東　海　大　學

工業設計學系碩士班

碩士論文

中國書法字型模擬系統平台開發之研究

A Study of Chinese Calligraphy Characters Simulation System Platform Development

研 究 生：王雅惠

指導教授：王中行

中 華 民 國 九十九 年 七 月

中文摘要

中國書法字型是中華文化中非常獨特而重要的資產，兼具實用性與藝術性，其豐富性和優美的外觀，更是令人讚嘆。目前電腦系統中的字型雖有相當多的種類可供使用者選用，但是提供給使用者自行變化的彈性較低，例如：無法依個人手寫習慣呈現書法字型風貌、自行造字及複合字的處理與應用...等。本研究以電腦幾何模型為基礎，透過筆劃特徵之辨識與分析，處理手寫輸入中文筆劃，並藉由電腦快速的處理能力，建立一個書法字型模擬平台，作為中國書法字型和使用者之間溝通的橋樑。

研究上，透過電腦書法字型筆劃的分析與處理，建立筆劃特徵模型資料庫，開發一個書法模擬系統平台。使用者可藉由滑鼠或數位筆…等輸入裝置，自由的書寫中文筆劃；系統會自動偵測輸入筆劃之座標值，並進行筆劃特徵辨識，再至筆劃資料庫中讀取所對應的筆劃特徵資料，模擬輸出書法字型樣貌。系統平台中並提供讓使用者將字型模擬結果輸出至 3D 繪圖軟體 SolidWorks 中的功能，以作為後續相關進一步設計與應用。操作上，以親和完善的使用者介面，讓設計師能很容易模擬出書法字型；也藉由此書法字型模擬系統平台的開發，讓一般的使用者或書法的初學者，能更進一步的親近書法，提高書法的學習意願，有利於中國書法的發揚與宣導。

關鍵詞：電腦幾何模型、中國書法、筆劃特徵辨識、筆劃資料庫、字型模擬平台。

Abstract

The Chinese calligraphy is a very unique and important asset in Chinese culture. It enriches and makes people highly praise gracefully. There are quite a lot of kinds in the computer system and are suitable for user's selecting for use at present, but the elasticity of offering the user to change is relatively low, for example unable to get used to presenting the style and features of calligraphies handwritten in accordance with individual. This research is through the fast treatment ability of the computer, set up the simulation platform for calligraphy with the user-friendly interface, let designers can produce calligraphy style of calligraphy easily that offer all field designed to apply to. Furthermore, let general user or beginner of the calligraphy can love Chinese calligraphy more and more and raise the study will, help the developing and declaring and leading of the Chinese calligraphy.

This research is to set up stroke feature model database with the analysis of the style of calligraphy of the computer calligraphy, set up a set of independent calligraphy simulation system platforms, enable users to input the device through the mouse or the digit pad ..etc., the free one writes the strokes of a Chinese character of characters, the computer system detects the coordinate value which examines the introduction stroke and carries on stroke characteristic recognition automatically, and then read corresponding stroke characteristic materials from the stroke database, one kind of outline of style of calligraphy that simulation outputs calligraphies to the Calligraphy characters platform, store the style of calligraphy or export the result in the SolidWorks for the use relevantly.

Keyword: Computer geometric modeling, Chinese calligraphy, Stroke features recognition, Stroke database, Calligraphy characters platform.

誌　　謝

猶記得甫考上東海大學工業設計學系研究所時的歡欣與喜悅，時光荏苒，今日得以獲致豐碩的成果。碩班修業的這一段歷程，是生命中很難忘的記憶。學術殿堂裡人文薈萃的氛圍與沈浸，著實令人著迷；課堂上腦力激盪出智慧的火花亦令人振奮；同儕間惺惺相惜的情感更是令人回味。

論文之完成，首要感謝指導教授王中行教授，悉心的引導學生研究方向，開拓論文寫作的視野，協助學生將青澀的研究構思轉化成厚實的成果；並於論文初稿完成階段，字字推敲琢磨的嚴謹治學態度令學生感佩；在口試準備階段，更是犧牲假日時光，指導學生簡報製作及發表，深切感激在心；感謝口試委員杜瑞澤教授、劉大銘博士、劉文縉博士及蕭肇殷博士，在口試時提供諸多寶貴意見，俾使學生論文品質更臻完備。求學期間，承蒙老師們在課堂上的知識傳授與同學們生活上的關心呵護，謹此致上誠摯的謝意。此外，王組同窗好友惠華、永鴻、仁傑的互勉；學長姐及學弟妹的加油打氣，讓我在學術生涯上未曾感到孤寂，並能更積極的面對人生。

感謝這一路走來，陪伴我的家人、師長、同學、好友及卓中同事們，特別感謝大學學長於程式撰寫上的協助及指導。併感謝王永森主任、劉瑞圓主任、黃國軒校長及陳永昌校長的支持與鼓勵。

感謝所有關心我及幫助我的人，因為有您，生命得以更加美好！

雅惠 謹識

中華民國九十九年七月八日

目　　錄

圖 目 錄

表 目 錄

第一章　緒論

東方設計元素在近代引領整個設計風潮，中國古典風格（Chinese Traditional Style）不但包含了有形的造型元素，還蘊藏無形的文化內涵深度，而在近代設計領域中佔有一席之地。其中，書法更是中華文化的代表之一，其獨特的藝術美，自古以來就被廣泛運用在許多生活領域與設計中，多數經典的書法作品是以書簡、繪畫、篆刻、碑帖、牌匾及室內裝飾物...等各類形式流傳至今。

數位科技的發展，也開始應用於中國書法字型，電腦字型廣泛應用在各設計領域中。雖然目前電腦系統已開發有許多字型供使用者選擇，但仍不免缺乏個性化及變化性，無法完全展現特有字型的豐富性；且某些電腦字型常常面臨罕見字及特殊字的缺字問題，使用者往往須耗費許多的時間進行處理。因此本研究希望開發一套中國書法字型模擬平台，讓使用者可以快速模擬書法字型加以運用。

1-1 研究背景與動機

中國書法的特點在於不論任何人書寫，均可表現個人獨特的個性與風格。傳統的書法創作需使用毛筆、墨汁與宣紙來書寫，有很大的不便性與限制；而且一旦書寫中途發生錯誤，便要全部重新寫過，這對於書法字型的運用有很大的障礙。目前已有一些書畫及書寫學習模擬平台系統，雖然已經可以模擬出虛擬毛筆的書寫特性；但使用者若想要寫一手好字，必須具備書法字型書寫能力或透過長時間的練習，因此若要快速運用書法字型，仍有一定的難度。例如：套用電腦系統中已有的字型，再透過影像軟體加以變形或處理，對於一般的使用者

而言，仍具有其困難和不便性。因此本研究希望開發一套中國書法字型模擬系統平台，讓使用者能透過這個軟體平台快速的模擬特有書法字型特徵，並可將結果輸出至 SolidWorks 中，作為後續相關開發與應用。

1-2 研究目的

目前電腦、電子翻譯機及 PDA 等書寫系統，都是利用辨識使用者書寫的結果（筆劃或是文字），輸出系統中建置好的文字。這樣的系統並無法達到個人化的目的，且多數的辨識系統並無法達到百分之百的辨識率，使用者仍須手動來選出正確的文字。而目前提供個人化字型的系統，多半是將使用者書寫的結果加以輪廓向量化，因此產生的字型並無法像由專家建置出來的那樣美觀。

本研究目的希望透過分析專家建置好的書法字型檔案，找出字型特徵並建構字型特徵模型，建立字型特徵資料庫，開發一套獨立模擬系統平台。使用者可透過此系統平台，自由的輸入中文筆劃，系統依據使用者書寫的情形，自動利用座標運算加以判讀筆劃特徵，並模擬輸出字型。因此相較於目前將書寫輪廓向量化的系統較為美觀；而相對於辨識書寫系統，則是增加了個性化及即時化的優點。

1-3 研究架構流程

本研究的架構流程，主要是分為建立筆劃特徵資料庫及使用者操作介面兩部份，如圖 1-1 所示。由現有字型進行分析，經過字型筆劃特徵擷取、細線化處理及歸納分析，建立筆劃特徵資料庫；藉由設計使用者操作介面，由使用者利用滑鼠或數位筆…等輸入設備，書寫輸入筆劃，系統會自動擷取筆劃進行細線化處理及特徵擷取，再比對所

建立的筆劃資料庫，找出相對應的筆劃進行輸出。系統並提供將模擬結果輸出成 jpg 圖檔格式及可自動輸出至 3D 繪圖軟體 SolidWorks 中的功能，讓模擬字型可再進行後續的處理與運用，發揮更多元的應用層面。系統平台開發所使用的相關軟體，如表 1-1 所示。

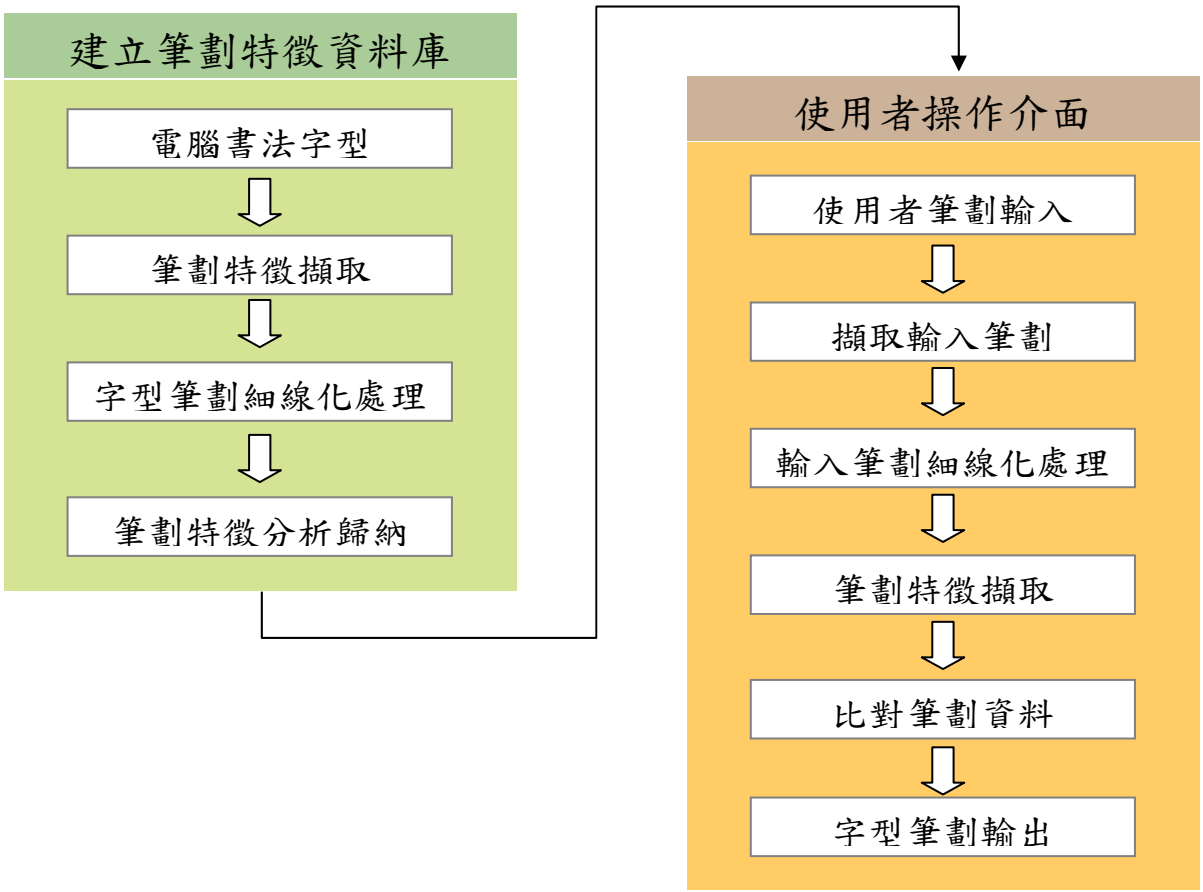


圖 1-1 研究架構流程

表 1-1 模擬平台開發所使用的軟體

項目	軟體名稱
作業系統	Windows XP
程式語言	Borland Delphi 7.0
字型特徵資料庫	Microsoft Access 2003
3D 輸出軟體	SolidWorks 2010 SP0

1-4 全文架構

本文分為五章，各章節內容概述如下：

第一章　緒論

包括研究背景與動機、研究目的及研究的方法與步驟說明。

第二章　文獻探討

包括書法字型筆劃特徵、書法字型分類與描述、字型辨識與特徵擷取、SolidWorks 二次開發及書法字型應用等相關研究領域進行探討。

第三章　研究方法與步驟

依據研究的相關步驟，依序將各階段之進行方式加以說明。第一部份說明透過電腦字型筆劃分析、細線化處理及特徵的擷取與分析並建立筆劃特徵資料庫的方法；第二部份為使用者操作介面的說明，包括筆劃輸入、擷取、細線化處理，以及筆劃特徵的比對、字型筆劃輸出等方面。

第四章　系統平台實例操作驗證

介紹系統平台操作介面配置與操作步驟說明，以實例操作驗證系統平台可用性，並針對系統平台開發時之困難點與問題進行討論。

第五章　結論與建議

針對論文所得結果進行彙整，並提出後續相關研究之方向。

第二章　文獻探討

本章節將就中國書法字型相關研究領域進行探討，以作為系統平台開發時，訂定研究步驟與方法的參考。概略分為下列五個方面：(1) 書法字型筆劃特徵 (2) 電腦字型分類與描述 (3) 字型辨識與特徵擷取 (4) SolidWorks 二次開發 (5) 書法字型應用，茲分述如下：

2-1 書法字型筆劃特徵

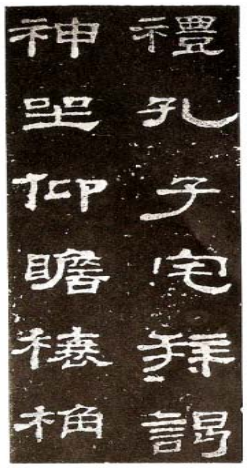
2-1-1 書法字型簡介

書法是漢字的書寫藝術，其歷史源遠流長，從新石器時代的雕刻符號開始，流傳至今五千多年的歷史。從古代開始已發展成一門成熟的藝術，有豐富、完整及一脈相承的理論體系，隨著朝代的演進，書法字型也隨之以不同的形態展現，散發其藝術的魅力。

從甲骨文發明以來，中國書法的字型經歷了由篆書到隸書、草書、行書、楷書等發展階段，每一階段都產生了為數眾多的書法家及書法作品，構成了中國書法的深厚傳統，並影響後人的書法學習和創作。在電腦應用領域中，也根據書法字型特徵發展變化出許多相關應用字型，如表 2-1 所示。其中楷書為唐朝書法主流，一直流行至今，成為最常用的字型，因此楷書也稱為正楷或正體，楷書名家有歐陽詢，顏真卿，柳公權等典範（馮議徹，2007）。

本研究選擇採用了 Microsoft Windows XP 作業系統中內建的「標楷體」字型。主要原因為標楷體的筆劃結構與一般使用者常用的筆劃書寫方式最為接近；且其造字的字元數較為完整，對於建立筆劃資料庫上能有較好的結果。

表 2-1 古代字型碑帖與電腦字型

書法字型	古代字型碑帖	電腦字型
篆書	 齊白石 篆書五言聯	華康新篆體
隸書	 史晨前碑（局部）	華康隸書體W3
草書	 懷素 自敘帖（局部）	金梅毛草書體

行書	 王羲之 蘭亭序	行書 華康行書體
楷書	 歐陽詢 九成宮醴泉銘	楷書 Windows系統標楷體

2-1-2 字型筆劃特徵

書法字型特徵結構基本上是以筆劃所組成，而基本筆劃的名稱約略可分為點、橫、 豎、勾、挑、厥、撇、捺等，組合成一個字。「永字八法」是書法結構的標準，因「永」字具備有側、勒、努、趯、策、掠、啄、磔等八種最基本的筆劃 [湯嘉明，2001]，如圖2-1所示；陳喜榮（1981）解釋「永字八法」中筆劃的基本特色，如表2-2所示。

圖 2-1 永字八法筆劃結構圖

表2-2 永字筆劃的基本特色

筆劃分類	說明
側（點）	指由上側筆而下，似側身飛下之鳥。
勒（橫）	像似書法上的楷則迴鋒，猶如勒馬之用韁。
努（豎或直）	原意為弓彎而用力的樣子。
趯（勾）	如踢球的腳尖。
策（挑或左上撇）	仰橫之意，下筆時如策馬揚鞭狀。
掠（長撇）	如彎彎的犀角，又似秀女梳掠秀髮模樣。
啄（短撇）	掠的一種，下筆時筆鋒著紙後，稍停筆後即是撇，要快且準，有如鳥之啄物。
磔（捺）法	為破裂牲物之意，亦謂之波。若像游魚 穩逸徐徐，為平捺，能上載千鈞，底如船腹，有穩然不斜之態。

對於書法字型的基本結構「筆劃」，諸多學者曾經加以定義，例如：筆劃是指執筆單一運動所作的記號 [Hornby, 1972]；筆劃是由點、線及曲線組成，其中點是指絕對的黑點，線為線性的筆端運動，曲線則是筆端的曲線運動 [Stallings, 1976]；筆劃為執筆而未提起所能畫的線 [Stallings, 1977]；筆劃為執筆從接觸紙面到離開紙面間所產生的圖

樣 [Chung & Tseng, 1992]；筆劃可以用運作點及參數值加以定義，運作點定義筆劃的長度、方向等，而參數值用來定義筆劃的形狀、寬度及位置等 [Wang & Hsu, 1995]；馮議徹（2007）參考西文字型的特徵定義系統 PANOSE SYSTEM 2.0，對於漢字字型所具有的特徵做了定義，將漢字的筆劃特徵定為種類、襯線、重量、樣式、筆劃粗細對比、傾斜角度、工具及比例等八個特徵；蔡登傳（2001）綜合 Stalling（1976）及顧大我（1994）之分類，則將中文筆劃分為 32 類，如圖 2-2 所示；丘永福（2005）對華康特粗明體的筆劃加以詳細定義，如圖 2-3 所示。

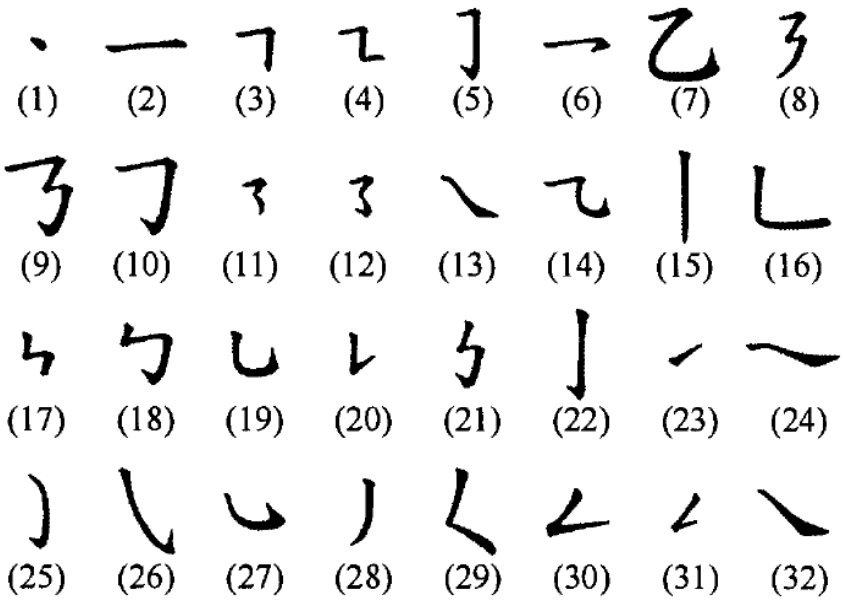


圖 2-2 中文筆劃分類

圖 2-3 華康特粗明體的筆劃

中文書法字型筆劃變化多端，所以造成筆劃描述方法十分複雜。蔡登傳（2001）指出即使是相同的筆劃結構，僅改變其中的一個筆劃形狀即可構成不同的文字，另外筆劃位置、長度及比例等，也都是影響字型結構的重要變項，如表 2-3 所示。

表 2-3 筆劃在不同文字中變化

筆劃變項	實例
不同形狀	
不同位置	
不同長度	
不同比例	

2-2 電腦字型分類與描述

2-2-1 電腦字型分類

數位字型（Digital Type）是指電腦排版系統發明以後，使用於電腦上的字型，由於電腦發展之快速，因此字型種類不斷增加，更有筆劃粗細及長、平、斜等變化的字型。電腦中文字型主要分為兩種，一為點陣字型（Bitmap），一為伸縮字型（Scalable），而伸縮字型依造字技術可分向量字和外框字（或描邊字），其關係如圖 2-4 所示 [呂昭慧，1994]。

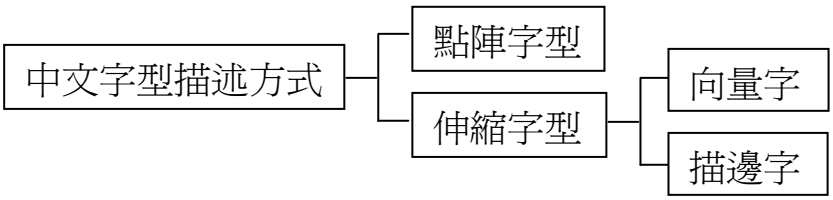


圖 2-4 電腦中文字型分類

呂昭慧（1994）對於電腦中文字型之點陣字型及伸縮字型的特點分述如下：

1. 點陣字型

以 n × n 網格狀所構成，文字佔用與不佔用之網格分別以 0、1 來表示，每個網格代表一個位元（bit）的儲存空間，即採用點矩陣的方式來表現一個字，為早期電腦中文系統的標準配備，如圖 2-5 所示。其優缺點如表 2-4 所示。

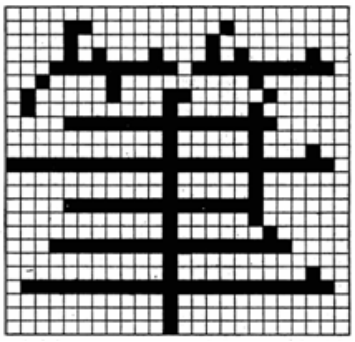


圖 2-5 點陣字型

表 2-4 點陣字型的優缺點

優點	1. 在螢幕上顯現速度快速。 2. 在快速列印草稿上表現突出。
缺點	1. 字型所需之儲存空間太大。 2. 只能以倍數放大，但超過字型所提供之點數大小上限，即有鋸齒狀現象。 3. 旋轉、傾斜、中空、加網等效果不理想。 4. 螢幕所見與列印結果不一致。

點陣字型因有顯現速度快的優點，故現今仍然被使用於螢幕顯示上，以提供較高效率的顯示效果。例如：目前電腦系統中所採用的預設字型大小（8~14pt）等尺寸較小的中文字型。不過，點陣字型主要的功能只作為輔助用途，當使用者設定的字型尺寸大小在電腦系統中並沒有點陣圖像時，字型便會以向量方式顯示；另外，列印文字時，印出字型無論大小亦會使用向量字型列印。常見的純點陣字型有 bdf，pcf，fnt，hbf 等格式，常用於 Linux console，Windows 修復控制台和嵌入式系統中。

2. 伸縮字型

以數學方式記錄字型的變化，當在螢幕顯示或列印時，才根據其大小計算。所以，對於字型可作任意大小的改變，且可節省字型的儲存空間。依造字技術可分為下列二種：

（1）向量字

向量字的特色在其字型的數學描述方式，常採用的為貝茲曲線（Bezier Curve），而最簡單的一種則是使用許多頭尾相連的直線段描述字型，這種字型便稱為向量字。其優點為佔用儲存空間小，儲存速度快。而缺點則是，當描述字型的線段愈多，字型的轉折愈平滑；相對所佔的儲存空間也變大，當任意放大時，向量字並不會產生鋸齒狀，但卻不一定能保持平滑。

（2）外框字

外框字是伸縮字型的一種。它以函數的方式來描述字型的外框，當顯示或列印時才進行邊框的運算，最後將框內空間填滿，可非常精確的描述出字的外形。

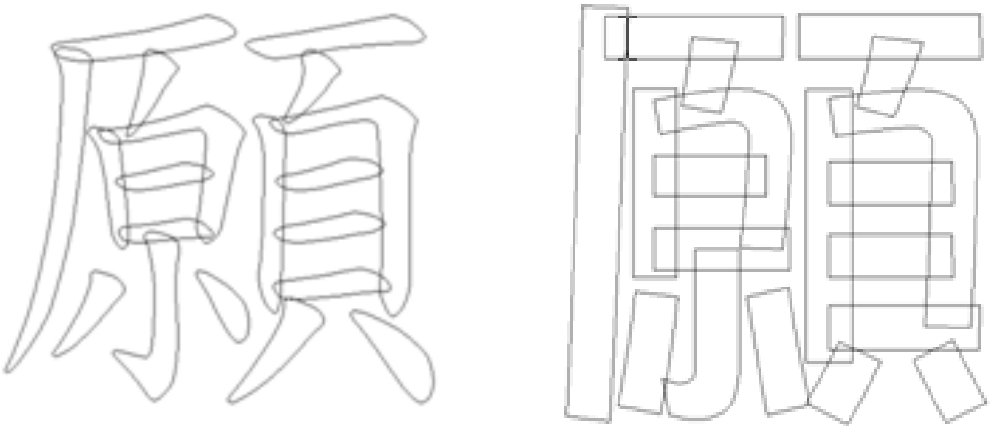
- 優點：運用函數描述，因此曲線永遠是曲線，不會有鋸齒現象、轉折點，字型十分漂亮。

- 缺點：須用相當多的函數才能正確地描述字的外形，而字型愈複雜時，資料量便愈大。

向量字型另一種分類方式，根據造字方式的不同，又可以分成筆劃型及字元型造字兩類，分述如下：

（1）筆劃型造字

筆劃型造字方式是將每一個中文字元分成不同的筆劃，由這些筆劃構成字元。字型的描述方式，是以字型的每一個筆劃為單位分別描述，如圖 2-6 所示，(a)標楷體 (b)華康 POP1 體兩種字型即是屬於筆劃型造字方式。在本研究中以筆劃型造字方式字型較為合適，所採用的字型即為「標楷體」字型。



(a) 標楷體　　(b) 華康 POP1 體

圖 2-6 筆劃型造字方式的字型

（2）字元型造字

字元型造字方式是將書寫好的字元利用程式讀取邊緣曲線資料，所有的字元是由單一筆劃所構成。字型的描述方式，是以字型的每一個字元為單位描述，如圖 2-7 中的 (a)文鼎中圓體 (b)中國龍海報體兩種字型是屬於字元型造字方式。

(a) 文鼎中圓體　　(b) 中國龍海報體

圖 2-7 字元型造字方式的字型

2-2-2 書法字型描述

字型的描述，主要用途是作為字型顯示或列印時，系統畫出實體的顯示文字的依據。中文字型變化多端，因此若要對中文字型進行描述，相較於英文字型而言，困難度相當高且十分複雜。在電腦字型應用上，英數羅馬字型是以用一個位元組（Byte）描述一個字元；而中文字型則需用二到六個位元組來描述。關於書法字型的描述方面，列舉相關的研究如下：

1. Lim 等（1995）先依筆劃元件以圓錐曲線及多項式曲線進行擬合步驟，再配合筆劃的軌跡進行輪廓掃掠，最後將筆劃元件輪廓組成完整的字型輪廓曲線。
2. 邱煥鐘（1998）利用字型輪廓的座標資料，將中文字型加以數值化，並以貝茲曲線進行字型輪廓曲線擬合，以避免 B-Spline 曲線其外凸的特性所可能造成的 CNC 加工空間設定困難的缺點。
3. Wong 等（2000）利用幾何模型理論找出字型書寫的軌跡線，並建立不同的筆刷特性參數模組，再以虛擬筆刷（Virtual Brush）模擬合成書法字型。

4. 盧建智（2000）將書法字型輪廓分割成較小的片段，並以直線及貝茲曲線分別描述，使其和字型片段達到良好的切合度，找出最佳的字型特徵描述。
5. 陳宗輝（2002）利用曲線數學理論方法，將中文字型影像利用過濾法，斜率取點捨棄法等方式將字型轉換成可加工之數值化及曲線平滑化，利用 MASTER CAM 加工軟體進行刀具路徑模擬，最後轉換成 NC 碼，執行實務加工，如雕刻、切割加工等。
6. Sarfraz 等（2002）以貝茲曲線為基礎，利用演算法自動找尋字型輪廓，透過控制點的搜尋及增減，找出最佳擬合曲線。
7. Sarfraz 等（2003）運用演算法將不同的字型特徵加以數位化，如邊緣偵測、稜角尋找及分隔控制點及曲線擬合等，並將結果透過 Client/Server 模組及網路平台上傳輸。

2-2-3 向量字型格式

向量字型在電腦應用下，主要有以下常用的幾種格式，分述如下：

1. PostScript 字型

PostScript 字型是由 Adobe 公司於 1984 所推出，支援 PostScript Type 1 和 PostScript Type 3 兩種不同類型的字型。主要應用於專業排版、影像輸出列印等，並以三次式貝茲曲線描述字型。在 Windows 作業系統中，PostScript Type 1 字型資料以下列兩個檔案儲存，但 Windows 系統並不支援 PostScript Type 3 字型。

（1）向量外框資料放於一個副檔名是 pfb 的檔案中。

（2）字型度量資料放於在一個副檔名是 pfm 的檔案中。

2. TrueType 字型

TrueType 字型是由蘋果（Apple）電腦公司開發的字型技術。是

屬於向量外框字型，主要是利用貝茲曲線描述字型，並可內置點陣字型，使用上十分普及，相容在一般主要的作業系統上，如 Windows、Mac OS 及 Unix 系統中，字型的副檔名為.ttf。在研究上所使用的 Windows 系統內建「標楷體」即是屬於 TrueType 字型的一種。

3. OpenType 字型

OpenType 是由 Microsoft 公司與 Adobe 公司所共同開發，其規格於 1997 年發表，並於 2000 年推出首套 OpenType 字型。OpenType 字型相似於 TrueType 字型，但可同時包含 TrueType 和 PostScript 兩種格式字型資料，對於異體字切換功能強大，OpenType 字型的副檔名為.otf。

2-2-4 造字程式現況

目前電腦系統中，因中文字字數繁多且多變，有些非經常使用的字元或特殊字元，並未納入電腦字型檔內定的字元中，因此必須透過造字方式來解決。造字方式通常都是透過手動或掃描的形式，把造字的點陣圖存入電腦內。1980 年代後期到1990 年代初期造字型式，都是採取點陣造字方式，通常是由書法家或系統編程員本身在一張大型的方格字上寫字（如圖 2-8 所示），接著再把表格數位化加以編碼儲存進造字檔中。

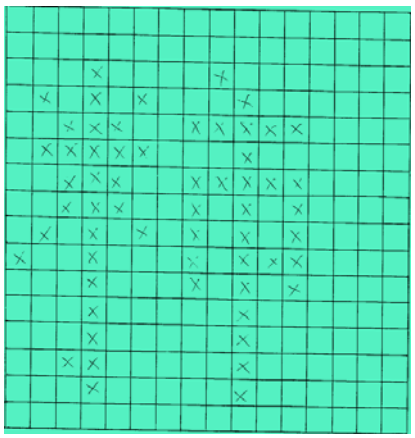


圖 2-8 早期造字用的表格

Windows 2000 時期開始，造字系統開始支援外框字型的造字，系統內建的造字程式，主要是透過點陣造字介面造出來的點陣字，透過描邊的演算法轉換成為外框字型，如圖 2-9 所示為 Windows XP 系統內建的 TrueType 造字程式畫面；另外向量造字的造字程式也開始發展，如圖 2-10、圖 2-11 所示，分別為文鼎圖王造字系統和全真造字精靈的畫面。

