

國立交通大學

資訊科學與工程研究所

碩士論文

一種新式的印刷電路板定位方法

A new positioning method for PCB

研究生：鄭遠威

指導教授：荊宇泰 教授

中華民國 102 年 8 月

一種新式的印刷電路板定位方法
A new positioning method for PCB

研究生：鄭遠威

Student : Yuan-Wei Cheng

指導教授：荊宇泰

Advisor : Yu-Tai Ching

國立交通大學
資訊科學與工程研究所
碩士論文



A Thesis
Submitted to Institute of Computer Science and Engineering
College of Computer Science
National Chiao Tung University
in partial Fulfillment of the Requirements
for the Degree of
Master
in
Computer Science

June 2012

Hsinchu, Taiwan, Republic of China

中華民國 102 年 8 月

一種新式的印刷電路板定位方法

學生：鄭遠威

指導教授：荊宇泰

國立交通大學資訊科學與工程研究所

摘 要

目前現在印刷電路板的定位誤差校正，最常用的方式就是用相機拍攝印刷電路板上的圓或是某些記號，若是尋找印刷電路板上的圓，就要利用使用 Hough Transform 得到相機拍攝的影像中的圓，藉此判斷相機拍攝的印刷電路板是否有擺放正確。

但是讓相機為了拍到印刷電路板上完整的記號，往往需要經過多次的調整後，才有辦法拍攝到完整的記號。

在本論文，希望放多個相機拍攝欲調整印刷電路板中的不同區塊，並且利用 Geometric Hashing，去尋找欲參考圖片中的區塊，藉此做影像對齊，如此可以不用去拍攝印刷電路板上的特定記號，就能夠達到判斷印刷電路板的擺放是否正確。

A new positioning method for PCB

student : Yuan-Wei Cheng

Advisors : Yu-Tai Ching

Institute of Computer Science and Engineering
National Chiao Tung University

ABSTRACT

Now, people correct the location error of the printed circuit board, the most common way is to use camera to capture the circle or some mark on the printed circuit board.

If we find the circle on the printed circuit board, it is necessary to find the circle using the Hough Transform to specify the circle to determine whether the printed circuit board is correctly placed.

But let the camera capture full mark on the printed circuit board, we often need to go through repeated adjustment, and then we can capture the full mark.

I hope to put multiple camera on the printed circuit board in different blocks, and apply Geometric Hashing to find the desired reference picture blocks to make the image alignment, so you can't find mark on the print circuit board.

誌 謝

感謝荆宇泰教授這兩年來的指導，讓我了解如何到如何能夠去做好一個研究，並且學習到在這個領域上專業的知識，最後協助我完成碩士論文。感謝實驗室學長們不厭其煩的從旁協助，並且耐心的協助我在求學過程中所遇到的問題，並且從旁指出我的問題所在。

也感謝 MIP 實驗室的同學們，在這段時間內能夠一起奮鬥，一起歡笑，在這兩年中一起成長，互相砥礪。陪伴我渡過在交通大學這段求學的日子。最後感謝我的父母，在背後支持我，給予我協助，鼓勵，讓我可以無後顧之憂的完成我的學業。



目 錄

| | | |
|----------|---------------------------------------|-----|
| 摘要 | | I |
| Abstract | | II |
| 誌謝 | | III |
| 目錄 | | IV |
| 圖目錄 | | VI |
| 表目錄 | | VII |
| 一、 | 前言..... | 1 |
| 二、 | 理論基礎..... | 3 |
| 2.1 | 切割參考圖片..... | 3 |
| 2.2 | 基於 α Shape 的特徵點提取..... | 5 |
| 2.2.1 | α Shape..... | 5 |
| 2.2.2 | DCEL(doubly connected edge list)..... | 7 |
| 2.3 | 取得欲對齊圖片使用 geometric hashing..... | 11 |
| 2.4 | 特徵點匹配..... | 14 |
| 2.5 | 求得影像之間的變換函數..... | 15 |
| 2.5.1 | 2D motions..... | 15 |
| 2.5.2 | 最小平方法..... | 16 |
| 2.6 | 除去錯誤的 match..... | 17 |
| 2.6.1 | RANSAC..... | 17 |
| 2.7 | 平移修正..... | 18 |
| 2.8 | 決定參數..... | 19 |
| 三、 | 實驗..... | 20 |
| 3.1 | 實驗結果..... | 20 |
| 四、 | 結論與未來展望..... | 30 |



圖 目 錄

| | | |
|--------|--------------------------|----|
| 圖 2.1 | 系統流程 | 1 |
| 圖 2.2 | 系統流程示意圖 | 2 |
| 圖 2.3 | 切割參考圖片 | 3 |
| 圖 2.4 | 切割後的參考圖片 | 4 |
| 圖 2.5 | 隨著 α 值的改變，造成形狀的改變 | 6 |
| 圖 2.6 | voronoi diagram 1 | 7 |
| 圖 2.7 | voronoi diagram 2 | 8 |
| 圖 2.8 | 由 α Shape 得到特徵點 | 9 |
| 圖 2.9 | 去除碰到邊界的元件結果 | 10 |
| 圖 2.10 | geometric hashing 流程 | 11 |
| 圖 2.11 | 轉換示意圖 | 12 |
| 圖 2.12 | 參考圖片特徵點 | 14 |
| 圖 2.13 | 欲對齊圖片特徵點 | 14 |
| 圖 2.14 | 欲對齊圖片與欲對齊圖片特徵點 | 14 |
| 圖 2.15 | 平面中的一群點 | 17 |
| 圖 2.16 | 印刷電路板旋轉後的影像 | 18 |
| 圖 2.17 | 系統取得參數的畫面 | 19 |
| 圖 3.1 | 印刷電路板旋轉 1 度的影像 | 21 |
| 圖 3.2 | 印刷電路板(a) | 22 |
| 圖 3.3 | 印刷電路板(b) | 23 |
| 圖 3.4 | 印刷電路板(c) | 24 |
| 圖 3.5 | 印刷電路板(d) | 25 |

表 目 錄

| | | |
|-------|-------------|----|
| 表 3.1 | 對齊結果 1..... | 26 |
| 表 3.2 | 對齊誤差 1..... | 27 |
| 表 3.3 | 對齊結果 2..... | 28 |
| 表 3.4 | 對齊誤差 2..... | 29 |



第一章 前言

現在一般的調整印刷電路板定位誤差的方法，是透過相機拍攝印刷電路板上面的特定記號，但必須以手動的方式去調整印刷電路板，以確保能夠拍攝到完整的記號，才能夠進行定位誤差的調整，所以希望能夠不用去拍攝特定的記號，但多用幾個相機，就可以完成校正的工作。

先將電路板對齊左上角再拍攝印刷電路板的某幾個區塊，並且找出相機拍攝的照片中的特徵點，再與參考圖片做影像對齊，來做定位誤差的調整。

透過多個相機，分別拍攝欲調整印刷電路板不同的區塊，再利用 geometric hashing 去找尋參考圖片中對應的區塊，並且做影像對齊的動作，以得到調整的參數。

使用多個相機是因為不保證由一個相機拍攝的影像找出來的調整參數一定正確，所以利用多個相機，以增加正確性，由每個相機分別拍攝的影像做影像對齊後找出來的調整參數，再從中選擇最有可能的參數。

流程

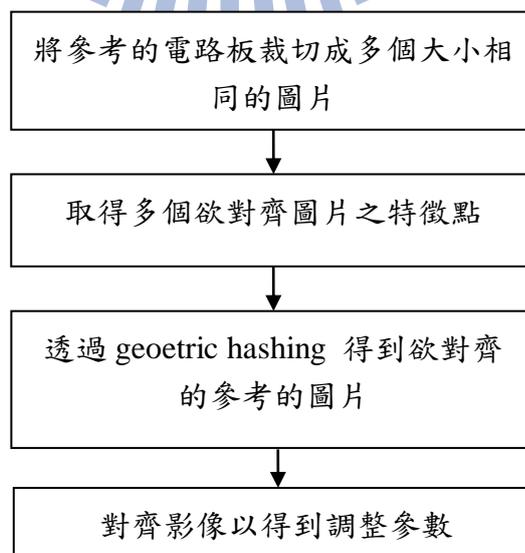
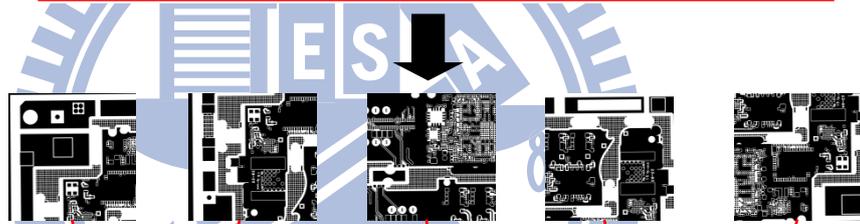
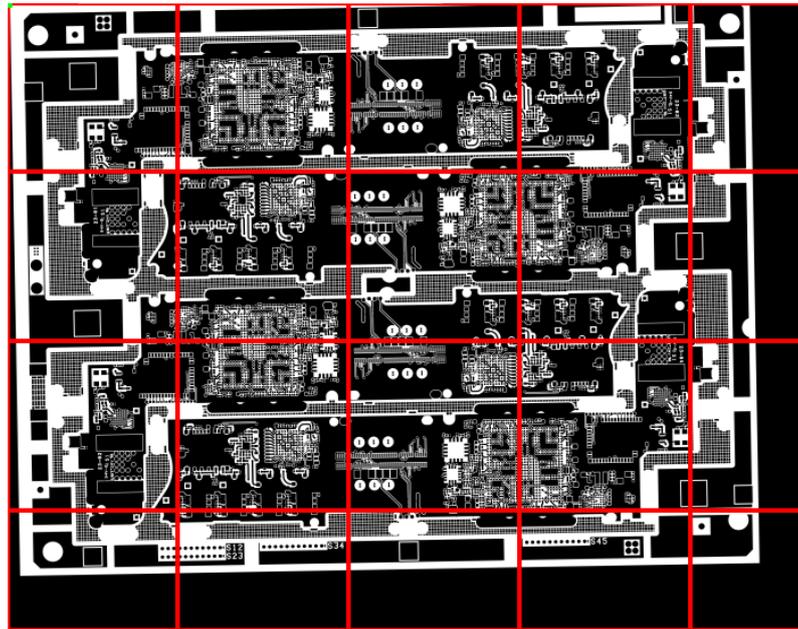


