# Vfx Project 1

B00901017電機四 王敦儒

B00901002 電機四 賴映安

**問題描述**:

由於非HDR影像在亮度動態範圍很大的場景效果較差，因此我們利用在同一場景，不同曝光時間下的多張非HDR的影像，透過解出response curve的方式，將這些非HDR的影像還原成該場景的一張HDR影像，最後再利用不同種類的tone mapping方式合成出一張視覺效果較佳的影像。

**學習目標:**

* + 利用MTB的演算法將同場景，多張不同曝光時間的影像對齊
  + 利用多張不同曝光時間下的影像，解出非線性的response curve
  + 依據解出的response curve將不同曝光時間的影像還原成該場景的 HDR影像
  + 將HDR影像用Photographic Tone Reproduction for Digital Images的 方法降低對比，合成出一張視覺效果佳的影像
  + 將HDR影像用Fast Bilateral Filtering for the Display of High-Dynamic- Range Images的方法降低對比，合成出另一張視覺效果佳的影像

**演算法:**

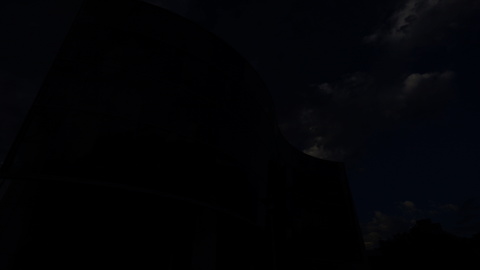
* + Response Curve:
    - 定義Objective function
    - 在N張影像中，選定P個同樣座標的點
    - 利用SVD分別對RGB三個顏色解出各自的Response Curve
  + HDR:
    - 場景中的每個點，將其在所有影像中的值加權平均
  + 對齊:
    - 目的是使輸入的多張相片對齊，避免因震動造成影像在同一個像 素(pixel)位置不一致的情況
    - 採用Median Threshold Bitmap(MTB)演算法
      * 將圖片皆轉為灰階(Luminance)的影像，並找出中位數指定 為Threshold
      * 先選定一張圖作為參考基準點
      * 每一張圖都建立金字塔，由底部開始每往上一層長寬就各縮 小為1/2倍
      * 由最頂端，也就是縮到最小的圖開始，整個畫面移動9個方 向（包含維持不動）比對參考圖
      * XOR計算出差異最小的方向，並將這個位移結果往下一層傳， 重複比對步驟直到最底層
  + 降低對比:
    - 用Photographic Tone Reproduction for Digital Images:
      * 先對影像做Color Space轉換: RGB -> XYZ -> Yxy
      * 對Y作降低對比
        + 先計算出ln Y的平均a，將Y先縮小exp(a)倍，即Y’ = Y/exp(a)
        + 定義一個Threshold = th
        + 對每個點做一大一小的Gaussian Blur，得到V1, V2， 計算V = (V1-V2)/(x+V1)，x是一個可調整參數，避免 V1過小時發散
        + 增加Gaussian Blur的標準差，值到V超過th
        + 此時的V1應是與這個點相似的最大範圍的平均值
        + 計算Y\_low = Y’/(1+V1)
        + 此時Y\_low為大幅降低對比後之亮度
      * 再將影像轉回RGB即可得到視覺效果佳之影像
    - 用Fast Bilateral Filtering for the Display of High-Dynamic-Range Images:
      * 將HDR影像轉為灰階(Luminance)，作為影像的強度層 (Intensity layer)
      * 對強度層取10為底的對數得到 log Intensity layer
      * 將 log Intensity layer 和其做完Bilateral Filtering的結果相 減即可獲得影像的細節層(Detail layer)
      * 為了降低對比

**實作細節:**

* + 語言: python
  + 利用的module:
    - opencv for python
    - matplotlib
    - numpy
  + Python較C++來說較為高階，因此實作上較方便，加上opencv for python的底層一樣是用C++來實做，再加上numpy的使用，讓我們可 以對矩陣做快速的運算。因此用Python可以說是融合C++的速度，加 上python的實作方便性，還有Matlab一樣的矩陣運算，一舉數得。
  + 解Response Curve步驟中的Over-determined系統我們是利用opencv 的SVD來解，在tone mapping步驟中的Gaussian Filter也是使用 opencv的函數

**結果:**

* + 原圖：









* + Curve：
  + Photographic tone reproduction 結果：
  + Fast Bilateral 結果：
  + 利用Photomatix結果：
  + 時間：

**有實作的Bonus:**

* + 利用MTB做alignment
  + 利用兩種不同的方法做tone mapping