

Intro. Image Processing

LAB 1

謝侑哲

112550069

1. Method

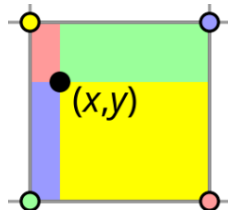
A. Interpolation

(1) Nearest Neighbor Interpolation

Nearest Neighbor 的方法實作來非常簡單，只需要把計算出來的座標做四捨五入，就可以得到最近的點，取其值即可做到最近鄰插值。另外對於邊界問題也做了處理，採用 zero padding 的方式:

(2) Bilinear Interpolation

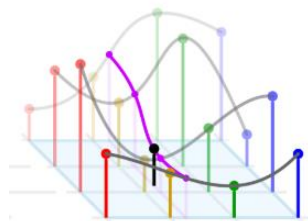
Bilinear Interpolation 在做完公式的簡化之後，可以發現對於周圍四個點的插值權重，就是該點越過目標點的對面那塊面積。如圖所示:



因此實作上，只需要計算 4 點座標與目標點形成的區域面積，即可得到不同點的顏色的權重。另外，對於邊界的插值，我採用 zero padding 的方。

(1) Bicubic Interpolation

Bicubic Interpolation 使用類似於 bilinear 的方式做成，但透過取 4×4 個點的方式，每次用 4 個點求解一個三次函數，再利用這個三次函數去近似某一點的值。實作上，對 x 軸做 4 次插值後，得到 4 個在同一條 y 軸線上的點，再去對這 4 個點做插值，即可得到參考了 16 個點的插值。如圖:



B. Deformation

(1) Rotation

Rotation 的實作基於線性代數中的 rotation matrix。不過線性代數中的 rotation matrix 是對於向量的旋轉，要把他應用到圖片像素的旋轉時，因為圖片是一個整體，

為了保持 pixel 的相對關係，可以將其視為在三維空間繞 z 軸旋轉一個完整物體。因此把 rotation matrix 擴展如圖：

$$\mathcal{R}_z(\theta_z) = \begin{bmatrix} \cos \theta_z & -\sin \theta_z & 0 \\ \sin \theta_z & \cos \theta_z & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

另外，因為單純套繞 z 軸的 rotation matrix 會繞(0,0)座標旋轉，但我們想要繞圖片中心旋轉，所以以中心當原點變成(x-cx, y-cy)重算座標，因此矩陣前 2 row 的後面加上把 cx, cy 項減去的值。

```
rotate = np.array([[np.cos(angle), -np.sin(angle), cx - cx * np.cos(angle) + cy * np.sin(angle)],  
                  [np.sin(angle), np.cos(angle), cy - cx * np.sin(angle) - cy * np.cos(angle)],  
                  [0, 0, 1]]) # rotate according to the center points
```

