

DIP Term Project --- 微笑濾鏡

R04944053 張鈺萃

R05944019 張嘉豪

R05922130 王瀚磊

I. 動機

生活在這個 3C 產品興盛的時代，人們紀錄生活的方式也從傻瓜相機逐漸轉變成數位相機，由實體的相片走向數位相片。數位時代崛起後，開始有人發現可以在相片或者影片當中加上許多的特效，讓鏡頭裡的一切變得更有趣。

這次參與的期末project主題為微笑四方，要替微笑四方增加不同的可能性。微笑四方是一個兼具趣味性以及互動性的藝術品，我們認為最主要的互動為微笑時的蟲洞互動，然而我們發現在蟲洞連結時，顯示的影像並沒有加上太多的特效，而圖片、影片的特效正好是現今流行的應用。因此我們也希望能夠加上一些具有不同風格的濾鏡效果，讓人與人之間的互動變得更有趣，也希望這些特殊效果能夠讓站在微笑四方前的你，或會心一笑，或驚喜萬分，並且留下深刻的印象！

II. 實作

A. 哈哈鏡

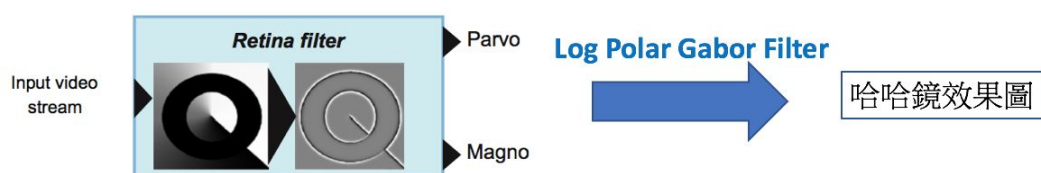


圖1：哈哈鏡效果流程圖

$$G_{ik}(f, \theta) = \frac{1}{\sigma\sqrt{2\pi}} \left(\frac{f_k}{f}\right)^2 \exp\left(-\frac{\ln\left(\frac{f}{f_k}\right)^2}{2\sigma^2}\right) \left(\frac{1 + \cos(\theta - \theta_i)}{2}\right)^{50}$$

f_k : GLOP filter中心的頻譜值

Θ_i : GLOP filter中心的頻率方向

f : 頻譜值

Θ : 頻率方向

σ : scale參數

Log Polar Gabor Filter (GLOP)公式

在哈哈鏡的部分，我們使用OpenCV中Retina的函數，當結果圖（

Parvo or Magno) 輸出時，會對其頻譜圖apply 一個Log Polar Gabor Filter，利用其特性即可使圖片產生類似哈哈鏡的效果。

B. 色鉛筆畫

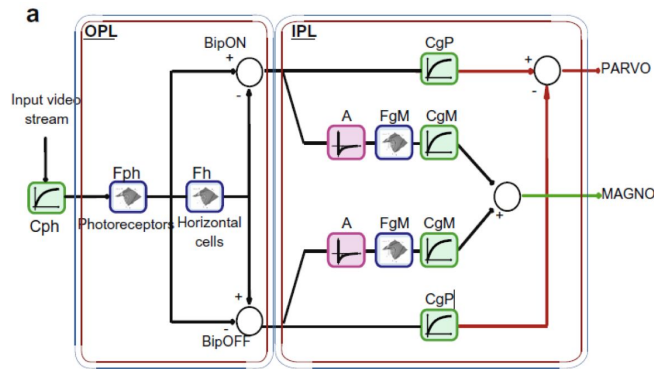


圖2:Retina類別中Channel示意圖

首先為了讓照片中的noise訊號在處理之後被放大，我們先apply一個Gaussian Filter，接著使用OpenCV中的Retina模組，先產生一個retina的類別後將照片輸入這個類別進行處理，在OPL時會先針對local contrast做第一次的調整，並且增強Contour資訊，接著在IPL時對Contour做第二次的強化。接著針對處理的內容設計適當的參數值，增強Local Contrast達成色鉛筆的效果，將參數photoreceptorsLocalAdaptationSensitivity設定為0.9（範圍：0 - 1），由於處理過程內容包含兩種Channel：Parvo、Magno且兩個的處理時間都不短，因此我們會先關掉Magno的Channel只取Parvo進行輸出，最後得到結果圖。

