**ADIP 2016年專題報告**

賴笙昊、洪育佳

學號：105318032實驗室：張文中教授

學號：105318036實驗室：張文中教授

摘要—本系統為一款影像處理的MFC程式，可讓使用者自行讀檔並選擇相應模式做影像處理。本系統利用連接元件邊緣演算法以及道格拉斯-普可算法找出影像物件的最大輪廓，並減少輪廓上的點來加速電腦的運算速度。

# 簡介

本篇文件是介紹一款影像處理的程式，此程式透過Visual Studio2012平台以及OPENCV程式庫為基礎所撰寫出來的一款C語言程式，使用者可給予一張影像，並選擇三種模式：第一種模式為無光處理，第二種模式為有光書本，第三種模式為有光紙張。使用者可自行判斷所需要的模式，系統會將影像中的文件或書本封面標記出來並顯示出運算時間。

# 基本指導方針

**2.1色彩空間轉換**

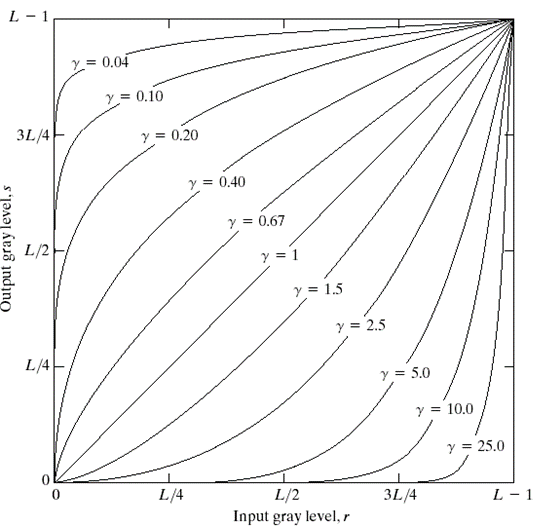
若影像使用RGB色彩空間進行處理，處理過程會比灰階影像複雜，因此本系統將RGB色彩空間轉換成灰階影像來處理。其色彩空間轉換如(1)式。Y輸出圖片，R為紅色通道，G為綠色通道，B為藍色通道。

(1)

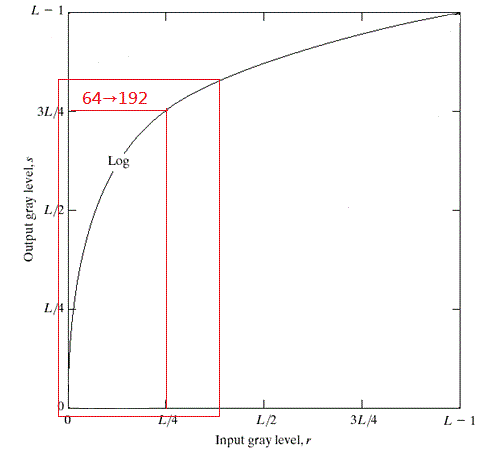
由於影像會有不均勻的光源，在RGB色彩空間上會有過度曝光或者是陰影存在，因此轉換到灰階影像時須作基本的強度轉換，最常使用的三種基本函數為：負片、Log transformations以及Power-Law，為了將影像較暗的部分，提升成較亮的像素值，故本系統採用Power-Law作為強度轉換的函數如(2)式，透過對數轉換，將過曝的照片暗一點，讓曝光度不足的照片亮一點。如圖一及圖二所示。圖二的紅色框框裡都是屬於比較暗但偏黑的部分，經過對數後，數值將會變大。s為輸出圖片，為調整值，r為輸入圖片，c為常數。

(2)

為了找出物體輪廓，因此需要先將灰階圖片作二值化的運算，分成感興趣的部分，以及不感興趣的部分，通常將某個閥值作為分佈的標準。由於每張影像會有不同的閥值，故使用Otsu自動門閥值決定法則，要先計算影像的直方圖，把直方圖強度大於閥值的像素分成一組，把小於閥值的像素分成另一組，再分別計算這兩組的組內便藝術，並把兩個組內變異數相加，最後將0~255依序當作閥值來計算組內變異數和，總和值最小的就是結果閥值



