

Digital Visual Effects hw1 Report

R06944059 林昀宣 R06922033 陳映紅

1. Image Setting

出外拍攝照片時選定的地點，有兩組照片在寶藏巖（圖一、圖二），另外兩組在系館（圖三、圖四），主要使用寶藏巖其中一組照片（圖一）作為呈現，其他三組為程式除錯時使用，以下會以圖一當作介紹。



（圖一）



（圖二）



（圖三）



（圖四）

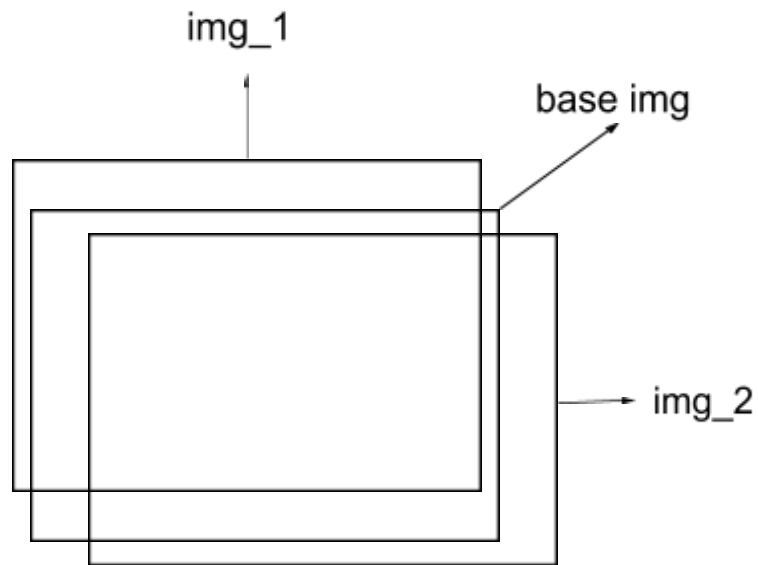
在拍照時我們先架設腳架，將相機設定為auto的模式，找到一組可以參考的IOS、光圈和快門速度的設定後，再調整不同的快門速度取得不同曝光時間下的照片。ISO設為800、光圈設為f/18.2，快門時間介於[1/50, 1/1600]，總共拍出15張照片，畫素為3456*5184，並同時存有CR2和JPG兩種格式，實作部份僅用到JPG，以上照片皆為快門時間為1/400得到的照片。

2. Image Alignment

a. 實作方法：

從這15張照片img_1~img_15中，依照快門時間選定排序過後，選定中間的那張img_8照片當作基準base_img，將其餘的14張照片img_i (i != 8) 分別與base_img做比對，找出img_i的水平位移h_move_i以及垂直位移v_move_i，使得該照片img_i相對於base_img可以達到最好的alignment效果

，找到全部照片的 h_move_i 以及 v_move_i 後，取出全部照片交集的位置，因此輸出的照片畫素會小於 $3456*5184$ ，實驗結果為 $3412*5159$ 。



在計算 h_move_i 和 v_move_i 的時候，使用到Median Threshold Bitmap (MTB) alignment technique，先算出每張圖的grayscale img_gray_i ，再依照grayscale的平均找出閾值 $thres_i$ ，高於 $thres_i$ 的設為1，低於 $thres_i$ 的設為0，建構出bitmap（如下圖五~圖八） img_mtb_i ，由圖可見，不管快門時間如何，照片的輪廓大致上呈現一致。



（圖五）快門時間 1/ 50



（圖六）快門時間 1/ 160



（圖七）快門時間 1/ 640



（圖八）快門時間 1/ 1600

img_mtb_i當中，一樣取第八張img_mtb_8當作基準base_img_mtb，每次對 (base_img_mtb, img_mtb_i) , $i=[1, 15]$, $i \neq 8$ 這個pair作alignment，先分別對兩張照片作五種縮放，每次縮放的比例為0.5，對於原本的照片會有的縮放比例為1、0.5、0.25、0.125、0.0625、0.03125的六張照片img_mtb_i, $j=[1, 6]$ ，先從base_img_mtb和img_mtb_i中各選最小的照片base_img_mtb_6和img_mtb_i_6進行比較，每次將img_mtb_i_6作九宮格位移，包括 (-1, -1)、(-1, 0)、(-1, 1)、(0, -1)、(0, 0)、(0, 1)、(1, -1)、(1, 0)、(1, 1) 9個方向的位移後與base_img_mtb_6，選定其中一個位移使得alignment的效果最佳，將此位移量拿來當base_img_mtb_5和img_mtb_i_5的初始位置進行一樣的操作，循序漸進算出base_img_i_1和img_mtb_i_1 (即原圖大小) 的位移量當作img_mtb_i的h_move_i和v_move_i。

b. 實驗結果：

alignment後的照片 (圖九~圖十二)，以這樣的呈現方式很難以肉眼分辨出實際alignment的效果如何，因此拿沒有alignment (圖十三) 和有alignment (圖十四) 的的照片分別做出的HDR照片經tone mapping後的結果來呈現，由於我們的照片在拍攝時均有使用腳架固定，但依舊會有些微差距，從實驗過程中看到因為alignment產生的位移量最多在大概10個pixel左右，因此經過alignment後，從有點模糊變成較清晰的照片可知，alignment對於建構出HDR照片的重要性。



