VFX PROJECT1

HIGH DYNAMIC RANGE IMAGING

電機三 B03901156 黃于瑄

VFX PROJECT1

HIGH DYNAMIC RANGE IMAGING

Introduction

High dynamic range imaging (HDRI) is a high dynamic range (HDR) technique used in imaging and photography to reproduce a greater dynamic range of luminosity. The aim of HDRI is trying to present a similar range of luminance to the experienced through the human visual system. HDR images can represent a greater range of luminance when scenes containing very bright, direct sunlight to extreme shade, or very faint nebulae in real world shoot by capturing and combining several different, narrower range, exposures of the same scene. Therefore, to do HDR imaging in this project, the first step will be to take photographs under different exposures by changing shutter speeds and do image alignment. The second step is to reconstruct the response curve of each color channel and obtain the radiance map by utilizing these raw image data. After obtaining the radiance map, tone mapping has to be implemented to generate the LDR image.

In this project, I use **C++ (Visual studio 2015)** with **OpenCV 3.1.0** to implement the algorithms for images alignment and tone mapping as well as **MATLAB_R2013a** for HDR as well as tone mapping. In addition, the whole system is tested on **Windows 10**.

Taking Photographs

In this project, I have shoot four sets of photographs with different exposure time by CANON EOS 70D, and utilized Canon's EOS utility to remotely control the shutter speeds and shooting. The following are the shooting conditions of each shooting image set:

imageSet1 → Size: 3648 X 5472 pixels, ISO: 500, Aperture: f / 14
imageSet2 → Size: 5472 X 3648 pixels, ISO: 500, Aperture: f / 14
imageSet3 → Size: 5472 X 3648 pixels, ISO: 500, Aperture: f / 10
imageSet4 → Size: 5472 X 3648 pixels, ISO: 320, Aperture: f / 18

Image Alignment (BONUS)

Although I have utilized the remote control software to avoid images shift while shooting, there may still be some vibrations caused by the camera at the moment of shooting scenes, which brings the result of pixel shifts.

In this part, I have implemented the algorithm **Median Threshold Bitmap (MTB)** mentioned in the course. The whole design of algorithm is on the below:

Read Input Images

Read the image set from input_image directory and save them in a Mat vector.
Choose a specific image from the set to be the standard for alignment.

Convert the images to grayscale

Convert each image from RGB to grayscale.

Function getExpShift

This function will recursively resize the two images for comparison to the 1/2 size of their original ones, building the images pyramid and compute the bitmaps based by thresholding the input images using the median of intensities.
From the bottom of the image pyramid, search for the optimal offset in horizontal and vertical directions from all possible nine directions {(-1, -1),(-1, 0), (-1, 1), (0,-1), (0, 0), (0, 1), (1,-1), (1, 0), (1, 1)} by doing bitwise xor (pixel difference)/and (pixel similarity)operations and choosing the offsets which make the fewest difference.
Return the optimal offset result to the above layer images and save the final offset results in an array.

Function align

According to the offset array, implement pixel-shift correction on the input images.

Assembling HDR image

The algorithm is based on the **Debevec's** method mentioned in the course.

Assembling HDR image

a. Math for recovering response curve :

$$Z_{ij} = f(E_i \Delta t_j)$$

let us define function $g = \ln f^{-1}$

$$g(Z_{ij}) = \ln E_i + \ln \Delta t_j$$

Design an objective function

$$\mathcal{O} = \sum_{i=1}^{N} \sum_{j=1}^{P} [g(Z_{ij}) - \ln E_i - \ln \Delta t_j]^2 + \lambda \sum_{z=Z_{min}+1}^{Z_{max}-1} g''(z)^2$$

b. Recovering response curve

Solve the linear equation that minimize the objective function by SVD and we can get the function g including gR,gB,gG for RGB channels with very small domain size.

c. Reconstruct the HDR image

After deriving the g function, we can apply the formulas on the below to reconstruct radiance maps of the HDR images.

$$\ln E_i = \frac{\sum_{j=1}^{P} w(Z_{ij})(g(Z_{ij}) - \ln \Delta t_j)}{\sum_{j=1}^{P} w(Z_{ij})} \quad w(z) = \begin{cases} z - Z_{min} & \text{for } z \le \frac{1}{2}(Z_{min} + Z_{max}) \\ Z_{max} - z & \text{for } z > \frac{1}{2}(Z_{min} + Z_{max}) \end{cases}$$

Tone Mapping (BONUS)

I have implemented the algorithm Photographic Tone Reproduction (Reinhard's) to do tone mapping on the HDR image obtained from the above steps.

