

DigiVFX

HW1 HDR Imaging

B02901072 楊喬諳
B02901065 李洛曦

Description

程式碼包括

— myHDR.m

使用 `hdr = myHDR(ImageDir, format, algorithm, arg);`

`ImageDir` 為圖片檔資料夾位置

`format` 為圖片格式，如 `' .jpg'`，`' .png'`，`' .tif'`等

`algorithm`可為1或2

若為1，則使用Debevec's method

`arg` 為exposure time的vector

若為2，則使用Mitsunaga and Nayar's method

`arg` 為M 所假定的多項式最高次

`hdr` 回傳一組 圖片長x圖片寬x3的 `hdr radiance matrix`

會在程式碼中呼叫到`gsolve.m` `gsolve2.m` `myMTB.m`函式

— `gsolve.m`

Debevec's method 同講義中所介紹的一樣

— `gsolve2.m`

使用 `g = gsolve2.m(Z,M);` 實作上假定 `M = 10`

Mitsunaga and Nayar's method 假設為M次多項式並解之

經過反覆運算後，得出最吻合的係數 C_M

#reference : *<High Dynamic Range Imaging>* p.175開始

$$C_N = 1 - \sum_{n=0}^{N-1} C_n$$

The final $N \times N$ system can be written:

$$\begin{bmatrix} \sum_{q=1}^{Q-1} \sum_{p=1}^P d_{p,q,0} (d_{p,q,0} - d_{p,q,N}) & \dots & \sum_{q=1}^{Q-1} \sum_{p=1}^P d_{p,q,0} (d_{p,q,N-1} - d_{p,q,N}) \\ \dots & \dots & \dots \\ \sum_{q=1}^{Q-1} \sum_{p=1}^P d_{p,q,N-1} (d_{p,q,0} - d_{p,q,N}) & \dots & \sum_{q=1}^{Q-1} \sum_{p=1}^P d_{p,q,N-1} (d_{p,q,N-1} - d_{p,q,N}) \end{bmatrix}$$
$$\times \begin{bmatrix} C_0 \\ \dots \\ C_{N-1} \end{bmatrix} = \begin{bmatrix} - \sum_{q=1}^{Q-1} \sum_{p=1}^P d_{p,q,0} d_{p,q,N} \\ \dots \\ - \sum_{q=1}^{Q-1} \sum_{p=1}^P d_{p,q,N-1} d_{p,q,N} \end{bmatrix}$$

where

$$d_{p,q,n} = M_{p,q}^n - R_{q,q+1} M_{p,q+1}^n$$

— myMTB.m

使用 $X = \text{myMTB}(X);$

輸入一個 圖片數x圖片長x圖片寬x3 的pixel value matrix

回傳一個 已aligned的pixel value matrix

— myTonemap.m

我們使用的是講義上的Photographic Tone Reproduction

使用方式有兩種

- (1) $\text{rgb} = \text{myTonemap}(\text{hdr}, 'global', \text{key}, L_{\text{white}});$
hdr 為一組 圖片長x圖片寬x3的 hdr radiance matrix
可以是myHDR所回傳的值，或是內建的hdrread所回傳的值
rgb 為tonemap出來的結果 可用imshow(rgb)加以顯示
key 為下圖中所使用的"a"

User-specified; high key or low key

$$L_m(x, y) = \frac{a}{L_w} L_w(x, y) \quad L_d(x, y)$$

L_{white} 為下圖中所使用" L_{white} "，可填可不填
default值為整張圖片中原有最高的 L_m 值

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

- (2) $\text{rgb} = \text{myTonemap}(\text{hdr}, 'local', \text{key}, \text{phi}, e);$

$$V_s(x, y) = \frac{L_s^{\text{blur}}(x, y) - L_{s+1}^{\text{blur}}(x, y)}{2^\phi a/s^2 + L_s^{\text{blur}}}$$
$$s_{\text{max}} : |V_{s_{\text{max}}}(x, y)| < \varepsilon$$

phi 為上圖中的2的指數

e則為測量 s_{max} 的threshold

此處我們使用了fspecial('gaussian', hsize, sigma)

(以下還未有實際成效)

— myGhost.m

使用myGhost(X, threshold)

