

中華民國第 65 屆中小學科學展覽會

作品說明書

科 別：電腦與資訊學科

組 別：高級中等學校組

作品名稱：以特徵點主導之圖像輪廓分段貝茲曲線擬合與節點簡化
系統

關 鍵 詞：向量化、貝茲擬合、演算法

編 號：

摘要

本研究提出一套基於貝茲曲線的手繪圖形向量化系統，透過自動節點擷取與分段擬合機制，在降低貝茲節點的同時維持高度幾何精度，本研究整合自行設計的線段向量與曲率特徵處理演算法（SVCFP），自動分割手繪筆劃並擷取關鍵節點，搭配最小平方法（LSM）進行高效貝茲曲線擬合。整體流程搭建於網頁互動介面，可即時繪圖並獲得向量化結果。研究結果顯示，相較於傳統工具，本研究在控制貝茲節點數量上平均減少約 84.6%，最高可減少 90.8% 的資料量，同時在精準度僅有輕微變動的前提下（BMND 分數平均僅變動 28 至 36 分），仍能保持流暢準確的輪廓重建，亦適用於大量或即時處理情境，為圖形向量化提供一種高效率、低冗餘、高視覺精度維持且實用價值高的解決方法。

壹、前言

一、研究動機

當使用觸控螢幕或數位畫板進行手繪時，筆劃通常是由離散的像素點組成，這些像素點的排列會受到解析度或設備規格的影响，進而導致筆劃邊緣不夠平滑，或在放大縮小時出現鋸齒狀失真，並在旋轉時會極大程度的破壞原有圖像。市面上有許多向量繪圖軟體，使用者可以藉由鋼筆工具直接進行向量作畫、使用貝茲繪圖工具手動調整控制點擬合圖像或使用轉換工具將像素影像轉換為向量圖像，但在本研究使用現有軟體（如 Adobe Illustrator、Inkscape）後發現以下問題：

（一）貝茲曲線（Bézier curve）節點過多，造成資料冗餘與計算負擔

現有轉換工具為了保留圖像細節，往往產生大量貝茲曲線控制節點。以 Inkscape 為例，單一筆劃輪廓即可能產生數百甚至上千個節點，導致儲存成本提高，傳輸與渲染效率下降，亦不利於後續壓縮或深度學習模型整合，以一張 64KB 的圖為例，在經過向量化後 SVG 檔竟高達 91KB，不僅沒有壓縮反而增長資料量。

（二）貝茲曲線工具使用耗時高

傳統的貝茲曲線繪製工具(鋼筆工具)往往仰賴使用者手動拖曳與調整控制點，以達

成曲線擬合的目的。然而，當面對筆劃變化複雜、曲率急遽變化的區域（如手寫字體中的筆鋒轉折、書法筆觸的收尾與轉折）時，這種人工操作方式不僅難以快速對應實際筆勢變化，亦無法確保局部結構的精確擬合，常導致反覆調整與耗費大量時間，進而大幅降低整體的作業效率與實用性。

（三）缺乏自動化能力，使用門檻較高

目前多數向量化工具需使用者手動調整貝茲曲線的控制點，此過程不僅耗時，亦需具備一定技術門檻，對非專業使用者而言不甚友善。

本研究針對以上的問題，並經過文獻探討後，本研究將使用自行設計的演算法，來改良現有貝茲曲線挑選節點、儲存等問題。貝茲曲線是一種數學模型，可以用來表示曲線，以下選擇最常見的三次貝茲曲線進行介紹，其公式如公式 1-1 所示，其中 P_0 及 P_3 代表貝茲節點、 P_1 及 P_2 則是貝茲控制點，由四個點的座標控制曲線的形狀，其繪製曲線的效果如圖 1-1 所示。

$$B(t) = P_0(1-t)^3 + P_1t^2(1-t) + P_2t^2(1-t) + P_3t^3 \quad 1-1$$

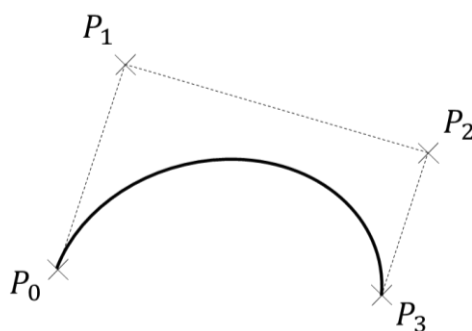


圖 1-1 貝茲節點與控制點繪製貝茲曲線（圖片來源：

https://en.wikipedia.org/wiki/B%C3%A9zier_curve）

貝茲曲線的特點在於可透過少量控制點描述平滑曲線，且在不同解析度下仍能維持穩定性。貝茲曲線在許多領域已有廣泛應用，例如字型設計、數位藝術、動畫製作等，並且因為其向量化特性，可以有效減少資料量並提高畫面呈現的效果。然而，儘管貝茲曲線在理論上具有很大優勢，但直接使用貝茲曲線來擬合手寫筆劃有幾項技術難點整理如下：

（一）如何自動選擇最佳的控制點？

1. 如果控制點選擇不當，擬合出的曲線可能會與使用者手繪圖形產生偏差。
2. 傳統的擬合方式可能會忽略使用者手繪的細微變化，導致手繪特徵失真。

（二）如何在降低資料量的同時保持手繪的完整性？

1. 向量化的過程需要在減少計算負擔與保持手繪特徵之間取得平衡。
2. 特徵保留愈多，雖可提升圖像還原度，卻也將導致檔案容量顯著增加，影響儲存與傳輸效率。

上述技術難點限制了貝茲曲線在手繪圖形數位轉換中的應用潛力，因此本研究致力於克服這些技術障礙，開發一套具備高效率、精確度與自動化能力的向量化方法，讓使用者的手繪輸入能更準確地轉換為電腦可處理的貝茲曲線，進而提升數位手繪技術的實用性與應用價值。

二、研究目的

本研究旨在開發一套手繪圖形向量化方法，運用貝茲曲線精確描述手繪筆劃之形狀輪廓。系統核心為自行設計的切割演算法「線段向量與曲率特徵處理（**Segment Vector and Curvature Feature Processing**）」，簡稱 **SVCFP**，可對使用者於網頁介面繪製之筆劃序列進行特徵提取與曲線分段，藉此生成出高擬合度的貝茲曲線。此外，為了提升擬合精度與相似度，在本研究中亦設計圖形比對損失函數，用以輔助貝茲控制點的最佳化，使擬合曲線更貼近原始筆跡的細節特徵，整理研究目的如下：

