

中華民國第 57 屆中小學科學展覽會

作品說明書

國小組 生活與應用科學科

第二名

080821

罰寫不再痛苦了～快速創造個人化手寫字體

學校名稱：新北市永和區秀朗國民小學

作者： 小六 潘楷翔	指導老師： 朱中梧 謝雅雯
-------------------	-----------------------------

關鍵詞：部件、手寫字、分解

摘要

當要抄寫文章的時候，光是看到很多字就會讓人覺得厭煩，其中還會發現很多字重複出現（例如：的、都、會...等）；不只如此，同偏旁或是部首的部分也重複出現頗多。因此想藉由這次的研究，把手寫的文字，藉由電腦程式將其拆解成多個部份，再和其他部份重新結合，即可組成更多不同的字，若計劃一下輸入之文字，便可簡便的組成一句話甚至一篇文章，再者，這些字都是由自己手寫出來的，因此，創造出來的字便如同一組自己的手寫字體，不但使用上具有個人特徵，而且列印出來之後，就如同一個一個我的個人化手寫字。

壹、研究動機

一天，我被老師罰寫一百遍的「先舉手再發言」這一句話，我當時就想，如果我把這六個字在紙上手寫一遍後掃描下來，再利用電腦的「複製」與「貼上」功能重複一百遍，就可以變成我所寫的字後再列印出來，這樣就不用擔心被罰寫了！但是實作後發現問題百出，掃描出現的字體背景有淡淡的黃色、貼上數個字串後又發現字串會同時偏向同一個角度，以及一行字結束要換行時，得切斷字串並重新組合或分離，這些都要花很多時間做調整，反而不符合經濟效益。因此，引發我想研究出能簡單創造手寫字體方法的動機。

五年級下學期，學校電腦課開始教 Scratch 程式設計，當時便想利用它來設計這個程式，但是當程式執行中要輸入手寫字體圖檔時出現了困難，且無法做到圖形分割及處理；為達到這個目的。於是，我開始自學 Visual C++ 程式語言，從圖形輸入、複製貼上、圖片分割處理等，雖經多次的版本修訂與改進，但還是有瑕疵，因此參考網路上的相關研究，並重新進行多次的實驗與實作，希望能改善手寫字體的品質，快速創造個人化手寫字體。

貳、研究目的

- 一、讓電腦重組出的字與使用者寫的字能更為相似。
- 二、加快運作速度。
- 三、以最少的手寫字來創造出最多的電腦模擬字。
- 四、透過系統的拆解與重組，創造出大量個人化的字體。

參、研究器材

一、桌上型電腦

(一) 硬體

1. 中央處理器：Intel(R) Core(TM)2 CPU 6400 @2.13GHz
2. 記憶體：4 GB
3. 硬碟：160 GB

(二) 軟體

1. 作業系統：MS-Windows 7 專業版
2. 程式語言：Microsoft Visual Studio 2013
3. 視覺資料庫：OpenCV 3.0

二、筆記型電腦

(一) 硬體

1. 中央處理器：Intel(R) Core(TM)2 Duo CPU T6570 @2.10GHz
2. 記憶體：3 GB
3. 硬碟：500 GB

(二) 軟體

1. 作業系統：MS-Windows 7 專業版
2. 程式語言：Microsoft Visual Studio 2013
3. 視覺資料庫：OpenCV 3.0

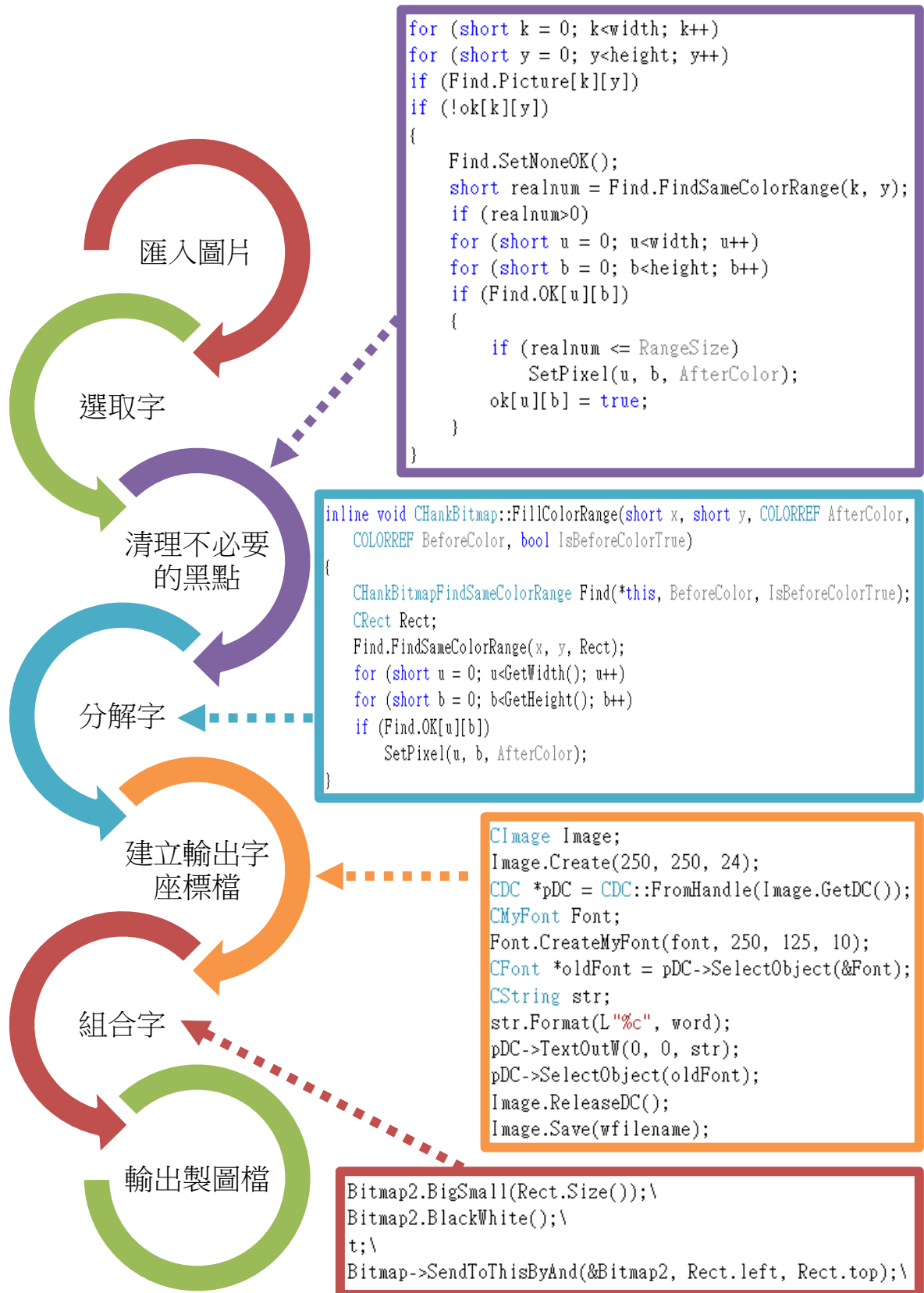
三、事務機：HP CLJ1312

四、手寫板

五、觸控筆

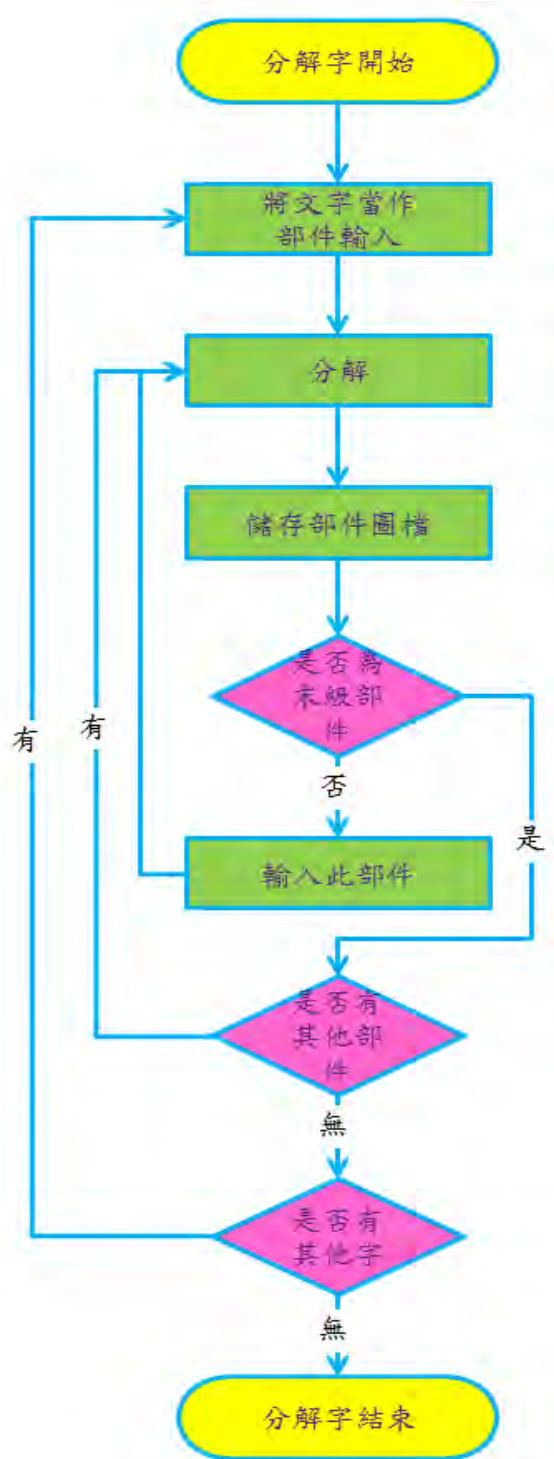
肆、程式流程

一、總流程圖

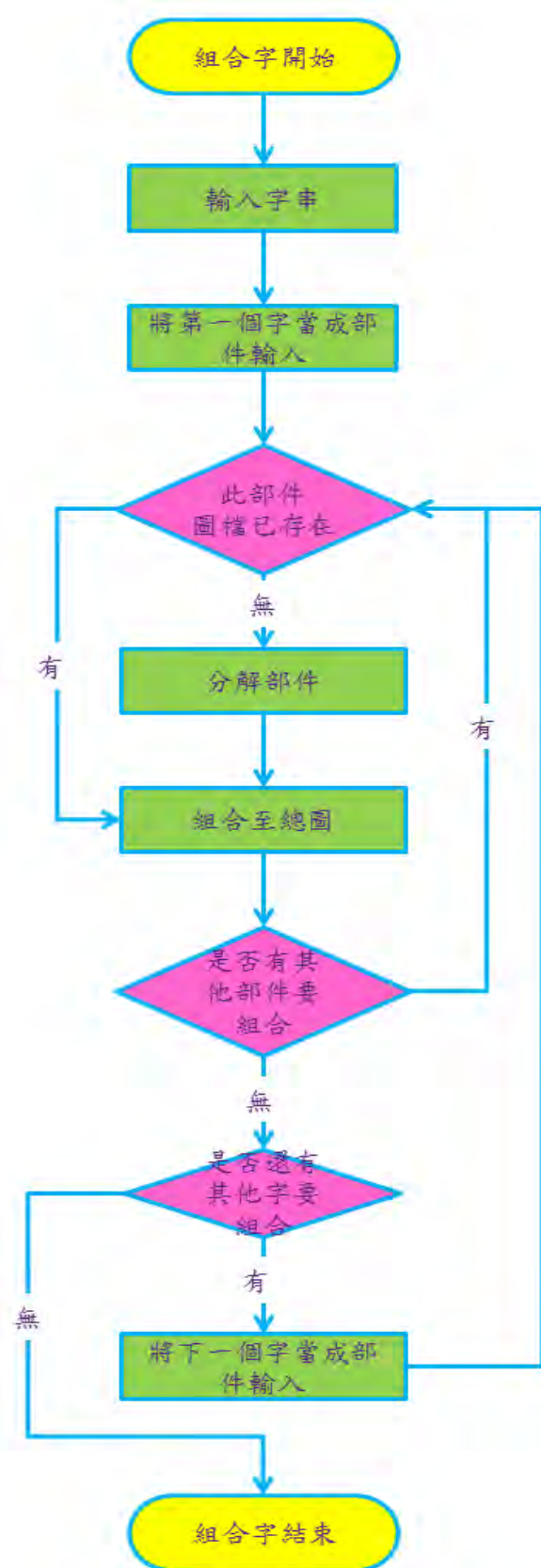


二、函數流程圖

(一) 分解字



(二) 組合字



伍、研究過程與方法

一、名詞解釋

(一) 部件

由一個手寫字拆解出來的部分就稱為部件，如【噁】是由【口】、【因】、【心】這三個部件所組成。

(二) 座標檔

副檔名為 .bh 的檔案稱作座標檔，內容是記載著一個組合字內部的各種部件種類編號、座標位置和長寬比例，做為組合一個字時的依據。

(三) 圖檔

圖檔包含所有未拆解的字及其部件，利用座標檔中的詳細資料便可將圖檔中分解完成的部件放置在正確的位置以組成新的組合字。

(四) 末級部件、次級部件

如果一個手寫字，已經無法再被分解，就稱作末級部件；相反的，如果一個部件，還可以再分解，就稱作次級部件。例如：【型】可以分解成【土】和【刑】，【土】已經不能再分解了，所以【土】是屬於末級部件，而【刑】則還可以再分解成【开】和【丩】，所以【刑】是屬於次級部件。

