

# VFX HW1 Report

R05944013 高滿馨 R05922069 陳思穎

## Project Description

因為相機硬體的限制，所以一般的相片只能呈現某個亮度範圍內的場景，暗部跟亮部細節不能並存，無法完整的還原人眼所見的景象。因此我們透過拍攝多張同個場景，但有不同快門時間的照片，重建出相機內部將亮度轉換成rgb值的方程式。利用這個方程式重建出radiance map，再透過這個radiance map來做tone mapping，重建一張HDR的影像，藉由這樣的方式完整地呈現人眼所見的場景。

實作的部分有三大步驟，拍攝完照片後會先對照片做alignment，然後再重建這些照片的radiance map，最後再做tone mapping得到最後的結果。

## Part1. MTB

做alignment是為了避免因為拍攝時的晃動或反光鏡反彈造成影像的偏移，所以必須先將照片對齊。我們使用MTB( Median Threshold Bitmap ) algorithm來做alignment，從相片中挑選一張覺得位置最好的照片，再把其他的照片跟它對齊。

MTB algorithm的概念是以圖片的intensity的median當作threshold，把圖片都變成bitmap，再把這些bitmap做數次downsampling，去計算原圖與待對齊圖片的9個方向(不動與往8個鄰居位移，如圖1)位移的error，找出error最小的方向並回傳給下一層，下一層再使用上一層回傳的資訊繼續計算最好的移動方向，藉由這樣的方式來做alignment。

另外我們有建一個exclusion map，把與threshold太過接近的值都設為0，因為這些點很容易造成誤差，所以我們不去考慮這些點。我們最後把這個map跟做完XOR的結果做AND，就可以把這些點忽略掉。

Error function:  $\text{sum}(\text{sum}(\text{XOR(Bitmap ImgA, Bitmap ImgB)} \& \text{exclusion A} \& \text{exclusion B}))$

( -1, -1 )	( 0, -1 )	( 1, -1 )
( -1, 0 )	( 0, 0 )	( 1, 0 )
( -1, 1 )	( 0, 1 )	( 1, 1 )

圖1

Source Image



level 0  
bitmap



level 1  
bitmap



level 2  
bitmap



level 3  
bitmap



level 4  
bitmap



level 5  
bitmap



level 6  
bitmap



Result (黑邊是被crop掉的區域)



## part2. HDR

照片的pixel值可以表示為，這個pixel實際的能量乘上這張照片的曝光時間，經過一個函數 $f$ 的作用後得到的值。（ $Z_{ij}$  表示第j張照片的第i個pixel）

$$Z_{ij} = f(E_i \Delta t_j)$$

然後把這個式子變成：

$$\ln f^{-1}(Z_{ij}) = \ln E_i + \ln \Delta t_j$$

使用  $f^{-1}$  的優點有兩個：

1. 因為  $f : \mathbb{R} \rightarrow [0,255]$ ，所以取  $f$  的 inverse 變成  $f^{-1} : [0,255] \rightarrow \mathbb{R}$  會比較好處理
2. 我們要求的是  $E_i$ ，所以要把input變成  $Z_{ij}$

把  $f^{-1}$  改寫成一個  $g$  函數：

$$g(Z_{ij}) = \ln E_i + \ln \Delta t_j$$

我們目標是要找到一條  $g$  curve 讓所有的點都落在這條curve上，而且這條curve要smooth，因此可以寫成一個 objective function，然後 minimize 這個 objective function：

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

所有的點都要落在線上

讓curve smooth

$$g''(z) = g(z-1) - 2g(z) + g(z+1)$$

加上 weight 是因為靠近0和255的地方會比較不reliable，可能會對應到很多個實際亮度，而中段的部分比較reliable，因此乘上不同的weight：

$$w(z) = \begin{cases} z - Z_{min} & \text{for } z \leq \frac{1}{2}(Z_{min} + Z_{max}) \\ Z_{max} - z & \text{for } z > \frac{1}{2}(Z_{min} + Z_{max}) \end{cases}$$