（一）改善傳統貝茲曲線擬合節點過多問題

本研究透過貝茲節點的自動優化，能以更少的節點達到相同甚至更高的擬合精度，從而減少計算負擔與儲存成本，並可調整閾值，選擇使用者所需要的精確度。

（二）在減少大量點位下保留手繪筆劃的細微變化

本研究方法結合筆劃長度與曲率動態分析，並利用 **SVCFP** 演算法，自動調整貝茲曲線的節點數量與位置，使轉換後的向量曲線更精確貼合原始筆跡，得以保留手繪

中的細微特徵。在有效減少點位數量的同時，亦能維持筆劃的細節變化與特徵完整性。

（三）降低使用者的使用門檻

本研究透過演算法自動化選擇貝茲節點與擬合曲線，使用者無需進行手動調整，即可獲得高品質的向量化結果，提升使用體驗與轉換效率。

本研究期望在提高手繪圖形轉數位格式的精度與效率的同時，思考其在各種數位內容創作領域中的應用潛力，並促進人工智慧技術在手寫辨識和數位繪圖中的發展。

三、文獻回顧

本研究之文獻探討，分為貝茲曲線、手繪線條向量化與演算法等，相關內容如下：

（一）貝茲曲線

1. Baydas 與 Karakas (2019) ^[1] 提出了一種新的方法，能夠確定任何一條曲線是否可以被定義為貝茲曲線，該研究介紹了創建矩陣的概念，並證明了該矩陣的反向矩陣可用於精確求解貝茲曲線的控制點，其優點在於提供了一種精確而非近似的方法，適用於多領域如計算機輔助幾何設計，但缺點是該方法依賴於矩陣反推，可能對於計算資源的需求較高，並且對於高階多項式的處理較為複雜，對於已被破壞之圖像難以反推回數學函式，需要在加上其他演算法輔助判斷。
2. Pastva (1998) ^[2] 研究了如何使用總體最小平方法 (Total Least Squares, TLS) 擬合貝茲曲線，使其能夠同時最小化水平方向與垂直方向的殘差 (residual)，該研究詳細探討了伯恩施坦多項式 (Bernstein polynomial) 與貝茲曲線的數學基礎，並提出了一種基於高斯-牛頓法 (Gauss-Newton Method) 的優化演算法，以求得最佳的 control 點與節點，文中提及的演算法，其優點在於提供了一種更準確的擬合方法，適用於數據分佈不均的情境，並具有較高的幾何穩定性；然而，該演算法在計算上較為複雜，尤其是在處理高維數據或實時應用時，計算成本可能較高。此外，該作者主要聚焦於數學方法的發展，對於具體的工程應用討論較少，故較難依其理論進行實作應用。

3. Mad、Zain 與 Miura (2023) ^[3] 探討了如何利用廣義分數貝茲曲線進行曲線擬合。該研究指出傳統貝茲曲線在靈活性和可調整性方面存在侷限，因為其形狀無法在不改變控制點的情況下進行調整，而為了解決這一問題，研究者們引入了具有形狀參數和分數參數的廣義分數貝茲曲線，這些參數允許在不改變控制點的情況下調整曲線形狀，從而提高了曲線擬合過程的靈活性和效率，此外，還利用了分數連續性，允許在曲線的公共點或連接處進行位置調整，進一步增強了曲線擬合的靈活性；然而該研究主要集中在數學模型的構建與理論分析，對於實際應用中的性能和效果尚需進一步驗證。

(二) 手繪線條向量化

1. 池品軒、林文杰與莊榮宏 (2013) ^[4] 提出了一種利用擴散曲線技術，將點陣圖自動轉換為向量圖的方法。研究者們從點陣圖中提取輪廓、顏色和模糊像素點，並將其轉換為擴散曲線所需的幾何元素，使生成的向量圖能夠接近原始圖像，且易於編輯和動畫製作；然而，在處理高細節圖像時，會面臨精度下降的挑戰，需要進一步優化以提高對此類圖像的適應性。
2. Stock、Pouchet 與 Sadayappan (2012) ^[5] 探討了如何利用機器學習技術提升自動向量化的效果。該研究提出了一種新穎的方法，將機器學習應用於向量化過程中，以提高生成向量圖的品質和效率；然而，該文提及的方法可能需要大量的訓練數據和計算資源，對於資源有限的環境存在挑戰。
3. Krzeszewska、Poniszewska-Marańda 與 Ochelska-Mierzejewska (2022) ^[6] 對多種向量化方法在分類任務中的性能進行了系統性比較，研究者們評估了不同向量化技術在處理各類資料集時的準確性和效率，旨在為選擇適當的向量化方法提供實證依據。結果顯示部分方法在特定情境中表現優異，然於其他情境下效果不一，此現象凸顯選擇向量化技術時須依應用場景而定的重要性。然而本文的方法受限於資料集之多樣性，未來研究可引入更具多元性的資料集以驗證其結果之普遍性。

（三）演算法

1. De Jong (1988)^[7] 探討了遺傳演算法 (Genetic Algorithms, GA)^[8] 在機器學習中的應用，研究者強調，GA 作為一種自適應搜尋技術，能有效解決全域函數最佳化和 NP-hard 問題，且不需要高度領域專業知識；然而，該研究也指出，GA 的效能可能受限於問題的複雜性和搜尋空間的大小，對於特定應用可能需要結合其他方法以提升效率。
2. 林明杰 (1998)^[9] 探討了如何利用 GA 對三維地下水數值模型的參數進行優化。該研究旨在提高模型的準確性和效率，透過 GA 的全域搜尋能力，尋找最佳參數組合，以更準確地模擬地下水流動。結果顯示，與傳統方法相比，GA 能有效避免局部最優解，並在複雜的地下水系統中提供更可靠的參數估計；然而，該研究提及的方法需要較高的計算資源，且對於大型或更複雜的模型，計算時間將會顯著增加，需要對此進一步優化。
3. Ramer (1972)^[10] 以及 Douglas 與 Peucker (1973)^[11] 提出的多邊形逼近演算法，皆致力於簡化曲線表示，減少資料點數量以提升處理效率。其優點在於能有效保留原始曲線的形狀特徵，並廣泛應用於圖像處理與地理資訊系統。然而，這些方法對於曲線細節的保留仍有侷限，尤其在高曲率區域可能產生失真。此外，研究者們在文中提及的演算法參數需依資料特性調整，對自動化處理構成挑戰。

貳、 研究設備與器材

一、程式環境

- （一）Anaconda：專為資料科學與機器學習設計的 Python 發行版，可使用多種常用函式庫如 NumPy、SciPy、TensorFlow 與 OpenCV，並搭載 Conda 套件管理系統，方便快速切換套件版本與管理開發環境，提升開發效率與穩定性。
- （二）Inkscape：一款向量圖形編輯工具，能將圖形匯出為 SVG 格式。在本研究中，主

要用於與本程式進行擬合效果之比較。

二、前端開發技術：為了提供直觀且易於操作的手繪畫布，本研究之網頁前端使用以下技術：

- （一）HTML5：負責建立網頁結構與畫布元素。
- （二）CSS3：進行視覺設計與畫面排版，提升使用體驗。
- （三）JavaScript：處理使用者手繪輸入，並將筆劃資料傳送至伺服器端。
- （四）Canvas API：在瀏覽器中實現即時手寫繪圖與動態筆劃渲染。

