**Image Processing Homework 1 Report**

**P76094282 方郁文**

**使用語言 C#**

**RGB Extraction & Transformation**

**Problem**

把 RGB 的圖片轉換成只有單獨的R、G、B的Channel以及Grayscale

**Method**

將RGB的圖片讀入Bitmap，分別取出其R、G、B 的Channel；Grayscale圖片的轉換參考以下公式



Reference: [opencv documentation](https://docs.opencv.org/3.4/de/d25/imgproc_color_conversions.html)

**Results**

Origin Image



R Channel G channel



B Channel Grayscale



**Smooth filter (mean and median)**

**Problem**

利用Smooth Filter把圖片中的噪點移除

**Method**

原圖先經過Padding 1後進行下列filter

Mean filter

|  |  |  |
| --- | --- | --- |
| 1/9 | 1/9 | 1/9 |
| 1/9 | 1/9 | 1/9 |
| 1/9 | 1/9 | 1/9 |

對原圖filter大小內的pixel進行取平均 ( Mean )

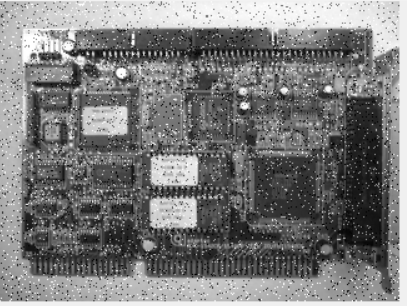
Median filter

|  |  |  |
| --- | --- | --- |
| a | b | c |
| d | e | f |
| g | h | i |

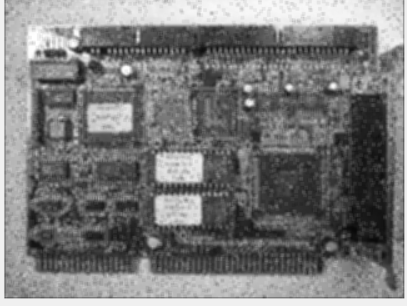
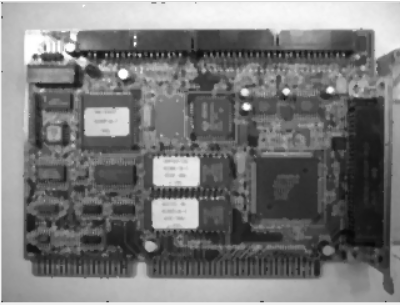
將原圖 a~f 的pixel進行sorting，取出中位數 ( Median ) 後，把原圖filter大小內的pixel設為中位數。

**Results**

原圖



Mean filter Median Filter

**Discussion**

Mean filter會讓圖片變模糊，Median Filter 圖片較清晰

Median Filter 對於消除噪點效果較好

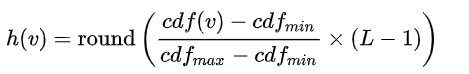
**Histogram Equalization**

**Problem**

Implement histogram equalization

**Method**

計算input image中每個pixel的grayscale值出現的頻率，把計算出每個pixel出現的pdf和cdf，最後套入下列公式

where L = 256

Reference: [wiki](https://zh.wikipedia.org/wiki/%E7%9B%B4%E6%96%B9%E5%9B%BE%E5%9D%87%E8%A1%A1%E5%8C%96)