此外還要再加上一個限制，因為算出來的解應該會有無限多組，因此  $g$  curve會上下移動，所以要把固定住，才會有唯一解：

$$g(Z_{mid}) = 0, Z_{mid} = 1/2(Z_{max} + Z_{min})$$

要解這個 function，可以把問題轉換成一個解一個Linear System：

$$\min \sum_{i=1}^N (\mathbf{a}_i \mathbf{x} - \mathbf{b}_i)^2 \rightarrow \text{least-square solution of } \begin{bmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \\ \vdots \\ \mathbf{a}_N \end{bmatrix} \mathbf{x} = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \vdots \\ \mathbf{b}_N \end{bmatrix}$$

我們的變數如下：

$$Ax=b$$

我們使用Matlab的gsolve來解這個式子，最後會得到g和我們所使用的N個sample點的E，但我們的目標是要得到所有的點的E。我們已知

$$\ln E_i = g(Z_{ij}) - \ln \Delta t_j$$

理論上用這個式子可以算出來所有點的E，但是因為靠近0和255的地方會比較不reliable，因此這邊仍然要乘上一個weight：

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

得到這個 radiance map 後，使用 matlab 的 `hdrwrite()`，就可以得到一張 .hdr 影像。

參數： $\lambda = 0.1$ , sample 200 個點

### Part3. Tone Mapping

tone mapping的部份我們選擇了photographic tone reproduction 這個方法，這個algorithm有global operator和local operator。

global operator的部份我們套用下面的公式，來將HDR算出的luminance轉換成相片呈現的顏色。

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

a: keyValue，可以調整相片亮度，a  
越大相片越亮

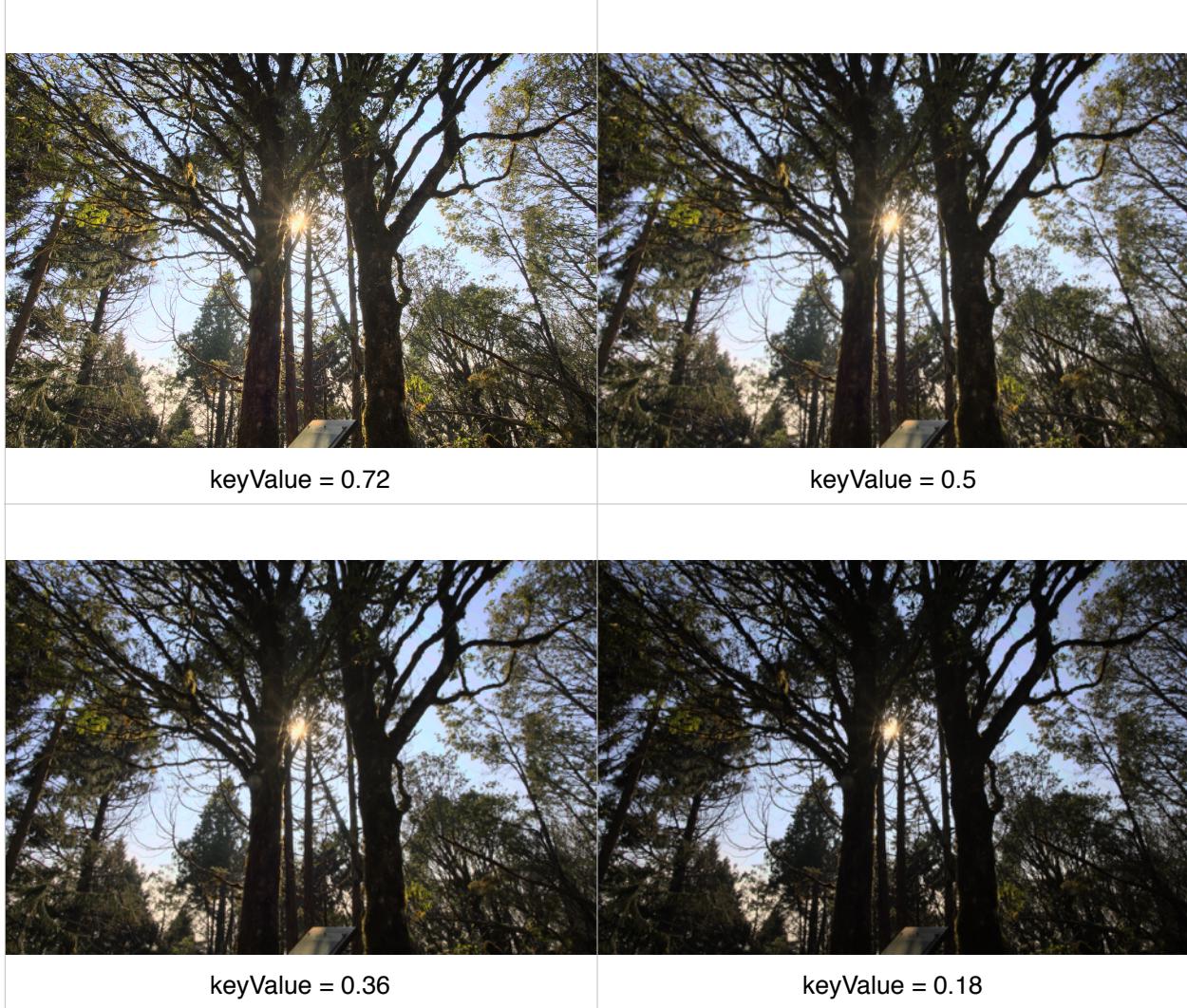
$$\bar{L}_w = \exp\left(\frac{1}{N} \sum_{x,y} \log(\delta + L_w(x, y))\right)$$