C. 油畫

油畫是採用渲染的方式做的，針對每一個 pixel 會有一個矩形的範圍，在範圍內分別計算RGB的值還有範圍內的 pixel 個數，最後再以範圍內RGB的平均強度加上原來的值來當作這個 pixel 的值，用平均強度來讓畫有模糊且有油畫一層一層疊起來的感覺出現。

D. 卡通



圖3:卡通風格流程圖

在模擬卡通效果的過程，我們使用OpenCV中stylization函數，調整sigma_s參數可產生不同程度的blur、sigma_r參數要選擇neighborhood的平均顏色，最終造成水彩的效果，而且同時帶有edge enhancement的效果，但是我們覺得不是我們想要的卡通效果，因為平坦的部分處理的不是很好，後來我們決定將結果圖與原圖進行alpha-blending，以達到最後

的效果。

E. 拼貼畫

拼貼畫的效果主要在於兩個部分，一個是影像分割，另一個是給予每個區域一個顏色。

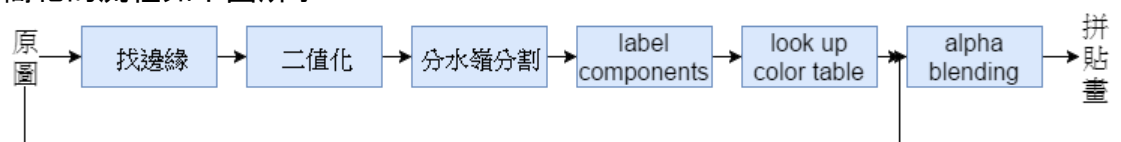
影像分割：我們先將原影像灰階化，爾後採用Sobel找水平和垂直的邊緣，再利用morpholog的opening和closing運算，讓輪廓變得更平滑，我們先執行opening讓比較纖細的連結斷開，分離兩個連接不緊密的物體，還可以消除物體外圍的細毛；之後再進行closing，填補一些細小的空洞，連接相鄰的物體。因為經過上述的過程得到的是一張灰階影像，必須先將影像二值化，而我們希望影像能自動找出適合自己的threshold，因此我們剛開始使用Otsu演算法，但是因為環境光線不均的問題，所以影像每個區域的灰階值基準不同，很難找到一個threshold能適用於整張影像，所以最後效果並不如預期。後來我們改用adaptive thresholding，可以將影像分成幾個區域，每個區域有各自的閾值，再分別將各個區域進行二值化，如此一來就可以得到一張輪廓明確的二值影像。最後我們利用watershed做image分水嶺分割，將輪廓圖再細分成很多個小區域，就可以得到我們的拼貼草圖。

填顏色：我們先label每個components，然後建立一個color lookup table，將每種label對應到一種顏色：若label是0者表示其為分水嶺，因此給予255的color value，其餘具有相同label的區域片段都給相同的顏色，其最後的結果如下圖所示。雖然拼貼畫的效果已經出來的，但是我們覺得畫風有些過於抽象，所以最終決定將這個結果和原圖做alpha-blending，得到了兼顧抽象和真實的作品。



圖4:抽象的拼貼畫風格

簡化的流程如下圖所示：



F. 新海誠風

我們首先調整原圖的飽和度，利用OpenCV `cvtColor`的function，轉到HSV色彩空間，然後做saturation維度的equalization，最後再轉回BGR的色彩空間，得到圖6。再來是加強邊緣，利用Sobel找出邊緣，然後再利用Ostu演算法的二值化，將pixels反轉成一張有白底黑邊的底圖(圖7)，最後只要把增加飽和度的圖和底圖blending在一起，就可以得到圖8，但是飽和度因為blending的原因所以下降了一些，所以我們最後再調整了飽和度，得到最後的結果(圖9)。



圖6

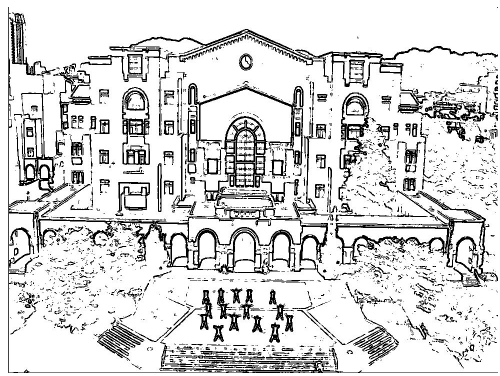


圖7



圖8



圖9

之前有把前景割出來貼到背景，我們採用seamless clone的方法，但是需給予一個mask，為了要能用在影片中，所以mask必然需要能自動產生，因此我們先利用canny找邊緣，然後用形態學的方法把一些輪廓找出來，最後trial and error後，得到一個還不錯的mask，可以剛好把圖書館(前景)切出來。