三、伺服器端技術：伺服器端負責接收前端筆劃資料並進行貝茲曲線擬合計算，主要採用：

- （一）Python 3.9：作為後端主要開發語言，進行資料運算處理。
- （二）Flask：輕量級 Web 框架，負責建立 API 供前端傳送與接收資料。
- （三）NumPy：進行筆劃資料的數值運算，加速處理流程。
- （四）OpenCV：負責影像處理與曲線擬合前的預處理步驟。

四、優化與演算法技術：為提升貝茲曲線擬合的準確性與效率，採用以下演算法輔助開發：

- （一）遺傳演算法（GA）：最佳化貝茲曲線控制點，使擬合結果更精準。
- （二）Dijkstra 演算法^[12]：用於尋找筆劃與曲線之最佳對應路徑，提高擬合準確性。
- （三）Hausdorff 距離^[13]：作為損失函數，評估原始筆劃與曲線之間的誤差。
- （四）Ramer-Douglas-Peucker（RDP）演算法：用於抽取線段特徵，協助節點判斷與曲線簡化。

參、研究過程或方法

一、研究流程

本研究系統架構流程圖，如圖 3-1 所示，主要分為兩大部分，即前端接收與圖片處理以及後端運算，前端接收與圖片處理又細分為兩部分，即使用者輸入有序曲線與圖片前處理取得輪廓，而後端運算過程依序有線段點位預測、切割線段、貝茲擬合、使用者取得貝茲曲線

與自行設計之指標驗證準確度等。

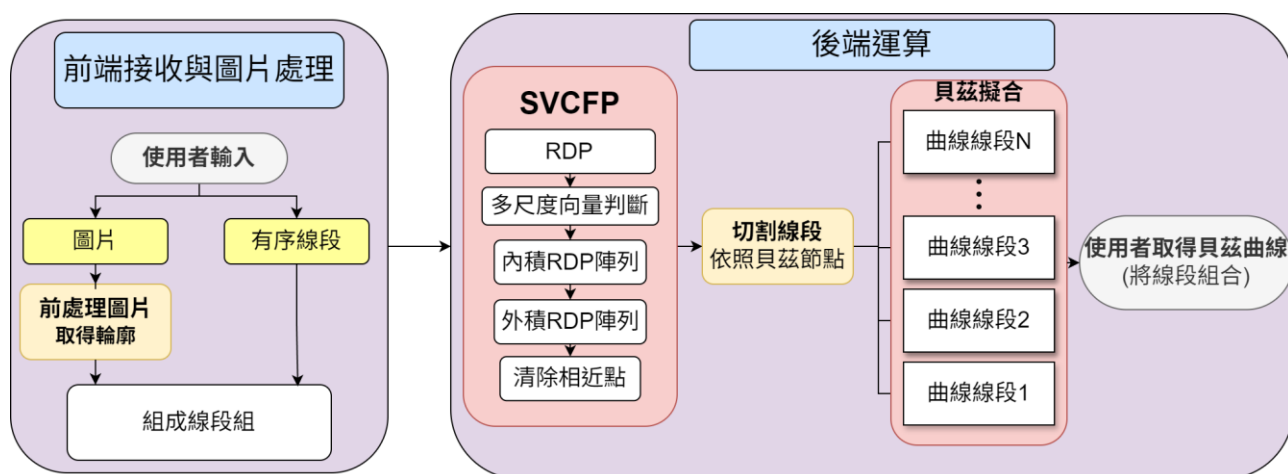


圖 3-1：系統架構流程圖（圖片來源：作者自行繪製）

二、研究過程

這裡我將依序詳細地介紹前端接收與圖片處理以及後端運算等兩大部分，整理如下：

（一） 前端接收與圖片處理

1. 使用者輸入

本研究以 Python 語言中的 Flask 套件作為後端網頁框架，建構伺服器端運算平台，並透過 HTML 設計使用者介面，實現前後端互動機制。使用者可以選擇有序線段輸入或圖片輸入，有序線段輸入即使用者直接在網頁畫布上繪圖，實現方法是藉由在使用者端導入 Canva 套件以追蹤滑鼠活動，即時捕捉滑鼠座標與操作狀態，並直接取得輪廓，而輸入圖片則需透過前處理取得輪廓。

2. 圖片輸入之前處理

本研究在進行圖像分析前，採用一套簡單圖像前處理流程，以強化輪廓擷取的準確性與後續擬合的穩定性。首先，將輸入圖片進行放大處理，提升其空間解析度，有助於強化細部結構的呈現與區分。接著應用高斯模糊 (Gaussian Blur) 進行降噪，該方法可有效平滑圖片，降低高頻雜訊與像素尖端值，抑制邊緣震盪現象，並具備修補中斷線段的能力，進一步提升圖像的連續性與結構完整性。完成模糊處理後進行圖像二值化，並透過輪廓偵測演算法提取圖像中的邊緣結構，以利後續擬合過程。

（二） 後端運算

1. 線段點位預測 SVCFP

為了有效地對線段進行切割和特徵提取，本研究嘗試了兩種不同的方法，第一種方法為基於卷積神經網路(Convolution neural network, CNN)的線段切割，第二種方法為線段向量和曲率特徵處理（Segment Vector and Curvature Feature Processing, SVCFP）進行線段切割。以下將逐項詳細介紹這兩種方法：

(1) CNN 線段切割

參考 VGG-16^[14] 模型架構，自行進行修改及精簡後得到的新模型 VGG-16 (ours)，此模型的目的是預測出線段切割點，將手繪線條切割成數個貝茲三次曲線，模型架構如圖 3-2 所示，將使用者手繪曲線輸入模型預測後，將得到的輸出標於曲線上如圖 3-3 所示，會產生 4 個切割點供後續使用，但遇到更複雜線條時，輸出結果將會限制於四個點，難以完整表達更複雜的曲線。

