VFX Homework 1 - High Dynamic Range Imaging

R10942152 游家權

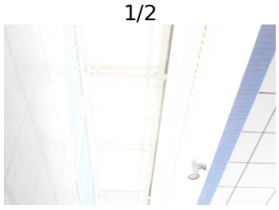
Algorithm that I implemented:

1. Paul Debevec's method for HDR
2. Robertson's method for HDR(Bonus)
3. MTB image alignment algorithm(Bonus)
4. Reinhard's method for Tone-mapping(Bonus)

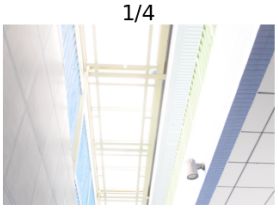
Original images

These images were shot at different shutter speeds.

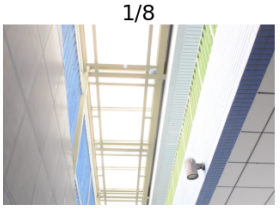
1/2



1/4

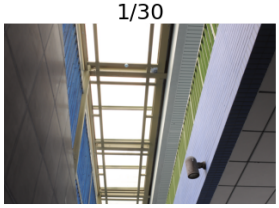


1/8

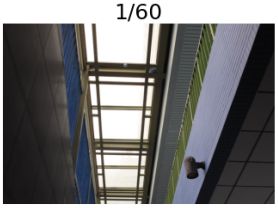


1/15

1/30



1/60



1/125

1/250

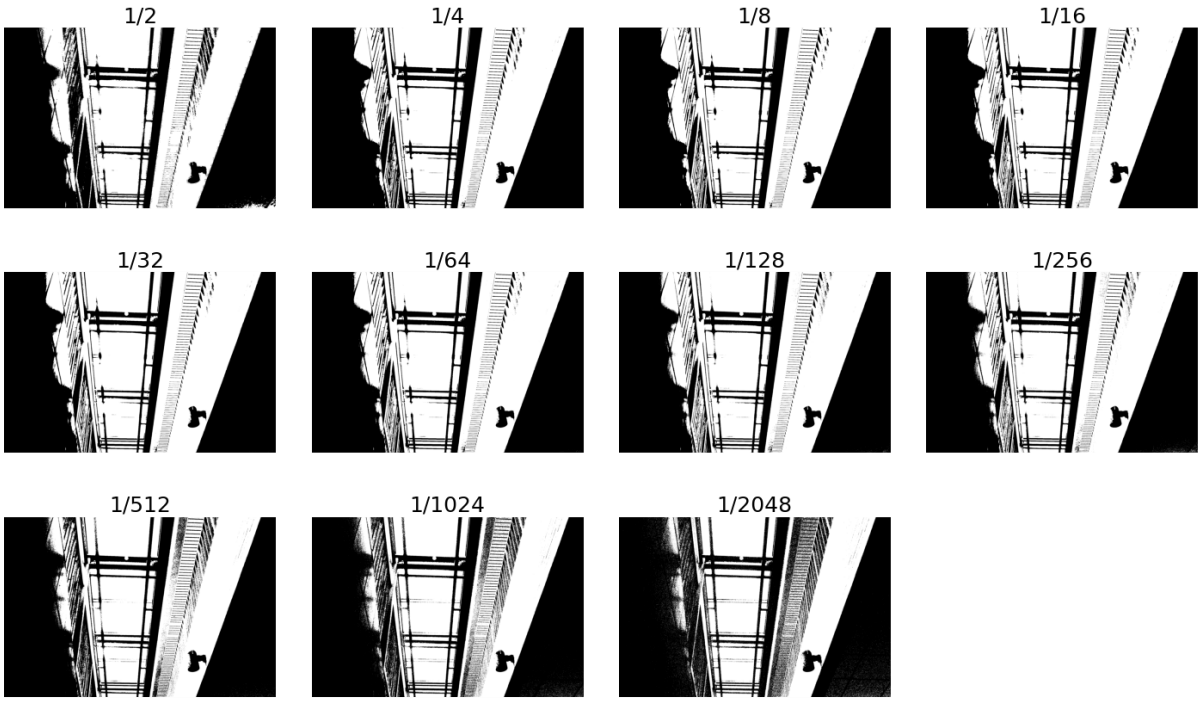