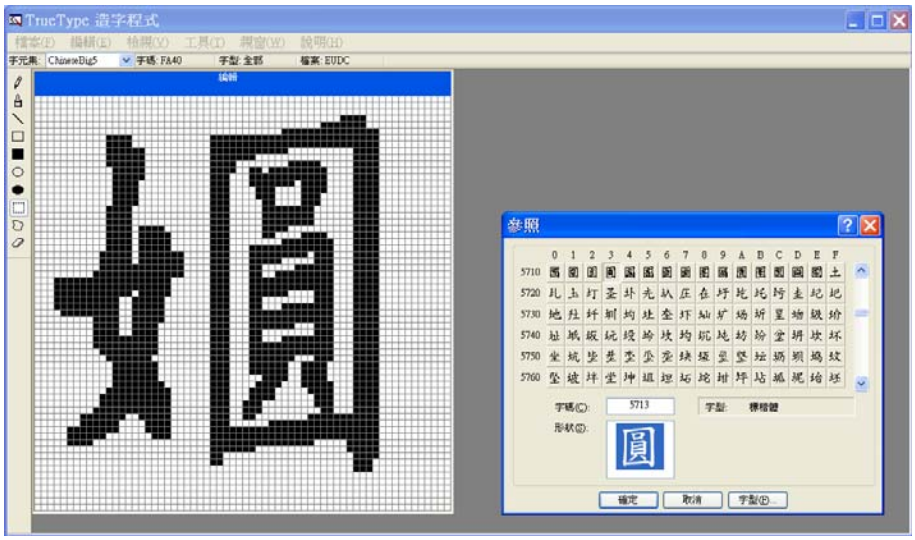


圖 2-9 Windows XP 系統內建 TrueType 造字程式

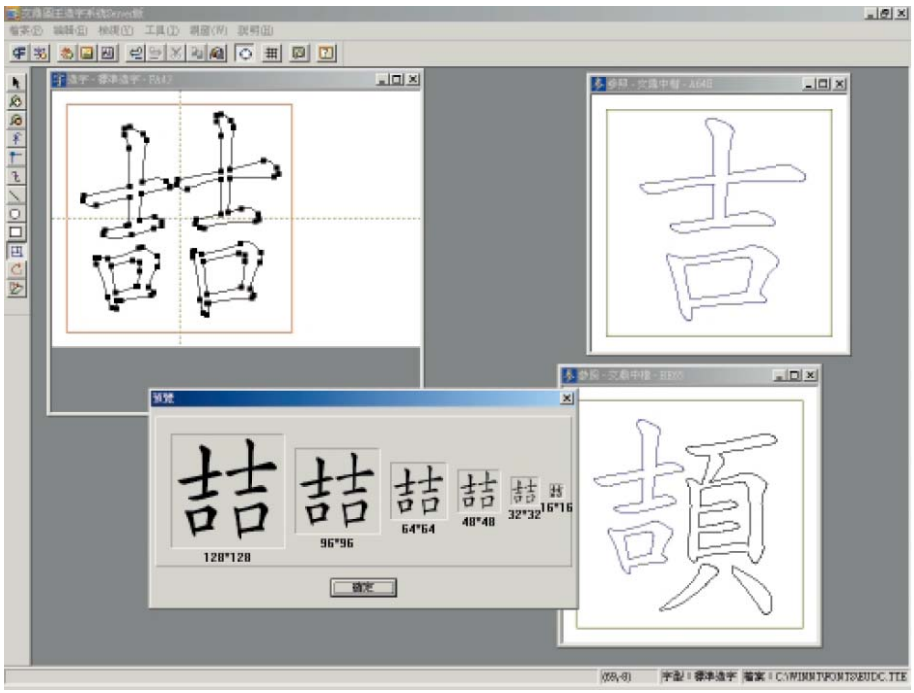


圖 2-10 文鼎圖王造字系統

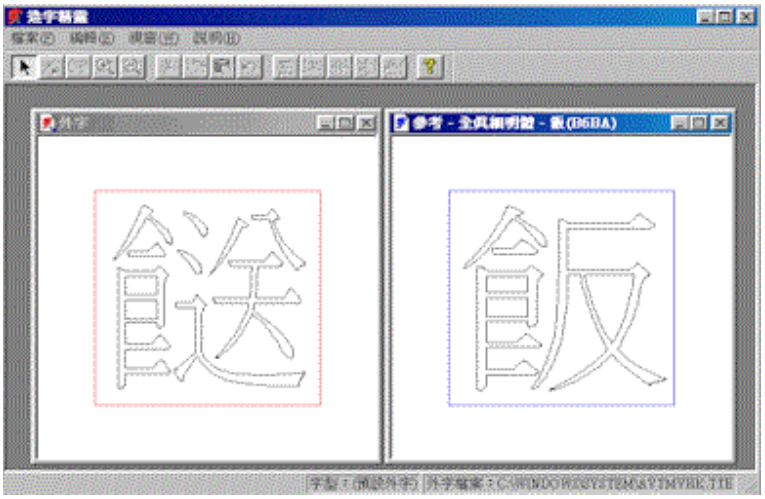


圖 2-11 全真造字精靈

2-3 字型辨識與特徵擷取

2-3-1 書法字型辨識

書法字型之辨識，或稱為逆向辨識，其目的在於以有效的方法取得字型的筆劃特徵、輪廓曲線或是點群相關資料，作為字型相關用途。關於書法字型的辨識方面，列舉相關研究如下：

1. Chuang 等（1995）以啟發式演算法來辨識印刷字型，並利用字型的輪廓特性來提高辨識率。
2. Lin 等（1996）利用字型趨勢導引（Trend-Followed）轉譯技術找尋字型的輪廓片段及整體特性資訊。
3. Romero 等（1997）以類神經網路理論來訓練相近字型的辨識率，並利用不同的搜尋方向來修正類神經參數以增進辨識速度的提昇。
4. 王舜正（2000）針對書法影像做二值化處理，將二值化的結果應用於書法輪廓描述，利用非等方向性擴散的技巧來模糊原始的書法影像，使得字型輪廓得以保留而不被模糊處理所影響，最後利用字元寬度的資訊來去除雜訊。

5. 陳映舟（2001）建構書法影像最短距離的對應圖以紀錄影像中每一個點的最短路徑，利用限制切割路徑與中文草書書法字的特性除掉去多餘的文字切割路徑，以找出文字切割模組；並利用不同的統計式文字特徵擷取及計算特徵距離的方法，找出實驗影像有最高辨識率的組合。
6. Lin 等（2001）利用自動門檻值設定辨識字型並建立常用字元資料庫，以用資料庫搜尋方式來辨識文件中字型，偵測及辨識常用字元並可區分文件中的手寫字及印刷字。
7. 方建中（2001）利用結構式特徵與多層分類法架構做光學字元辨識。在擷取結構式特徵方面，先進行細線化處理獲得骨架，從中擷取子筆劃，再從子筆劃中抽出可代表直線和曲線的性質者做為特徵。在兩層式分類架構裡，先使用二元樹分類法去降低系統處理時間，再利用鬆弛比對法達到較高的辨識率且結構式特徵可容忍多種不同的字型。

2-3-2 特徵辨識方法

Ivind 等（1996）提到在特徵擷取上，特徵可分為統計式特徵（Statistic Feature）及結構式特徵（Structure Feature）兩種。針對不同的特徵，在辨識時會使用不同的方式。統計式特徵常用分類法做辨識，一般稱為統計式處理；結構式特徵則常用比對法做辨識，稱為結構式處理。此兩種方式，亦可用於離線中文字辨認的研究 [吳偉賢，2000；方建中，2001]。在圖樣辨識（Pattern Recognition）領域中，辨識的方法主要分為下列三大類：

1. 統計法（Statistical Method）

統計法為利用機率密度函數（Density Function）將圖樣空間

（Pattern Space）分割成許多區域，每一個區域即代表某個分類，最具代表性的統計法為 Bayes 分類器 [Pandya & Macy, 1996]。莊志鴻（1999）採用統計法作為車牌字元的辨識方法，其所利用的統計辨識方法為 WMMS（Weighted Mask Matching System），WMMS 分為兩個部份，第一部份是訓練模式，第二部份則是辨識模式。在訓練模式時，以機率統計的方法計算參考圖樣每一點的出現機率，並藉此完成參考圖樣的訓練與建立；當參考圖樣建立之後，以圖樣比對的方法找出待辨識字元的答案。

統計法著重對文字圖形的分析，使用圖樣一些特性的統計分佈做為特徵，多由一串有次序、長度固定的數值來表示，稱為特徵向量。主要依賴統計決定理論，提供一個分群演算法去切割特徵向量空間來辨識字元，其缺點是很難從 n 維的向量空間看出圖樣的特性。另外，如何去導出符合每一個字元的機率分佈函數，是十分困難的工作 [吳偉賢，2000；方建中，2001]。

2. 結構法（Structural Method）

結構法是利用字元結構特徵作為辨識參考依據，常見之結構特徵包括：水平線段、垂直線段、封閉區間數量等。結構法之優點在於可將字元結構和各特徵之間的關係，加以公式化處理。公式化處理可視為某種編碼動作，將各個特徵編碼前，需建立一個解碼的規則。編碼過程必須保證同一字元在不同情況下，例如：有雜訊、位移及旋轉等，所得到的編碼是一致的，而解碼的過程也必須保證不同的編碼不會造成同樣的解碼結果；而同樣的編碼結果也必須對映到相同的解碼結果。編碼與解碼的技巧決定了結構法的優劣 [Pandya & Macy, 1996]。

結構法著重於文字中的線段（或筆劃）本身彼此之間的連結方

式，從中取得特徵資訊。其方法主要是從擷取文字中的筆劃著手，取得與筆劃相關的資訊，例如：筆劃的端點、中心點、筆劃傾斜度、筆劃間距離、角度差以及交叉方式等。辨識方法常採用比對方式，例如：直接對基本結構做比對及利用語法規則對基本結構做語法分析。語法分析的方法，較常因為雜訊而造成擷取的基本結構的誤差，或是限於文法規則過於嚴謹，而造成表現大幅降低。結構式特徵辨識方法的優點是較為接近人類直覺的辨識方式，且在處理不規則變形的問題方面，比統計方法更有容忍度。但基本結構是否擷取適宜及特徵之間關係的描述是否能清楚區分字元，都將是影響結果很大的因素 [吳偉賢，2000；方建中，2001]。

3. 類神經網路法（Artificial Neural Networks Method）

類神經網路法具有適應性、非線性處理、平行運算及抗雜訊等優點，在類神經網路法中，辨識的過程一如類神經的精神：包含訓練期與回憶期（辨識期），如果一個類神經網路具有簡單的架構、快速且高效率的演算法，同時在訓練期所挑選的訓練圖樣恰當，則必定是一個快速且高辨識率的網路 [溫福助，2000]。

2-3-3 細線化方法

細線化（Thinning）又稱骨架化（Skeletonizing），應用於二元化之影像，其操作目的是將影像中之線條特徵，縮化成一個單位寬的線條。經由細線化後的線特徵，仍能保留原影像中線條圖形的所有結構性資訊，例如：線條的位置、方向和長度。具有亮度灰階之影像，可以使用局部最大值追蹤方法擷取線條，或以二元化之程序，例如：採用門檻值，轉化為二元化影像 [劉育儒，1992]。

李昭慶（2002）亦指出細線化的意義就是將不等寬度的圖形或字

型的周圍各點，經過多次循環的取捨，而留下寬度只有一個點素所連成的線。細線化主要是用於影像圖形辨識（Pattern Recognition）的前置處理，其作用是將一個二值化影像，刪除圖形或文字影像中心骨架以外之部份，只保留其中心細線化之骨架簡化影像，以作為影像辨識、特徵擷取或運算之用。在資料儲存上，細線化可以有效降低儲存空間，有助於影像處理速度的提昇，在細線化的過程當中，必須要注意到的前提有以下幾點：

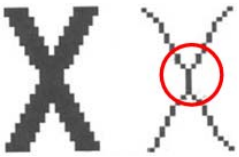
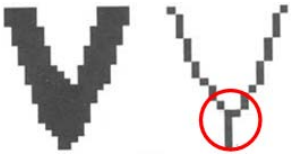
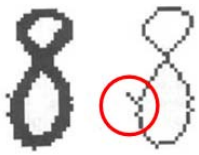
（1）端點（End-Point）不可被消除。

（2）整個圖形的連結性不可被破壞。

（3）不能造成過度消除，造成端點浸蝕問題。

常見的細線化問題，主要如表 2-5 所示。其中，在本研究中因採用的是 Windows 系統中內建的標楷體字型，不會有雜點現象，故研究中並不有第 3 項問題；另外，因主要是針對中文字型及手寫筆劃的單一筆劃進行細線化處理，故第 1 項問題，亦不存在。

表 2-5 常見的細線化問題

問題	說明	圖例
1. 頸部化 Necking	在兩個線條的交界處的點，會形成一個短線段。	
2. 尾端化 Tails	當兩個線段以銳角相交時，尾端會過度細化 。	
3. 贅餘線段 Hairs 或稱 Line Fuzz	當影像有雜點時，亦產生額外的線段。	

以下介紹兩種常用的細線化方法：中軸法（The Medial Axis Approach）及剝皮法（The Peeling Approach）。

1. 中軸法

Blum 於 1967 年定義 MAF（Medial Axis Function）函數，提出中軸轉換法（Medial Axis Transformation，MAT） [Gonzalez & Woods，1992]。在這個方法中視所有物件邊界像素為點波源，這些邊界上每一個像素會觸發相鄰的像素，當兩波相遇時會彼此抵消，即產生所謂的角（Corner），而中軸就是這些角的軌跡，也就是物件的骨架。中軸轉換的功能主要是將影像的骨架擷取出來。只要與影像邊界上至少兩個像素距離相等，則表示此像素在影像的中軸上。中軸的計算方法，可利用在影像中畫圓，圓周必須觸及影像邊界上至少兩個點，所有圓圈的圓心的點組成線的中軸，如圖 2-12 所示 [McAndrew, 2004]。

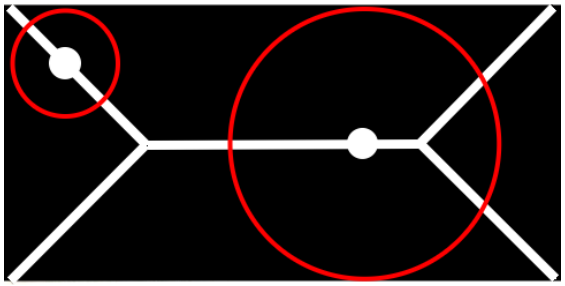
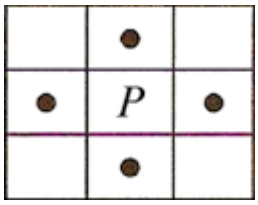
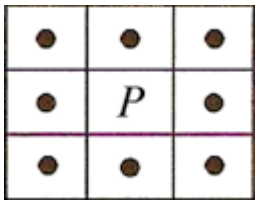
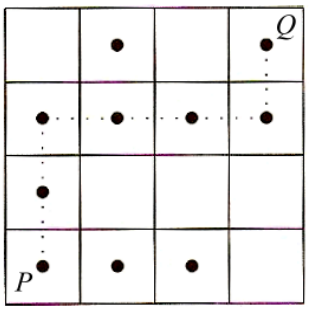
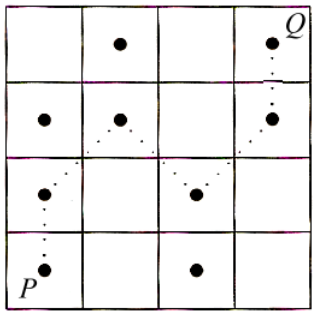
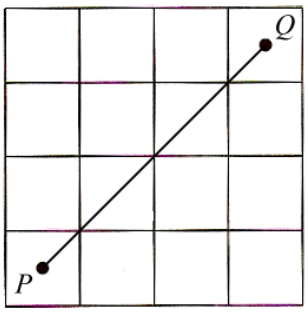
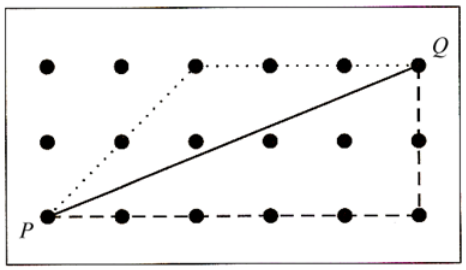
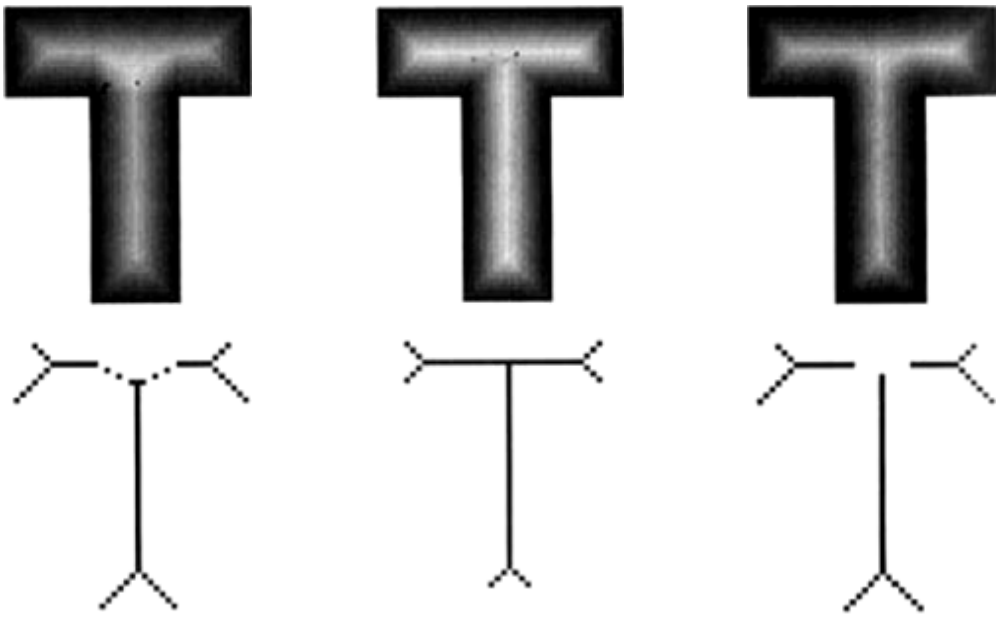


圖 2-12 影像的中軸計算

中軸距離之計算可以採用 4 鄰接（4-Connected）、8 鄰接（8-Connected）及或歐氏距離（Euclidean Distance）等方式。如表 2-6 所示，4 鄰接是指像素點間的連結方式只能上下左右像素點連結方式；8 鄰接則是像素點的八個相鄰像素皆可作為連結點；歐式距離則是採用兩個像素點對角距離的計算方式。而採用不同的距離函數來計算中軸會因影響可能會產生不同的細線化結果，如圖 2-13 所示，[Parker, 1999]。

表 2-6 計算中軸常用的距離函數

4 鄰接	8 鄰接	歐氏距離
		$P=(x1,y1)$ $Q=(x2,y2)$ $d(P,Q)=\sqrt{(x1-x2)^2+(y1-y2)^2}$
		
	4 鄰接： 7 8 鄰接： 5 歐式距離： $\sqrt{5^2+2^2}\approx 5.39$	



(a) 4 鄰接　　(b) 8 鄰接　　(c) 歐氏距離

圖 2-13 不同的距離函數對中軸的影響

中軸法是許多細線化演算法的基礎，中軸法有以下幾個缺點，[廖振偉等，2001]。而中軸法的部份缺點，例如：骨架像素的遺失及無法保證線條連續性等，對於本研究中必須確保筆劃完整性有不利的影響，因此未採用此種細線化方法。

（1）計算時間很長。

（2）影像的解析度要非常高，否則某些狀況會導致骨架像素的遺失。

（3）當物件的像素只有少許差異會產生截然不同的骨架，如圖 2-14 所示可能因一個像素的差異，而產生多餘物件骨架。

（4）無法保證線條連續性，如轉彎處或線條交叉處都有斷線的可能。

（5）因處理過程必需遵照掃描順序，以致於無法使用平行處理技巧。

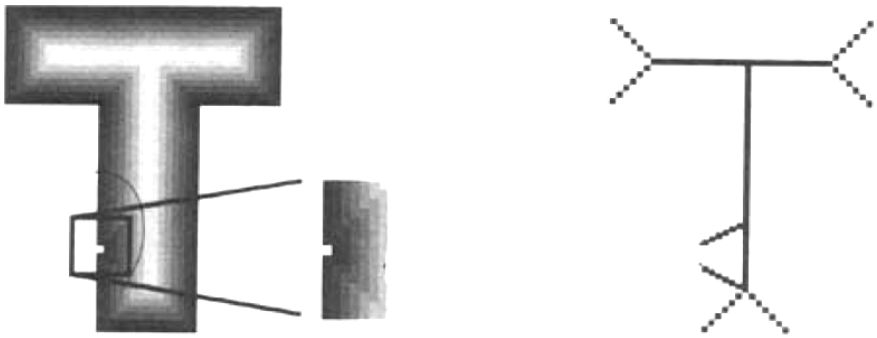


圖 2-14　中軸轉換的缺點

2. 剝皮法

剝皮法又稱為迭代形態法（Iterative Morphological Method），其方法是以非零像素（代表線資料存在的位置）與其鄰近像素之間關係決定此像素是否位於線的邊緣，若是則捨去（即將值以零取代）。以這種類似剝皮的方式由線條兩邊依次縮減，直到剩下一個像素寬，並且保持線的連結狀態。剝皮法的優缺點如下 [廖振偉等，2001]：

- 優點：

（1）具有平行化處理之能力。

（2）線段交接處經細線化後仍能保存完整及總線段長對執行時間只有極小影響。

- 缺點：

（1）因每次作業只去除線條兩側各一個像素，因此線條寬度對執行效率有很大影響。

（2）若線段一側受到誤差干擾，會造成結果呈波浪狀（Wave）或形成小圓圈情況。

（3）線條若太寬，則在轉彎或接合處很難找到真正中心線位置，且易造成原線條縮短情況發生。

Zhang-Suen 細線化演算法 [Zhang & Suen, 1984] 是屬於剝皮法的一種。其方法是在二值化黑白影像中，以 3×3 矩陣為一運算單位區域，對整個影像逐點作運算，並以遞迴方式由外而內逐漸減少點數來達到細線化的目的。在本研究中字型筆劃及手寫輸入筆劃的細線化方法即是採用此法。

假設像素點的值為 1，背景點（白點）的值為 0。對於運算區域中心點P_1的周圍參考點定義為P_2、P_3、P_4、P_5、P_6、P_7、P_8、P_9，運算單位區域參考圖，如圖 2-15 所示。

P_9	P_2	P_3
P_8	P_1	P_4
P_7	P_6	P_5

圖 2-15 細線化運算單位區域參考圖

細線化的過程是判斷某一影像黑點是否應予保留或將予以刪除。首先對整個影像逐點作運算，若P_1為背景點時（即$P_1 = 0$），該點非屬於影像部分，不作任何動作；若P_1為影像黑點時（即$P_1 = 1$），則判斷是否刪除該像素點P_1，相關的步驟如下。如圖 2-16 為以一個 10×10 筆劃影像為例，利用Zhang-Suen細線化演算法的細線化過程。

（1）若滿足下列條件，則記錄P_1須刪除：

(a) $2 \leqq sum \leqq 6$

(b) cycle = 1

(c) $P_2 \times P_4 \times P_6 = 0$

(d) $P_4 \times P_6 \times P_8 = 0$

$sum = P_2 + P_3 + P_4 + P_5 + P_6 + P_7 + P_8 + P_9$。

若 sum = 1，只有 1 個近鄰像素，表示該像素可能位於骨架末端。

若 sum = 7，該像素有 7 個近鄰像素，若刪除會破壞原有基本形狀。

cycle：依 P2～P9～P2 順序，像素點由 1（黑）變 0（白）的次數。

（2）一次刪除步驟（1）全部有記錄的像素點。

（3）若滿足下列條件，則記錄P_1須刪除：

(a) $2 \leqq sum \leqq 6$

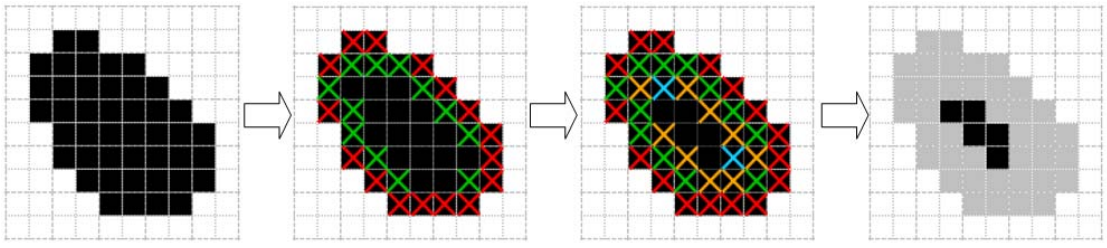
(b) cycle = 1

(c) $P_2 \times P_4 \times P_8 = 0$

(d) $P_2 \times P_6 \times P_8 = 0$

（4）一次刪除步驟（3）全部有記錄的像素點。

（5）重複步驟（1）到步驟（4），直到剩下單一像素點骨架為止。



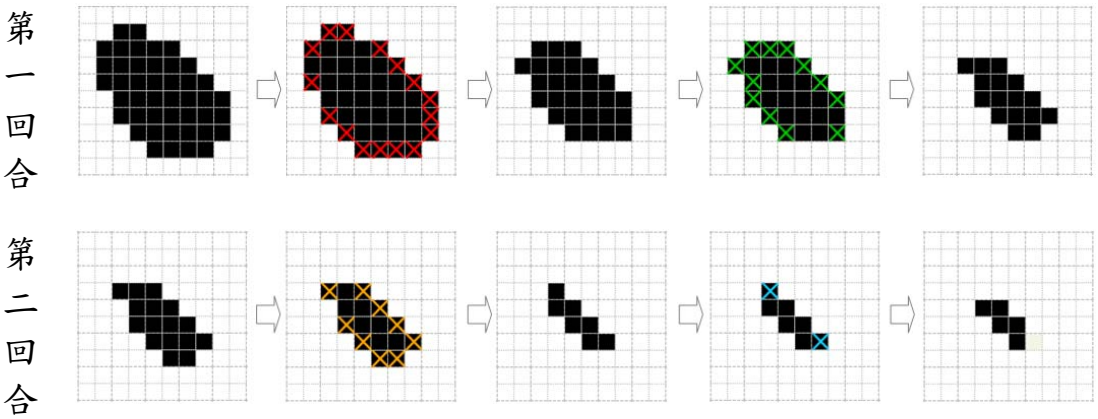


圖 2-16　筆劃影像細線化過程

李昭慶（2002）應用 Zhang-Suen 細線化演算法於電腦化離線筆跡鑑定上，如圖 2-17 所示為字型標楷體粗體之「大」字的字型影像及該字型影像細線化骨架。

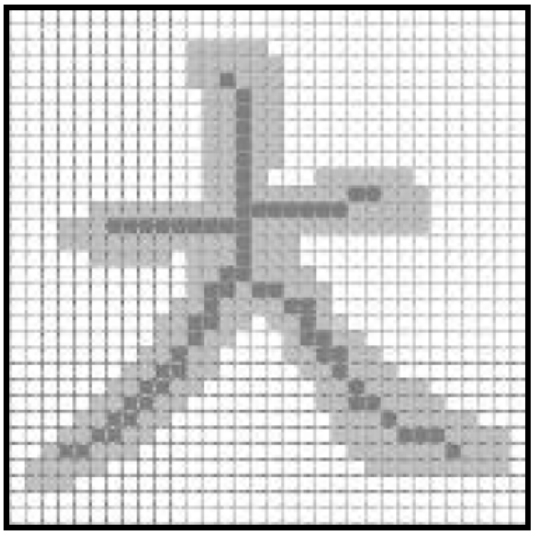


圖 2-17 原始字型影像和字型影像細線化骨架

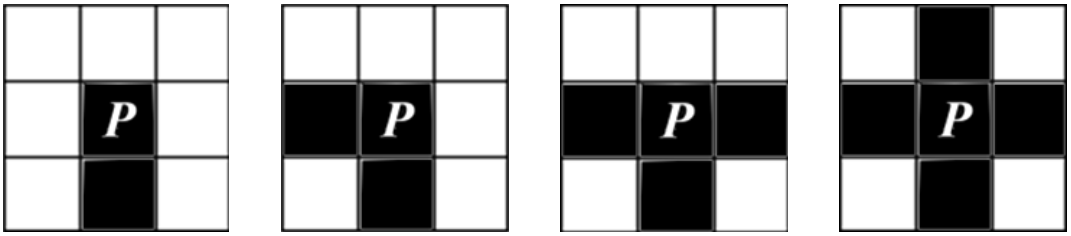
2-3-4 特徵擷取與編碼方法

特徵為辨識及比對作業時最關鍵的要素。在中國書法字型中，不同之筆劃在結構上可能具備相同之結構特徵；因此在辨認流程上，通常會應辨識目標不同而採取幾種不同的特徵擷取配合。饒忻等（2003）認為特徵擷取（Future Extraction）技術在字型筆劃的辨識中是很重要的一環，因為特徵資料的優劣對於辨識能力會有很大的影響。特徵擷取的技術其主要的著重點在於特徵資料容易運算、重要資訊保留、避免減化現象以及提高辨識效能等。以下介紹特徵值擷取與編碼方法相關研究：

1. 特徵擷取方法

Nakamura 等（1999）提出節點特徵值擷取演算法（Skeleton Revision Algorithm），利用一個 3x3 遮罩，針對細線化後的影像，由左至右、由上至下進行掃描，藉由計算各像素之 A(*P*)值來判定節點

型態。所謂 A(*P*)就是計算中心 *P* 點的八鄰接的像素，沿著 *P* 點環繞一圈，從背景點轉換為黑點的次數。若所計算出來的 A(*P*)=1，則判定該像素點具有端點（End Point）特徵；A(*P*)=2 為連結點（Connect Point）；A(*P*)=3 為三叉點（Branch Point）；A(*P*)=4，則為四叉點（Cross Point）。以阿拉伯數字 4 為例，影像經過細線化處理後可由影像中擷取出 3 個端點、1 個三叉點以及 1 個四叉點三種特徵，如圖 2-18 所示[饒忻等，2003]。



A(P)=1，端點　*A(P)*=2，連結點　*A*(*P*)=3，三叉點　*A*(*P*)=4，四叉點

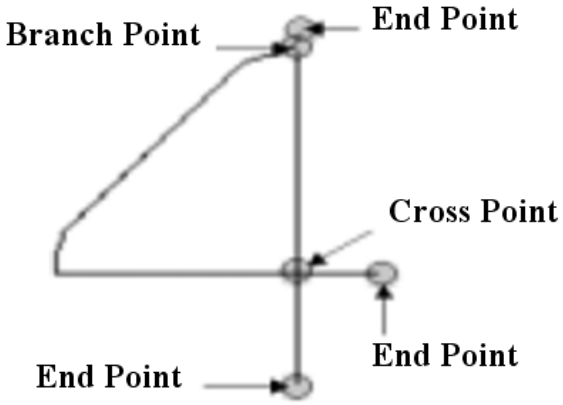


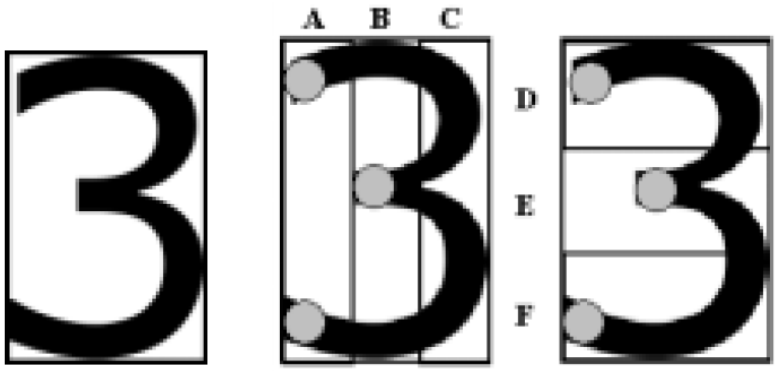
圖 2-18　節點特徵值判定

2. 特徵編碼方法

（1）水平垂直座標軸相似編碼

水平垂直座標軸相似編碼主要分成兩部分：第一部份是依字元端點之垂直與水平座標是否有位於相同區域進行編碼，共編成 2 碼；第二部分是以相同區域字元端點其水平及垂直區域所在位置來編碼，共編成 6 碼。最後將兩部分的二進制值轉換成十進制值當作特徵值。