— mySobelEdge.m

使用 Y = mySobelEdge(X, threshold)

edge detection，下圖參照維基百科

$$\mathbf{G}_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} * \mathbf{A} \quad \text{and} \quad \mathbf{G}_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix} * \mathbf{A}$$

若兩個G的平方和大於threshold，則標記為邊界

— myImageVariance(X, Y);

Y 即是上式所計算出來的Y

計算X這P組圖片的Variance，並利用Y作為其每個點的權重

即為weighted variance

Image alignment -> myNTB.m

我們使用的是講義中所介紹的MTB alignment technique
先找到一張圖的中位數median，將threshold設為此，以上為1，以下為0
然後再進行八個方向（加上原本不動共九個）可能進行微調，得出變化最小的
先將圖片縮小到1/32，九個方向進行比較，然後是1/16, 1/8...以加快速度
實作中，我們已曝光量中間程度那張（假設有九張，就是用第五張）作為標準圖
將其他影像依據標準圖做移動



上圖無使用MTB，下圖有。可看出下圖的圖片經過align之後較上圖清晰許多。

以下使用我們Artifact所使用的照片為範例：



曝光時間由短至長分別為

0.125, 0.25, 0.4, 0.6, 0.8, 1.3, 2.0, 3.0, 5.0

光圈設為f-5.0

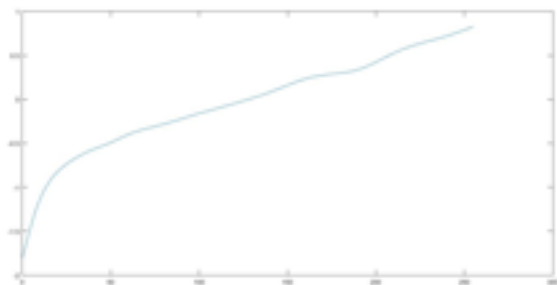
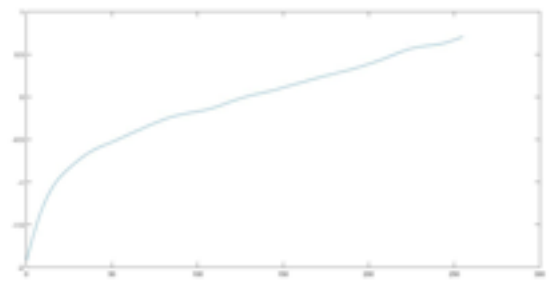
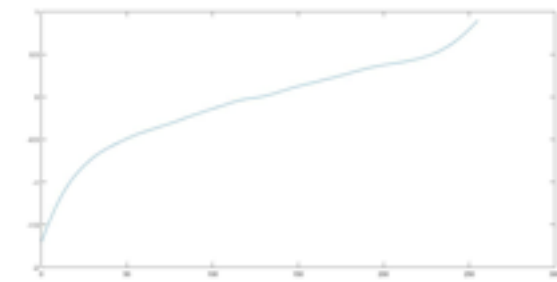
解析度為1500x1000

P = 9

N = 50(我們利用rand()函數，隨機在圖中取50點)