**圖一： 強度轉換函數圖**



**圖二： 強度轉換示意圖**

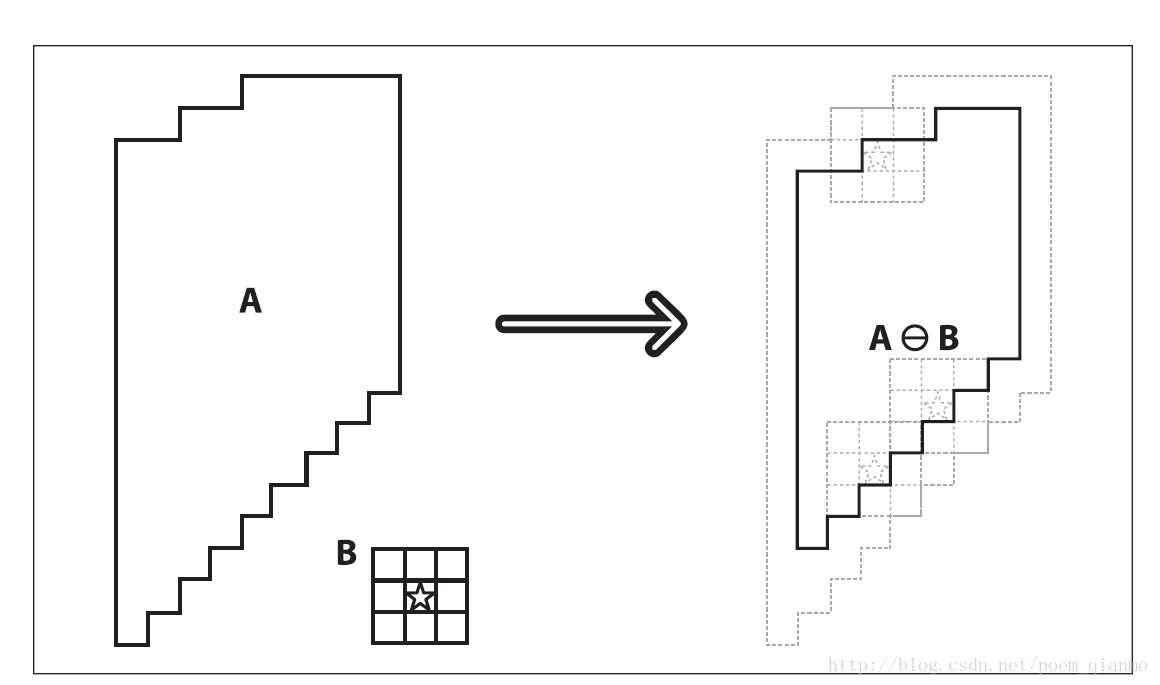
**2.2侵蝕與膨脹**

經由色彩空間轉換後獲得的二值化影像有許多的雜訊需要去除，本系統採用大小為10\*10的遮罩對影像做處理。erode侵蝕顧名思義為消融物體的邊界，如圖三所式，如式(3)，B對A進行侵蝕。

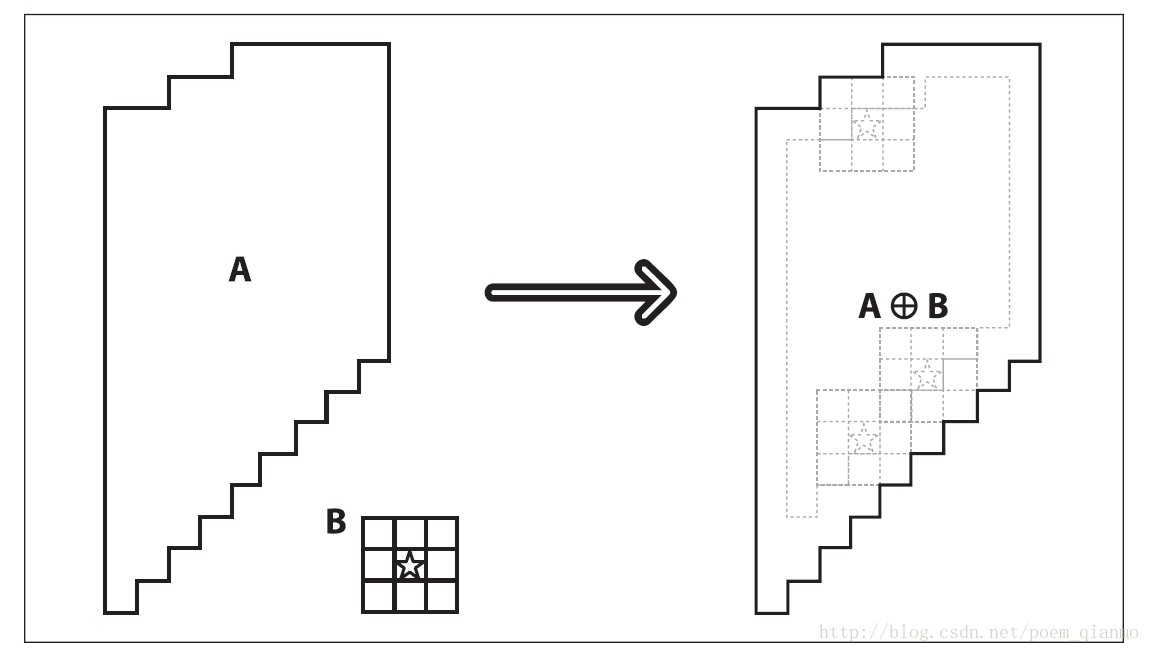
dilate膨脹為擴大物體的邊界，如圖四所式，如式(4)，H對I進行膨脹。

(3)