其方法為首先將文字影像切割至最小區塊，即方框邊界皆緊密靠在文字上，分別依據垂直及水平方向將文字影像等分成 A、B、C 與 D、E、F 六個區域，再依據各端點在水平及垂直座標上是否落於同一區域加以編碼。例如：字元「3」經過細線化處理後可得到 3 個端點，第一部份是依端點之垂直與水平座標是否有位於相同區域進行編碼，垂直方向的分割區域 (A,B,C) 中 A 區域內有出現兩個頂點，所以值為 1，水平方向的分割區域 (D,E,F) 中並未有兩個（含以上）的頂點在同一區域，所以值為 0，第一部份編碼值為二進制值為「10」，轉成十進制值為「2」；第二部份則是將分割的六個區域分別以 0 或 1 值表示有無兩個（含以上）的頂點在同一區域，第二部份編碼值為二進制值為「000001」，轉成十進制值為「1」，故字元「3」之水平垂直座標軸相似編碼為「21」，如圖 2-19 所示 [饒忻等，2003]。



第一部份編碼爲二進制值「10」

1	0
垂直	水平

第二部份編碼爲二進制值「000001」

0	0	0	0	0	1
F	E	D	C	B	A

圖 2-19 字元「3」之水平垂直座標軸相似編碼

（2）節點方向性編碼

節點方向性編碼是針對端點彼此間垂直座標相對位置，依循八連通之 5、6、7 角度方向為特徵來編碼，角度方向如圖 2-20 所示，共

編成三碼 [邱顯達，1992]。以數字 2、5 及大寫字母 C 為例：數字 2 有兩個端點特徵。端點 2 位於端點 1 之右下方，且依循八連通之 7 角度方向，故可求得方向性特徵值為 1。同樣地，數字 5 及大寫字母 C 端點 2 與端點 1 之相對位置分別依循八連通之 5 及 6 角度方向，故可求得方向性特徵值為 4 及 2，如圖 2-21 所示 [饒忻等，2003]。

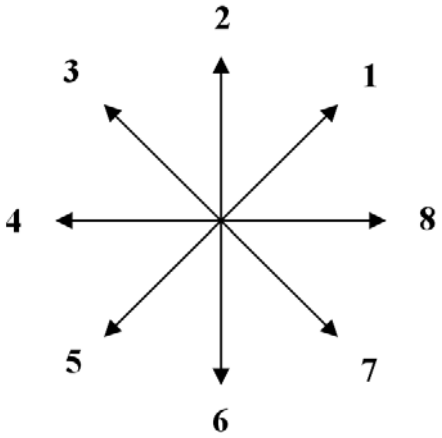


圖 2-20 八連通方向

5	6	7
0	0	1

編碼值爲 **001**

轉換成十進制值爲 **1**

5	6	7
1	0	0

編碼值爲 **100**

轉換成十進制值爲 **4**

5	6	7
0	1	0

編碼值爲 **010**

轉換成十進制值爲 **2**

圖 2-21 字元「2」、「5」及「C」之方向特徵編碼

（3）節點四角定位編碼

節點四角定位編碼是依據順時針方向將影像區分成 A、B、C、D 四個區塊，同時依據字元頂點分佈的位置來編碼，共編成 4 碼，如圖 2-22 所示 [顏技文，1994]。以大寫字母 G、S 及數字 7 為例，大寫字

母 G 有兩個端點特徵。端點 1 落在區域 B 中而端點 2 落在區域 C 中，可求得節點四角定位特徵值為 6。同樣地，大寫字母 S 及數字 7，其端點 1 與端點 2 分別落在區域 B、D 及 A、D 中，可求得節點四角定位特徵值分別為 10 及 9，如圖 2-23 所示 [饒忻等，2003]。

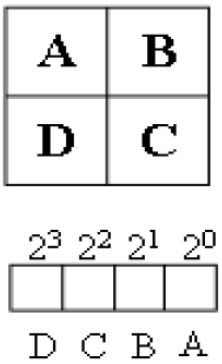
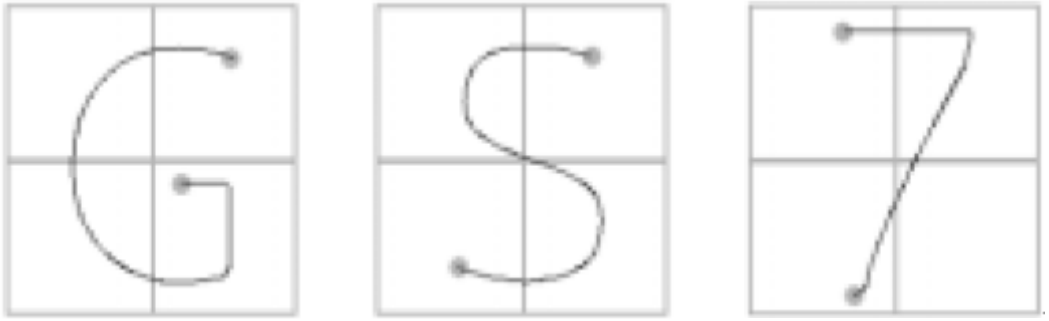


圖 2-22 節點四角定位編碼



D	C	B	A
0	1	1	0

編碼值爲 **0110**

轉換成十進制值爲 **6**

D	C	B	A
1	0	1	0

編碼值爲 **1010**

轉換成十進制值爲 **10**

D	C	B	A
1	0	0	1

編碼值爲 **1001**

轉換成十進制值爲 **9**

圖 2-23 字元 G、S 及 7 之節點定位特徵編碼

2-4 SolidWorks 二次開發

2-4-1 二次開發簡介

二次開發（Secondary Development）功能是指 CAD 軟體提供讓使用者可以依據個人需求設計程式模組，利用高階程式語言控制

CAD 軟體執行所需的功能，對 CAD 功能進行擴充及修改，達到客製化的結果。目前常見的 CAD 軟體，例如：SolidWorks、Pro/E、AutoCAD、CATIA、Rhinoceros、UG、MasterCam…等都有提供二次開發的工具或模組，讓使用者可用來撰寫客製化模組的功能。

本研究中是利用 SolidWorks 提供的二次開發工具 SolidWorks API（Application Programming Interface，應用程式介面），結合 Delphi 程式語言開發工具，將螢幕上字型模擬的結果輸出到 SolidWorks 中，讓書法字型模擬結果能在 3d 繪圖軟體中進行後續處理及設計應用，如圖 2-24 所示。

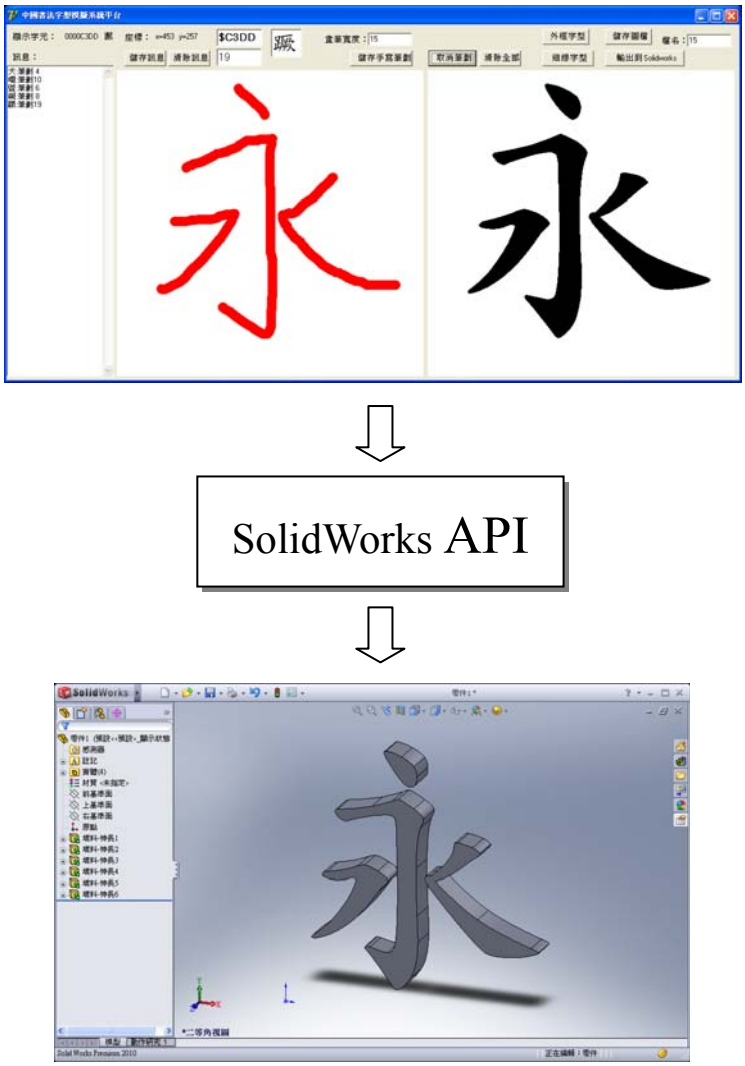


圖 2-24 字型模擬系統平台中 SolidWorks API 應用

2-4-2 應用 SolidWorks API 的方法

余奕昌（2006）指出應用 SolidWorks API 的方法主要分成 OLE 自動化控制、COM 連結技術及巨集 VBA 程式三種，分述如下：

1. OLE 自動化控制

OLE（Object Linking and Embedding，物件連結與嵌入）是物件連結與嵌入技術的簡稱。OLE 提供了方便的技術將來自不同程式的各種類型的功能和資料結合起來。利用 OLE 技術，使用者可以使用來自兩個或多個 Windows 應用程式的資源來解決複雜的問題。SolidWorks 是一套建構於 Windows 環境下的 CAD/CAE/CAM/PDM 多功能 3D 繪圖系統，其架構支援了 OLE 標準，使用者可利用 OLE 自動化控制技術利用其內建的 OLE 物件來編寫程式。SolidWorks API 是 SolidWorks 的 OLE 應用程式開發接口，使用者可以利用它建立所需的或專用的 SolidWorks 功能，以 OLE 自動化控制方式製作*.EXE 檔進行外部的連接控制。只要是支援 OLE 的程式語言，例如：Visual Basic、Visual C++、VB.NET、C#、Delphi、Java 等語言都可以利用 SolidWorks API 開發 SolidWorks 的功能模組。OLE 最主要的作用為連結與嵌入，其功能比較，如表 2-7 所示［陳彥名，2006]：

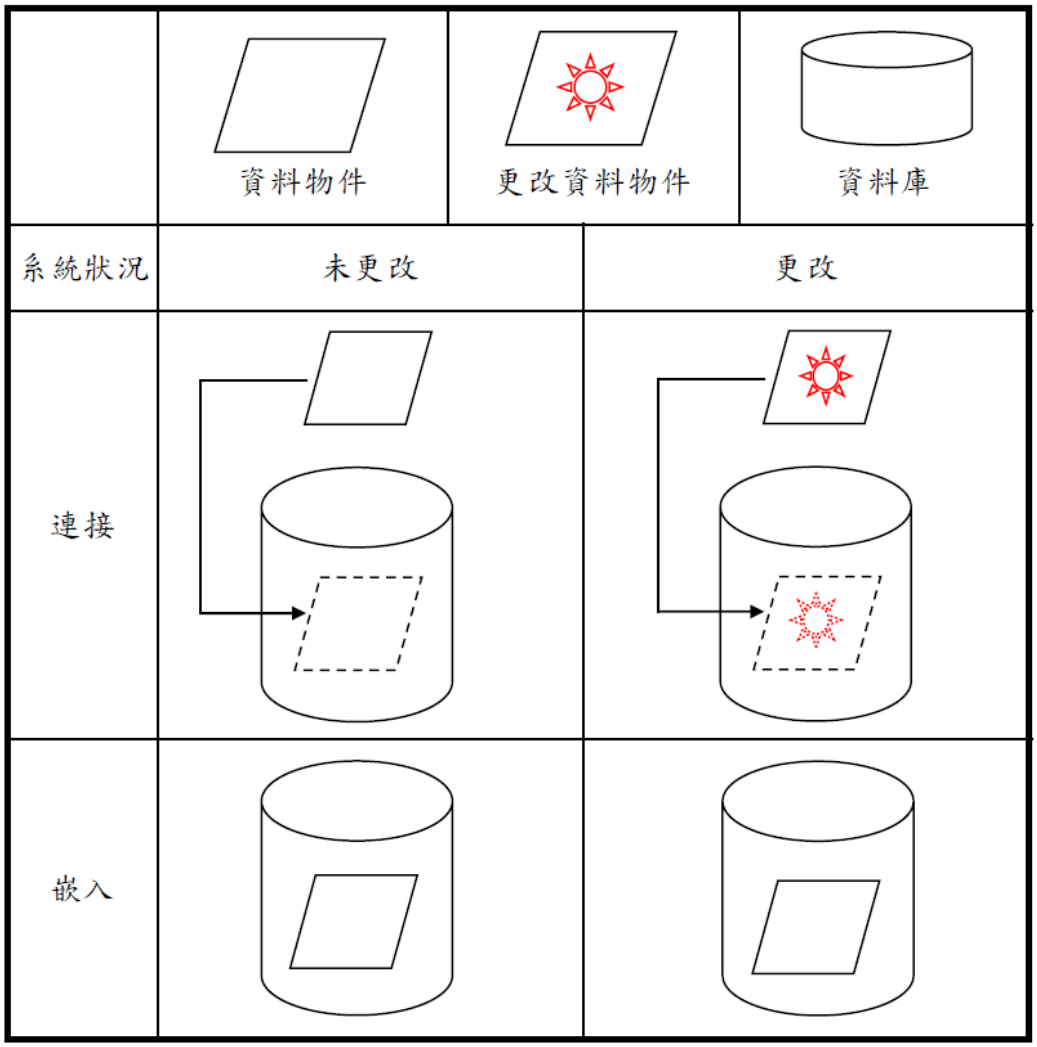
（1）連結

資料本身是獨自存在於自己的物件之中，資料會連結到所需的資料庫系統中做相關連結，然而連結的資料物件內容產生變化，而資料庫系統也可預見內容的變化。

（2）嵌入

將資料物件整個複製一份放置在資料庫系統中，其中資料庫系統只與在資料庫中的資料物件做連結，不跟原始資料物件做連結，所以當資料物件改變時，資料庫中的資料物件不會有任何改變。

表2-7 OLE 功能比較



SolidWorks 相當於一個 OLE 伺服器，提供大量的 OLE 物件，以及這些物件所擁有的方法和屬性，使用者可以利用程式對這些 OLE 物件進行控制，即可在自己開發的應用軟體中執行 SolidWorks 中的各項功能，例如：建立曲線、構造實體、檢查曲面表面參數…等。圖 2-25 為 SolidWorks API 的 OLE 物件層級，從圖中可看到 SolidWorks API 的物件分為若干層，每一個物件又都有自己的屬性、方法和事件。其中 SldWorks 為 SolidWorks API 中最上層的物件，它能夠直接或間接的呼叫 SolidWorks API 中所有其他的物件。

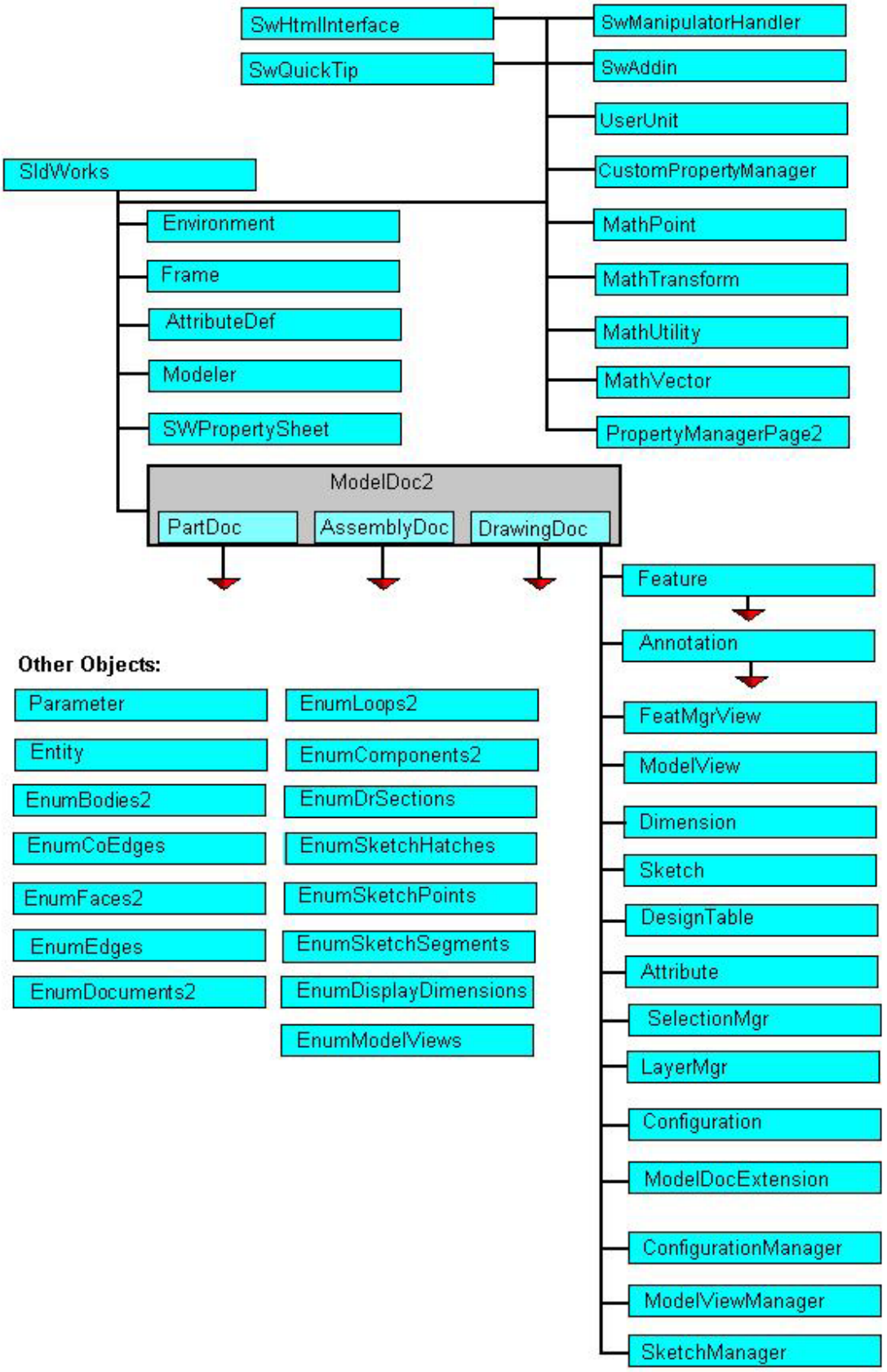
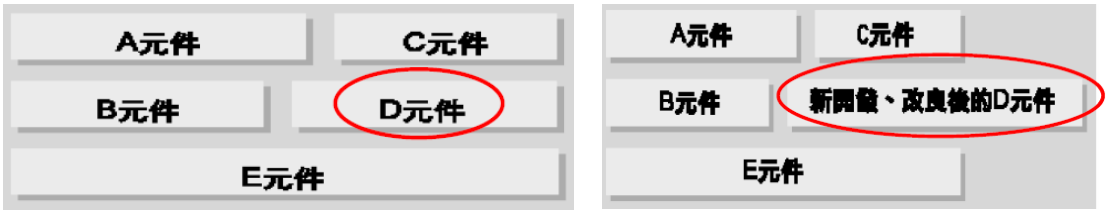


圖 2-25 SolidWorks API 物件層級

2. COM 連結技術

COM（Component Object Model，元件物件模組）是一種建立可動態交替更新組件的規範，可用來作為不同程式間溝通的規範。傳統的應用程式是由單一的程式組成，當程式撰寫到可以利用且上市販售的同時，便是該程式的結束，因為其不得再進行程式的修改。隨著對程式語言的進步與要求程度，傳統的程式撰寫方法已經不合時代需求，因此有了元件式程式設計方法的產生。如圖 2-26(a)所示，主程式由五個元件程式所組成，這些程式以*.DLL 或*.EXE 作為元件程式的格式，這種建構程式的方法稱為元件架構法（Components Architecture），而 COM 就是切開主程式的方法。利用這個建構程式的方法的好處就是可以加入或替換新元件的方式來達到程式的更新的目的。如圖 2-25(b)所示，將重新開發的 D 元件程式取代舊有的 D 元件程式的位置，如此便可不必變更主程式的內容且達到元件程式的更新。SolidWorks API 亦可以 COM 連結技術的方式來進行自動化應用。



(a) 元件式的程式　　(b) 元件程式的更新

圖 2-26 COM 連結技術

3. 巨集 VBA 程式

巨集（Macro）操作的環境為 VBA（Visual Basic for Application），在 Microsoft Office 中，所有使用巨集的應用程式都共享一種建立在 Basic 語言之上的通用巨集語言。雖然其特性相當的便利，但是此種

語言的語法不夠完整且受到很多的限制；因此 Microsoft 在 1993 年推出一種可以讓多種應用程式共享並可針對應用程式內部編譯的語言，就是 VBA。但是 VBA 並不能歸類為程式語言，它雖然具有完整的程式語言基本結構，且其程式語法與 VB 十分相似，但是其主要功能在於記錄操作過程。因此使用者在 SolidWorks 下將自動化操作的步驟錄製成巨集，在錄製完成後就會生成 SolidWorks API 的 VBA 程式碼。在 SolidWorks 中使用巨集的步驟如下：

1. 錄製巨集：如圖 2-27 所示，按下錄製的功能指令，使用者在 SolidWorks 中所進行的任何步驟都會被錄製下來，形成 VBA 的程式碼。

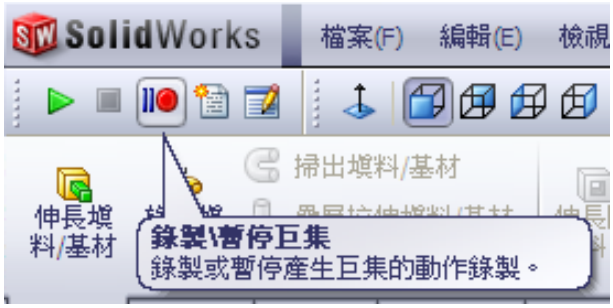


圖 2-27 錄製巨集

2. 停止巨集：如圖 2-28 所示，當使用者在錄製巨集以後，不在需要錄製時，按下停止功能指令，此時錄製巨集便會結束且會進行存檔的動作不再錄製。

圖 2-28 停止巨集

3. 編輯巨集：當開啟編輯錄製的巨集功能時，便會自動開啟 VBA 的

編輯器並且顯示錄製的 VBA 程式碼，如圖 2-29 所示，圖 2-30 為編輯巨集的畫面，其程式碼為在 SolidWorks 中繪製一條不規則曲線，如圖 2-31 所示。

圖 2-29 編輯巨集

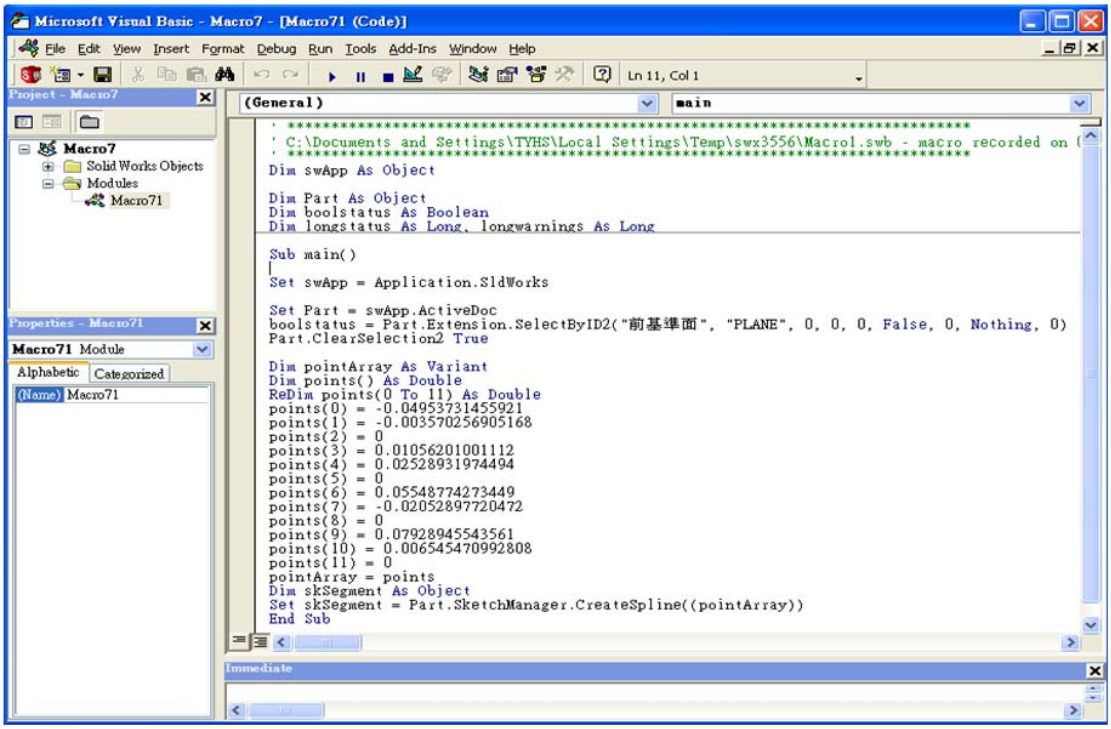


圖 2-30 VBA 程式碼

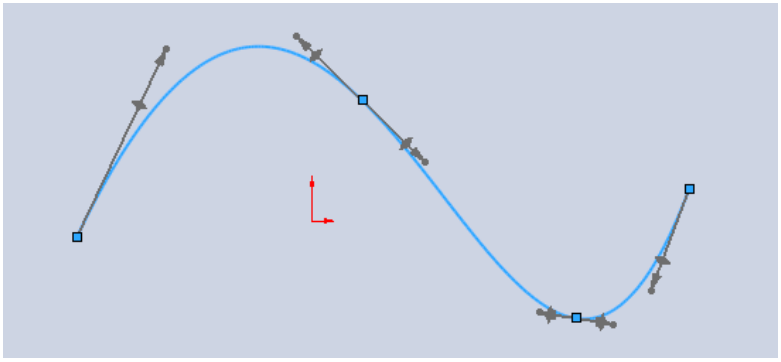


圖 2-31 VBA 程式碼（圖 2-29）所繪製的圖形

雖然使用巨集來進行 SolidWorks API 的功能開發十分的容易且方便，但是也有其功能的不足的地方。最大的問題點在於錄製巨集所產生的 VBA 程式碼和 SolidWorks 的動作指令間，並不是雙向溝通方式，巨集內產生的 VBA 程式碼只具有傳值給 SolidWorks 的單向動作指令，而不能得到 SolidWorks 的回傳值指令。

上述所介紹的三種 SolidWorks API 的應用方法各有其優缺點，整理如表 2-8 所示。

表 2-8 SolidWorks API 的方法比較

方法	優點	缺點
OLE 自動化控制	操作易懂且可以利用 VBA 程式碼輔助功能模組開發。	無法直接整合至 CAD 軟體，須靠程式控制。
COM 連結技術	可以將開發的模組整合至 CAD 軟體功能。	操作較複雜且無法利用 VBA 程式碼輔助功能模組開發。
巨集 VBA 程式	開發方便且使用上較容易。	不具有雙向溝通的能力，且其在操作上必須對軟體有所基礎。

本研究的中國書法系統模擬平台是利用 Delphi 7.0 程式語言所開發，在三種開發方式中，因錄製巨集的方式缺乏雙方溝通性，並不適用本系統；而亦不適用 COM 的方式，因此採用的 SolidWorks API 應用方式為以 OLE 自動化的方式作*.EXE 檔進行外部的連接控制。

錄製巨集的方式雖然不適用於平台開發所需，但仍有其可用性。因為可以利用錄製巨集的方式得到 VBA 的程式碼，在藉由 VBA 程式碼得到相對 SolidWorks 功能的指令。例如：錄製不規則曲線所產生的巨集，可由 VBA 程式碼中得知其功能指令為「CreateSpline」；

再利用 CAD 軟體 SolidWorks API 所提供的「Help」說明，可以知道各種指令的限制和參數，對於程式的開發有其參考價值。

2-5 書法字型應用

2-5-1 書法字型資料庫建立

羅鳳珠、周曉文（1998）完成小篆字型字庫的建立，開發高效率字型造字平台，完成字型認別、替代、比對及檢索等作業，達到古籍數位化的目標，如圖 2-32 所示。

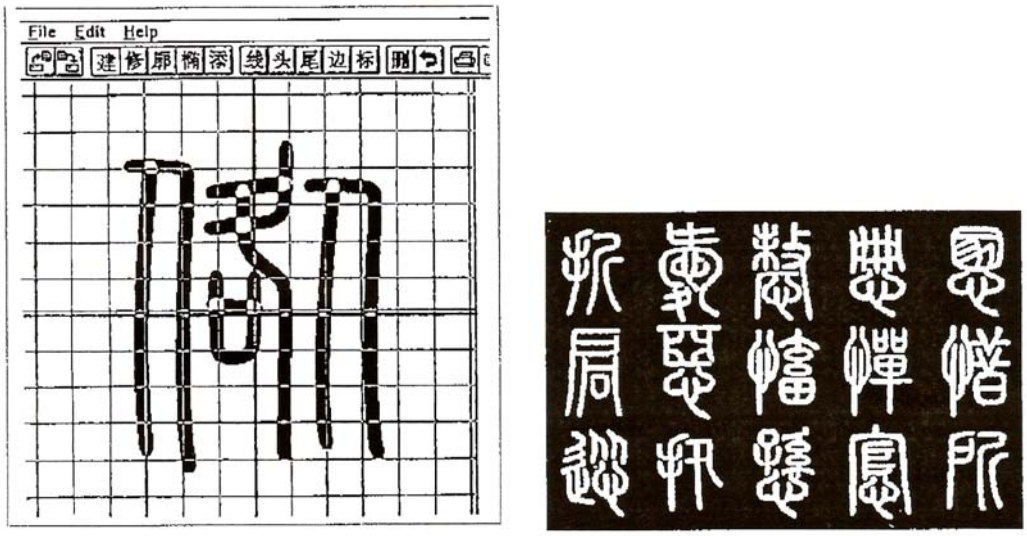


圖 2-32 小篆字型造字平台及字庫圖樣

2-5-2 書法書寫模擬應用系統

以下介紹幾個關於書法書寫模擬應用系統相關研究：

1. 毛筆筆鋒之模擬及其水墨書畫系統

連挺佑（2002）開發毛筆筆鋒之模擬及其水墨書畫系統，以彎曲彈簧的力學模型為基礎，模擬出毛筆筆頭隨著力道作用而變化；水墨傳輸系統則是模擬出水墨在毛筆跟宣紙上傳輸的情形。系統採用多層的水墨結構，配合水墨傳遞、水墨遞減與水墨擴散的演算法，表現出

