

I. Quick Usage

1. `python .\HDR.py --alignment=True --path=./dataset --index=0`
path 為還沒對齊的 image dataset
2. `python .\HDR.py --lambda_=10 --path=./alignment_result --index=0 --time=explosure_time.txt --gamma=1.5`
path 為對齊過後的 image dataset

II. Code Work

1. Image Alignment

- 實作課堂上講述的 MTB 演算法，一開始輸入要求選擇第 index 張的相片作為基準圖(base)，以便 alignment
- 輸入的圖皆以灰階讀取，取得各個 image 的 median pixel value，並製作 binary threshold image
- 我以間隔 1,2,4,8,16,32 取點去縮小各 binary threshold image 1,2,4,8,16,32 倍
- 用 numpy 中的 logical_or 去產生 base image median 正負值 10 的 exclusion mask，目的是減少干擾帶來的影響
- 並用上述的手法將 exclusion mask 縮小 1,2,4,8,16,32 倍
- 由最小到最大順序將各個影像用 cv2 中的 wrapAffine 去做 9 個方向的位移並與相同 size 的 base image 去做 xor 後取 exclusion mask，然後只取中間區塊計算差異，求得最小值後，增加位移量並進行下一輪

2. HDR Reconstruction

- 產生對圖 sample 的 matrix，再對圖像位置平均取點
- 使用的 weighted function: `return 256-value if value>127 else 1+value`，與課堂講述得有點不同，這裡是為了不讓有任何一個 pixel 的權重等於零
- 之後用 Debevec's method 還原 RGB 三條 response curve
- 並用 response curve 對每個 pixel 加權平均建出 ln radiance map
- 再對 ln radiance map 做 exponential 建構 radiance map，即為 HDR image，並存成 .hdr 檔

3. Tone mapping

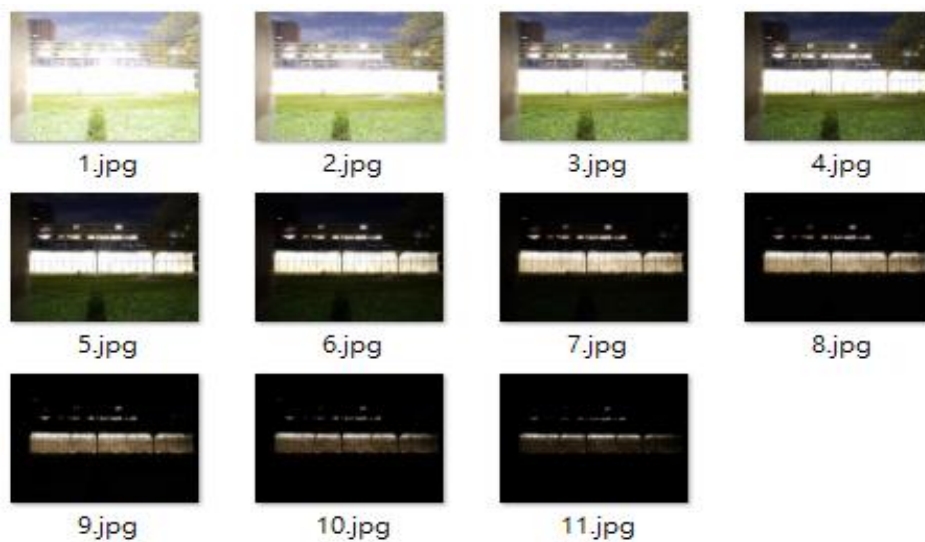
- 實作上課講述的 global tone mapping，設 $\delta = 1e-6$, $a = 0.5$ ，先對 Radiance map 加上 δ 後取 log 再取 mean 再做 exponential 得到 L_{bar} ，將 $radiance\ map * (a/L_{bar})$ 得到 L_m ，設 L_w 為 L_m 的最大值，最後 $L_d = L_m * (1 + (L_m / (L_w * L_w))) / (1 + L_m)$ ，得到一張 LDR image
- 再對還原後的 LDR image 進行 gamma correction，設 $\gamma = 1.5$ ，得到一張還不錯的圖