圖 2.1 系統流程

示意圖：

 拍攝欲對齊影像



 : Geometric hashing

圖 2.2 系統流程示意圖

第二章 理論基礎

2.1 切割參考圖片

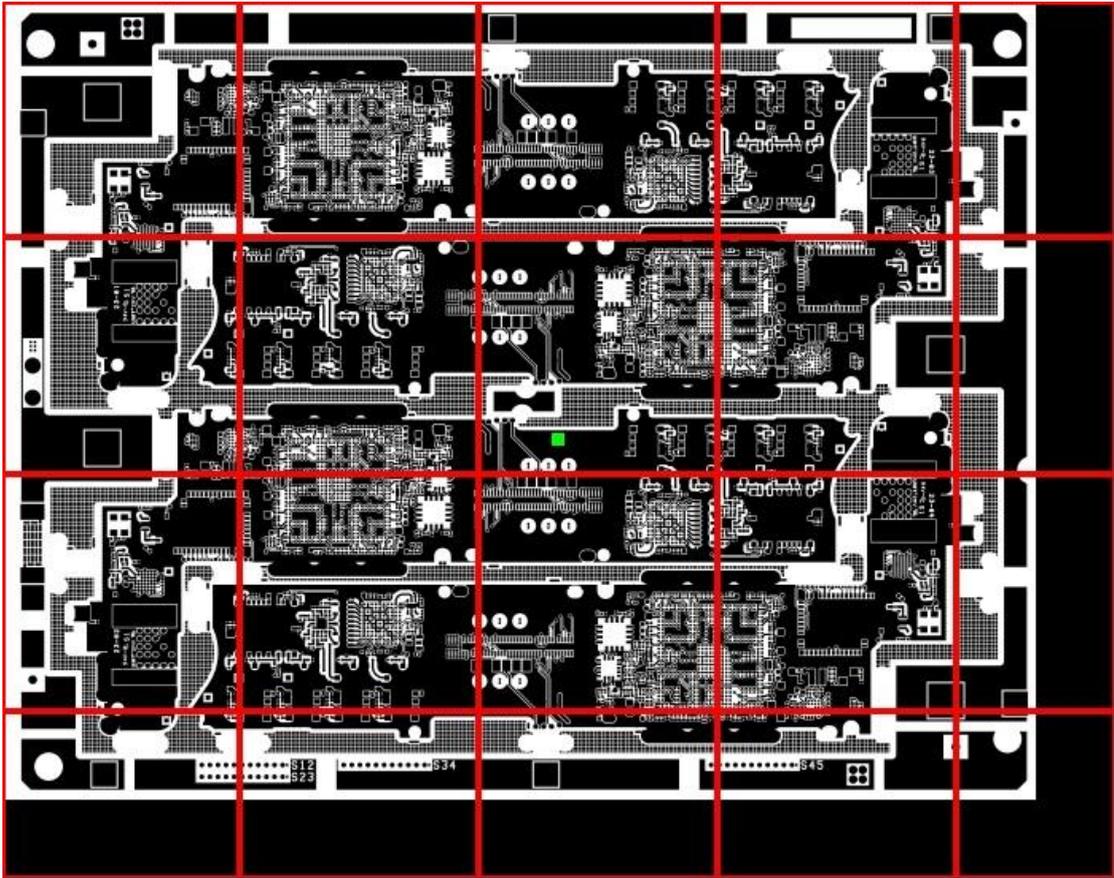


圖 2.3 切割參考圖片

假設此為擺正的情況下的圖片，將此參考圖片切割成許多塊相同大小的圖片，並且記錄每一個圖片在大圖片中的座標，以供最後調整平移參數。邊緣的部分圖片資訊不夠，所以不採用。

假設切割大小為 750*750

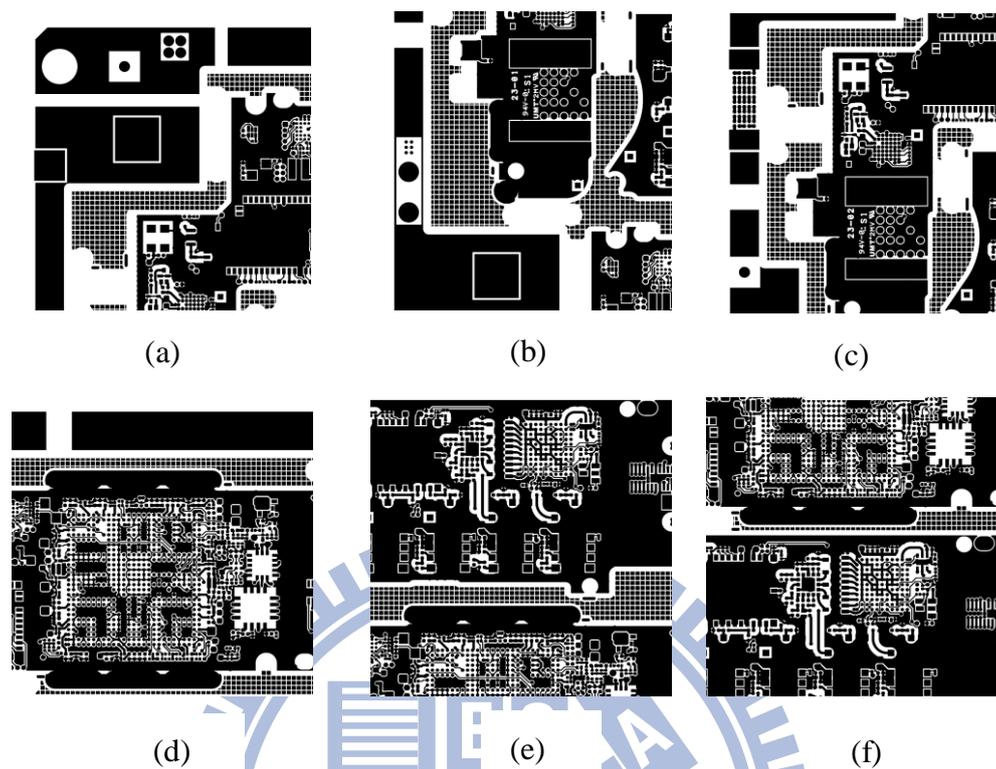


圖 2.4 切割後的參考圖片

(a) 座標為 (0 , 0) (b) 座標為 (0 , 750) (c) 座標為 (0 , 1500)
(d) 座標為 (750 , 0) (e) 座標為 (750 , 750) (f) 座標為 (750 , 1500)
其他圖片依此類推。

2.2 基於 α Shape 的特徵點提取

2.2.1 α Shape

利用電路板上面的元件擺放的相對位置不會因為經過旋轉、平移、縮放而改變其相對位置，所以利用這個想法將元件的中心點，當作特徵點。

找尋圖片中的元件，是使用 Edelsbrunner, Kirkpatrick & Seidel(1983)所提出的 α Shape。

以下簡述求得 α Shape 的演算法：

1. 先從影像中計算出 voronoi diagram 接著再導出 delaunay triangulation。
2. 調整 α 值，以得到最後的 alpha shape。

其中 α Shape 的形狀是根據 α 值而來，而 α 值是依照經驗來調整的，並沒有公式可求得最佳的 α 。

隨著 α 值的變化，可歸納出以下 3 種情況：

1. 當 α 無限大時， α Shape 是一個 convex hull。
2. 當值遞減， α 會產生凹洞。
3. 當 $\alpha=0$ 則 α Shape 為點集合。

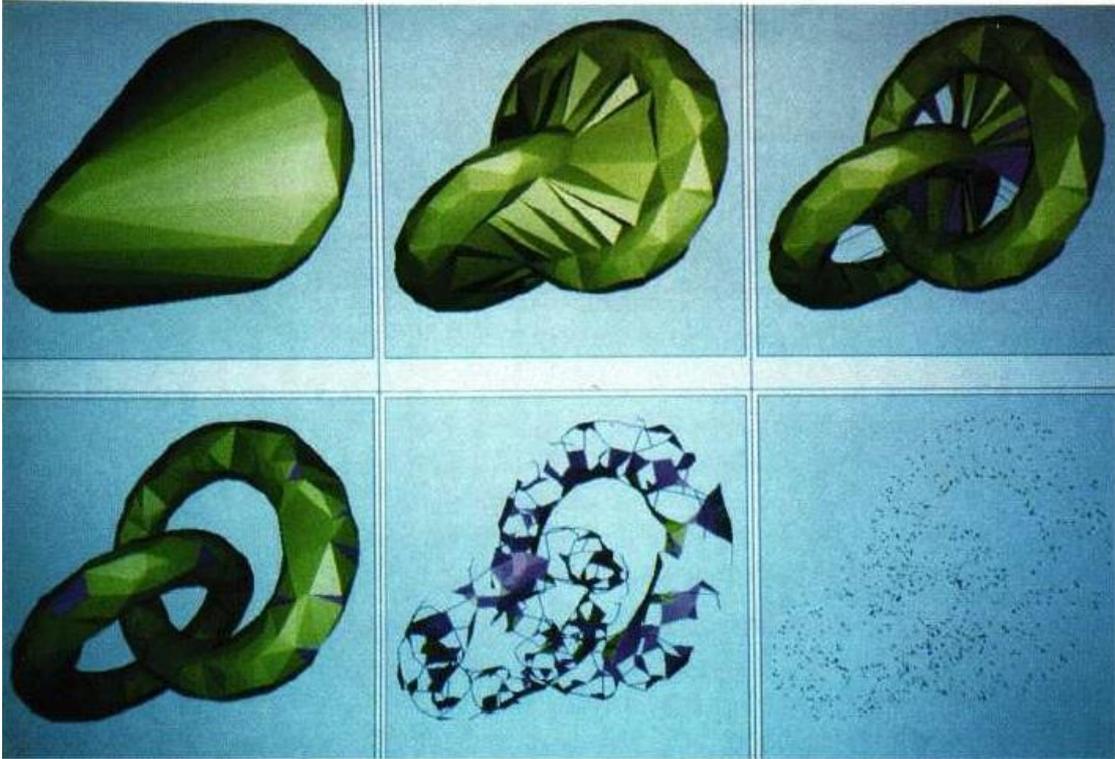
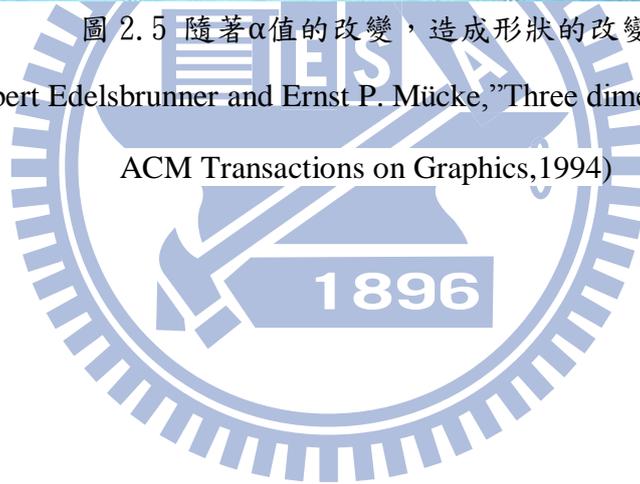