(五) 圖片黑白化

通常字的圖檔只會有兩種顏色，文字色和背景色

，系統中預設文字色為黑色

，背景色為白色，所以在輸入圖檔的時候，系統都會自動將圖檔黑白化，深色的部分轉變為黑色，淺色的部分則轉變為白色。其方法如下，將圖片中的每個點進行紅、綠、藍色彩分析，若三色中有其中一色之數值小於 128，即改變此點為黑色，其餘的點則改為白色(註：電腦裡點陣圖的色彩是使用紅、綠和藍的三原色所組成，數值越小顏色越深，數值越大則顏色越淺。顏色的數值由 0 到 255 組成，所以選擇以 256 的一半 128 作為區分顏色的基準)。因此，繪圖



常用的十二色相環中的顏色都會被轉換成黑色，如此系統才能將所有不同色彩的筆跡都轉換為黑色。

(六) OpenCV

OpenCV 的全名是 Open Source Computer Vision Library，是一個跨平台的電腦視覺庫，可以在商業和研究領域中免費使用。OpenCV 可用於開發即時的影像處理、電腦視覺以及模式識別程式。本研究使用的是 OpenCV 3.0 版本。

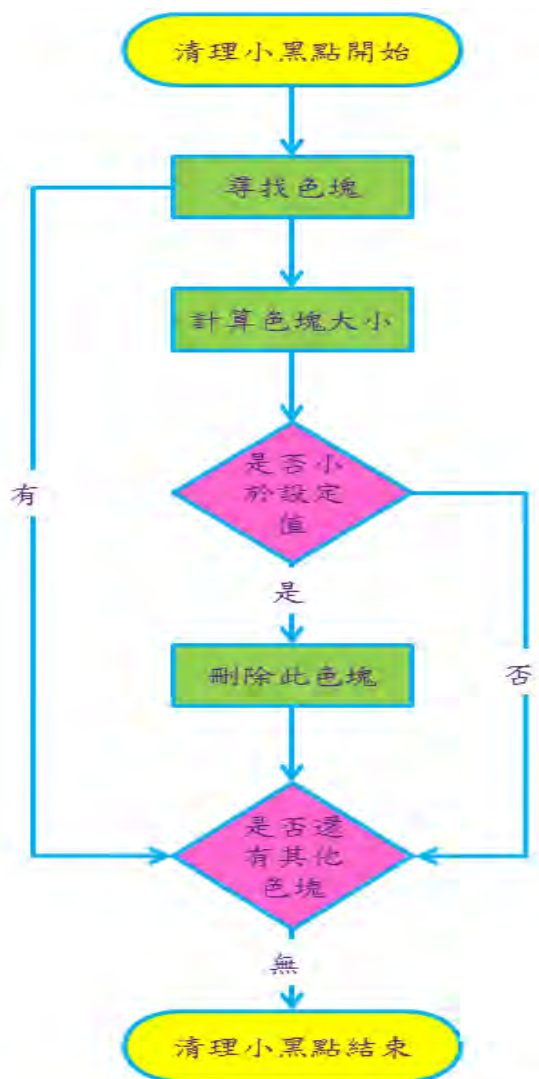
二、系統運作說明

(一) 使用點陣圖輸入手寫字

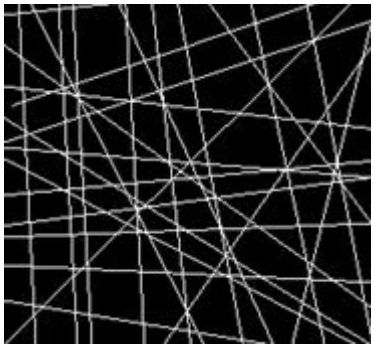
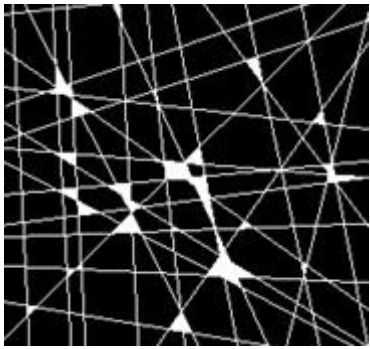
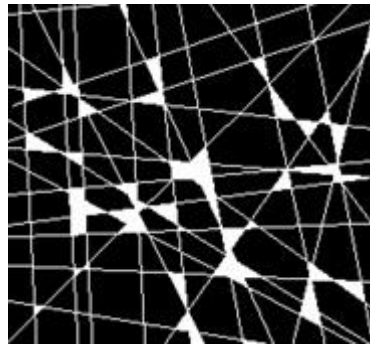
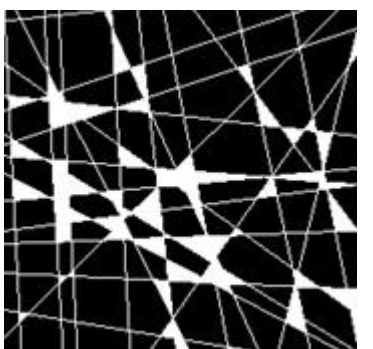
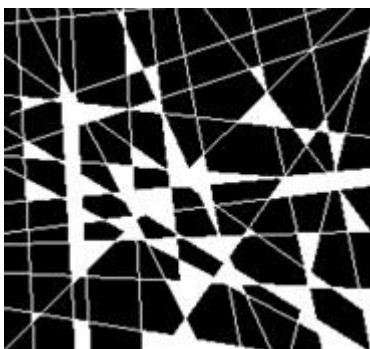
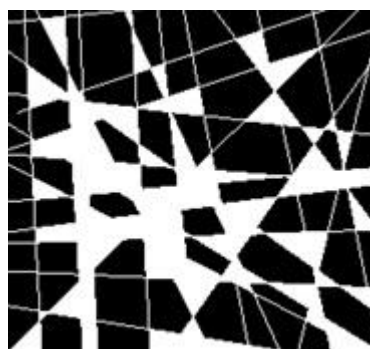
原始的手寫字點陣圖不一定要在自己的系統裡產生，可由掃描機、相機、手寫板、甚至小畫家產生的也都可以使用，而且不會因為原始是向量的圖形而消失了許多細節，雖然點陣圖會需要較多的人力來做辨識的動作，但是圖像能較為精細，會使用到的電腦計算功能較少，而且不佔據太多空間。

(二) 圖形淨化

在分解手寫字之前，系統會先將整個圖片黑白化，接著將圖片最靠邊的黑點設定為新的邊界。之後需要透過程式來去除掃描或拍照時產生的多餘小黑點，方法是由使用者設定一個數值作為去除黑點的大小基準，程式只要搜尋圖形中色塊面積小於所設定的數值，那個色塊就會被去除。這是運用計算色塊大小的方法，將圖形中每個色塊都帶入函數來計算色塊的大小，要是計算結果小於設定的數值，就把那一整個色塊用白色代替，這樣就能消除圖形中不必要的色塊或是斑點。



以圖片說明如下：

設定數值	0	50	100
結果			
設定數值	150	200	250
結果			

(註：黑點被刪除後會變成白色區塊)

(三) 選取部件

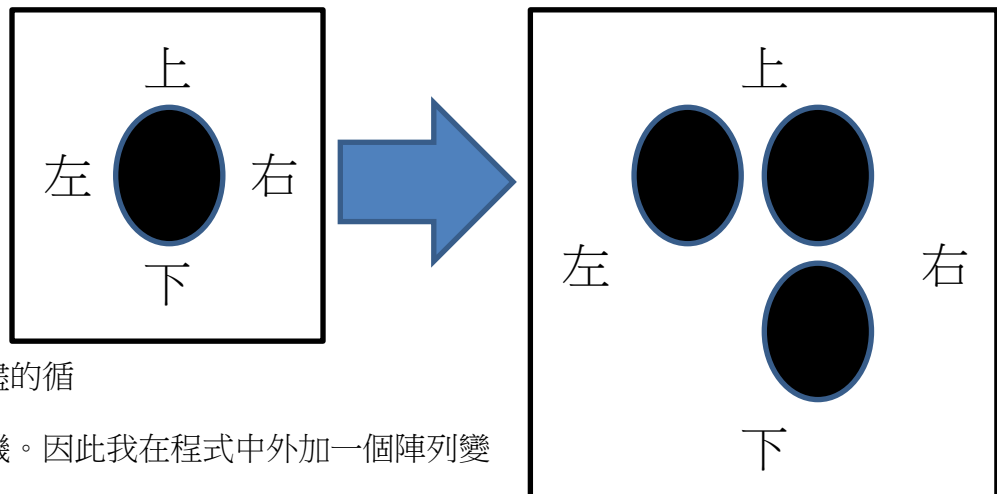


系統設計讓使用者可以用滑鼠來點擊想要的部件色塊，然後告訴電腦要分解什麼部分，電腦就會把點擊的那個色塊取出，要用點擊而不是用選取的方式是因為一個部件通常就是一個色塊，而且如果用選取的話，四周

有可能會同時框到其他部件的角落(如右圖紅圈所示)，之後還得想辦法刪除。至於要從一個點如何找出一個色塊其實並不難，由使用者點擊之後，系統就會用擴展的方式將四周也一樣是黑色點的座標記錄下來，完成後再依序以剛剛找到的四周黑色點為中心，再把周圍一樣是黑色點



的座標都記錄下來，如此一直重複到找不到黑點為止，但是這種方法會出現一個非常大的問題，就是有可能右邊的黑點找到左邊的黑點，左邊的黑點又找回右邊的黑點，這樣會造成無止盡的循環，結果就是當機。因此我在程式中外加一個陣列變數，用來紀錄到底哪些黑點曾經被系統找到過，如果



找到已經紀錄過的點，就會跳過。在執行這個程式的時候，會順便紀錄邊界的座標，以便分割部件的圖形，做法是在一開始先用使用者所點擊點的上、下、左、右四個邊界設定成此部件的邊界，一樣用擴展的方式尋找其他的點，當有新的點出現在邊界以外，就調整部件的邊界，把這個點含括在部件內，這樣就可以計算出此部件最小的邊界。

(四) 挑選末級部件

只要是不能再分解的部件，就會將它歸類為末級部件，但並不是所有看起來能分解的字都可以被系統分解。例如【東】這個字不可以分解出【木】，因為在寫【東】時，移除【日】後，會和平常寫的【木】外型上相差甚遠；同時【東】也不可以分解出【田】，因為在寫【東】字時，字的中心是一筆由上而下連成的，如此就無法分出中間的【田】了，因此，就將【東】直接歸類為末級部件。相反的，將【元】這個字經過系統切割後即可分解出上面的【一】與下面的【兀】兩個末級部件，故將【元】歸類為次級部件。

(五) 相似的部件

分解一些字的時候，同一個部件常常有不同的寫法，像是【熾】裡面的【火】的右下角是點，而【炎】中的【火】的右下角是捺，原先我將這兩個【火】當成相同的部件，哪一種是後儲存就直接使用，為了讓組合出來的字更近似於使用者手寫的字，現在我將它區分成不同的部件；若是相同的寫法而有不同的外觀，則會儲存在同一個圖檔中，使用時由系統依據座標檔內要求的部件長寬比例，再自動尋找和篩選，選擇出比例最相近的來使用。

(六) 壓縮並儲存末級部件圖檔

通常字只有黑色和白色兩種顏色，因此我沒有將末級部件圖檔存成一般的 bmp、jpg、png 或是 gif 檔案格式，而另自創一種檔案格式，裡面每 1 個 byte 存 8 個點，也就是 1 個 bit 存 1 點，我把 1 和 0 分別代表黑和白，如此每個點可節省約 96%(23/(8 bit*3))的儲存空間。

(七) 產生座標檔、並建立座標檔字庫

當要分解一個完整的手寫字(或是次級部件)時，系統會產生並儲存那些末級部件的圖檔，同時並製成座標檔，座標檔的內容會記錄這個字所需要的各種部件代碼，以及這些部件在這個字裡面所存在的位置和大小，例如：

保	代碼	座標	大小
亻	4EBB	(0,0)	(3,10)
呆	5446	(3,0)	(7,10)

如果想要用很少的手寫字來創造出很多個組合字時，就必須要有大量的座標檔。由於不同的人寫同一個字的架構都類似，因此產生的座標檔都差不多，差異較多的反而是末級部件的圖檔，因此我想到用建立座標檔字庫的方式，只要是分解過以前我所寫的手寫字，或是分解過別人的手寫字，甚至於是利用 WINDOWS 系統中內建的字體來分解，都可以將分解後得到的座標檔加入座標檔字庫，如此即可達到快速累積座標檔的目的。

(八) 進行組字

有了座標檔和末級部件圖檔後，便能有效的組字了，一開始先建立一個空白的輸出點陣圖檔，然後依座標檔把指定的末級部件圖檔轉換為點陣圖後並複製到暫存空間，之後將暫存的點陣圖縮放到座標檔內所要求的大小，再把調整大小後的點陣圖形複製到輸出圖檔，同時將輸出圖檔的黑色部分保留，白色則轉換為透明，並一直重複到複製完所有座標檔內所要求的部件圖檔為止。

