Image process

hw1 p76081433 資工所碩二潘文傑

1. RGB Extraction & Transformation

Methods:

將圖片存成bitmap透過getpixel來取得每個pixel的r,g,b值

Results:

Methods:

將圖轉成grayscale的方法是將r,g,b的值相加之後，取平均代回原圖，便會產生出灰階影像

Results:



Discuss:

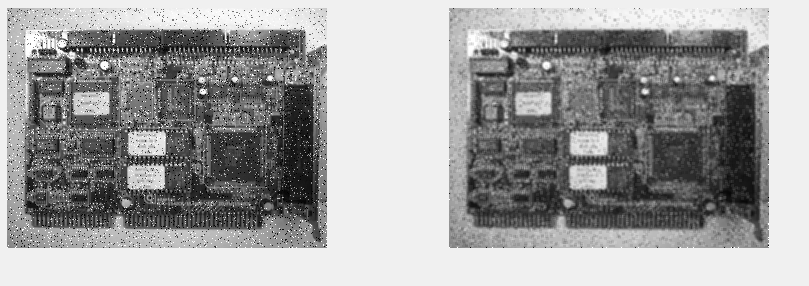
分別看r,g,b channel會看到同一個pixel上面不同channel會有不同的強度，如此組出一張彩色影像

若是三個channel值都相同就會變成一張灰階影像

2.Smooth filter (mean and median)

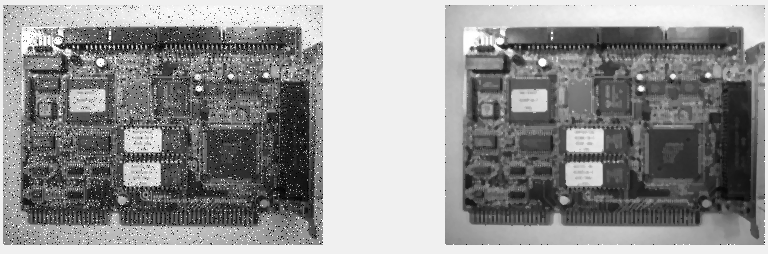
Methods:  
Mean filter是透過一個3x3 filter在圖上滑動 filter中心那個pixel的強度值為9個值相加取平均

Results:



Methods:

Median filter 是透過一個3x3 filter在圖上滑動 filter中心那個pixel的強度值為9個值做排序後取第五個數



Discuss:

在pepper and salt雜訊中，雜訊多半是黑點或白點，透過mean filter將雜訊與其他像素強度做平均，稍為降低了雜訊但圖片依然很模糊

當使用median filter 他會將過強或過低的訊號給消除，所以pepper and salt雜訊就被消除了，圖片也變清楚了

3.histogram equalization

Methods:

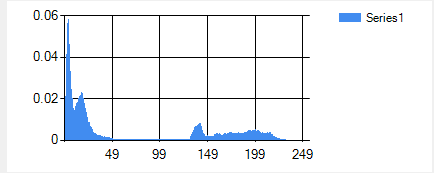
直方圖等化法分為三個步驟

第一步 計算輸入影像的機率分布函數 pmf

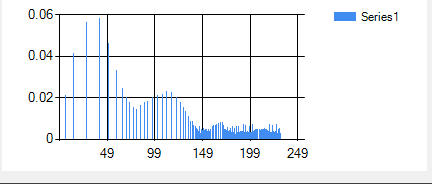
第二步 計算累積分布函數cdf

第三步 計算轉換函數 s= T(r) s=round(cdf(r)\*max-min)

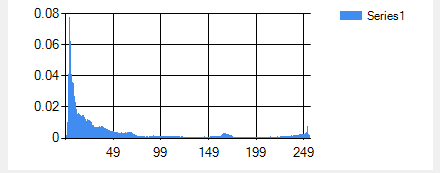
Results:

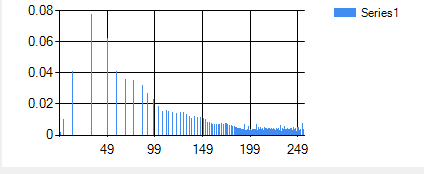












Discuss:

透過此方法可以增加影像的對比度，原先的影像若是集中在某個區域透過此方法可以將向速強度拉開，影像的對比會增強，但是若是原本影像強度就強的地方，有可能產生過曝的情況

4.A user-defined thresholding

Method :

檢查圖片的pixel如果小於threshold的都改成0反之改成255

Results:



Discuss:

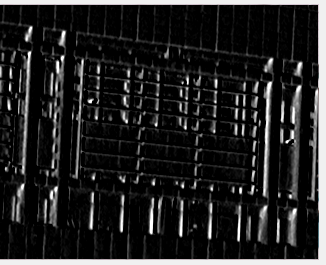
此作法會產生出非黑即白的影像，透過設定不同的threshold值可以調整我們

要保留圖片的那些部分

5. Sobel edge detection (vertical, horizontal, and combined)

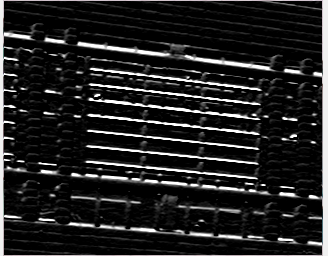
Methods:透過vertical sobel的filter在空間域上做convolution可以找出圖片的垂直線