(4)



**圖三： 侵蝕示意圖**



**圖四： 膨脹示意圖**

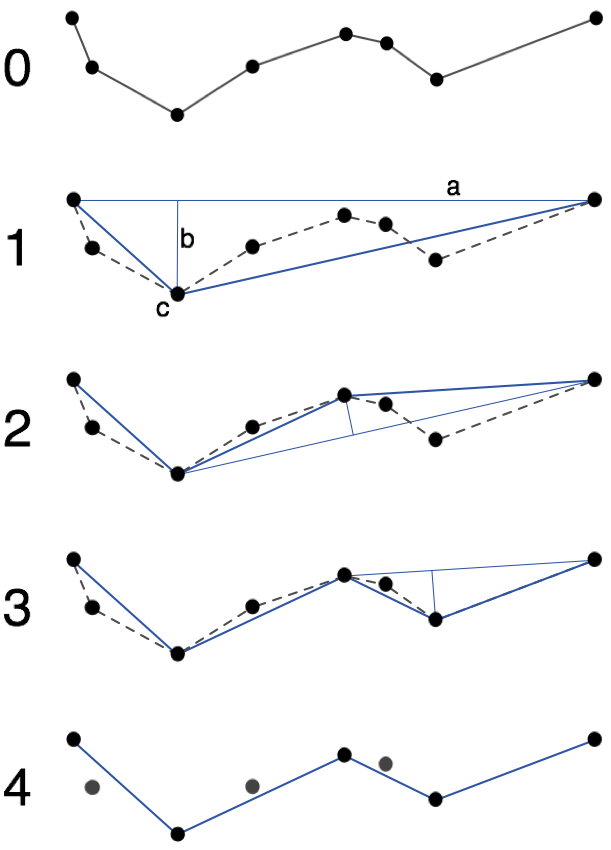
**2.3找出輪廓**

連接元件邊緣演算法，將相連的點串聯起來形成一個輪廓，接著在對每一個輪廓進行後續判斷。此演算法可尋找輪廓的最外圍，而不會在向內繼續搜尋。

**2.4輪廓逼近**

道格拉斯-普可算法，該算法遞歸地劃分線，最初，先由使用者設定，然後在第一點和最後一點之間點出所有線上的點，並找出第一點和最後一點作為終點線段的最遠的點，該點顯然距離端點之間的近似線段的曲線最遠。如果距離線段最遠的點大於來自近似的，則必須保持該點，如果點比線段更接近於，則可以丟棄當前未被標記為要保留的任何點，而不會使簡化曲線比更差。

當遞歸完成時，可以生成由那些被保留下來的點所輸出的新曲線。流程圖如圖五所示。



**圖五： 道格拉斯-普可算法平滑一條已線性分段的曲線**

**2.5找出四個對應角點**

取得輪廓後，將所有點依依計算並比較。因為座標的原點是在最左上角，所以推得輪廓中最左上角的點座標之是最小值，反之輪廓中最右下角的點座標之是最大值，接著根據的最小最大值依序找出輪廓中的最右上角以及最左下角。找到這四個點之後，就能進行透視變換，將此平面轉移到另一個平面。

**2.6透視變換**

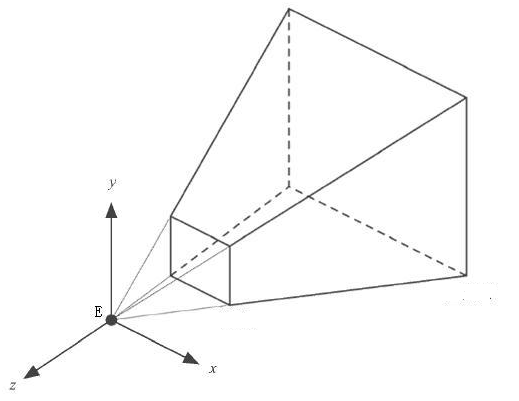
找到輪廓後，可以找到邊緣的最左上點、最左下點、最右上點以及最右下點，再將這四個點透過透視變換轉換到新的圖像裡，就可以產生出結果圖，即書本或紙張的封面。透視變換是將圖片投影到一個新的透視平面，也稱做投影映射，通用的變換公式如(5)：

