影像處理 Term project report

一、動機

我們的主題是結合 Content-Aware Rotation、Color transfer 以及 Transportation Map Regularization(TMR)，許多影像因為拍攝角度不對或是其他因素而導致影像內的物體是歪斜的，所以需要用 Content-Aware Rotation 進行校正，另外 Color transfer 可以被應用在很多方面，其中幾乎所有人都有用過的就是濾鏡修圖，最後 TMR 可改善 Color transfer 後帶有瑕疵的影像。基於此，我們決定做這三種主題。

二、方法

Content-Aware Rotation:

為了能夠有 content-aware 的效果，第一步必須先將圖中線條找出來，這方面可以直接套用 open source 的套件去處理即可(Line Segment Detector)。

而這些線條其實只是為了要給如何對圖片進行轉換提供一個方向，實際上要將原圖等分成數百個小格子(mesh)，所有轉換都是以格子為單位進行轉換，使用 bilinear interpolation 的方式去扭曲原本的格子。

由於照片中有些東西本來就該是斜的，如我們本次實驗的對象比薩斜塔，不能盲目地把所有斜線都調成水平線/垂直線，因此需要由使用者輸入一個他想轉動的角度，令這個角度為 Δ，將所有找出的線條根據他們對水平線的角度定義為$[-\Delta, \pi-\Delta]$，並區分成 90 個 bins，每個 bin 涵蓋兩度的範圍的線條。這麼做的目的是，我們可以知道第一號的 bin（角度為$[-\Delta, 2-\Delta]$）在經過轉換後應該要是水平線，而第 90 號的 bin(角度為$[\pi-\Delta-2, \pi-\Delta]$）在經過轉換後應該要是垂直線，而我們就可以在計算 energy function 時對每個 bin 有不同的算法，例如特別關照這兩個 bin 裡的線條，迫使他們轉換完確實是水平線/垂直線。

做完這些前處理後，就可以開始定義 energy function，也就是事後要 optimize 的對象。這部分照著原 paper 的做法，總共定義了 4 個不同的 energy function 來完成不同的目的，最後再把這 4 個加起來以得到 total energy function，然後再對 total energy function 做 optimize。4 個 function 分別為：

1. Rotation Manipulation: 這個 function 會鼓勵第一個 bin 轉完後變水平線，而第 90 個 bin 轉完後變垂直線。至於其他的 bin，則會鼓勵編號相鄰

的 bin 轉換的角度盡量一樣。

2. Line Preservation: 將所有線段的 2 個端點視為是 4 個點做 bilinear interpolation 後的結果，目的在建立起前面用 LSD 找出的線段與劃分出來的 mesh 之間的關係。
3. Shape Preservation: 讓每個 mesh 經歷的轉換都盡量一樣，目的是要避免圖片被過度扭曲，改變原有的內容物的形狀。
4. Boundary Preservation: 如上所述，做這個 content-aware rotation 的目的即是要避免圖片的邊邊在旋轉的過程中被裁掉，因此要最大程度地保留邊邊的影像內容

定義完這些 function 後，就可以去跑迴圈做 optimize。與 machine learning 的 application 一樣，要跑幾個 iteration 可以由使用者自己定，而我們實驗的結果是跑 10 個 iteration 在幾乎所有圖片上都能有很好的效果。跑完之後的輸出檔就是每個 mesh 在經過轉換後，其四個端點的新座標，如此便可以透過 bilinear interpolation 去將每個 mesh 作轉換。

Color transfer:

我們會使用兩張影像，其中一張是要擷取色調的影像，稱作 source，另一張則是我們想要轉換色調的影像，稱作 target，其步驟如下：

1. 利用以下公式將兩張影像先從 RGB space 轉到 LMS space，再轉到lαβ space。

$$\begin{bmatrix} L \\ M \\ S \end{bmatrix} = \begin{bmatrix} 0.3811 & 0.5783 & 0.0402 \\ 0.1967 & 0.7244 & 0.0782 \\ 0.0241 & 0.1288 & 0.8444 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix},\ \begin{bmatrix} \mathbf{L} \\ \mathbf{M} \\ \mathbf{S} \end{bmatrix} = \begin{bmatrix} \log(L) \\ \log(M) \\ \log(S) \end{bmatrix},\ \begin{bmatrix} l \\ \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{3}} & 0 & 0 \\ 0 & \frac{1}{\sqrt{6}} & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & -2 \\ 1 & -1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{L} \\ \mathbf{M} \\ \mathbf{S} \end{bmatrix}$$