圖 2.5 隨著 α 值的改變，造成形狀的改變

(資料來源: Herbert Edelsbrunner and Ernst P. Mücke, "Three dimensional Alpha Shapes",
ACM Transactions on Graphics, 1994)



2.2.2 DCEL(doubly connected edge list)

利用 DCEL 這種資料結構來儲存找出的 α Shape

DCEL 是一種常常被用來表示 voronoi diagram 的一種資料結構。

1. 是一種 edge-based 的資料結構，將點、線、面互相連結在一起。
2. 更容易 traverse voronoi diagram 中的 face 以及 visit 在某一個點 V 附近的所有的邊(此 half-edge 的 origin 為 V)。

每一個 edge 都由兩個 half -edge 表示，內圈的 half -edge 為逆時針方向，外圈的 half -edge 為順時針方向。

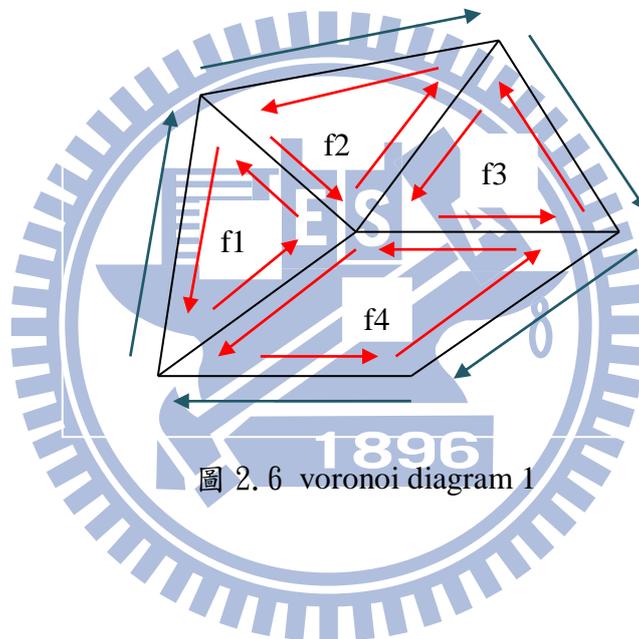


圖 2.6 voronoi diagram 1

DCEL 內部存放的資料：

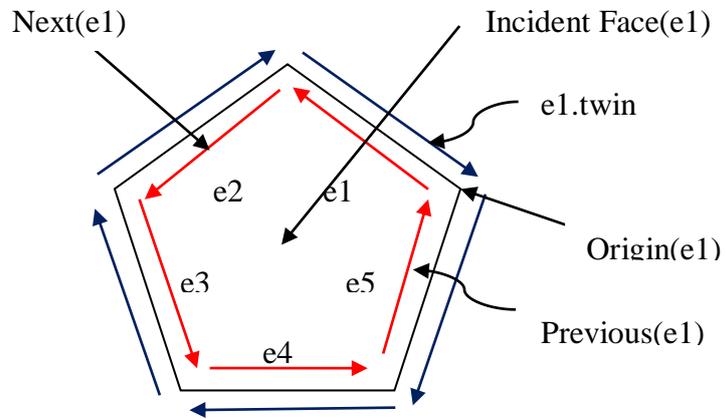


圖 2.7 voronoi diagram 2

1. 點：點 V 紀錄 V 的座標，另外記錄一個指標 $\text{IncidentEdge}(v)$ 指向任意一個把 V 當作 origin 的 half-edge。
2. 面：face F 紀錄記錄一個指標指到可以當作起點逆時針 traverse F 並且在邊界上的 half edge。
3. half- edge : half -edge e 紀錄以下指標：
 - origin(e)
 - twin
 - the face to its left($\text{IncidentFace}(e)$)
 - next(e):next half edge on the boundary of $\text{IncidentFace}(e)$
 - previous(e) : previous half edge

將 DCEL 輸出的多個元件（一群點），將座標值平均以得到需要的點。設某一元件共 n 個點： $(x_1, y_1)(x_2, y_2)(x_3, y_3) \dots \dots \dots (x_n, y_n)$
 則特徵點座標：

$$\left(\frac{\sum_{i=1}^n x_i}{n}, \frac{\sum_{i=1}^n y_i}{n} \right) \quad (2.1)$$

