

VFX project1

High Dynamic Range Imaging

b02901004 王士元
b02901101 林士庭

1 Introduction

在本次project中，我們將實作High Dynamic Range Imaging(HDR)。在一般自然環境中，我們的眼睛可以感知到相當大範圍的radiance，但當我們以相機等方式記錄下這些影像之後，往往會因為256 pixel levels、camera response的限制而喪失掉原始的radiance資訊，只能獲取low dynamic range的影像。因此，在本次的project中，我們希望能透過不同曝光時間(快門速度)的狀況下，針對同一個場景拍攝照片，再根據HDR的相關演算法重新產生出具有high dynamic range的radiance map，進而產生出新的影像。

一般而言，我們可以將相機的功能表達如下，lens會將scene radiance(L)轉換成sensor irradiance(E)，而後，藉由shutter來收集光線，產生sensor exposure(X)，再經由相機內部的功能(CCD、ADC、non-linear mapping)後產生最終的digital values(Z)。

$$Z_{ij} = f(E_i \Delta t_j) \quad (1)$$

而我們的目的在於，透過已知的曝光時間(Δt_j)，根據 Z_{ij} 來求解出原始的 E_i 。

在本次的project中，我們藉由python實做了下列功能以及演算法。

- **Alignment**

MTB(median threshold bitmap)

- **Reconstructing response curve**

Debevec's method, Robertson's method

- **Tone mapping**

Photographic tone reproduction(Reinhard) method, Fast bilateral filtering(Durand) method

2 Alignment

在製作HDR影像時，我們需要對同一場景拍攝在不同曝光時間下的多張照片。然而因為在拍攝時，相機可能因為快門震動等因素而使得各張照片有所偏移，造成計算response curve的誤差，因此在計算response curve之前，我們參考[1]先對各張照片進行alignment，以將偏移的照片調整回來。以下則是實作的方式。

2.1 Implementation

1. 首先，我們先對兩張要進行alignment的圖片，img_1, img_2計算其各自的median threshold map(MTB)，mtb_1, mtb_2。使用MTB的原因在於，因為不同於gradient，各張照片即使是在不同的曝光時間下MTB依然大致是相同的，因此對HDR的alignment來說是一不錯的feature。
2. 接下來，我們即可以一張圖的MTB(mtb_1)為基準，計算

$$\arg \min_{\Delta x, \Delta y} \text{NumDiff}(\text{mtb_1}, \text{mtb_2}^{(\Delta x, \Delta y)}) \quad (2)$$

其中 $\text{mtb_2}^{(\Delta x, \Delta y)}$ 為經向x, y方向移動 $\Delta x, \Delta y$ 個pixel後之mtb_2。然而式(2)的計算量是非常龐大的，因此此處我們對圖片建立了pyramid來增加搜尋速度，在pyramid中，每層的圖片的大小為上一層圖片之四分之一，即 $\text{dim}(\text{img}^{(n+1)}) = 0.5 \times \text{dim}(\text{img}^{(n)})$ 。對各層之圖片⁽ⁿ⁾，我們皆以下式計算 $\Delta x^{(n)}, \Delta y^{(n)}$

$$\begin{aligned} & \arg \min_{\Delta x^{(n)}, \Delta y^{(n)}} \text{NumDiff}(\text{mtb_1ff}^{(n)}, \text{mtb_2}^{(n)(\Delta x^{(n)}, \Delta y^{(n)})}) \\ & \Delta x^{(n)} \in \{2\Delta x^{(n-1)} \pm 1, 2\Delta x^{(n-1)}\} \\ & \Delta y^{(n)} \in \{2\Delta y^{(n-1)} \pm 1, 2\Delta y^{(n-1)}\} \end{aligned} \quad (3)$$

其中在計算function NumDiff(.)時，我們可利用bit-wise XOR很有效率的得到總差異數，如此一來，即可大大降低計算量並以遞迴的方式快速地計算出alignment的大致方向。

3. 在以MTB做alignment時，影像的雜訊有時會對結果產生不小的影響，因此在計算NumDiff(.)時，我們通常會將值在median附近的pixel排除掉，而為了達到這樣的目的，我們可以建立一個exclusion map，其中若pixel之value $\in [\text{median} - 4, \text{median} + 4]$ ，其在exclusion map中之值即為1，反之為0，而我們只要將MTB與exclusion map做bit-wise AND即可很有效率的排除雜訊的影響。