(5)

u,v是原始圖片左邊，對應得到變換後的圖片座標x,y，其中。

變換矩陣可以拆成四部分，表示線性變換，比如scaling，shearing和rotation。

用於平移， 產生透視變換。所以可以理解成仿射是透視面換的特殊形式。簡易概念圖如圖六所示。

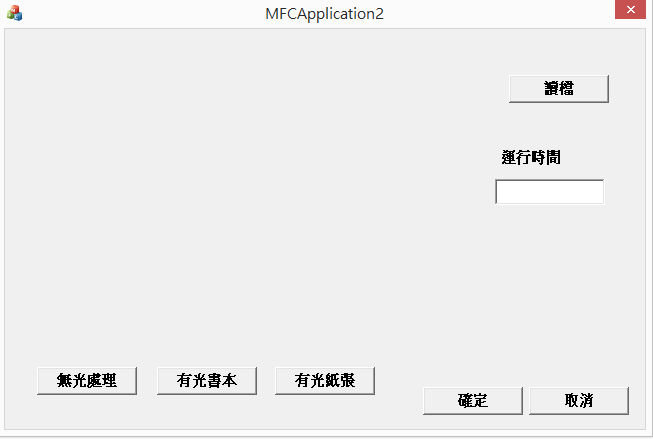


**圖六： 透視變換概念圖**

# 實驗結果

3.1 實驗架構

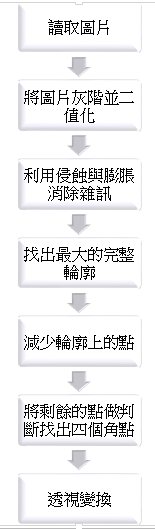
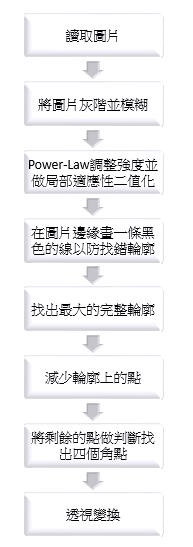
此系統是由Microsoft Visual studio2012所撰寫的MFC程式，此系統可由使用者自行讀取JPG圖檔，並且選擇模式使用，如圖七所示。