Model:

Basically, this method can only compress the bright side and doesn't help a lot for the contrast enhancement of the dark side. Since the whole algorithm is simple for researchers to understand, it is applied commonly for application.

a. Set small value of delta δ = 0.001 to avoid singularity if black pixels are present in the image
b. a is the key value of the image when applying scaling. Usually a is ranged from 0.045 to 0.72. Consider the shooting condition of the four image sets in this project, I use a = 0.18 here.
c. Compute log-average luminance ($L_w(x, y)$):

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

d. Compute scaled luminance (Lm):

$$L_m(x,y) = \frac{a}{\bar{L}_w} L_w(x,y)$$

e. Compute maximum luminance (Lwhite):
Lwhite = max($L_w(x, y)$);
f. The global operator of tone mapping ($L_d(x, y)$):

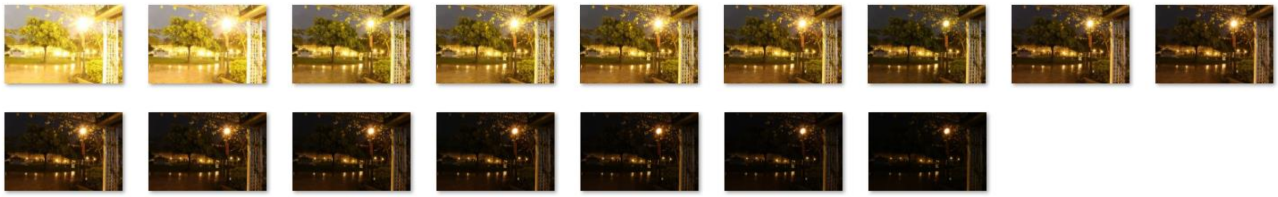
$$output = L_d(x,y) = \frac{L_w(x,y) * (1 + \frac{L_w(x,y)}{Lwhite^2})}{1 + L_w(x,y)}$$

Results

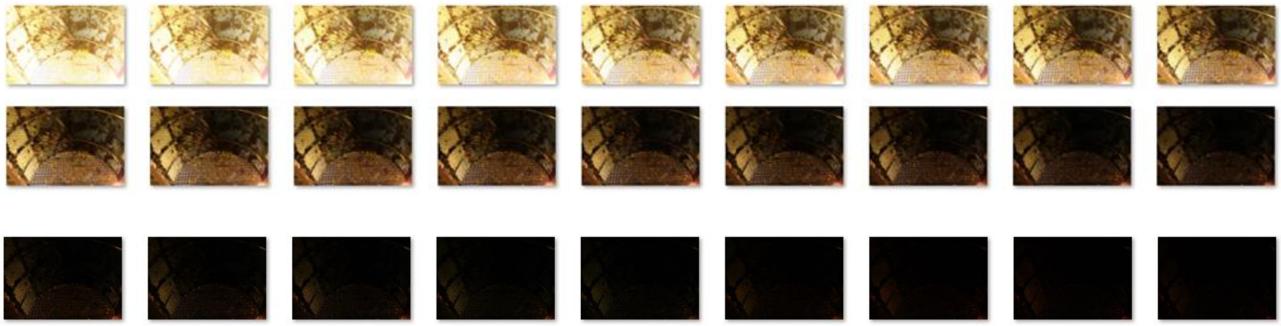
Taking Photographs

imageSet1 (Size: 3648 X 5472 pixels, ISO: 500, Aperture: f / 14)

imageSet2 (Size: 5472 X 3648 pixels, ISO: 500, Aperture: f / 14)



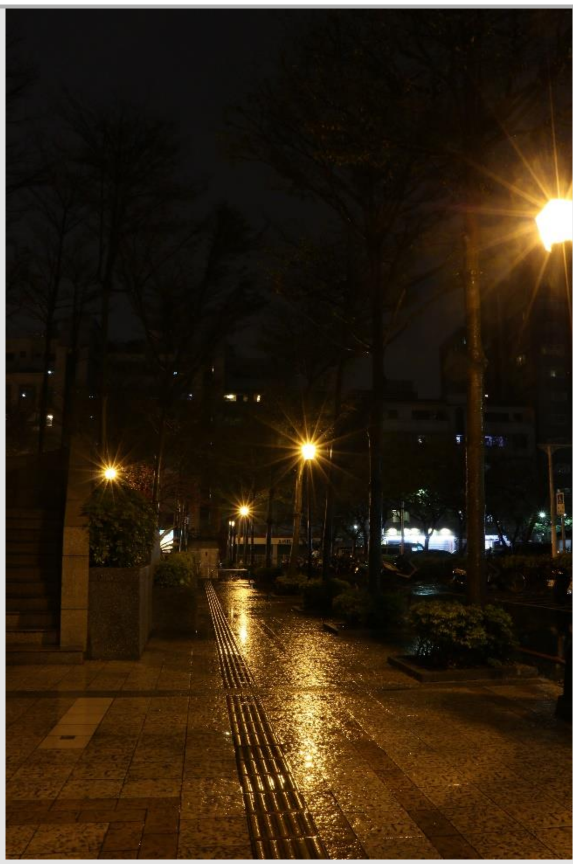
imageSet3(Size: 5472 X 3648 pixels, ISO: 500, Aperture: f / 10)