III. Data Set

1. Test data set



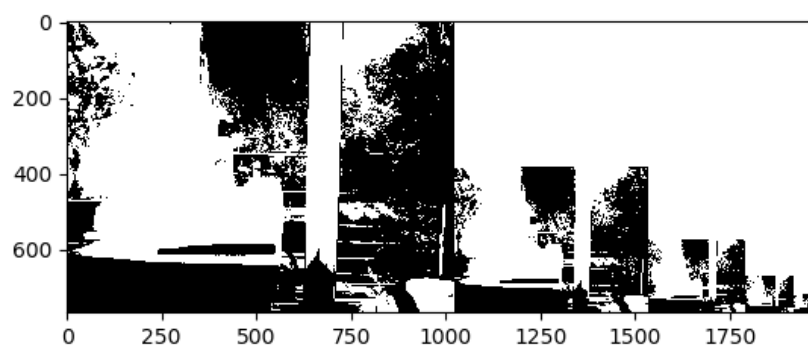
2. My data set

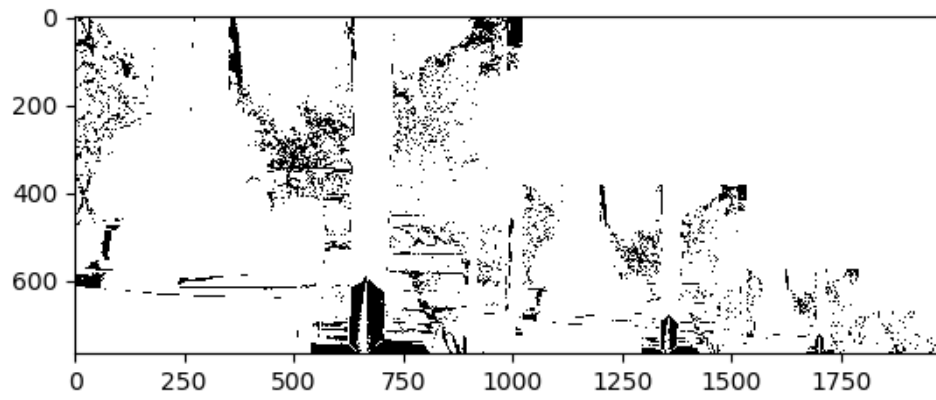


IV. Graph compare

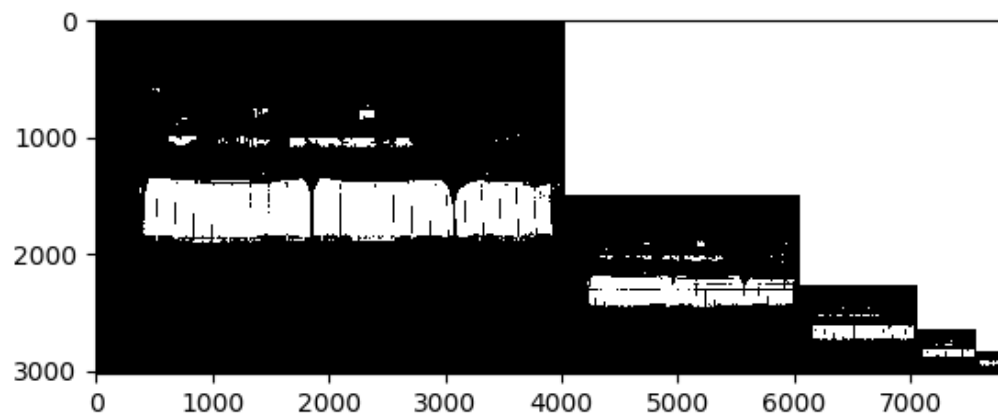
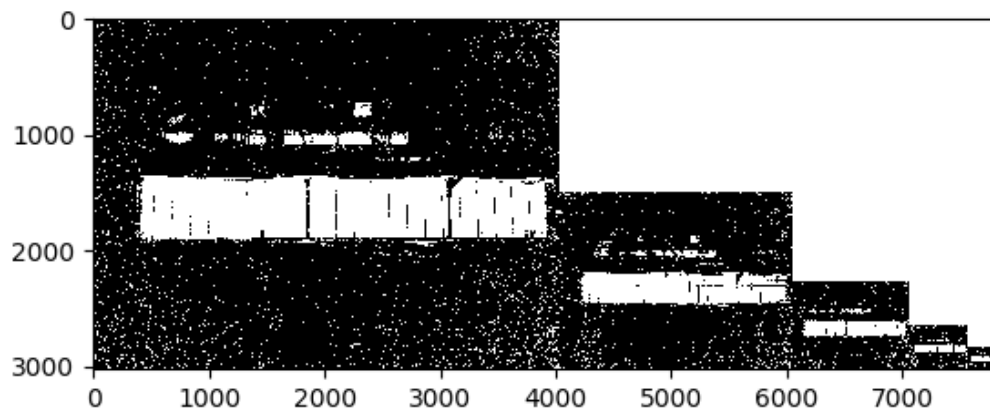
1. binary threshold image and mask