**圖七： 系統視窗**

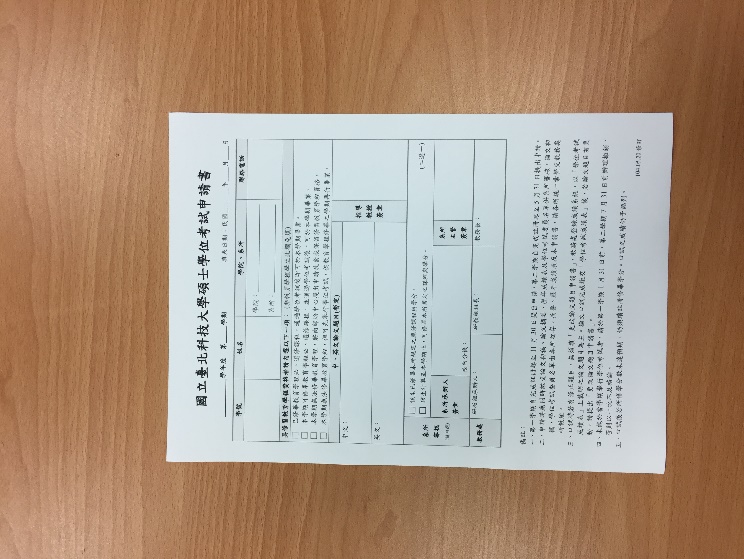
3.2 實驗流程

由於多種模式供選擇，因此各模式的流程分別以圖八、圖九做解說，較為直觀。

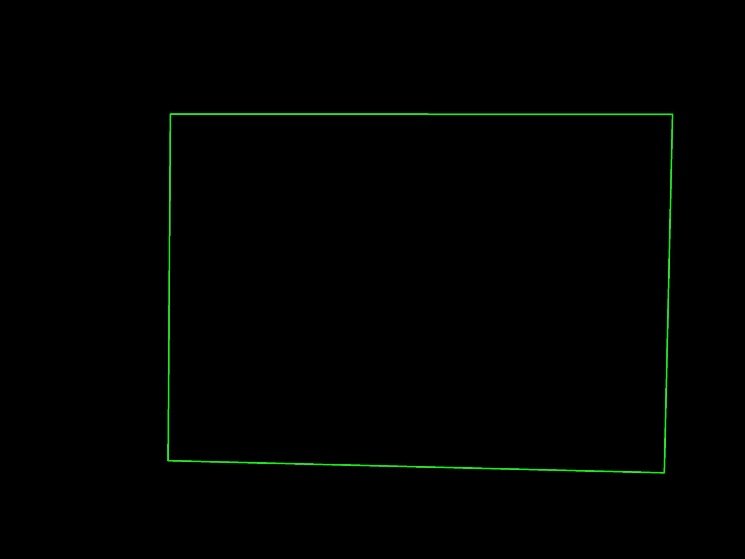
 

**圖八： 無光處理流程圖 圖九：有光書本和紙張流程圖**

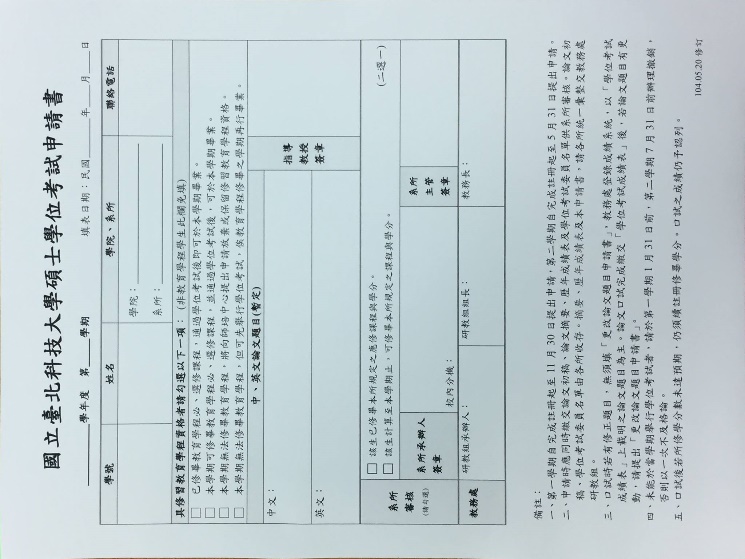
3.3 實驗結果



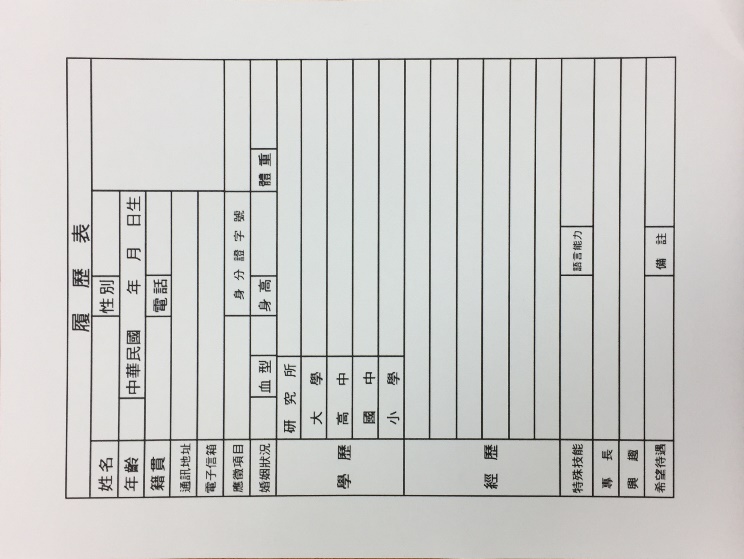
**圖十： IMG\_2343.JPG**

****

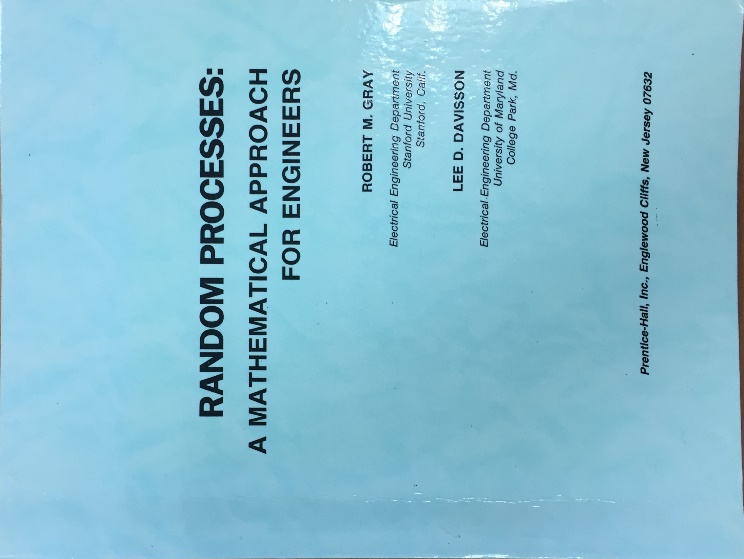
**圖十一： IMG\_2343.JPG的輪廓**



**圖十二： IMG\_2343.JPG取得結果**



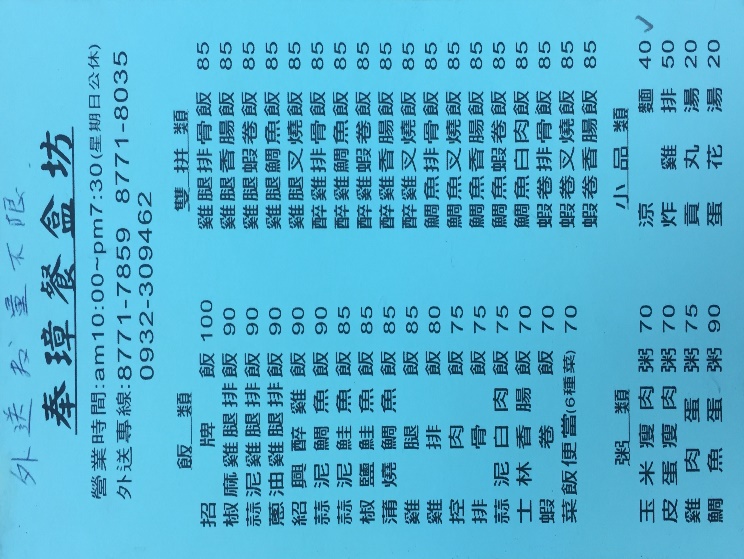
**圖十三： IMG\_2334.JPG取得結果**



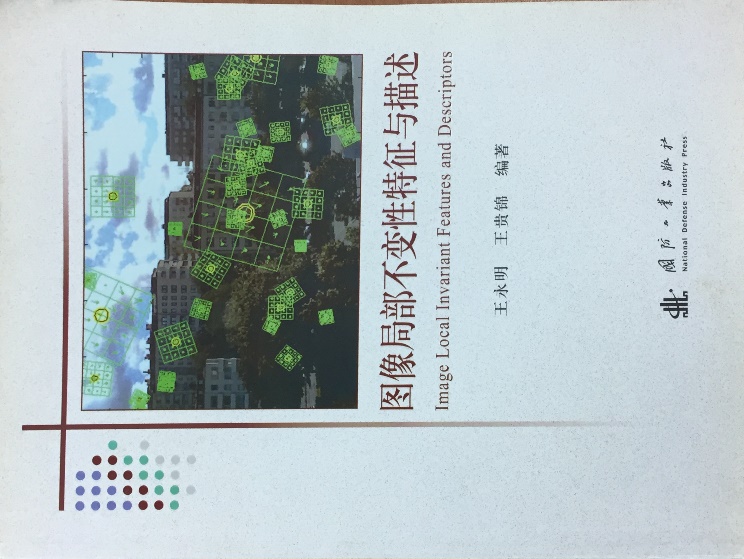
**圖十四： IMG\_2335.JPG取得結果**



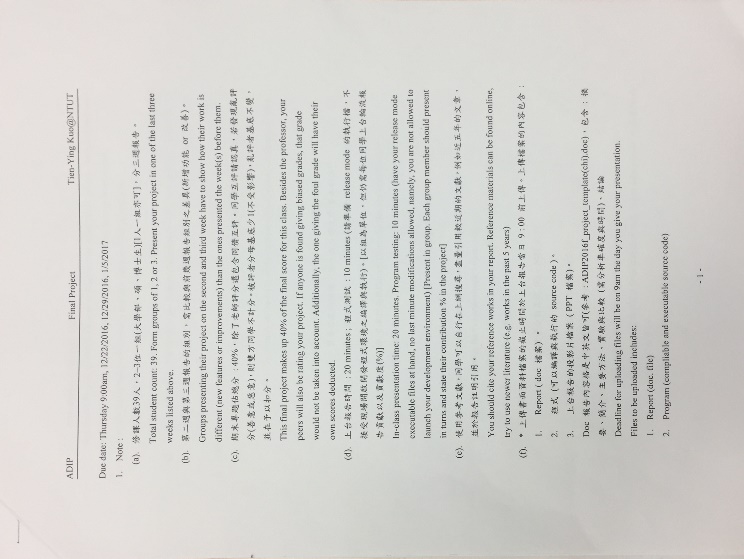
**圖十五： IMG\_2338.JPG取得結果**



**圖十六： IMG\_2339.JPG取得結果**



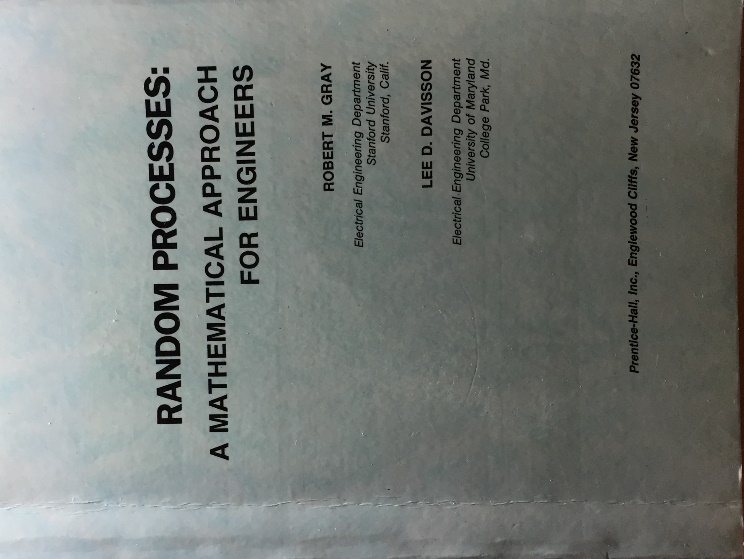
**圖十七： IMG\_2342.JPG取得結果**



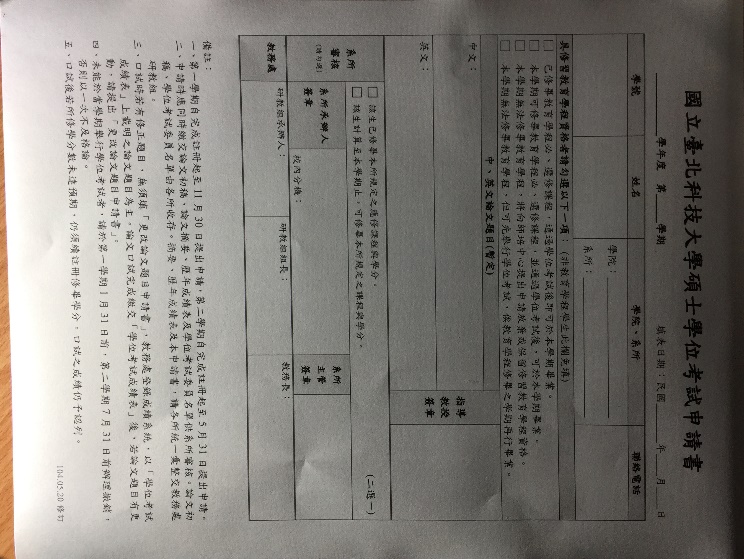