2.2 Results

2.3 Median Threshold Map



Figure 1: Median Threshold Map Pyramid, 最左上下兩張為原圖

2.4 Exclusion Map

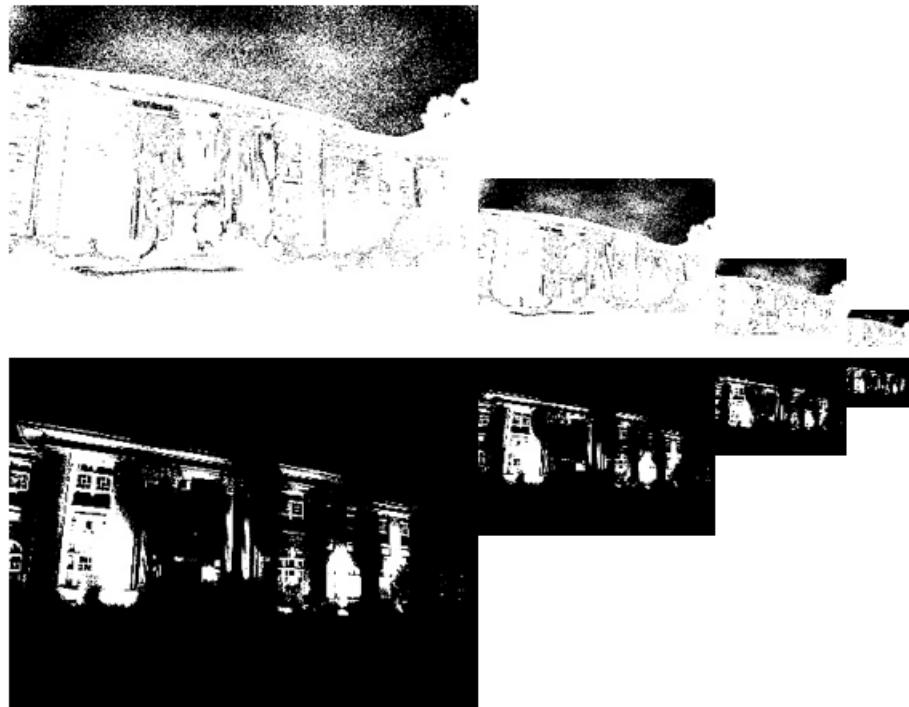


Figure 2: Exclusion Map Pyramid

2.5 XOR Map



Figure 3: 上排為未使用Exclusion Map的結果，下排為與Exclusion Map做bit-wise AND後的結果

3 Response curve reconstruction

3.1 Debevec's method [2]

為了要解出原始的 E_{ij} ，我們必須要先將原始的 camera response curve 近似出來，或是求解其反函數，可以表示如下

$$g(Z_{ij}) = \ln f^{-1}(Z_{ij}) = \ln E_i + \ln \Delta t_j \quad (4)$$

(4) 中我們定義一個新的函數 g ，當給定一個 pixel value 時，其 range 代表 log exposure。而要求解 inverse response curve 的方法，即可變為下述之 optimization problem:

$$\arg \min_{g,E} \mathcal{O} = \sum_i^N \sum_j^P [g(Z_{ij}) - \ln E_i - \ln \Delta t_j]^2 + \lambda \sum_{z=Z_{min}+1}^{z=Z_{max}+1} g''(z)^2 \quad (5)$$

除了 MSE 的項外，另外多出了 L2-regularization 的目的是希望可以讓解出來的 g curve 具有平滑的特性。

另外，由於 g curve 在極值處容易產生較大的斜率，我們並不希望這樣的曲線產生，因此會再額外乘上 weighting function (w)，希望在極值處仍然具有平滑的特性，並且不希望太過 fit data，同時也視中間處的 digital value 較具可信度，於是作出下列 weighting:

$$w(z) = \begin{cases} z - Z_{min} & \text{for } z \leq \frac{1}{2}(Z_{min} + Z_{max}) \\ Z_{max} - z & \text{for } z \geq \frac{1}{2}(Z_{min} + Z_{max}) \end{cases} \quad (6)$$