1/500

1/1000

1/2000

MTB Image Alignment algorithm

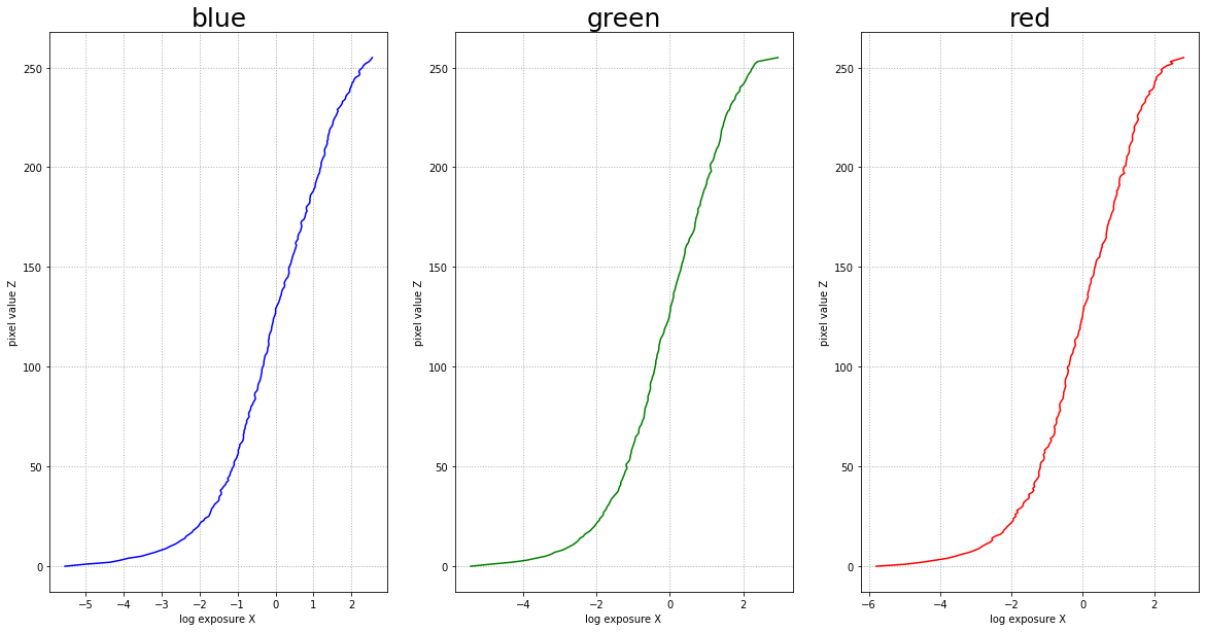
I used median thresholding to get similar binary images from different exposure images. The results are shown below. Even the brightest image(1/2) and the darkest image(1/2048) have very similar thresholding output.



Paul Debevec's methods

Paul Debevec's method uses some sample points to estimate the response curve. In my implementation, I used 300 randomly sample points to do this estimation. I also found that If I use more sample points the response curve could be even smoother.