要將點陣圖白色部分轉換為透明的方法很簡單，我們可以運用一些簡單的電腦基本邏輯運算 or 和 and，

	1	1	0	1	0		1	1	0	1	0
or	0	1	1	0	0	and	0	1	1	0	0
	1	1	1	1	0		0	1	0	0	0

來達成；簡單來說，or 就是把 1 和 0、0 和 1、1 和 1 都

變成 1，只有 0 和 0 變成 0；而 and 卻是相反，它是把 1 和 1 變成 1，其他的狀況都變成 0。

使用 or 和 and 運算便可達到將圖案中白色部分轉換為透明的效果，由於點陣圖裡的顏色是以數字來設定，黑的點是全 0，白的點是全 1，而且輸出圖檔中只有黑跟白，如果要把白色變成透明，那就和背景的颜色數值做 and 運算，白色的 1 就會改成背景的颜色數值，而黑色的 0 則會保留。

(九) 文字使用寬字元

在研究的過程當中，我發現許多部件有類似的狀況，例如【𠄎】、【𠄏】、【𠄐】、【𠄑】、【𠄒】.....等，用 ASCII 字元顯示時很容易當機，非常不好用，於是我使用 UNICODE 字元，也就是寬字元，一個字元佔據 2 個 byte，也就是一整個中文字的寬度，這樣運作的時候也比較容易排列，一個字元一個字元分的清清楚楚，不會有一個字元跨越兩個顯示空間，出現非常困難操作的情況。

ASCII	A	B	我	他	,	,	
UNICODE	A	B	我	他	,	,	

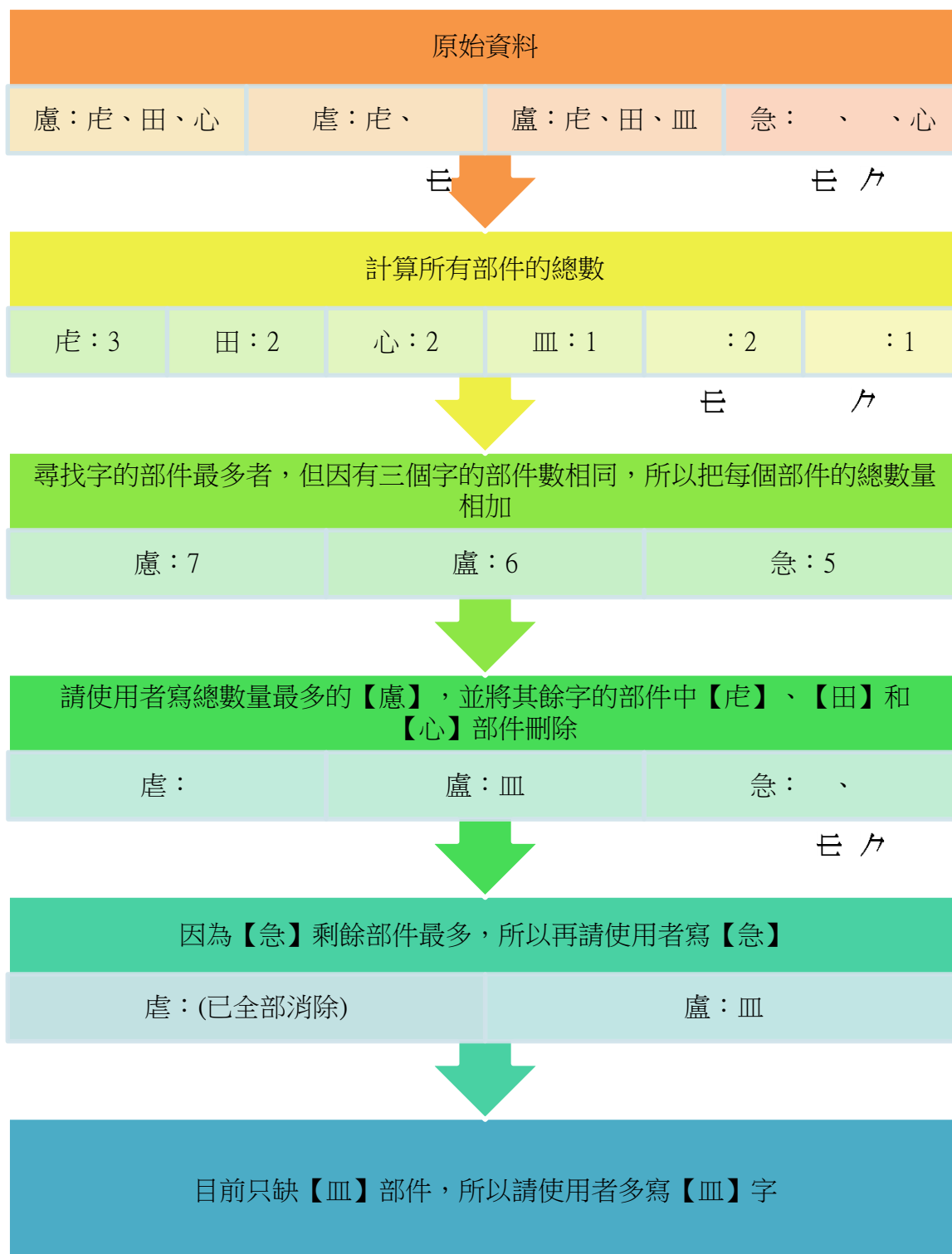
(十) 以最少的手寫字創造出最多的組合字

我使用教育部重編國語辭典修訂本找尋能夠分解出最多部件的字，利用筆畫索引，從筆畫最多的字開始，然後一一尋找，最後得到並不是筆畫多的字部件就會多，像是【麤】這個字的部件量就等於【鹿】的部件量，因為有三個重複的【鹿】部件，因此，要以最少的手寫字創造出最多的組合字，必須找出部件最多的字，例如【灑】這個字，就可以找到 13 個末級與次級部件，這些部件便可以用來組合更多的組合字。

(十一) 以多的字尋找最少的字來拼湊

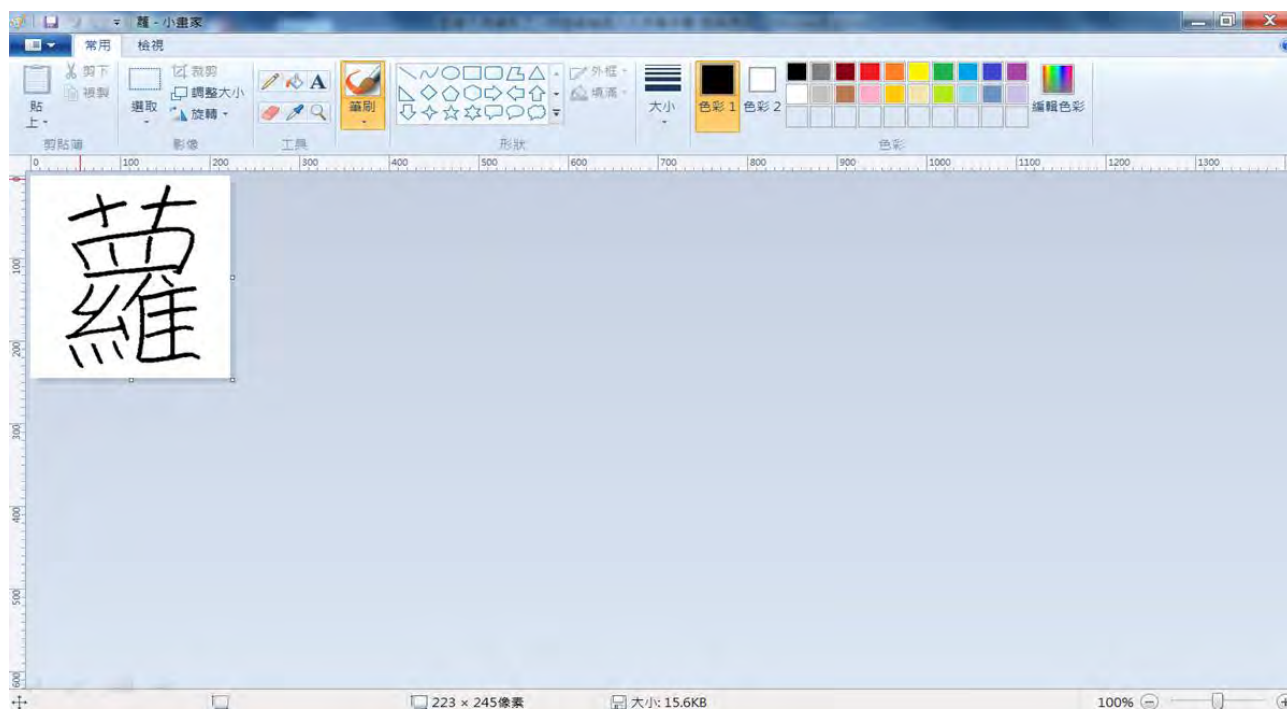
當我想要以系統呈現一些字、一句話或一篇文章時(例如被罰寫)，我想要知道怎樣才能以寫最少的手寫字來產生整篇文章所需要的部件，於是本研究使用以下方法來解決這個問題：先請使用者輸入目標文章，系統會自動分解出文章中每個字的部件並進行統計，同時選出文章中部件數最多的數個字(用 A 表示)，再將 A 所使用到的不同部件數相加後選出其中不同部件總數值最大的一個字(A1)，即為使用者該寫的第一個字；其餘所需的部件，系統會先扣除(A1)的部件，再重新選出文章中所需部件數最多的數個字(用 B 表示)，再將 B 所使用到的不同部

件數相加後選出其中部件總數值最大的一個字(B1)，又可列出下一個使用者該寫的字；重覆以上步驟，直到目標字串所需要的所有部件都產生了為止。此時系統便可以使用者剛剛所寫的字來組合出目標文章。