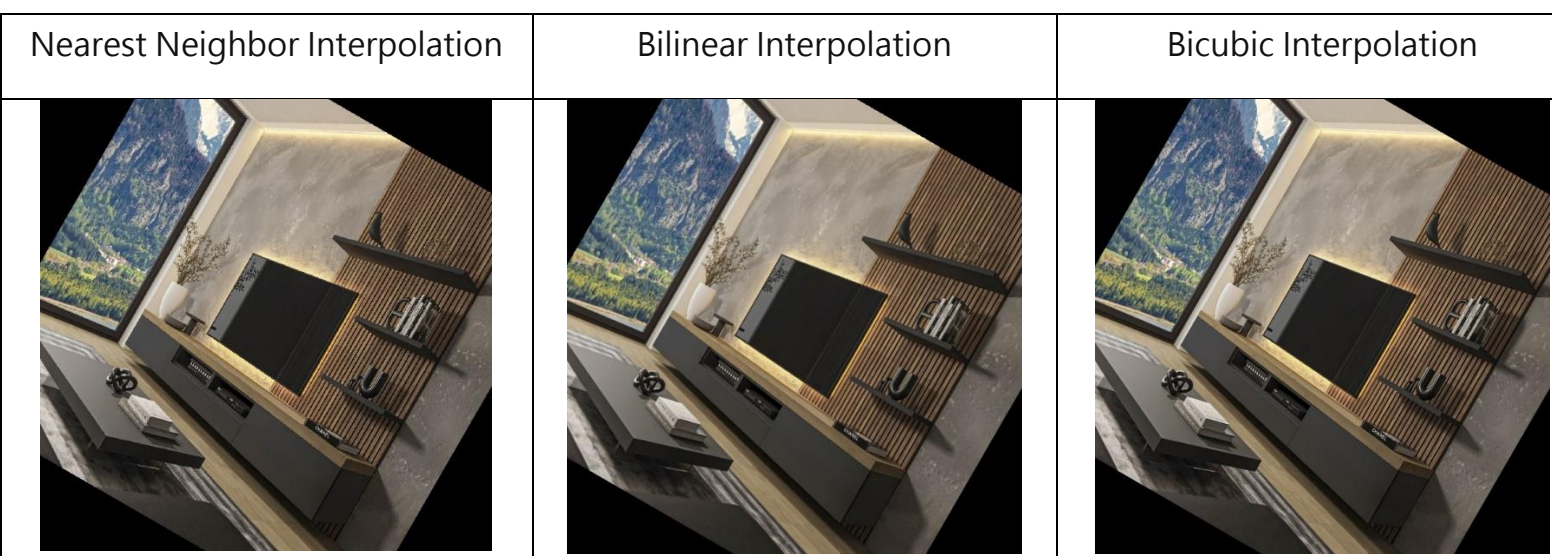
得到 rotation matrix 後，因為從原圖直接計算 rotation 後的點，會有浮點數而產生 mapping 位置模糊的問題。所以實作上，從處理後的圖片的像素開始，用反矩陣回推原圖的位置，再套上前面提到的 interpolation 方法插值。

(2) Wrapping

Wrapping 是可以把平面上任意點映射到另外一點的方法，透過計算 homography 矩陣 H。可 $HA=B$ 的映射。同樣的，如果要把整張圖做形變映射，只要形變後的至少 4 個點(才能有解)，與原本圖片的對應點。就可以近似出共通的 homography 矩陣。再利用跟 rotation 同樣的方法，從目標圖透過反矩陣回推原圖位置，搭配插值即可做到 warpping。

2. Result

A. Rotation



Interpolation type	Nearest Neighbor	Bilinear	Bicubic
Time consume (s)	9.5122	12.7305	48.4914

B. Warpping



Interpolation type	Nearest Neighbor	Bilinear	Bicubic
Time consume (s)	9.2872	9.4669	11.9943

從圖片品質的方面看，在 Rotation 與 Warping 套用不同插值方式的圖片中。可以看出 nearest neighbor interpolation 的方法做出來的圖片，會有明顯的雜訊感，直接取一點的值來用讓插值效果不夠平滑。Bilinear 與 Bicubic 的方法在圖片上就明顯的平滑很多，圖像品質更好。而 Bicubic interpolation 得到的圖片，因為參考了 16 個點，參考的資料量更多，因此得到的圖片保留的細節更多。像是在 rotation 中，窗戶中樹林及電視旁的植物，在 Bicubic interpolation 下，輪廓跟陰影、對比被保留的程度比起 Bilinear 更多，兼顧了平滑卻不會模糊。另外，Bilinear interpolation 在邊界的一些地方會有明顯的鋸齒感，但 Bicubic interpolation 就比較不會有這個問題。

從運行速度的方面看，Bicubic 在運行上會花費大量的時間，在 rotation 任務中，甚至是 bilinear 的 4 倍。因此對於需要更快速度的任務，像是 real time 影像處理等，選用前 2 種方法可能是更好的選擇，而沒有時間限制的情況下，可以選用 bicubic 來達到最好的圖片品質。

3. Feedback

本次課程中，讓我了解到修圖背後的原理與實作，看到圖片的變化感覺非常有成就感，發現了線性代數應用在影像處理的有趣之處。