G. 漣漪

在模擬水波的過程，我們採用了sin函數來模擬。假設現在有一張圖片，針對圖中的每一點P，先計算出P對圖片中心點C的距離，再利用

sin 函數以及預設好的振幅係數A 和頻率係數 B，計算出新的位置 N，接著將 N 對稱於 $y=C_y$ 找到 N'。此時，原本的 P 點在經過轉換後會變到 N'，而顏色的計算是利用 Bilinear的方式。

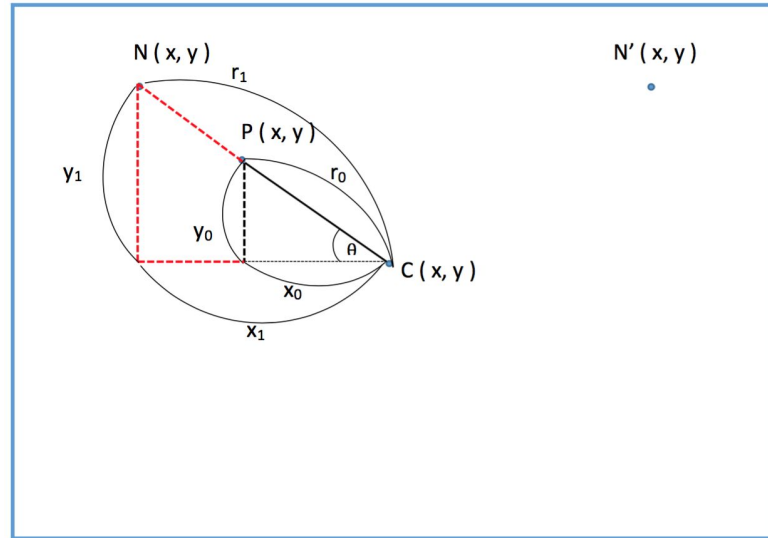


圖10: 漣漪示意圖

$$x_0 = C_x - P_x \quad (1)$$

$$y_0 = C_y - P_y \quad (2)$$

$$theta = \tan^{-1}(y_0 \div x_0) \quad (3)$$

$$x_1 = r_1 \times \cos\theta \quad (4)$$

$$y_1 = r_1 \times \sin\theta \quad (5)$$

$$r_0 = \sqrt{x_0^2 + y_0^2} \quad (6)$$

$$r_1 = r_0 + A \times \sin(B) \quad (7)$$

$$N'_x = C_x + x_1 \quad (8)$$

$$N'_y = C_y + y_1 \quad (9)$$

C_x : 圖片正中間像素C的X座標

C_y : 圖片正中間像素C的Y座標

P_x : 像素P的X座標

P_y : 像素P的Y座標

N_x : 經過sin運算後的X座標

N_y : 經過sin運算後的Y座標

N'_x : N對稱於Y軸的像素N'的X座標

N'_y : N對稱於Y軸的像素N'的Y座標

A: sin波的振幅變量因子

B: sin波的頻率變量因子

theta: P到C的夾角

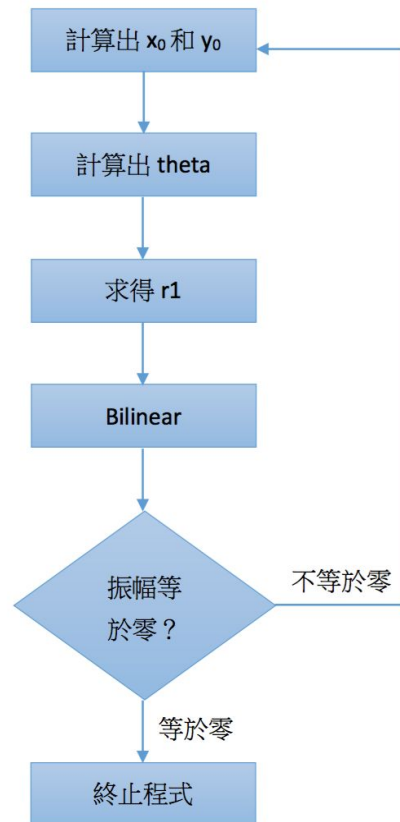


圖11:: 漣漪計算流程

III. 結果與討論

A. 哈哈鏡



圖12: 哈哈鏡效果

1. 邊緣的白色效果：當影像做完Log Sampling後，雖然成功作出圓形視覺效果，卻可以發現在邊緣的部分會有一圈明顯的白色效果。
2. 雜訊：外圍的圓圈可以看到有些pixel value的值並不正確，原本旁邊應該為整面牆壁的部分受到一些noise影響。
3. 即時效果：在opencv實作中，這類module的應用非常耗時，無法達成有效的即時效果。