於是(4)可以修改為，

$$\arg \min_{g,E} \mathcal{O} = \sum_i^N \sum_j^P \{w(Z_{ij})[g(Z_{ij}) - \ln E_i - \ln \Delta t_j]\}^2 + \lambda \sum_{z=Z_{min}+1}^{z=Z_{max}+1} [w(z)g''(z)]^2 \quad (7)$$

進一步的，上述之 optimization problem 可以視為是 least-square problem，並且透過 singular-value decomposition (SVD) 即可求解，但由於這是一個 over-determined system，為了找到一組解，我們令 $g(127) = 1$ 。實作上，我們可以直接透過 numpy 中的 linalg 直接解得 g 。

此外需注意的是，由於在此處 SVD 的複雜度會和參與計算的 pixel 數量大幅增加，因此較為實際的作法是針對原始的照片取出適當的 samples，而我們使用的方法是隨機取樣，但將照片邊界的點排除掉，以免受到 alignment 結果的干擾。

3.2 Robertson's method [3]

在本篇 paper 中，作者在 exposure 的計算中加入了 noise 的成分，以此將原式描述成，

$$f^{-1}(Z_{ij}) = E_i \Delta t_j + N_{ij} \quad (8)$$

where N_{ij} is the noise. 如此一來，即可將原本的問題改寫為以 gaussian 作為 likelihood function 的 estimation problem，

$$P(f^{-1}, E_i | Z_{ij}, \Delta t_j) \propto \exp\left\{-\frac{1}{2} \sum_{i,j} w(Z_{ij})(f^{-1}(Z_{ij}) - E_i \Delta t_j)^2\right\} \quad (9)$$

於是可以在求解時，可以利用所有的data進行下列計算。

- 假設 f^{-1} 為已知

$$\hat{E}_i^{(n)} = \frac{\sum_j w(Z_{ij}) f^{-1}^{(n-1)}(Z_{ij} \Delta t_j)}{\sum_j w(Z_{ij}) \Delta t_j^2} \quad (11)$$

- 假設 E_i 為已知

$$\hat{f}^{-1}^{(n)}(m) = \frac{1}{|E_m|} \sum_{Z_{ij}=m} E_i \Delta t_j \quad (12)$$

以此方式反覆進行iteration計算，直到收斂。而在實作中，所使用的收斂條件是當ML的objective function下降比例低於某一個threshold。

使用此方法的好處在於，我們不需要對inverse response function f^{-1} 做出額外的假設，並且使用所有的data進行計算，可以獲得較為robust的結果，但相對的，收斂時間也會增加，並且在python的實作時，以我們目前的做法，會需要大量的時間計算。因此，我們採用了藉由Debevec進行initialization，希望藉此加速收斂速度。

3.3 Comparison

針對同一組相片，分別採用Debevec's method以及Robertson's method來重建response curve，其結果如Figure 4所示，兩種方法最大的差異在於極端值處的斜率，正如Debevec's method所希望的，極端值的表現確實沒有出現極大的斜率，較為平滑。