**Results**

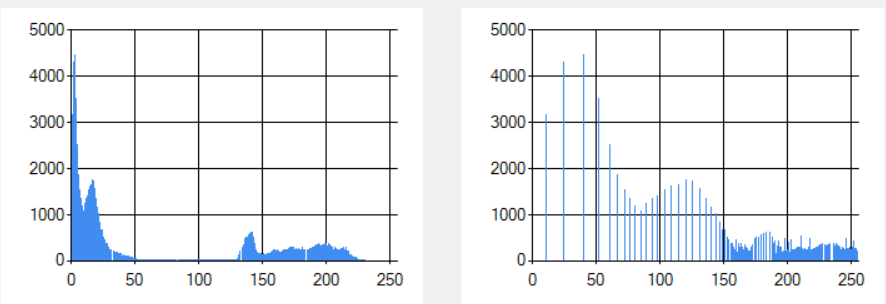
原圖 處理後

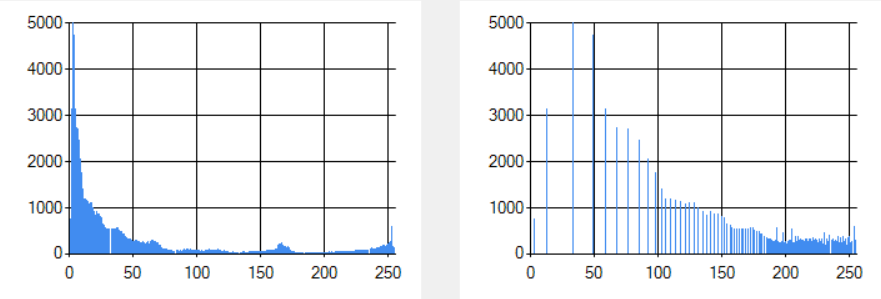
****

原圖 處理後

**Discussion**

****

****

從Histogram中可以看出，Intensity較高的區域被拉較開，透過Histogram Equalization可以讓顏色糊成一塊的區域變較清晰

**A user-defined thresholding**

**Problem**

Implement Binary Thresholding

**Method**

輸入threshold t，將pixel intensity低於t的值設為0，大於等於t的值設為255

**Results**

**原圖 處理後**

****

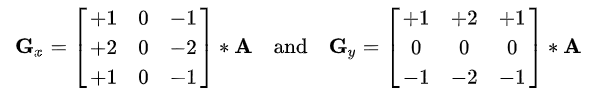
**Sobel edge detection**

**Problem**

Implement Sobel Edge in vertical, horizontal and combined

**Method**

設計Vertical 和 Horizontal 的Filter分別如下



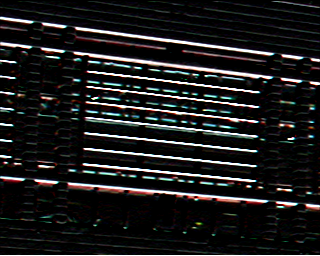
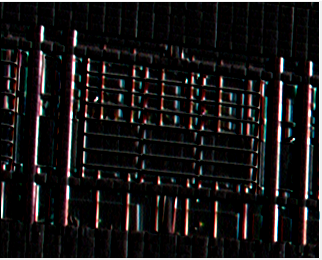
將input圖片Padding 1後，利用上述filter對圖片做convolution

**Results**

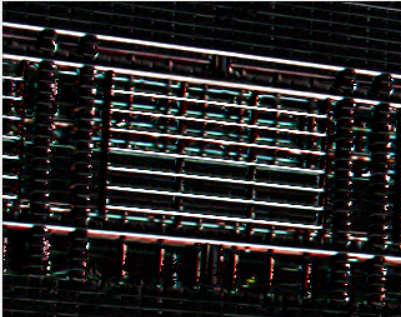
原圖



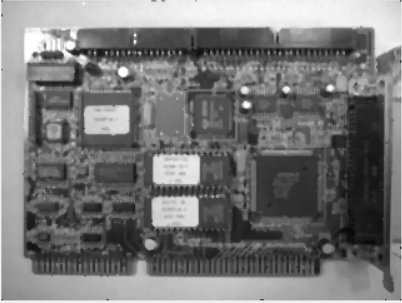
Vertical Horizontal



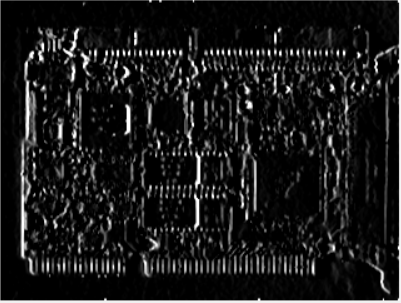
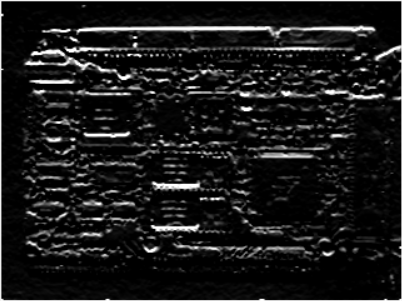
Combined