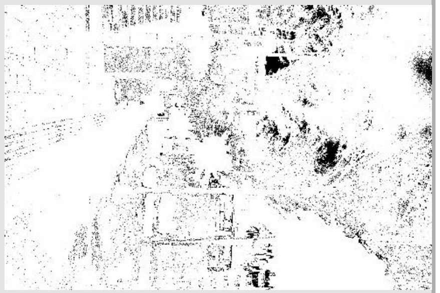
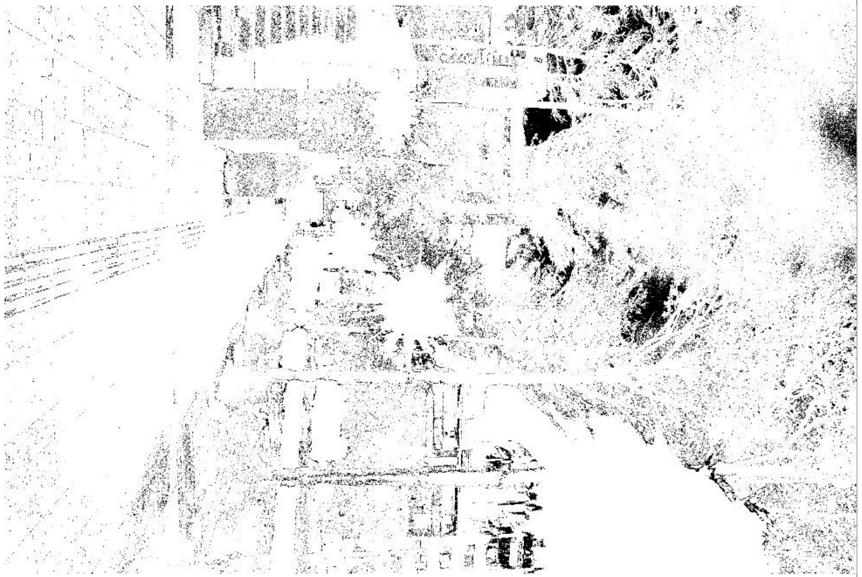
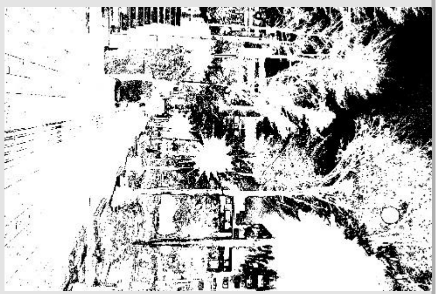
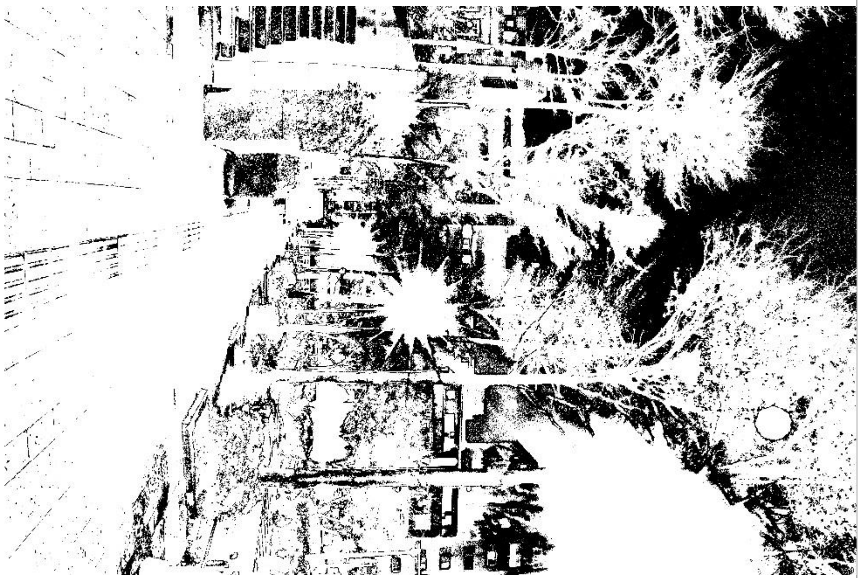
imageSet4 (Size: 5472 X 3648 pixels, ISO: 320, Aperture: f / 18)