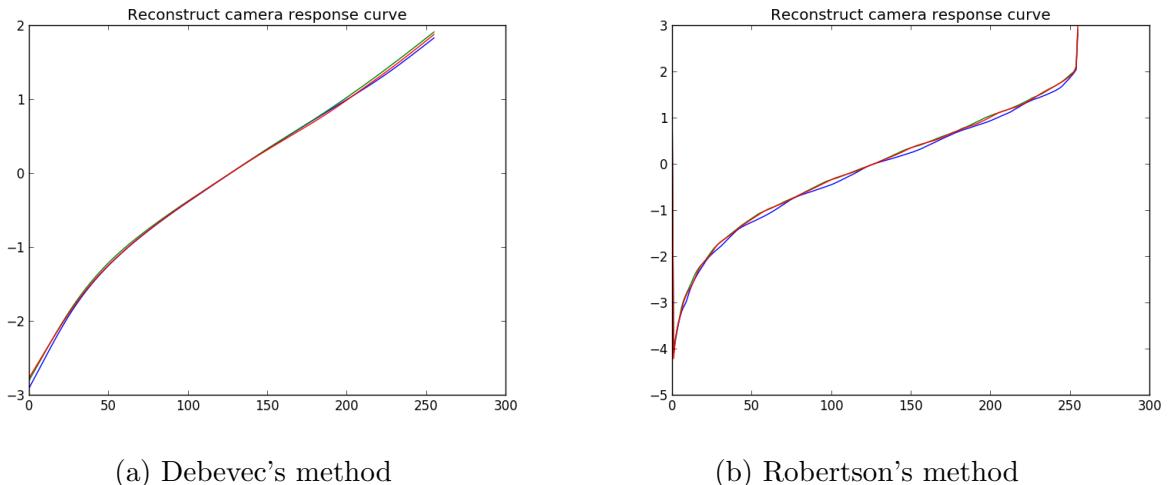


Figure 4: Recovered response curve

而相對應的，從radiance map的結果上來看，也可以觀察到靠近極端值時，Robertson's method會有更明顯的強度差異出現。

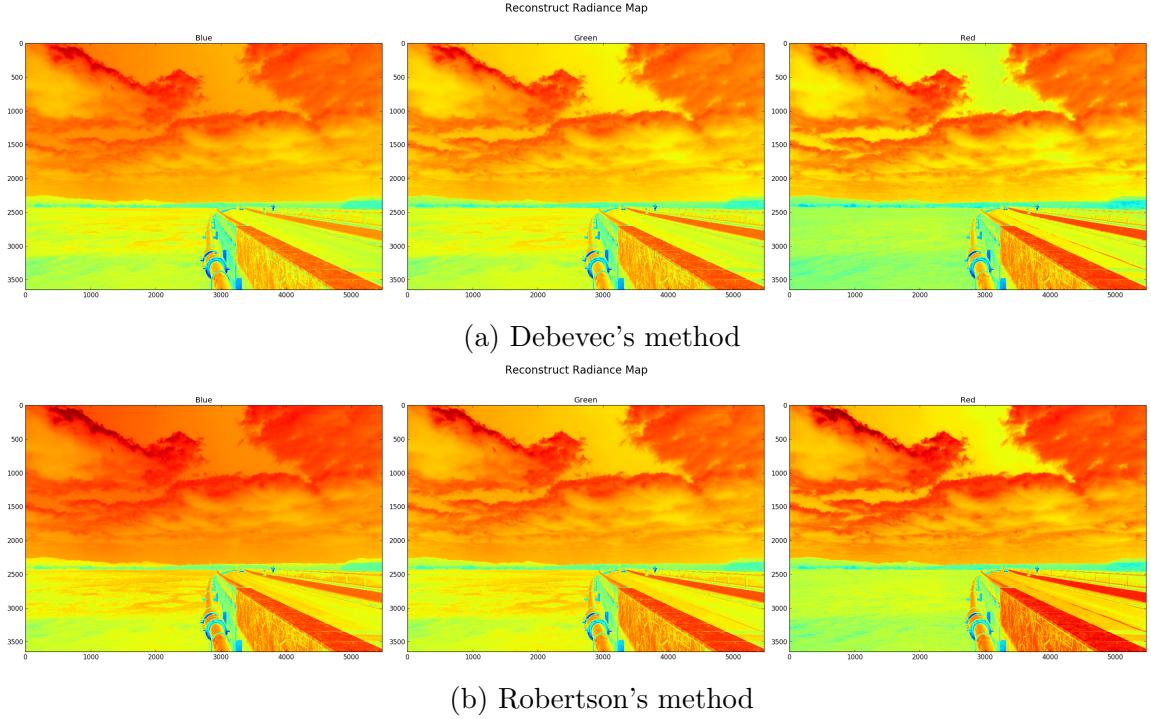


Figure 5: Radianc map

4 Tone mapping

根據radianc map，我們可以近似出拍照環境當下的world luminance，但如何將radianc map轉換成人眼可以感知、符合平常人類對色彩感受的影像，便需要tone mapping的轉換。

4.1 Photographic tone reproduction(Reinhard) method[4]

在本演算法中，我們會將radianc map的資訊透過color的轉換得到luminance domain的資訊。在實驗中我們所使用的轉換關係為：

$$L = 0.2126R + 0.7152G + 0.0722B \quad (13)$$

再將獲得的world luminance以及world color透過這樣的關係轉換成display luminance以及color:

$$\begin{bmatrix} R_d \\ G_d \\ B_d \end{bmatrix} = \begin{bmatrix} L_d \frac{R_w}{L_w} \\ L_d \frac{G_w}{L_w} \\ L_d \frac{B_w}{L_w} \end{bmatrix} \quad (14)$$