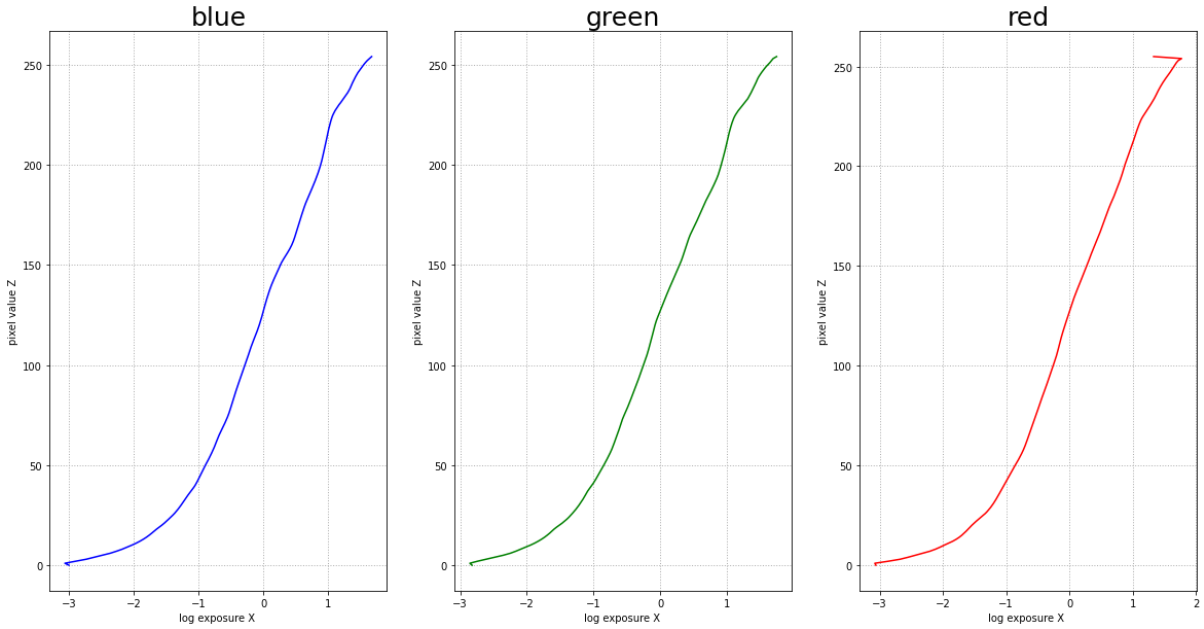
Response curve derived by Paul Debevec's method.



