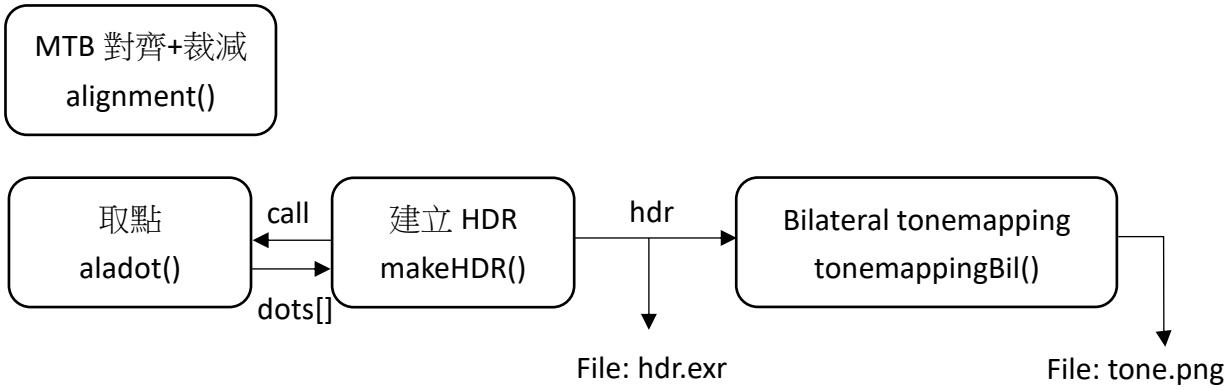
VFX project #1: High Dynamic Range Imaging 第 18 組 潘楷翔 吳沛昀

Bonus: 實作 MTB 對齊 5%

1. 流程 function



2. Function 說明

(1) alignment(self):

目的：每種曝光時間的圖片都對齊，裁掉沒重疊的部分後統一大小。

做法：

先把圖片依照曝光時間(ltime)排序，用 opencv cvtColor 轉成灰階。

MTB，依設定的 resize 次數(預設 5 次)從小到大 loop：

Scale 圖片(1/32 -> 1/16 -> 1/8… each loop)後取每張圖的百分級數 30 和 70 亮度值做為門檻，定義亮部是大於 70%門檻的點，暗部是小於 30%門檻。接著每兩張相鄰圖做九次位移 x[-1, 0, 1] * y[-1, 0, 1]，算圖 i 和圖 i+1 中分屬不同亮暗部的點。紀錄九次位移中不同點數量最小的位移存在 dx 和 dy 陣列，dx[i]表示第 i+1 張圖相對第 i 張的 x 位移。

五次 resize 跑完後，累加 dx dy 前綴並在最前增加一項[0]，從相鄰兩張圖位移量轉成以第一張圖為基準的位移量。每項減去最小值讓 dx, dy 存的資料 >= 0，轉成以位移量最小值的圖為基準。算出所有圖重疊的最大長寬(szx szy)後，每張圖 i 裁切[dx[i]:dx[i] + szx, dx[i]:dx[i] + szx, :]得到統一大小(szx, szy)的位移後圖片 。

(2) aladot(self, dot_num: int):

目的：回傳 dot_num 個適合做 HDR 分析的點。

做法：

建立兩種 bit mask，一個是附近(設定為 20*20)對比不大的 smooth_msk，另一個是保留中間亮度的 middle_msk(設定為亮度 84~170 的點)。找對比的 smooth_msk 在所有圖片上都必須成立，用 and；middle_msk 因為曝光時間會影響每張圖亮度，所以用 or。

找出滿足兩個條件的點存進陣列，shuffle 後取陣列前 dot_num 個點。

(3) makeHDR(self, filename: str = None, smooth_coustant: float = 100.0):

目的：重建 HDR

做法：

建立加權函數 wnp 和 w，依照和中間值 128 的差作加權。呼叫 aladot 取點後對每個顏色 channel 填入 A、B 矩陣求 g 函數。

每張圖 i 對所有點 j 取 w()得 w(Zij)，w(Zij)填入 A[(現在行數), Zij]作為乘上 g(Zij)的係數，-w(Zij)填入 A[(現在行數), 256 + Zij 的 index(i)]是 ln(Ei)係數。B 填上每張圖的 ln(曝光時間) = ltime，順帶一提 ltime 輸入時紀錄為 2 為底的指數。