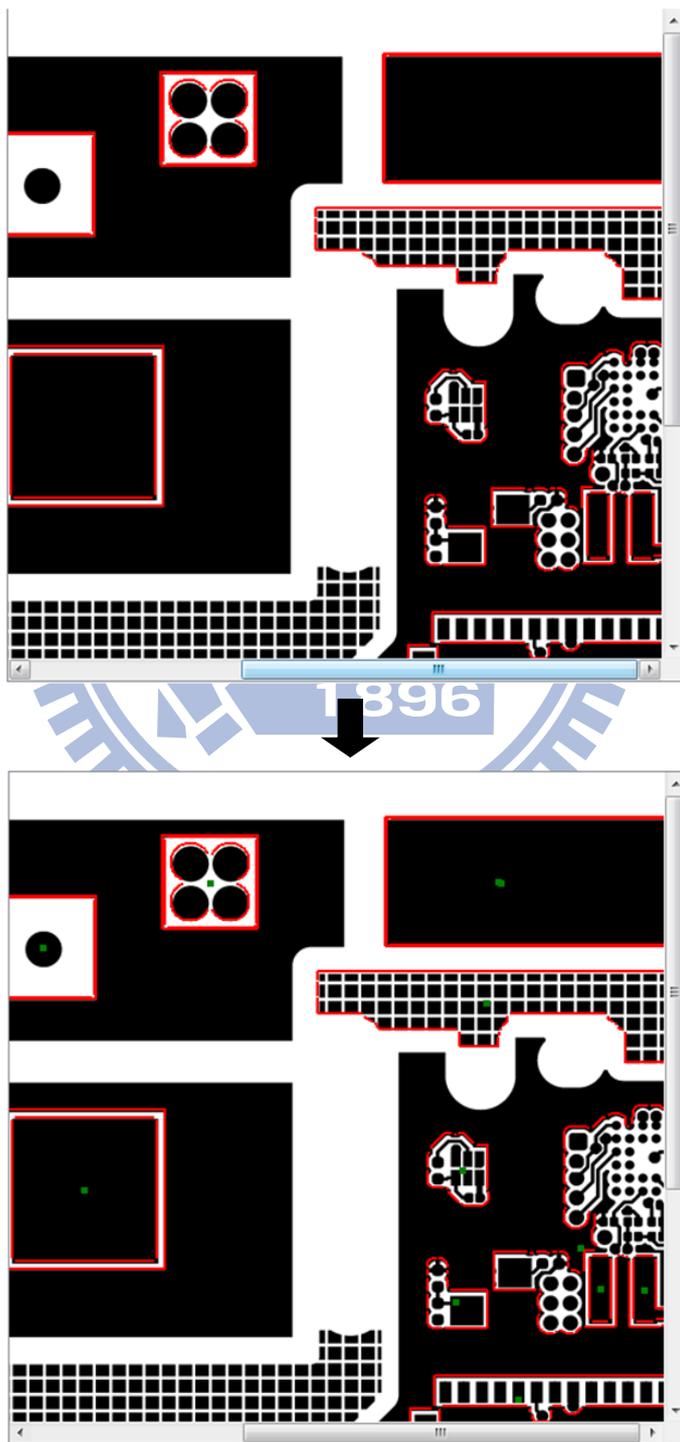


圖 2.8 由 α Shape 得到特徵點

將碰到邊界的元件去掉，因為經過平移、旋轉、縮放後，不完整的元件所求出的特徵點，不正確，並非是整個完整的元件的中心，最後會影響到最後做影像對齊的正確性。

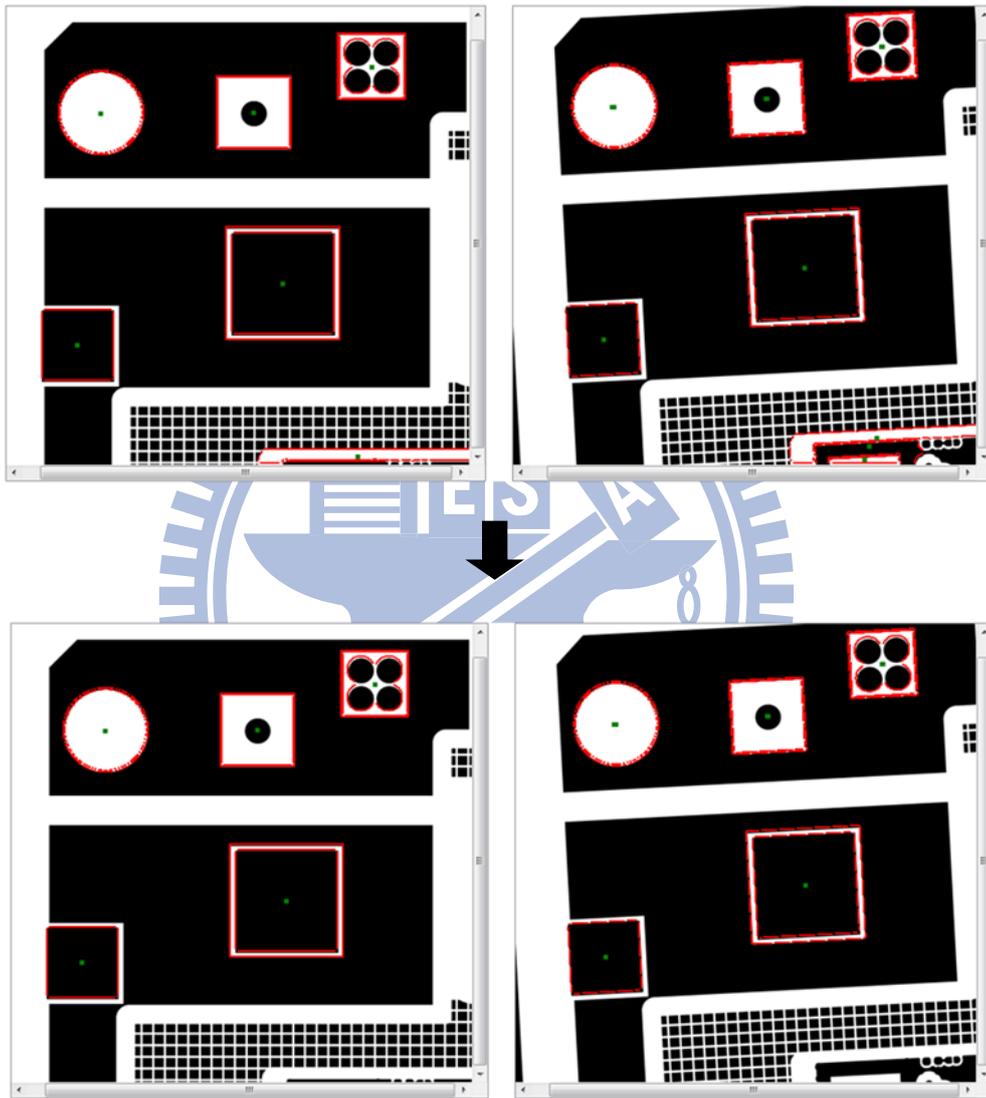


圖 2.9 去除碰到邊界的元件結果

2.3 利用 geometric hashing 取得欲對齊圖片

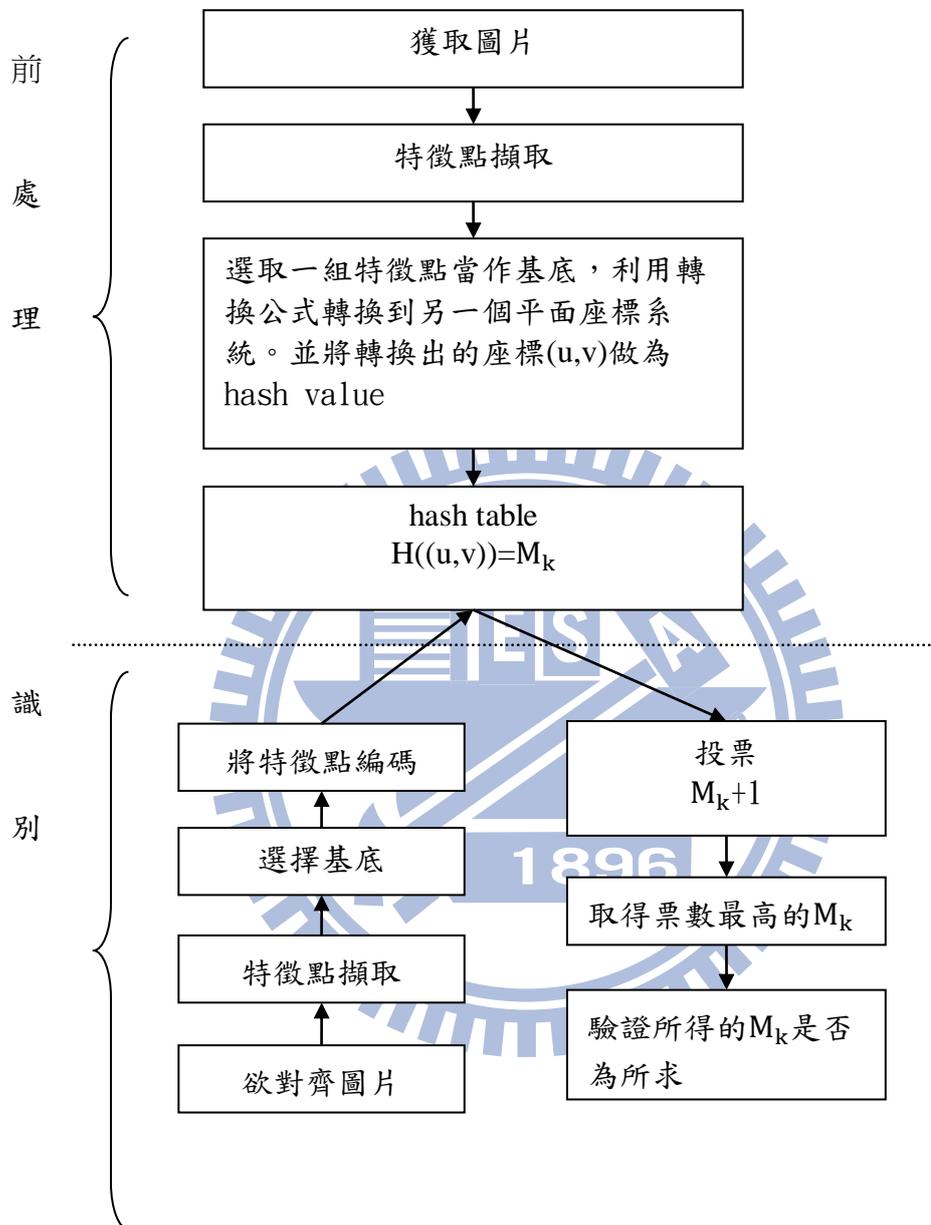


圖 2.10 geometric hashing 流程

1. 前處理

(1) 特徵點擷取：

利用之前敘述的基於 α Shape 的特徵點擷取方式，來取得參考圖片的特徵點。

(2) 建立 hash table

將所有參考圖片的所有特徵點，兩兩當作基底，把所有其他除了基底外的特徵點轉換到另一個平面座標系統共 $c(n,2) \cdot (n-2)$ 個點，因為欲對齊的影像得到的特徵點，並不一定完全與參考影像相同。

轉換公式

$$p - p_0^s = up_x^s + vp_y^s \quad (2.2)$$

解出之 (u,v) 即為轉換到另一個座標平面之座標，為 hash table 的 hash value

p 為除了基底外的其他特徵點

其中：

$$p_0^s = (p_{u1} + p_{u2}) \quad (2.3)$$

$$p_x^s \triangleq (p_{u2} - p_{u1}) \quad (2.4)$$

$$p_y^s \triangleq \text{Rot}_{90}(p_x^s) \quad (2.5)$$

$$p_{u1}、p_{u2} \text{ 為基底。} \quad (2.6)$$

概念為由絕對位置轉換為相對位置。

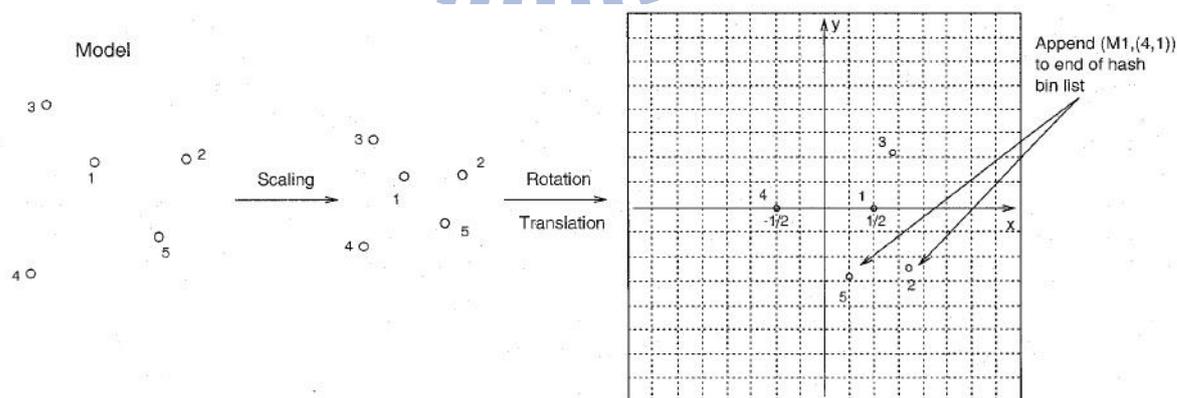


圖 2.11 轉換示意圖

(資料來源：Wolfson, H.J. ; Rigoutsos, I. ,“Geometric Hashing: An overview.”

Computational Science & Engineering, IEEE Volume: 4 , Issue: 4 , pp. 10-21,1997.)

2. 識別

- (1) 取出欲對齊圖片的特徵點，利用之前敘述基於 α Shape 的找特徵點方式。
- (2) 選取基底，利用轉換公式將所有除了基底外的點編碼，將編碼後的 hash value 到 hash table 尋找對應的 hash value。
- (3) 雜湊值 = (u,v)
元素 m
 $H((u,v)) = \text{元素} \rightarrow H((u,v)) = M_k$
- (4) 得到圖片名稱(M_k)後，到另外一個記錄所有已編碼的參考圖片名稱的表格中投票，此表格記錄圖片的相關資訊。
- (5) 投票的步驟結束後，取出得到最高票的圖片。
- (6) 驗證取出的圖片(M_k)，是否是要和欲對齊圖片對齊的參考的圖片。
若是則接著進行影像對齊。
若否則提供攝影機在圖片中的座標，以比對得取正確的圖片。



2.4 特徵點匹配

利用 greedy 的想法來找 Matching

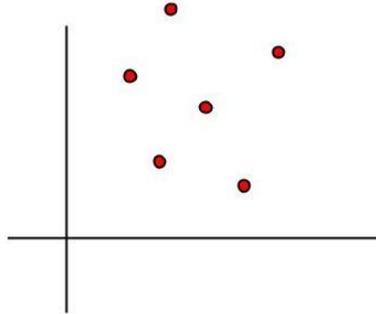


圖 2.12 參考圖片特徵點

假設此為參考的圖片所找出的特徵點

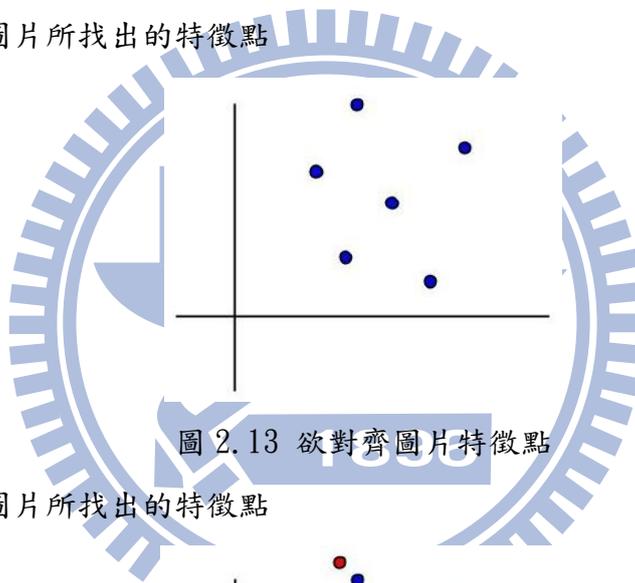


圖 2.13 欲對齊圖片特徵點

假設此為欲對齊圖片所找出的特徵點

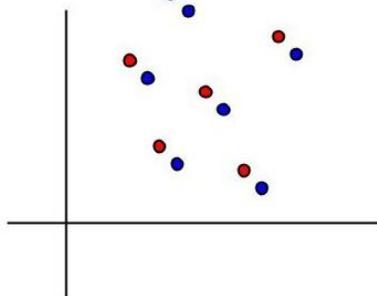


圖 2.14 參考圖片與欲對其圖片特徵點

尋找 match 的方法為，以參考圖片的特徵點為基準，每一個參考圖片特徵點都尋找距離 (歐式距離) 離自己最近的欲對齊圖片的特徵點，來當作是一個 match。直到所有參考圖片的特徵點都找到對應到欲參考圖片的特徵點為止。

