

VFX HW1

資工三 B09902117 林榮盛

資工三 B09902108 石容居

reference 1:<https://github.com/bchao1/high-dynamic-range-imaging>

reference 2:<https://github.com/JCly-rikiu/HDR>

reference 3:[Photographic Tone Reproduction for Digital Images](#)

作業內容

本次作業通過將多張曝光時間不同的照片經計算後，生成一張 hdr 影像，並通過 tone mapping 演算法，將此 hdr 影像轉為效果更佳的 ldr 影像

實作項目

Alignment

Generate HDR image via Paul Debevec's method

Photographic global tone mapping

Photographic local tone mapping

Adaptive logarithmic mapping

Alignment

在此步驟中，algorithm 會根據 input 的 parameter 選定 standard image，其他 image 都會以 standard image 作為基準進行 alignment。

由於 standard image 會被使用多次，因此先將 standard image 縮小成各種比例，EX $2^1, 2^2, \dots, 2^{\text{depth}}$ ，再根據每種比例的 standard image 做出對應的 binary image 與 mask 並存在 table 裡面，有助於加速 algorithm 的運算。再來就是依照上課教授教講的。將需要 align 的 image I 縮小產生出 $I^1, I^2, \dots, I^{\text{depth}}, I^k$ 為縮小 2^k 的 I ，讓 $N = \text{depth}$ 。

Step 1：對 I^N 產生出九個 shift 方向的 binary image 與 mask，透過取 AND 與 XOR 來算出每個方向與 standard image 的 difference，取 difference 最小的 shift 值 S_{\min}

Step 2：將 I^{N-1} shift $S_{\min} \times 2$ ，然後讓 $N = N - 1$ 重新做 step 1 與 step 2 直到 $N < 0$ 最後 shift 完的 image 就是 align 完成的 image

Paul Debevec's method

在此步驟中，我們須先 sample 出多個點，因此選擇在 $1/8$ 長(寬)至 $7/8$ 長(寬)範圍中隨機取 50 個點，以避免選取邊界作為取樣點

根據公式：

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

所求為 O 的最小值，可視為解 $Ax = b$ 的 least square solution

後續將 A 矩陣填入適當值，結合填好的向量 b 求出最終結果，求 least square solution 時，我們以 numpy 的 module numpy.linalg 中 numpy.linalg.lstsq 作為解 solution 的方式

分別對三個 channel 解出 x 後，x 的前 256 項元素即為真實對應 hdr image 的數值，接下來回填每個 pixel 真實對應的值

現考慮其中一個 pixel 的一個 channel:

根據公式

$$\ln E_i = \frac{\sum_{j=1}^P w(z_{ij})(g(z_{ij}) - \ln \Delta t_j)}{\sum_{j=1}^P w(z_{ij})}$$

將結果再取 exponential 後可得實際應填入的值，對圖片中所有 pixel 計算後即可得到最終 hdr image

本次作業中所取 lambda 為 20，此為實際試驗和參考 reference 1 後的結果，詳細原因會在下方結果討論說明

根據投影片中，weight function 如下

$$w(z) = \begin{cases} z - Z_{min} & \text{if } z \leq Z_{mid} \\ Z_{max} - z & \text{otherwise} \end{cases}$$

$$\text{for } Z_{min} = 0, Z_{max} = 255, Z_{mid} = 127$$

但可發現在計算 $\ln E_i$ 時，若所有 z 皆為 0 或 255，則可能導致分母部分和分子部分皆為 0，因此我們在這裡有做出細微調整

$$w(z) = \begin{cases} z - Z_{min} + 0.001 & \text{if } z \leq Z_{mid} \\ Z_{max} - z + 0.001 & \text{otherwise} \end{cases}$$