Image Alignment

Alignment Implementation: Pyramid Structure

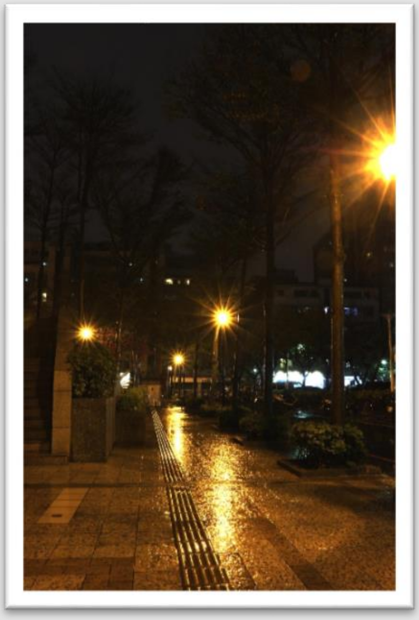
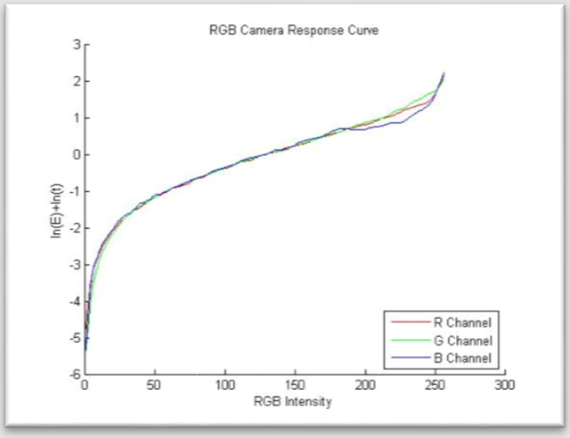
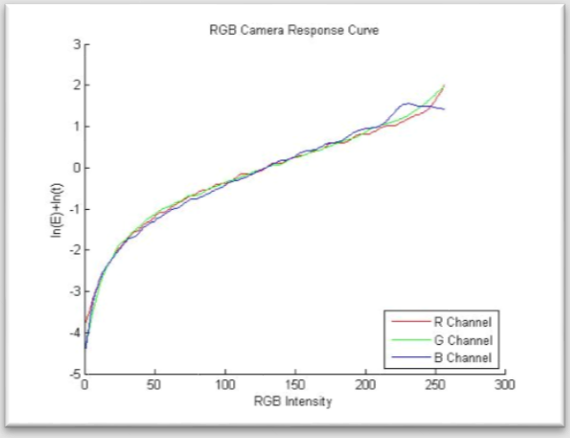
	Exposure time = 30 sec	Exposure time = 1.33 sec
Input Image		
Threshold Bitmap(layer 6)		

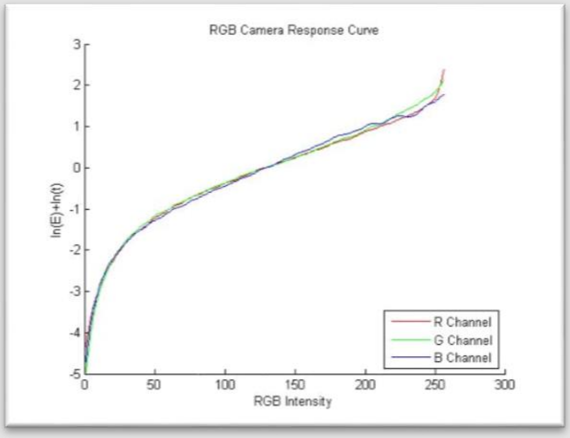
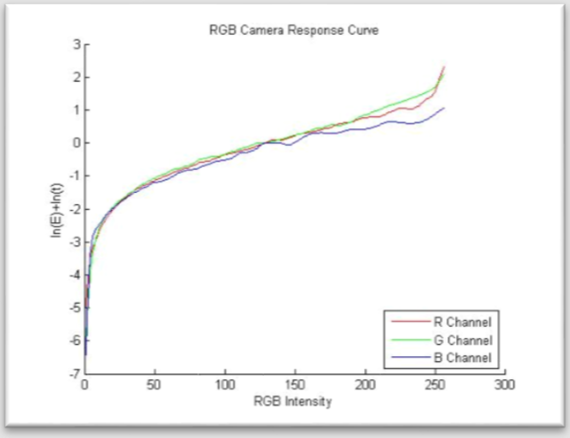
Threshold Bitmap (layer 5)		
Threshold Bitmap (layer 4)		
Threshold Bitmap (layer 3)		
Threshold Bitmap (layer 2)		
Threshold Bitmap (layer 1)		
Threshold Bitmap (layer 0)		

EB (2)	EB (3)	EB (layer 4)	Exclusion Bitmap (layer 5)	Exclusion Bitmap (layer 6)
				