**圖十八： IMG\_2344.JPG取得結果**



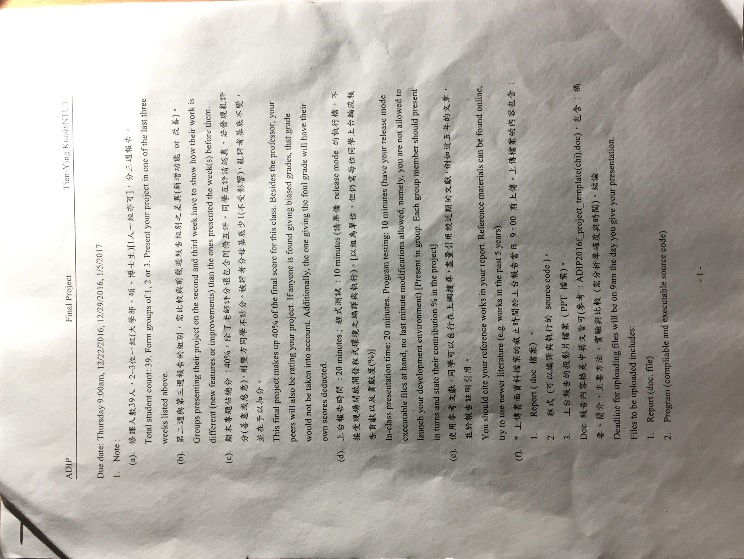
**圖十九： IMG\_2355.JPG取得結果**



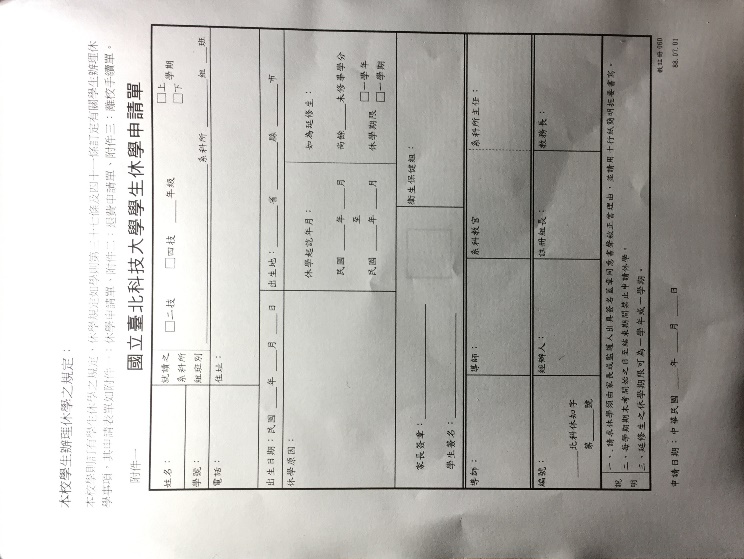
**圖二十： IMG\_2356.JPG取得結果**

****

**圖二十一： IMG\_2364.JPG取得結果**



**圖二十二： IMG\_2365.JPG取得結果**



**圖二十三： IMG\_2366.JPG取得結果**

**表I**

實驗結果報表

|  |  |  |
| --- | --- | --- |
| **檔名** | **時間(sec)** | **是否成功** |
| **IMG\_2334** | **0.313** | **是** |
| **IMG\_2335** | **0.359** | **是** |
| **IMG\_2337** | **0.344** | **否** |
| **IMG\_2338** | **0.344** | **是** |
| **IMG\_2339** | **0.328** | **是** |
| **IMG\_2342** | **0.359** | **是** |
| **IMG\_2343** | **0.32** | **是** |
| **IMG\_2344** | **0.359** | **是** |
| **IMG\_2355** | **0.969** | **是** |
| **IMG\_2356** | **0.991** | **是** |
| **IMG\_2358** | **0.922** | **否** |
| **IMG\_2359** | **0.901** | **否** |