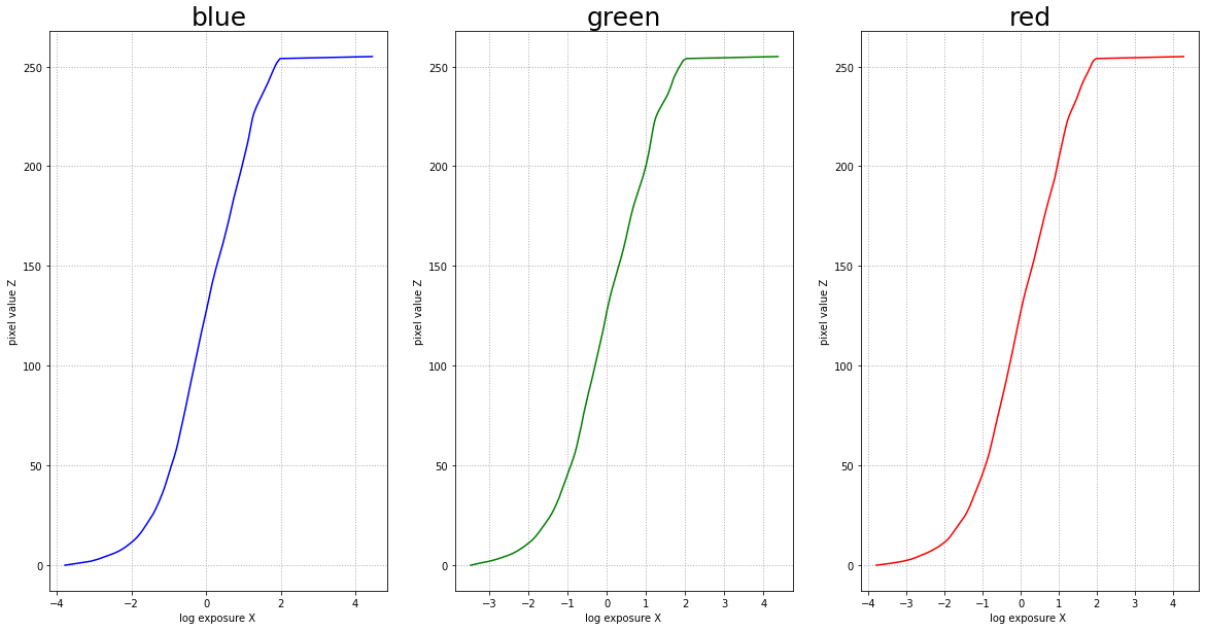
Robertson’s method

Robertson’s method uses all the pixels in the images to estimate the response curve, and uses an iterative optimization scheme to approach the response curve. In my implementation, I use a straight line(y = 28.33x + 170) to initialize the response curve. As a result, the response curve converges very quickly; it barely changes after second iterations.

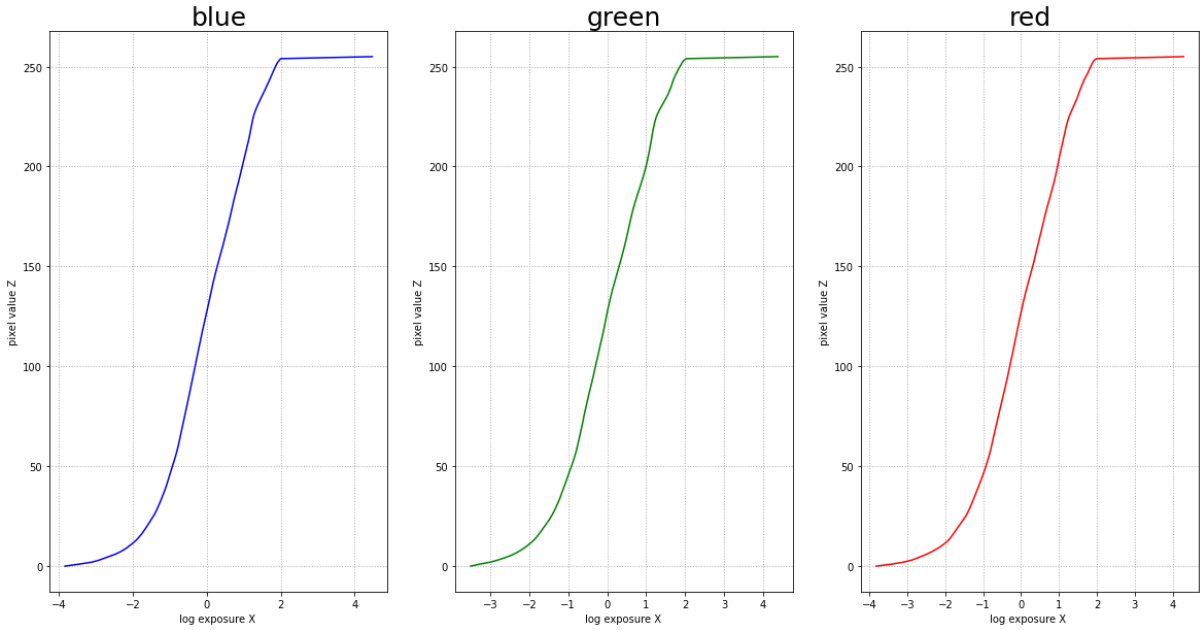
Response curve derived by Robertson’s method.(1st Iteration)



Response curve derived by Robertson’s method.(2nd Iteration)



Response curve derived by Robertson's method.(3rd Iteration)