- Test



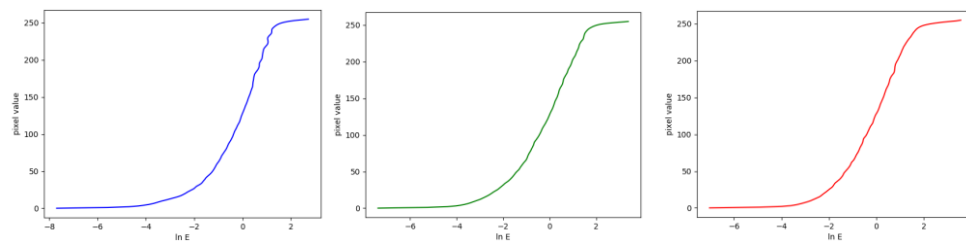


- My

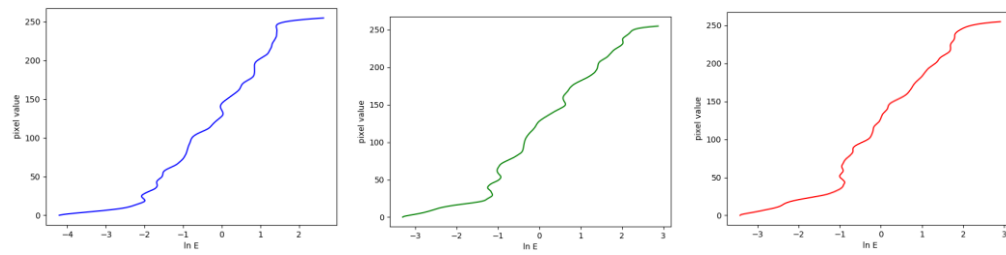


2. Response Curve

-- Test

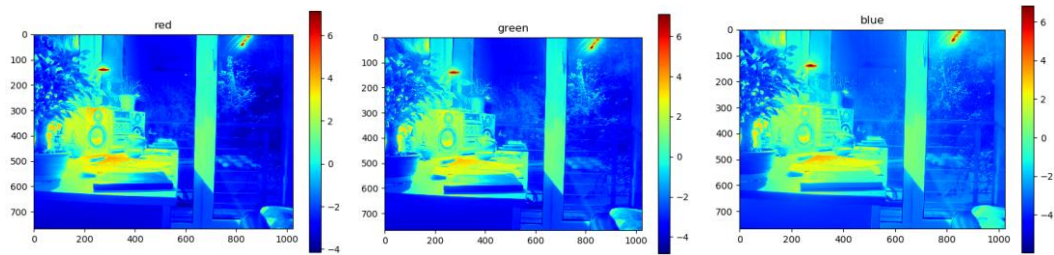


- My

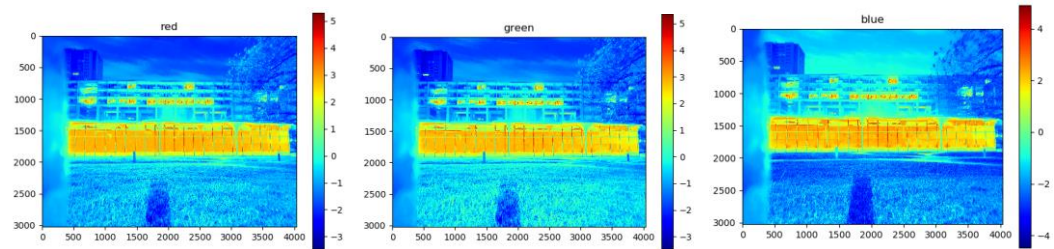


3. Ln radiance map

- Test

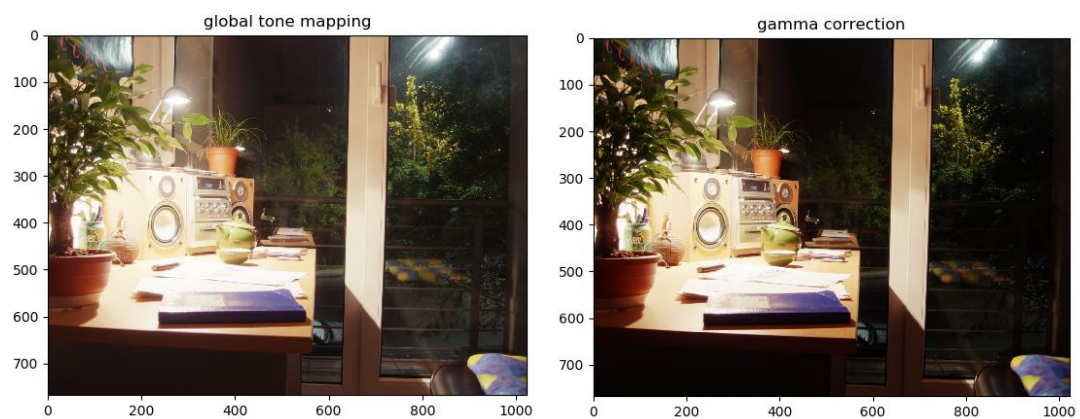


- My



V. Result

- Test



- My