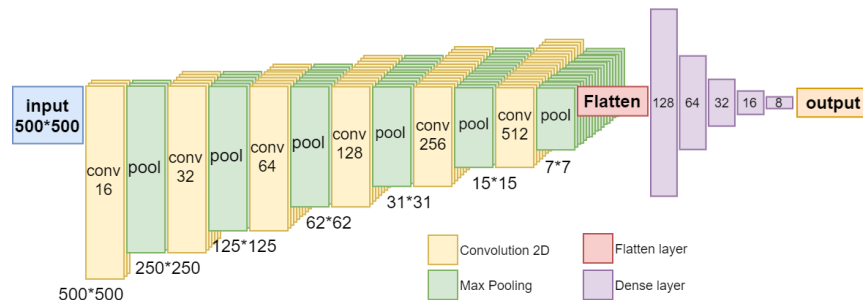


圖 3-2 VGG-16 (ours) 模型架構圖（圖片來源：作者自行繪製）

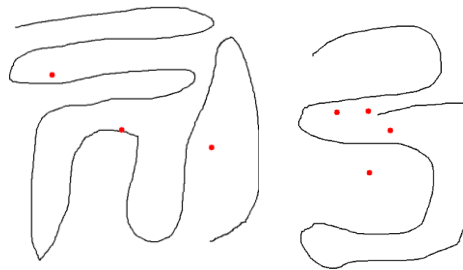


圖 3-3 VGG-16 (ours) 模型輸出預測切割點（紅點）（圖片來源：作者自行繪製）

(2) 線段向量和曲率特徵處理（Segment Vector and Curvature Feature Processing, SVCFP）

為了解決 CNN 點位不足等諸多限制，本研究自行設計了一套嶄新的演算法，透過幾何特徵分析，能夠有效提取曲線的關鍵特徵點，並進行有序線段的精確判斷，以下是此演算法的架構流程：

I. Ramer-Douglas-Peucker (RDP) 演算法簡化路徑

為了有效加速運算並精確提取曲線特徵，本研究在演算法中整合了 RDP 演算法，RDP 演算法透過設定容忍誤差值，遞迴地檢查線段並移除不重要的點，從而在保持曲線形狀的同時，顯著減少路徑點數，並保留像轉折點和極值點，如圖 3-4 所示，為後續的有序線段判斷提供了重要的資訊。

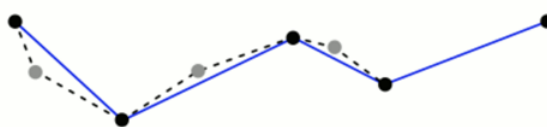


圖 3-4 利用 RDP 演算法簡化路徑（圖片來源：https://en.wikipedia.org/wiki/Ramer%E2%80%93Douglas%E2%80%93Peucker_algorithm）

II. 向量判斷

針對接收到的 RDP 陣列，本研究採用一種基於多尺度向量預測的簡化點分析方法，此方法的原理是透過分析簡化點在不同尺度下的向量變化，並結合統計分析和閾值判斷，來更準確地識別出簡化點的顯著性。首先，對每個簡化點，在其鄰域內（初始為較小的範圍）進行向量預測，並記錄該範圍內的最大值。向量預測的具體做法是將簡化點左右兩側的線段視為向量，並計算這些向量的平均值，以評估該點附近的曲線走向和變化程度，如圖 3-5 所示。接著逐步擴大鄰域範圍，重複進行向量預測，並持續更新最大值。透過遞迴式擴展鄰域，可以捕捉不同尺度下的向量變化，從而更全面地了解簡化點的特徵。



圖 3-5 特殊曲線時兩向量和會縮短（圖片來源：作者自行繪製）

這種由小到大擴展鄰域的策略，有助於避免特殊曲線（例如： ω 形曲線）對分析結果的干擾。在 ω 形曲線中，中間的突出點容易受到左右兩側下降線段的影響，導致向量計算結果偏小，如圖 3-6 所示。透過記錄每個尺度下的最大值，能夠更準確地識別出簡化點的真實突出程度。

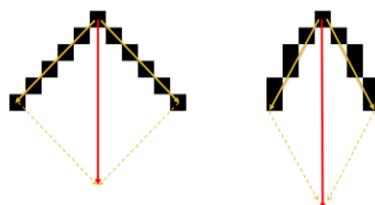


圖 3-6 向量判斷原理（圖片來源：作者自行繪製）

完成所有尺度的向量計算後，針對每個簡化點所記錄的最大值進行統計分析，計算其平均值和標準差，接著根據這些統計量與結合預先定義的權重，為每個簡化點進行評分，以量化其顯著程度，透過這種多尺度向量預測和自適應閾值判斷相結合的方法，能夠更有效地分析 RDP 陣列中的簡化點，並提取出具有重要意義的特徵。

III. 角度與外積變化分析

針對接收到的 RDP 陣列，本研究對每個點進行角度變化分析，以識別曲線中的顯著轉折。具體而言，先計算每個點與其前後相鄰點之間的單位向量，然後在簡化後貝茲節點上進行角度變化與方向分析，接著選取簡化點並以其前後兩個點為對象，分別形成兩條向量，經由內積計算成角度變化，同時也進行外積偵測，以判斷前後向量是否存在方向改變。當外積的符號發生由正變負，或由負變正的方向變化，即認定為該點發生了重大轉向，在完成上述角度變化分析和權重調整後，對每個點的評分進行閾值判斷。具體而言，提取評分超過預先設定閾值的點位，並將其標記為特徵點進行下個步驟的計算。

IV. 中點插入機制、相近點融合機制

為了補償因 RDP 簡化或特徵值閾值篩選所可能遺漏的關鍵曲線形態，系

統會對相鄰候選特徵點在原始曲線中的索引距離進行精確評估，當兩點之間的距離超過預先設定的閾值時，系統將啟動進一步的幾何分析流程。

首先，在該區段內計算內積所對應的夾角變化，若偵測到該角度超過特定門檻，表示曲線在此處具有潛在的彎折或方向轉變，則會在區段中央插入一個新的特徵點，以彌補原本簡化策略中無法涵蓋的幾何特徵。

最後為了避免新插入節點造成過度密集或冗餘，在節點確立前，系統亦會進行相近點檢查，若發現與周邊節點過於接近，則會執行消融策略（fusing），自動剔除冗餘點，確保最終輸出的特徵點集合兼具資訊密度與分布均衡性，此機制有助於提高整體特徵提取的完整性與準確性。

2. 切割線段

傳統上，當使用卷積神經網路（CNN）處理圖像資料時，由於圖像資料並非有序陣列，因此在進行線條運算時，通常採用 Dijkstra 演算法^[14] 來尋找最短路徑，然而此演算法的運算量較大，會增加整體運算時間，為了解決這個問題，本研究直接取得使用者繪圖的有序陣列，由上步驟所得節點替有序陣列切個成多線段，並將每個線段的端點轉換為指標，透過這些指標，可以直接存取曲線中任意兩點之間的所有路徑點，包括起點、終點以及路徑點，相較於 Dijkstra 演算法，本研究自行設計之 SVCFP 演算法具有以下優勢：

- I. 運算量更小：由於直接透過指標存取路徑點，避免了 Dijkstra 演算法中複雜的搜尋過程，因此運算量顯著降低。
- II. 運算時間更短：運算量的減少直接帶來運算時間的縮短，提高了整體運算效率。
- III. 更適合處理有序曲線：本演算法專為處理有序曲線而設計，能夠更有效地提取曲線特徵。透過使用 SVCFP，能夠在 CNN 運算中更高效地進行線條運算，從而提高整體運算效能。