High Dynamic Range -> myHDR.m

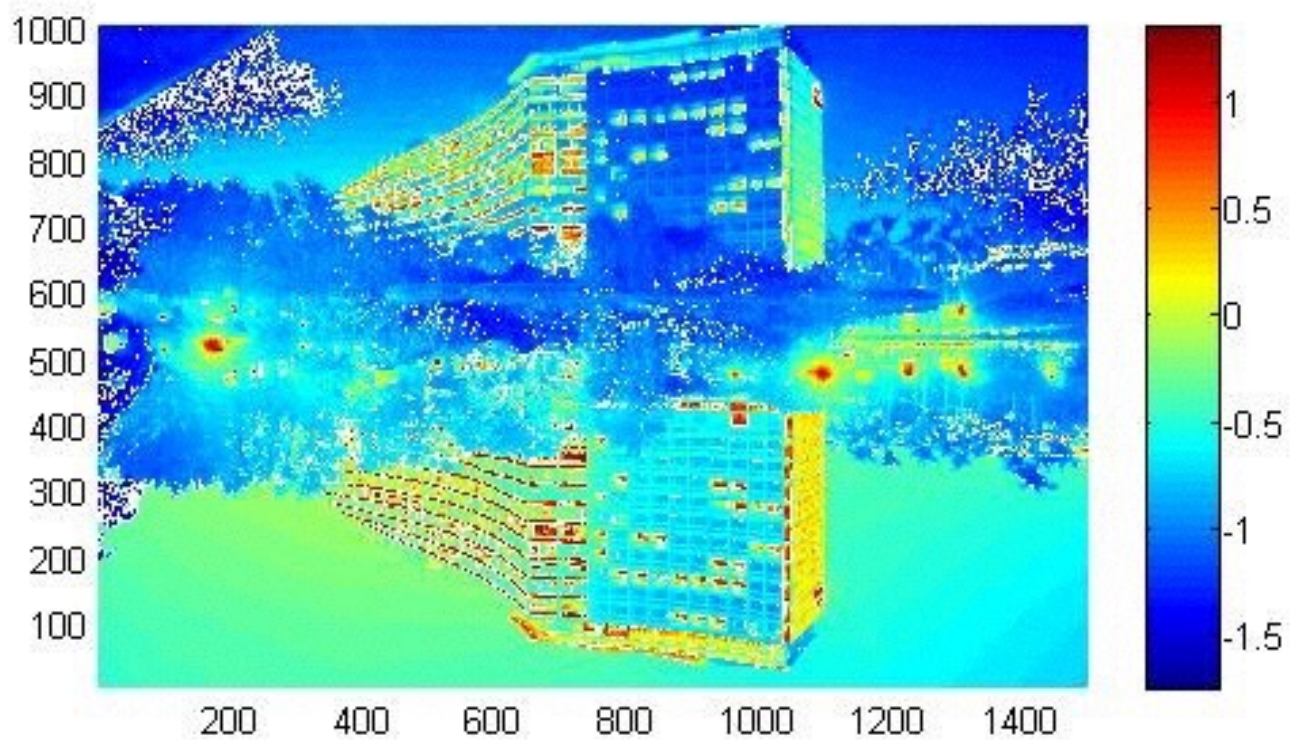
(a) Debevic's method -> gsolve.m



左上 右上 左下 分別為 R G B

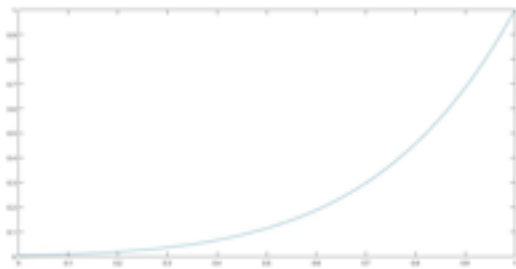
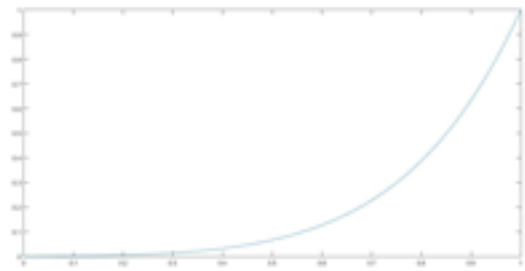
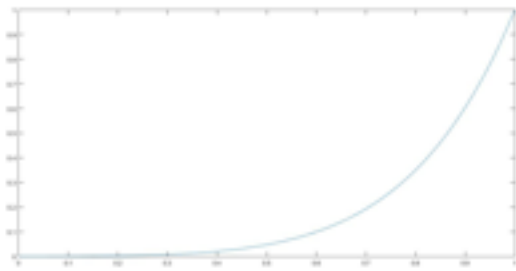
$\log(\text{Exposure}) - \text{pixel value}$