| **IMG\_2364** | **0.985** | **是** |
| **IMG\_2365** | **0.922** | **是** |
| **IMG\_2366** | **0.938** | **是** |

# 結論

此系統為一款影像處理的MFC程式，可以由使用者自行選擇圖片以及模式對該影像做處理，其核心重點為連接元件邊緣演算法以及道格拉斯-普可算法，可以找出最大的輪廓並減少輪廓上的點加速程式的運算。目前程式上還是會有部分圖片無法處理的問題，因此未來可能參考更多的論文來讓此程式發揚光大。

五、參考文獻

1. [Mustafa Ali Abuzaraida](http://ieeexplore.ieee.org/search/searchresult.jsp?searchWithin=%22Authors%22:.QT.Mustafa%20Ali%20Abuzaraida.QT.&newsearch=true), [Salem Meftah Jebriel](http://ieeexplore.ieee.org/search/searchresult.jsp?searchWithin=%22Authors%22:.QT.Salem%20Meftah%20Jebriel.QT.&newsearch=true) “The detection of the suitable reduction value of Douglas-Peucker algorithm in online handwritten recognition systems” [*IEEE International Conference on Service Operations And Logistics, And Informatics (SOLI)*](http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=7360270),2015,pp.82-87
2. Rafael C. Gonzalez, Richard E. Woods, “Image Processing”,3/e, 2008, pp. 108–113.
3. OpenCV入門教程<http://blog.csdn.net/poem_qianmo/article/details/23710721>
4. Intel Corporation,“Open Source Computer Vision Library –Reference Manual”,2001,pp.10-6 - 10-7
5. 阿洲的程式教學http://monkeycoding.com/?p=690#BGR\_Gray
6. 王冠智，「雲端筆記之混和式文字切割與辨識」，國立政治大學資訊科學系，2012，pp.30
7. 逍遙文工作室https://cg2010studio.com/2013/03/22/opencv-%E9%80%8F%E8%A6%96%E8%AE%8A%E6%8F%9B-perspective-transform/
8. 學習堅持，堅持學習-影像處理筆記<https://dotblogs.com.tw/dragon229/2012/12/28/86092>
9. S. Suzuki , K. Abe “Topological structural analysis of digitized binary images by border following” Computer Vision, Graphics, and Image Processing, Volume 30, Issue 1, 1985, pp. 32–46.