B. 色鉛筆畫



圖13: 色鉛筆效果

1. 色彩對比不足：當在做完預處理後使用opencv的module實作後，會將原先圖片的對比降低，雖然加強了local contrast的效果整張圖卻依然感覺很暗沈，應該再加入Gamma Correction調整整張圖的明亮程度。
2. 陰影：仔細觀察陰影的部分如樹木底下，可以發現有點blur的效果，使得影子看起來並不真實。

C. 油畫



圖14: 油畫效果

1. 影片版中有閃爍點：光點和倒影的部份若是移動，在影片中會造成閃爍點的問題。
2. 在Contour較密集的部分如樹葉密集處會使得油畫效果不明顯，真實的油畫效果應該會達到一片一片暈開的感覺，在前面幾組的報告也有同學提到並解決這類問題。

D. 卡通



圖15: 卡通效果

1. 後方的建築物過於blur：
若是原圖中有很多平坦的部分，但有少量的細節成份，可能會無法讓兩個地方視覺上的blur程度達到一致的效果。
2. 改進的方法：
後來我們覺得可以直接利用local gaussian blur來模糊細節和sobel filtering來留下edges，過edge enhancement，再和原圖進行alpha-blending，可能會更趨於卡通的效果。

E. 拼貼畫



圖16: 拼貼畫效果

1. 不適用於影片版：
 - a. 因為當鏡頭晃動時，整張圖的輪廓也會跟著改變，色塊也會隨之改變，所以會造成令人不舒服的跳動。
2. 不以色塊構成：
 - a. 原本是建立一個color lookup table，給予每個區域同一個顏色，但是看起來效果不佳，會有一些髒的顏色出現，可能要再改進每個區域顏色挑選的演算法。
 - b. 後來將結果和原圖做alpha blending後發現效果出乎意料的好，看起來就跟拼貼畫一樣，還有一種像碎玻璃的效果。

F. 新海誠風



圖17: 新海誠風格

1. 色彩不夠鮮豔：
 - a. 因為用OpenCV實作的seamless clone function，將前景(圖書館)和背景(有雲的藍色天空)接在一起，會導致前景(source)的顏色會受背景(destination)的顏色影響，導致整張圖會呈現藍藍的感覺。
 - b. 若是做post-processing去調整整張圖的飽和度或者色相，可能會讓雲的部分失去動畫"你的名字"中的感覺，所以我們也不採用這個方法。
2. 圖書館的白色部分有一點雲的痕跡被合在一起：
 - a. 要將此風格放在影片中，所以給seamless clone的mask要自動產生才行。
 - b. 產生mask的方式是利用我們自己寫的演算法進行trial and error調整參數，所以mask沒有處理得很好，就會把比較大的平坦部分也當作是天空，而留下少量雲的成分在畫面上。
3. 改進方法：因為我們使用的方法是把圖書館切下來貼到天空上，這樣如果要使用seamless clone就會產生上述的問題，所以可能可以改成把天空影像clone到原本的圖書館影像的天空上，就可以避免改變顏色和前景太平坦的部分也會出現雲的問題。而mask的產生方式可以改成利用deep learning來學習天空，再將原始影像(圖書館的圖)中的天空部分mark起來，將天空圖crop成mask的形狀，clone到原始圖的天空區域。