三、程式說明

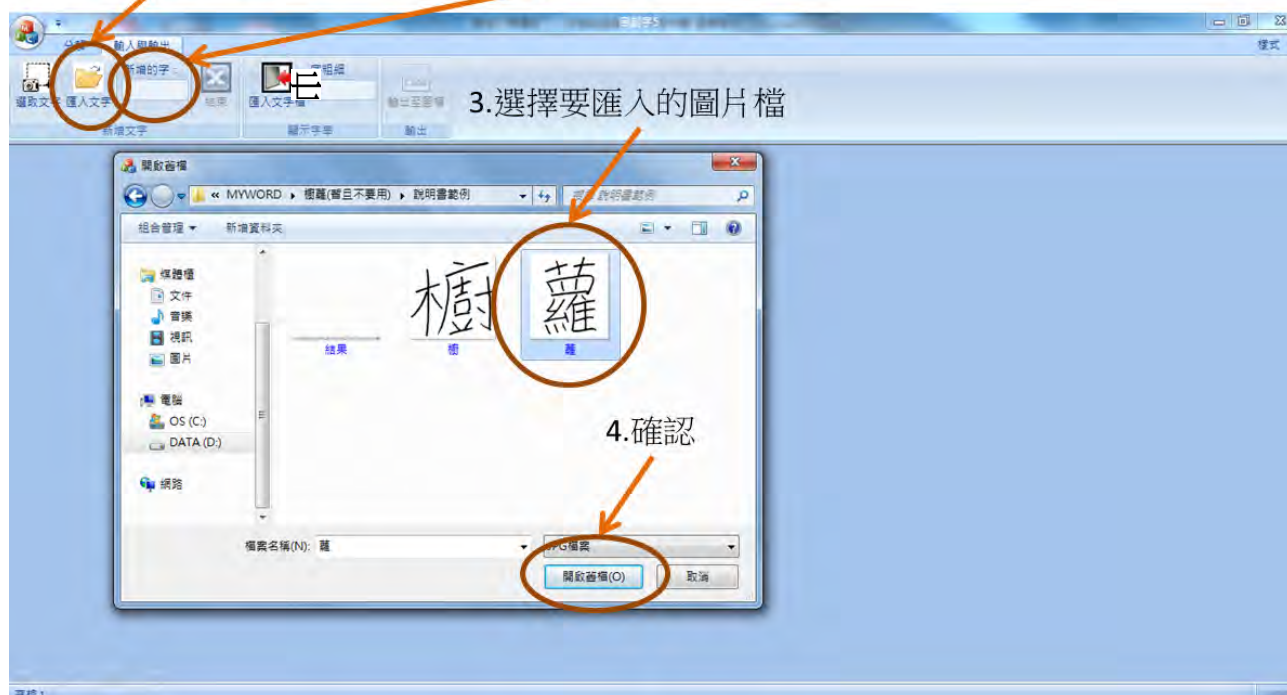
(一) 在手寫板上寫字用小畫家呈現並另存圖檔



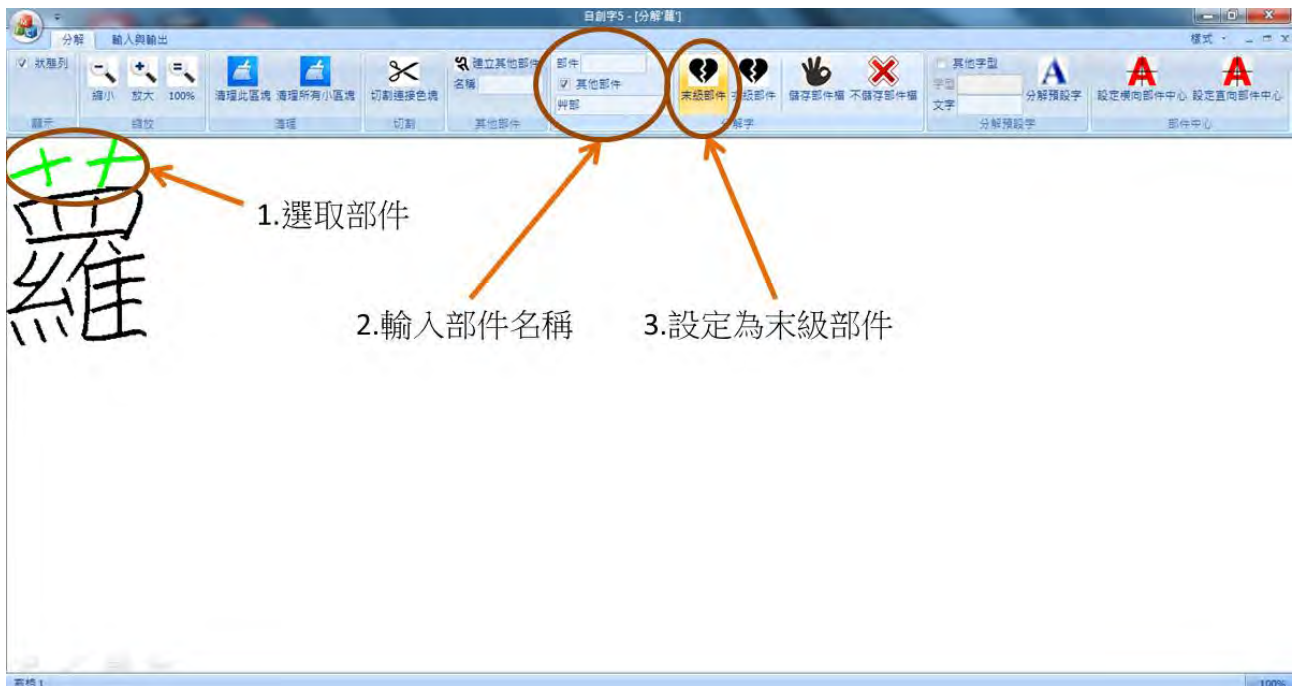
(二) 匯入字【蘿】

2. 點選【匯入文字】

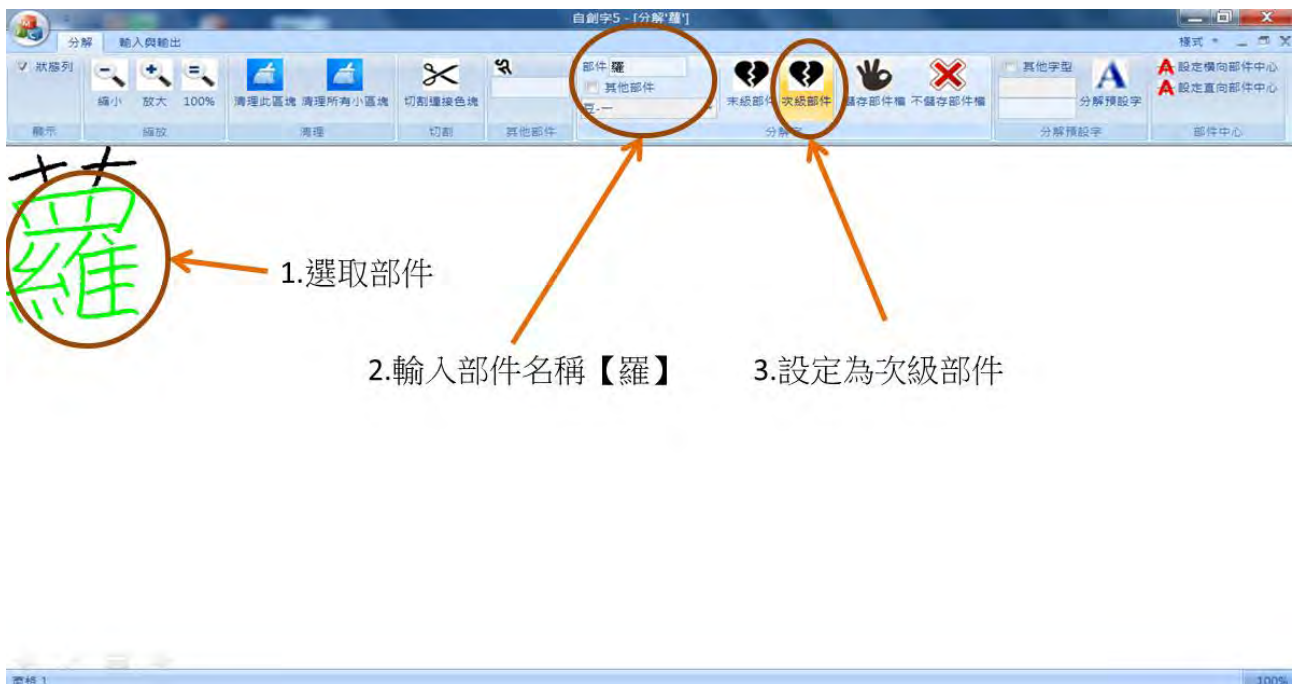
1. 輸入要新增的字



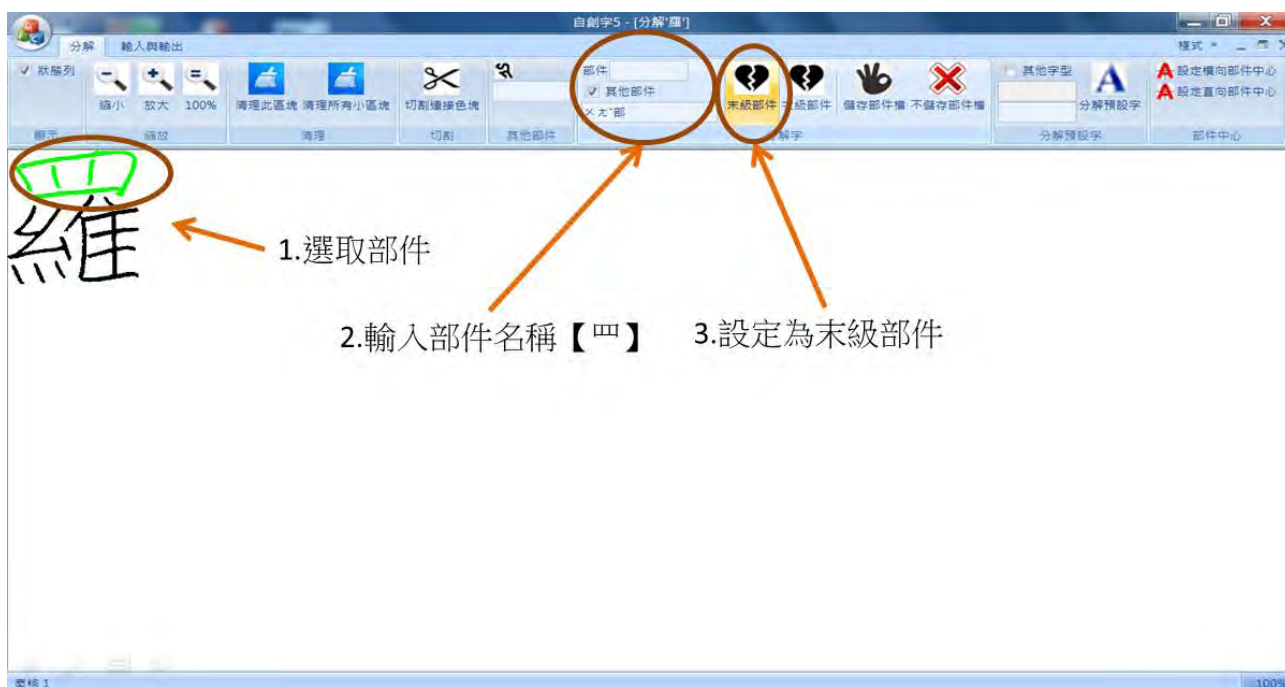
(三) 設定【++】部件為末級部件



(四) 設定【羅】部件為次級部件



(五) 分解【羅】，設定【𠃉】部件為末級部件



(六) 以此類推，選取【維】、【糸】、【隹】為次級部件和【幺】、【亻】、【宀】、【一】為末級部件。

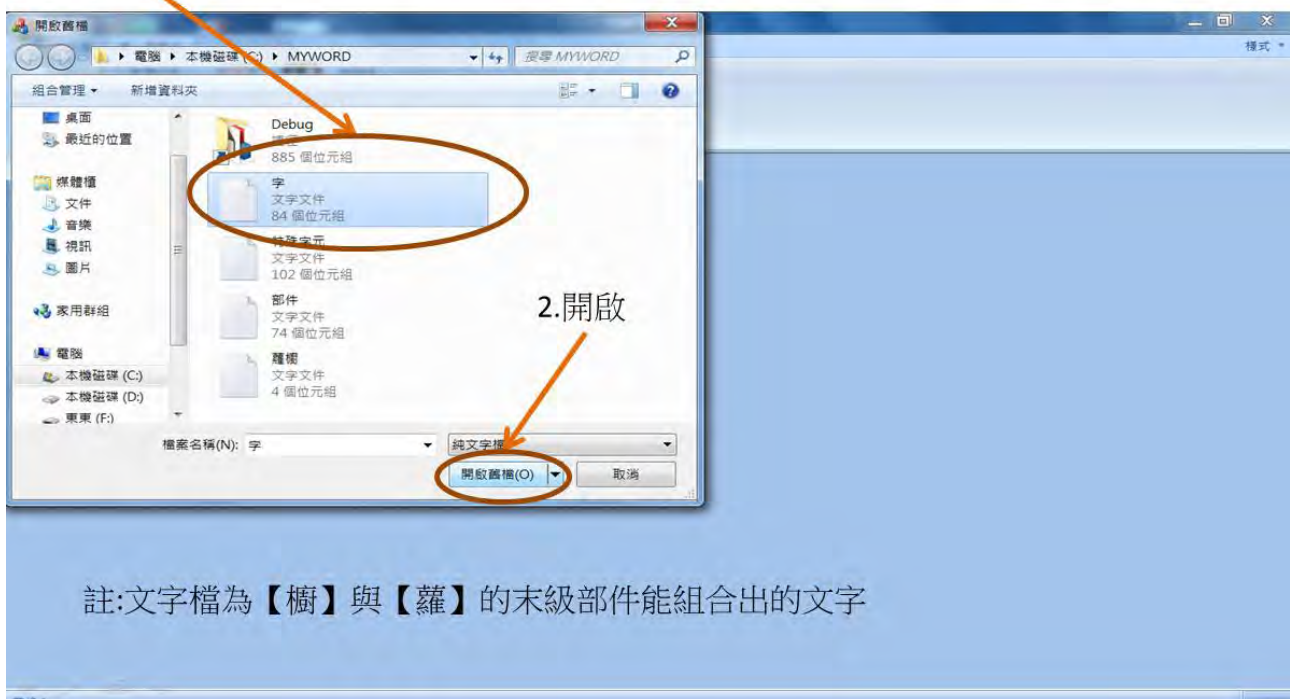
(七) 匯入【櫛】，選取【廚】、【广】為次級部件和【木】、【厂】、【宀】、【土】、【口】、【止】、【寸】為末級部件。

(八) 匯入要輸出的文字檔(如：村)