$$\text{for } Z_{min} = 0, Z_{max} = 255, Z_{mid} = 127$$

以避免最終出現 divided by zero

Photographic global tone mapping

本項目實作 Reinhard tone mapping 的 global 版本

首先需先計算 L_w ，我們採用 B:G:R = 0.06:0.67:0.27 的比例計算 L_w

$$L_w = 0.06B_w + 0.67G_w + 0.27R_w$$

計算出 L_w 後，加上 delta(1e-6)，取 log

對每個 pixel 作相同事情後，將所有計算的值相加，公式如下

$$sum_{log} = \frac{1}{N} \sum_{x,y} \log(\delta + L_w(x,y))$$

其中 N 為所有 pixel 數，將其取 exponential 得到

$$\bar{L}_w = e^{sum_{log}}$$

接著計算所有 pixel 的 L_m

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

再來得到 L_d

$$L_d(x, y) = \frac{L_m(x, y)(1 + L_m(x, y)/L_{white}^2)}{1 + L_m(x, y)}$$

最終填入每個 pixel 的各個 channel 值

$$B_d = \frac{L_d}{L_w} B_w, G_d = \frac{L_d}{L_w} G_w, R_d = \frac{L_d}{L_w} R_w$$

本次作業中， a 值設定為 0.35， L_{white} 值為 1.1，此為經由多次測試後所得，將在下方結果討論詳細敘述

Photographic local tone mapping

根據公式

$$\begin{aligned} L_s^{blur}(x, y) &= L_m(x, y) \otimes G_s(x, y) \\ V_s(x, y) &= \frac{L_s^{blur}(x, y) - L_{s+1}^{blur}(x, y)}{2^\phi a/s^2 + L_s^{blur}} \\ s_{max} : |V_s(x, y)| < \epsilon \end{aligned}$$

根據 reference 2 和 reference 3，我們用較簡易的方式求出 L_s^{blur} ，具體細節如下
首先將 s 定為 0.35，對整張圖做 GaussianBlur，存下此圖

再將 s 乘上 $\alpha=1.6$ 後，再次對整張圖做 GaussianBlur，存下此圖

再次將 s 乘上 α 後，再做 Blur，重複此步驟直到有 16 張圖為止

現考慮一個 pixel，對這 16 張圖算出 15 個 V_s (因要取 $L_s^{blur} - L_{s+1}^{blur}$ ，所以只有 15 個)，
找出 $V_s(x, y) < \epsilon$ 中 s 最大的 $L_s^{blur}(x, y)$ 作為 $L_s(x, y)$

最後計算 L_d

$$L_d = \frac{L_m(x, y)(1 + L_m(x, y)/L_{white}^2)}{1 + L_s(x, y)}$$

其中 $L_m(x, y)$, L_{white} 和 global 中的數值相同，最後和 global 進行相同處理即可

本次作業中， a 為 0.18， ϵ 為 0.05， ϕ 為 6，kernel size 為 5，部分來自於原始論文參數，部分為實際測試後考量參數

Adaptive logarithmic mapping

Step 1：先將 HDR image 轉成 XYZ type，再算出 Y channel 的最大亮度 Y_{wmax} 與平均 Y_{wavg}

Step 2：套入公式算出每個 pixel 的 Y_d ，其中 L_{dmax} 為螢幕的亮度顯示能力這裡依照 paper 設為 100， b 為 bias 依照 paper 的實驗這裡也設為 0.85， Y_w 為此 pixel 的 Y channel 亮度。

$$\begin{aligned} Y_d &= \frac{L_{dmax} \times 100}{\log_{10}(Y_{wmax} + 1)} \times \frac{\log(Y_w + 1)}{\log(2 + ((\frac{Y_w}{Y_{dmax}})^P) \times 8)} \\ P &= \frac{\log(b)}{\log(0.5)} \end{aligned}$$

Step 3：依照 Y_d 填入每個 pixel 的各個 channel 值

$$X_d = \frac{Y_d}{Y_w} X_w, Z_d = \frac{Y_d}{Y_w} Z_w$$

Step 4：做 gamma correction，這裡實作兩種 type。做完 gamma correction 後將 image 從 XYZ type 轉成 BGR type 再透過 numpy 內建的 function ”np.clip()” 轉成 uint8