				

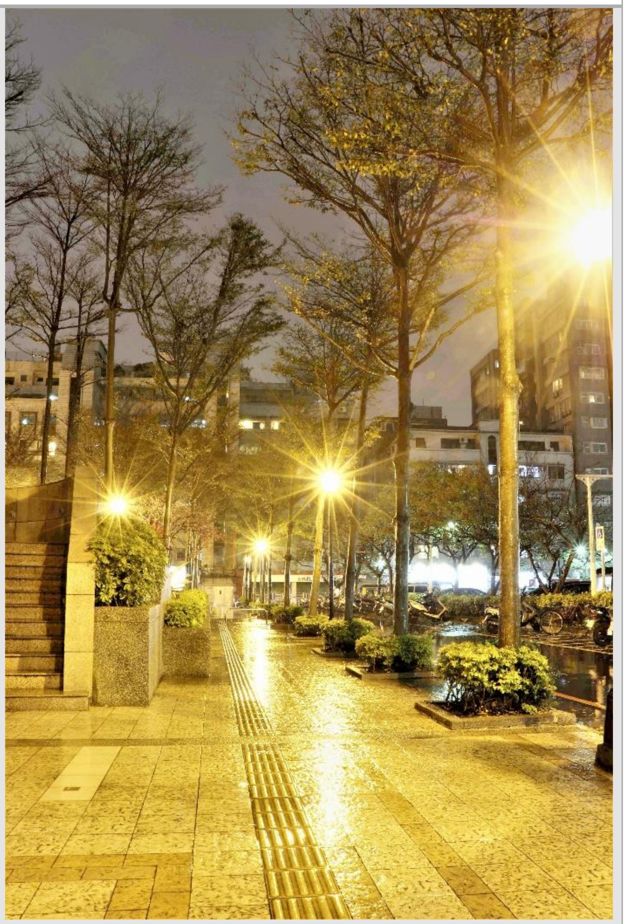
EB (1)		
EB (0)		