4.1.1 Global operator

首先，會將得到的world luminance透過(15)log-average的方式來近似出場景的key，key的意義代表影像的亮暗程度。

$$\bar{L}_w = \exp\left(\frac{1}{N} \sum_{x,y} \ln(\delta + L_w(x,y))\right) \quad (15)$$

另外，再引入一個user-specified key value(a)後，可以產生(16)中的scaled luminance(L_m)。

$$L_m(x, y) = \frac{a}{L_w} L_w(x, y) \quad (16)$$

接著，希望可以將high luminance區域加以壓縮，但同時又希望有部分的luminance可以到達最亮的程度，因此display luminance可以定義如下：

$$L_d(x, y) = \frac{L_m(x, y)(1 + \frac{L_m(x, y)}{L_{white}^2})}{1 + L_m(x, y)} \quad (17)$$

其中 L_{white} 是最小會使得 $L_d = 1$ 的 L_m 數值。

4.1.2 Local operator

除了對整張圖片進行key以及world to display的mapping，也可以增加局部區域的對比程度，此種方式相當類似於dodge-and-burning。因此，我們需要針對影像中對比程度進行適當的估計，其核心的想法是，由於我們會將這樣的方式利用在被高對比度所包圍的區域上，此一區域的大小就可以作為局部對比度的衡量標準。而一個常用的方式是difference of Gaussian，利用不同scale的Gaussian kernel(18)產生出blurred images(19)，再將scale相鄰的gaussian blurred images相減(20)作為評估。

$$G_s(x, y) = \frac{1}{\pi(\alpha s)^2} \exp\left\{-\frac{x^2 + y^2}{(\alpha s)^2}\right\} \quad (18)$$

$$L_s^{blur}(x, y) = L_m(x, y) \otimes G_s(x, y) \quad (19)$$

$$V_s(x, y) = \frac{L_s^{blur}(x, y) - L_{s+1}^{blur}(x, y)}{\frac{2^{\phi_a}}{s^2} + L_s^{blur}(x, y)} \quad (20)$$

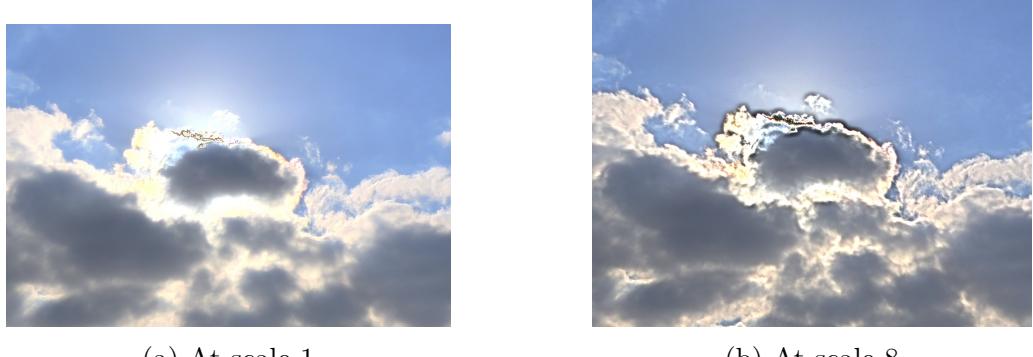
這裡的想法是，根據(20)，如果此一區域沒有大對比度差異存在，則 $V_s(x, y)$ 應該會相當小，而接下來的目標是要找到一個最大的區域能夠保有小於某個threshold的對比度差異，可以表示為：

$$\begin{aligned} & \underset{s}{\text{maximize}} && s \\ & \text{subject to} && |V_s(x, y)| < \epsilon \end{aligned} \quad (21)$$

而在實作上，我們採用了原始paper的建議參數， $\alpha = \frac{1}{2\sqrt{2}}$ ，相鄰的gaussian kernel scale相差1.6倍，並且在1至43個pixel的範圍限制中。