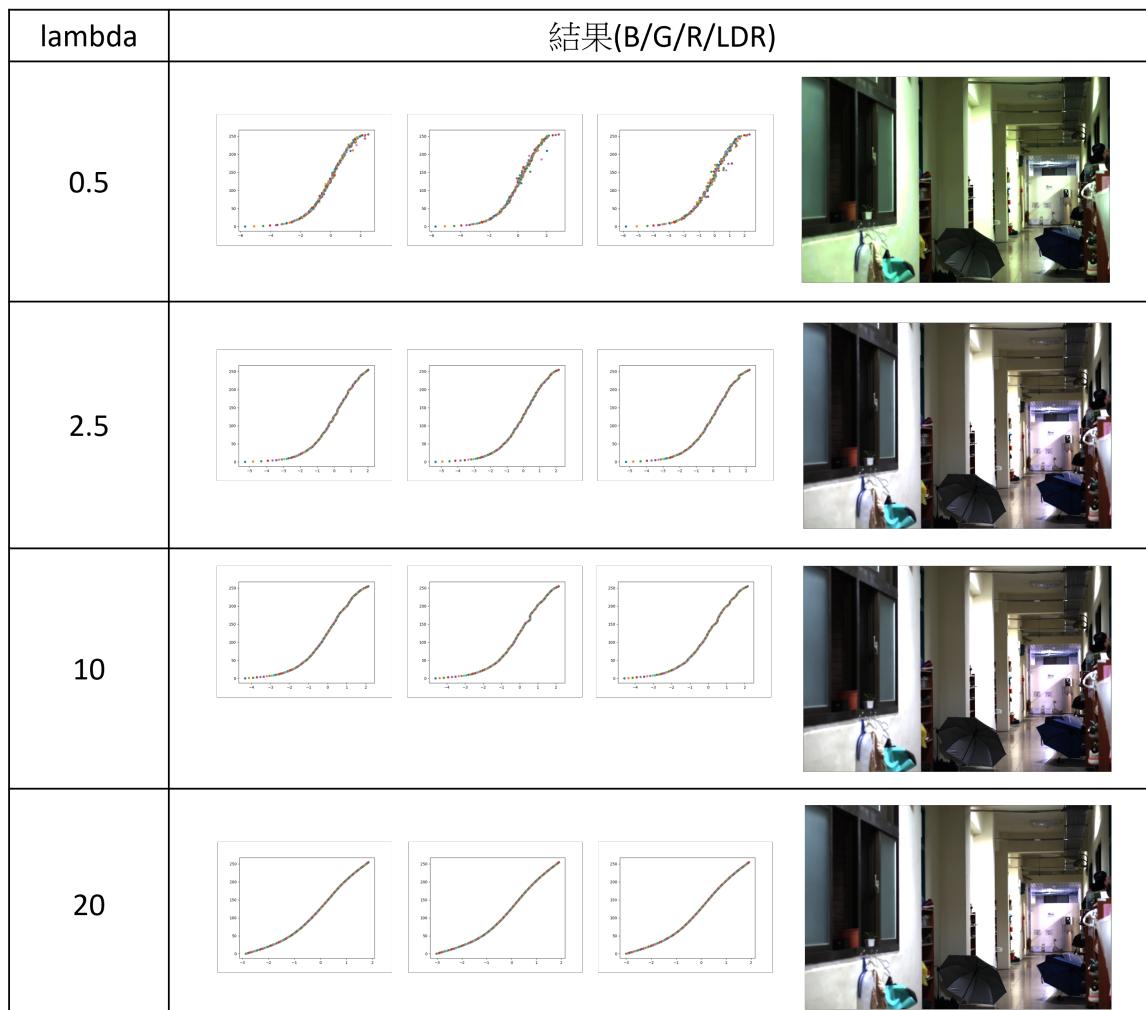
Type 1 :

$$L' = \begin{cases} L \times 4.5 & \text{if } L \leq 0.018 \\ 1.055 \times L^{\frac{1}{24}} + 0.055 & \text{otherwise} \end{cases}$$

Type 2 :

$$L' = \begin{cases} L \times 12.92 & \text{if } L \leq 0.0031308 \\ 1.099 \times L^{0.8} - 0.099 & \text{otherwise} \end{cases}$$