Results:



Methods:透過horizontal sobel的filter在空間域上做convolution可以找出圖片的垂直線

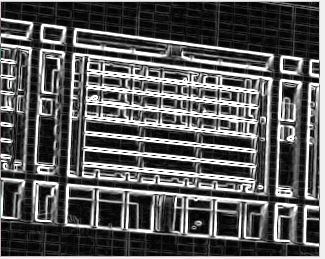
Results:



Methods:透過兩個sobel filter對原圖計算出值之後，做平方相加在開根號

就可以找出圖片的水平和垂直邊緣，來做到邊緣偵測

Results:



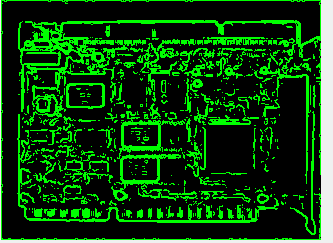
Discuss:

我們可以發現透過不同的filter可以找出圖片在不同方向強度變化的部分，filter之間也可以互相結合，找出圖片中不同的細節

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

Methods:將六做完的圖片進行二值化處理後，留下來的就是物件的邊緣，在將它畫到原圖上

Results:



Discuss:

可以觀察到物件的邊緣都被加強在原本的圖上，隨者threshold的值往上調會留下來的是較粗的邊緣

7. Image registration

Methods:

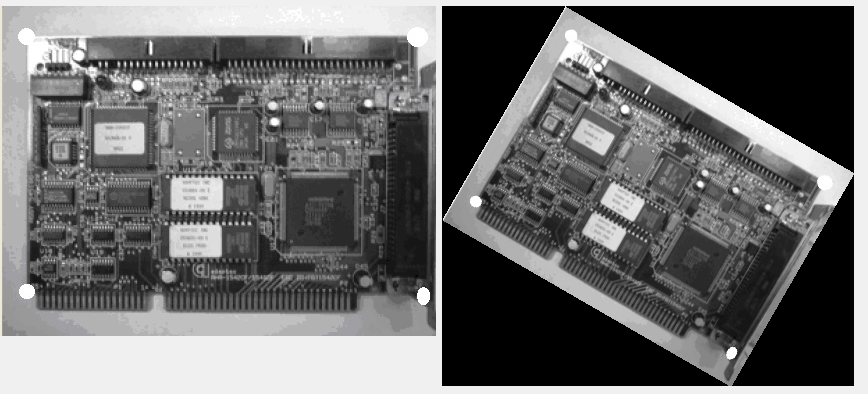
這提要做的是圖片的定位

一開始先測試多組的縮放大小，以及旋轉角度

最後觀測x,y縮放應為0.75

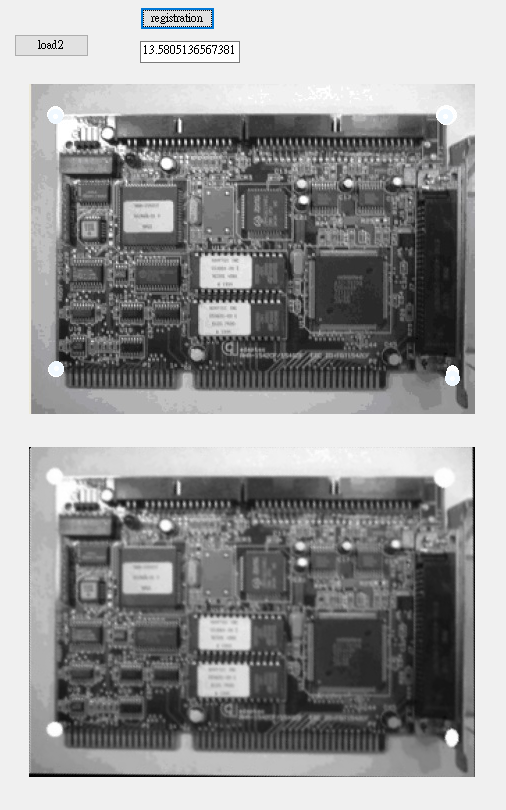
旋轉30度

Results:



Methods:

透過四組點，可以解出bilinear approximation的未知數，接下來在透過內差法計算圖的強度，並計算和原圖的intensity difference



Discuss:

原本是想透過旋轉縮放後的點反推旋轉跟縮放矩陣，但是因為坐標系根本不同所以算出來的矩陣一直都是無解，所以才自己去嘗試了很多縮放跟旋轉來找到當初到底是旋轉以及縮放多少

Image registration則是透過旋轉縮放後的點把他轉回原圖，因為需要經過放大所以做了內差法去逼近，但差異還是有點明顯，圖片看起來

Conclusion:

透過這次的做業讓我將課本上所教的影像處理方法從數學公式轉換到實際的程式，也從自己所寫的程式看到圖片的變化，表面上數學公式可能只是簡單的計算，例如反矩陣或是解聯立，但是在程式上面沒有套件的幫忙確實麻煩許多!!