2. 計算兩張lαβ影像各個 channel 的平均值以及標準差後進行以下公式之運算。

$$l_t^* = \left(l_t - \bar{l}_t\right)\frac{\sigma_s^l}{\sigma_t^l} + \bar{l}_s,\ \alpha_t^* = (\alpha_t - \bar{\alpha}_t)\frac{\sigma_s^\alpha}{\sigma_t^\alpha} + \bar{\alpha}_s,\ \beta_t^* = \left(\beta_t - \bar{\beta}_t\right)\frac{\sigma_s^\beta}{\sigma_t^\beta} + \bar{\beta}_s$$

3. 將計算過後的 target 影像從lαβ space 經由以下公式轉回 RGB space。

$$\begin{bmatrix} \mathbf{L} \\ \mathbf{M} \\ \mathbf{S} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & -1 \\ 1 & -2 & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{3}} & 0 & 0 \\ 0 & \frac{1}{\sqrt{6}} & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} l \\ \alpha \\ \beta \end{bmatrix},\ \begin{bmatrix} L \\ M \\ S \end{bmatrix} = \begin{bmatrix} \log(\mathbf{L}) \\ \log(\mathbf{M}) \\ \log(\mathbf{S}) \end{bmatrix},\ \begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 4.4679 & -3.5873 & 0.1193 \\ -1.2186 & 2.3809 & -0.1624 \\ 0.0497 & -0.2439 & 1.2045 \end{bmatrix} \begin{bmatrix} L \\ M \\ S \end{bmatrix}$$

另外，為了解決 global transfer 造成色調轉換錯誤的現象，我使用 local tranfer 的方式，利用lαβ space 中數值差異較符合實際感知差異的特性來對影像分群，先從 target 影像手動挑選幾個參考區域並計算該區域的lαβ，然後對每個 pixel 計算與各區域的歐氏距離，D_i值最低的就會屬於 i 區域。

$D_i = \sqrt{(l - \bar{l}_i)^2 + (\alpha - \bar{\alpha}_i)^2 + (\beta - \bar{\beta}_i)^2}$　　i：參考區域

最後從 source 影像手動挑選要轉換的色塊，並將先前在 target 分的區域對應到 source 挑選的色塊進行 color transfer。

Transportation Map Regularization:

論文以 Non-local filter 的原理為出發點，並使用 Yaroslavsky filter 的公式進行推導，最後整理出以下公式作為進行 TMR 的實作:

$$TMRu(T(u)) = Yu(T(u)) + u - Yu(u)$$

其中 u 代表原圖、T(u)為經過 color transfer 的圖片、$Y_u(u)$為經過 Yaroslavsky filter 的圖片，整個公式原理可以理解為將過濾後的 Color transfer 圖片加上原圖的細節。並且 $Y_u(T(u)) - Y_u(u) = Y_uM(u)$這個值會收斂到一個定值，所以在實作中會讓一個像素點跑多次迴圈，直到 $Y_uM(u)$數值無法再收斂，也因為每一個像素點收斂的速度不一樣，因此每個像素點都會進行不同次數的 TMR。

因為論文以 Non-local filter 的原理為基礎，以及我對 Yaroslavsky filter 並不了解，網路上相關資料也不多，於是我決定使用教授在上課中講過的 Non local mean filter 進行實作。在這邊我自己實作了一個 Non local mean filter 的函式，讓我能夠自行調整 Searching kernel, neighborhood kernel, gaussian kernel variance 以及 Degree of smoothing 的數值，如此也能夠對單一像素點進行過濾達到上面提到的需求，這是 matlab 內建函式無法做到的。

實作過程先將圖片從 RGB space 轉到 CIE 1976 L*a*b* space，這邊使用 matlab 內建函式達成，接著就是照著 TMR 的原理進行實作，並且用迴圈對每一個像素點進行不同次數的 TMR 處理。Degree of smoothing 是先找出整張圖每個像素與原點(強度為(0, 0, 0))的歐氏距離，再計算像素點間的標準差得到的，其他參數則為手動調整。