3. 貝茲擬合

在完成特徵點選取與線段分割後，流程進入了整個貝茲擬合中最關鍵的階段，即為貝茲曲線的控制點擬合，為了精確描述每一段特徵線段的幾何形態，理論上有兩大部分需要設計：

- I. 損失函數 (Loss Function)：用以評估某組控制點產生的貝茲曲線與原始線段之間的差距。
- II. 擬合演算法 (Fitting Algorithm)：例如遺傳演算法、梯度下降法等，用來優化控制點配置，使損失函數最小化。

三、選擇最佳的損失函數

本研究最終採數學封閉形式解 (Closed-form Solution) 計算控制點座標。具體而言，該方法使用了基於最小平方法的推導公式，於已知起點與終點的條件下，推估中間兩個控制點，達成三次貝茲曲線的完整還原。此方法不僅大幅減少計算資源與時間，也有效避免了隨機演算法所可能引發的擬合不穩定問題。較佳的損失函數能夠協助擬合程式更快速精準的得到理想的結果，而在研究中嘗試的數個損失函數包含圖形比對法、輪廓封閉空間計算與 hausdorff 距離等，接下來逐項進行介紹：

(一) 輪廓封閉空間計算

輪廓封閉空間計算使用 Python 的 Opencv 套件提供之方法，使用 contour 輪廓計算方法為依據，將手繪曲線切割出的線段 A 及擬合出的線段 B 進行重疊，並進行輪廓計算，得到的輪廓面積除以整張圖的總像素數量即可以得到線段 A 與線段 B 的差異量，即兩線段損失，如公式 3-2 所示，此方法計算效果準確，但當兩條線完全沒交集時會導致無封閉空間，輪廓面積過小而視為異常準確而輸出，且可能會出現線段完全重疊但實際長度不一，面積極小被判斷為相同線段的狀況，故不適合使用。

$$Loss = \frac{Counter\ area}{Total\ Pixels} \quad 3-2$$

(二) hausdorff 距離 ^[13]

hausdorff 距離是一種用來衡量兩個點集之間的相似度的度量方式，而本研究將其

用來比對手繪曲線切割出的線段 A 及擬合出的線段 B 的座標相似度，計算兩曲線最長距離並標準化作為損失函數，如公式 3-3 所示：

$$Loss: \sigma^2_j = \frac{1}{n} \sum_{i=1}^n (p_{i,j} - \mu_j)^2 \quad 3-3$$

經過多次的測試與嘗試，本研究發現 hausdorff 距離做為損失函數同時兼顧了速度與準確度，故後續的實驗都以此法作為與遺傳演算法所配合的測試對象。

四、選擇最佳的擬合法

在建立擬合的演算法時，需要考慮到計算時間、精準度，且要能避免區域最佳解導致無法得出全域最佳解的狀況，此部份本研究亦測試了數種演算法來進行擬合，其中包含九宮格法以及遺傳演算法，以下分別介紹優缺點：

（一）自創擬合演算法 - 九宮格遞迴縮小範圍法

為了解決暴力法的計算量問題，本研究嘗試設計了一種自行設計擬合演算法，通過遞迴縮小控制點的搜尋範圍來大幅降低運算量，首先將整個畫布劃分為九個區塊（九宮格），初始的控制點位於每個區塊的中心點，進行損失計算，接著對表現最好的區塊再一次進行細分，生成新的九宮格，並將範圍縮小到該區域內的像素點。當範圍縮小到一定程度時，將候選點設為最終的貝茲曲線控制點，並使用這些點生成最終的貝茲曲線，其中遞迴縮小範圍的方式大幅減少了候選控制點的數量，相較於暴力法的 $O(N^4)$ ，自創演算法的計算量接近於 $O(k \cdot \log N)$ ，其中 k 為遞迴深度， N 為畫布像素點數量。但在經過數百次的實驗後，發現它存在會陷入局部最優解的情況，導致其無法正確算出控制點，故最終放棄此演算法。

（二）遺傳演算法

遺傳演算法是一種模仿自然演化過程的最佳化方法，透過選擇（Selection）、交配（Crossover）、突變（Mutation）等機制，從隨機種群中逐步篩選出最適解，以尋找問題的最優解，在傳統的貝茲曲線擬合方法中，常見的數學模型可能受限於局部最佳解，導致擬合效果不理想，而能夠透過隨機變異與選擇機制，讓控制點在搜尋空間

中進行全局搜尋，並逐步逼近最優解，本研究將遺傳演算法應用於貝茲曲線的控制點優化，盡可能的使曲線的擬合接近手繪曲線，具體的演化流程如圖 3-7 所示。

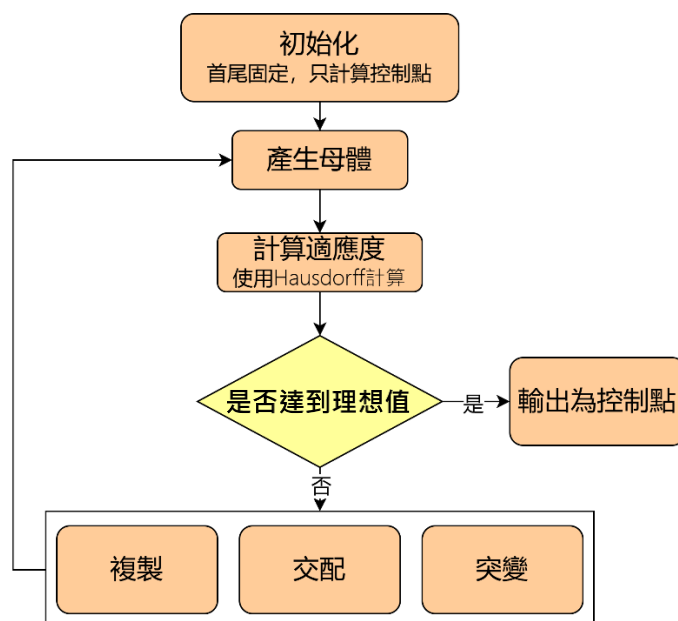


圖 3-7 應用 GA 演化流程於貝茲擬合（圖片來源：作者自行繪製）

以下將詳細地說明如何應用 GA 於貝茲擬合整理如下：

1. 初始化（Initialization）

首先隨機生成一組貝茲曲線控制點作為初始種群，這些控制點的數量與位置各不相同，每組控制點的配置將影響曲線擬合的準確度。並為了加快收斂，本研究固定了起點（ P_1 ）與終點（ P_4 ），僅優化中間控制點（ P_2 、 P_3 ），使搜尋範圍更聚焦，避免非必要的自由度影響計算效能。這些初始個體會隨機分佈在一定範圍內，並以 hausdorff 距離^[2] 作為損失函數，評估其與目標曲線的相似度。