計算log domain上的平均值  
Lw: 相機捕捉到的luminance

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

轉換成display的luminance

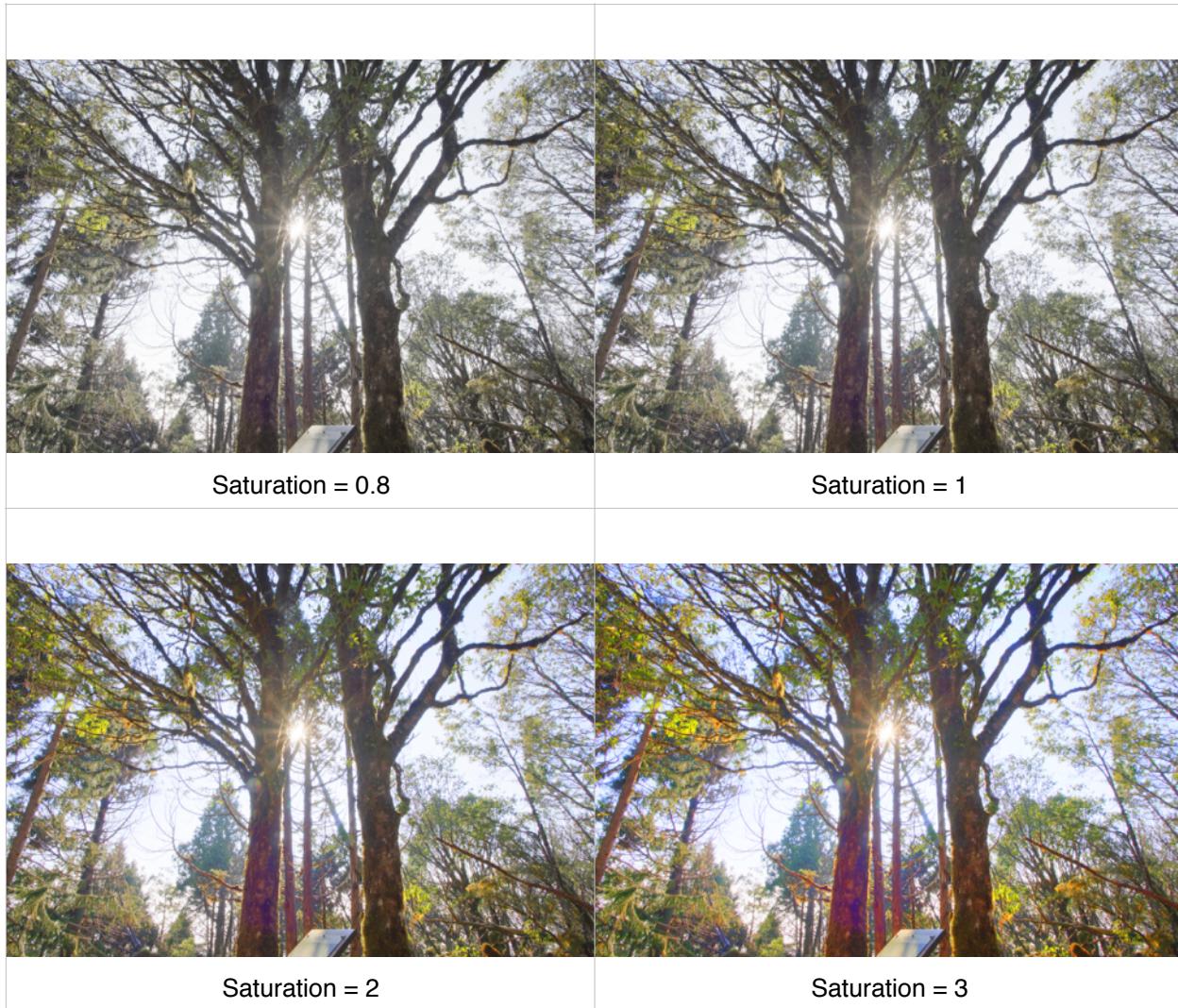
## Global Operator Result 1



不過我們發現這樣做出來的圖片，在過亮的部分會出現問題。



我們覺得是因為那些點在3個channel的luminance差距過大，所以導致轉換的時候出現一些問題，因此我們分別對三個channel做normalize，並把數值compress至0~1之間，這樣做確實可以解決過亮的問題，但彩度就會降低，因此需要提高飽和度，做出來後得到不錯的效果。



## Global Operator Result 2

	
keyValue = 0.72	keyValue = 0.5
	
keyValue = 0.36	keyValue = 0.18

## Local Operator

Local operator相較於global operator來說，可以增加對比度，讓一些細節更明顯。他會找出與自己pixel值相近的最大區域，求得這個區域的平均值，來作轉換。如此一來可以讓暗的點變得更暗，亮的點變得更亮。

首先他會與鄰近的pixel做不同scale的gaussian，

$$L_s^{blur}(x, y) = L_m(x, y) \otimes G_s(x, y)$$

接著再計算pixel在不同的scale的gaussian間的差值，

$$V_s(x, y) = \frac{L_s^{blur}(x, y) - L_{s+1}^{blur}(x, y)}{2^\phi a/s^2 + L_s^{blur}}$$