4.1.3 Experiments

當選用的gaussian kernel scale不同時，會發現到scale越大，對比度會被提升的越明顯，甚至出現異常的黑環於高對比區邊界上，一如paper中所提及的。



(a) At scale 1

(b) At scale 8

Figure 6: Local operator under different Gaussian scale

如Figure 7所示，在雲和天空交界處會發現到隨著scale上升，亮區和暗區的界線會更加明顯，甚至有黑環環繞在旁邊。

而目前實作上遇到最大的問題在於，對於高亮度的區域，例如Figure 7的陽光區域，很容易產生異常的顏色塊。

4.2 Fast bilateral filtering(Durand) method[5]

在tone mapping中，另一個常見的演算法則是gamma compression。對一pixel之world luminance，我們可以下式將其轉換為display luminance：

$$L_d = A \cdot L_w^\gamma \quad (22)$$

如此一來即可將 $L_d \in [0, 1/A^{1/\gamma}]$ map至 $L_w \in [0, 1]$ ，但是，此種做法會造成細節的模糊，因此通常會保持高頻的luminance，而只對低頻的luminance做gamma compression，然而這樣的方式又會在邊界上產生Halo Artifacts。為了解決這樣的問題，Fredo Durand et al.在[5]中則提出了利用Bilateral Filter做blurring的tone mapping演算法，如此gamma compression亦會作用在明顯的edge上，也就可避免產生Halo Artifacts。

4.2.1 Fast Bilateral Filtering

在filtering的演算法中，一般常見的方式如gaussian blurring。此方式是以一spatial的gaussian kernel

$$f(x, y) = e^{-\frac{(x^2+y^2)}{2\sigma^2}} \quad (23)$$

對圖片做low pass filtering，意即pixel s在經過filter後之pixel數值

$$J_s = \frac{1}{k(s)} \sum_{p \in \Omega} f(s_x - p_x, s_y - p_y) I_p \quad (24)$$

但這樣會使一個很pixel數值差異極大的邊界變得平滑，要解決這樣的問題，我們可以引入另一個在intensity domain的gaussian function

$$g(\Delta I) = e^{-\frac{\Delta I^2}{2\sigma^2}} \quad (25)$$

來作為計算時額外的weight，意即

$$\begin{aligned} J_s &= \frac{1}{k(s)} \sum_{p \in \Omega} f(s_x - p_x, s_y - p_y) g(I_s - I_p) I_p \\ \text{where } k(s) &= \sum_{p \in \Omega} f(s_x - p_x, s_y - p_y) g(I_s - I_p) \end{aligned} \quad (26)$$

這樣一來與pixel s intensity差距很大之pixel對 J_s 的數值貢獻就會降低，也就可以防止很明顯的邊界在blurring的過程中被模糊。

4.2.2 Implementation Details

在式(24)中，因為kernel與pixel數值無關，因此可將其拆成圖片與kernel之convolution並以FFT進行快速的計算。但是在式(26)中，其weight與 I_s 有關，無法簡單的拆解成convolution。不過因為其weight也只有與 I_s 有關我們可以把所有可能的 ΔI 離散化成j個segment，並對各個pixel s都去計算在每個segment下之 J_s ，即

$$J_s^j = \frac{1}{k^j(s)} \sum_{p \in \Omega} f(s_x - p_x, s_y - p_y) g(i^j - I_p) I_p \quad (27)$$

where $k^j(s) = \sum_{p \in \Omega} f(s_x - p_x, s_y - p_y) g(i^j - I_p)$

經過這樣的改寫，我們可以看出27中的兩式都可以被拆成convolution的形式，已就可以以FFT快速的計算得。而要得到 J_s 之值，我們可利用與 I_s 最靠近的兩個 i^j 計算得之 J_s^j 做linear interpolation來得到估計。最後我們也可以先對原本的圖片先做subsample，以上述方式做blurring，再同樣以linear interpolation做upsampling來得到更進一步的加速。