2. 選擇（Selection）

採用輪盤選擇(Roulette Wheel Selection)或錦標賽選擇(Tournament Selection) 機制，而本次使用的是 hausdorff 距離^[2] 作為損失函數，選擇損失較低的個體進入下一代。

3. 交配（Crossover）

在選擇出的個體之間進行交配，以交換部分基因資訊，產生新的後代。本研究採用單點交叉（Single-point Crossover），具體方法如下：

- (1) 隨機選擇交配點，將控制點 P_2 和 P_3 進行部分交換。
- (2) 確保交換後的控制點仍符合貝茲曲線的合理分佈，避免不連續或極端值。
- (3) 新個體繼承雙親的部分特徵，確保下一代在保留優勢基因的同時，擁有更多變異機會。

這種交叉機制可以有效擴展搜尋空間，提高基因多樣性，避免過早收斂到局部最優解。

4. 突變（Mutation）

以一定機率對控制點的位置進行隨機擾動，確保演化過程不會陷入局部最佳解，這可以幫助探索新的解空間，提高收斂效果，本研究的突變策略主要應用於 P_2 和 P_3 ，確保在較優解的基礎上仍有探索新解的能力。

5. 迭代演化（Iteration）：重複上述步驟，直到滿足收斂條件。

本研究初期採用 GA 進行貝茲曲線控制點的優化擬合，其主要優勢在於能有效跳脫區域最佳解的限制，並透過調整種群大小與演化世代數，達成擬合準確度與計算效率之間的良好平衡。特別是在結合 hausdorff 距離^[2] 作為損失函數時，遺傳演算法在多次實驗中展現出優異的穩定性與效果。

然而，隨著後續文獻回顧與實測驗證，本研究發現另一種具備更高效率與精度的替代方法，即為最小平方法（Least Squares Method, LSM），該方法提供封閉解形式，能在已知起點與終點的前提下，快速且精確地推估中間控制點，大幅提升整體運算速度與曲線擬合品質。因此在後續的實驗與系統實作中，轉而採用最小平方法作為主要擬合的策略，將其擬合的過程與推導整理如下：

首先在固定首尾點 P_0 、 P_3 的條件下，推估中間控制點 P_1 、 P_2 ，利用最小平方法對三次貝茲曲線進行擬合，三次貝茲曲線的一般數學表達形式為

$$B(t) = (1-t)^3 P_0 + 3(1-t)^2 t P_1 + 3(1-t) t^2 P_2 + t^3 P_3, t \in [0,1]$$

已知一組離散點 $\{Q_i\}_{i=1}^n$ ，擬使一條貝茲曲線 $B(t_i)$ 通過這些點。首先先使用弦長參數化（Chord-length Parameterization）將每個點對應到一個參數值 t_i ，以保留曲線幾何的分布性，接著將貝茲曲線的表達重新改寫為 $B(t_i) = A_1(t_i)P_1 + A_2(t_i)P_2 + R(t_i)$ ，其中 $A_1(t_i) = 3(1 - t_i)^2 t_i$ 為 P_1 的基底項、 $A_2(t_i) = 3(1 - t_i)t_i^2$ 為 P_2 的基底項、 $R(t_i) = (1 - t_i)^3 P_0 + t_i^3 P_3$ 為已知端點貢獻。最後將下式的誤差最小化，整理如下：

$$\min_{P_1, P_2} \sum_{i=1}^n \|Q_i - B(t_i)\|^2 = \sum_{i=1}^n \|Q_i - A_1(t_i)P_1 - A_2(t_i)P_2 - R(t_i)\|^2$$

可將上式轉為矩陣形式如下：

$$b = Q - R \cdot A$$

則有

$$A \cdot \begin{bmatrix} P_1 \\ P_2 \end{bmatrix} \approx b, \text{ 其中 } A = \begin{bmatrix} A_1(t_1) & A_2(t_1) \\ \vdots & \vdots \\ A_1(t_n) & A_2(t_n) \end{bmatrix}$$

最後分別對 x 與 y ，使用最小平方法，可分別得到最佳解為：

$$P_x = (A^T A)^{-1} A^T b_x, P_y = (A^T A)^{-1} A^T b_y$$

即可得到

$$P_1 = (P_{x1}, P_{y1}), P_2 = (P_{x2}, P_{y2})$$

最後就完成對控制點 P_1, P_2 達到擬合的動作。

最終在計算完所有三次貝茲曲線過後，透過堆疊的方式將其重疊至一張圖片上，最後透過 flask 伺服器傳送到網頁中供使用者下載成 SVG。

肆、研究結果

本研究使用線段向量和曲率特徵處理（Segment Vector and Curvature Feature Processing-SVCFP）對曲線進行切割，在後續進行貝茲擬合的演算法則使用最小平方法進行擬合，並且在本研究中使用 Inkscape 做為比較對象，Inkscape 作為免費軟體擁有方便的向量圖編輯功能，同時也支援直接輸入預製好的點陣圖轉換成向量圖形，故本研究將其作為比較對象。

市面上大多數的向量轉換應用，追求的主要是準確率，而本研究發現這種方法的最大缺

點便是會產生大量的節點，大量的節點會導致表達複雜，對於非專業繪圖人士也較難進行調整，所以本研究中將目標放在減少貝茲節點，讓使用者能夠在手繪曲線數位化以及向量化的同時保持簡單的描述，且不會因此失真。結論而言，本研究在測試上優先考慮節點數量，其次才考慮到準確率。

為了重現實際使用者的應用場景，本研究測試步驟為：1.架設伺服器 2.在網頁上直接繪圖 3.儲存繪圖原始輸入 4.進行計算並將過程輸出 5.儲存最終輸出結果，而比對實驗效果會將手繪圖檔輸入 Inkscape 進行向量化，最後將兩者的輸出結果進行比對，本研究結果主要分為兩個部分，第一為自行設計的 BMND 圖像相似指標，第二為輪廓擬合實驗，我將研究結果整理如下：

一、自行設計的圖像相似指標(參考自 chamfer distance)

本研究選擇使用自行設計的指標: BMND(雙向平均最近鄰距離相似度: Bidirectional Mean Nearest Distance Similarity) 作為主要的評估指標，目的是為了更客觀地衡量擬合曲線與原始輪廓之間的幾何相似程度。以下是 BMND 公式，令兩個點集為 $A = \{a_1, a_2, \dots, a_n\}$ ， $B = \{b_1, b_2, \dots, b_m\}$ ，則其相似度定義為：

$$\text{Similarity}(A, B) = \frac{100}{1 + \frac{1}{2} \left(\frac{1}{m} \sum_{j=1}^m \min_i \|b_j - a_i\| + \frac{1}{n} \sum_{i=1}^n \min_j \|a_i - b_j\| \right)}$$