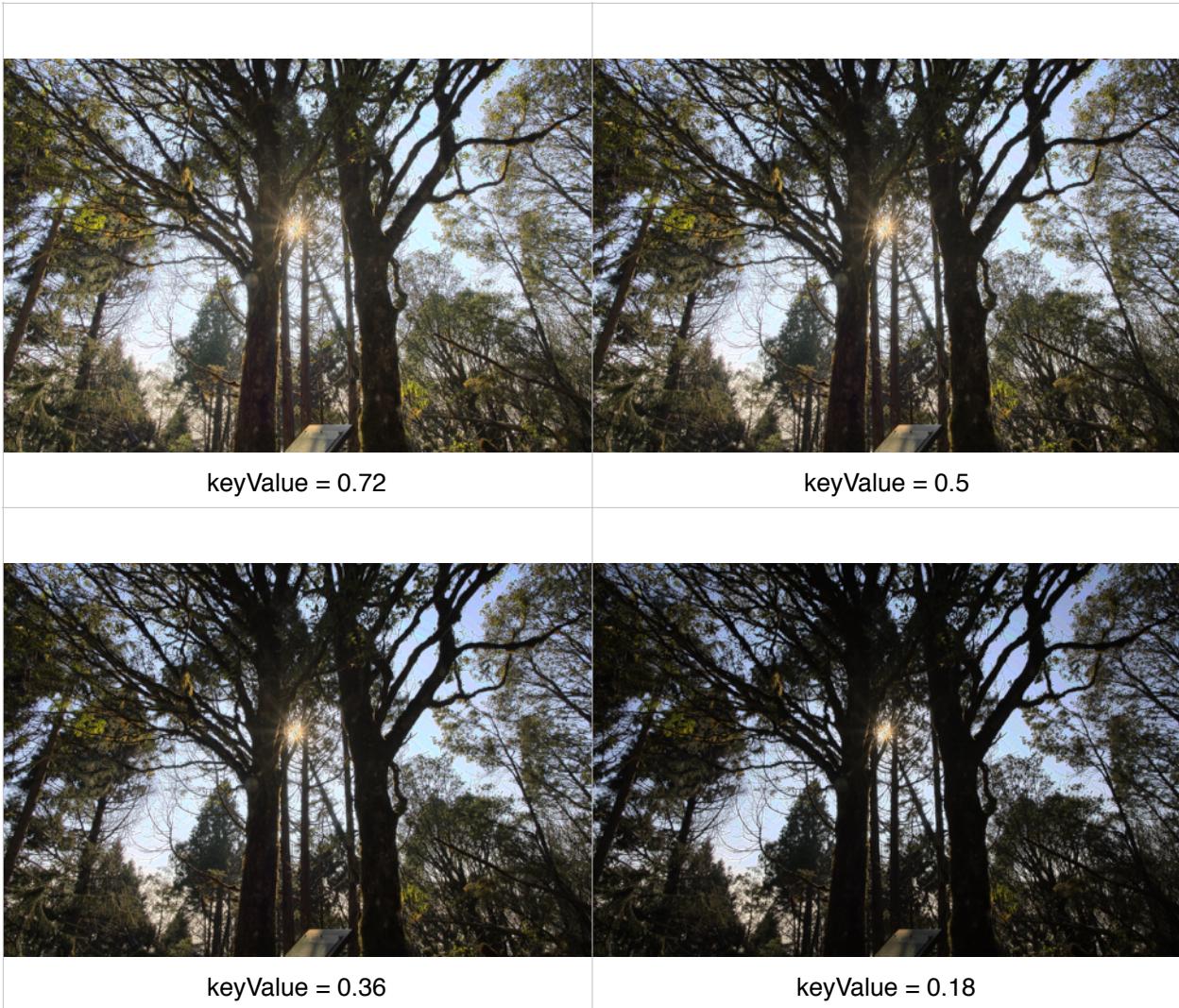
找出最大的scale，使得兩個gaussian間的差距小於一個定值

$$s_{max} : |V_{s_{max}}(x, y)| < \epsilon$$

最後再轉換成display的luminance，

$$L_d(x, y) = \frac{L_m(x, y)}{1 + L_{s_{max}}^{blur}(x, y)}$$