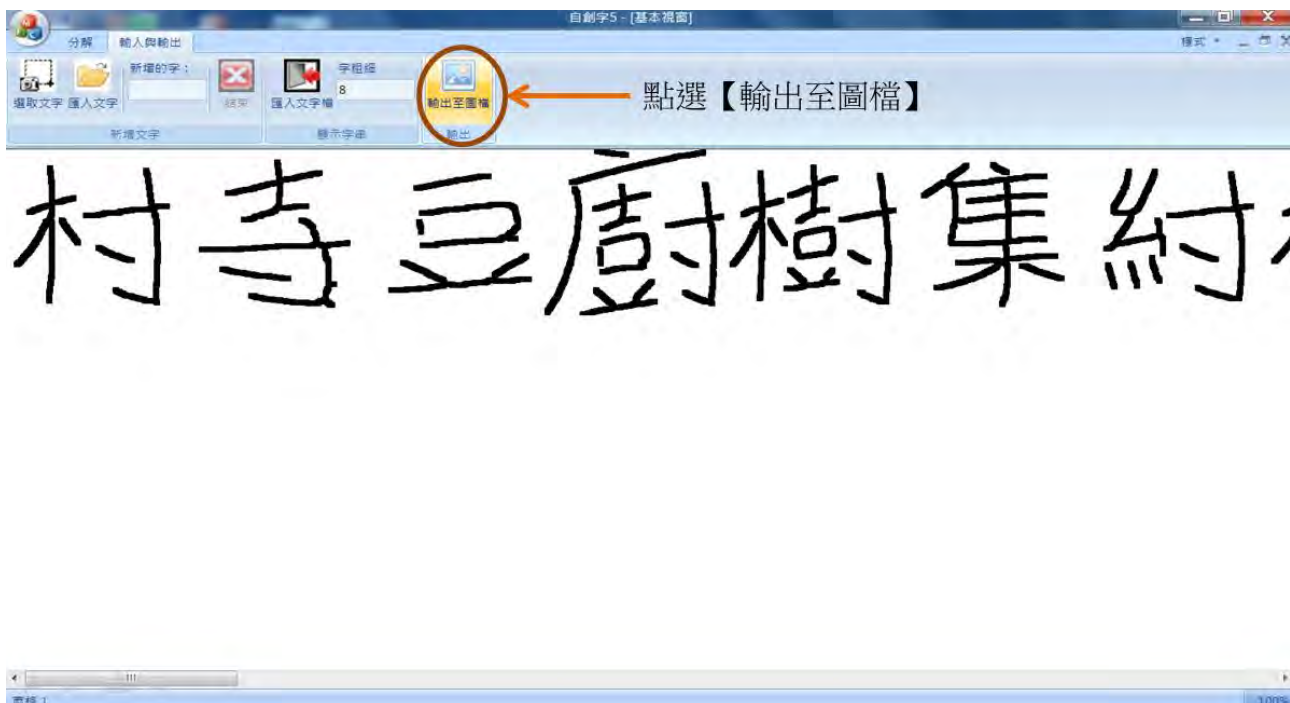


(九) 設定文字檔

1. 選取文字檔

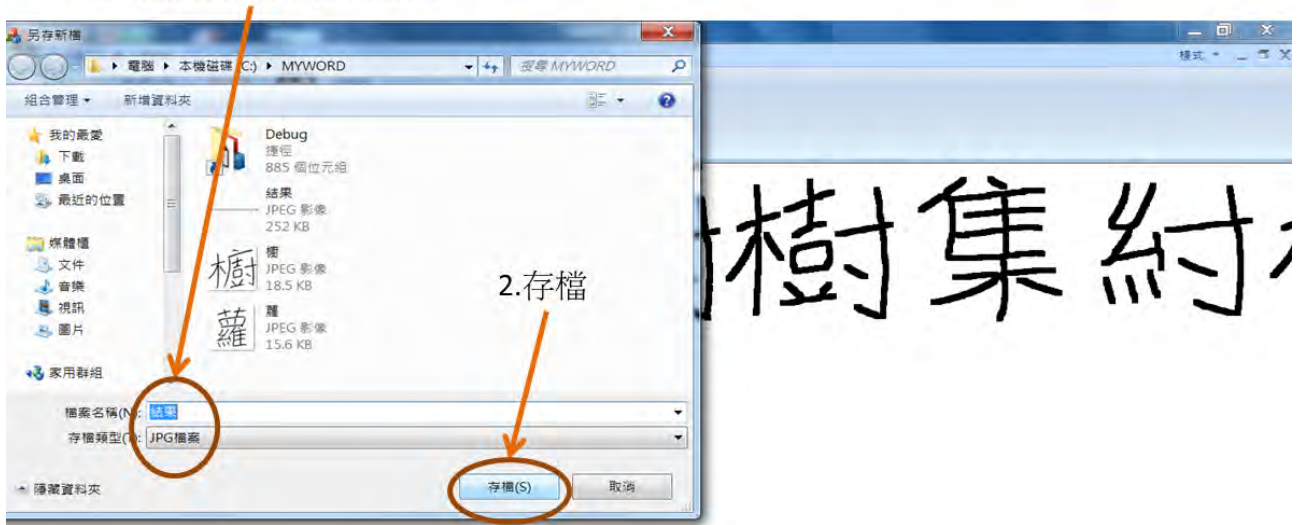


(十) 輸出至圖檔

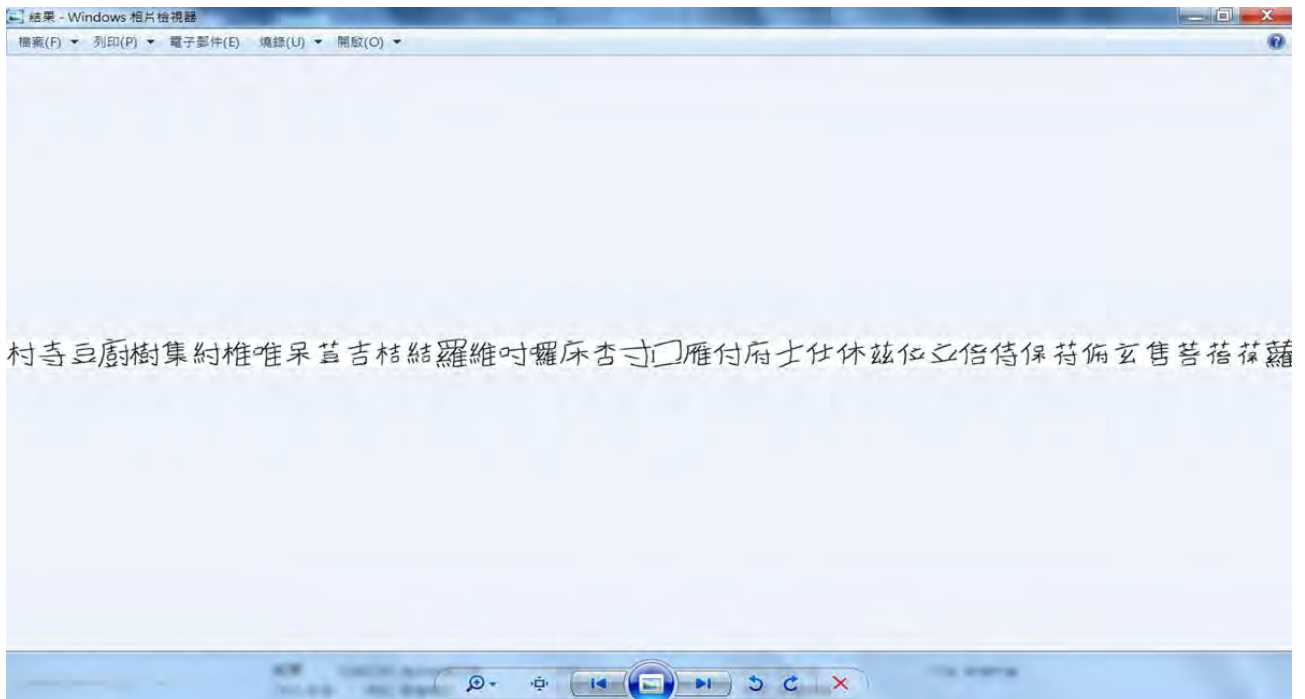


(十一) 設定儲存位置

1. 輸入檔案名稱和檔案種類



(十二) 成功顯示出所有的組合字



陸、研究成果

一、使用者提供兩個手寫字樣本



二、細部的分解與顏色



三、系統可產生的組合字結果

組字結果	實際寫字	組字結果	實際寫字

寺	寺	雁	雁
豆	豆	付	付
廚	廚	府	府
樹	樹	士	士
集	集	仕	仕

紂	紂	休	休
椎	椎	茲	茲
唯	唯	位	位
呆	呆	立	立
萱	萱	倍	倍

吉	吉	侍	侍
桔	桔	保	保
結	結	符	符
羅	羅	俯	俯
維	維	玄	玄

吋	吋	售	售
囉	囉	菩	菩
床	床	蓓	蓓
杏	杏	葆	葆
吋	吋	蘿	蘿

柒、討論與結論

- 一、藉由這次的研究，讓我對電腦文字顯示、圖形處理和程式設計有了更多的了解。雖然期間遇到很多的挫折和失敗，但我幾乎都有想出辦法來解決這些問題。例如：
 - (一) 在用 Visual C++顯示中文字的過程中，發現使用 ASCII 碼(美國國家標準交換碼 - 字的內碼)的話，時常會造成電腦當機，因此改為使用 UNICODE 碼。
 - (二) 剛開始時，相同的末級部件但圖形不同的檔案會互相覆蓋，例如【呆】這個字裡面的【木】和【林】這個字旁邊的【木】，這兩個木明明就長得不一樣卻會互相覆蓋，如此就會造成組合字體的不完美，所以我改為保留全部的末級部件圖檔，然後以末級部件圖檔的長寬比例來篩選出較為接近座標檔內部件長寬比的圖檔來使用。
 - (三) 選取部件的色塊時會造成選取時無止盡的循環，所以我在程式中外加一個陣列變數，用來紀錄到底哪些色塊曾經被系統找到過，以避免重複選取。
 - (四) 圖片若設定只有黑和白，便可以將黑與白分別設定為 1 和 0 來進行儲存，這樣便可以減少非常多的儲存空間，並加快運算效率。
- 二、在這次研究中雖然有達到預定的成果，未來還是希望能繼續朝以下的方向努力研究，讓系統更加完美：
 - (一) 在縮放部件的時候，常常會需要把小的部件變大，由於系統使用點陣圖的方式運作，就容易使字的筆劃產生鋸齒狀的現象，減少相似度，以後想要再思考如何讓放大後的筆畫維持平滑的方法。
 - (二) 研究中發現有時候要把兩個部件組合成字時，這兩個部件的某些筆畫必需連在一起，但是只利用座標檔內的資料無法完美的做到這一點，有時候角度會歪斜或者字體會變形，而使得組合字顯現得不太自然(例如【古】這個字)，以後想要再研究如何克服這個缺點。
 - (三) 這次的研究是以點陣的手寫圖形為運作的基礎，但是現在電腦的字型大部分都是向量字，故也想試試以向量字重新製作系統，看組合的結果能不能增加和手寫字的相似度。

三、雖然這次的研究最初是以簡化罰寫的心態為出發點，但是隨著一遍又一遍的改進程式，讓我從當初的偷懶心態，轉變為挑戰如何讓電腦快速產生具有個人特色的手寫字體，這樣的轉變，讓我學習到要多方嘗試來解決問題，也讓我喜歡上程式設計，更喜歡每次解決語法和邏輯錯誤後，能讓程式順利執行的成就感。

捌、研究計畫執行進度表

		105 年										106 年			
		2 月	3 月	4 月	5 月	6 月	7 月	8 月	9 月	10 月	11 月	12 月	1 月	2 月	3 月
學習 Visual C++		√	√												
第一版 只複製和貼上字	撰寫		√	√	√	√									
	除錯						√	√							
	成功							√							
第二版 可以組裝字	撰寫								√	√	√				
	成功										√				
第三版 開始分解	撰寫										√	√	√		
	成功												√		
第四版 有多個圖檔	撰寫												√	√	
第五版 全部使用寬字元	撰寫													√	√
	成功														√

玖、參考資料

一、參考網站

階層架構圖表 · 取自 <https://msdn.microsoft.com/zh-tw/library/ws8s10w4.aspx>

CImage Class · 取自 <https://msdn.microsoft.com/zh-tw/library/bwea7by5.aspx>

LoadImage 函数与 LoadBitmap 函数的使用 · 取自

<http://thgenius.blog.51cto.com/1042803/1188984>

Cimage 使用概要 · 取自 <http://blog.yam.com/swwuyam/article/15333660>

在刷新窗口时经常要调用重绘函数 · 取自

<http://www.cnblogs.com/X8023Z/archive/2008/12/09/mfc33.html>

MFC 下使用 Ribbon (我的第一个有 Ribbon 的程序!) · 取自

<http://blog.csdn.net/giantchen547792075/article/details/6875276>

簡單的 MFC 多執行緒寫法 · 取自

<http://blog.chinson.idv.tw/2009/03/mfc.html?showComment=1323667950396>

Hands-On Lab Ribbon with MFC-Native · 取自

<http://az12722.vo.msecnd.net/windows7trainingcourse1-0/Labs/Ribbon%20-%20MFC/Lab.pdf>

逐步執行程式碼 · 取自 [https://msdn.microsoft.com/zh-tw/library/aa291565\(v=vs.71\).aspx](https://msdn.microsoft.com/zh-tw/library/aa291565(v=vs.71).aspx)

中華民國第 54 屆中小學科學展覽會 作品說明書 · 取自

<http://activity.ntsec.gov.tw/activity/race-1/54/pdf/040815.pdf>

教育部重編國語辭典修訂本 · 取自 <http://dict.revised.moe.edu.tw/cbdic/>

教育部國語辭典簡篇本 · 取自 <http://dict.concised.moe.edu.tw/jbdic/>

C++資料寫入檔案，或讀取檔案資料 · 取自 <http://jiannrong.blogspot.tw/2015/08/c.html>

[OpenCV]在 Windows 環境下安裝 OpenCV、設定 Visual Studio 及 Python · 取自

<https://archer1609wp.wordpress.com/2015/06/02/opencv%E5%9C%A8windows%E7%92%B0>

<https://archer1609wp.wordpress.com/2015/06/02/opencv%E5%9C%A8windows%E7%92%B0%E5%A2%83%E4%B8%8B%E5%AE%89%E8%A3%9Dopencv%E3%80%81%E8%A8%AD%E5%AE>

<https://archer1609wp.wordpress.com/2015/06/02/opencv%E3%80%81%E8%A8%AD%E5%AE%9Avisual-studio%E5%8F%8Apython2-7/>

(原創) 如何使用 ISO C++讀寫 BMP 圖檔? (C/C++) (Image Processing) · 取自

<http://www.cnblogs.com/oomusou/archive/2007/02/03/639074.html>

為了找工作的 C++課程--作業 5 · 取自 <https://disp.cc/b/11-94>

有關 BMP 圖檔的處理 · 取自

<http://www.programmer-club.com.tw/showsametitle/c/39568.html>

筆畫 · 取自 <https://zh-yue.wikipedia.org/wiki/%E7%AD%86%E7%95%AB>

檔案讀寫 fread()/ fwrite() (C/C++) · 取自

<http://myprogrammingguide.blogspot.tw/2010/07/fread-fwrite-cc.html>

徹底解密 C++寬字符：1、从 char 到 wchar_t · 取自

<http://www.cppblog.com/lf426/archive/2010/06/25/118707.html>

[小知識] 判斷是否為繁體字(Big5) · 取自

<https://dotblogs.com.tw/as15774/2012/10/29/79734>

如何：在各種字串類型之間轉換 · 取自

<https://msdn.microsoft.com/zh-tw/library/ms235631.aspx>

二、參考書籍

韓丹(2005) · 擁抱 MFC：新世代的視窗程式設計（5 版）· 臺北市：儒林

位元文化(2008) · 精通 MFC 視窗程式設計：Visual Studio 2008 版 · 台北市：文魁資訊

普羅席塞(2006) · Windows 程式設計：使用 MFC · 臺北市：文魁資訊

溫植燁(2006) · 使用 MFC 設計 OpenGL · 臺北市：文魁資訊

坦普爾曼(2000) · MFC COM 程式設計教學手冊 · 臺北市：碁峰資訊

林隆煥(2007) · MFC 函式庫程式架構技術手冊 · 臺北市：金禾資訊

普悠瑪數位科技(2004) · Visual C++ 6 範例教本（2 版）· 臺北市：學貫行銷

溫瑞烘、洪正聰、徐玉梅(1993) · 資料結構使用 C 語言（9 版）· 台北市：全賢圖書

王豐緒、許嘉純、范雲波(1988) · Microsoft Quick C 執行時期程式庫參考手冊 · 台北市：

松崗電腦

賀元、賴明宗、劉燈(1994) · 世紀末的軟體革命 · 台北市：傳徵

施威銘研究室(1994) · Borland/Turbo C++語言實務 · 台北市：旗標

蔡明志(1980) · Turbo C 程式語言——入門與應用（6 版）· 台北市：松崗電腦

【評語】 080821

作者挑戰如何讓電腦快速產生具有個人特色的手寫字體，既有趣又創意十足並且獲得初步結果。藉由本研究，作者顯然對電腦文字顯示，圖形處理及程式設計有更深入的認識，假以時日，成果可期。宜再思考字體辨識科學量化的評估標準。