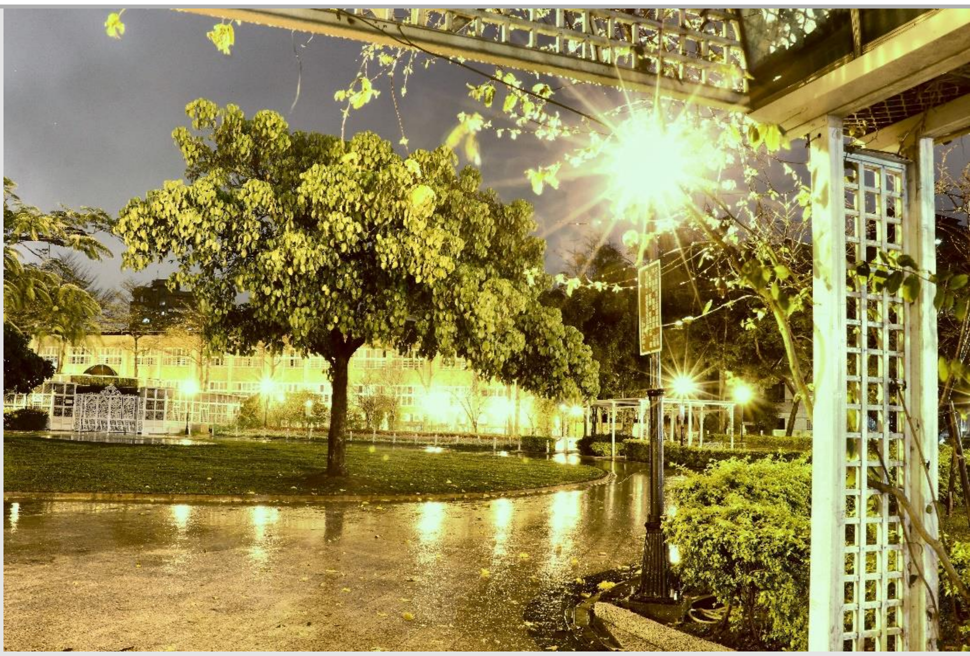
Assembling HDR image

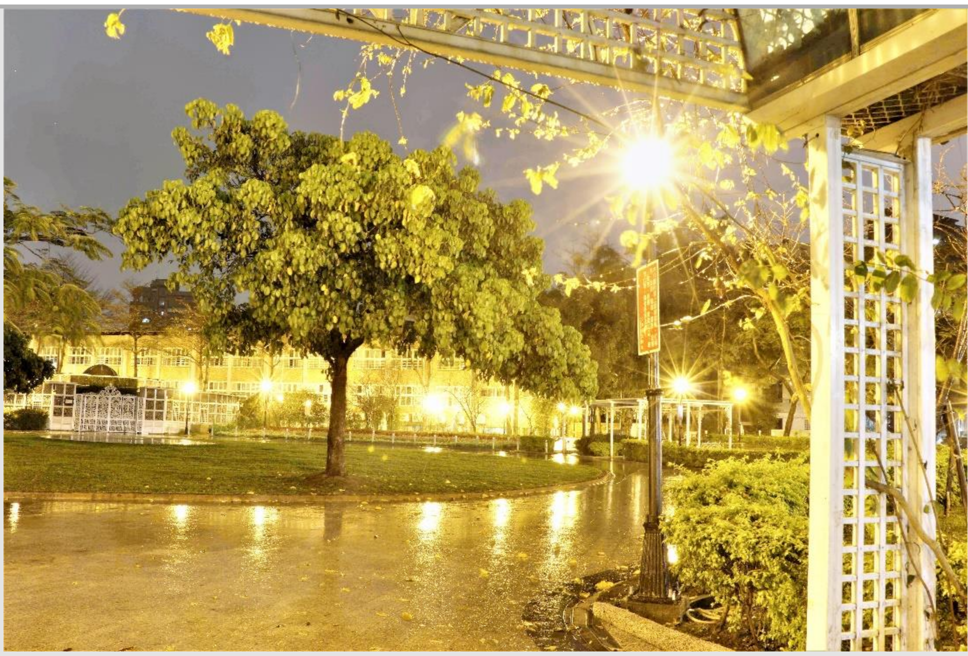
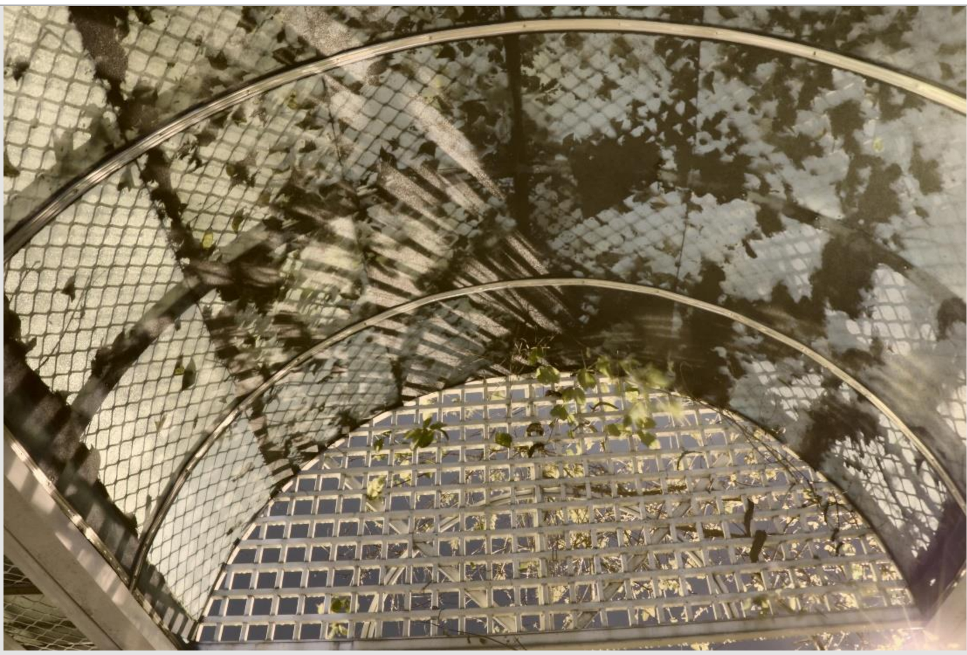
imageSet1_hdr_image	imageSet1_response_curves
	