也就是 $\log(g(Z))$ curve



log尺度下的HDR radiance map

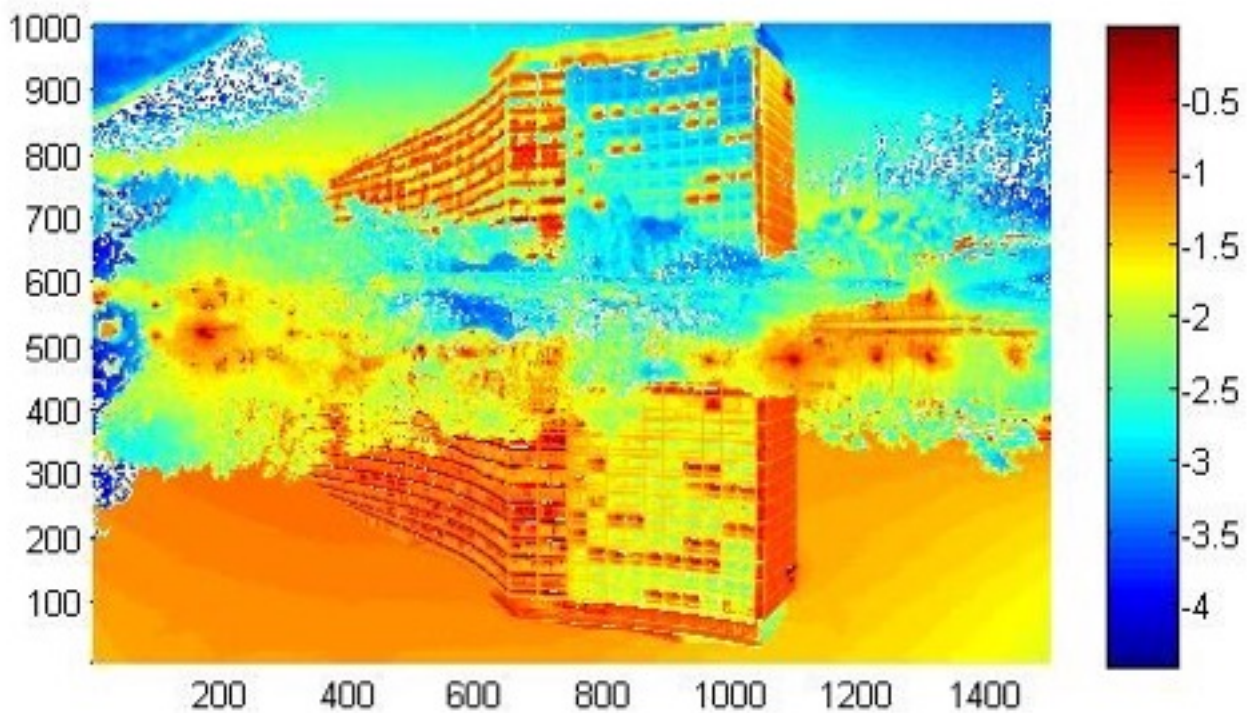
(b) Mitsunaga and Nayar's method -> gsolve2.m



左上 右上 左下 分別為 R G B

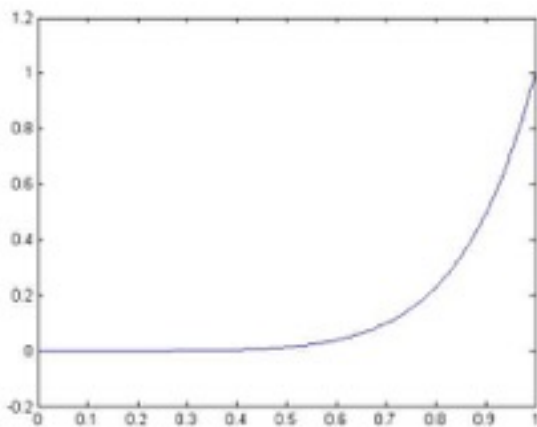
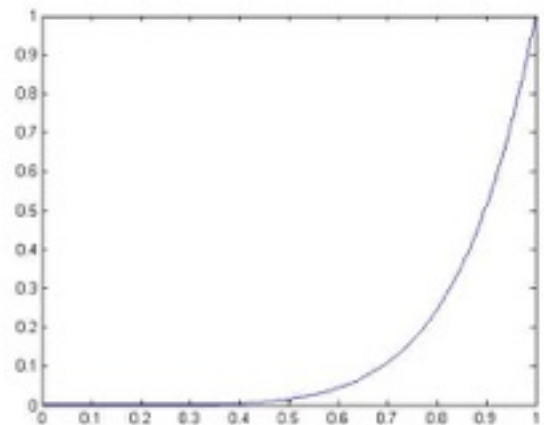
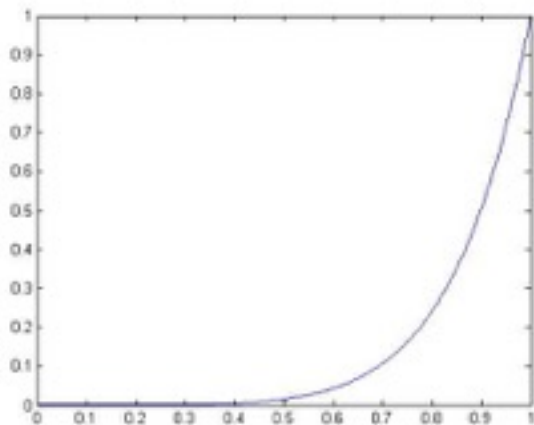
Exposure - pixel value

