謝侑哲

**Intro. Image Processing**

**LAB 1**

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:

1. **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:

一張含有 螢幕擷取畫面, Rectangle, 行, 圖形 的圖片

AI 產生的內容可能不正確。

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:

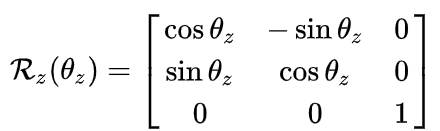
一張含有 行, 鮮豔, 藝術 的圖片

AI 產生的內容可能不正確。

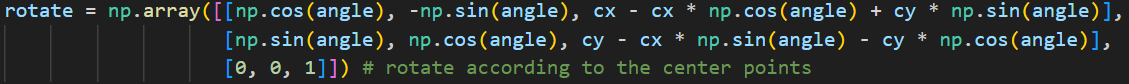
**B. Deformation**

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:



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。



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。

1. **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**

1. **Rotation**

|  |  |  |
| --- | --- | --- |
| Nearest Neighbor Interpolation | Bilinear Interpolation | Bicubic Interpolation |
|  |  |  |

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

**B. Warpping**

|  |  |  |
| --- | --- | --- |
| Nearest Neighbor Interpolation | Bilinear Interpolation | Bicubic Interpolation |
|  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| 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neareast 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。