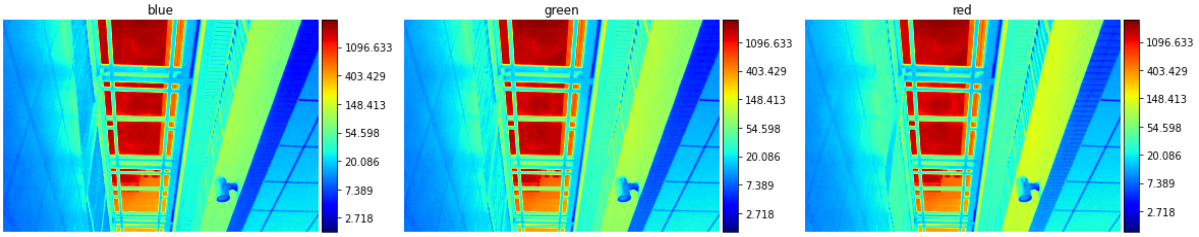
Comparision:

Paul Debevec's result and Robertson's result are mostly the same. Paul Debevec's curves are a little bumpy; but it can be eliminated if I use even more sample points. On the other hand, Robertson's curves are very smooth, since it directly uses all the pixels to do the optimization, its result is more robust and smoother.

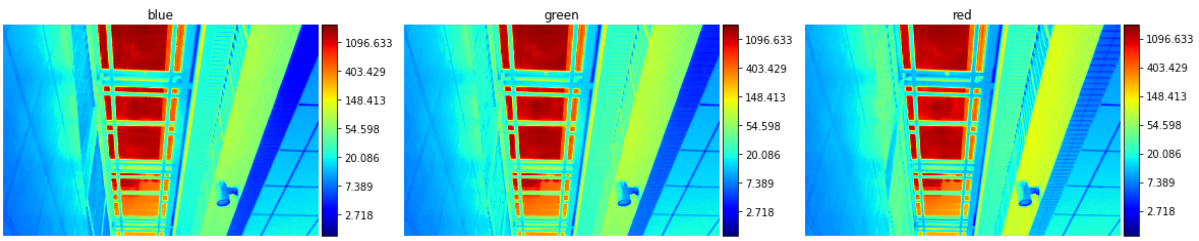
The other difference between the two methods is the values that are near 255 and 0. I used a triangular weight function to make extreme value less important when implementing Paul Debevec's method; however, in Robertson's method, I didn't use the weight function. As a result, Paul Debevec's curve is more stable at these extreme values and Robertson's curve is sometimes jumpy at these values.

Recovered Radiance Map

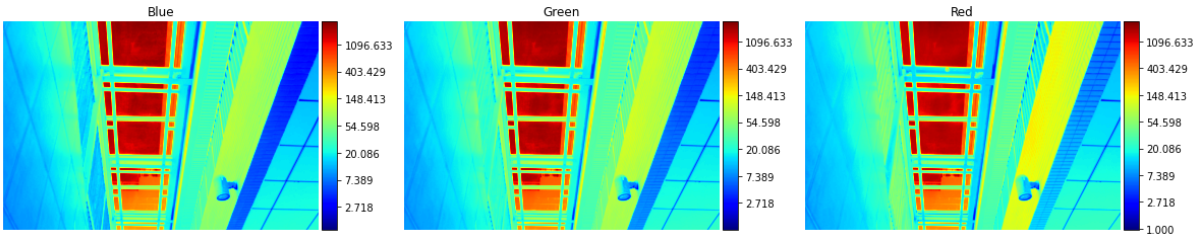
Paul Debevec's method recovered radiance map(My Implementation)



Robertson's method recovered radiance map(My Implementation)



Opencv's Paul Debevec method recovered radiance map



The red part of the radiance map represents the high energy part of the scene; in this case, it's the sky. The energy of the bright part is very high, since the camera almost shot directly to the sun.
The yellow and orange parts are the scene with less energy, which usually means they're reflecting the sunlight. And the blue part shows the dark or shadow in the scene, representing the least energetic part.
Comparing Paul Debevec's results and Robertson's results, they're almost the same. Can't see any visually major difference in terms of these results.