2.5 求得影像之間的變換函數

2.5.1 2D motions

在此問題中，需求得 2 張影像之間的的平移、旋轉、縮放的關係

1. Translation:

可寫為:

$$(1) X = X' + t$$

$$(2) X = [I \ t] X'$$

其中 $I: 2 \times 2$ 單位矩陣， $t: 2 \times 1$, 代表平移矩陣， $X' = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$

2. Rotation + Translation

可寫為

$$(1) X = RX' + t$$

$$(2) X = [R \ t] X'$$

其中 $R: \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$

3. Scale + Rotation + Translation 也被稱為 similarity transformation

可寫為

$$(1) X = sRX' + t$$

$$(2) X = [sR \ t] X' = \begin{bmatrix} a & -b & tx \\ b & a & ty \end{bmatrix} X'$$

其中 s 為任意的縮放係數

2.5.2 最小平方法

令所得的 match set 為 $T(X_i) = Y_i, i = 1 \dots n$

$$\begin{cases} (x_{11} & x_{12}) \rightarrow (y_{11} & y_{12}) \\ (x_{21} & x_{22}) \rightarrow (y_{21} & y_{22}) \\ (x_{31} & x_{32}) \rightarrow (y_{31} & y_{32}) \\ & \vdots \\ & \vdots \\ (x_{n1} & x_{n2}) \rightarrow (y_{n1} & y_{n2}) \end{cases}$$

$$\begin{aligned} T(x_1 \ x_2) &= \begin{bmatrix} a \\ d \end{bmatrix} + \begin{bmatrix} b & c \\ e & f \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = [1 \ x_1 \ x_2] \begin{bmatrix} a & d \\ b & e \\ c & f \end{bmatrix} \\ &= \begin{bmatrix} a + b * x_1 & c * x_2 \\ d + e * x_1 & f * x_2 \end{bmatrix} = (y_1 \ y_2) \end{aligned}$$

令 transform 為 X 上式可寫為 $AX=Y$

則所求為

$$\min_x \|AX - Y\| \quad (2.7)$$

其中 $A = \begin{bmatrix} 1 & x_{11} & x_{12} \\ 1 & x_{21} & x_{22} \\ 1 & x_{31} & x_{32} \\ \vdots & \vdots & \vdots \\ 1 & x_{n1} & x_{n2} \end{bmatrix}$

$$Y = \begin{bmatrix} y_{11} & y_{12} \\ y_{21} & y_{22} \\ y_{31} & y_{32} \\ \vdots & \vdots \\ y_{n1} & y_{n2} \end{bmatrix}$$

解為

$$(A^t A)^{-1} A^t Y \quad (2.8)$$

取轉置後：

$$Y^t A (A^t A)^{-1} = \begin{bmatrix} tx & a & -b \\ ty & b & a \end{bmatrix} \quad (2.9)$$

$\begin{bmatrix} tx & a & -b \\ ty & b & a \end{bmatrix}$ 此即為所求之矩陣

2.7 除去錯誤的 match

2.7.1 RANSAC(RANdom SAMple Consensus)

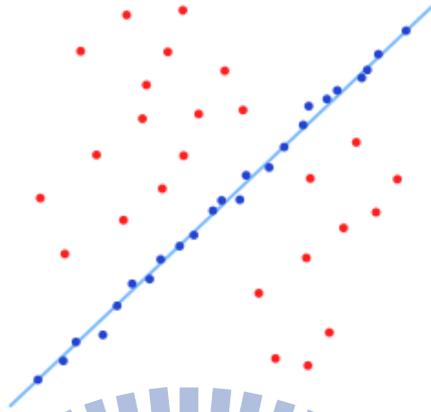


圖 2.15 平面中的一群點藍色:內群點 紅色:離群點
(資料來源: http://zh.wikipedia.org/wiki/File:Fitted_line.svg)

在一組點中找到一條最適合的線。假設，此有一組集合包含了內群以及離群，其中內群為可以被擬合到線段上的點，而離群則是無法被擬合的點。如果用最小平方法來找此線，我們將會無法得到一條適合於內群的線，因為最小平方法會受離群影響而影響其結果。而 RANSAC，可以只由內群來計算出模型，而且機率還夠高。但 RANSAC 無法保證結果一定最好。

使用 RANSAC:

1. 在先前找到的 match set 中，隨機的選一組小的集合。
2. 用這群小的集合利用 least square 算出一個 transformation。
3. 將除了這群小的集合的 match set 帶入，看有多少組能夠符合這個 transformation。
4. 重覆 1~3 足夠多次，以找出一個最佳的 transformation 能夠讓最多的 match 符合。

2.7 平移修正

若圖片有平移、旋轉均會有此問題

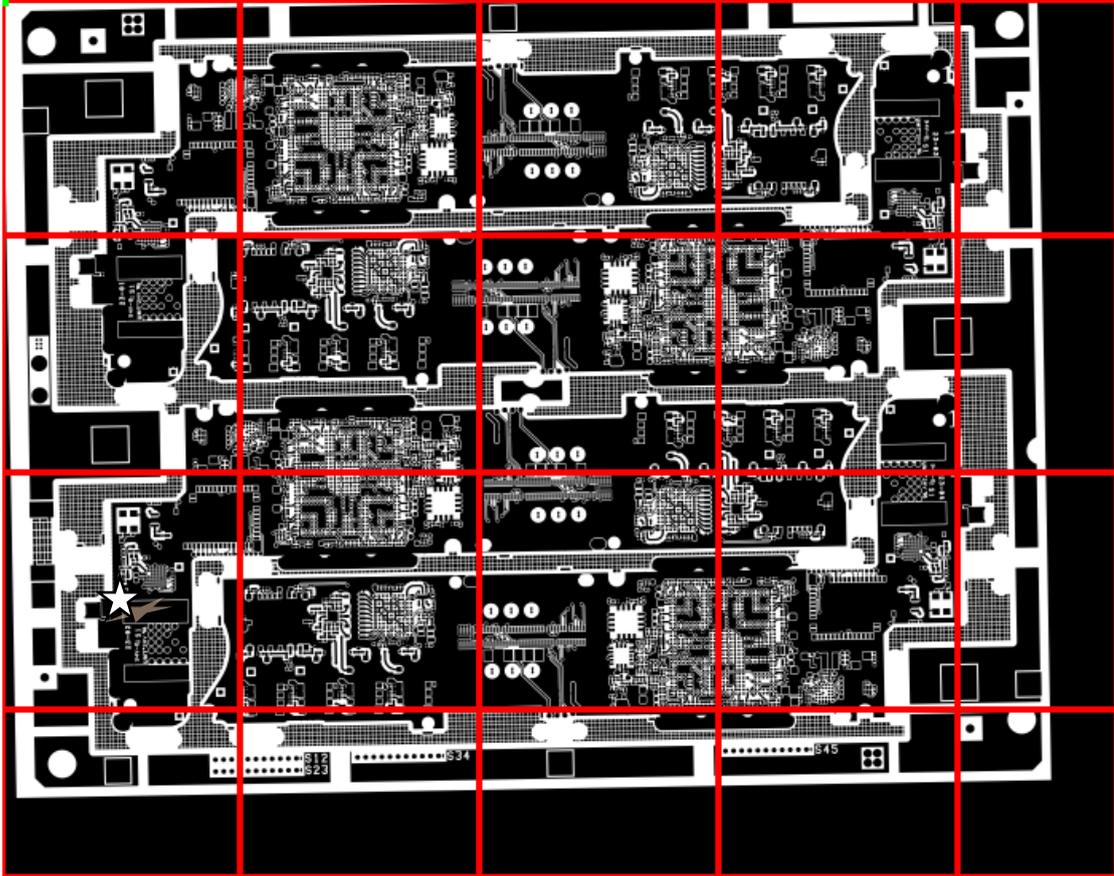


圖 2.16 印刷電路板旋轉後的影像

此為以左上角向右旋轉 1° 的圖片，在☆圖片，跟參考的區塊的圖片做對齊後，會產生平移的參數，但是不應該有平移的參數，只有旋轉，所以需要進行調整。

$$\begin{bmatrix} \alpha & \beta & (1 - \alpha) * \text{center.x} - \beta * \text{center.y} \\ -\beta & \alpha & \beta * \text{center.x} + (1 - \alpha) * \text{center.y} \end{bmatrix} \quad (2.10)$$

center 為真正旋轉圓心的座標

$$\alpha = \text{scale} * \cos \theta$$

$$\beta = \text{scale} * \sin \theta$$

假設☆左上角的座標為(0, 750)，則帶入center為(0, -750)，則最後算出來的平移(x, y)，真正的平移(Real x, Real y)。

$$\text{Real } x = x - [(1 - \alpha) * \text{center.x} - \beta * \text{center.y}]$$

$$\text{Real } y = y - [\beta * \text{center.x} + (1 - \alpha) * \text{center.y}]$$

2.8 決定平移、旋轉、縮放參數

將調整過後的圖片與參考圖片的每一個像素兩兩做互斥，像素值 0 為 0，像素值 255 為 1，最後再將每個像素互斥的結果全部相加，此數值表示 2 張圖之間的相似度，5 張調整過後的圖片與原圖互斥總和其中最小的為最佳的調整參數。