## Local Operator Result 1

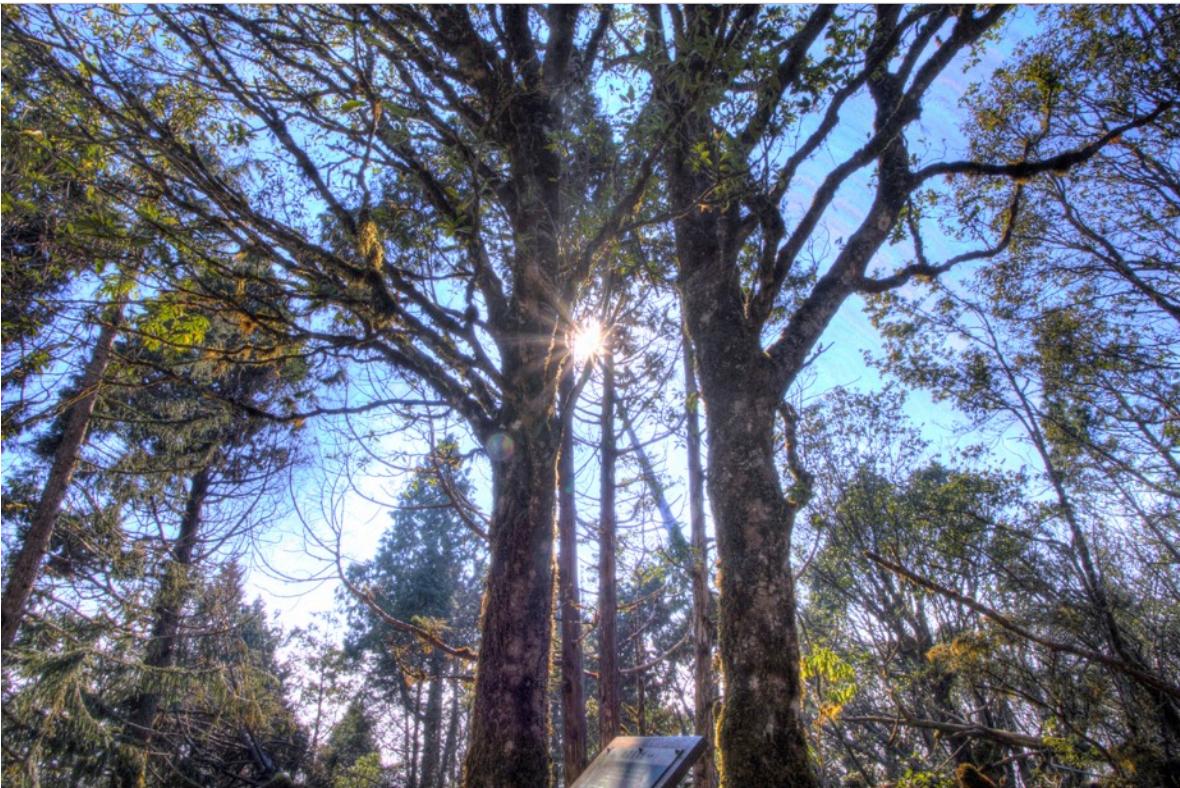


## Local Operator Result 2

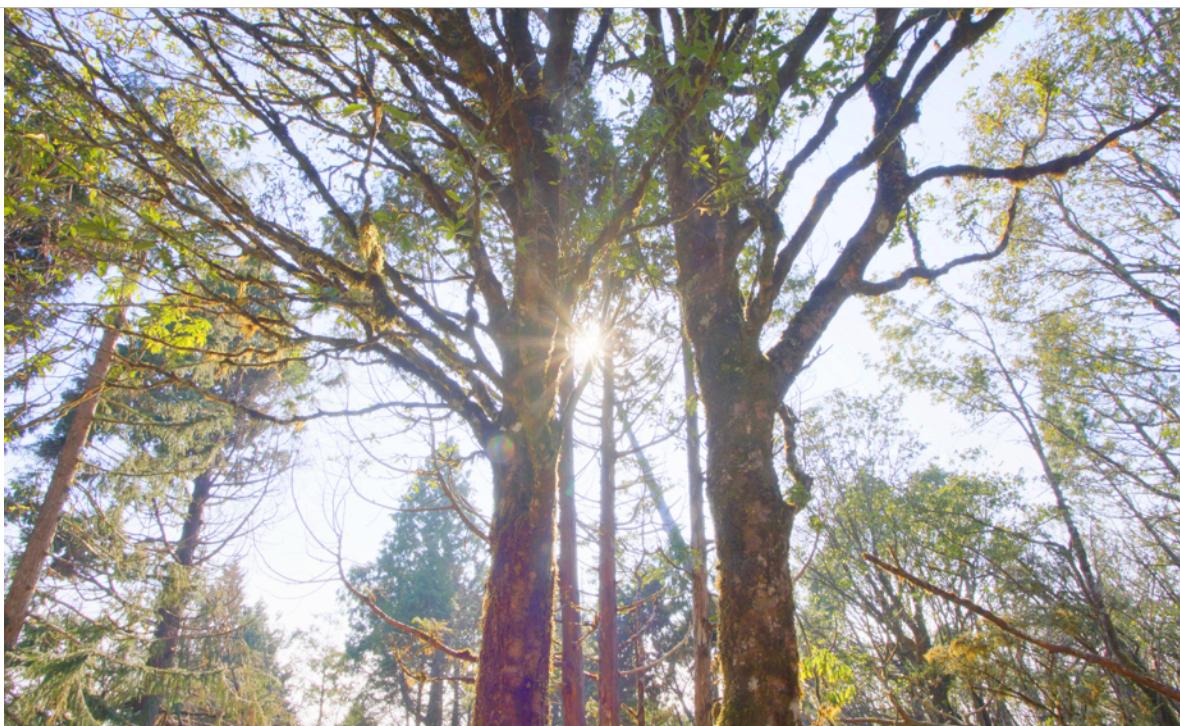
	
keyValue = 0.72	keyValue = 0.5
	
keyValue = 0.36	keyValue = 0.18

和Photomatix做出來的結果比較：

Photomatix



Ours (Global, Key Value = 0.72, Saturation = 2)



Photomatix



Ours (Global, Key Value = 0.72)



## Final Best Result



我們最後挑選使用photographic tone reproduction的global operator做的圖片當作最佳圖片。

Parameter:

keyvalue = 0.72, Saturation = 2