三、結果

Content-Aware Rotation:

Before

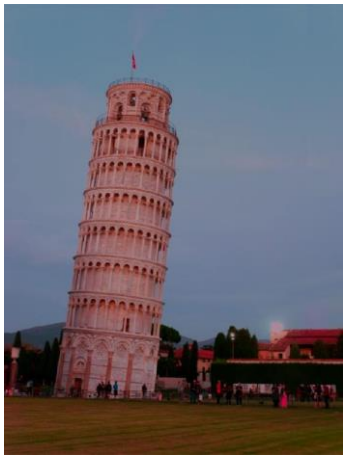
After

由以上的實驗結果可以看出，在經過轉換後，原本是斜的草地、屋頂、樹叢等等都變為水平的，但比薩斜塔仍是歪斜的，也就是我們的程式並不會盲目地將所有斜線調成水平線/垂直線。此外，圖片的邊角都沒有肉眼可辨識的裁切。

Color transfer:

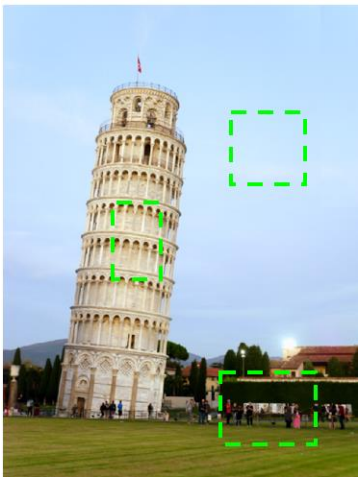
target source global transfer

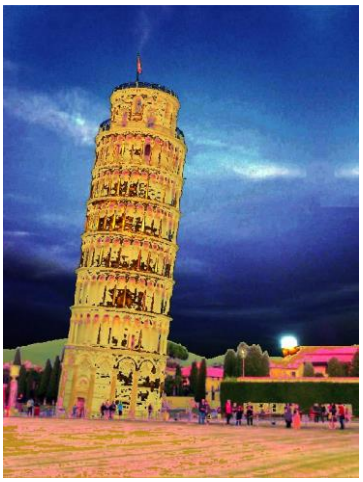




如上圖，如果我們直接使用整張影像的平均值以及標準差進行運算，轉換後的風格會和 source 完全不一樣，這是因為 global tranfer 並不能反應到各個區域的色塊，有點像是將影像上的顏色都混在一起的感覺，因此我改用 local transfer 的方式，如下圖，其中 target 和 source 中的方框就是我手動挑選的區域，target 的天空會轉換成 source 的天空的色調，斜塔會轉換成房子的色調，而建築物以及草地則是地板。

target source local transfer

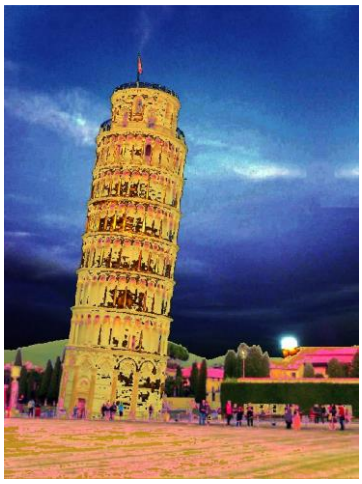
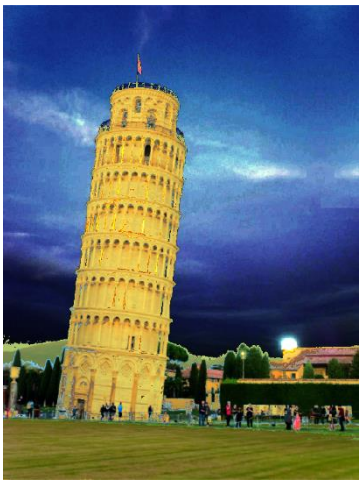