水墨畫的筆觸效果，讓使用者可利用數位手寫板作為輸入裝置，透過直覺式的操作介面模擬與創作，如圖 2-33 所示。

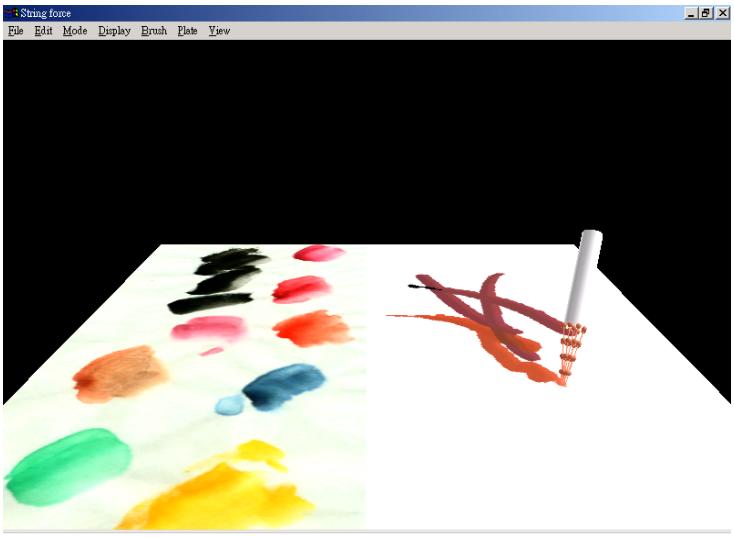


圖 2-33 毛筆筆鋒之模擬書畫系統

2. e 筆書法

淡江大學研發團隊融合了書法藝術與電腦科技，開發了 e 世代嶄新的書寫應用系統「e 筆書法」。利用電腦筆（e-Pen）在平版電腦或數位手寫板上書寫，不再需要傳統書寫的用筆、紙張、墨水及字帖等，即可以模擬出原子筆、簽字筆及毛筆等各種字型質感；另外也可運用於習字、練字、書寫，進而各種書法作品佈局上。與日本書道協會（日本通信教育連盟 U-CAN）合作開發漢字藝術素材總庫，將中國文字各種書體的字型，透過數位典藏程式，歸納成素材庫，提供學習者、欣賞者及設計者自由套用，如圖 2-34 所示。

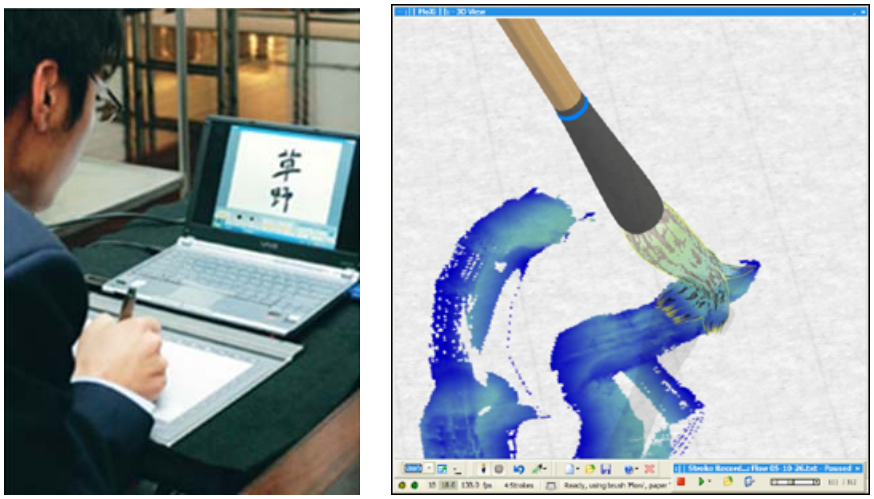


圖 2-34 淡江大學 e 筆書法系統

3. 數位虛擬毛筆軟體

香港科技大學 Nelson Chu 開發數位虛擬毛筆軟體寫意軒（Bristle Studio，又稱為 MoXi），使用者可透過數位手寫板及虛擬 3D 毛筆擬擬水墨筆觸寫毛筆或畫水墨畫，如圖 2-35 所示。

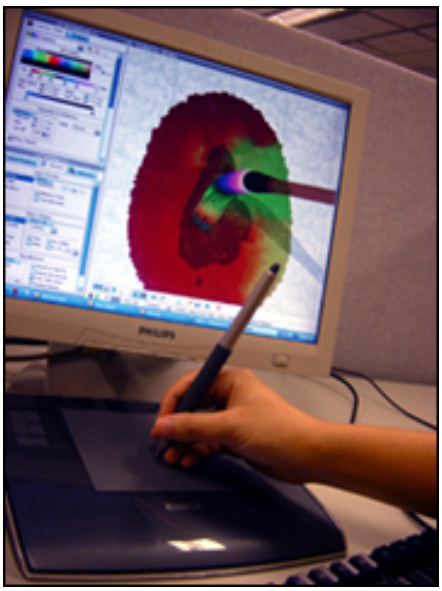


圖 2-35 數位虛擬毛筆軟體

2-5-3 個性化書法應用

個人獨特風格在現代逐漸被重視，因此關於書法的應用如何展現個性化，也是十分熱門的研究。謝忠勳（2006）提出個人化向量字型方法，經由擷取手寫字元特徵，據以變更標準字型筆劃長度、斜率及

字根長寬比例，產生獨一無二個人化之字型供使用者使用，如圖 2-36 所示。文鼎圖王造字管理系統的智慧型自動描邊造字系統，建置有數位化簽名、個人印章等功能，如圖 2-37 所示。

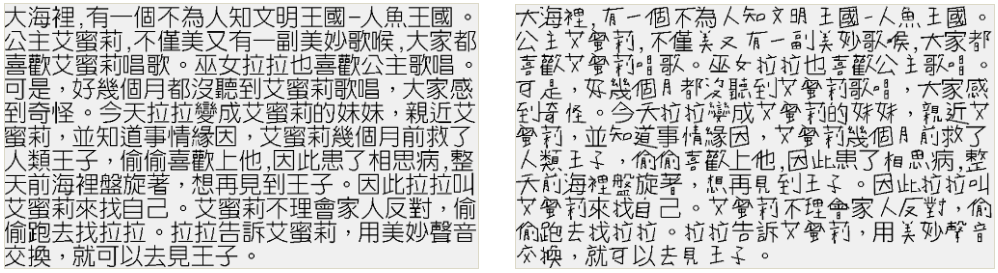


圖 2-36 個人化向量字型

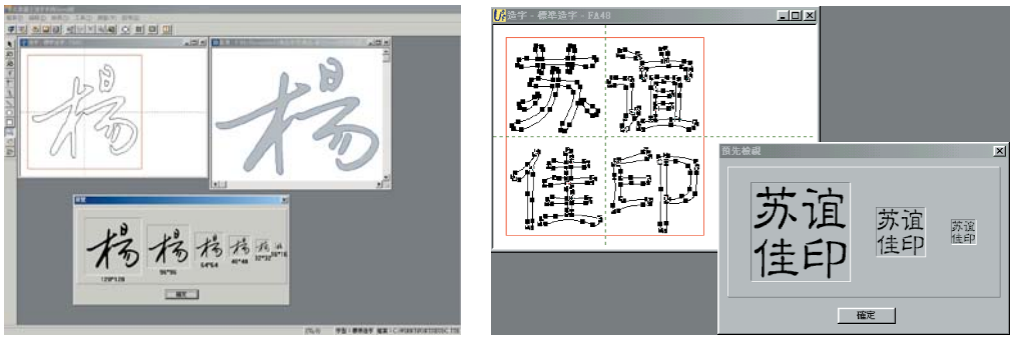


圖 2-37 文鼎圖王造字管理系統

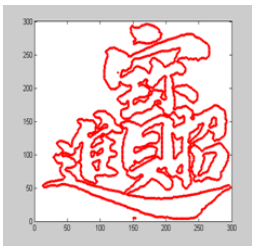
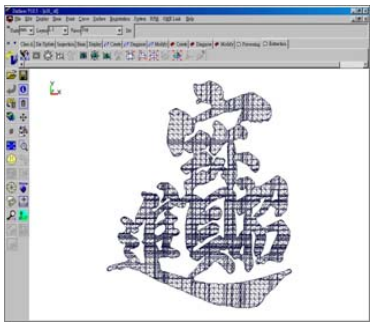
2-5-4 書法學習系統

日本任天堂公司於 2008 年開發文字學習軟體「DS 美文字訓練（DS 美文字トレーニング）」，讓使用者可透過觸控螢幕與觸控筆，依照軟體所設計的訓練課程，來學習正確美觀的漢字書寫技巧，如圖 2-38 所示；

圖 2-38 任天堂 DS 美文字訓練展示網頁

2-5-5 書法合成字型設計

王中行等（2004, 2006）以如何建立中國書法及碑刻的數位化保存為方向，結合影像辨識處理的逆向設計技術，作為中國書法之數位化工程轉換依據，讓中華傳統書法藝術得以延續保存，並發揚光大為目標。在書法字型逆向設計研究上，首先，藉由掃瞄器讀取中文書法字型，並以灰預測理論導入影像處理，利用 Matlab 語言撰寫影像辨識程式，完成中國書法字型邊緣偵測；進而將字型影像邊緣輪廓轉換成點群資料，再將其匯入逆向 CAD 軟體- Surfacer，以重建字型曲線輪廓；再轉換成 STL 格式，藉由快速原型機輸出字型實體，以達成字型之完整逆向設計與數位化的目標。在實例驗證上，並以中國書法中獨具特色的吉祥合成字型為案例，以毛筆書寫「招財進寶」複合字，再經由掃描器將書寫於紙上之字型轉換為影像檔案；藉由灰預測逆向辨識技術取得字型之影像邊緣，再將書法字型轉為點群文字檔，進行 Spline 曲線建構；最後再利用快速原型，將其以 3D 立體模型呈現，使中國書法字型能以數位化方式，並提供相關領域應用，如圖 2-39 所示。


(a)

(b)

(c)

(d)

(e)

(f)

(a) 招財進寶影像檔　(b) 招財進寶之影像輪廓
(c) 招財進寶轉成*.stl 格式
(d) 招財進寶文鎮　(e) 招財進寶印章
(f) 招財進寶中國結吊飾造型

圖 2-39 中國書法吉祥合成字之逆向設計研究

第三章　研究步驟與方法

本研究目的為中國書法字型模擬平台的開發，首先的問題便是不同字型的筆劃取得的問題。相同的字型在不同的書法家呈現的方式亦各自不同。如何建立一個可以包含多種書法字型的模擬平台，需先建立一個雛型作為後續研究之用。目前開發的電腦中文字型相當多，對建立的筆劃資料庫的完整性有相當大的助益。所以研究上是針對目前筆劃型造字的電腦中文字型進行分析，快速取得研究所需的字型筆劃，建立一套雛型系統，以對後續其他字型的研究有所助益。

研究上是利用自行開發的程式，建立一套模擬系統平台。從現有的電腦字型檔案進行分析，擷取字型特徵並加以細線化處理及特徵編碼，建構書法字型筆劃特徵資料庫；在使用者操作介面操作上，當使用者利用滑鼠或數位筆…等輸入設備，書寫每一個筆劃後，系統便會自動擷取手寫筆劃進行細線化處理及特徵擷取，再比對筆劃資料庫，輸出模擬書法字型。系統並提供將結果輸出至 SolidWorks 中的功能，以利模擬字型後續的設計與應用，以下就相關步驟說明。

3-1 建立筆劃特徵資料庫

3-1-1 筆劃特徵擷取

要建立使用者書法模擬系統首先就是要建立相對應字型的筆劃資料庫，筆劃特徵資料庫與書法模擬系統的效能和辨識的準確性有密不可分的關係。而建立筆劃特徵資料庫的方法首先為取得筆劃資料，取得的方式可分為下列三種方式：

（1）由掃描的書法點陣圖形，透過邊緣輪廓的擷取和向量化建立。
（2）透過人工輸入建立各種不同的筆劃。
（3）由現有的電腦書法字型中擷取。

由於現有的電腦書法字型都由專人書寫建立且筆劃資料完整，所以研究上採用的筆劃資料來源為現有的電腦書法字型，即為微軟 Windows 作業系統中內建的「標楷體」字型；且此字型是採用教育部所提倡的國字標準字型，與國人的書寫習慣較為一致。

標楷體字型是屬於 TrueType 向量字型，其造字方式為筆劃型的造字方式，其字型架構資料可由 Microsoft（微軟）的 MSDN（Microsoft Development Network）中取得。研究上是利用 Win32 API 中的 GetGlyphOutline 函數讀取指定字元的字型相關資料。TrueType 向量字型中的字元資料結構排列方式如下，一個字元（Character）可以是由多個筆劃（Contour）組成，而每一個筆劃又由多個曲線（Curve）組成，如圖 3-1 所示。經計算統計標楷體共有 23,230 個字元及 178,196 個筆劃資料。

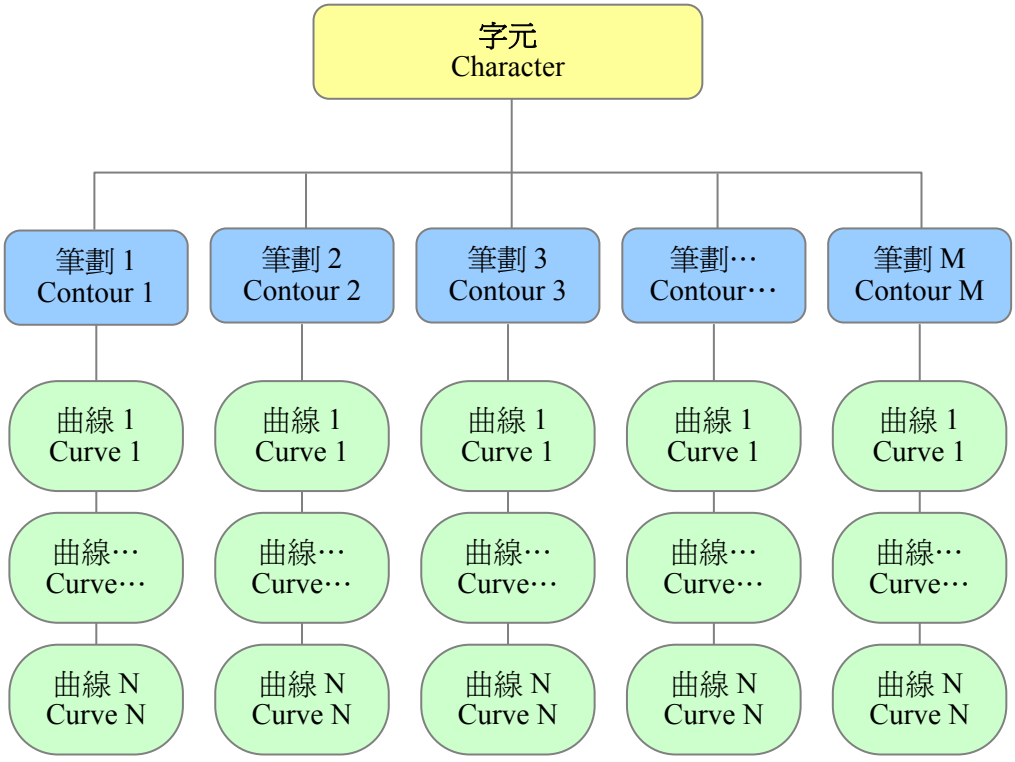


圖 3-1　TrueType 字型的字元資料結構

利用 GetGlyphOutline 函數所得到的字元資料結構，排列方式如下所示。即每一個字元可以分成一個或多個筆劃（Contour），而每一個筆劃的開頭就是一個 TTPOLYGONHEADER 的結構。

■ 字元資料結構

Contour 1's TTPOLYGONHEADER ➔ TTPOLYCURVE 1-1….. TTPOLYCURVE 1-N1 ➔
Contour 2's TTPOLYGONHEADER ➔ TTPOLYCURVE 2-1….. TTPOLYCURVE 2-N2 ➔
Contour 3's TTPOLYGONHEADER ➔ TTPOLYCURVE 3-1….. TTPOLYCURVE 3-N3 ➔
……
Contour M's TTPOLYGONHEADER ➔ TTPOLYCURVE M-1….. TTPOLYCURVE M-NM

其中 TTPOLYGONHEADER 和 TTPOLYCURVE 的資料結構如下所示。TTPOLYGONHEADER 的結構中，分別記錄筆劃所佔的記憶空間（cb）、型態（dwType）以及起始點位置（pfxStart）。

■ TTPOLYGONHEADER 資料結構

```
typedef struct _TTPOLYGONHEADER {
   DWORD     cb;          // 佔用 4 Bytes，cb 的值通常爲 16
   DWORD     dwType;   // 佔用 4 Bytes
   POINTFX pfxStart;     // 佔用 8 Bytes，記錄筆劃的起始位置
}TTPOLYGONHEADER, *LPTTPOLYGONHEADER;
```

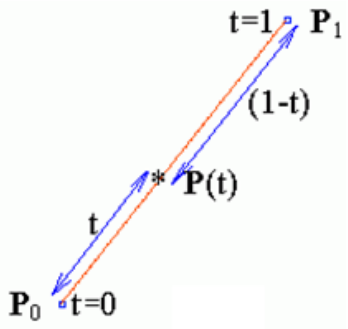
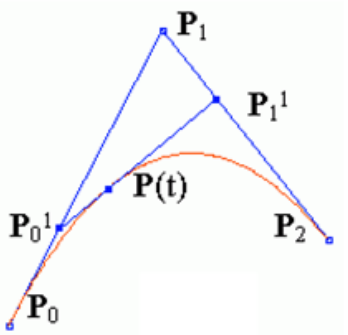
在 TTPOLYGONHEADER 後接著的是一個或多個的 TTPOLYCURVE 的結構，其中 TTPOLYCURVE 的結構分成三個部分：

■ TTPOLYCURVE 資料結構

```
typedef struct tagTTPOLYCURVE {
   WORD      wType; // 佔用 2 Bytes，記錄線條型態
   WORD      cpfx;    // 佔用 2 Bytes，記錄點的個數
   POINTFX apfx[1]; // 佔用 4+cpfx*8 個 Bytes
}TTPOLYCURVE, *LPTTPOLYCURVE;
```

1. 第一個部份為曲線型態（wType），可分成三種曲線類型：線性曲線（Polyline）及控制點一個和二個的貝茲曲線，如表 3-1 所示。

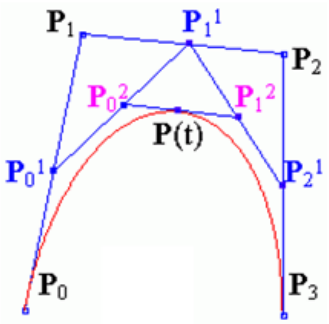
表 3-1　TrueType 字型的曲線型態

（1）Polyline（線性曲線）
由 0 至 1 的連續t，**P**(t)描述一條由$\mathbf{P}_0$至$\mathbf{P}_1$的直線，例如：當t=0.25時，**P**(t)即一條由點$\mathbf{P}_0$至$\mathbf{P}_1$路徑的四分之一處。 $\mathbf{P}(t) = (1-t)\mathbf{P}_0 + t\mathbf{P}_1 \ , \ 0 \le t \le 1$ 
（2）QuadraticBezier Spline （Quadratic 貝茲曲線）
P(t)描述一條由P_0至P_2之間加上一個控制點P_1所構成的曲線，建構的方式如下： $\mathbf{P}_0^1 = (1-t)\mathbf{P}_0 + t\mathbf{P}_1 \ , \ \mathbf{P}_1^1 = (1-t)\mathbf{P}_1 + t\mathbf{P}_2 \ ,$ $\mathbf{P}(t) = (1-t)\mathbf{P}_0^1 + t\mathbf{P}_1^1 = (1-t)[(1-t)\mathbf{P}_0 + t\mathbf{P}_1] + t[(1-t)\mathbf{P}_1 + t\mathbf{P}_2]$ $= (1-t)^2\mathbf{P}_0 + 2(1-t)t\mathbf{P}_1 + t^2\mathbf{P}_2 \ ,$ $\mathbf{P}(t) = \sum_{i=0,2} B_i^2(t)\ \mathbf{P}_i$ • 由$\mathbf{P}_0$至$\mathbf{P}_1$的連續點$\mathbf{P}_0^1$，描述一條線性貝茲曲線。 • 由$\mathbf{P}_1$至$\mathbf{P}_2$的連續點$\mathbf{P}_1^1$，描述另一條線性貝茲曲線。 • 由$\mathbf{P}_0^1$至$\mathbf{P}_1^1$的連續點P(t)，描述一條二次式貝茲曲線。 

（3）Cubic Bezier Spline（Cubic 貝茲曲線）

P(t)描述一條由$\mathbf{P}_0$至$\mathbf{P}_3$之間加上二個控制點$\mathbf{P}_1$及$\mathbf{P}_2$所構成的曲線，為建構高階曲線，需要相應更多的中介點，Cubic貝茲曲線屬於三次式曲線，可由線性貝茲曲線描述的中介點 $\mathbf{P}_0^1$、$\mathbf{P}_1^1$、$\mathbf{P}_2^1$，和由二次式曲線描述的點$\mathbf{P}_0^2$、$\mathbf{P}_1^2$建構而成。

$$\mathbf{P}(t) = \mathbf{P}_0 (1-t)^3 + 3\,\mathbf{P}_1\, t(1-t)^2 + 3\,\mathbf{P}_2\, t^2 (1-t) + \mathbf{P}_3\, t3\,,\ 0 \le t \le 1$$



2. 第二部分為曲線用的點數（cpfx）。
3. 第三部份為曲線使用的點資料（apfx），而點資料的資料結構為POINTFX，其中含有兩個 FIXED x,y 分別記錄點資料的 x,y 座標，而每一個 FIXED 結構又分成兩個部分：value（整數部分）及fract（小數部份）。

■ POINTFX 資料結構

```
typedef struct tagPOINTFX {
   FIXED x;
   FIXED y;
}POINTFX, *LPPOINTFX;
```

■ FIXED 資料結構

```
typedef struct _FIXED {
   WORD   fract;
   short value;
}FIXED;
```

以「永」字為例，在標楷體的造字中，共分為六個筆劃，其各筆劃原始資訊如下，如圖 3-2 所示，其筆劃資訊的內容所代表的意義如下所列，其中❶❷❸❹❺ 的數學關係式為：

$$❶ = 16 + (❷ + ❸) * 4 + (❹ + ❺) * 8$$

❶ 筆劃在字型檔佔用空間（Byte）

❷ 筆劃中直線的個數

❸ 筆劃中曲線的個數

❹ 筆劃中直線使用的點數

❺ 筆劃中曲線使用的點數

❻ 筆劃的 xy 角度

❼ 字元

❽ 筆劃在字元中的第幾筆

	176,0,8,0,16,392,永,1
	380,2,17,2,34,707,永,2
	148,1,6,1,12,487,永,3
	416,0,20,0,40,451,永,3
	216,0,10,0,20,383,永,4
	216,0,10,0,20,382,永,5

圖 3-2 標楷體「永」字的原始筆劃資訊

以「永」字的第一筆「點」為例，該筆劃中共有 0 條直線，8 條曲線，曲線的部分使用 16 個點（一個曲線使用 2 個點），夾角為 39.2 度，使用空間為 16 + (0+8) * 4 + (0+16)* 8 = 176 個 Bytes，如圖 3-3 所示。

❶	❷	❸	❹	❺	❻	❼	❽
176	0	8	0	16	392	永	1

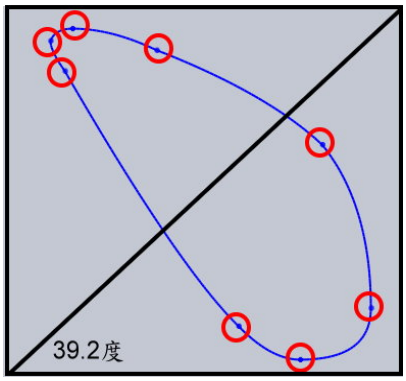


圖 3-3 「永」字的第一個筆劃資訊

在研究過程中，發現許多的字型並非以筆劃的方式造字，所以在應用上需配合其他的演算法將已經合併的筆劃擷取出來。而以筆劃方式造字的字型上，如本系統所採用的「標楷體」中，有些筆劃與正常的書寫習慣不同，因此在筆劃的應用上需將不同的筆劃合併的處理，如圖 3-4 所示。例如：「乃」字的第一個筆劃（藍色標示）和第二個筆劃（紅色標示）在一般書寫習慣會是同一筆劃，但字型筆劃造字時是分成兩部分，所以在定義筆劃資料庫時，必須合併筆劃以符合一般的書寫習慣。

圖 3-4 與書寫習慣不同的造字筆劃

合併筆劃是以筆劃所佔空間大小、直線及曲線個數、點個數…等來分析判斷前後筆劃是否需要合併輸出。例如以筆劃「所佔空間的大小」為例，以下為標楷體字型筆劃判斷是否要合併的判斷方式。而實際的合併必須對輸出的筆劃進行各種組合的觀察，不同的字型檔案可能有不同的組合。

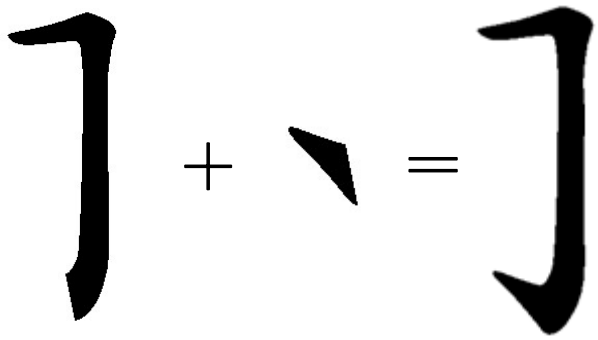
（1）88 前面筆劃空間大小為 368,468

（2）108 前面筆劃空間大小為 388,408,428,548,628,648,768

（3）128 前面筆劃空間大小為 508,588,628,648

（4）148 前面筆劃空間大小為 380,408,428,448,468,608,648,668

以「永」字為例，第二筆及第三筆必須進行合併的作業，屬於上述組合中的第（4）項 148 前面的筆劃空間為 380 種類型。筆劃資訊的合併方式採用❻❼❽設定為合併筆劃中第一筆的資訊，其他❶❷❸❹❺項目則為相加的方式，如圖 3-5 所示。



合併前筆劃資訊	合併後筆劃資訊
380,2,17,2,34,707,永,2 148,1,6,1,12,487,永,3	**528,3,23,3,46,707,永,2**

圖 3-5「永」第二筆及第三筆劃合併作業

3-1-2 字型筆劃細線化處理

細線化處理的步驟是將取得的字型筆劃特徵進行細線化處理，除了可以減少筆劃資料庫的資料量外，主要是作為和使用者手寫輸入筆劃細線化結果的比對依據；另外，也提供螢幕細線字型輸出模式的用途。研究上所採用的細線化的方法為本文 2-3-3 中所述之 Zhang-Suen 細線化演算法。如圖 3-6 為「永」字的第一個筆劃「點」細線化示意圖；圖 3-7 為「願」全字型筆劃細線化處理結果。

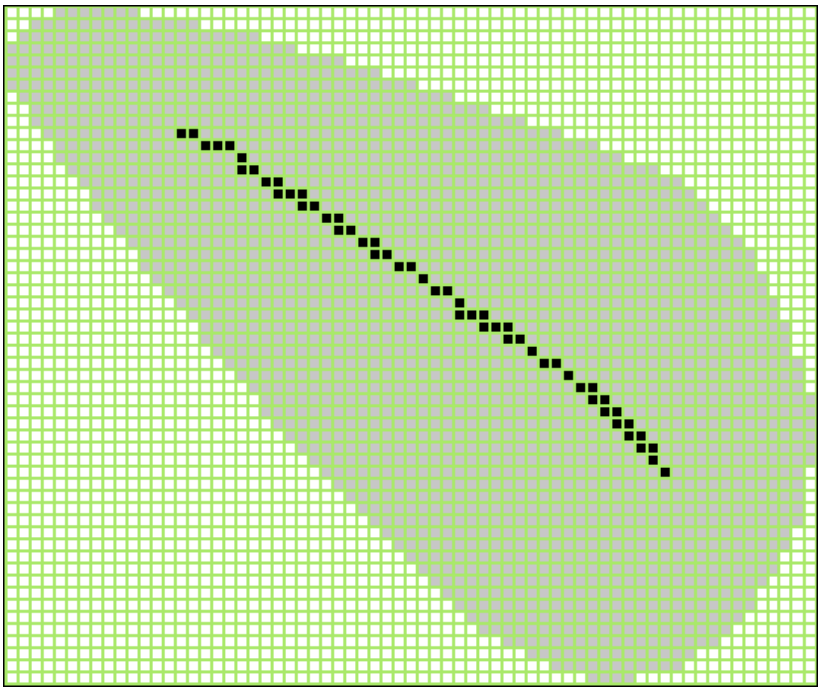


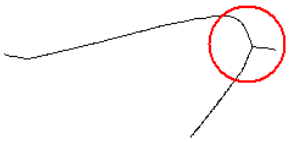
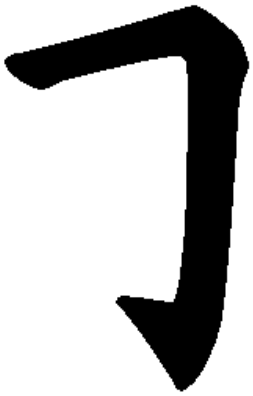
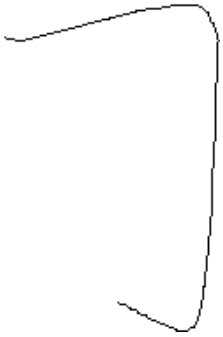
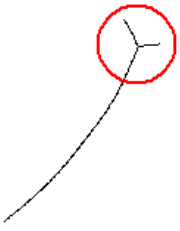
圖 3-6 「永」字的第一個筆劃「點」細線化示意圖

(a) 原字型　　(b) 細線化結果

圖 3-7 字型筆劃細線化處理

在細線化過程中，發現有些筆劃轉折點會形成一些多餘的線條骨架的情形，因此需透過優化細線化的方法加以改善。研究上採用的方法是依據本文 2-3-4 所述之特徵擷取方法用以區別特徵的型式。由筆劃端點沿著路徑偵測每一個像素點，一般中間連結點只有兩個相鄰點，若有兩個以上的相鄰點則為分岔點。尋找路徑的長度的結束判斷即以相鄰點超過兩個的為結束點，形成的路徑長度會比分岔點的長度較長，留下最長的兩個即是筆劃端點。找出筆劃端點後，再將過程中多餘的線條骨架消除，如表 3-2 所示。

表 3-2 字型筆劃細線化處理

原來的筆劃	不良細線化	優化細線化
		
		
		

3-1-3 筆劃特徵分析歸納

特徵為字型筆劃辨識時最為關鍵的要素。在中國書法字型中，不同之筆劃在結構上可能具備相同之結構特徵，因此在辨認流程上，通常會因應辨識目標不同而採取幾種不同的特徵擷取配合。一般常用的特徵值擷取與編碼方法已在第二章文獻中加以探討。

研究上使用的字型為 Windows XP 中文系統中之標楷體，使用此字型的原因首先為該字型為筆劃型造字方式較適合本系統的實作；其次為標楷體與一般人書寫習慣的字型較為接近。在中文系統 BIG-5 編碼中，有 5,401 個常用字，7,652 個次常用字，441 個特殊符號，而每個字元的筆劃平均約有 10 筆以上，所以全部字型中的筆劃共十幾萬筆資料，例如：標楷體字型共有 23,230 個字元及 178,196 個筆劃。如果將所有的筆劃納入筆劃資料庫中，在後續筆劃辨識搜尋時需耗費大量的時間，不符效益；因此必須將重覆的筆劃予以刪除，減少比對的次數，以增加系統運作效率。而精簡的方法是藉由分析字型中共通的筆劃結構，將不同的字元中具有相同結構的筆劃刪除。

研究上中利用 GetGlyphOutline 函數讀取指定字元的字型資料後，再依下列幾個字型筆劃資料的項目加以分析及判斷，作為刪減筆劃的依據：

1. 所佔用空間大小

因為標楷體造字是採用筆劃型的造字，結構相同的筆劃只是經過放大縮小形成，所以相同結構的筆劃使用的資料空間大小一樣。例如：「永」的第一筆、「汁」和「汀」的第一筆及第二筆的筆劃空間都是 176，同屬於「、」結構；但除了空間大小外，尚須配合其他的判斷項目，如 xy 角度，才決定該筆劃是否加以刪除，如圖 3-8 所示。