作品海報

摘要

抄寫文章時，時常重複抄寫相同的字、偏旁或部首。因此想藉由這次的研究，把手寫的文字，藉由電腦程式將其拆解成多個部份，再和其他部份重新結合，即可組成更多不同的字。再者，這些字都是由自己手寫出來的，故創造出來的字便如同一組具有個人特色的手寫字體。

壹、研究動機

一天，我被老師罰寫一百遍的「先舉手再發言」，我當時就想，如果可以先寫一遍再利用電腦的「複製」並「貼上」一百遍就交差了事，那該有多好？但是實作後問題百出。因此，我想研究出能簡單創造手寫字體的方法。

貳、研究目的

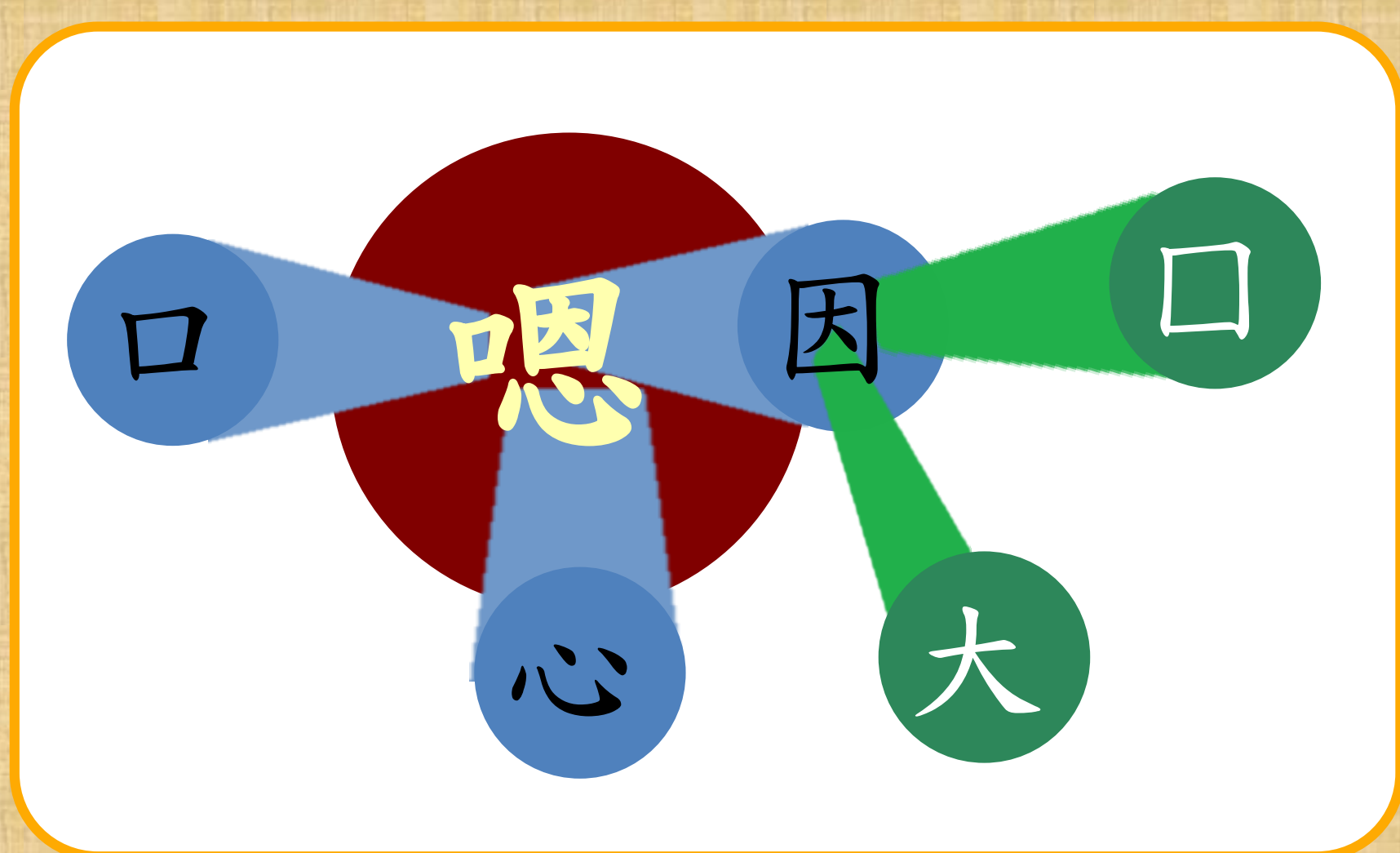
利用手寫板將少量手寫字轉入電腦，再透過系統的拆解與重組，創造出大量個人化的字體，研究目標為：

