VFX Project 1 High Dynamic Range Imaging

B06902072 李謙 & B06902074 柯宏穎

Note1: We have taken two set of pictures and have shared them with another group(B06902016 & B06902066). Note2: More detail about how to execute can reference the README file in github repo

To-do List

1. Taking photographs (O)
2. Write a program to assemble an HDR image (O)
    - Bonus: Other HDR creation method -- Robertson's algorithm[1]
3. Develop radiance map using tone mapping (O)

Method for Recovering HDR Image and Tone-mapping

We used $Robertson$'s algorithm to get the response recovering curve and the radiance map. This algorithm finds better E and g iteratively, fixing E to update g, then fixing g and update E. We have run about 20 iterations to make these values converge. We used Dodging and Burning techniques to accomplish tone-mapping task, with only global operators applied. We also tried adding Gamma Correction on the graph, but found $\gamma = 1$ is still the best between different γ values. To save our results to HDR-formatted files, we take `pyexr` [2] package as our reference.

Results

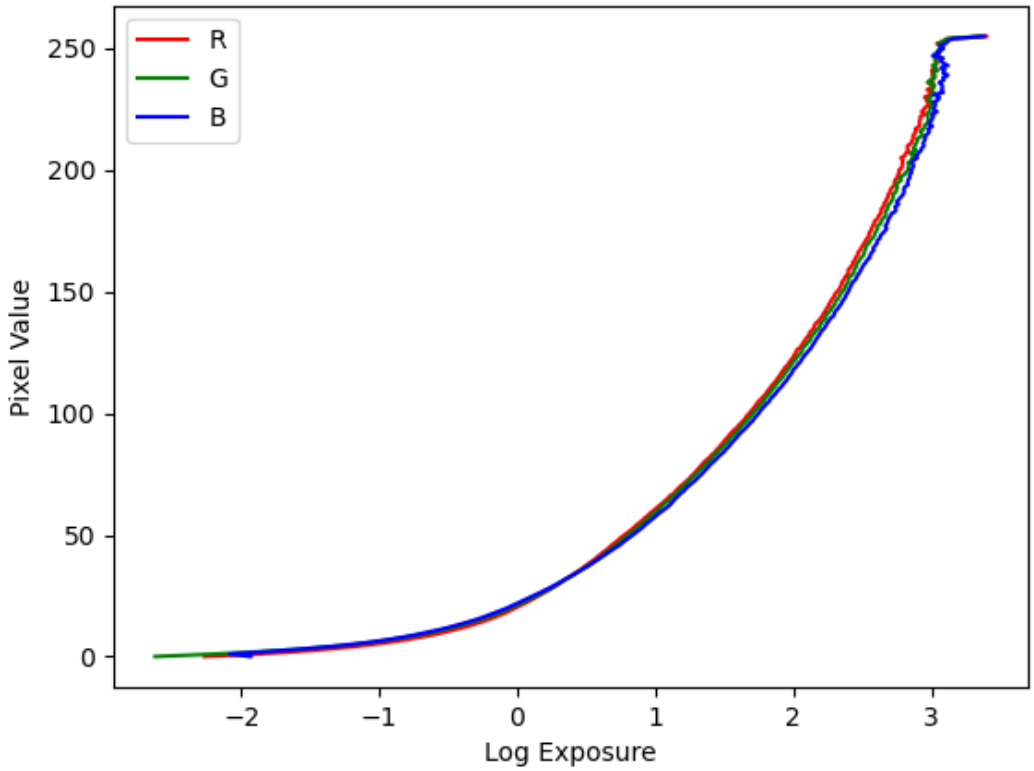
We have run our processing program on two sets of photos. We represent our results respectively below.

1.

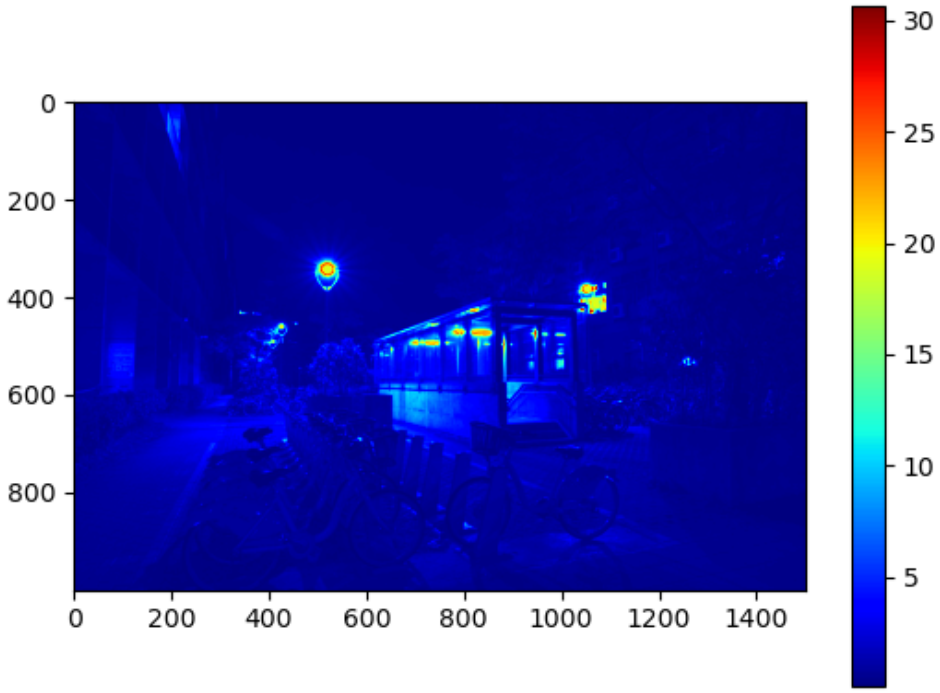
Original photos taken:

Recovered HDR image:

Curve:



Radiance map:

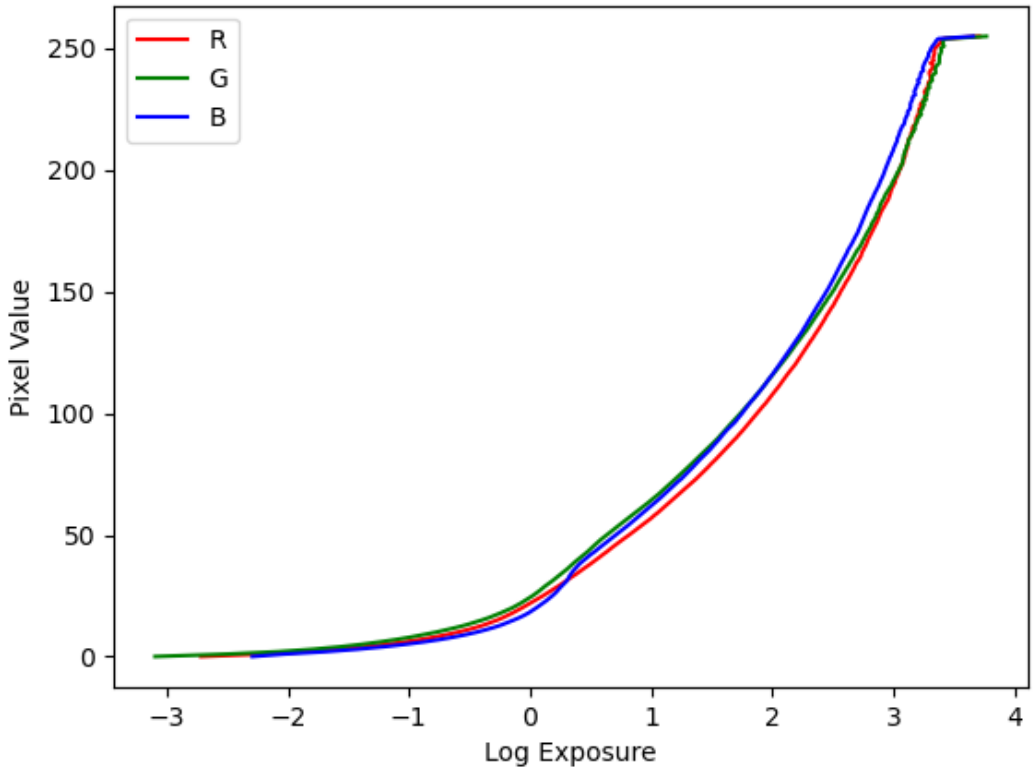


2.

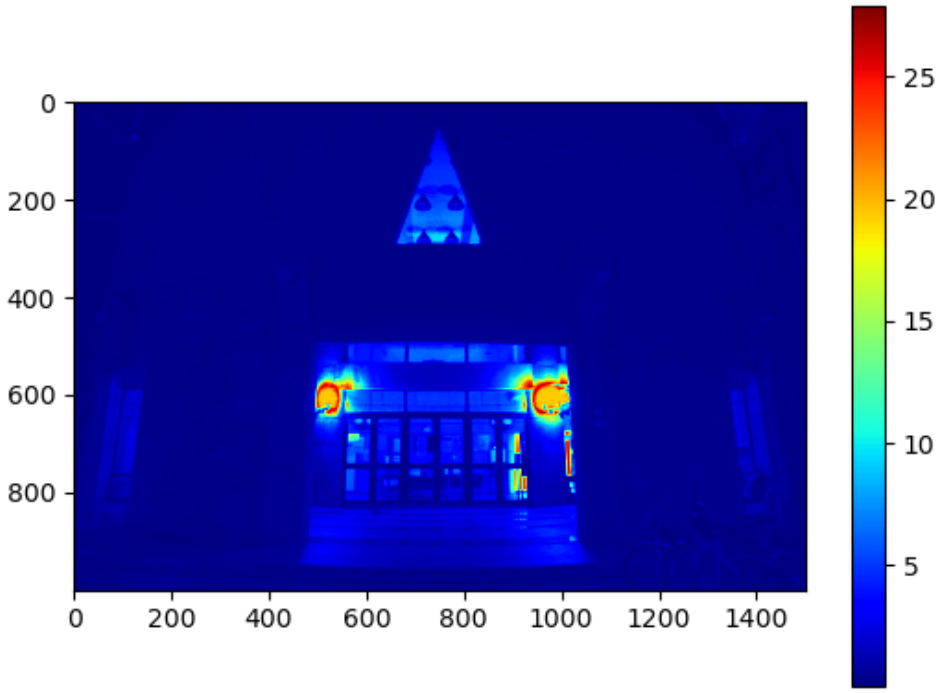
Original photos taken:

HDR image:

Curve:



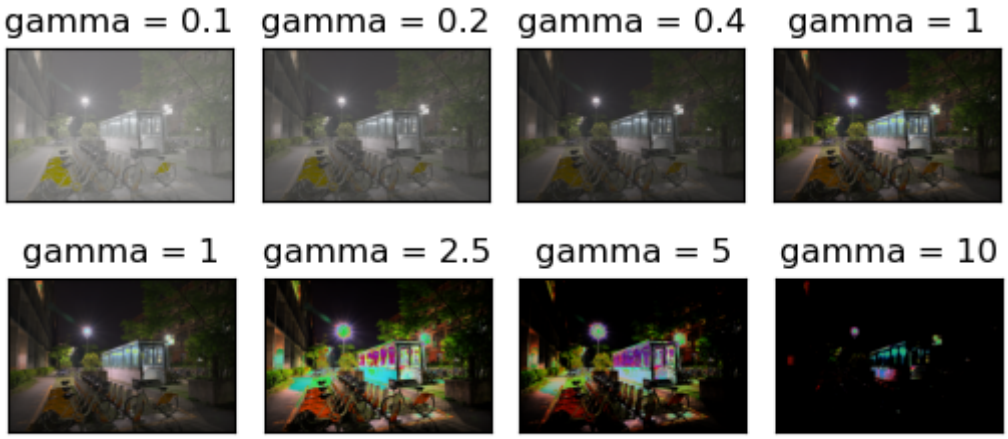
Radiance map:



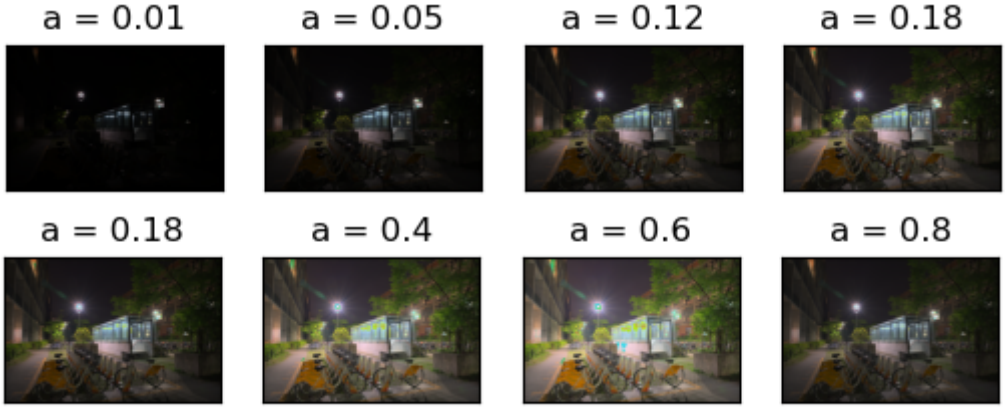
We think the first image is better-presented after HDR-imaging than the second one. One reason is that the first image has more details(or feature points) to observe. Second, it is generally brighter than the second image. Third, there are more objects in it, making the whole scene comparable between pre and post HDR-imaging.

Experience:

1. Different γ in tone mapping:



2. Different a in tone mapping:



Other Things to Share

This is the first time we tried to deal with raw images, which is different to . jpg or . png images. These images can't be read into program with normal methods in libraries. And there are lots of color distortions between these images read by different tools. We tried to recover them in different ways after they are read into programs, and have leart more with white balance, brightness, and so on, since we need to modify the parameters with these read-in functions.

Instead of solving linear system to find the optimal solution, we used $Robertson$'s algorithm to get the recovering curve. This algorithm need to traverse all of pixels in the picture and iterate several times to make it converge. In this case, the effectiveness is very important for the algorithm. We have used $numpy$ function, multi-thread, and list comprehension to speed up the process. By adding these tools and techniques, we have reduced the running time at least 90%.

Reference

1. S. Borman M. A. Robertson and R. L. Stevenson, Estimation-theoretic approach to dynamic rangeenhancement using multiple exposures, J. Electronic Imaging, 12 (2003), pp. 219–228.
2. https://github.com/tvogels/pyexr
3. http://users.eecs.northwestern.edu/~ollie/eecs395/HW4/HW4.htm
4. https://blog.csdn.net/u010165147/article/details/85232710