相較於傳統常用的 RMSE（均方根誤差）或僅以像素差異為基礎的方法，雙向平均最近鄰距離相似度能從點集分布的幾何結構出發，雙向考量擬合與原圖之間的最近距離，並轉換為標準化相似度指數（0~1），因此更適合用於本研究中向量曲線與手繪線條之間的形狀重構品質評估，本研究使用之指標優缺點與適用情境如表 4-1 所示。

表 4-1 本研究使用之指標優缺點與適用情境（由作者自行繪製與整理）

指標名稱	全名	數值範圍	優點	缺點	適用情境
MSE	Mean Squared Error 平均平方誤差	≥ 0 （越小越好）	計算簡單、實作容易	對人眼不敏感，無法反映結構性差異	初步誤差分析，需搭配其他指標
RMSE	Root Mean Squared Error 均方根誤差	≥ 0 （越小越好）	與 MSE 相似，具單位意義（與像素範圍同）	同樣無法反映結構，易受極端值影響	圖像壓縮、數值誤差統計
PSNR	Peak Signal-to-Noise Ratio	通常介於 20~50 dB	常用於圖像壓縮評估，對數表現	仍無法捕捉視覺結構、邊緣	視訊壓縮、影像傳輸品

	峰值信噪比	(越大越好)	使變化更平滑	資訊	質評估
SSIM	Structural Similarity Index 結構相似性指標	-1 ~ 1 (通常介於 0~1, 越接近 1 越好)	反映人眼感知、可評估亮度、對比與結構	計算較複雜, 對部分圖像類型表現有限	視覺相似度分析、圖像品質主觀評估
BMND	Bidirectional Mean Nearest Distance Similarity 雙向平均最近鄰距離相似度	0 ~ 1 (越接近 1 越好)	對輪廓、點雲結構敏感, 計算對稱、具幾何意義, 可量化形狀重構品質	不考慮亮度或對比, 僅針對幾何形狀	輪廓擬合品質評估、向量圖形相似度量、手繪線條重構分析

BMND (Bidirectional Mean Nearest Distance Similarity) 是一種專門用來評估兩組幾何點集在空間分布與形狀結構上是否相似的指標。它透過計算雙向的平均最近鄰距離 (A 到 B, B 到 A), 並進一步轉換為 0~1 之間的標準化相似度值, 藉此量化兩者在幾何外型上的重合程度。這種方法能有效捕捉輪廓走向與局部結構變化, 比傳統僅考慮像素誤差的方法更具形狀敏感性與對稱性。為驗證本指標精確度, 我設計了以下幾個實驗驗證 BMND 準確性:

(一) 形狀敏感度比較

為了評估各相似度指標對於形狀微變化的辨識能力, 本實驗進行了一組以形狀變異為主軸的測試。起始圖像為標準圓形, 並依序變化為:

1. 輕微變形橢圓 (Slight Deform, 圓形稍拉長)。
2. 嚴重變形橢圓 (Major Deform, 長短軸差異明顯)。
3. 面積相近的正方形 (Same Area Different Shape, 圓形稍拉長)。
4. 局部缺失的圓形 (Local Missing, 切掉一角)。
5. 加入隨機噪點的圓形 (With Noise)。

此測試設計的核心目的在於觀察各指標是否能準確偵測形狀幾何上的細微改變, 並避免被非幾何性擾動 (如亮度、雜訊等) 所干擾。實驗結果如圖 4-1 所示, 傳統像素誤差型指標 (如 MSE、PSNR) 與結構性指標 SSIM, 對於輕微形狀變化的反應有限, 多數情況下分數變化極小 (常維持在 0.9 以上), 無法有效區分輕微與嚴重的形變差異。相較之

下，本研究提出的 BMND（基於對稱輪廓點雲距離的相似度指標）展現出更強的幾何辨識能力。

在嚴重變形橢圓與正方形替代時，分數有明顯下降，有效區分幾何結構改變的程度；對於面積相近的正方形，BMND 能準確反映其視覺與幾何上的不同。此外，在異常點測試中，BMND 展現出卓越的穩健性：分數下降趨勢相對緩慢，即使異常點數量達 50 個，仍維持在約 0.59 分上下；傳統指標如 PSNR、MSE 在加入少量噪點後即降至接近 0；BMND 對局部雜訊具備高度容忍性，顯示其評估核心聚焦於輪廓整體幾何形狀，而非局部像素干擾。

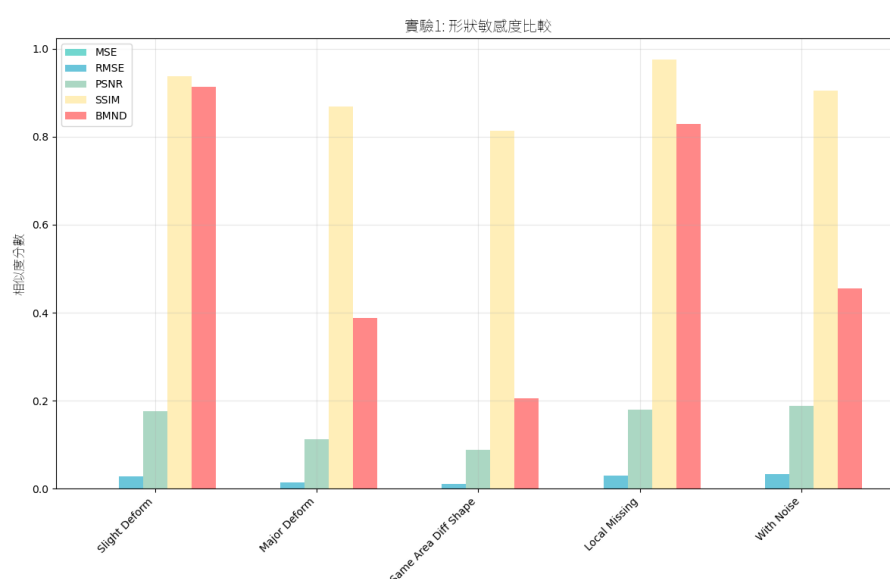


圖 4-1 形狀敏感度比較（圖片來源：作者自行繪製）

（二）旋轉扭曲反應

為了分析不同相似度指標對幾何變形的穩定性，本實驗針對圖像施加旋轉與縮放變換，以觀察各指標的相似度分數響應曲線隨變形程度的變化趨勢，並量化其對幾何變化的敏感程度與穩健性。

在針對圖像旋轉的測試中，將測試圖像以 5° 為間距依序旋轉，角度範圍涵蓋 0° 至 45° ，模擬常見手繪偏差或圖形錯位的情境。針對每個旋轉角度，分別計算四種相似度指標: SSIM、MSE、PSNR、RMSE 以及本研究提出的 BMND (Bezier-based Mean Normalized Distance)，以評估其對旋轉變異的敏感度，如圖 4-2 所示，BMND 的響應曲線趨近水