圖 3-8 資料空間大小相同的筆劃

2. 直線和曲線個數及點數

每一筆劃均可能包含數個直線或是曲線，依照每個筆劃的直線和曲線的個數及點數可以區別不同結構的筆劃。例如：「刀」、「刁」、「力」的第一筆的筆劃直線和曲線個數及點數相同，是屬於相同的結構，如圖 3-9 所示。

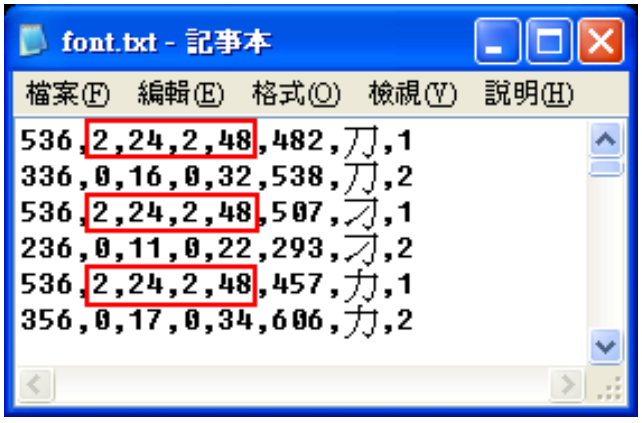
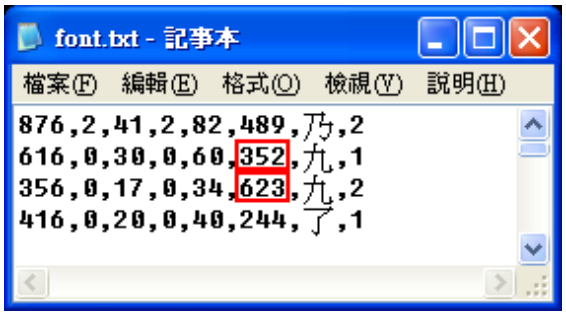


圖 3-9 直線和曲線個數及點數相同的筆劃

3. xy 角度

依據筆劃寬高形成的夾角亦可用來判斷筆劃的結構，如圖 3-10 中，「九」的第一筆為 35.2 度，第二筆為 62.3 度，書法字型中相同筆劃不同寬高的筆劃可以用此來區別。



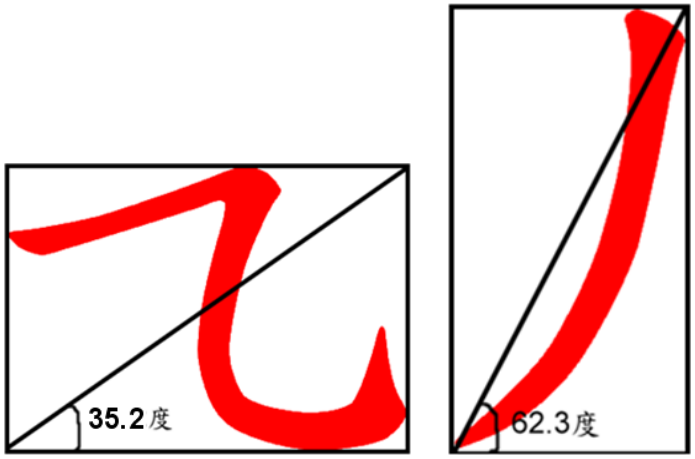
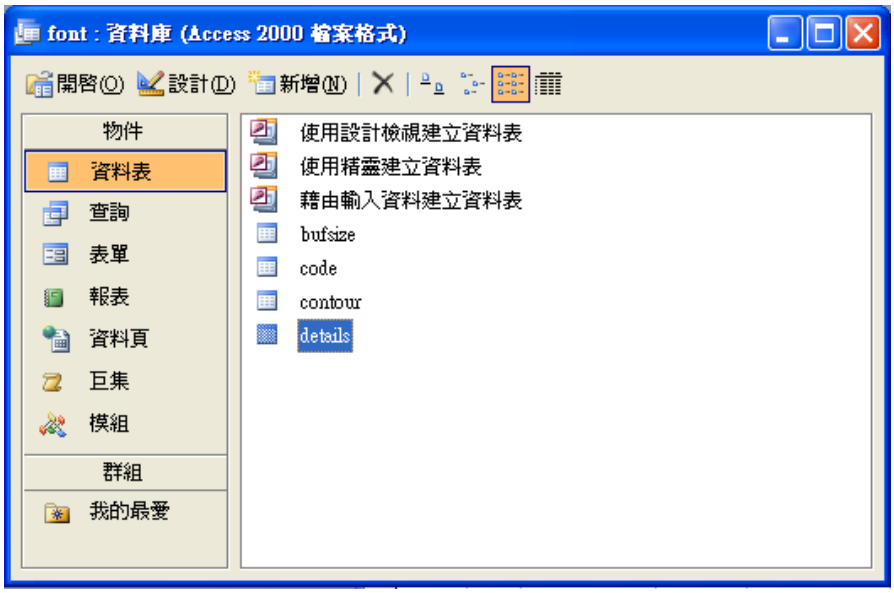
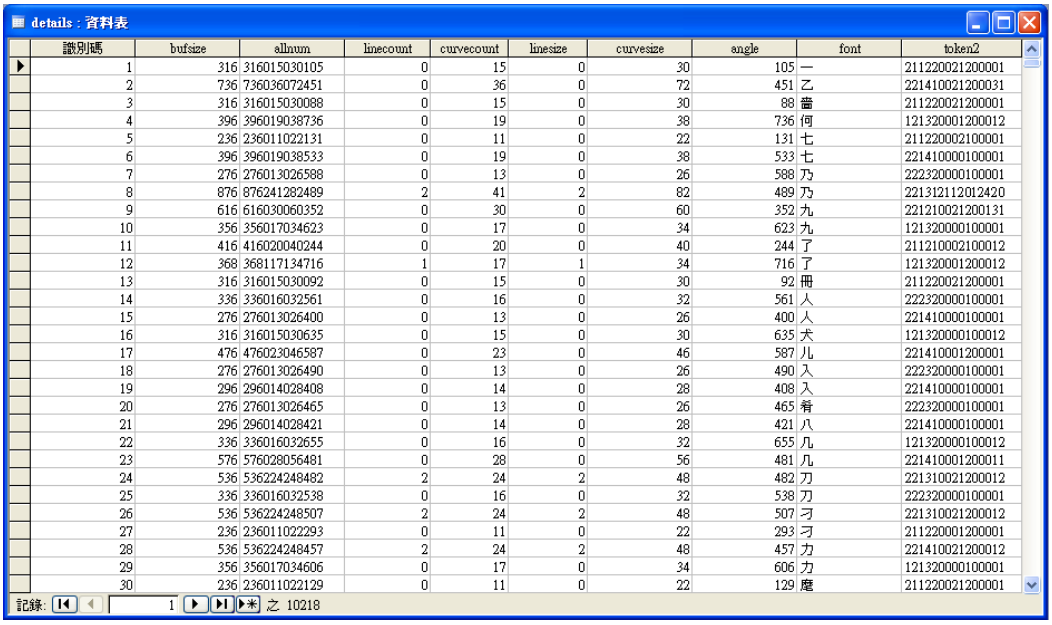


圖 3-10 「九」筆劃的 xy 角度

根據以上三個項目將從字型檔案中擷取的筆劃相同的部分刪除。若有三項數值皆相同，則取寬高較大者為代表筆劃，因為較小的筆劃在擷取筆劃特徵時比較容易有誤差。經過分析並刪減後結果最後剩餘筆劃數為 10,218 筆劃資料，而筆劃特徵資料庫使用 Microsoft Access 的格式加以儲存，如圖 3-11 所示。





details : 資料表

識別碼	bufsize	allnum	linecount	curvecount	linesize	curvesize	angle	font	token2
1	316	316015030105	0	15	0	30	105	一	211220021200001
2	736	736036072451	0	36	0	72	451	乙	221410021200031
3	316	316015030088	0	15	0	30	88	畫	211220021200001
4	396	396019038736	0	19	0	38	736	何	121320001200012
5	236	236011022131	0	11	0	22	131	七	211220002100001
6	396	396019038533	0	19	0	38	533	七	221410000100001
7	276	276013026588	0	13	0	26	588	乃	222320000100001
8	876	876241282489	2	41	2	82	489	乃	221312112012420
9	616	616030060352	0	30	0	60	352	九	221210021200131
10	356	356017034623	0	17	0	34	623	九	121320000100001
11	416	416020040244	0	20	0	40	244	了	211210002100012
12	368	368117134716	1	17	1	34	716	了	121320001200012
13	316	316015030092	0	15	0	30	92	冊	211220021200001
14	336	336016032561	0	16	0	32	561	人	222320000100001
15	276	276013026400	0	13	0	26	400	人	221410000100001
16	316	316015030635	0	15	0	30	635	犬	121320000100012
17	476	476023046587	0	23	0	46	587	儿	221410001200001
18	276	276013026490	0	13	0	26	490	入	222320000100001
19	296	296014028408	0	14	0	28	408	入	221410000100001
20	276	276013026465	0	13	0	26	465	脊	222320000100001
21	296	296014028421	0	14	0	28	421	八	221410000100001
22	336	336016032655	0	16	0	32	655	几	121320000100012
23	576	576028056481	0	28	0	56	481	几	221410001200011
24	536	536224248482	2	24	2	48	482	刀	221310021200012
25	336	336016032538	0	16	0	32	538	刀	222320000100001
26	536	536224248507	2	24	2	48	507	刁	221310021200012
27	236	236011022293	0	11	0	22	293	刁	211220001200001
28	536	536224248457	2	24	2	48	457	力	221410021200012
29	356	356017034606	0	17	0	34	606	力	121320000100001
30	236	236011022129	0	11	0	22	129	麼	211220021200001

記錄: 1 之 10218

圖 3-11 筆劃資料庫

筆劃資料經刪減及細線化處理後，接著針對筆劃特徵值加以編碼並計算出筆劃分佈特徵值。本研究編碼的方式是將每一個筆劃特徵以「15」位數加以編碼，以作為筆劃在資料庫中的索引值，在資料庫中的欄位名稱設定為「token2」，編碼方式分成四個項目，如表 3-3 所示；而字型筆劃分佈特徵值，主要作為用來與手寫筆劃分佈特徵值的比對依據，亦分成四個項目，如表 3-4 所示，分述如下：

表 3-3 筆劃特徵索引值編碼方式

項目		位數
1. 筆劃長寬比		2
2. 筆劃頂點位置		2
3. 頂點間夾角		1
4. 筆劃分佈值	(1) 在 y 軸向 x 軸方向掃描，符合條件分割通過的線段數序列	5
	(2) 在 x 軸向 y 軸方向掃描，符合條件分割通過的線段數序列	5

表 3-4 筆劃分佈特徵值

項目	欄位名稱
1. 在 x 軸向 y 軸方向掃描，通過不同線段區域百分比	bnx0～bnx4
2. 在 y 軸向 x 軸方向掃描，通過不同線段區域百分比	bny0～bny4
3. 在 x 軸向 y 軸方向掃描，符合條件分割百分比	cx0～cx4
4. 在 y 軸向 x 軸方向掃描，符合條件分割百分比	cy0～cy4

1. 筆劃的長寬比

依筆劃的長寬比分成三類：2x2、1x2、2x1 三種，如圖 3-12 所示，編碼依序為「22、12、21」，如圖 3-13 中「乃」字的第二個筆劃的長寬比分類編碼為第一類「22」。

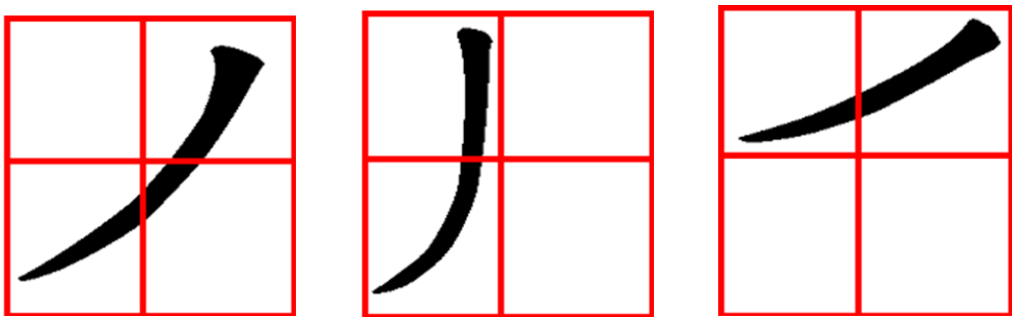


圖 3-12 筆劃的長寬比分類：依序為「22、12、21」

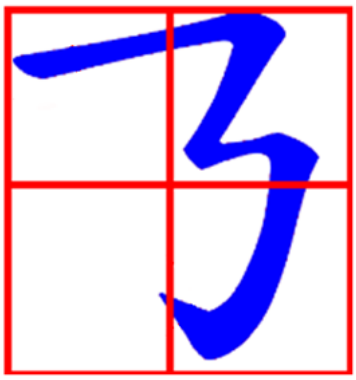


圖 3-13 「乃」字的第二個筆劃之長寬比編碼值「22」

2. 筆劃的的頂點位置

將筆劃細線化的結果依寬高分成四等分，去除多餘的頂點，最後

會得到兩個頂點，依頂點分佈的區域分類以兩個數字值編碼，如圖 3-14 所示，頂點分佈編碼分別為「23、13、12」，以圖 3-15 中「乃」字的第二個筆劃為例，頂點位置的編碼值為「13」。。

1	2
3	4

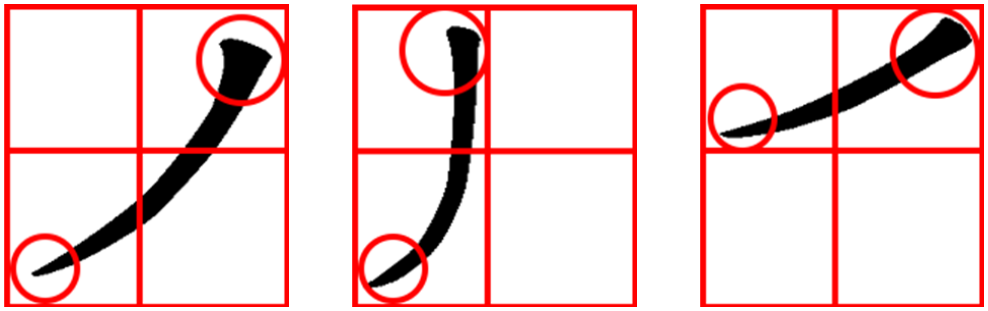


圖 3-14 筆劃的頂點位置：依序為「23、13、12」

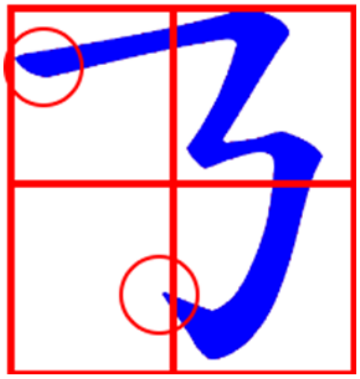


圖 3-15 「乃」字的第二個筆劃之頂點位置編碼值「13」

3. 頂點間的夾角

根據細線化得到的兩頂點的夾角，分成兩類，分別用「1」和「2」編碼，假設兩頂點座標為 (x1,y1)、(x2,y2)，如果 (x2-x1)*(y2-y1) < 0，則編碼值為「2」，否則為「1」，垂直或水平筆劃，編碼值為「1」。如圖 3-16 所示，「乃」字的第一筆劃座標為 (215,132) 及 (85,379)，因 (85-215)*(379-132) < 0，編碼值為「2」；第二筆劃座標為 (113,119) 及 (251,351)，(251-113)*(351-119) > 0，編碼值為「1」。

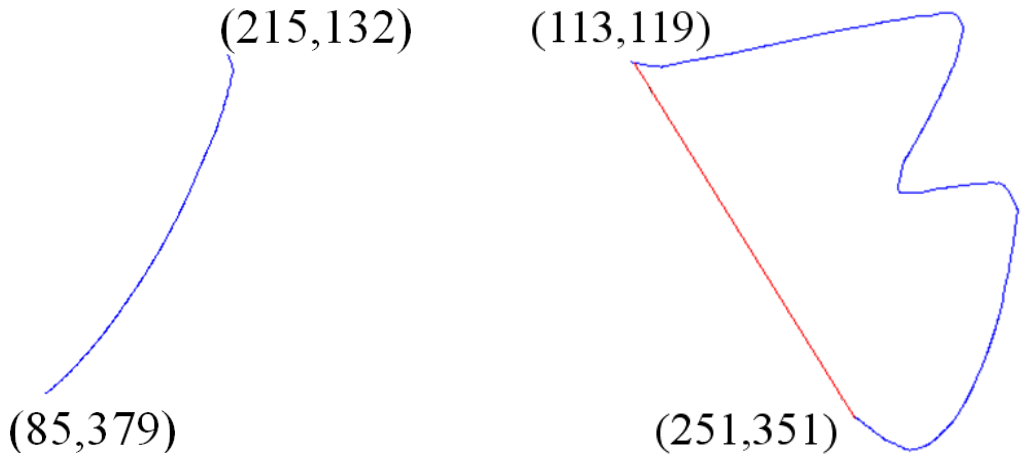


圖 3-16 筆劃的兩頂點間的夾角分類

4. 筆劃的分佈特徵

系統平台所使用的座標系統為電腦的座標系統，其表示方式如圖 3-17 所示，系統中字型最大點數設定為「480 x 480」點。

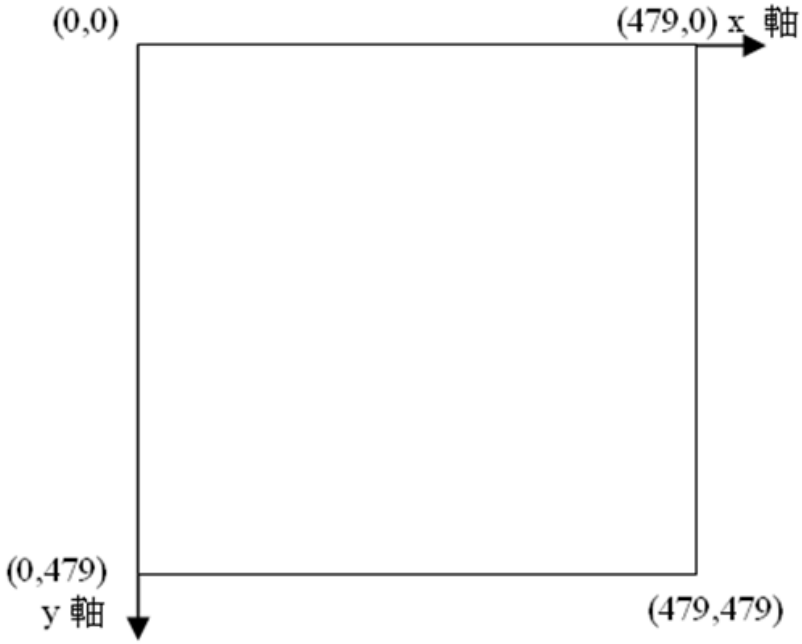


圖 3-17 座標系統

依據筆劃細線化的結果，在 x 軸方向及 y 軸方向進行掃描計算取出筆劃分佈特徵值。以下將以「乃」字的第二個筆劃為例，分別說明如下：

（1）在 x 軸向 y 軸方向掃描，通過不同線段區域百分比：

如圖 3-18 所示，分別定義各區域如下，其中「bnx0」及「bnx4」為必要項目，若分割區域超過 5 個區域，則中間部份只取出所佔百分比較大的 3 個部份，依序為「bnx1」、「bnx2」、「bnx3」。「乃」字的第二個筆劃，經計算結果 (bnx0,bnx1,bnx2,bnx3,bnx4) = (11,22,3,9,53)。

bnx0：向 y 軸方向掃描最初的空白區域的點數佔全部圖形的百分比，如圖中「粉紅色」部分。

bnx1：第一組藍點和第二組藍點中間的區域的點數佔全部圖形的百分比，如圖中「綠色」部份。

bnx2：第二組藍點和第三組藍點中間的區域的點數佔全部圖形的百分比，如圖中「藍色」部份。

bnx3：第三組藍點和第四組藍點中間的區域的點數佔全部圖形的百分比，如圖中「橙色」部份。

bnx4：向 y 軸方向掃描最後的空白區域的點數佔全部圖形的百分比，如圖中「灰色」部分。

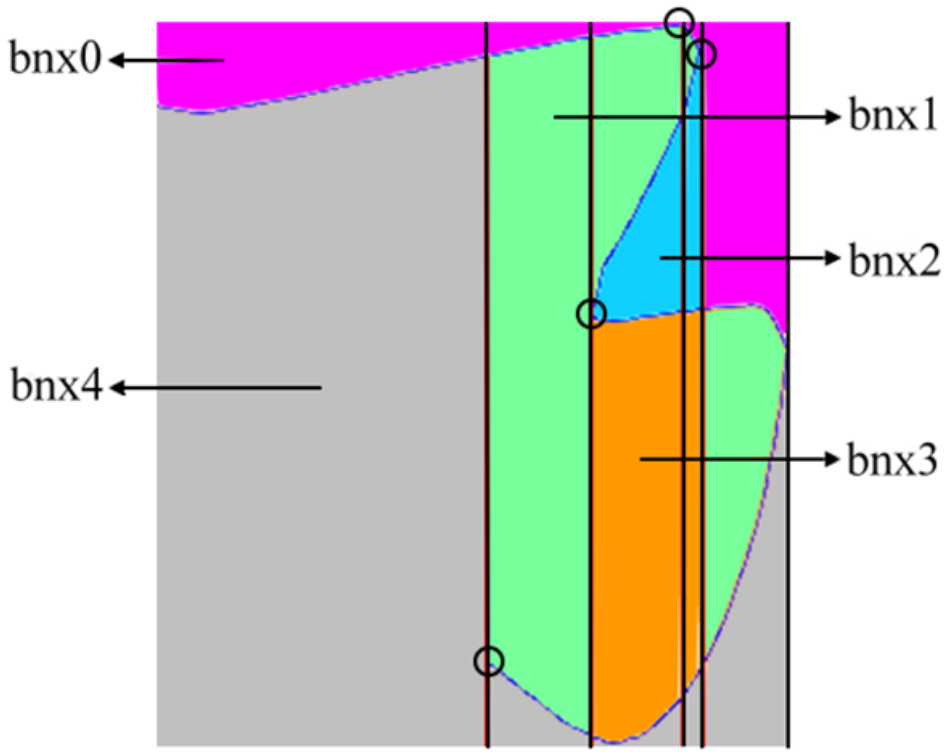


圖 3-18 在 x 軸向 y 軸方向掃描，通過不同線段區域百分比

（2）在 y 軸向 x 軸方向掃描，通過不同線段區域百分比：

如圖 3-19 所示，分別定義各區域如下： 其中「bny0」及「bny4」為必要項目，若分割區域超過 5 個區域，則中間部份只取出所佔百分比較大的 3 個部份，依序為「bny1」、「bny2」、「bny3」。「乃」字的第二個筆劃，經計算結果 (bny0,bny1,bny2,bny3,bny4) = (78,7,1,0,13)。

bny0：向 x 軸方向掃描最初的空白區域的點數佔全部圖形的百分比，如圖中「粉紅色」部分。

bny1：第一組藍點和第二組藍點中間的區域的點數佔全部圖形的百分比，如圖中「綠色」部份。

bny2：第二組藍點和第三組藍點中間的區域的點數佔全部圖形的百分比，如圖中「橙色」部份。

bny3：第三組藍點和第四組藍點中間的區域的點數佔全部圖形的百分比（在本例未含此組分類）。

bny4：向 x 軸方向掃描最後的空白區域的點數佔全部圖形的百分比，如圖中「灰色」部分。

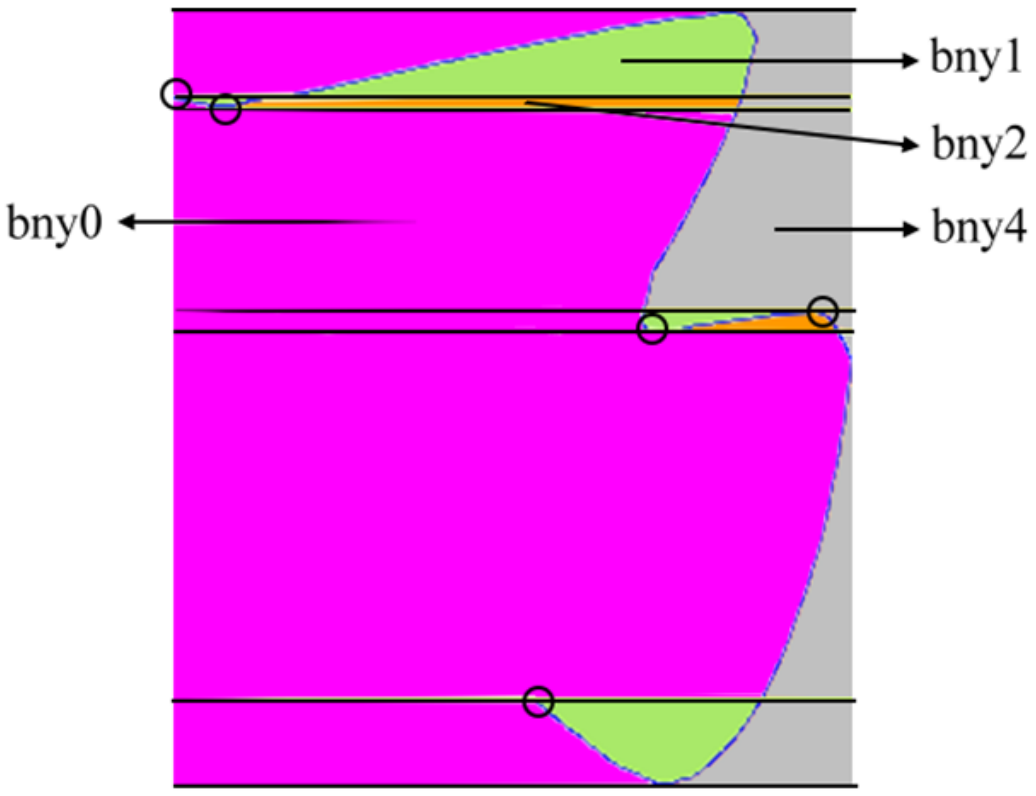


圖 3-19 在 y 軸向 x 軸方向掃描，通過不同線段區域百分比

（3）在 x 軸向 y 軸方向掃描，依通過的線段數量分割：

在 x 軸向 y 軸方向掃描，依通過的線段數量分割，分割的方式為通過相同的線段數量的區域為一分割。其中區域所佔百分比若小於筆劃寬度 1/16，則將其捨棄。若分割超過 5 個區域，則取佔百分比最大的 5 個區域，依序定義為「cx0~cx4」；相對應符合條件分割的通過線段數序列定義為「cnx0~cxn4」。例如：在圖 3-20 中的第 4 個區域（xx 標示處），由於比例小於 1/16，所以會被捨棄，最後只採用了 4 個符合條件的分割「cx0~cx3」；以下為「乃」字第二個筆劃所求得的計算值。

在 x 軸向 y 軸方向掃描，符合條件分割百分比：

(cx0, cx1, cx2, cx3, cx4) = (53, 16, 16, 14, 0)

在 x 軸向 y 軸方向掃描，符合條件分割通過的線段數序列：

(cnx0, cnx1, cnx2, cnx3, cnx4) = (1, 2, 4, 2, 0)

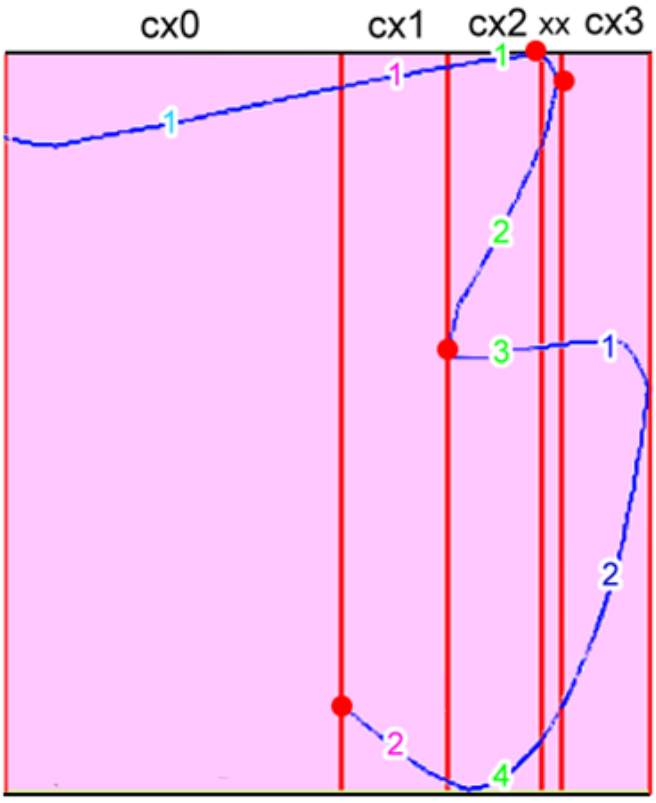


圖 3-20 在 x 軸向 y 軸方向掃描，依通過的線段數量分割

（4）在 y 軸方向向 x 軸方向掃描，依通過的線段數量分割：

在 y 軸向 x 軸方向掃描，依通過的線段數量分割，分割的方式為通過相同的線段數量的區域為一分割，其中區域所佔百分比若小於筆劃寬度 1/16，則將其捨棄。若分割超過 5 個區域，則取佔百分比最大的 5 個區域，依序定義為「cy0~cy4」；相對應符合條件分割的通過線段數序列定義為「cny0~cny4」。例如：在圖 3-21 中有二個區域比例小於 1/16 會被捨棄（xx 標示處），最後採用了 4 個符合條件的分割「cy0~cy3」；以下為「乃」字第二個筆劃所求得的計算值。

在 y 軸向 x 軸方向掃描，符合條件分割百分比：

(cy0, cy1, cy2, cy3, cy4) = (11, 26, 48, 11, 0)

在 y 軸向 x 軸方向掃描，符合條件分割通過的線段數序列：

(cny0, cny1, cny2, cny3, cny4) = (2, 1, 1, 2, 0)

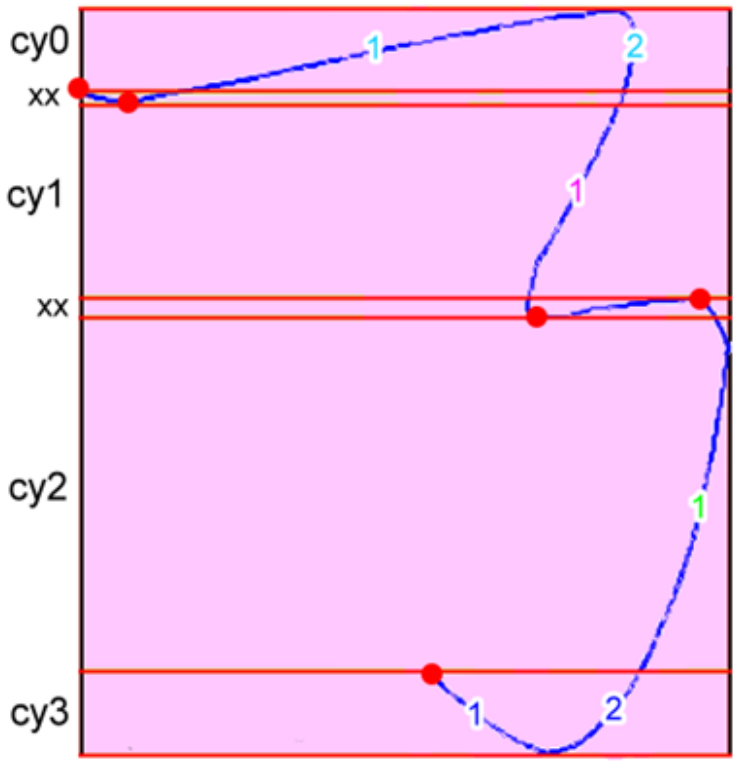


圖 3-21 在 y 軸向 x 軸方向掃描，依通過的線段數量分割

在筆劃資料庫中的各筆劃的特徵值，依上述的方式，透過程式自動擷取並分析，最後以下列四個部份加以編碼，並作為筆劃在資料庫中的索引值。如圖 3-22 中的「乃」字的第二筆劃。其編碼值為「221312112012420」。

