DFX Project: HDR

B00901022祝成豪＆B00902004徐粲翔

1. Project Description

The topic of this project is HDR (High Dynamic Range) image recovering. By taking multiple images of a scene, but with different amount of exposure time. We can recover the response function of our camera’s imaging process. With the response function, we can convert pixel values to radiance values. Thereby construct a High Dynamic Range Radiance Map.

1. Algorithms
2. )Image Alignment:

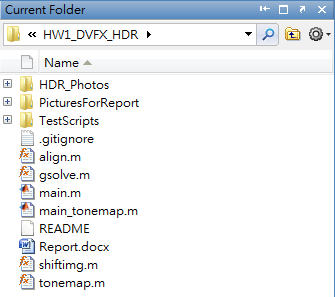
For images alignment, we implemented the Ward's [MTB (Median Threshold Bitmap) algorithm](http://www.csie.ntu.edu.tw/~cyy/courses/vfx/papers/Ward2003FRI.pdf). In which we convert our taken photos into grayscale images. And then use the median of intensities of each image as thresholds to create binary images. By shifting bit maps and XORing with each other. We can get the errors between maps. The shift with smallest error will be adopted as the shift of the image. Therefore achieve the goal of images alignment. After the shifting of pixels, we don’t crop the blank borders in the images since we used a tripod and the shifts are extremely small.

1. )Recover Response Function:

To recover the response function. We implemented the algorithm in Paul E. Debevec and Jitendra Maliks’ paper - Recovering High Dynamic Range Radiance Maps from Photographs. Since we already have the exposure time and the pixel values. We can recover a function g using this algorithm. Where g = ln(f^(-1)), and f is the function in the film reciprocity equation.

1. )Tone Mapping:

We implemented the dodging and burning. First, we calculate Lw by using the EMap.mat which is made before. Second we calculate average Lw. In the end, we can implement local operator and global operator respectively. In the local operator, we calculate Lsblur by using Gaussian. In the global operator, we calculate Ld by using Lwhite. Both operators have the formula in lecture slide, so that we can use it to generate two images (global and local).

1. Code Implementation

Our code is written in MATLAB. There are six code file in total. The main.m file includes all the processes before tone mapping, it will call the align.m function, shifting function and glove function to achieve our goal. After running main.m, we can recover a HDR radiance map. And then we can run the main\_tonemap.m which will call the tonemap.m by passing our radiance map as an argument. We will then receive two images as our final results.

Fig.Project Repository

The first thing we do is to use the matlab Canny edge detection function on all of our images and by ORing all the result bit maps, construct a bit map which have pixel value 0 for places that are detected as edges.

Fig.Edge detection result bit map

Fig.One of the original image

Second, we align our photos using MTB algorithm and take our photo number 4 as a reference photo. However, since we used the tripod when we were taking the photos. The result of MTB shows that there is no need to shift any of the images…… (One photo actually needed to shift a little bit when we used the original 6000x4000 resolution photos. We switched to 4500x3000 resolution later because of the extremely long time period our program needs to run.)

Fig.One median threshold bitmap

Fig.One grayscale image used in MTB

We then randomly pick 100 sample pixels for the recovering of our camera’s response curve. This is where we take our edge detection bit map into use. Since we do not wish to pick points that are on some edges to recover response curve. We check the random points with our edge bit map to make sure it is not on any edge. Otherwise it may lead to low accuracy. After that, we can recover the response function.

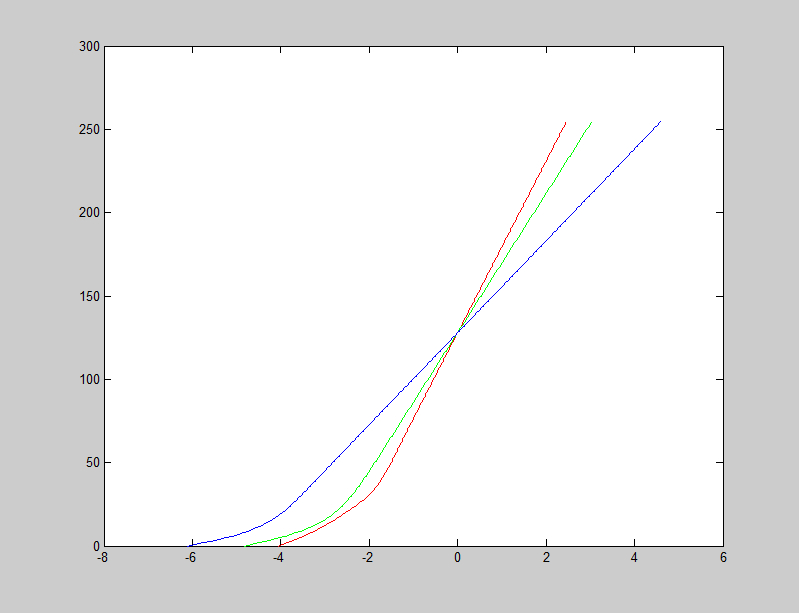


Fig. Recovered response function in green, red and blue.

Finally, we use the response function to reconstruct a high dynamic range radiance map from our original photos.

1. Results
2. Summary