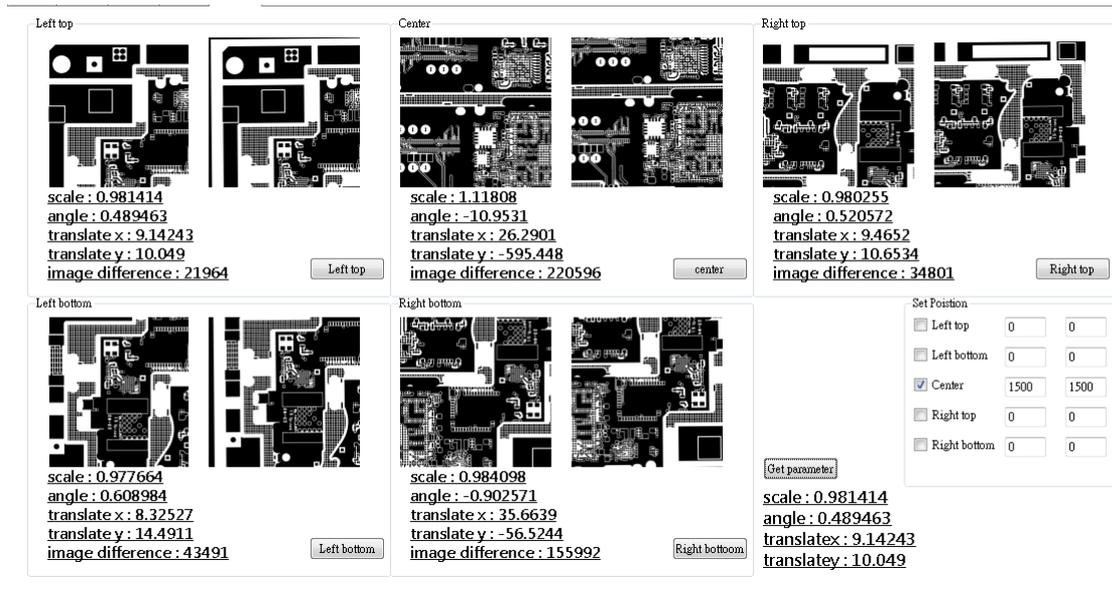


圖 2.17 系統取得參數的畫面

第三章 實驗

3.1 實驗結果

本系統為 QT 搭配 opencv 2.4.5 作業系統為 windows 7。

本實驗使用 4 種較不相同的印刷電路板分別為(a)(b)(c)(d)，來進行對齊的精確度的實驗。

四張圖的解析度分別為 (a) 12448*10805 (b) 12992*10111 (c) 15327*11727 (d) 8542*8208

實驗分為兩種情況第一種情況為有五組參數，固定縮放、 平移係數，改變旋轉係數，因為實際情況，縮放係數是固定不變的，而平移又受旋轉係數影響，所以只改變旋轉係數，第二種情況有四組參數縮放、旋轉、平移係數隨機產生。

旋轉的係數範圍在 ± 1 ，因為若超過前述的範圍，肉眼即可分辨出沒有擺正，所以不考慮超過前述列舉出的範圍的情況。

實驗時將原圖縮小 1/4，並切成多塊 750*750 的圖片，因為考慮到找尋圖片中元件的時間，並且將原圖對齊左上角，當做縮放、旋轉的圓心。

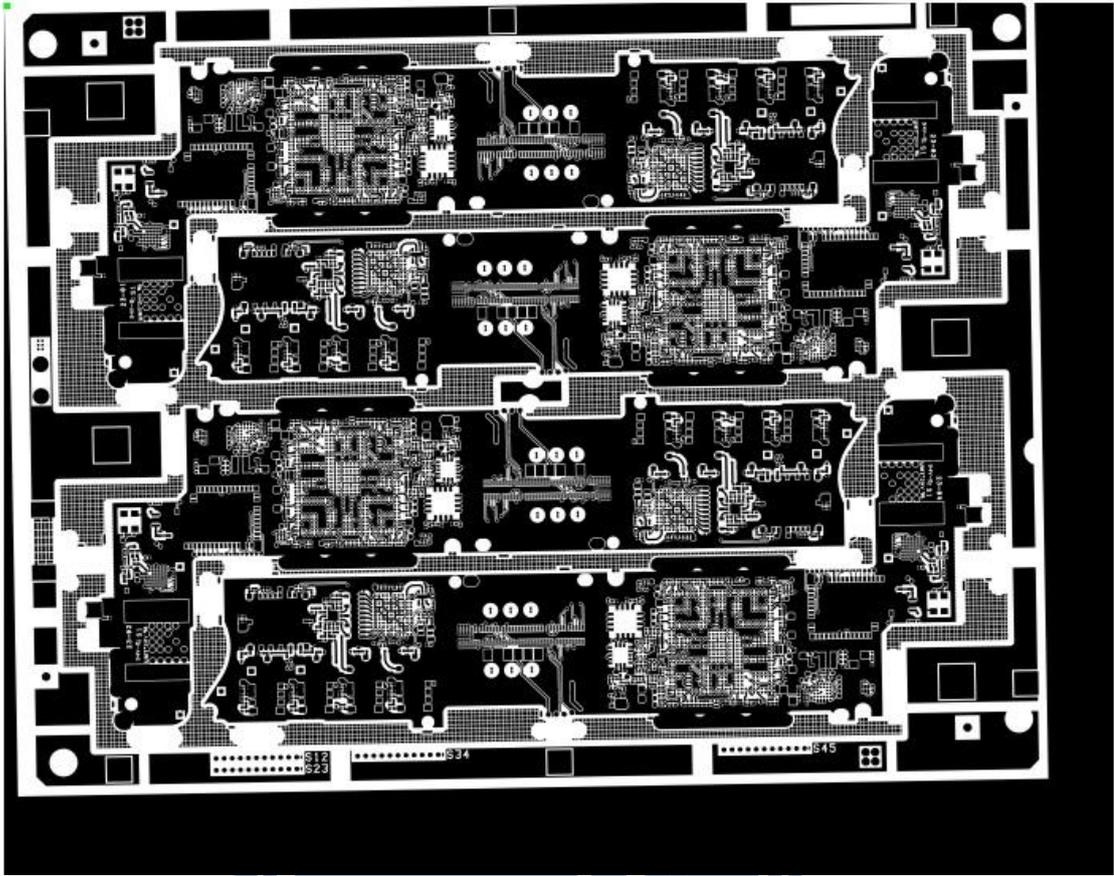


圖 3.1 印刷電路板旋轉 1 度的影像



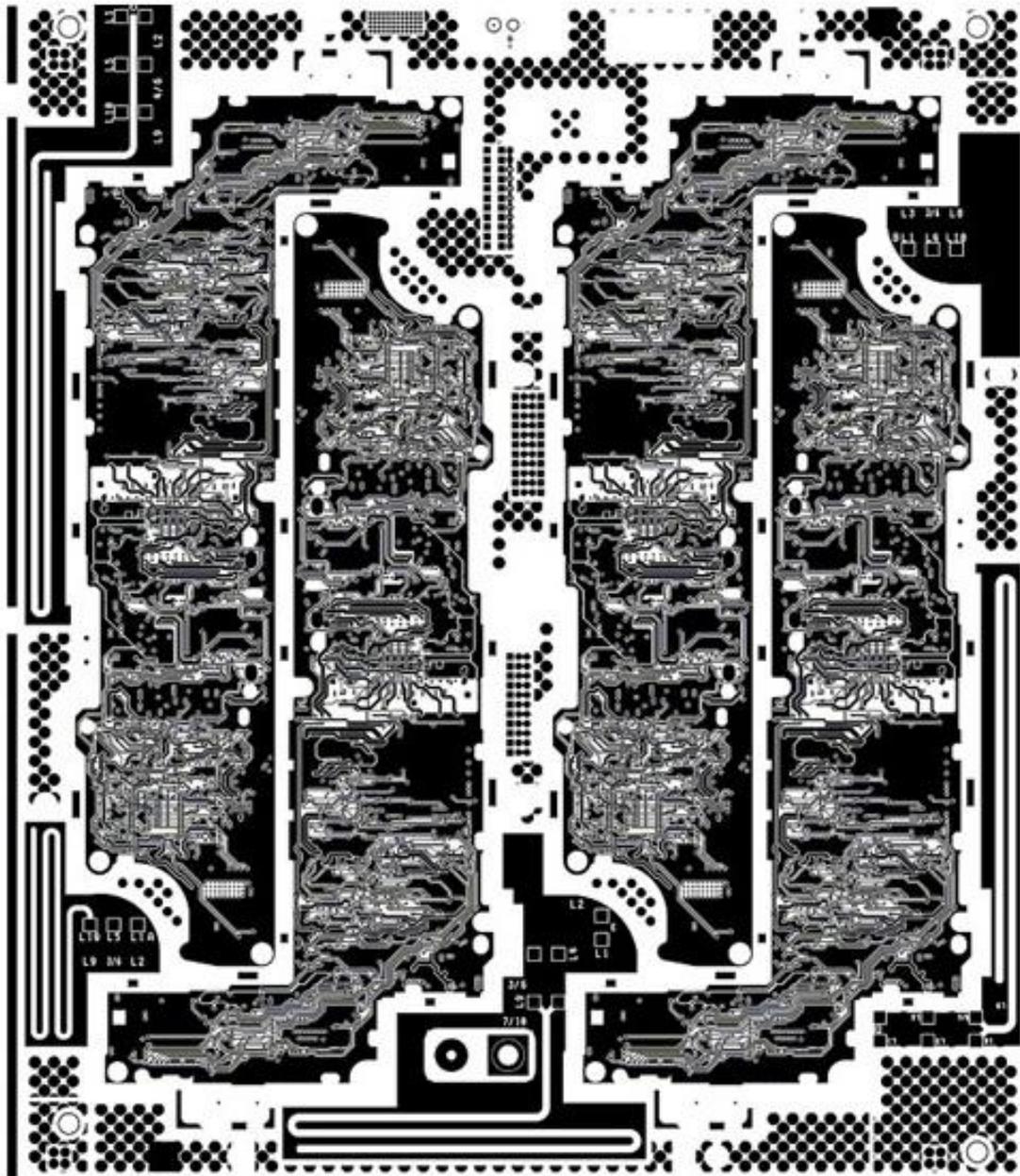


圖 3.2 印刷電路板(a)