1. 筆劃長寬比：22
2. 筆劃頂點位置：13
3. 頂點間夾角：1
4. 筆劃分佈值：

(1) 在 y 軸向 x 軸方向掃描，符合條件分割通過線段數序列(cny0~cny4)：21120

(2) 在 x 軸向 y 軸方向掃描，符合條件分割通過線段數序列(cnx0~cnx4)：12420

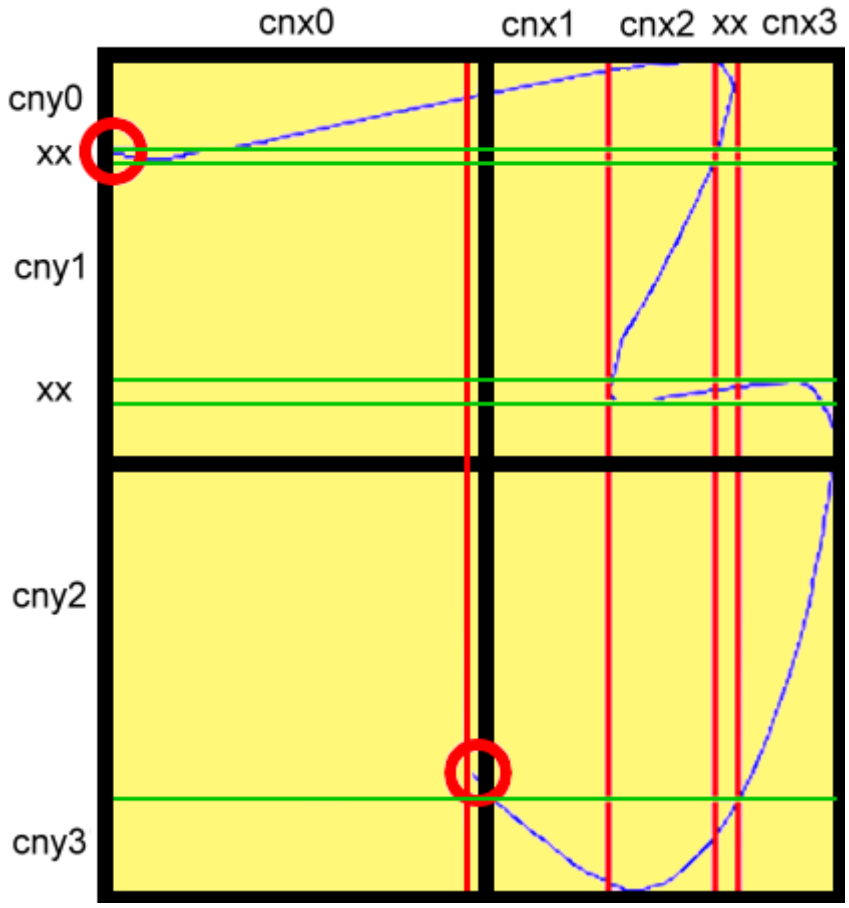


圖 3-22「乃」字第二個編碼值「221312112012420」

3-2 使用者操作介面

3-2-1 使用者筆劃輸入

在模擬系統平台中，設定使用者的操作方式是利用滑鼠或數位手寫板等輸入設備進行筆劃的書寫輸入；系統會自動針對每一個筆劃加以細線化處理，擷取其筆劃特徵，再與筆劃資料庫進行比對作業，並即時模擬輸出字型筆劃。

在操作上，在輸入筆劃時使用者可以自行設定「畫筆寬度」，預設值為「15 pt」，在輸入完筆劃系統會依輸入的筆劃找出最符合特徵值的筆劃。若不滿意可以選擇「取消筆劃」或「清除全部」，直到輸出筆劃符合需求。圖 3-23 為分別設定畫筆寬度為「1 pt」、「15 pt」及「35 pt」，輸入「永」字的畫面。另外系統提供將手寫筆劃儲存成 jpg 圖檔格式的功能，可讓使用者將系統模擬出的字型，匯入影像處理軟體中，進行後續的處理及應用。

圖 3-23 設定不同的輸入畫筆寬度

3-2-2 擷取輸入筆劃

使用者手寫輸入筆劃的方式，設定為一次將筆劃輸入完畢。系統平台會根據使用者的輸入設備的狀態，例如：滑鼠設備為「按下」和「放開」，作為擷取輸入筆劃的判斷標準。從使用者按下開始記錄，系統會記錄滑鼠移動的事件，記錄每一個移動的點，直到使用者放開滑鼠為止，將這些連續直線線段組合起來即是輸入筆劃。圖 3-24 為輸入「口」字的筆劃時三個筆劃的擷取。

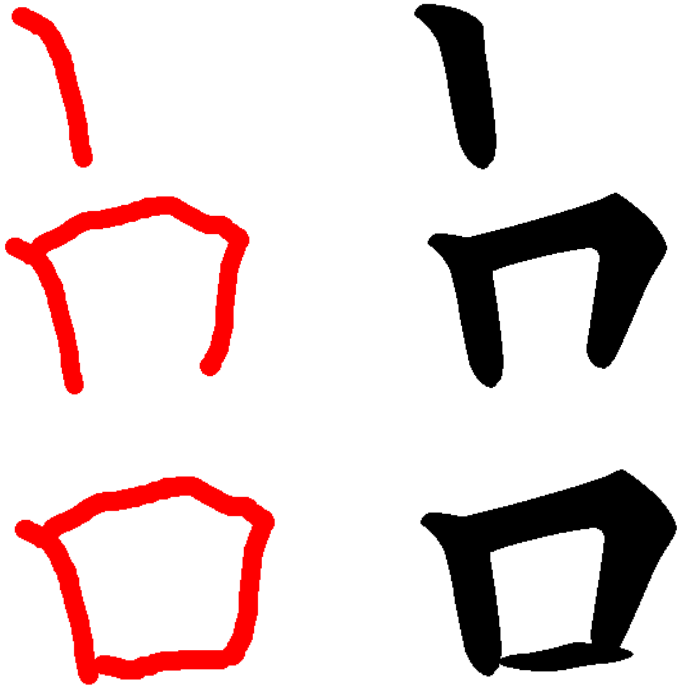
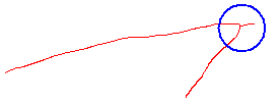
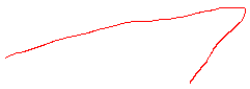
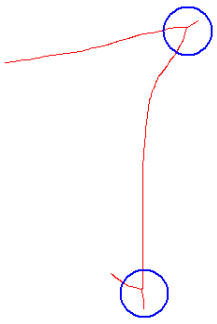
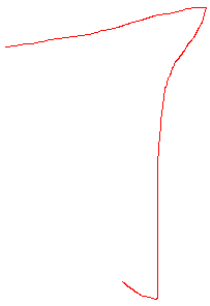


圖 3-24 輸入筆劃的擷取

3-2-3 輸入筆劃細線化處理

在輸入筆劃的細線化處理上和電腦字型筆劃一樣是採用 Zhang-Suen 細線化演算法，細線化的結果可以有效降低儲存空間，有助於影像處理及筆劃比對速度的提昇。對於輸入筆劃細線化處理過程中，也同樣會發生一些多餘的線條骨架的情形，因此仍需透過優化細線化的方法加以改善，如表 3-5 所示；圖 3-25(a)(b)為設定畫筆寬度為「30 pt」使用者輸入的筆劃及系統自動細線化的結果。輸入筆劃細線化是用於系統程式內部控制用途，因此使用者於操作過程並不會看到輸入筆劃細線化的結果。

表 3-5 輸入筆劃細線化處理

原來的筆劃	不良細線化	優化細線化
		
		
		

(a) 原輸入筆劃字型　　(b) 細線化結果

圖 3-25 輸入筆劃細線化處理

3-2-4 筆劃特徵擷取

手寫輸入筆劃特徵的擷取及編碼方式，如 3-1-3 所述之方法，將擷取特徵值主要分成下列兩個項目，以作為與筆劃特徵資料庫的字型筆劃資料比對的依據。

1. **索引值**：若索引值為「221312112012420」，表示：

（1）筆劃長寬比：22
（2）筆劃頂點位置：13
（3）頂點間夾角：1
（4）筆劃分佈值：

- 在 y 軸向 x 軸方向掃描，符合條件分割通過線段數序列：21120
- 在 x 軸向 y 軸方向掃描，符合條件分割通過線段數序列：12420

2. **筆劃分佈特徵值**：主要包含下列四個項目。

（1）在 x 軸向 y 軸方向掃描，通過不同線段區域百分比
（2）在 y 軸向 x 軸方向掃描，通過不同線段區域百分比
（3）在 x 軸向 y 軸方向掃描，符合條件分割百分比
（4）在 y 軸向 x 軸方向掃描，符合條件分割百分比

3-2-5 比對筆劃資料

每當使用者手寫輸入一個筆劃後，系統便會自動擷取筆劃、細線化處理，並進行特徵擷取與分析編碼及筆劃的比對作業。首先，將輸入筆劃的索引值與筆劃資料庫比對，若有相同者表示有相同結構的筆劃；接著將輸入筆劃的每一項分佈資料與筆劃特徵資料庫中的分佈值相減，取其絕對值再相加，數值最小者表示誤差值最小，即為最符合的筆劃。

假設輸入筆劃的索引值為 bindex，其筆劃分佈值的變數分別為：

（1）在 x 軸向 y 軸方向掃描，通過不同線段區域百分比：
dbnx0～dbnx4

（2）在 y 軸向 x 軸方向掃描，通過不同線段區域百分比：
dbny0～dbny4

（3）在 x 軸向 y 軸方向掃描，符合條件分割百分比：
dcx0～dcx4

（4）在 y 軸向 x 軸方向掃描，符合條件分割百分比：
dcy0～dcy4

在筆劃資料庫中，字型筆劃的索引值欄位名稱設定為「token2」，其相對應筆劃分佈的欄位名稱如下，比對流程圖及程式碼，如圖 3-26 所示：

（1）bnx0～bnx4

（2）bny0～bny4

（3）cx0～cx4

（4）cy0～cy4

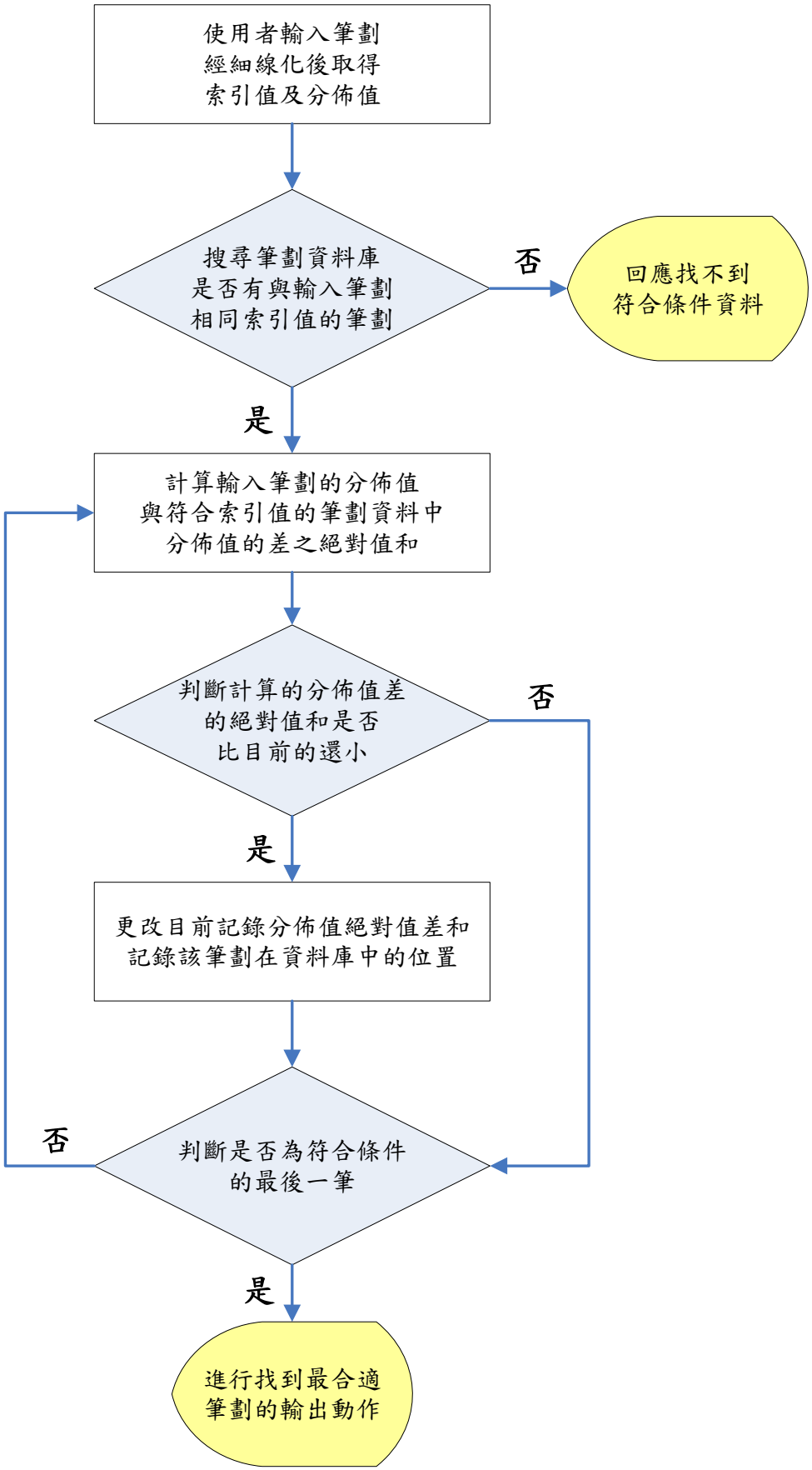


圖 3-26 輸入筆劃與筆劃資料庫比對流程圖

輸入比劃與筆劃資料庫比對程式碼片段

```
t:= 'Select * from details where token2 = '"+ bindex + '"';
   // token2 爲資料庫索引值欄位名稱，bindex 爲輸入筆劃的索引值
   ADOQuery1.Close;
   ADOQuery1.SQL.Clear;
   ADOQuery1.SQL.Add(t);
   ADOQuery1.Open;
   // 查詢是否有符合輸入筆劃索引值的資料
   dd := 999999;
   // 記錄目前符合輸入筆劃索引值的資料經與輸入筆劃分佈值比對的結果
   // 數值越小表示越符合，越大表示越不符合，若爲 999999 表示沒任何符合
   ADOQuery1.First;
   // 移到第一筆符合輸入筆劃索引值的位置，如果沒有任何符合的資料，則在
   // 下面 ADOQuery1.Eof 會成立，直接結束查詢
   while not ADOQuery1.Eof do
   begin
   // dd2 爲計算符合輸入筆劃索引值的資料與輸入筆劃分佈值的差值
   // 以下爲計算每一分佈值的差值，取差值的絕對值相加爲其結果
      dd2 :=   abs(dbnx[0]-StrToInt(ADOQuery1.FieldByName('bnx0').Text));
      dd2:= dd2 + abs(dbnx[1]-StrToInt(ADOQuery1.FieldByName('bnx1').Text));
      dd2:= dd2 + abs(dbnx[2]-StrToInt(ADOQuery1.FieldByName('bnx2').Text));
      dd2:= dd2 + abs(dbnx[3]-StrToInt(ADOQuery1.FieldByName('bnx3').Text));
      dd2:=dd2+abs(dbnx[4]-StrToInt(ADOQuery1.FieldByName('bnx4').Text));
      dd2:=dd2+abs(dbny[0]-StrToInt(ADOQuery1.FieldByName('bny0').Text));
      dd2:=dd2+abs(dbny[1]-StrToInt(ADOQuery1.FieldByName('bny1').Text));
      dd2:=dd2+abs(dbny[2]-StrToInt(ADOQuery1.FieldByName('bny2').Text));
      dd2:=dd2+abs(dbny[3]-StrToInt(ADOQuery1.FieldByName('bny3').Text));
      dd2:=dd2+abs(dbny[4]-StrToInt(ADOQuery1.FieldByName('bny4').Text));
      dd2:=dd2+abs(dcy[0]-StrToInt(ADOQuery1.FieldByName('cy0').Text));
      dd2:=dd2+abs(dcy[1]-StrToInt(ADOQuery1.FieldByName('cy1').Text));
      dd2:=dd2+abs(dcy[2]-StrToInt(ADOQuery1.FieldByName('cy2').Text));
      dd2:=dd2+abs(dcy[3]-StrToInt(ADOQuery1.FieldByName('cy3').Text));
      dd2:=dd2+abs(dcy[4]-StrToInt(ADOQuery1.FieldByName('cy4').Text));
      dd2:=dd2+abs(dcx[0]-StrToInt(ADOQuery1.FieldByName('cx0').Text));
      dd2:=dd2+abs(dcx[1]-StrToInt(ADOQuery1.FieldByName('cx1').Text));
      dd2:=dd2+abs(dcx[2]-StrToInt(ADOQuery1.FieldByName('cx2').Text));
      dd2:=dd2+abs(dcx[3]-StrToInt(ADOQuery1.FieldByName('cx3').Text));
      dd2:=dd2+abs(dcx[4]-StrToInt(ADOQuery1.FieldByName('cx4').Text));
   // 如果計算完 dd2 的結果小於 dd 表示有更符合的資料出現，將 dd 設定新值
      if (dd2<dd) then dd := dd2;
  // 繼續查詢下一筆符合輸入筆劃索引值的資料
      ADOQuery1.Next;
   end;
   // 如果比對完所有符合輸入筆劃索引值的資料後，dd 不等於 999999，表示有符
   // 合的資料，若 dd=999999 表示無任何一筆符合的資料
   if (dd<999999) then
      begin
         { 進行輸出筆劃的動作 }
      end
   else
         { 回應使用者沒發現符合條件的資料 }
```

3-2-6 字型筆劃輸出

本研究所建立的書法字型模擬平台，設計有兩種字型筆劃輸出模式提供使用者自由選擇：螢幕輸出模式及 SolidWorks 輸出模式，分述如下：

1. 螢幕輸出模式

研究上是利用 Win32 API 中的 GetGlyphOutline 函數讀取 TrueType 字型的輪廓資料。TTPOLYGONHEADER 結構中的 pfxStart 提供筆劃輪廓的起始點座標。而使用 TTPOLYCURVE 結構用來定義筆劃輪廓資料，可分為下列兩種類型，直線與曲線的描述，如表 3-6 所示：

（1）TT_PRIM_LINE 記錄一系列的點，點之間繪製描述字型的輪廓。

（2）TT_PRIM_QSPLINE 記錄一系列定義二次式 Quadratic 貝茲曲線的點。

表 3-6 TrueType 字型輪廓資料的描述方式

輪廓型式	描述方式
直線	以上一線段的終點為起點（最初的起點為 TTPOLYGONHEADER 中字義的筆劃輪廓起始點座標），再以 TTPOLYCURVE 中定義的一系列點（每一線段使用一個點）來形成一系列的直線。
曲線	以上一線段的終點為起點，再加上 TTPOLYCURVE 中定義的一系列點（每一曲線使用兩個點）共三個點，形成一系列的二次式 Quadratic 貝茲曲線。

在 TrueType 字型中，二次式 Quadratic 貝茲曲線被定義成三個點 A、B 和 C，點 A 和 C 在曲線上，點 B 為控制點，一般並不在曲線上面（在曲線上會形成直線）。其表示的公式如下，xA 表示點 A 的 x 座標，yA 表示點 A 的 y 座標，其餘類推：

x(t) = (xA-2xB+xC)*t^2 + (2xB-2xA)*t + xA

y(t) = (yA-2yB+yC)*t^2 + (2yB-2yA)*t + yA

其中 t 的範圍從 0.0 到 1.0

（1）A 點的座標最初是 TTPOLYGONHEADER 中定義的 pfxStart，或是 TTPOLYCURVE 中直線的終點或是二次式 Quadratic 貝茲曲線中的 C 點。

（2）B 點則是 TT_PRIM_QSPLINE 中曲線定義的目前的點，為曲線的控制點。

（3）C 點的表示有兩種情形：如果該筆記錄不是 TTPOLYCURVE 的最後一筆，則 C 點是 B 點和記錄中的下一個點之間的中點。否則，點 C 是 B 點後的一個點。

舉例說明，假設上一個線段的的終點為A，二次式Quadratic貝茲曲線定義的點為 ($Q_{11,}$ Q_{11} $Q_{21,}$ Q_{22} $Q_{31,}$ $Q_{32,}$……, $Q_{n1,}$ Q_{n2})，共定義了n個二次式Quadratic貝茲曲線，其中各線段表示二次式Quadratic貝茲曲線的點 (A,B,C) 分別為：

線段 1：(A, Q_{11}, (Q_{11}+Q_{12})/2)

線段 2：((Q_{11}+Q_{12})/2, Q_{21}, (Q_{21}+Q_{22})/2)

……

線段n：(($Q_{(n-1)1}$+$Q_{(n-1)2}$)/2, Q_{n1}, Q_{n2})

根據以上的公式，即可在程式中模擬筆劃在螢幕輸出或是直接利用 WIN32 API 提供的 PolyBezier 函數來進行輸出；在本研究中是採用 PolyBezier 函數進行筆劃在螢幕上的輸出。由於 PolyBezier 函數在繪製曲線時使用的定義為三次式 Cubic 貝茲曲線，所以必須將原來字

型檔中的二次式 Quadratic 貝茲曲線資料轉換為三次式 Cubic 貝茲曲線資料才能繪製，假設二次式 Quadratic 貝茲曲線定義的三個點為(A,B,C)，轉換後的三次式 Cubic 貝茲曲線的點為 (D,E,F,G)，其轉換公式如下。

$$D = A$$
$$E = A + 2*(B-A)/3$$
$$F = B + 2*(C-B)/3$$
$$G = C$$

除了一般實心模擬字型輸出外，系統另外提供字型輪廓外框字型及細線字型輸出選擇。其中外框字型的處理是針對整個實心字型影像逐點作運算，當上下左右都是黑點則將該像素點設定為白色；反之，若上下左右有不是黑點的像素點，表示其為外框邊緣點，則保留。以圖 3-27 參考運算單位單元為例，假設黑色像素點的值為 1，背景點（白點）的值為 0，中心點P_1的周圍參考點定義為P_2、P_3、P_4、P_5、P_6、P_7、P_8、P_9，當P_2、P_4、P_6、P_8中有「0」值時，則表示P_1為外框邊緣點；如圖 3-28 中的筆劃，先找出紅色標示外框像素點加以保留並刪除內部像素點後，即可得字型的外框；圖 3-29 為「王」字外框字型示意圖。

P_9	P_2	P_3
P_8	P_1	P_4
P_7	P_6	P_5

圖 3-27 外框字型參考運算單位單元

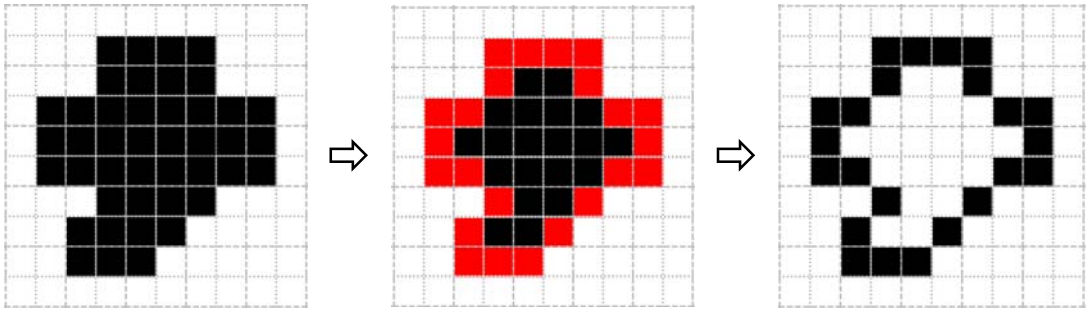


圖 3-28 外框字型的輸出處理

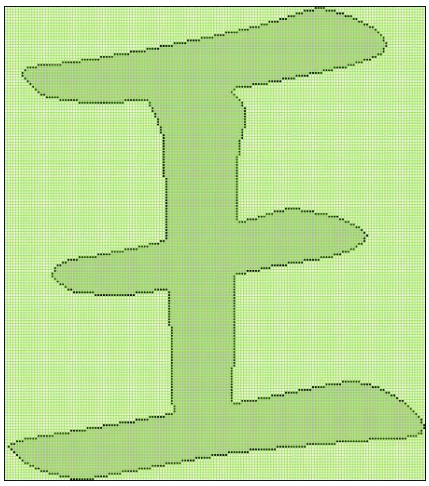
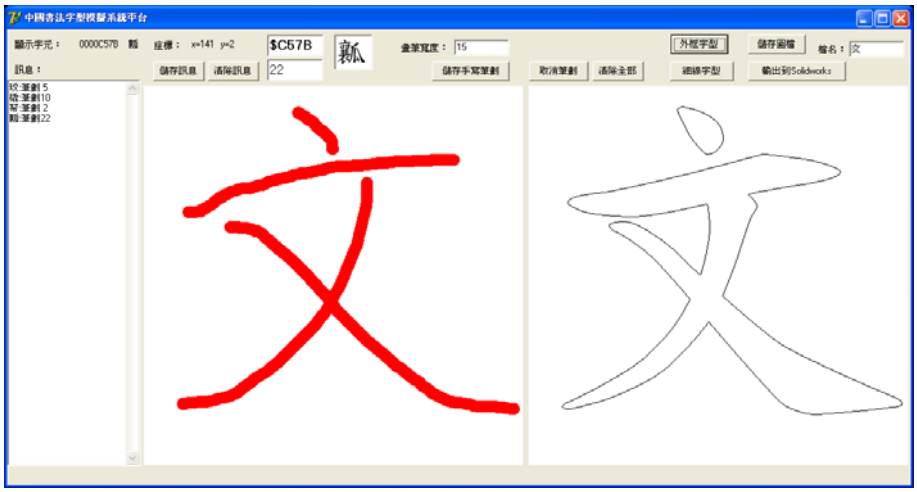


圖 3-29 「王」字外框字型示意圖

系統平台中，除了螢幕的輸出外，亦提供將輸出的字型儲存成 jpg 格式圖檔的功能，可讓使用者再匯入影像處理軟體中進行後續應用。圖 3-30 為模擬「文」字實心、外框及細線字型的螢幕的輸出畫面。



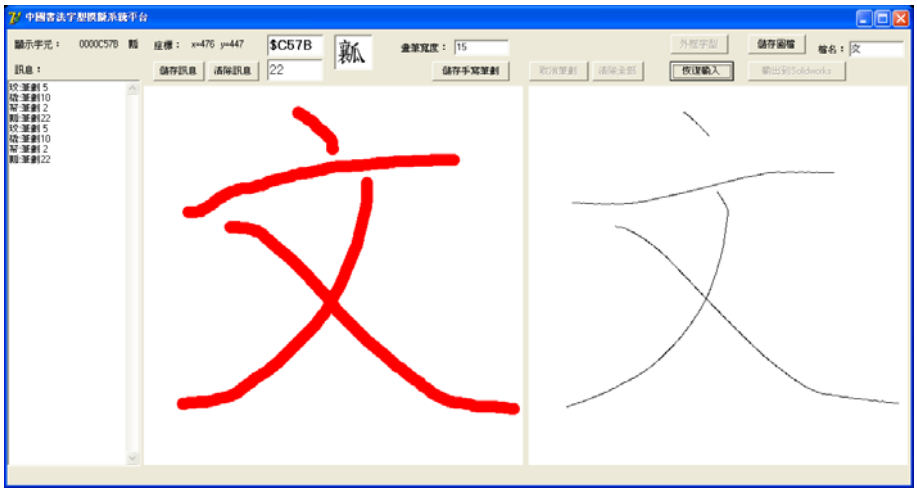


圖 3-30 模擬「文」字型的螢幕輸出畫面

2. SolidWorks 輸出模式

在本研究中所應用 SolidWorks API 方式是採用 DLL（Dynamic Link Library，動態連結函式庫），DLL 是具有函式的共用程式庫功能的可執行檔，提供一種方法讓處理程序（Process）呼叫不是可執行程式碼部分的函式；且多個應用程式可以同時存取記憶體中的 DLL 的內容，有助於資源與資料的共用。在實作上，必須先導入相關的類型庫，用以呼叫 SolidWorks 的對象、方法與屬性等。在 Delphi 中，導入類型庫的步驟如下所示：

（1）選取「Project／Import Type Library」命令，在彈出的如圖 3-31 所示的【Import Type Library】對話方塊中，對 SolidWorks 類型庫進行註冊，將類型庫文件 SldWorks.tlb 裝載到 Delphi 中。在清單中選取「SolidWorks 2010 Type Library (Version 12.0)」，按 Create Unit 按鈕，Delphi 會自動在其「Import」子資料夾中產生該類型庫的 Object Pascal 文件「Sldworks_TLB.Pas」，該文件包括 SolidWorks 提供的所有 API 函數。

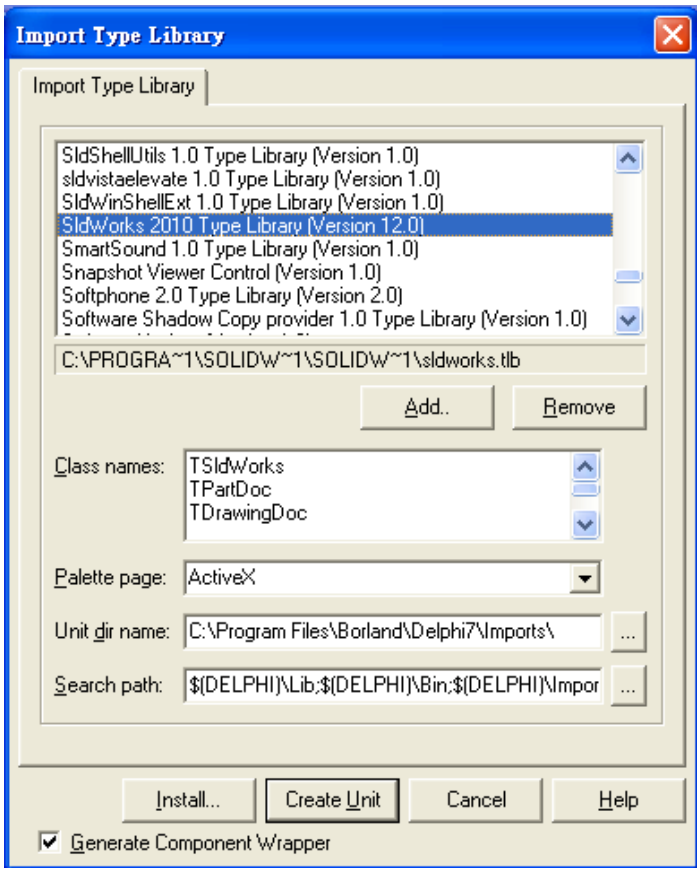


圖 3-31 在 Delphi 中加入 SolidWorks 類型庫對話方塊

（2）選取「Project／Add to Project」命令，在彈出的【Add to Project】對話方塊中，選擇 Delphi 的「Import」子資料夾中的 SolidWorks 類型庫單元文件「Sldworks_TLB.pas」，將其加入程式中，如圖 3-32 所示。