4.2.3 Gamma Compression

以bilateral filtering結合gamma compression之tone mapping演算法如Alg.1。

Algorithm 1 Tone Mapping by Fast Bilateral Filtering

Input: Radiance map of R, G, B, targetContrast

Output: HDR Image

- 1: $L_w = 0.0722 * B + 0.7152 * G + 0.2126 * R$
 - 2: $\log L_w = \log_{10}(L_w)$
 - 3: $\log Base = \text{Bilateral}(\log L_w)$
 - 4: $\log Detail = \log L_w - \log Base$
 - 5: $\text{CompressionFactor} = \log(\text{targetContrast}) / (\max(\log Base) - \min(\log Base))$
 - 6: $\log AbsoluteScale = \max(\log Base) * \text{CompressionFactor}$
 - 7: $\log L_d = \log Base * \text{CompressionFactor} + \log Detail - \log AbsoluteScale$
 - 8: $L_d = 10^{\log L_d}$
 - 9: $R_d = L_d^{\frac{R_d}{L_w}}, G_d = L_d^{\frac{G_d}{L_w}}, B_d = L_d^{\frac{B_d}{L_w}}$
 - 10: $R_d = R_d^\gamma, G_d = G_d^\gamma, B_d = B_d^\gamma$
-

其中之參數targetContrast可調整影像之對比，而我們使用的數值為[5]中建議的5。在經過演算法處理後，gamma compression就不只會針對低頻，也會作用在明顯的edge上，也就可避免產生Halo Artifacts，進而得到更漂亮的HDR影像。

5 Results

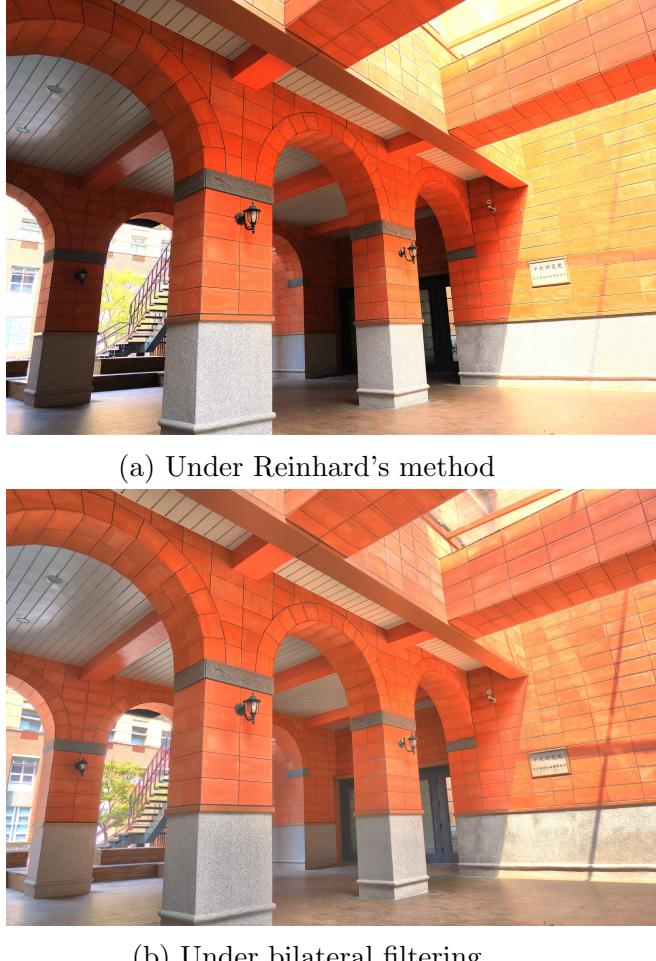


Figure 7: Our final HDR image under different tone mapping

References

- [1] Greg Ward, Fast, Robust Image Registration for Compositing High Dynamic Range Photographs from Handheld Exposures, 2003
- [2] Paul E. Debevec, Jitendra Malik, *Recovering High Dynamic Range Radiance Maps from Photographs*, SIGGRAPH 1997.
- [3] Mark A. Robertson, Sean Borman, Robert L. Stevenson, *Estimation-theoretic approach to dynamic range enhancement using multiple exposures*, Journal of Electronic Imaging 2003.
- [4] Erik Reinhard, Michael Stark, Peter Shirley, Jim Ferwerda, *Photographics Tone Reproduction for Digital Images*, SIGGRAPH 2002.
- [5] Fredo Durand, Julie Dorsey, Fast Bilateral Filtering for the Display of High Dynamic Range Images, SIGGRAPH 2002