G. 漣漪

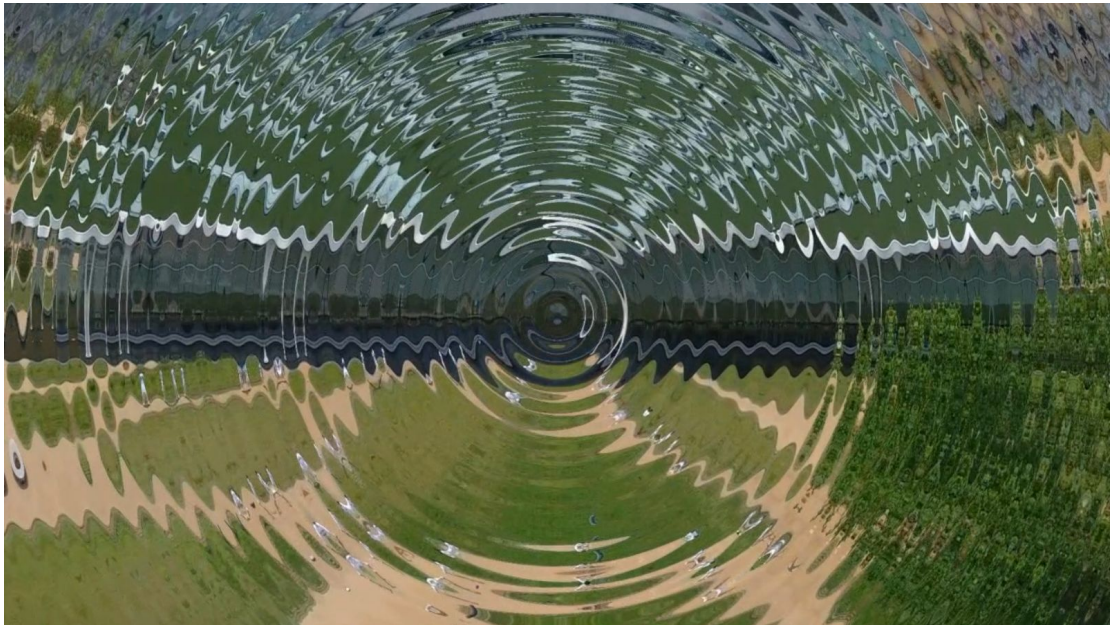


圖18: 漣漪效果

1. 水波結束要變平坦的時候較不自然：
 - a. 現在的做法是每一個 frame 的時候會調整 sin 的振幅和頻率，適用等差的方式讓水的波紋有向外擴散的效果。但是利用等差的方式，在最後要趨於平坦的時候會有些不自然。之後會嘗試用高斯曲線遞減，而不是等差的方式。
2. 即時效果：
 - a. 原本以為即時動作會處理的很慢，但是很意外地並不會，在影片每個 frame 處理的過程中也很迅速，若是要在蟲洞使用這個效果感覺也會不錯。

IV. 結論

經過這次的專題後，發現原來在影像中可以做到這麼多的效果呈現。每個效果都讓人有耳目一新的感覺，不同的效果在實作中遇到的困難自然也不一樣。有的時候為了追求逼真，反而會犧牲掉許多的效能。

V. 未來展望

A. 即時處理

除了微笑四方外，現在有很多的通訊軟體都有即時的互動，如果能夠在這些即時的互動當中，加入一些特效，也是一件很有趣味的事呢！我們實作出來的效果，有些會處理得很慢，未來我們希望可以好好思考如何增進效能，讓這些好玩的特效可以即時展現！

B. 改善新海誠seamless clone的顏色會改變的問題

第一個改善的方法，就是只處理 seamless clone 的周圍一定範圍，不不要一直往裡面做內積，這樣可以讓周圍與目標的圖片不會產生太大的違和感，又可以保有 clone 物體原本的顏色。

第二個改善方法，就是盡量讓背景的色調一致，這樣做完 seamless clone 之後顏色不會被影響的太多。

C. 改進哈哈鏡&色鉛筆的缺點

基本上都是採用相同的影像處理方式，只差在多做一次Log Polar Gabor Filter，因此他們都存在著顏色明亮度不足以及雜訊的問題，未來可以針對他們去設計一些去除陰影雜訊的filter並且加上正確的亮度校正後達成更好的效果。

D. 卡通效果的改進

雖然我們在stylization後為了符合卡通圖案邊緣較粗的特色，我們將邊緣增強後再進行疊合，然而在人物的部分邊緣效果過於強，我們認為可以再針對像圖片中人物這些叫細微的項目，對其內部做一些flattening的處理，使得人物的衣服可以達到整片白色而不是穿插著許多粗黑線條。

E. 新海誠色彩飽和度問題：

之前是利用YCrCb的Y維，藉以調整lumination，但發現結果的飽和度不佳，所以我們後來有再改成從HSV的色彩空間來調整Saturation，而結果也已呈現在之前的實作部分。

VI. 展示影片連結

<https://drive.google.com/file/d/0B2yFRocnz6U9Y1ZUUmT0X1NzZkE/view?usp=sharing>