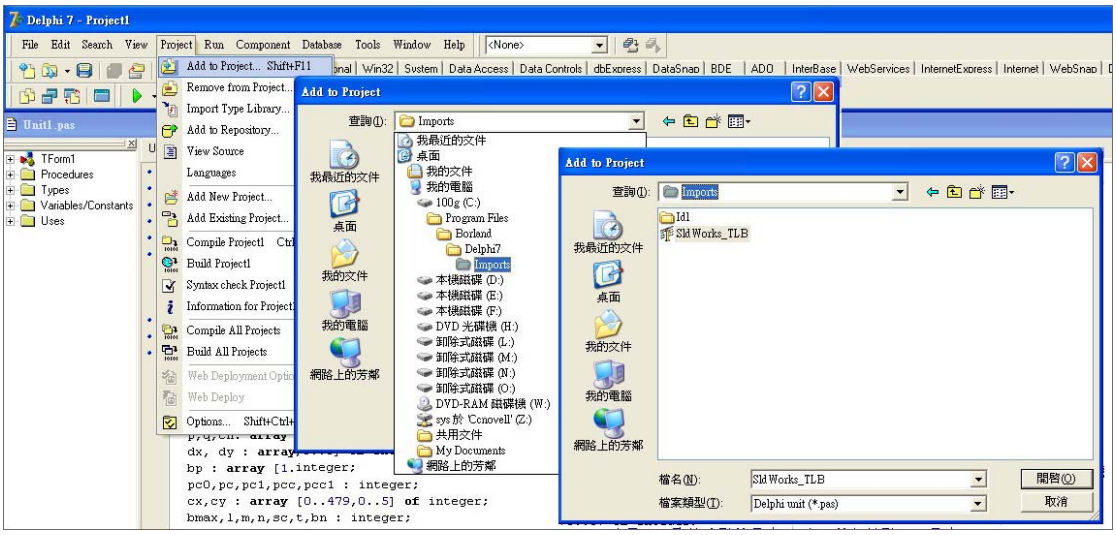


圖 3-32 在 Delphi 中加入 SolidWorks 類型庫單元文件

在 SolidWorks 中，若要產生貝茲曲線（在 SolidWorks 中，稱為不規則曲線），主要有下列二個函式：

（1）CreateSpline：CreateSpline 函數定義不規則曲線的方式為通過曲線上的點，是採用曲線擬合（Spline Fitting）的方式。

（2）CreateSplinesByEqnParams：定義不規則曲線的方式採用內插法（Spline Interpolation）。

TrueType 字型中所讀取的曲線資料為帶有一個控制點的二次式 Quadratic 貝茲曲線，所以在輸出到 SolidWorks 時，採用 CreateSplinesByEqnParams 函式較為適合。以下為在 SolidWorks 輸出二次式 Quadratic 貝茲曲線的程式碼片段，其產生的二次式 Quadratic 貝茲曲線如圖 3-33 所示。圖 3-34 (a) 為在系統平台中模擬「字」的字型畫面；而圖 3-34 (b) 為選擇將結果輸出到 SolidWorks 中後的結果畫面。模擬系統為字型預設加上「10 mm」伸長填料，使用者可依需要自行修改或作為後續應用與設計。

在 SolidWorks 輸出二次式 Quadratic 貝茲曲線的程式碼片段

```
swapp:=CoSldWorks_.Create;   // 產生 SolidWorks 物件
swapp.Visible :=True;
swdoc :=swapp.IActiveDoc2;
swdoc.Extension.SelectByID2('Front plane','PLANE',0,0,0,False,0,n,bres);
swdoc.ClearSelection2(true);
nPtdata :=VarArrayCreate([0,18],varDouble);
nPtData[0]:= 3;        // Dimension
nPtData[1]:= 3;        // Order，二次式 Quadratic 貝茲曲線
nPtData[2]:= 3;        // Number of control points，控制點個數
nPtData[3]:= 0;        // Periodicity
// Knots
nPtData[4] := 0;
nPtData[5] := 0;
nPtData[6] := 0;
```

```
nPtData[7] := 1;
nPtData[8] := 1;
nPtData[9] := 1;
// Control points
nPtData[10]:=-66.6/1000;    nPtData[11] := 0 /1000;    nPtData[12] := 0;
nPtData[13]:=-19./1000; nPtData[14]:=58.5 /1000;    nPtData[15] := 0;
nPtData[16]:=54.9/1000;    nPtData[17] := 0 /1000;    nPtData[18] := 0;
swdoc.InsertSketch2(True);
swdoc. CreateSplinesByEqnParams (nPtdata);    // 產生曲線
swdoc.InsertSketch2(True);
```

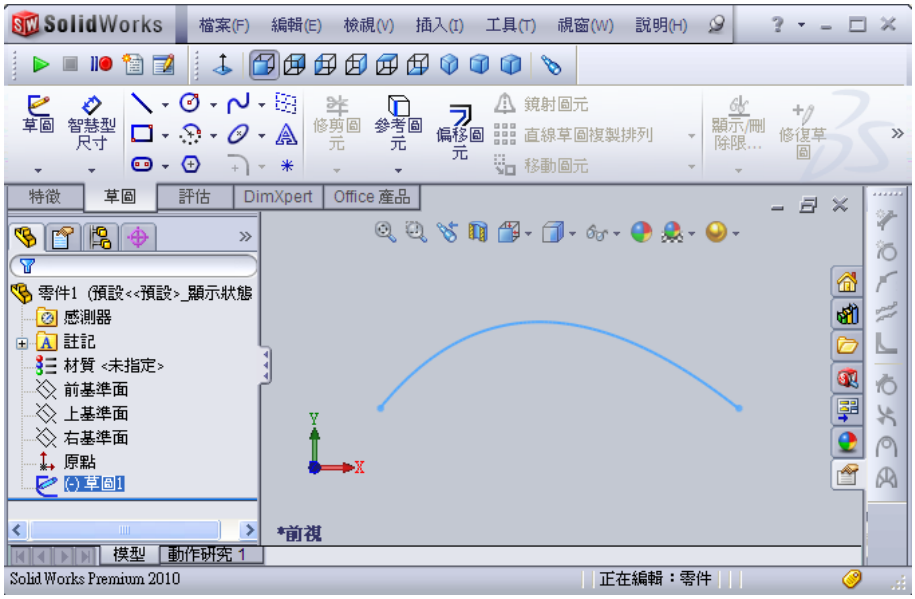
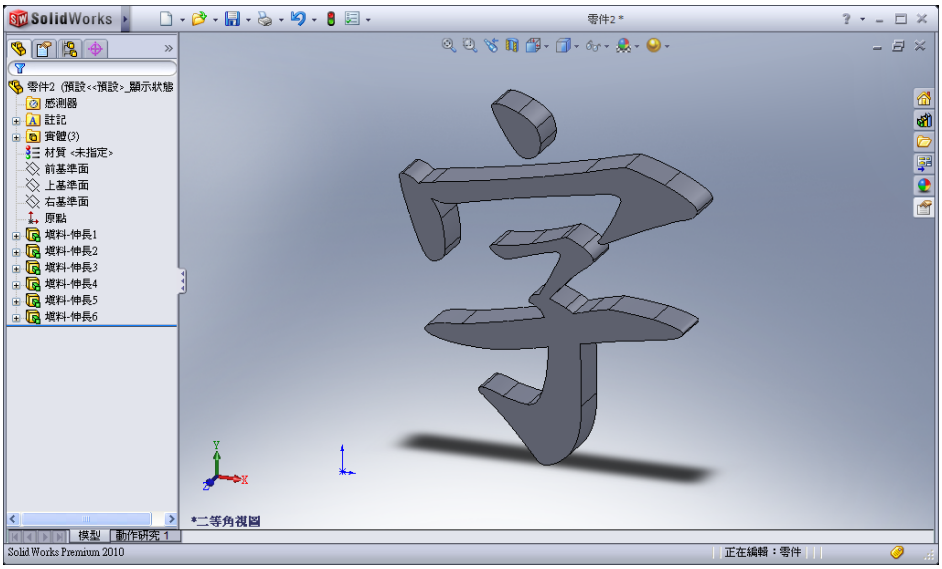


圖 3-33 在 SolidWorks 產生二次式 Quadratic 貝茲曲線

(a)



(b)

圖 3-34 輸出字型到 SolidWorks 中

第四章　實例操作驗證

本研究所開發的模擬平台，透過分析目前電腦字型檔案，建立筆劃特徵資料庫；接著擷取使用者輸入之筆劃，經過細線化和筆劃特徵比對的結果，選取最適合的筆劃進行字型建構並輸出。操作上，可由使用者自行判斷是否採用或刪除模擬出來的筆劃重新輸入。在字型輸出模式上，除了螢幕輸出外，並結合了 SolidWorks API 功能，讓字型可輸出至 SolidWorks 中，以作為字型後續設計、輸出應用。以下分別就系統平台操作介面、螢幕輸出模式及 SoildWorks 輸出模式，以實例操作驗證系統平台的可行性，並列舉模擬系統平台開發之困難與問題。

4-1 系統平台操作介面

使用者可由執行檔「Project.exe」開啟模擬系統平台視窗，如圖 4-1 所示，操作介面相關功能說明如下：

圖 4-1　系統平台操作介面

❶ 訊息區：用來顯示系統所擷取出的字型筆劃相關資料。

❷ 儲存訊息 清除訊息：用來儲存及清除訊息區中的資料，儲存格式為純文字檔*.txt 格式，主檔名為和字型圖檔檔名一樣。

❸ 手寫筆劃輸入區：使用者可利用滑鼠或數位筆…等輸入設備，在此區域手寫筆劃。

❹ 畫筆寬度：15：畫筆寬度設定區，使用者可依自己的喜好設定畫筆的寬度，系統預設值為「15 pt」。

❺ 儲存手寫筆劃：用來儲存使用者的手寫筆劃，儲存格式為 jpg 圖檔格式，主檔名為字型圖檔檔名後加入「user」。例如：字型圖檔若設定為「永」，則手寫筆劃的儲存檔名為「永 user.jpg」。

❻ 取消筆劃 清除全部：用來取消一筆或全部的筆劃，可允許使用者對於比對擷取結果不佳的筆劃進行重新輸入的作業。

❼ 字型筆劃輸出區：當使用者每輸入一筆筆劃，系統自動擷取筆劃後與筆劃資料庫中的字型筆劃比對後，輸出字型筆劃。

❽ 外框字型 細線字型 恢復輸入：系統除了提供實心字型輸出，另外提供外框字型及細線字型供使用者選擇，且提供恢復一般輸入模式機制。

❾ 儲存圖檔 檔名：15：使用者可將螢幕模輸出的字型加以儲存，儲存格式為 jpg 圖檔格式。其中外框字型及細線字型存檔時，主檔名為字型圖檔檔名後加入「ou」及「ln」。例如：字型圖檔若設定為「永」，則外框字型及細線字型儲存檔名為「永 ou.jpg」及「永 ln.jpg」。

⓾ 輸出到Solidworks：當使用者輸入完成時，可選擇輸出至 SolidWorks 中，所採用的版本為 SolidWorks 2010 SP0，當使用者啟動此功能時，系統會自動開啟 SolidWorks 軟體，並將字型模擬輸出。

4-2 螢幕輸出模式

在系統平台中是以電腦字型中的「標楷體」作為輸出模擬的字型，模擬過程中，如遇不滿意的筆劃，可利用取消筆劃功能並重新輸入，如圖 4-2 為「力」字的書寫模擬過程，其中在輸入第二筆時，產生的結果不佳，利用取消筆劃功能回到上一筆的狀態，再重新輸入。圖 4-3 為「永」字書寫模擬過程，輸出時除了實心字型，還可選擇輸出外框字型及細線字型；圖 4-4 為「永」字相關的儲存檔案資料，包括筆劃訊息、手寫筆劃、模擬字型、外框及細線字型。圖 4-5 則為「東海工設」四個字的全字模擬完成畫面。

圖 4-2 「力」字的書寫模擬過程

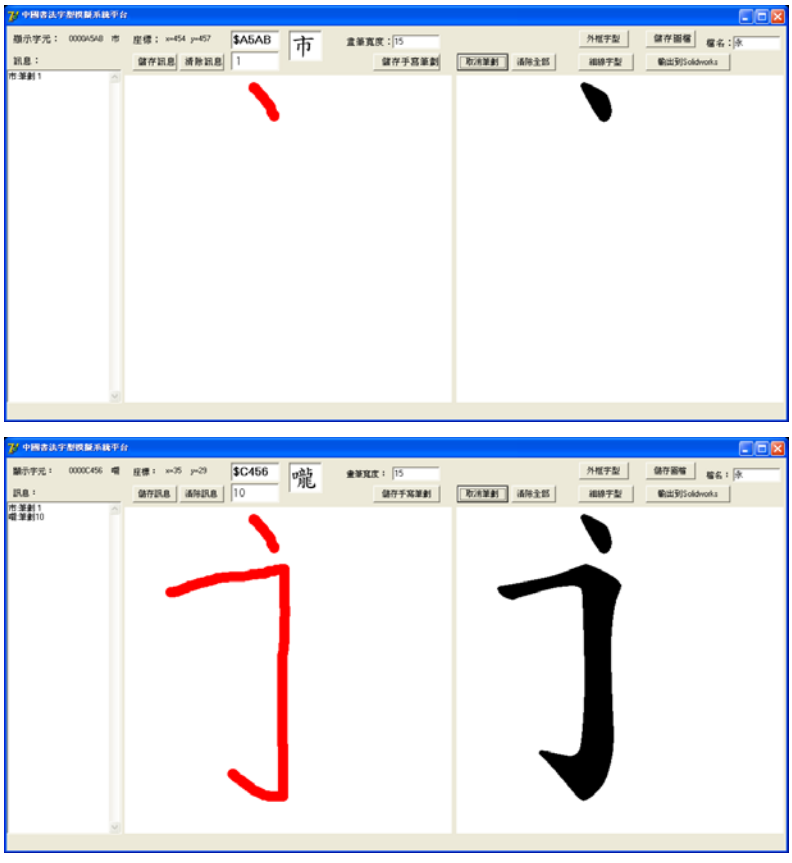
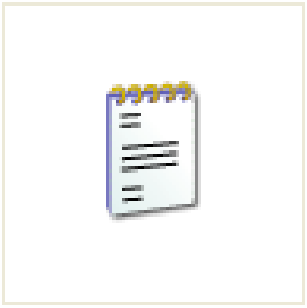
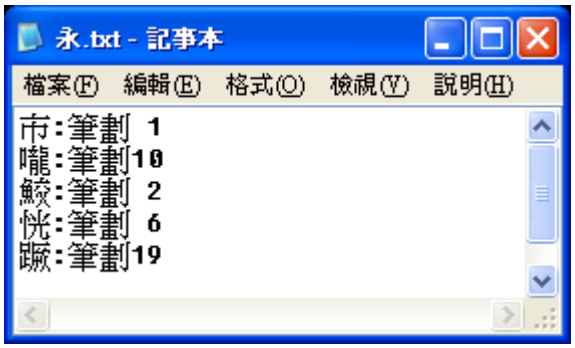


圖 4-3 「永」字書寫模擬過程及外框、細線字型輸出畫面



永.txt



永.jpg

永usr.jpg

永ou.jpg

永ln.jpg

圖 4-4 「永」字相關的儲存檔案

圖 4-5 「東海工設」全字模擬完成畫面

4-3 SolidWorks 輸出模式

在中國書法文字中，經常將一些帶有吉祥含義的短語，合寫成一個複合字，以祈求吉祥，例如：「招財進寶」、「囍」及「黃金萬両」…等。而這些複合字電腦字型系統中，常常面臨字型的缺字問題，因此使用者必須透過造字程式或繁複的影像處理過程才能取得；另外罕用字元及使用漢字的國家，例如：中國大陸、日本、越南及朝鮮半島…等，若要在設計軟體中使用這些字元，往往必須熟悉這些語系國家的輸入法或轉換程式，操作上十分不易。因此模擬系統平台除了提供螢幕輸出模式外，結合了SolidWorks API功能，將模擬出的字型輸出至SolidWorks中，讓設計師在字型的後續應用上能更加便捷；而輸出至SolidWorks中後，每一個筆劃為獨立的單元，可進行設計變更。如圖 4-6 中，「永」字輸出至SolidWorks中後，調整最後一個筆劃「捺」的位置。圖 4-6 為複合字「招財進寶」、「日日有見財」、「黃金萬両」、「喜喜」及「孔孟好學」螢幕及SolidWorks輸出畫面；圖 4-7 為罕用字螢幕及SolidWorks輸出畫面；圖 4-8 為日文漢字螢幕及SolidWorks輸出畫面。

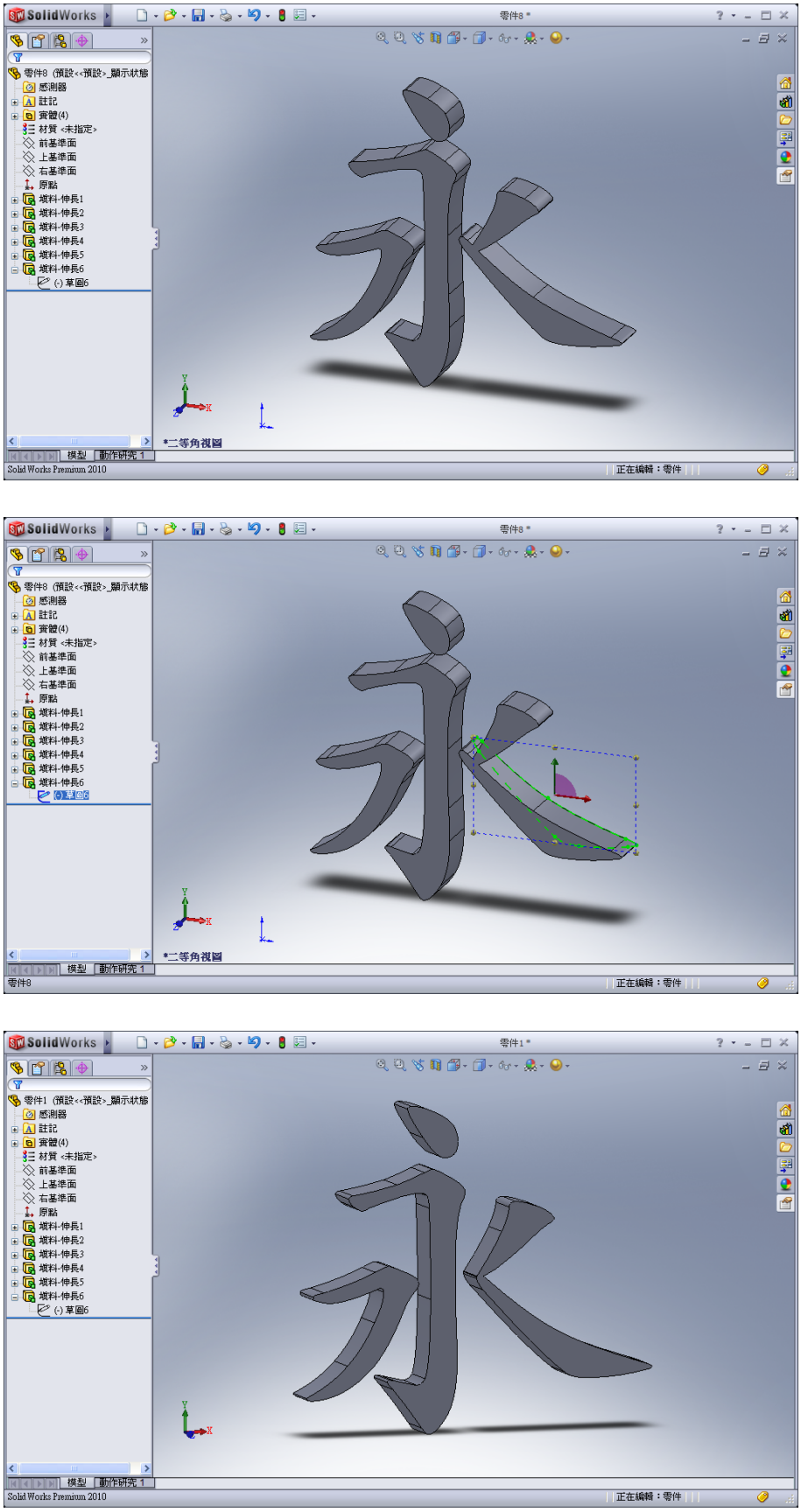
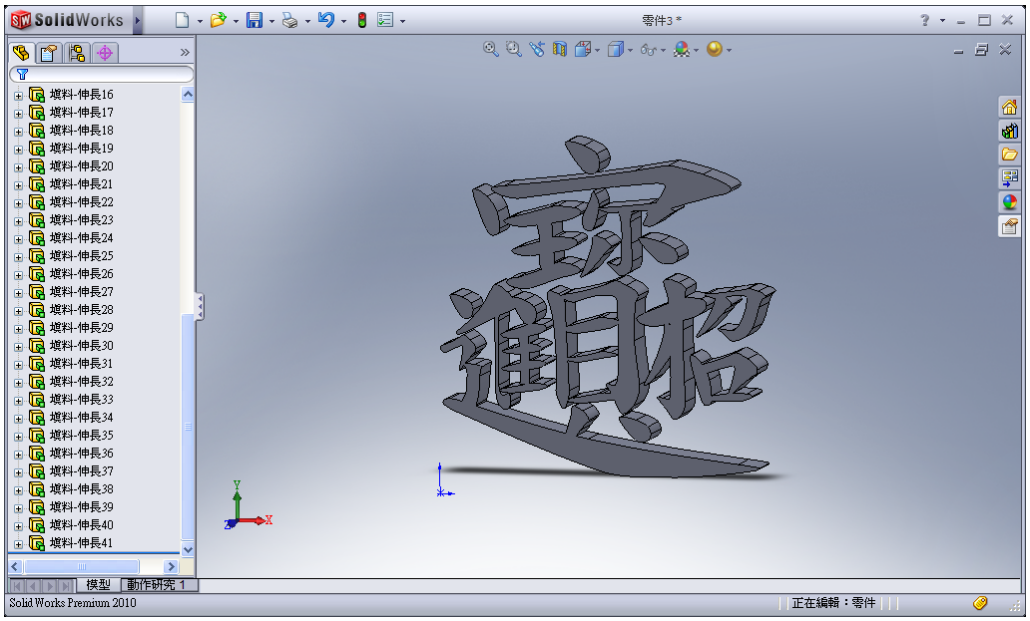
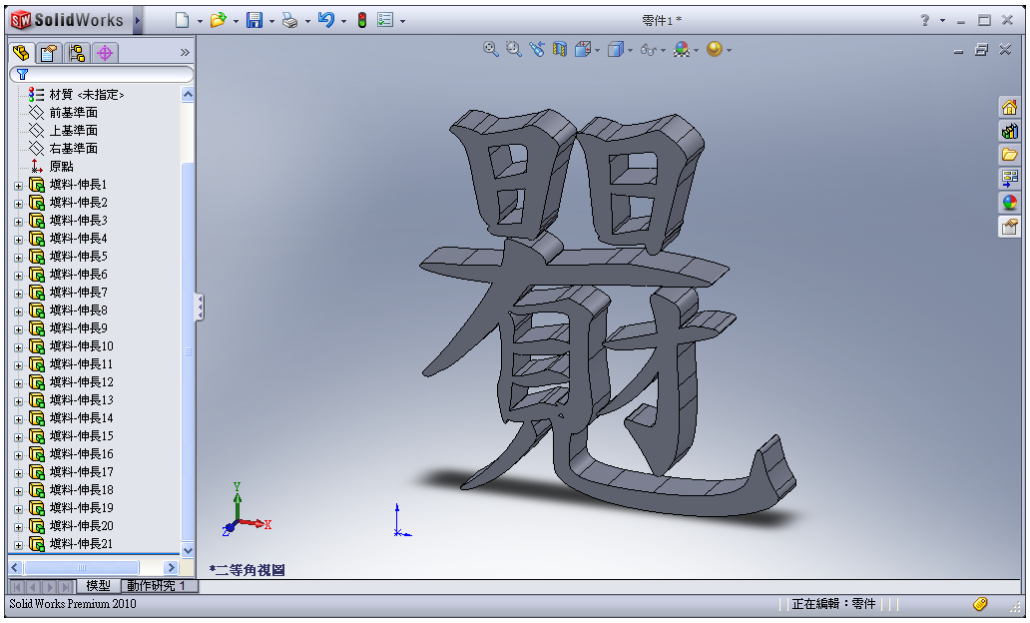
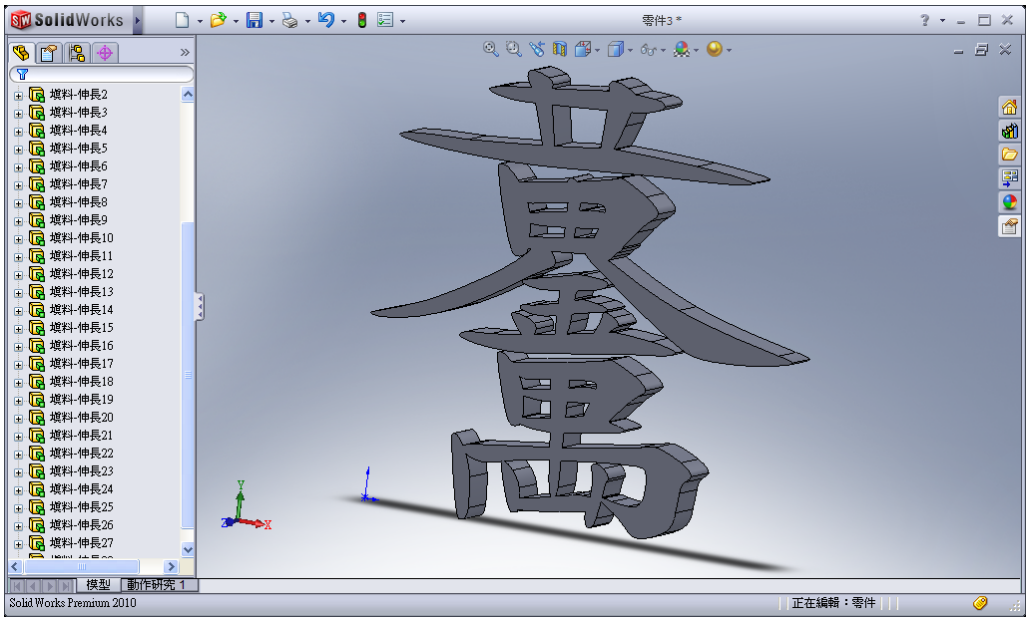
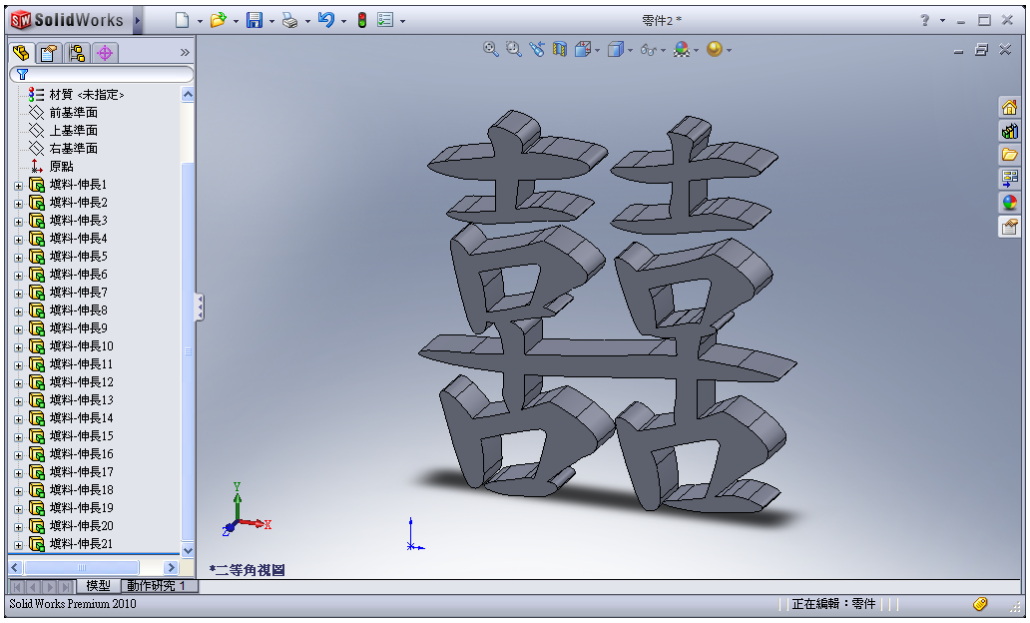


圖 4-6 「永」字輸出至 SolidWorks 後進行設計變更


SolidWorks
零件3 *
模型
動作研究 1
Solid Works Premium 2010
正在編輯：零件


SolidWorks
零件1 *
材質 <未指定>
前基準面
上基準面
右基準面
原點
*二等角視圖
模型
動作研究 1
Solid Works Premium 2010
正在編輯：零件


SolidWorks
零件3 *
模型
動作研究 1
Solid Works Premium 2010
正在編輯：零件


SolidWorks
零件2 *
材質 <未指定>
前基準面
上基準面
右基準面
原點
模型
動作研究 1
*二等角視圖
Solid Works Premium 2010
正在編輯：零件