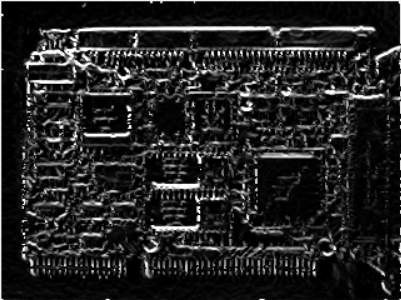
原圖



Vertical Horizontal

Combined



**Threshold the result of (5) to binary image and overlap on the original image**

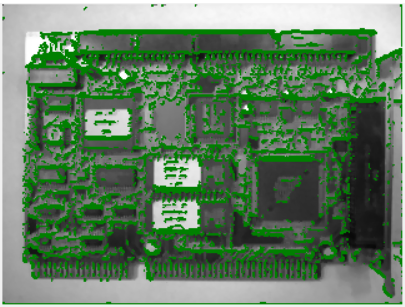
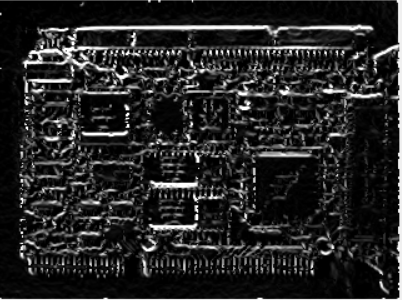
**Problem**