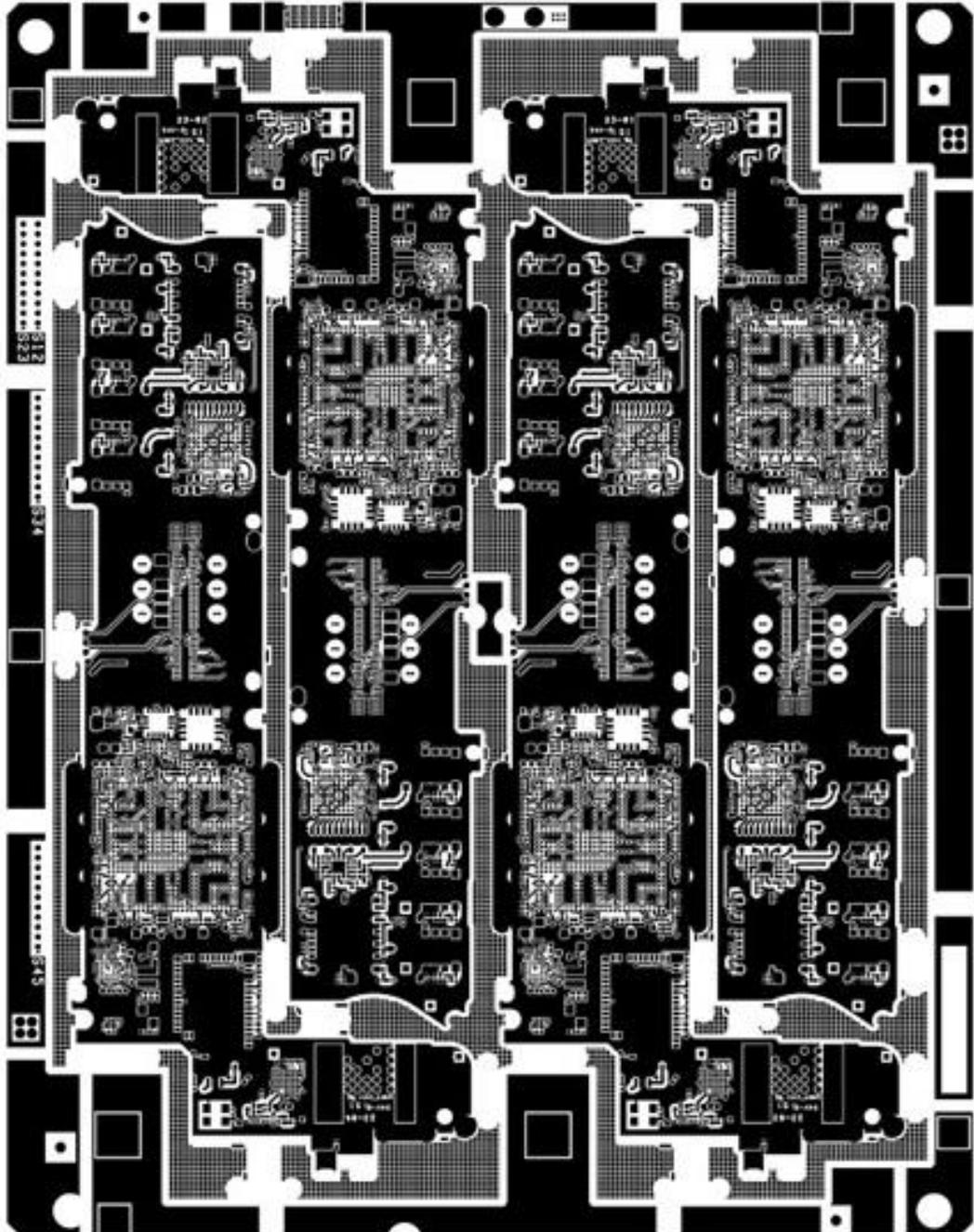


圖 3.3 印刷電路板(b)

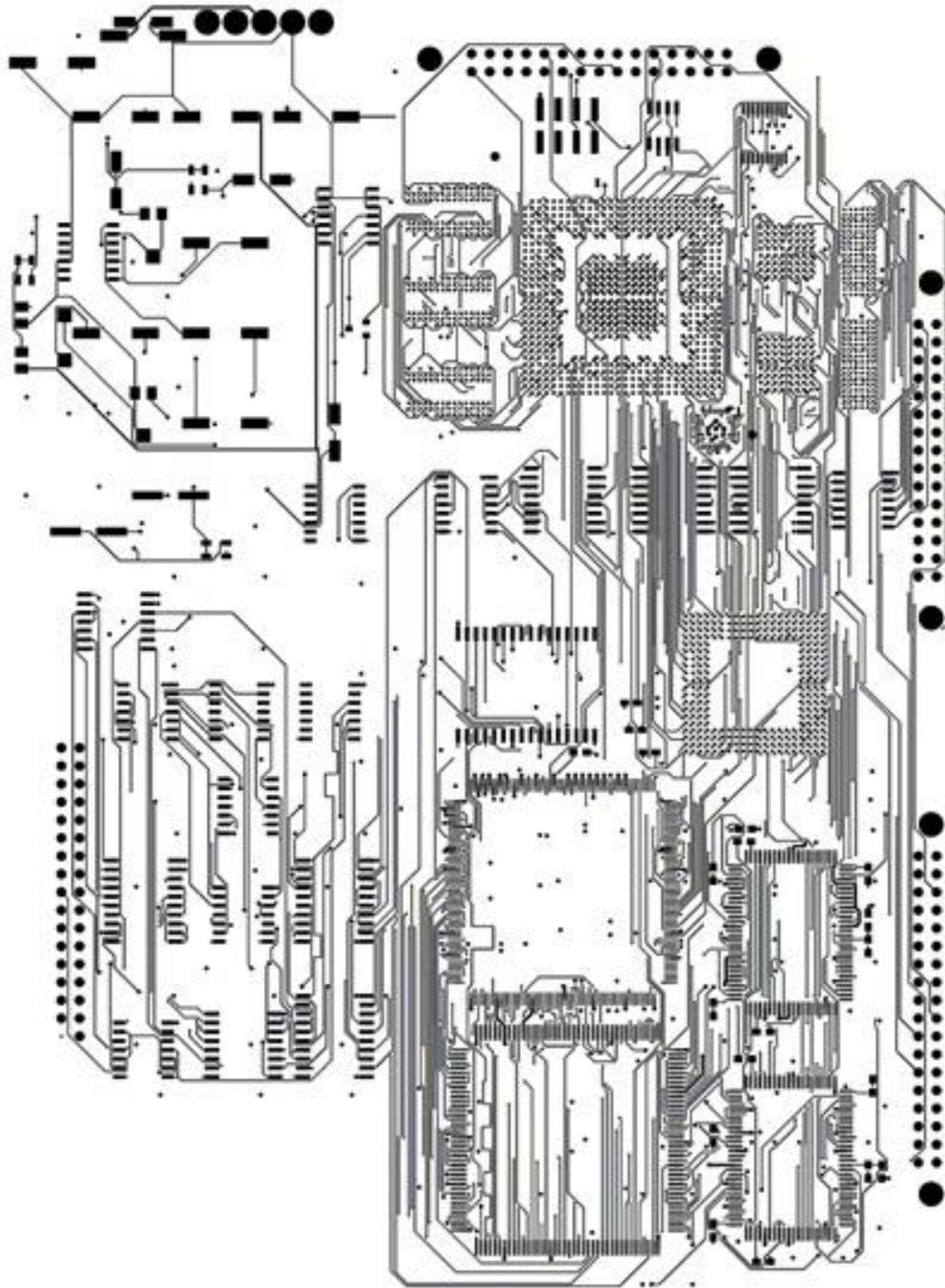


圖 3.4 印刷電路板(c)

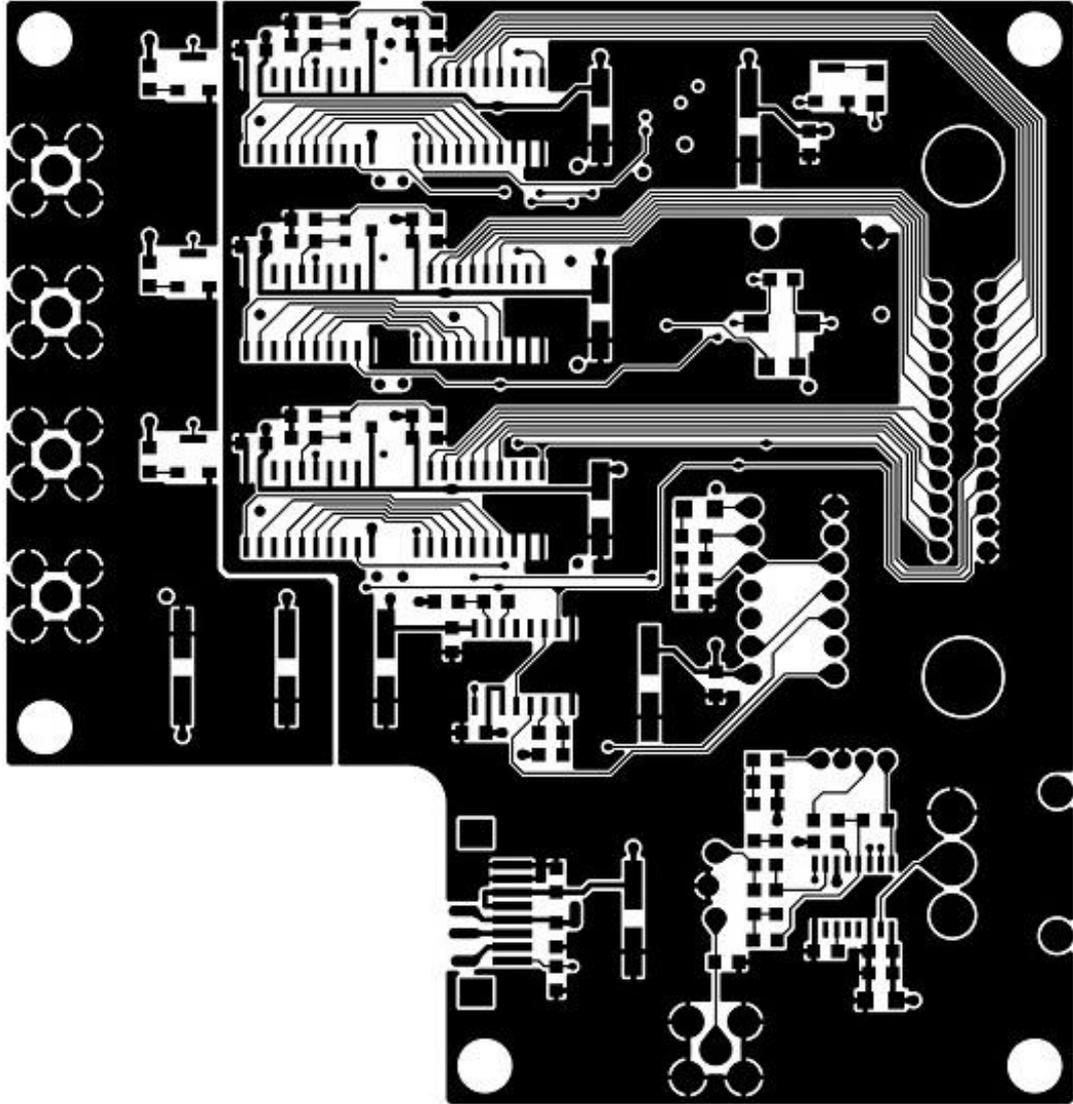


圖 3.5 印刷電路板(d)