也就是 $g(Z)$ curve



log尺度下的HDR radiance map

比較兩個HDR出來的Radiance Map
除了尺度上的MAX MIN有差異之外
其餘大多相似



這是 $M=15$ 的結果

$M \geq 10$ 以上所得出的
 $g(Z)$ 已經差不多了

下圖為Mitsunaga and Nayar's method
在我們嘗試多次下的global $S=0.8$ $key=0.05$
這樣的情況下得出的成果



可遇下一頁結果相比較，比起Debevec's method，其成品「紅」了許多
之後的圖片，我們都是用Debevec的成果

Tone Mapping -> myTonemap.m

- For color images

$$\begin{bmatrix} R_d \\ G_d \\ B_d \end{bmatrix} = \begin{bmatrix} L_d \frac{R_w}{L_w} \\ L_d \frac{G_w}{L_w} \\ L_d \frac{B_w}{L_w} \end{bmatrix}$$

$$C_{\text{out}} = \left(\frac{C_{\text{in}}}{L_{\text{in}}} \right)^s L_{\text{out}}$$

for $C = R, G, B$. L_{in} and L_{out} denote the luminance before and after HDR compression, respectively, and the exponent s controls the color saturation of the resulting image. We found values between 0.4 and 0.6 to produce satisfactory results.

上圖取自 *Gradient Domain High Dynamic Range Compression* 這篇paper
以下 global tonemap 為 Debevec 所做出來的 hdr
經過反覆測試，我們認為 $S=0.6$ ， $\text{key}=0.4$ 有比較好的效果