前 n*p 行填完後下一行是 constraint，A[(現在行數), 127] = 10000，B 此行填 0，10000*g(127) = 0。

接下來 254 行填 smooth 部分，加權後的亮度 w(z)乘上 1, -2, 1 和 smooth_coustant(設定為 100)作為 g(z-1), g(z), g(z+1)的係數。

A、B 係數填完後用 numpy.linalg.lstsq 解最小平方和，x 的前 256 項存成 g 函數 g_func。最後對每張圖同一個點做加權減少 noise，得到 ln(Ei)後以 2 為底還原 Ei。

以上程序對三個顏色 channel 各做一次後得到 hdr。

(4) tonemappingBil(self, filename: str = None, smooth_contant: int = 20,
smooth_min: int = 0, smooth_max: int = 256,
fre_contant: int = 160):

目的：用 Bilateral filtering tonemap hdr 影像，寫入 filename

做法：

hdr 三個顏色取平均值得到 intensity，存在二維陣列 inten，原圖除以 inten 得到 color 保留著。

開始 Bilateral filter，定義位置權重 msk(dx, dy)和亮度權重 s(N, p)：

a. msk(dx, dy) = $\frac{1}{\sigma\sqrt{2\pi}} e^{-(\frac{dx^2+dy^2}{2\sigma^2})}$

標準差 σ 是 smooth_contant 的一半，也是 mask 邊長的 1/4 (預設為 smooth_contant = 20，mask 大小 41*41，dx, dy 範圍[-20, 20]是和中心點的相對位置)。

形成中心點為 1 的二維常態分佈。

b. s(N, p) = $e^{-10 \cdot (\frac{f(N)-f(p)}{f(p)})^2}$

p 是現在所在的點，N 是鄰近某個點。f(p)是 p 點在 hdr 圖上的能量。

對 hdr 上每個點 p、41*41 範圍內每個鄰居 N，做 msk((Nx-px), (Ny-py)) * s(N, p)能量加權得到 Large scale 陣列 inten_smooth，完成 Bilateral filter。

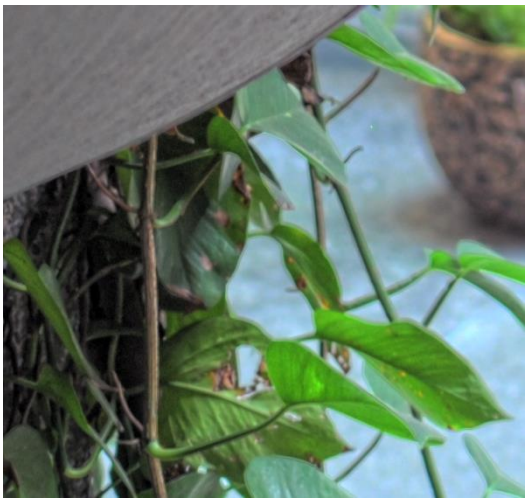
保存細節 inten_fre = (inten - inten_smooth) / inten_smooth。

降低 Large scale 陣列的對比，取 log 後減去最小值，(最大值, 最小值)重新 scale 到(smooth_min = 0, smooth_max = 256)。最後將細節乘上 fre_contant 參數(預設 160)，加回減少對比的 Large scale 陣列得到新的 intensity。intensity 乘上 color，再裁剪掉超過(0, 255)的資料就是輸出圖片。

3. Experiment and Comparison

(1) Tonemapping 參數 fre_contant=[256, 512, 1024, 128]

fre_contant 是細節乘回亮度時乘上的係數，原本預設 256。圖一到圖四分別是不同 fre_contant 的同一位置。



圖一(fre_contant=128)

圖二(fre_contant=256)

圖三(fre_contant=512)

圖四(fre_contant=1024)

觀察四張圖得知細節係數越大，物體描邊的現象越顯著，對比增加，顆粒 noise 也越強。實驗後決定將 fre_contant 預設改成 160。

(2) Tonemapping 參數[smooth_min, smooth_max]=[0, 256] / [0, 180] / [84, 256] / [84, 180]:

圖五([0, 256])

圖六([0, 180])

圖七([84, 256])

圖八([84, 180])

[smooth_min, smooth_max]是 Large scale 陣列取 log 後 map 到的亮度範圍，如圖七、圖六所示可以調高最小值點亮，或調低最大值壓暗整張圖。圖八的亮度範圍最小，造成對比小且有灰濛濛的感覺。