Tone Mapping

I implemented Reinhard's tone-mapping method. Figure below shows the change of tone when tuning the high/low key parameter. When 'a' is set to a higher value, the picture looks brighter, vice versa. We can see that through tone-mapping, image's details in bright and dark parts are both preserved in tone-mapped image.

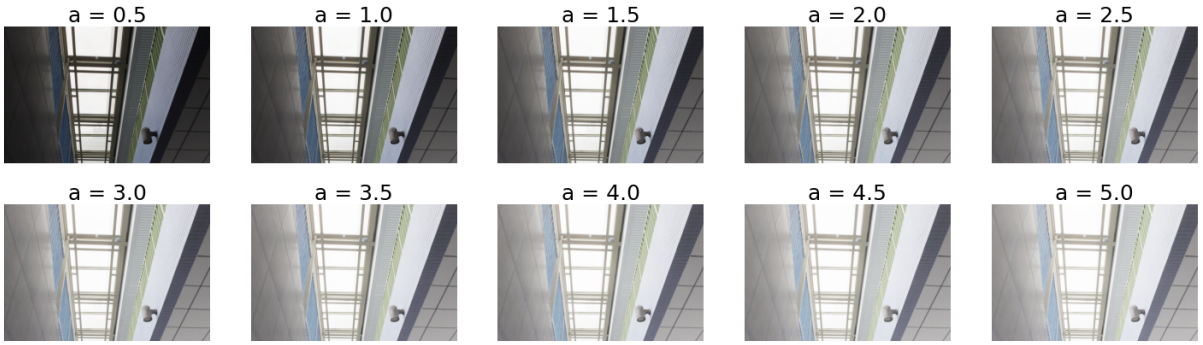
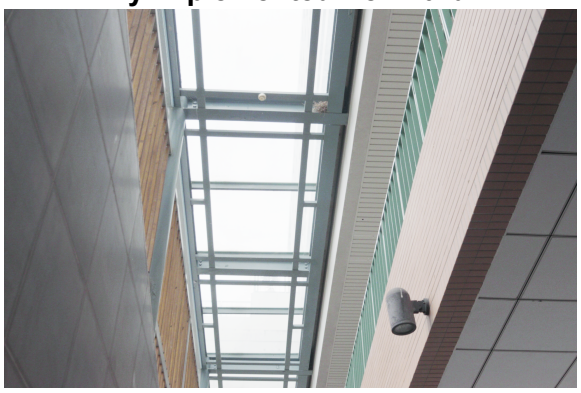
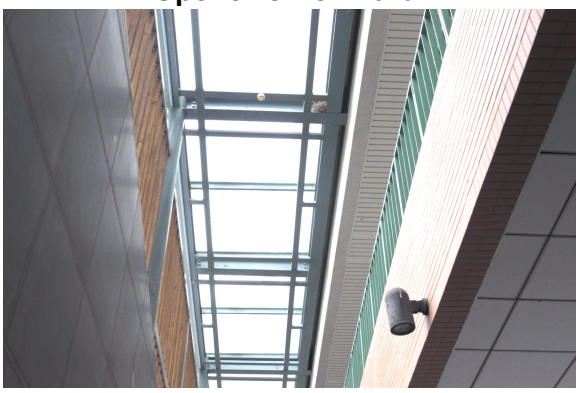


Figure below shows the comparison between different tone-mapping methods. All of these methods seem to produce good quality results. As to mention, all these tone mapping methods have lots of parameters to tune, this makes comparison even harder since these differences might be due to the different parameters set. Personally, I think Drago's method presented the best image, since it has the strongest contrast and I feel that It has more details inside the image.

My Implemented Reinhard	Opencv's Reinhard
	
Opencv's Drago	**Opencv's Mantiuk**