imageSet2_hdr_image	**imageSet2_response_curves**
	

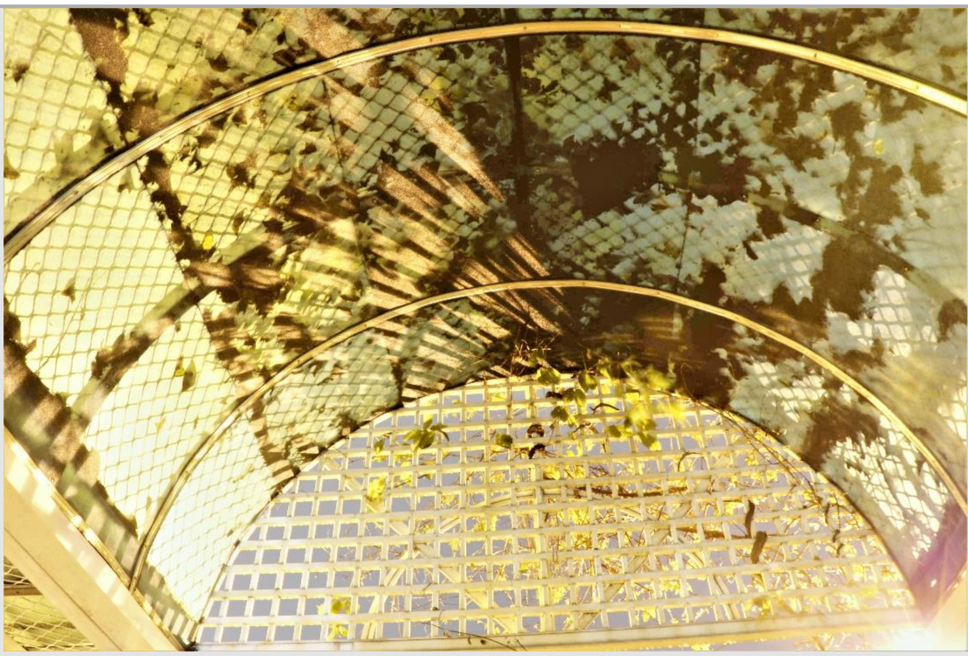
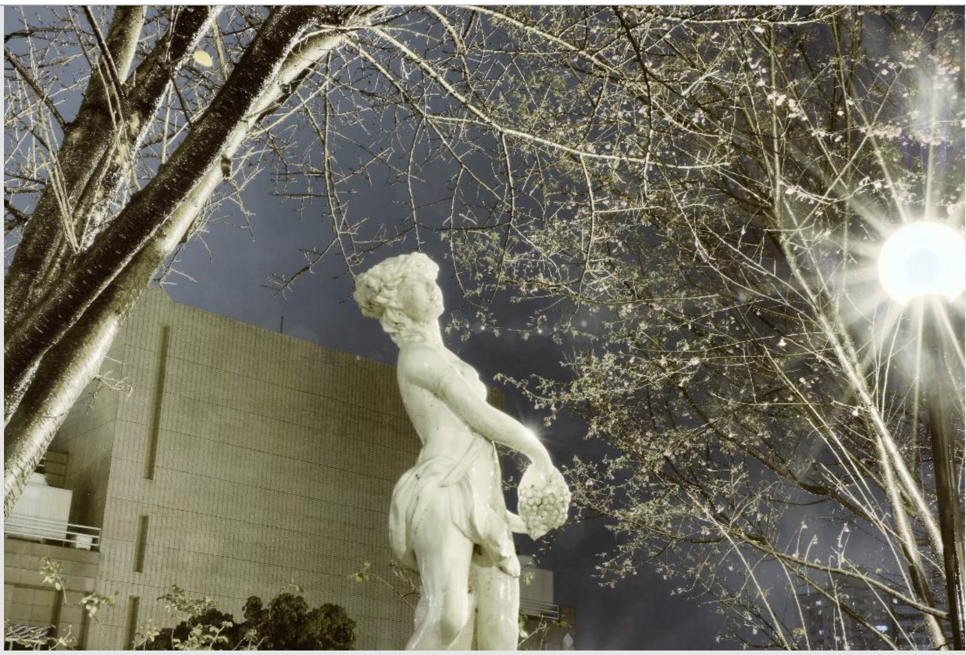
imageSet3_hdr_image	imageSet3_response_curves
	