Paul Debevec's method 中 lambda 對結果影響



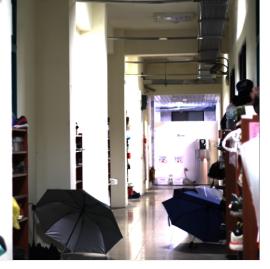
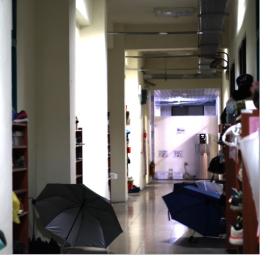
就 BGR 曲線而言，增加 lambda 值使曲線變得較為平滑，而在圖形上的效果顯示，lambda 值過低有機會使圖片出現色澤異常的情況，因此我們選擇 lambda=20 作為本次作業的參數

Photographic tone mapping method 中 key 對結果影響

key	結果(LDR global/local)	
0.1		
0.2		
0.35		
0.72		
0.9		

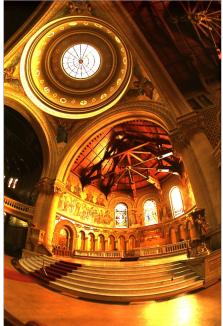
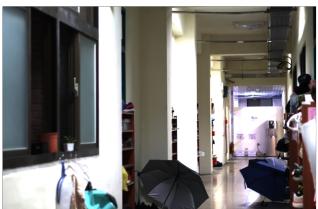
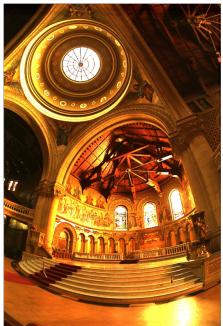
key 值影響整體亮度，由圖形可以看到，key 值過低將使整體亮度偏暗，但 key 值過高使圖片過亮，我們選擇 0.35，使整體亮度不會過亮或過暗

Photographic tone mapping method 中 L_{white} 對結果影響

L_{white}	結果(LDR global/local)			
0.5				
0.8				
1.1				
1.5				

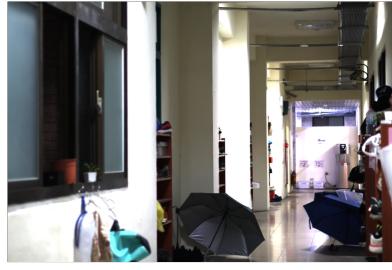
L_{white} 也影響亮度，但影響幅度偏向局部亮度，若數值過高則導致局部過亮而細節不清，過低可能導致無法凸顯亮處，此處選擇 1.1 作為 L_{white} 值

Photographic local tone mapping method 中 ϕ 對結果影響

Phi	結果(LDR local)		
2			
4			
6			
8			
10			

ϕ 值影響圖片銳利度，數值越大則越銳利，此處以 ϕ 為 6 作為最終選擇

對比 Photographic tone mapping method global 和 local 效果

LDR global	LDR local
	
	
	

對比 global 和 local 可以發現，local 對細節的處理相對好，證明 local 確實有較佳的效果

對比有無 alignment 的效果

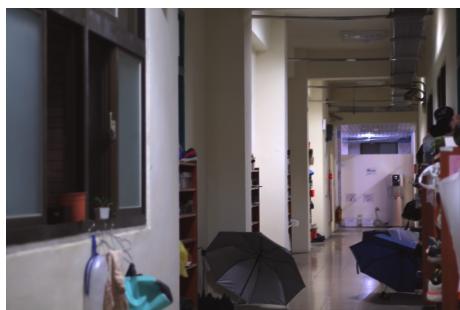


Figure 1: Without alignment



Figure 2: Apply alignment

由於我們在拍照時有故意做一點小小的移動，因此可以看出 figure 1 的飲水機附近非常模糊，而 figure 2 的並無此現象，由此可以看出 alignment 是真的有助於 HDR image 的產生

對比兩種 gamma correction 的效果



Figure 3: type 1



Figure 4: type 2

可以看出 type 1 還原後的 LDR 非常亮，而 type 2 的較為適中，我認為這就要依照各個圖片的特性來挑選合適的 gamma correction。