| 對齊結果 | | | | | | |
|------|----|--|--|---------------------------------------|---|---------------------------------------|
| 模擬參數 | | 縮放:0.98 旋轉:-1 平移 X:10 平移 Y:10 | 縮放:0.98 旋轉:-0.5 平移 X:10 平移 Y:10 | 縮放:0.98 旋轉:0 平移 X:10 平移 Y:10 | 縮放:0.98 旋轉:0.5 平移 X:10 平移 Y:10 | 縮放:0.98 旋轉:1 平移 X:10 平移 Y:10 |
| 對齊結果 | | | | | | |
| (a) | 縮放 | 0.9849 | 0.9799 | 0.9800 | 0.9799 | 0.9879 |
| | 旋轉 | -0.8473 | -0.5179 | 0.020 | 0.5179 | 0.6716 |
| | 平移 | 7.20 4.46 | 10.17 9.53 | 9.86 9.20 | 10.17 9.53 | 9.879 8.143 |
| (b) | 縮放 | 0.9717 | 0.9790 | 0.9794 | 0.9814 | 0.9788 |
| | 旋轉 | -1.1899 | -0.4022 | 0.0563 | 0.4894 | 0.98 |
| | 平移 | 7.01 9.06 | 7.16 12.44 | 9.124 11.263 | 9.14 10.04 | 12.16 9.24 |
| (c) | 縮放 | 0.9782 | 0.9811 | 0.9770 | 0.9805 | 0.9827 |
| | 旋轉 | -0.9382 | -0.3297 | -0.0805 | 0.5118 | 0.9041 |
| | 平移 | 8.39 12.90 | 8.25 12.58 | 11.86 9.36 | 9.22 11.31 | 8.29 4.64 |
| (d) | 縮放 | 0.9808 | 0.9806 | 0.9798 | 0.9796 | 0.9811 |
| | 旋轉 | -1.0289 | -0.4961 | 0.0577 | 0.5147 | 1.1353 |
| | 平移 | 8.56 10.01 | 9.16 9.49 | 9.92 11.64 | 10.23 10.09 | 9.83 13.59 |

表 3.1 對齊結果 1(平移單位:pixel)

| 誤差 | | | | | | | 平均 誤差 |
|------------|--|--|---------------------------------------|---|---------------------------------------|--------------|----------------|
| 模擬參數 誤差 | 縮放:0.98 旋轉:-1 平移 X:10 平移 Y:10 | 縮放:0.98 旋轉:-0.5 平移 X:10 平移 Y:10 | 縮放:0.98 旋轉:0 平移 X:10 平移 Y:10 | 縮放:0.98 旋轉:0.5 平移 X:10 平移 Y:10 | 縮放:0.98 旋轉:1 平移 X:10 平移 Y:10 | | |
| (a) | 縮放 | 0.0049 | 0.0001 | 0.0000 | 0.0001 | 0.0079 | 0.0026 |
| | 旋轉 | 0.1527 | 0.0179 | 0.0200 | 0.0179 | 0.3284 | 0.1433 |
| | 平移 | 2.79 5.53 | 0.17 0.47 | 0.14 0.80 | 0.17 0.47 | 0.12 1.85 | 0.67 1.824 |
| (b) | 縮放 | 0.0083 | 0.0010 | 0.0006 | 0.0014 | 0.0012 | 0.0125 |
| | 旋轉 | 0.1899 | 0.0978 | 0.0563 | 0.0106 | 0.02 | 0.0749 |
| | 平移 | 2.98 0.93 | 2.84 2.44 | 0.87 1.26 | 0.86 0.04 | 2.16 0.76 | 1.942 1.086 |
| (c) | 縮放 | 0.0018 | 0.0011 | 0.0030 | 0.0005 | 0.0027 | 0.0018 |
| | 旋轉 | 0.0618 | 0.1703 | 0.0805 | 0.0118 | 0.0959 | 0.084 |
| | 平移 | 1.60 2.90 | 1.75 2.58 | 1.86 0.64 | 0.78 1.31 | 1.71 5.36 | 1.54 2.558 |
| (d) | 縮放 | 0.0008 | 0.0006 | 0.0002 | 0.0004 | 0.0011 | 0.0006 |
| | 旋轉 | 0.0289 | 0.0039 | 0.0577 | 0.0147 | 0.1353 | 0.0481 |
| | 平移 | 1.44 0.01 | 0.84 0.51 | 0.08 1.64 | 0.23 0.09 | 0.17 3.59 | 0.552 1.168 |

表 3.2 對齊誤差 1(平移單位:pixel)

| 對齊結果 | | | | | |
|------|----|--|---|--|--|
| 模擬參數 | | 縮放:0.978 旋轉:1.29 平移 X:7 平移 Y:-6 | 縮放:0.992 旋轉:-0.287 平移 X:10 平移 Y:-5 | 縮放:1.01 旋轉:0.356 平移 X:-8 平移 Y:5 | 縮放:1.0005 旋轉:0.05 平移 X:1 平移 Y:1 |
| 結果 | | | | | |
| (a) | 縮放 | 0.973 | 0.9914 | 1.0106 | 1.0003 |
| | 旋轉 | 1.3172 | -0.3173 | 0.4080 | 0.1052 |
| | 平移 | 10.56 -5.96 | 10.46 -5.24 | -8.71 5.13 | 0.58 1.62 |
| (b) | 縮放 | 0.9782 | 0.9925 | 1.0096 | 1.0018 |
| | 旋轉 | 1.1902 | -0.5545 | 0.2882 | 0.0110 |
| | 平移 | 9.88 -8.58 | 8.80 -5.98 | -7.07 4.56 | -2.40 -1.51 |
| (c) | 縮放 | 0.9687 | 0.9925 | 1.0116 | 1.0002 |
| | 旋轉 | 1.1556 | -0.3090 | 0.2659 | 0.0383 |
| | 平移 | 16.98 -6.86 | 9.52 -4.40 | -8.87 4.33 | 1.77 0.89 |
| (d) | 縮放 | 0.984 | 0.9935 | 1.0097 | 0.9998 |
| | 旋轉 | 1.2399 | -0.3673 | 0.3698 | 0.0697 |
| | 平移 | 10.56 -5.34 | 8.57 -8.64 | -8.55 5.44 | 1.08 2.40 |

表 3.3 對齊結果 2(平移單位:pixel)

| 誤差 | | | | | | |
|------|--|---|--|--|--------------|----------------|
| 模擬參數 | 縮放:0.978 旋轉:1.29 平移 X:7 平移 Y:-6 | 縮放:0.992 旋轉:-0.287 平移 X:10 平移 Y:-5 | 縮放:1.01 旋轉:0.356 平移 X:-8 平移 Y:5 | 縮放:1.0005 旋轉:0.05 平移 X:1 平移 Y:1 | 平均誤差 | |
| (a) | 縮放 | 0.005 | 0.0006 | 0.0006 | 0.0002 | 0.0016 |
| | 旋轉 | 0.0272 | 0.0303 | 0.052 | 0.0552 | 0.0411 |
| | 平移 | 3.56 0.04 | 0.46 0.24 | 0.71 0.13 | 0.42 0.62 | 1.28 0.25 |
| (b) | 縮放 | 0.0002 | 0.0005 | 0.0004 | 0.0013 | 0.0006 |
| | 旋轉 | 0.0998 | 0.2675 | 0.0678 | 0.06 | 0.1237 |
| | 平移 | 2.88 2.58 | 1.2 0.98 | 0.93 0.44 | 3.40 2.51 | 2.1 1.62 |
| (c) | 縮放 | 0.0093 | 0.0005 | 0.0016 | 0.0003 | 0.0029 |
| | 旋轉 | 0.1344 | 0.022 | 0.0901 | 0.0117 | 0.06 |
| | 平移 | 9.98 0.86 | 0.48 0.6 | 0.87 0.67 | 0.77 0.11 | 3.025 0.56 |
| (d) | 縮放 | 0.006 | 0.0015 | 0.0003 | 0.0007 | 0.0021 |
| | 旋轉 | 0.0501 | 0.0803 | 0.0138 | 0.0197 | 0.0409 |
| | 平移 | 3.56 0.66 | 1.43 3.64 | 0.55 0.44 | 0.08 1.40 | 1.405 1.535 |

表 3.4 對齊誤差 2(平移單位:pixel)

經過以上實驗可以發現在旋轉±1度以內，此方法在這四種類型的電路版圖片可以提供不錯的精確度。不過對齊的精確度仍需依賴 α 值的選擇，需要多次嘗試各種可能的 α 值。圖(c)的最大誤差要比(a)(b)(c)來的大，因為圖(c)都是線，不像(a)(b)(c)有比較鮮明的特徵的元件。

第四章 結論與未來展望

利用此方法，可以不需要去尋找電路板中的特定記號，即可以完成定位誤差的調整，只要印刷電路板的擺放傾斜角度不超過 $\pm 1^\circ$ 、以及位移不要差太多，即可做到，並且有不錯的精確度。

利用基於 α Shape 的特徵點擷取方式，可以達到旋轉、平移、縮放、不變，找出的特徵點是元件在電路板上的相對位置，而不像一般常用的角點提取的方法，找出的特徵點較為雜亂而且很多。

系統中仍有部分參數在做誤差調整之前仍需要手動調整，才能夠達到較佳的結果，以及辨別圖片是否高度相關仍無找出準確度較高的方式。

在未來，希望可以讓系統自動決定較佳的參數如 α 值，以及把準確度較高、速度快的辨別圖片是否為高度相關的方法加入 Geometric hashing 中，讓系統可以接近全自動的去完成誤差調整的工作。



第五章 參考文獻

- [1] Yehezkel Lamdan & Haim J. Wolfson, “Geometric hashing: A general and efficient model-based recognition scheme” ,Computer Vision., Second International Conference on ,p. 238-249 , 1988.
- [2] Wolfson, H.J. ; Rigoutsos, I. , “Geometric Hashing: An overview.” Computational Science & Engineering, IEEE Volume: 4 , Issue: 4 , pp. 10-21,1997.
- [3] Barbara Zitova*, Jan Flusser, “Image registration methods: a survey” Image and Vision Computing, Volume 21, Issue 11, October 2003, Pages 977–1000,2003.
- [4] Martin A. Fischler and Robert C. Bolles. “Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography”. Comm. of the ACM 24 (6): 381–395,1981.
- [5] Lowe, D.G. , “Distinctive Image Features from Scale-Invariant ” International Journal of Computer Vision, 60, 2, pp. 91-110, 2004.
- [6] R. Szeliski, “Image Alignment and Stitching:A Tutorial1,” Handbook of Mathematical Models in Computer Vision, pp.273–292, 2005.
- [7] Edelsbrunner, Herbert; Kirkpatrick, David G.; Seidel, Raimund , "On the shape of a set of points in the plane", Information Theory, IEEE Transactions on Volume: 29 , Issue: 4 , Page(s): 551 - 559 ,1983.