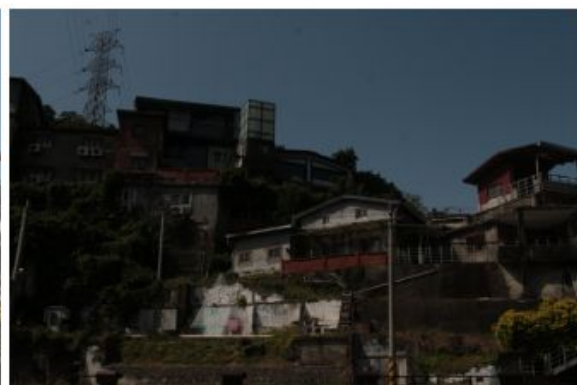
(圖九) 快門時間 1/ 50



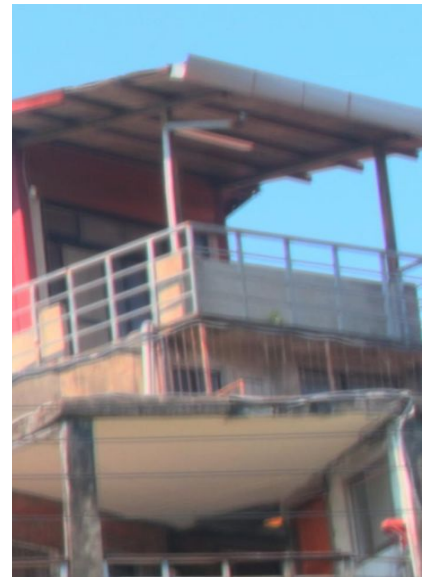
(圖十) 快門時間 1/ 160



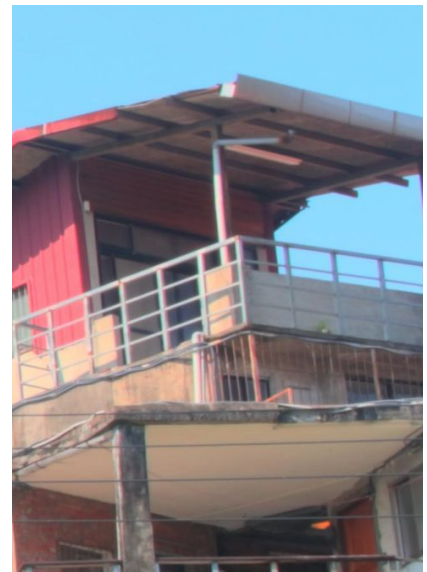
(圖十一) 快門時間 1/ 640



(圖十二) 快門時間 1/ 1600



(圖十三)



(圖十四)

3. Assemble HDR

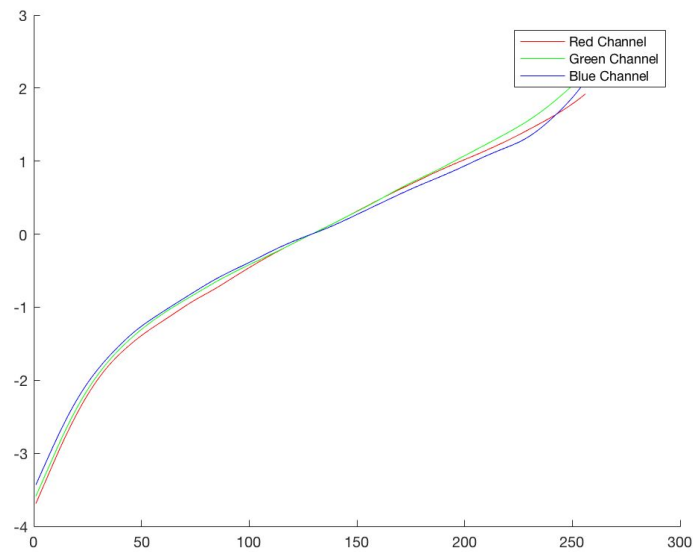
a. 實作方法：

我們實作Debevec提出的方法，整個流程可分為以下三部：重建g curve、重建InE和移除鬼影。

b. 重建g curve：

我們於每組圖中隨機抽樣100個點，以第一組圖為例，第一組圖有十五張圖，因此 $P = 15$ ， $N = 100$ 。將抽樣點分為RGB 3個channel做運算，利用投影片第45頁提供的gsolve函式重建A和B矩陣後，解出3個channel各自的g