- 一、讓電腦重組出的字與使用者寫的字相似度提升。
- 二、加快程式運作速度。
- 三、以最少的手寫字來創造出最多的電腦模擬字。

參、名詞解釋

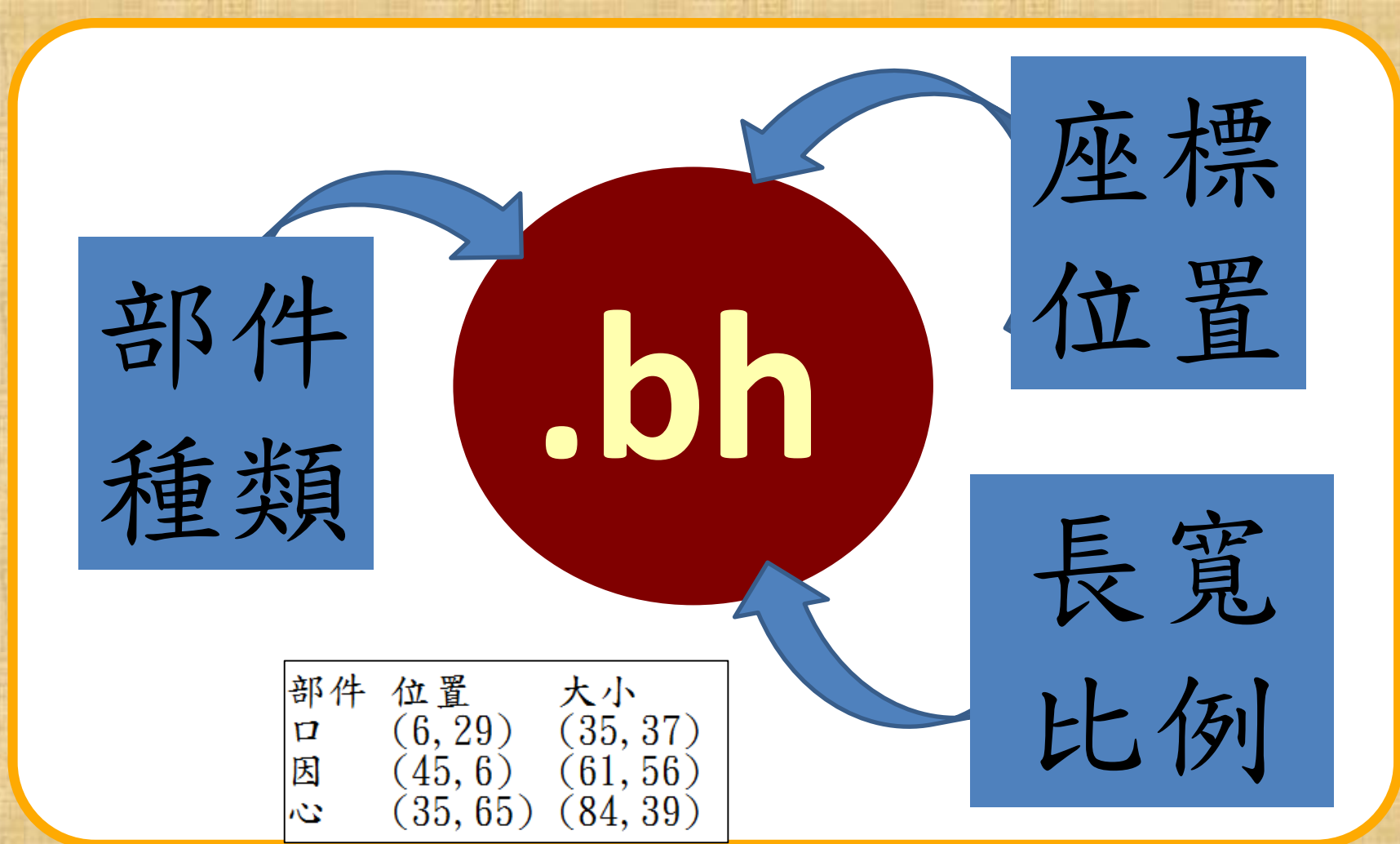
(一) 部件

由一個手寫字拆解出來的部分就稱為部件。



(二) 座標檔

其副檔名為.bh，內容作為組合字的依據。



(三) 次級部件

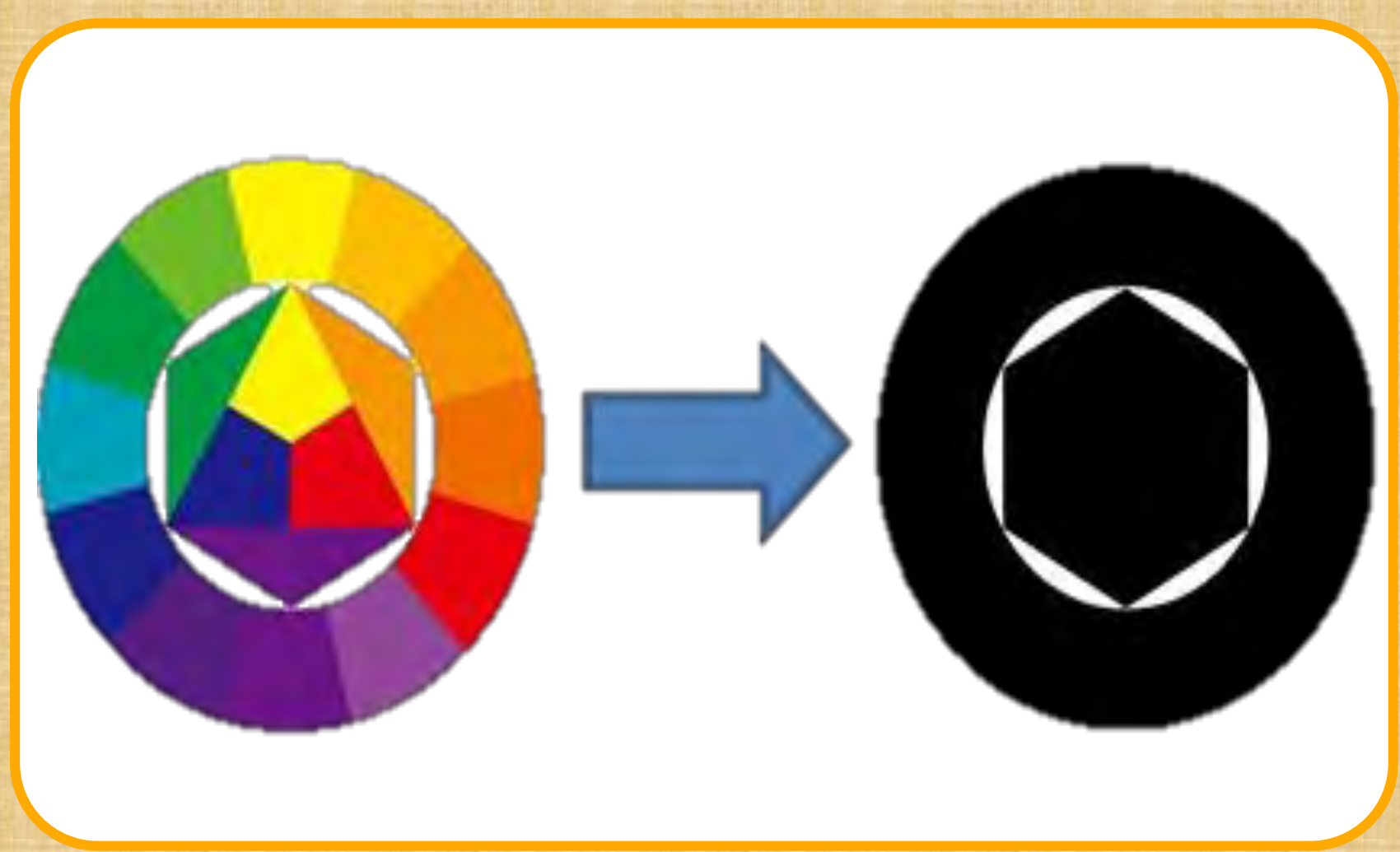
可以再被分解的部件。

(四) 末級部件

無法再被分解的部件。

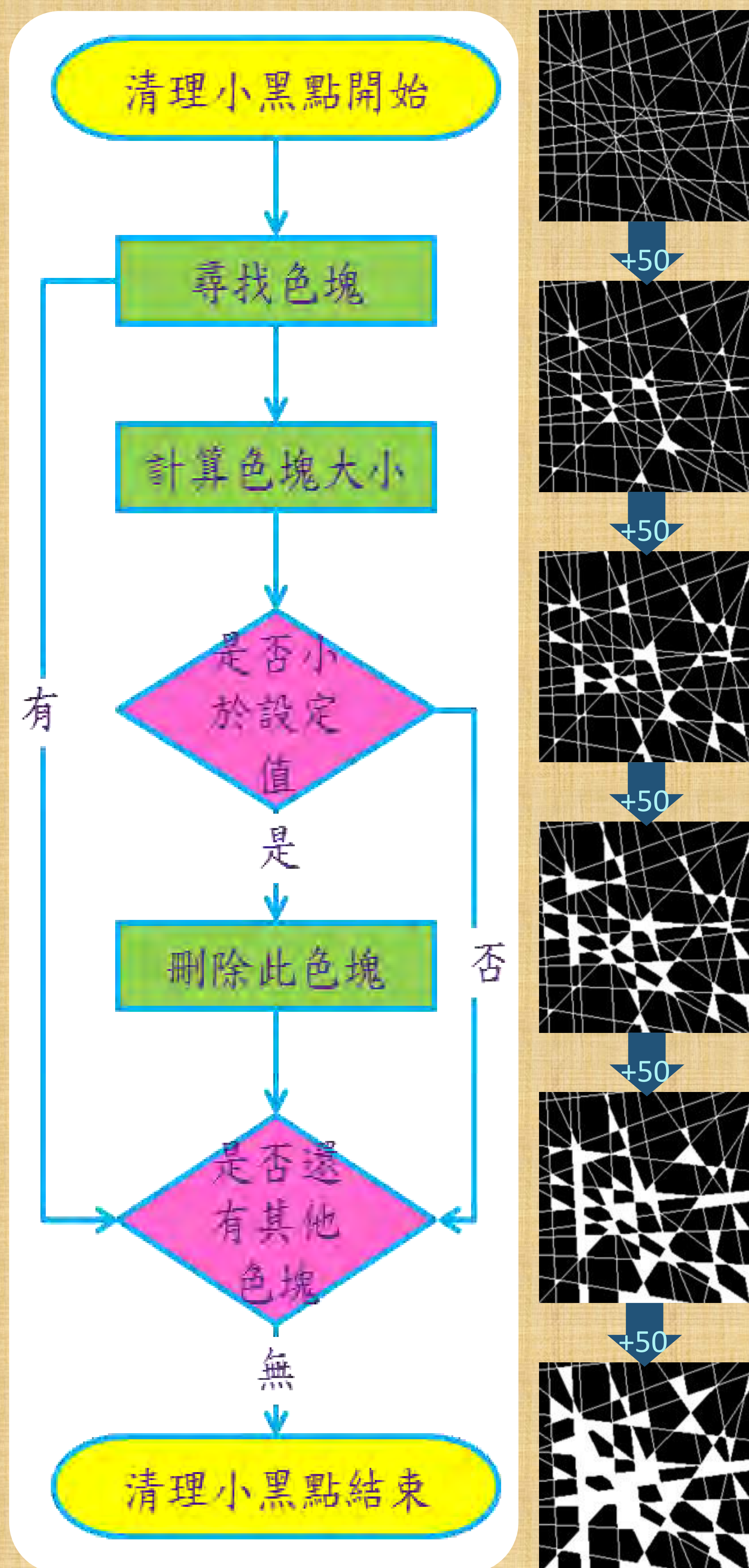
(五) 圖片黑白化

將字圖檔中有色部分轉變為黑色；無色部分轉變為白色。



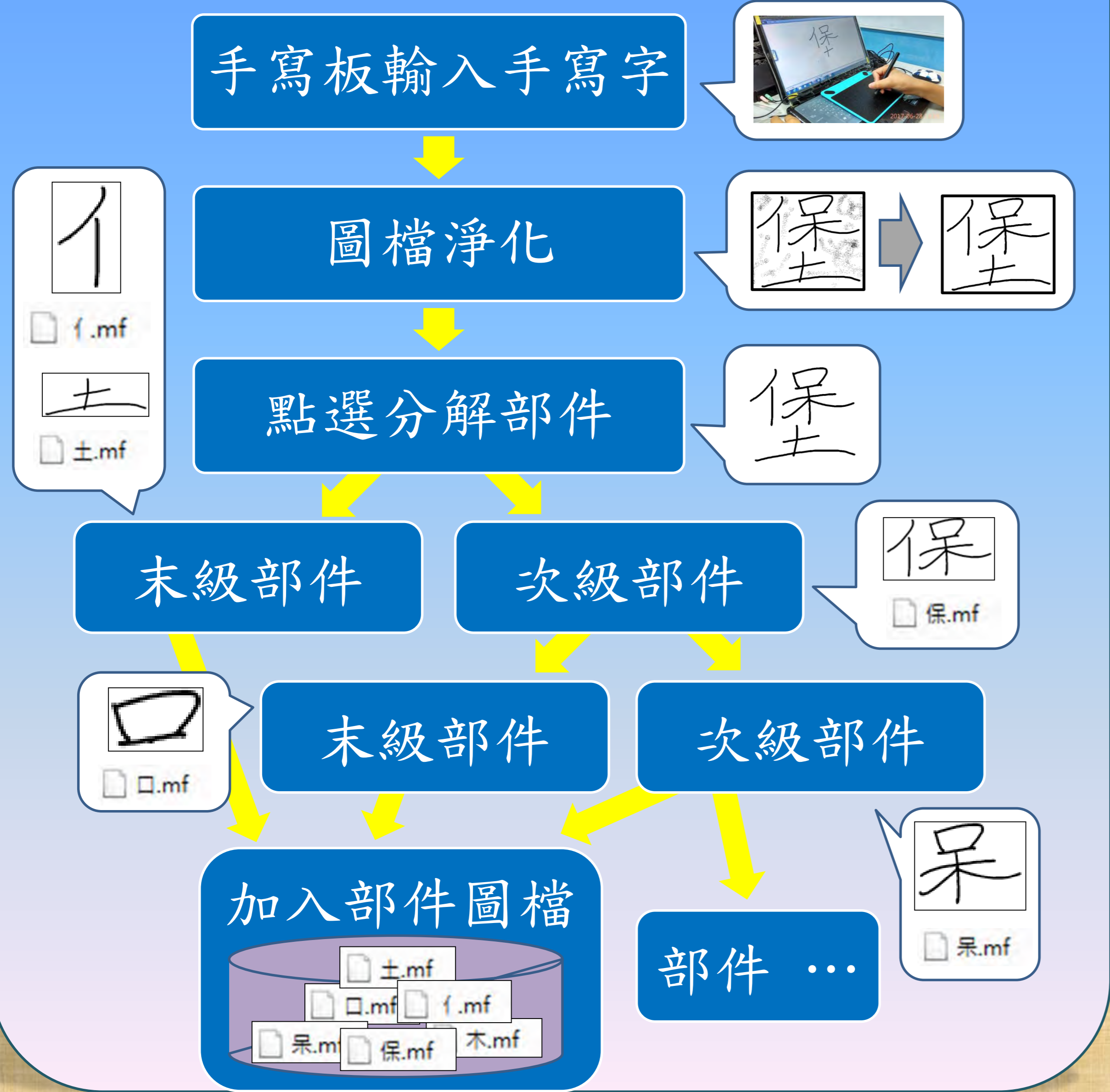
(六) 圖形淨化

運用計算色塊大小的方法，先設定一個數值作為去除黑點的大小基準，只要搜尋圖形中色塊面積小於所設定的數值，就把那一整個色塊用白色代替，這樣就能消除圖形中不必要的色塊或是斑點。

