iterationally update $R^{(k)}_{j,j+1}$

$$R^{(k)}_{j,j+1} = \frac{1}{P}\sum_{i=1}^{P}\frac{\sum_{n=0}^{N} c_n^{(k-1)} Z_{i,j}}{\sum_{n=0}^{N} c_n^{(k-1)} Z_{i,j+1}}$$

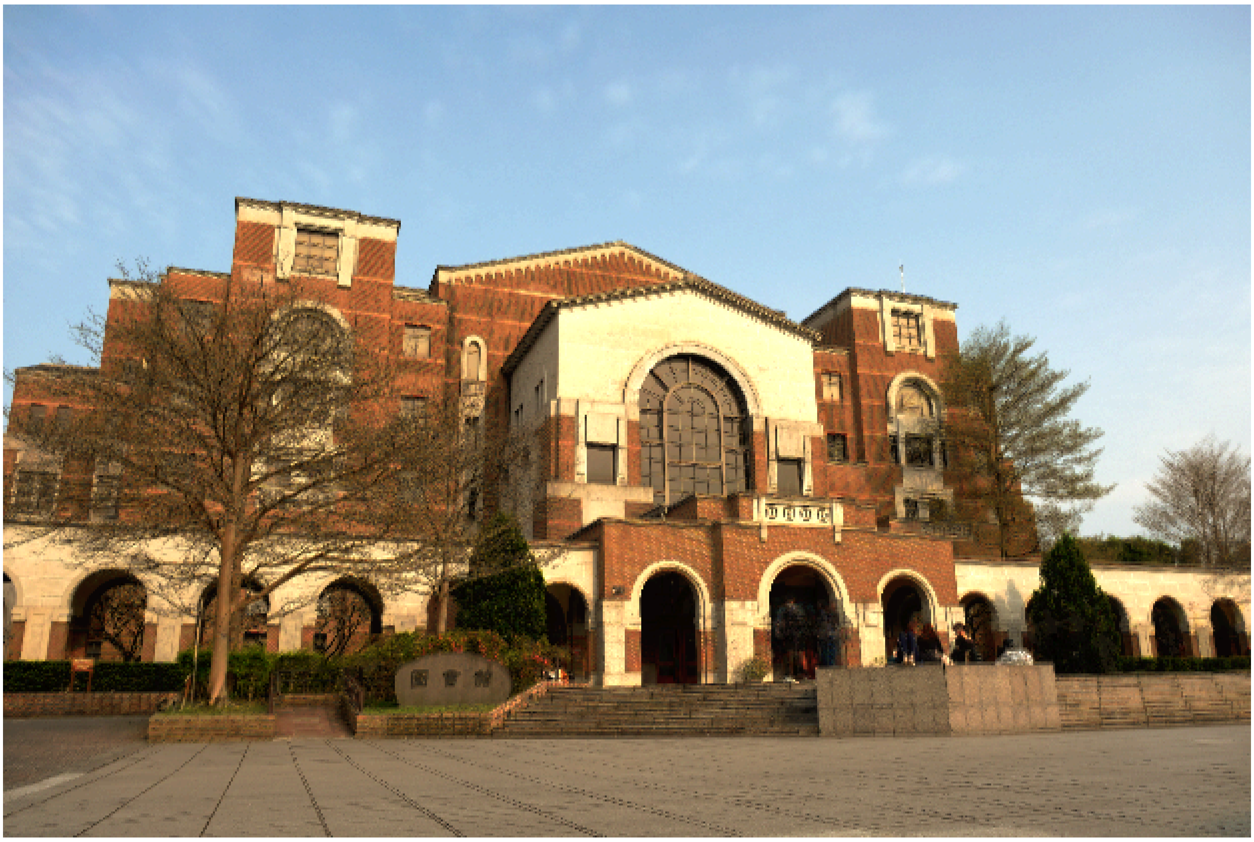
And solve the linear system of $\frac{\partial \varepsilon}{\partial c_n} = 0$ and let $c_M = E_{max} - \sum_{m=0}^{M-1} c_m$ to update $c^{(k)}$

Keep doing above formula until $\mid f^{(k)}(Z) - f^{(k-1)}(Z) \mid < \epsilon$

Calculate $E_{i,j}$ of a point $Z_{i,j}$ by $E_{i,j} = \frac{\sum_{p=1}^{P} w(Z_{p_{i,j}}) * f(Z_{p_{i,j}})}{\sum_{p=1}^{P} w(Z_{p_{i,j}})}$. We found out that $w(Z) = (Z - 220)^2$ perform the best by experiment.

At this point, we've only get $E = f(Z)$ for each color. We need to obtain the relative value of I of each color. We asume that the relationship of E between colors are same as it in Z. We have $Z = [Z_r, Z_g, Z_b]$, $E = [E_r, E_b, E_g]$. And assume that $E_c = [k_r E_r, k_g E_g, k_b E_b]$ which is the correct E of each color. We apply least-squares minimization on $\frac{|M|}{||M||} = \frac{|E_c|}{||E_c||}$.

Following is the result of this method using Reinhard global mapping.



Tone Mapping

1. OpenCV

Algorithm	Parameter	Image
Reinhard	gamma=1.5 intensity=0 light_adapt=0.5 color_adapt=0	
Drago	gamma=1.5 saturation=1 bias=0.85	
Mantiuk	gamma=1.5 scale=0.7 saturation=1	

2. Reinhard
   We handcrafted the Photographic Tone Repproduction method proposed by Reinhard. We implemented the global and local operators and used the same setting as mentioned in paper *"Erik Reinhard, Michael Stark, Peter Shirley, Jim Ferwerda, Photographics Tone Reproduction for Digital Images, SIGGRAPH 2002."* (phi=8, threshold=0.05, 8 Gaussian map increasing with factor of 1.6)

   We tried different combination of a and l_{white} and found $a = 0.8$, $l_{white} = 2.5$ yield the best result.

Global operator	**Global + Local operators**
	

3. Bilateral Filter
   We implemented the Bilateral Filter as follows.
   1. Get the intensity map by summing the BGR value with weights $(0.06, 0.67, 0.27)$.
   2. Divide the HDR image with intensity map to acquire the color map.
   3. Pass the intensity into bilateral filter to get large scale map.
   4. Divide the intensity map with large scale map to get detail map.
   5. Reduce contrast of large scale map in log domain.
   6. Combine large scale map and detail map with different weight multiply with color map to get the output

Ghost Removal

We have multiple images when constructing the radiance map, so we can calculate E_{ij} for pixel i in image j and average them to get the mean of E_i. Because the image causing the ghost effect at pixel i normally will have different value then the value of pixel i in other images, we can simply assign lower weight to images whose value in pixel i is far from the mean of E_i during the radiance map construction.

The result is depicted below.

With Ghost Removal	Without Ghost Removal