curve。我們的g curve如下：



c. 重建lnE：

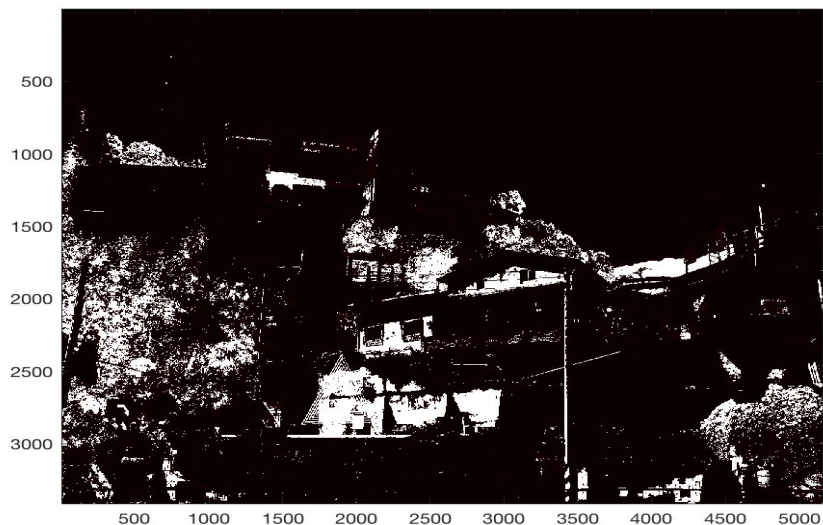
得到g curve後即可以以下公式得到整個畫面中每個點的平均lnE：

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

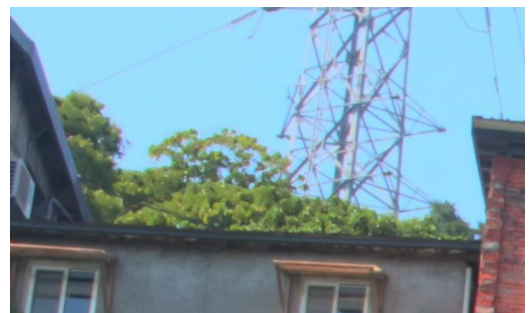
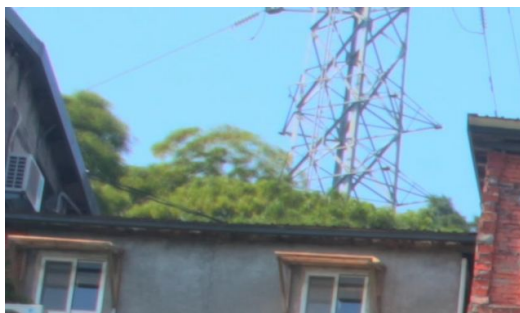
做exponential後即得到每個點的sensor irradiance，利用MATLAB內建函式hdrwrite()即可將結果存為.hdr檔。

d. 移除鬼影

拍攝地點有風，造成景物的晃動，所以將.hdr檔做tone mapping後可看到許多鬼影，因此我們實作ghost removal。首先算出每張照片的原始lnE（即 $g(Z) - \ln t$ ），並算出每個點的原始lnE的variance。我們將variance大於0.08的點視為包含移動物件的點，並標記這些點，標記結果如下：



接下來如上點計算平均InE，並將標記點的平均InE替換成第七張圖的原始InE，即可移除鬼影。我們擷取以下幾個移動物件來展示結果：



4. Tone Mapping

一開始我們使用MATLAB內建的tonemap()函式，結果如下：



如上圖，tonemap()確實可將HDR影像轉為LDR影像，但是色調與實際景象有很大的差距。因此我們實作photographic tone reproduction的global operator方法，實作結果色調偏紅但明亮許多，較tonemap()結果貼近實際景象。結果如下：



註：有另一組同學使用同一台相機拍攝，最終結果的色調也偏紅，因此我們推測和相機有關。

5. What I've learn

a. 林昀宣

我從這次的作業中，學習到的部份主要分為兩個部份，一個是Matlab操作，另一個是alignment的技巧，由於平時比較沒有機會接觸Matlab，操作起來格外生疏，那這也佔了這次作業滿多的時間，花時間看Youtube線上教學、Matlab官方文檔和許多Stack Overflow的文件，從一開始indexing從1開始和句尾加分號，matrices的操作，到imread和imwrite，以及寫完alignment後才發現的imtranslate，慢慢發現到許多好用的工具。另一部份，在alignment技巧上面，了解到MTB的作法比grayscale在找edge的時候有更佳的效果及更好的執行效率。在實驗過程中，也試著根據不同pixel的值用不同的weight來調整thres_i看能不能得到更好的結果，但實際上發現原本沒有weight的作法就已經達到很好的效果了。

b. 陳映紅

我依照演算法重建HDR照片和Tone Mapping時沒有遇到太大的問題，但仍然在實作中發現一些有趣的地方：

- i. 移除鬼影時發現RGB 3個channel的variance分佈有差異，若取threshold = 0.1，blue channel的variance高於threshold的pixel個數是另外兩個channel的幾十倍，並且涵蓋另外兩個channel高於threshold的所有點。
- ii. Global photographic tone mapping的參數key (a) 會因照片不同而有極大的變化，我的照片取1.5即可，其他組別的照片必須取20才夠亮。
- iii. 另一組同學與我們共用相機，最後實作結果的色調都偏紅（原始照片色調正常），而使用不同相機的組別則沒有這個問題，因此我推測重建HDR和tone mapping方法的表現因相機而異，或許這台相機比較適合其他我們沒有嘗試的方法。