$S = 0.6$ ， $\text{key} = 0.18, 0.5, 0.9$ -> 亮度變化



$\text{key} = 0.18$ ， $S = 0.4, 0.6, 1.0$ -> 彩度變化



以下local tonemap, $S=0.6$, $key=0.3$

$\phi = 7$, $e = 0.05, 0.5, 5$

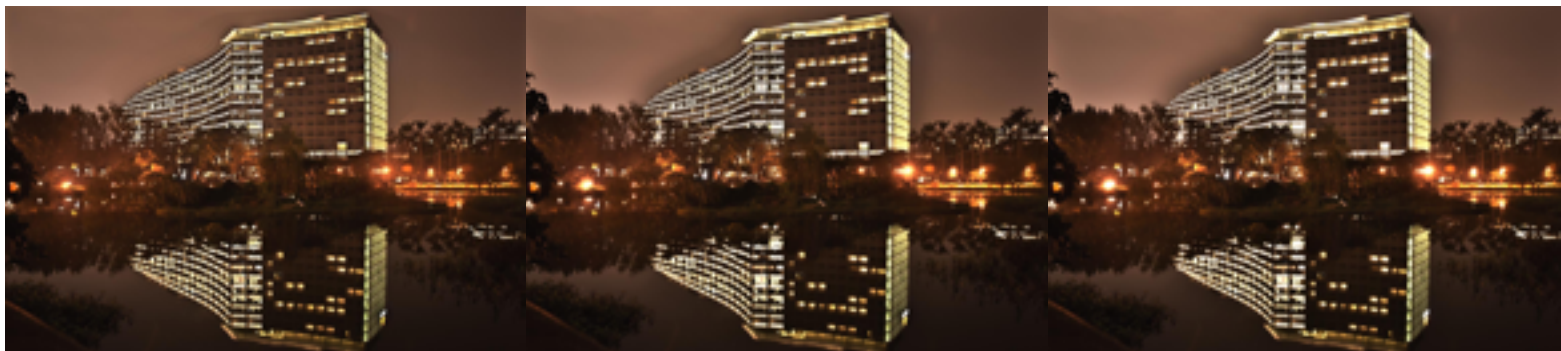
-> 可以注意到建築物周圍的黑影變化，還有光源附近的星芒大小



$\phi = 8$, $e = 0.05, 0.5, 5$



$\phi = 9$, $e = 0.05, 0.5, 5$



改變 ϕ 值並不會造成太大的更動
local相較global亮暗區隔更明顯



第一張是我們用matlab內建的tonemap做出來的
第二張是我們用photomatix tonemap的成果



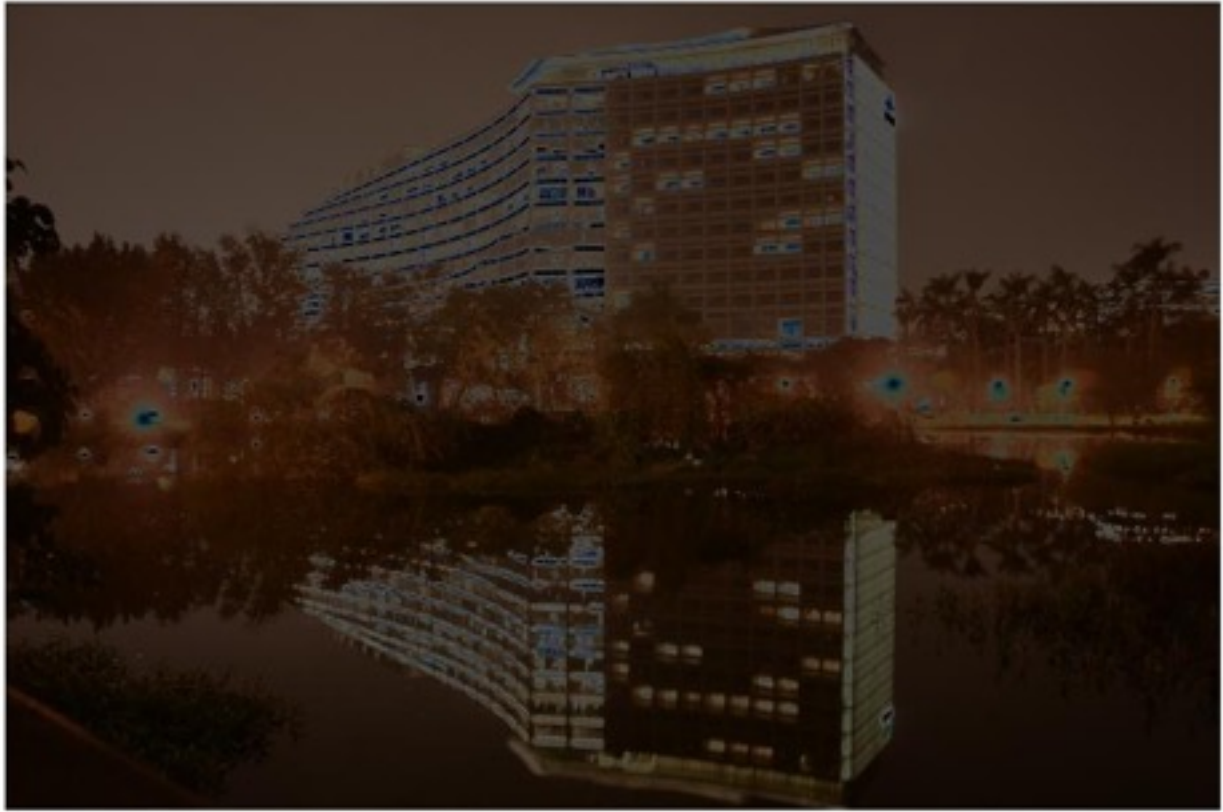
這張是我們自己寫的matlab myTonemap.m 做出來的成果

專業軟體所做出來的成果相較起來，我們的還是差了许多
最後我們選擇用Debevec's method的hdr以及
photomatix tonemap的成品作為我們的artifact

(*)Ghost Removal

-> `myGhost.m` & `mySobelEdge.m` & `myImageVariance.m`

這部分我們有嘗試做做看，但只有做到Image的Weighted Variance這一步而已



如同講義上所述，邊界與高變化的區塊會被標記起來
但之後的region analysis我們就沒有完成

References

除了教授的講義內容外，

我們也參考了 *<High Dynamic Range Imaging>* 這本教科書

以及Photographic Tone Reproduction for Digital Images這篇論文

http://www.cmap.polytechnique.fr/~peyre/cours/x2005signal/hdr_photographic.pdf