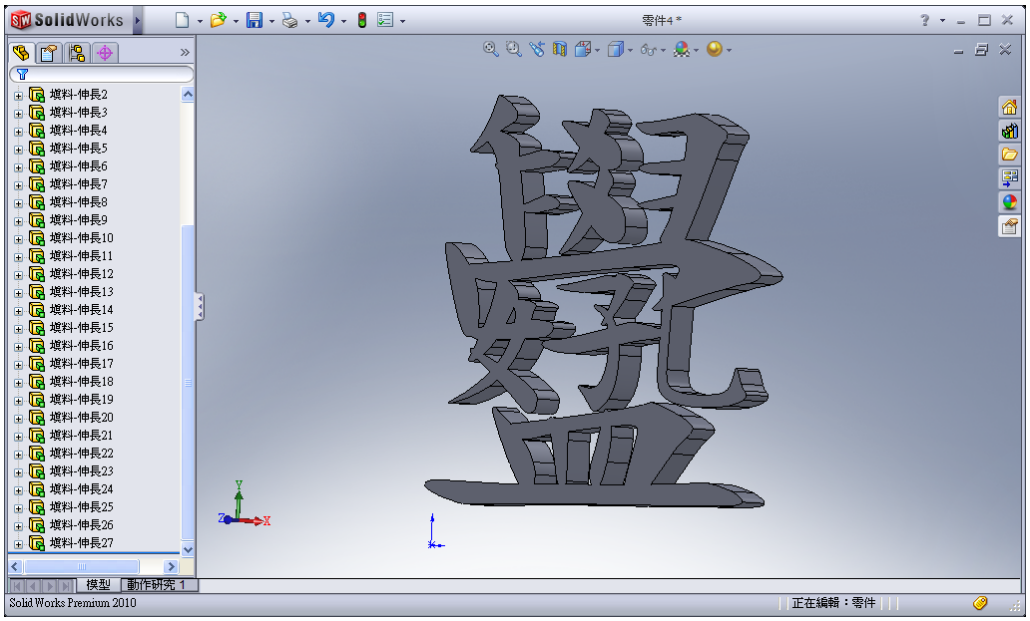
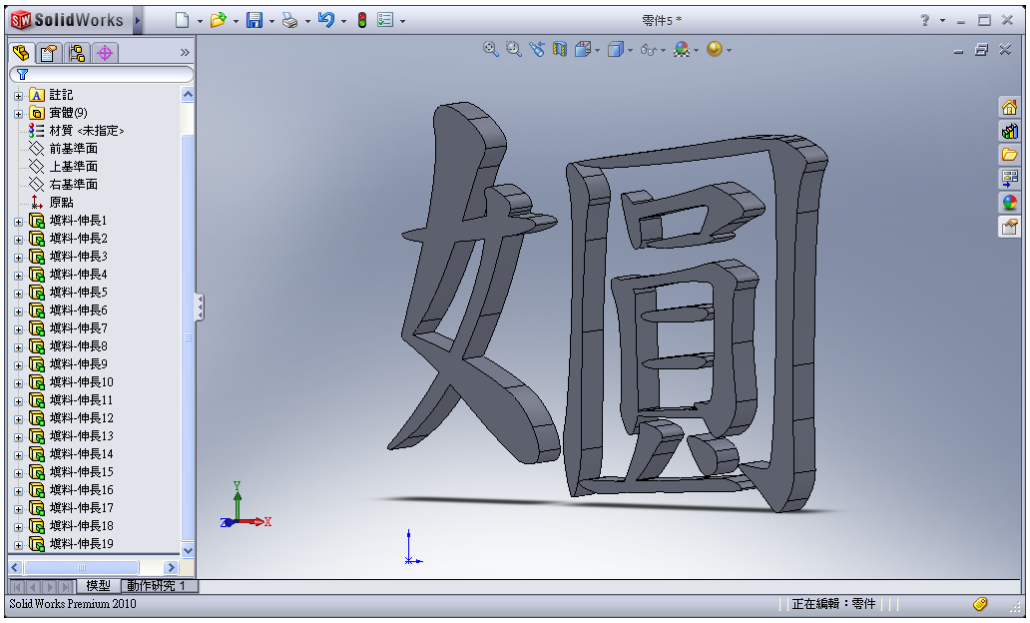
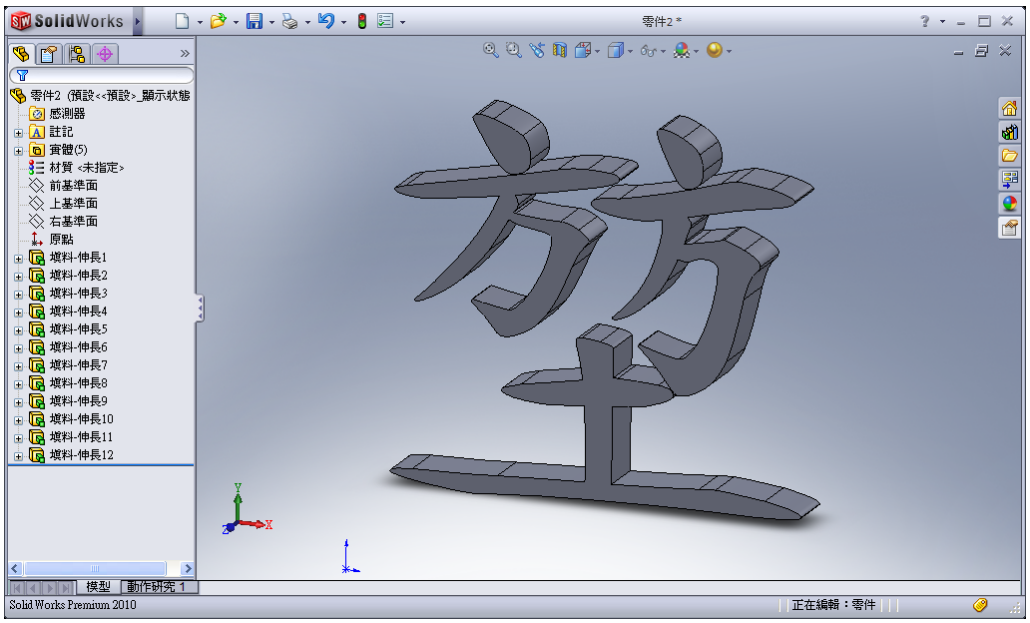


圖 4-7 複合字螢幕及 SolidWorks 輸出畫面





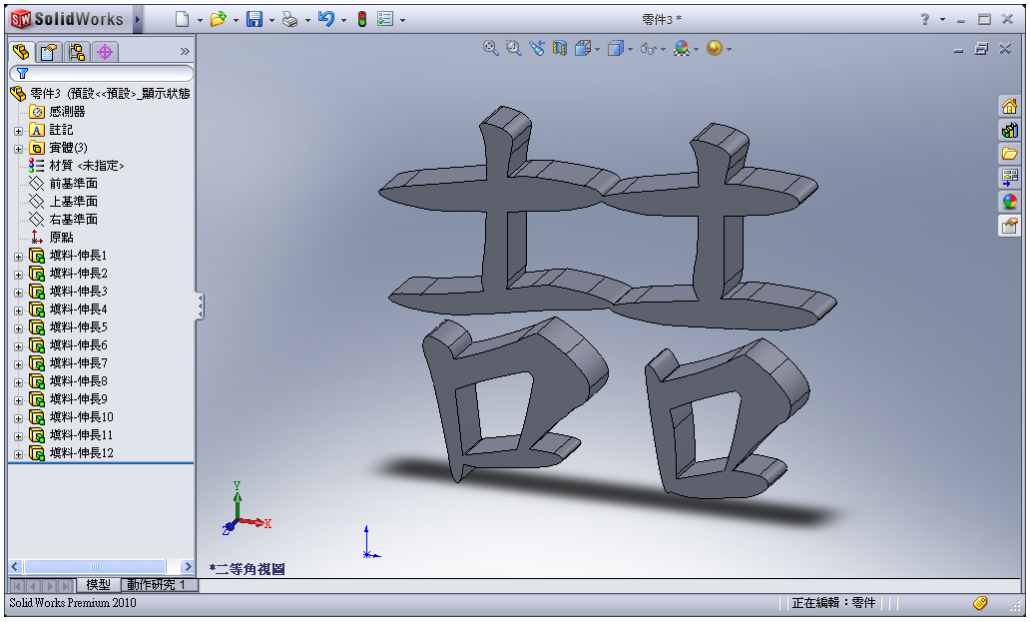
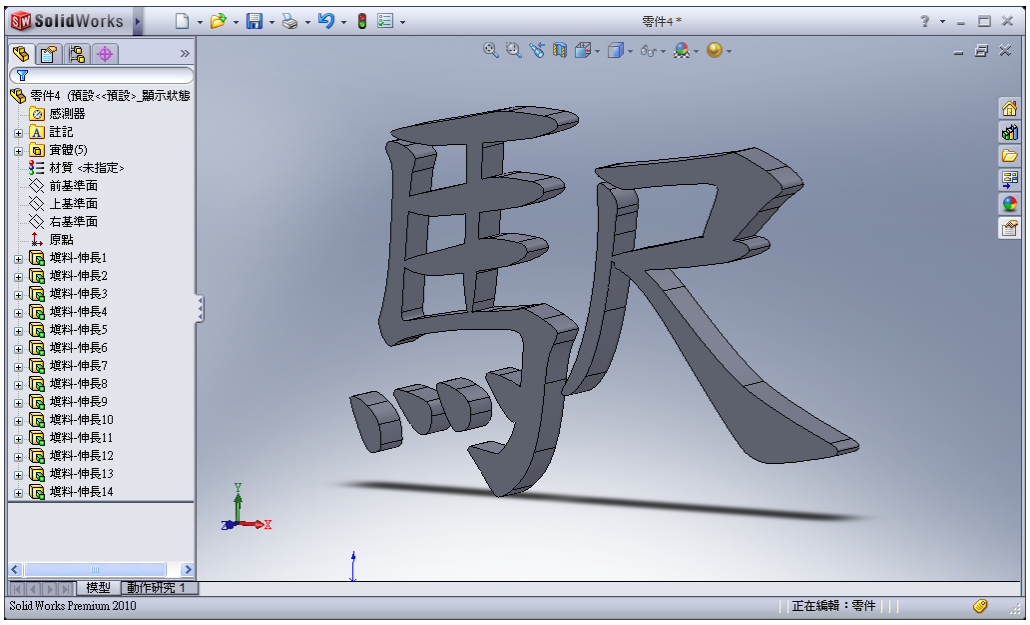
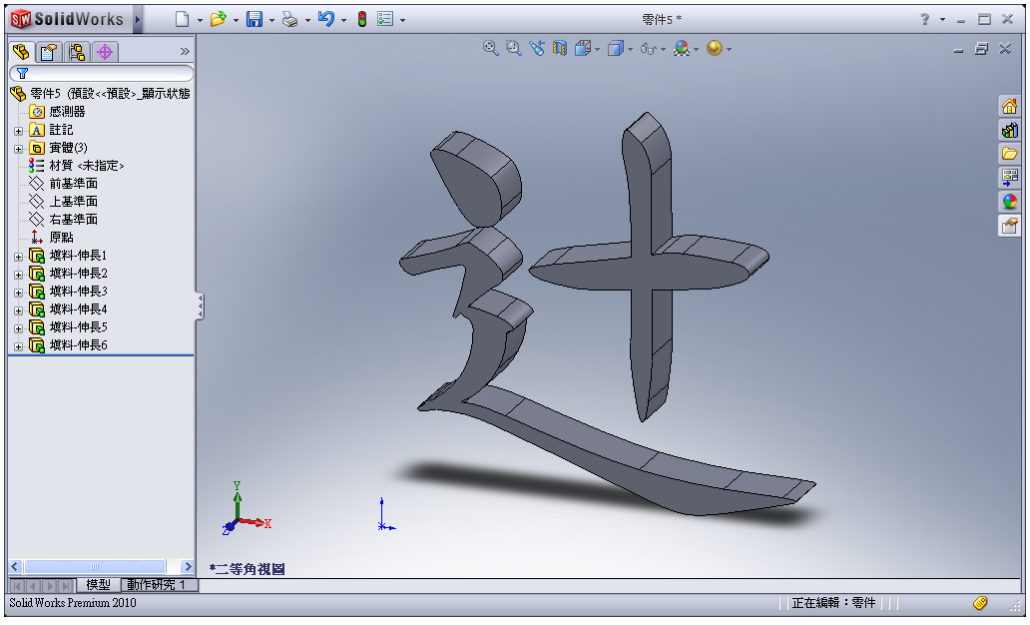


圖 4-8 罕用字螢幕及 SolidWorks 輸出畫面


SolidWorks
零件4 *
零件4 (預設<<預設>_顯示狀態
感測器
註記
實體(5)
材質 <未指定>
前基準面
上基準面
右基準面
原點
塑料-伸長1
塑料-伸長2
塑料-伸長3
塑料-伸長4
塑料-伸長5
塑料-伸長6
塑料-伸長7
塑料-伸長8
塑料-伸長9
塑料-伸長10
塑料-伸長11
塑料-伸長12
塑料-伸長13
塑料-伸長14
模型
動作研究 1
Solid Works Premium 2010
正在編輯：零件


SolidWorks
零件5 *
零件5 (預設<<預設>_顯示狀態
感測器
註記
實體(3)
材質 <未指定>
前基準面
上基準面
右基準面
原點
塑料-伸長1
塑料-伸長2
塑料-伸長3
塑料-伸長4
塑料-伸長5
塑料-伸長6
*二等角視圖
模型
動作研究 1
Solid Works Premium 2010
正在編輯：零件

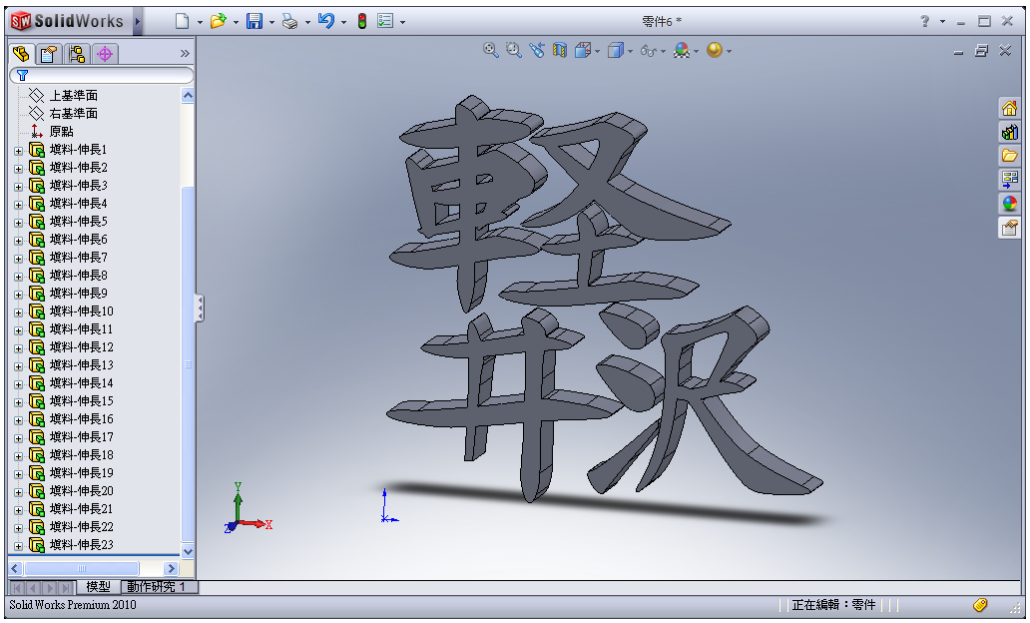


圖 4-9 日文漢字螢幕及 SolidWorks 輸出畫面

4-4 系統平台開發之困難與問題討論

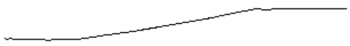
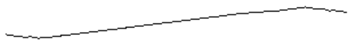
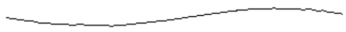
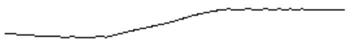
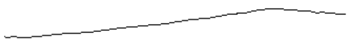
在進行中國書法字型模擬平台開發研究過程中，因中國書法字型的多樣化，延伸了許多研究上的困難與問題點，討論如下：

1. 電腦字型：在研究上採用的字型為 Windows XP 內建的標楷體，為筆劃方式的造字字型，但其筆劃分類部份與一般書寫習慣不同，所以在應用上需將與一般書寫習慣不同的筆劃加以處理，如本文 3-1-1 中所述。而不同的筆劃方式的造字字型及輪廓方式的造字字型在本研究中並未加以探討。

2. 細線化處理：由於標楷體部份筆劃會產生一些特別情形，易造成系統判斷頂點位置時誤判。輸入筆劃時需配合其狀況，系統才能

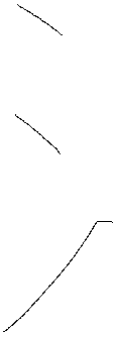
輸出較為符合的字型筆劃。例如:「橫」、「直」筆劃在同一種字型上可能書寫的方式可能就有幾十種，如表 4-1 列舉一些輸入「橫」、「直」的筆劃、系統比對判讀出來的字型筆劃及細線化的結果；又如表 4-2 中「水」部首的第三筆，因字型筆劃細線化後，前端會有小勾現象，使用者輸入時亦須以此方式輸入才能得到較佳的結果。因此研究上須根據可能發生問題的情形，加以進一步分析及處理，找出適用的細線化改良方法。

表 4-1 「橫」、「直」筆劃細線化處理問題

輸入的筆劃	輸出的字型筆劃	字型筆劃細線化
		
		
		
		
		
		

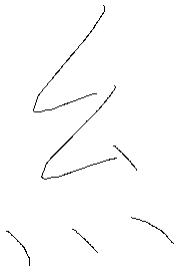
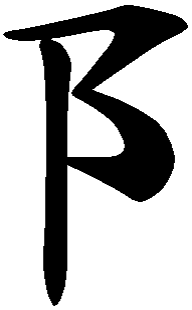
輸入的筆劃	輸出的字型筆劃	字型筆劃細線化
		
		
		
		

表 4-2 「水」部首的第三筆細線化問題

輸入的筆劃	輸出的字型筆劃	字型筆劃細線化
		
		

3. 筆劃特徵擷取：目前有關於字型辨識或筆劃辨識已經相當多的文獻，可提供參考，但由於不同的字型和書寫習慣會對不同的筆劃特徵有適用性的問題。如本文 3-1-1 中所述，針對「標楷體」勾的部份有進行筆劃合併作業；但因使用者書寫的習慣各有不同。如表 4-3 所示，「糸」及「阝」部首的第一、二筆筆劃，有些使用者習慣會同一筆劃完成，因在本系統中未進行筆劃合併處理；當使用者連筆輸入時，系統便無法找出相對應的筆劃。因此在筆劃特徵擷取及處理的部份，如何針對不同的書寫方式決定使用的筆劃特徵，是需要更深入研究的問題。

表 4-3 筆劃特徵合併的問題

輸入的筆劃	輸出的字型筆劃	字型筆劃細線化
		
		

4. 筆劃資料庫：通常一種中文字型大約有一萬多字，將其筆劃分開後約有二十萬筆左右，而在應用上筆劃資料庫如果筆劃太大，會造成搜尋速度較慢；而若筆劃資料庫太小，可能會有字型完整性不足的問題。如何決定適合的筆劃資料庫大小，刪除重覆的筆劃，和筆劃資料庫中要使用的筆劃特徵欄位及索引的使用，都會對最後的辨識速度和正確度有決定性的影響，是相當重要的一部分。

5. 辨識正確性：由於同一筆劃在書寫上可以有很多不同的方式，因此在辨識上也會有相當大的困難；因此如何增加筆劃辨識正確性及輸入的彈性，亦是研究的重點。例如：根據使用者輸入的筆劃，系統提供相似筆劃提供選擇，或是根據使用者輸入筆劃和選擇結果加以記錄，或是根據筆劃在一般書寫習慣中出現的順序，以增加辨識的正確性。

第五章　結論與建議

5-1 結論

本研究為中國書法字型模擬系統平台開發，完成具體成果如下列所示：

1. 建構一套中國書法字型模擬模式，並以電腦字型中的標楷體書法字型作為模擬的目標。
2. 使用者可以輕易的透過滑鼠或數位筆…等輸入設備，自由的書寫中文筆劃，系統平台會即時的模擬輸出書法字型。
3. 系統平台所模擬輸出的標楷體字型，除了保有原來書法字型的特徵外；因為加上使用者手寫輸入筆劃結構的差異化，讓電腦字型能跳脫刻版的一致性，更能突顯使用者書寫文字的個性化、電腦字型活潑性及增添書寫的樂趣。
4. 透過字型筆劃特徵的擷取、分析，對於不符合一般書寫習慣的筆劃進行合併作業，並將重覆的筆劃加以刪減；原本標楷體字型中共有 178,196 個筆劃資料，精簡成筆劃數 10,218 個筆劃資料；重建了標楷體字型的筆劃資料庫，對於系統平台運作效率的提昇有很大的助益。
5. 提出中國書法字型筆劃優化細線化法則，以改善現有書法字型經細線化後所產生不良細線化結果（例如：尾端化現象），增進筆劃擷取、分析及比對的準確性。
6. 建立一套適用於標楷體書法字型筆劃的編碼系統，以作為筆劃在

資料庫中的索引值及與手寫輸入筆劃比對用途。每個筆劃均以 15 位數進行編碼，編碼的項目分別四個部份：筆劃長寬比（2 位數）、筆劃頂點位置（2 位數）、頂點間夾角（1 位數）及筆劃分佈值（10 位數）。

7. 提出筆劃分佈特徵值擷取及分析方法，分別透過在 x 軸向 y 軸方向掃描及在 y 軸向 x 軸方向掃描方式，擷取筆劃分佈特徵值，以作為書法字型筆劃及手寫輸入筆劃比對依據。

8. 提出描邊字處理的演算方法，作為不同的螢幕輸出模式的選擇，以結省使用者在字型應用上的處理時間。在系統平台中，除了可模擬一般的實心字型外，亦可選擇描邊字的「外框字型」或是只有筆劃骨架的「細線字型」作為輸出。

9. 提供模擬字型及手寫輸入筆劃字型，以 jpg 圖檔格式存檔輸出功能，讓使用者可再匯入影像處理軟體中，進一步的處理及應用。

10. 導入 SolidWorks API 技術，使得模擬字型可快速自動的輸出至 3D 繪圖軟體 SolidWorks 中。這項功能除了讓一般常用中文字在 SolidWorks 中更方便被使用外；對於一些複合字、罕用字及不同漢字體系國家的文字在 SolidWorks 中的應用上，提供很好的靈活度，可節省許多繁複的處理時間，例如：中文造字、曲線擬合及輸入法的熟悉…等。

5-2 後續研究建議

如本文 4-4 中所述，仍有一些困難及問題點，有待後續研究加以改進，未來研究方向建議可就以下幾項進行探討：

：針對細線化方法再加以改良，使其能與手寫習慣更能吻合；筆劃特徵值分析、擷取與編碼使其更具彈性；將字型筆劃的書寫方向性、筆順及個人書寫習慣等因素，加入筆劃資料庫中，讓筆劃的辨識更精確完善。

3. 導入 PBT（Problem-Based Training，問題導向訓練）概念於系統設計中，將筆劃間相互關係，例如：相交比例、長度及角度等，建立至筆劃資料庫中，並提供引導輸入學習模式；當模擬出的字型筆劃不理想時，系統可自動的建議修正或由使用者自行設定參數進行調整。如此可增進系統平台的彈性，讓使用者能更輕易模擬出理想的筆劃。

4. 目前系統平台只能模擬標楷體字型的輸出，可加入更多的字型選擇。除了電腦字型外，古代碑帖或是個人筆跡亦是很好的研究選擇。可利用逆向的方式，透過字型邊緣輪廓之偵測，筆劃的擷取與分析，將筆劃資料導入資料庫中。

5. 目前系統僅提供模擬字型輸出成 jpg 圖檔格式，可加入向量圖檔的儲存格式，讓輸出的字型檔可匯入更多的設計軟體中再應用。

6. 結合系統造字程式功能，讓模擬出的字型可儲存成 TrueType 字型格式，以提供一般輸入使用。如此使用者所模擬出的字型，便可其他軟體中用打字的方式，直接輸入重覆使用，例如：文書處理軟體、繪圖軟體…等。

7. 結合字元辨識系統，讓一般有提供手寫輸入介面的設備，例如：PDA、行動電話等，使用者在輸入時，即能顯示漂亮的書法字型，而不再是粗糙的手寫線條。

8. 藉由使用對象的分析與設定，開發不同使用者功能模式，以適用於不同的使用者需求，例如：兒童學習模式、一般使用者模式及專業設計師模式…等。

9. 加強使用者介面設計(User Interface Design)，讓系統平台除了功能運作外，整體風格及操作介面品質能進一步提昇。

參考文獻

1. Pandya, A. S., Macy, R. B. (1996). Pattern Recognition with Neural Networks in C++, CRC PRESS & IEEE PRESS。
2. Drummond, J., Essen, R.V., and Boulerie, P. (1991). Some Considerations on Vectorization Algorithms, ITC journal, 3, 153-157.
3. Chuang, C.T., and Tseng, L.Y. (1995). A Heuristic Algorithm for the Recognition of Printed Chinese Characters. IEEE Transactions on System, Man and Cybernetics, 25(4), 710-717.
4. Leung, C. H. and Sze, L. (1997). Feature Selection in the Recognition of Handwritten Chinese Characters, Engng Applic. Artif. Intell., 10(5), 495-502.
5. Choi, B.K. (1991). Surface Modeling for CAD/CAM., Ch 3, Elsevier.
6. Hornby, A.S. (1972). The Advanced Learner's Dictionary of Current English, 2nd Ed., Oxford Press.
7. Holt, C. M., Stewart, A., Clint, M. and Perrott, R.H. (1987). An Improved Parallel Thinning Algorithm. Communications of the ACM, 30(2), 156-160.
8. Ivind, D.T., Anil, K.J. and Torfinn, T. (1996). Feature Extraction Methods for Character Recognition - A Survey, Pattern Recognition, 29(4), 641-662.
9. Toraichi, K., Kumamoto, T., Yamamoto, K. and Yamada, H. (1996). Feature Analysis of Handprinted Chinese Characters , Pattern Recognition Letters, 17, 795-800.
10. Heutte, L., Paquet, T., Moreau, J.V., Lecourtier, Y. and Olivier, C. (1998). A Structural / Statistical Feature Based Vector For Handwritten Character Recognition, Pattern Recognition Letters, 19, 629-641

11. Lim, S.B., Kim, M.S. (1995). Oriental Character Font Design by a Structured Composition of Stroke Elements. Computer Aided Design, 27(3), 193-207.

12. Lin, J.R., Chen, C.F. (1996). Stroke Extraction for Chinese Characters Using a Trend-Followed Transcribing Technique. Pattern Recognition, 29(11), 1789-1805.

13. Lin, C.F., Fang, Y.F., and Juang, Y.T. (2001). Chinese Text Distinction and Font Identification by Recognizing Most Frequently Used Characters. Image and Vision Computing, 19, 329-338.

14. Nakamura, T., He, L., and Itoh, H. (1999). SRAMCs：Skeleton Revision Algorithm Using Maximal Circles, Transactions of Information Processing Society of Japan, 40(2), 768-771.

15. Parker, J.R. (1999). Algorithm for Image Processing and Computer Vision, Wiley Computer Publishing, 176-188.

16. Romero, R.D., Touretzky, D.S., and Thibadeau, R.H. (1997). Optical Chinese Character Recognition Using Probabilistic Neural Networks. Pattern Recognition, 30(8), 1279-1292.

17. Sarfraz, M., and Razzak, F.A. (2002). An Algorithm for Automatic Capturing of the Font Outlines. Computer & Graphics, 26, 795-804.

18. Sarfraz, M., and Razzak, F.A. (2003). A Web Based System to Capture Outlines of Arabic Fonts. International Journal of Information Sciences, 150, 177-193.

19. Stentiford, F.W.M. and Mortimer, R.G. (1983). Some New Heuristics for Thinning Binary Handprinted Characters for OCR. IEEE Transactions on Systems, Man, and Cybernetics. 13(1), 81-84.

20. Stallings, W.W. (1976). Approaches to Chinese Character Recognition, Pattern Recognition, Pergamon Press, Great Britain, 8, 87-98.

21. Stallings, W.W. (1977). Chinese Character Recognition, in Fu, K.S.,

1977, Syntactic Pattern Recognition and Application, Spring-Verlag, New York, 95-123.

22. Wong, H.T.F., and Ip, H.H.S. (2000). Virtual Brush: a Model –based Synthesis of Chinese Calligraphy. Computers & Graphics, 24, 99-113.

23. Wong, P.Y.C & Hsu, S.C. (1995). Design Chinese Typeface Using Components, IEEE. 0730-3157/95, 416-421.

24. Wang, C.S., Hsiao, C.Y., Chang, T.R., and Teng, C.K. (2006). Product Development for Chinese Calligraphy Using Reverse Engineering and Rapid Prototyping, Virtual and Physical Prototyping, 1(4), 259-269.

25. Zhang, T.Y., and Suen, C.Y. (1984). A Fast Parallel Algorithm for Thinning Digital Pictures, Commun. ACM, 27(3), 236-239.

26. Zhang, S., and Fu, K.S. (1984). A Thinning Algorithm for Discrete Binary Images. Proceedings of the International Conference on Computers and Application, Beijing, China. 879-886.

27. 三維空間網站，用 Delphi 對 SolidWorks 進行二次開發，http://www.MCADtools.net。

28. 王中行、張庭瑞、柯俊宏（2004），中國書法吉祥合成字之逆向設計研究，東海學報，45，117-130。

29. 王志堅（1995），手寫中文字根之辨識，國立成功大學工程科學系碩士論文。

30. 王舜正（2000），書法影像內中文字的二值化和非等方向性擴散，交通大學資訊工程系碩士論文。

31. 方建中（2001），利用結構式特徵在多層次架構做光學字元辨識，國立中央大學資訊工程研究所碩士論文。

32. 文鼎科技網站，http://www.arphic.com/tw/index.htm。

33. 丘永福（2005），字學，台北市：藝風堂。

34. 呂昭慧（1994），電腦中文字型之意象調查與分析，國立台灣科技大學工程技術研究所碩士論文。

36.余奕昌（2006），應用 CAD 軟體 API 輔助膝下殘肢曲面之建模與編修，國立成功大學機械工程學系碩士論文。

37.吳冠峰（1987），手寫中文字型辨認之研究，國立成功大學電機工程研究所碩士論文。

38.吳偉賢（2000），筆劃特徵用於離線中文字的辨認，國立中央大學資訊工程研究所博士論文。

39.邱述文（2008），應用 CAD 軟體 API 建立刨齒加工模擬系統，清雲科技大學機械工程研究所碩士論文。

40.邱煥鐘（1998），中文字型資料結構之分析與數值化，技術學刊，13(4)，555-562。

41.邱顯達（1992），自動化 IC 檢視與辨認系統之研究，國立成功大學機械工程研究所碩士論文。

42.莊志鴻（1999），以 WMMS 法做車牌字元辨識，國立交通大學電機語控制工程學系碩士論文。

43.連挺佑（2002），毛筆筆鋒之模擬及其水墨書畫系統，國立臺灣大學資訊工程學研究所碩士論文。

44.張鴻楨（1993），以結構特徵為基楚之鬆弛比對法用於線上手寫中文字辨識，國立中央大學資訊及電子工程研究所碩士論文。

45.張皓傑（1996），個性化字型之筆劃特徵擷取系統，國立台灣大學電機工程研究所碩士論文。

46.張耀明（1999），灰色理論為基礎之影像邊緣偵測，中原大學電子工程研究所碩士論文。

47.張銘豪（1996），利用分割辨識方法之英文數字辨識系統，國立中山大學資訊工程研究所碩士論文。

48.唐榮錫等（2005），計算機圖學，台北市：網奕資訊。

49.徐曉珮譯（2005），數位影像處理（Alasdair McAndrew 原著，An Introduction to Digital Image Processing with MATLAB），台北市：湯姆生。

50.威鋒數位華康字型網站，http://www.dynacw.com.tw/index.asp。

51.許鴻基（1992），階層模式導引產生中文字型，國立交通大學資訊工程研究所碩士論文。

52.崔陟（2001），書法，台北市：貓頭鷹。

53.黃石老人（1984），中國歷代名家書法，台北市：星光。

54.湯嘉明（2001），液晶顯示器上動態中文字視認性與視覺動向之研究，雲林科技大學視覺傳達設計系碩士論文。

55.馮議徹（2007），漢字的筆劃特徵與風格意象，國立成功大學工業設計學系碩士論文。

56.陳世昇（2005），應用 CAD 軟體 API 建立設計概念具體化輔助系統，國立成功大學機械工程學系碩士論文。

57.陳宗輝（2002），中文字型平滑化應用於電腦輔助製造工程，中國機械工程學會第十九屆學術研討會，C9-022，965-972。

58. 陳彥名（2006），應用 CAD 軟體 API 與快速成型技術製作義肢承筒，國立成功大學機械工程學系碩士論文。

59.陳映舟（2001），中文草書書法字帖的文字切割與辨識，國立交通大學資訊工程系碩士論文。

60.陳喜榮（1981），中英文美術字型設計，高雄市：愛樂書店。

61.溫福助（2000），類神經網路樣板比對法於車牌字元辨識之研究，國立臺灣大學電機工程學研究所碩士論文。

62.蔡仲智（2003），應用輪廓線分析於中文字筆劃擷取，國立中興大學電機工程學系碩士論文。

63.蔡登傳（1998），以字型描述值建構中文視認度的公式研究，科技學刊，8(2)，145-156。

64.蔡登傳（2001），應用文字的字型描述值評估其視認度之研究，國立台灣科技大學工業管理系博士論文。

65.廖振偉、史天元、張崑宗（2001），細線化演算法比較，地籍測量，20(3)，1-18

66.劉浩群（2003），虛擬書法之設計與實作，大同大學資訊工程研究所碩士論文。

67.劉育儒（1992），由二元化影像繪製線畫圖之研究，成大航空測量研究所碩士論文。

68.盧建智（2000），中文字帖書法字之描述，國立交通大學資訊工程系碩士論文。

69.謝忠勳（2006），個人化向量字型，國立中山大學資訊工程學系研究所碩士論文。

70.顏技文（1994），應用多層分類方法於手寫數字辨認之研究，國立成功大學資訊工程研究所碩士論文。

71.羅鳳珠、周曉文（1998），古籍數位化的重要里程-大陸北京師範大學完成小篆字型字庫的建立，漢學研究通訊，17(3)，302-304。

72.顧大我（1994），漢字字型標準化之我見，華文世界，36，33-36.（Gu, D.W., 1994, A Study on the Standardization of Chinese Characters, Chinese World, 36, 33-36）.

73.饒忻、曾健維、呂理邦（2003），自動化 IC 印字瑕疵檢測(I)-字元辨識，Journal of the Chinese Institute of Industrial Engineers，20(4)，317-326。