Threshold the sobel filter result

**Method**

將第五題的輸入帶入第四題做的User Define Threshold，將非黑色的部分以綠色畫上原圖

**Results**



**Image registration**

**Problem**

Register Input image to reference Image

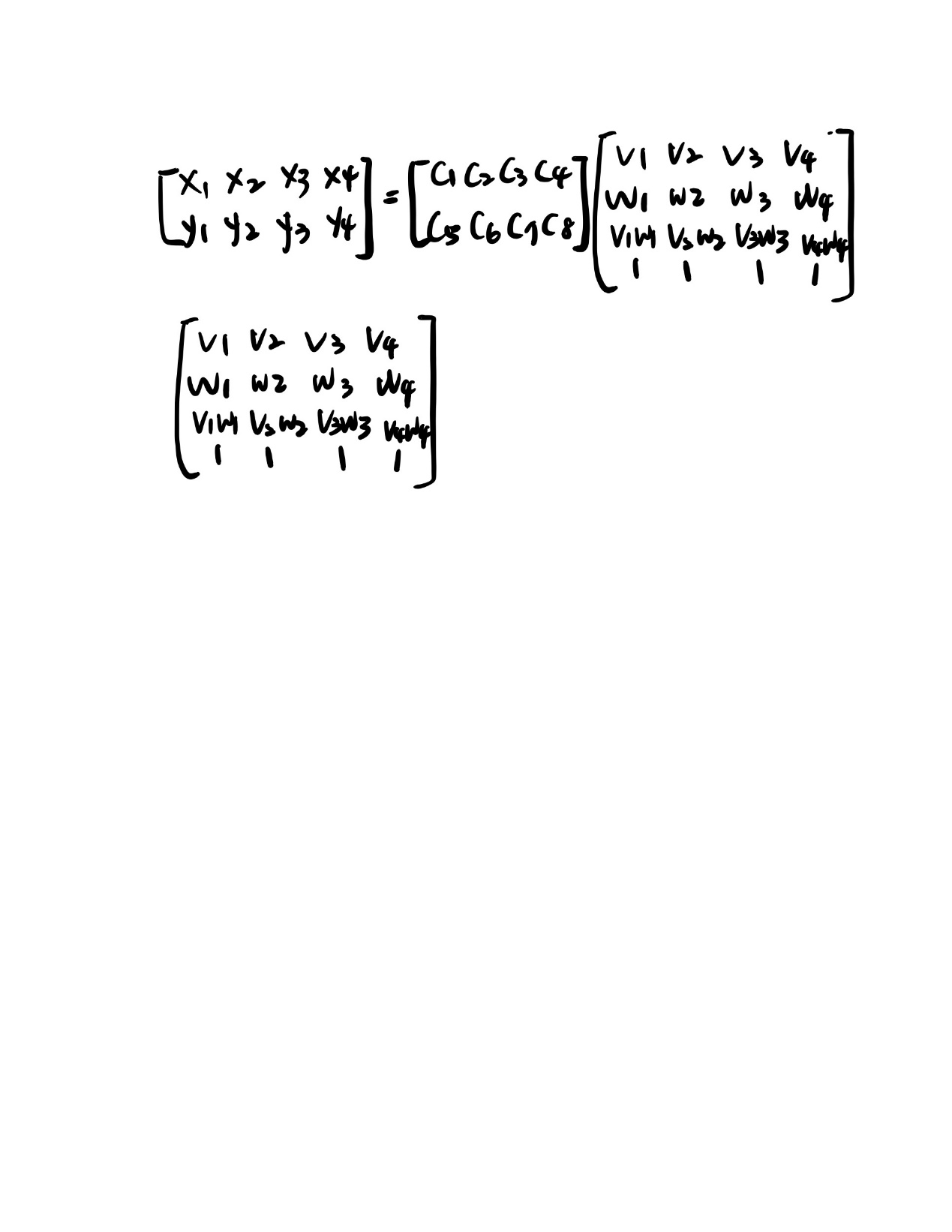
**Method**

參考課本，利用下列方程式

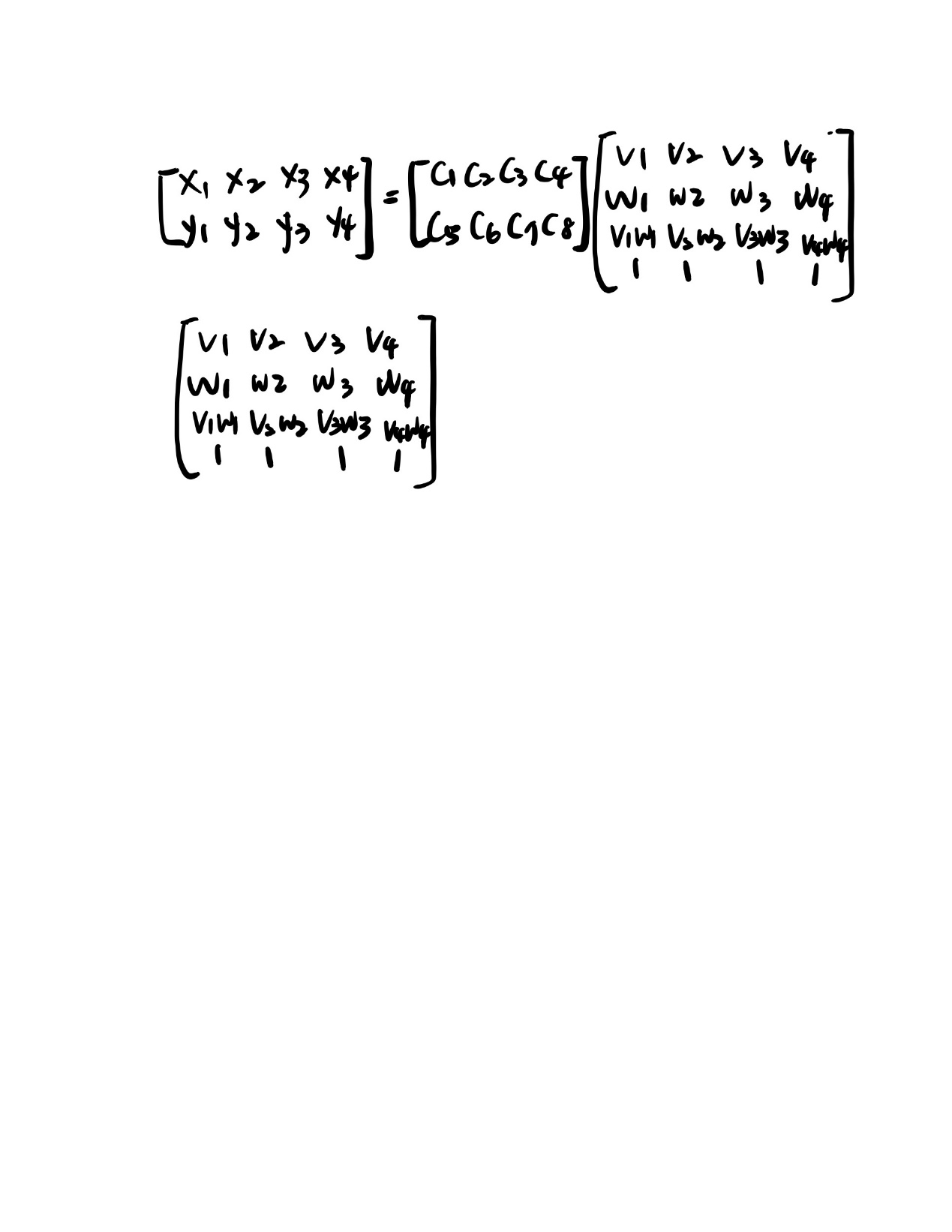


從原圖( Reference Image )中找到的4個點帶入上述式子( v, w)，在從Input Image中找出4個對應的點帶入上述式子的( x, y)，解出c1~c8。

實作上使用矩陣乘法



找到

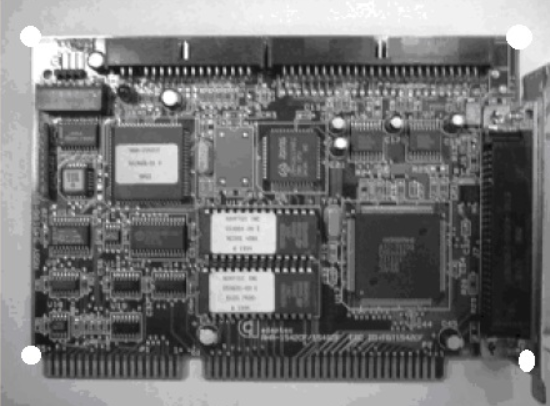