Demo 時展示的轉換結果會發現斜塔跟草地都有斑點，這是因為這些斑點在還沒轉換時被分類到錯誤的區域上，例如斜塔的斑點處被分類成和天空同一群；而草地不僅是分類問題，因為草地的顏色較複雜，其中包含了綠色、土色以及少許的紅色，所以轉換後顏色會有違和感，因此我在原始 target 影像中斜塔有斑點的位置又選取一小範圍的區域作為參考區域，並同樣轉換成房子的色

調，另外也選取草地作為參考區域，不過保留草地原始的色調不做轉換，其結果如下

改善前　　改善後

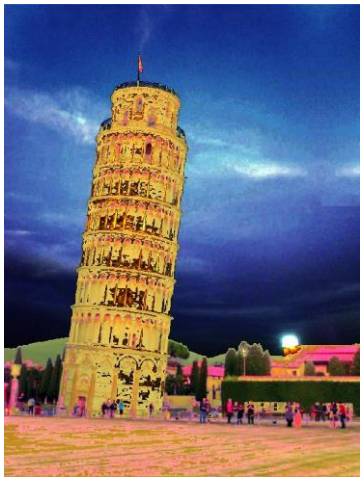



經過改善後塔的斑點幾乎不存在，另外草地在沒有經過轉換的情況下反而看起來比較自然。

Transportation Map Regularization：

由於對每一個像素點都要進行多次的 TMR，整個運行過程花費很長的時間，在效果與速度的權衡下，我把 Searching kernel 大小設為 21，Neighborhood kernel 大小設為 5，Gaussian kernel variance 設為 1.4，以下為 TMR 得到的結果與僅有 Color transfer 的結果比較圖:

僅經過 Color transfer　　經過 TMR 處理




可以看出 TMR 很順利的將斜塔、草地以及人的細節彌補了回來，天空中的顆粒感被消除，邊界也變得比較自然。不過斜塔內部的暗色斑點依然存在，這種 Color transfer 產生的色塊瑕疵僅靠 TMR 是無法解決的。同時也有注意到斜塔的邊界有明顯較亮的區塊，我認為是因為我直接計算整張圖的 Degree of smoothing 導致部分雜訊較少的地方被過度過濾。因此在未來改善的部分我提出如果把對整張圖計算 Degree of smoothing 改成僅對當下的 Searching kernel 區塊

進行計算應該可以解決這個問題，不過會讓運行時間大幅增加。

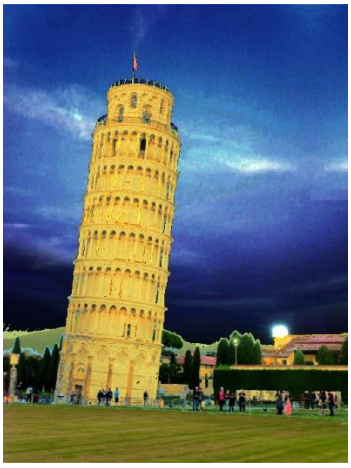
在 Demo 後我將程式稍作修改後得到了計算 local degree of smoothing 進行 TMR 的結果，與舊的方式比較如下:

整張圖計算 Degree of smoothing　　計算 local degree of smoothing


可以看到這兩張圖沒有任何差別，因此假設是失敗的。調整其他的參數例如 Neighborhood kernel 以及 Gaussian kernel variance 的數值大小也無法解決這類問題。

最後是附上 Color transfer 改善之後的結果:

僅經過 Color transfer　　經過 TMR 處理




四、分工內容

Content-Aware Rotation：107060008 張承勛

Color transfer：107011254 陳沛騏

Transportation Map Regularization：107011225 林致維

五、參考資料

1. E. Reinhard, M. Adhikhmin, B. Gooch and P. Shirley, "Color transfer between images,"

in *IEEE Computer Graphics and Applications*, vol. 21, no. 5, pp. 34-41, July-Aug. 2001, doi: 10.1109/38.946629.

2. J. Rabin, J. Delon and Y. Gousseau, "Regularization of transportation maps for color and contrast transfer," *2010 IEEE International Conference on Image Processing*, 2010, pp. 1933-1936, doi: 10.1109/ICIP.2010.5650823.
3. K. He, H. Chang and J. Sun, "Content-Aware Rotation," *2013 IEEE International Conference on Computer Vision*, 2013, pp. 553-560, doi: 10.1109/ICCV.2013.74.