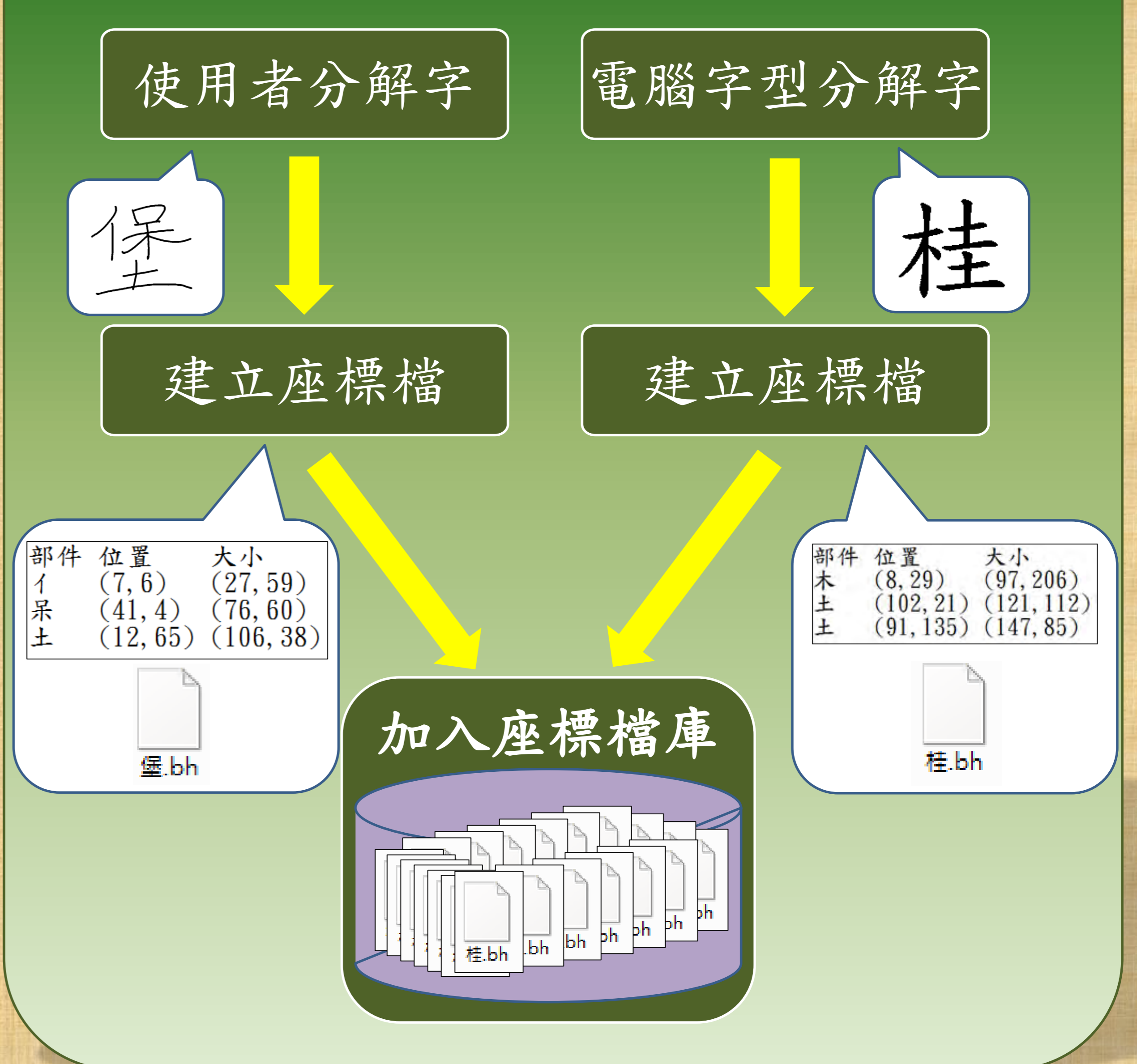


肆、系統流程圖

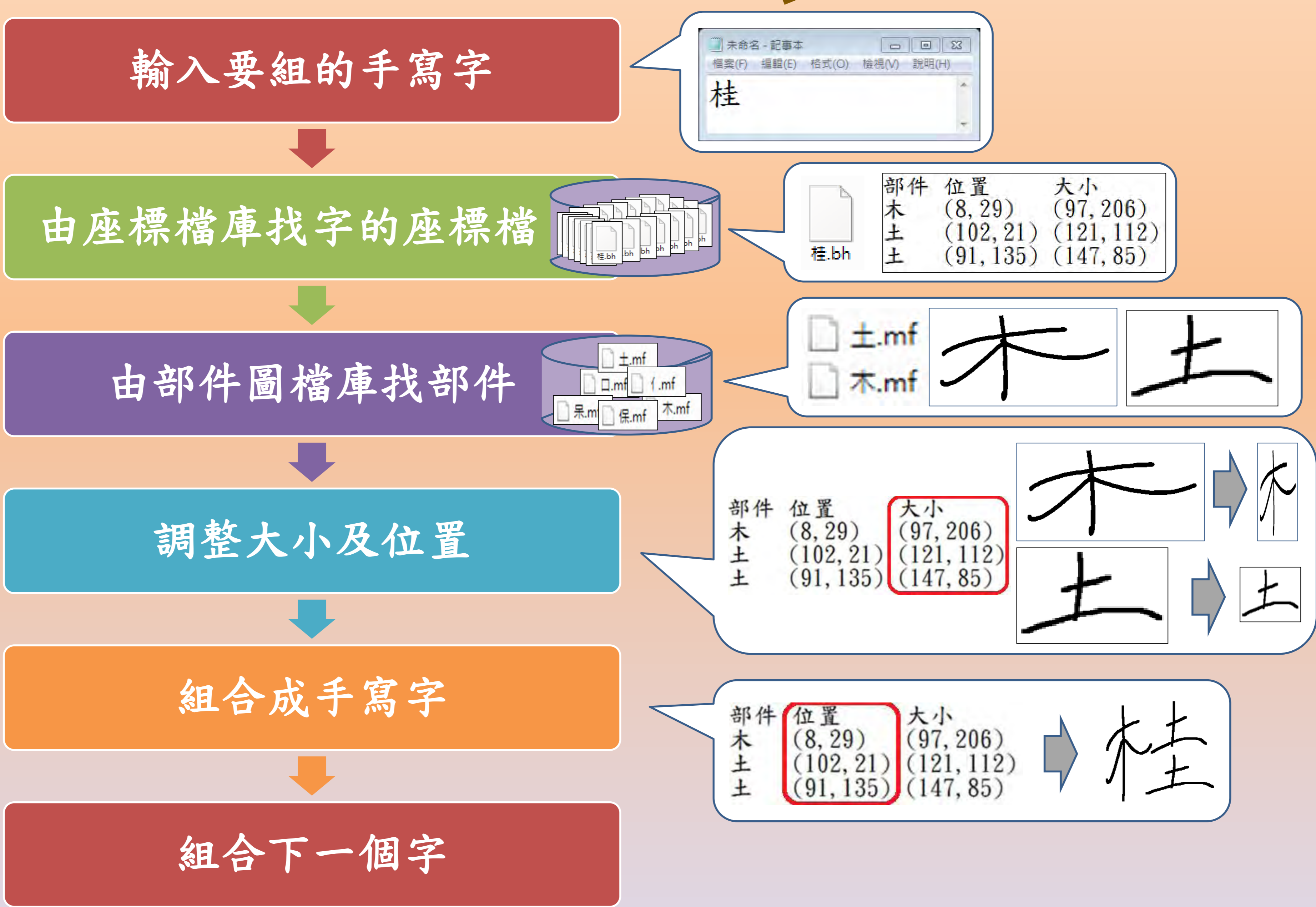
產生部件圖檔庫



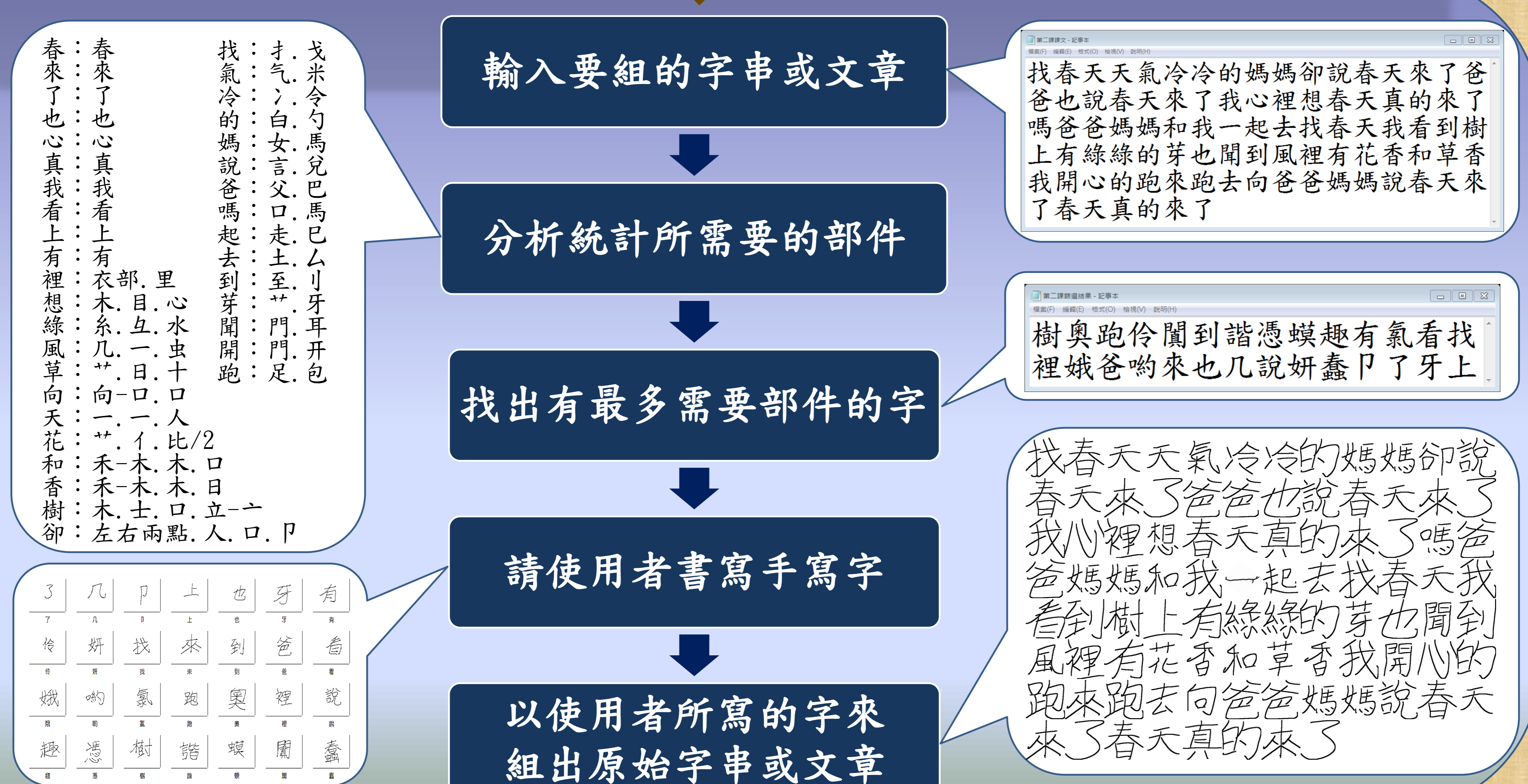
產生座標檔庫



組合手寫字



最少手寫字組合最多字



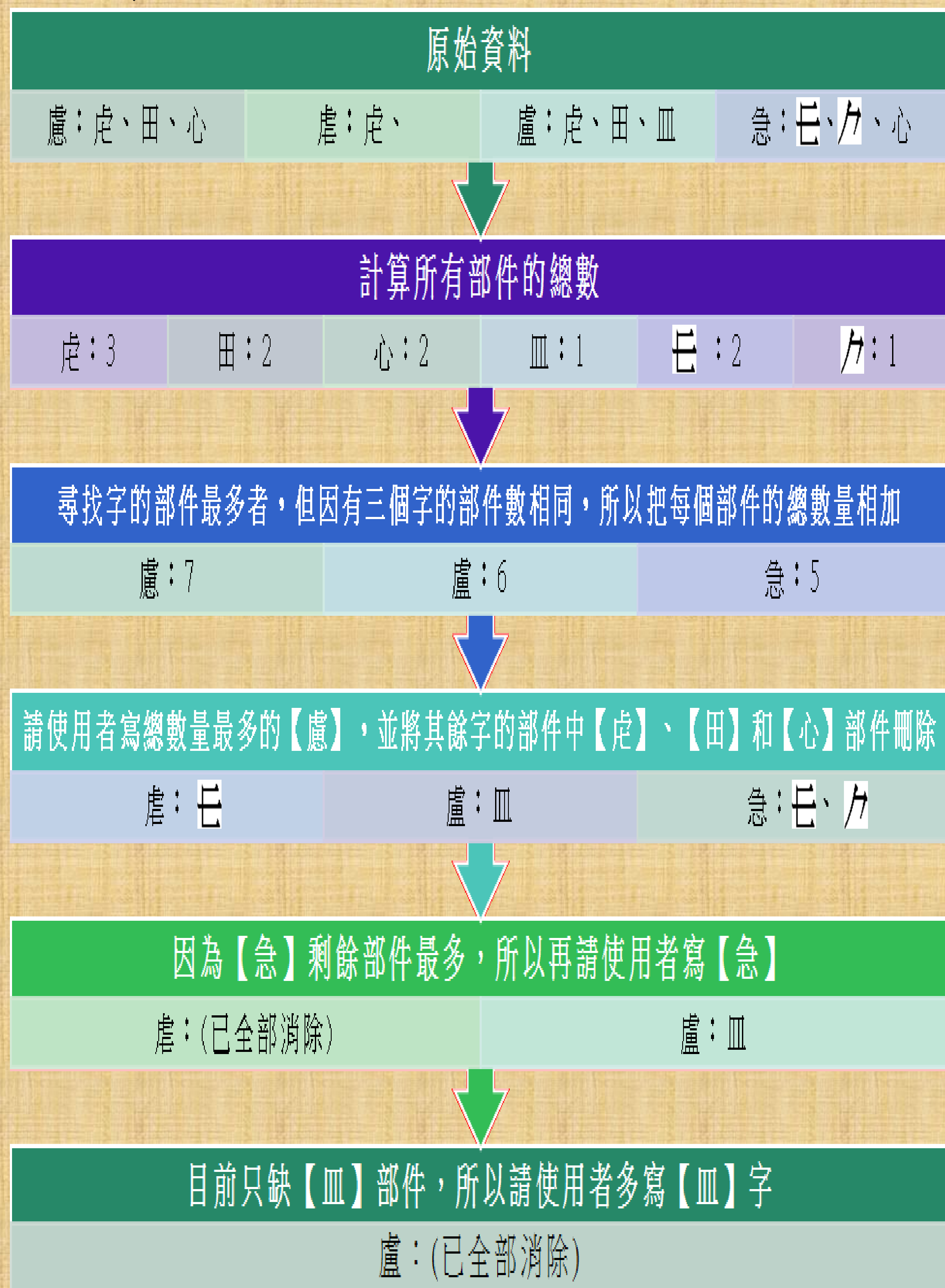
伍、系統說明

(一) 相似的部件

分解一些字的時候，同一個部件常常有**不同的寫法**，像是【熾】裡面的【火】的右下角是**點**，而【炎】中的【火】的右下角是**捺**，使用時由系統依據座標檔內要求的**部件長寬比例**，選擇出比例**最相近**的來使用。

(二) 自動篩選要寫的字

當使用者想要以系統呈現一些字、一句話或一篇文章時，常常不知道需要輸入哪些手寫字才能達成目的，於是使用下圖的方法來找出使用者該寫的字。



陸、討論與結論

- 藉由這次的研究，讓我對電腦**文字顯示**、**圖形處理**和**程式設計**有了更多的了解。雖然期間遇到很多的挫折和失敗，但是都努力的想出辦法來解決。例如相同的末級部件但圖形不同的檔案會互相覆蓋，【呆】這個字裡面的【木】和【林】這個字旁邊的【木】，這樣就會造成組合字體的不完美，所以改為保留全部的末級部件圖檔，然後以末級部件圖檔的長寬比例來篩選出較為接近座標檔內部件長寬比的圖檔來使用。
- 在這次研究中雖然有達到預定的成果，未來還是希望能**繼續朝以下的方向努力研究**，讓系統更加完美：
 - 在縮放部件的時候常常會需要把小的部件變大，就容易使字的筆劃變粗，減少相似度，以後想要再思考如何讓**筆畫粗細能夠維持一致**的方法。

- 研究中發現有時候要把兩個部件組合成字時，這兩個部件的某些**筆畫必需連在一起**，但是只利用座標檔內的資料無法完美的做到這一點，有時候角度會**歪斜**或者字體會**變形**，而使得組合字顯現得不太自然(例如【古】這個字)，以後想要再研究如何克服這個缺點。
- 這次的研究是以點陣的手寫圖形為運作的基礎，但是現在電腦的字型大部分都是向量字，故也想試試以**向量字**重新製作系統，看組合的結果能不能增加和手寫字的**相似度**。

三、雖然這次的研究最初是以簡化寫的心態為出發點，但是隨著一遍又一遍的改進程式，讓我從當初的偷懶心態，轉變為**挑戰如何讓電腦快速產生具有個人特色的手寫字體**，這樣的轉變讓我學習到要多方嘗試來解決問題，也讓我喜歡上程式設計，更喜歡每次解決語法和邏輯錯誤後程式能夠順利執行的成就感。

柒、研究成果

我們利用【蘿】和【櫥】這兩個字來測試，讓系統產生42個組合字，如以下所示：



捌、參考資料

- 教育部重編國語辭典修訂本·取自 <http://dict.revised.moe.edu.tw/cbdic/>
- 教育部國語辭典簡篇本·取自 <http://dict.concised.moe.edu.tw/jbdic/>
- 階層架構圖表·取自 <https://msdn.microsoft.com/zh-tw/library/ws8s10w4.aspx>
- 簡單的 MFC 多執行緒寫法·取自 <http://blog.chinson.idv.tw/2009/03/mfc.html?showComment=1323667950396>
- (原創) 如何使用 ISO C++ 讀寫 BMP 圖檔? (C/C++) (Image Processing)·取自 <http://www.cnblogs.com/oosou/archive/2007/02/03/639074.html>