的反矩陣後即可算出c1~c8。

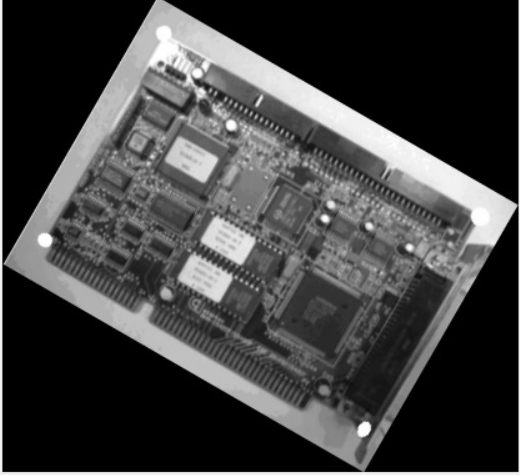
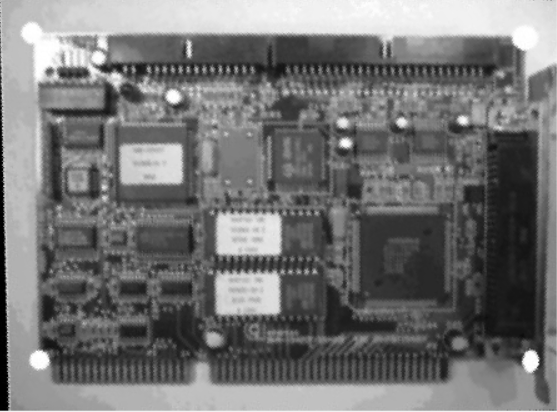
再來從Output Image反推每一個pixel所對應到的是input image上的哪一個點，將output上每一個點 (j, i) 帶入 (v, w)，即可求出對應input image的點。

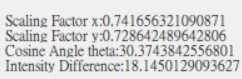
**Results**

Reference Image



Input Image Output Image



**Discussion**

點的4個點會影響到Registration的效果

**Conclusion**

從這次作業可以看出，有些task是需要好幾種演算法配合完成的 (例如 sobel前要先帶median filter )，而這些方法經常需要大學所學的必修科目(線性代數、機率學等等)。