imageSet4_hdr_image	**imageSet4_response_curves**
	

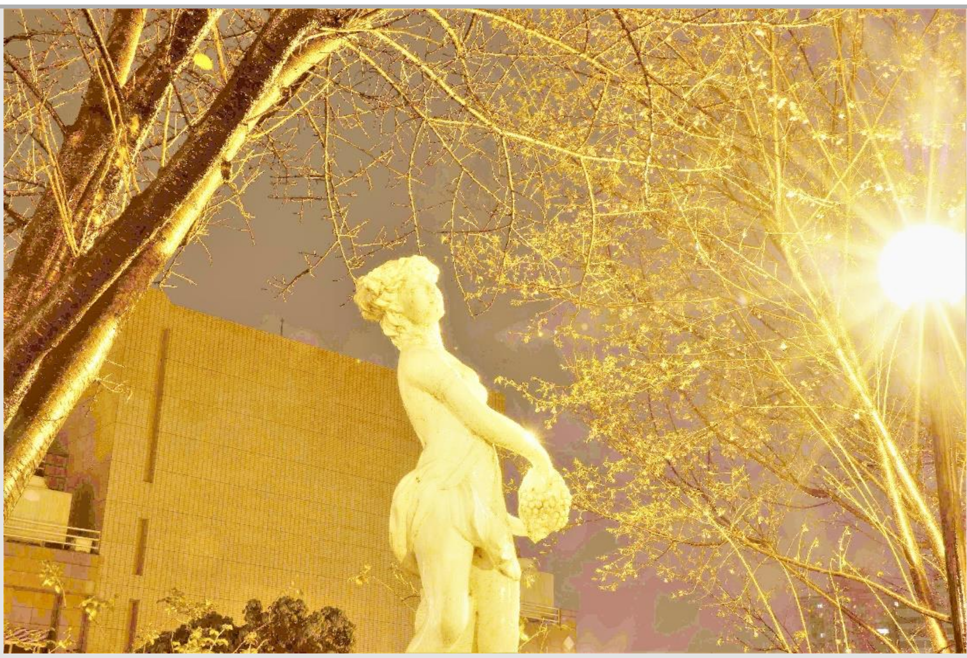
Tone Mapping

	MATLAB tonemap()	Reinhard
imageSet1		

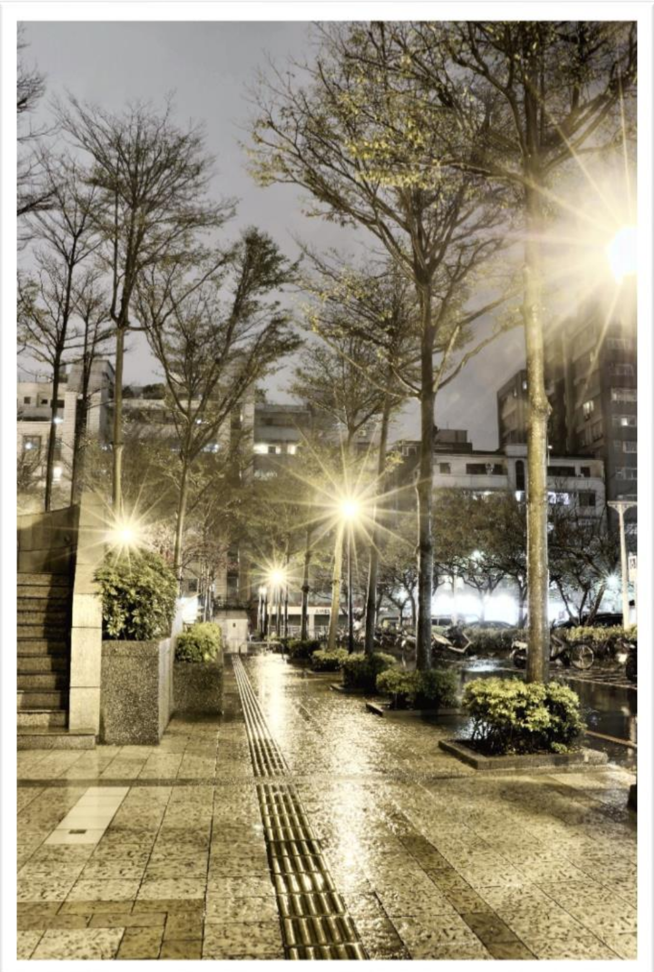
	MATLAB tonemap()
imageSet2	

	Reinhard
imageSet2	
	MATLAB tonemap()
imageSet3	

	Reinhard
imageSet3	
	MATLAB tonemap()
imageSet4	

	Reinhard
imageSet4	

Artifact



Reviews and Finding

a. The images shot by with tripod have better performance than hand-held ones even though the alignment method can be applied after photographing. The reason may be that bitmap technique focus on dealing with the images with small offset differences, and it can't handle zooming and flipping differences.

b. The reconstruction of the response curves of one camera are almost the same in spite of different color channels or exposure sequences.

c. From the result images of two different tone mapping methods, it's obvious that the colors of images tone-mapped by Reinhard's method are more close to yellow than the ones tone-mapped by MATLAB embedded function, which indicates that Reinhard's algorithm can convert the HDR images to the ones close to what human views in real scenes.

d. It can be observed that the MATLAB embedded function performs better than the other method when it comes to dealing with the details in images.

e. From the observation of the tone mapping result images of the imageSet4, it can be seen that the edge of the scenes has some white boundary no matter which tone mapping method is applied on them. I think the reason may be that the quality of original input images is low and the atmosphere light is insufficient even with long exposure time, which hides many details of the scene and leads to the poor result of the tone mapping image.

Reference

a. Greg Ward, Fast, Robust Image Registration for Compositing High Dynamic Range Photographs from Handheld Exposures, 2003

b. Paul E. Debevec, Jitendra Malik, Recovering High Dynamic Range Radiance Maps from Photographs, SIGGRAPH 1997

c. Erik Reinhard, Michael Stark, Peter Shirley, Jim Ferwerda, Photographics Tone Reproduction for Digital Images, SIGGRAPH 2002.

d. Course slides (High dynamic range images, Tone mapping)