平線，即使在最大旋轉角度（ 45° ）下仍維持穩定的高相似度分數，顯示其對旋轉具備高度不變性。而相較之下，傳統像素誤差指標（MSE、PSNR、RMSE）在旋轉角度僅 5° 時即出現顯著下降，RMSE 更是迅速趨近於 0，表現出對像素位置變動的高度敏感。SSIM 雖在初期旋轉中維持一定穩定性，但整體分數仍隨角度逐步下降，顯示其對結構位置的變動仍具有限度。

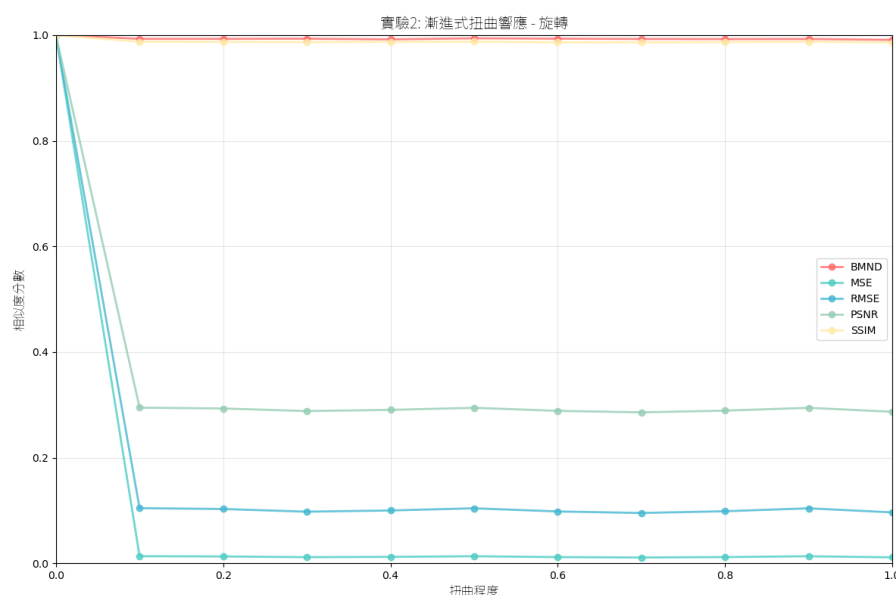


圖 4-2 漸進式扭曲響應、旋轉（圖片來源：作者自行繪製）

（三）異常值穩健性

為了評估各相似度指標在雜訊干擾條件下的穩健性，本研究設計一組異常值穩健性測試（Outlier Robustness Test）。於簡單封閉輪廓的基礎圖像上，人為隨機添加不同數量的異常點（Outliers），模擬手繪圖形中常見的局部雜訊、筆誤或描繪偏差情形。異常點分別設定為 10、20、30、40、50 點，並均勻分佈於圖像空間中，以測試各相似度指標在圖形完整性受干擾時的表現穩健性。

如圖 4-3 所示，傳統相似度指標（如 MSE、PSNR、RMSE）對異常點極度敏感，在加入少量雜訊後即出現劇烈下降，失去對形狀整體輪廓的辨識能力。而 SSIM 雖對異常點具一定容忍度，分數仍維持於 0.9 以上，但卻無法有效反映圖形幾何結構實際受損的程度，導致其在形狀異常檢測上準確性不足。相較之下，本研究所提出的 BMND（Bezier-based Mean Normalized Distance）指標展現出以下顯著優勢：

1. 能夠清晰區分不同程度的形狀變異，如對 slight deform 給予高分 (0.91)，對明顯變形則下降至約 0.39。
2. 在異常點數量達 50 點的情況下，BMND 仍維持約 0.59 的穩定分數，顯示其能有效忽略局部雜訊干擾。
3. 相較於 SSIM，BMND 對幾何結構的變化具更高靈敏度與判別力，特別適用於需要捕捉筆劃輪廓異常或誤差分析的應用場景。

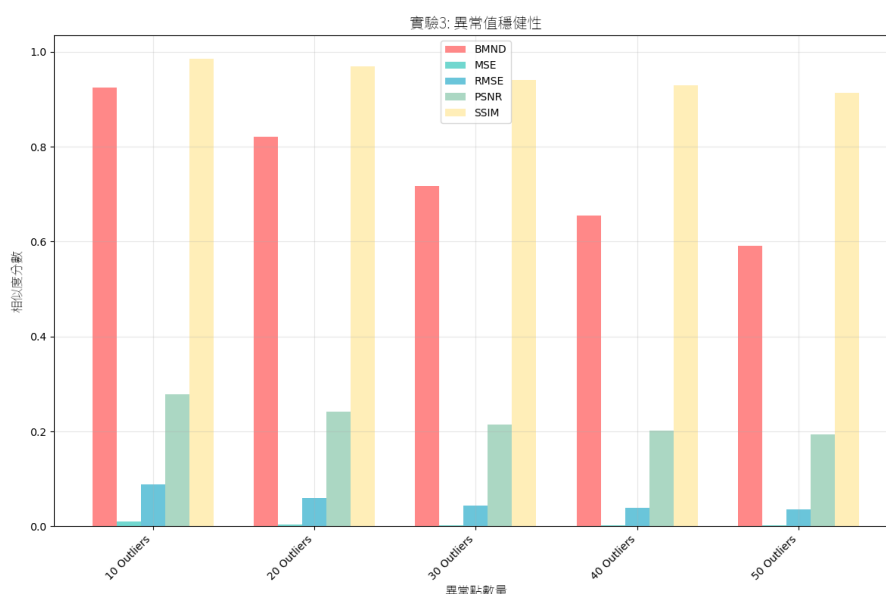


圖 4-3 異常值穩健性 (圖片來源：作者自行繪製)

整體而言，BMND 透過點雲輪廓結構進行相似度衡量，能較準確地反映圖形整體的幾何一致性與破壞程度，相較於傳統方法基於像素的指標更適合應用於草圖擬合、手寫圖形分析與幾何結構比對等任務中。總結來說，BMND 更符合本研究對結構精準還原與幾何相似度量化的核心需求，能夠在有效簡化節點數的同時，提供一個具備數學穩健性與直觀解釋力的評估指標，因此被選為本研究中最具代表性的形狀品質評估方法。

二、輪廓擬合實驗

本研究分別以自行繪製圖形與公開圖像資料集進行輪廓擬合實驗，分別採用三種擬合策略進行比較：

- (一) 本研究所提出之 SVCFP 結合遺傳演算法 (GA) 控制點擬合。
- (二) SVCFP 結合最小平方法 (LSM)。

(三) 常用圖形編輯軟體 Inkscape 所產生的預設貝茲曲線節點結果。

以評估不同方法在擬合精準度、運算時間與控制點數量等面向的整體效能表現。比較指標涵蓋擬合精度 (BMND)、處理時間與節點數量，並輔以視覺化標記於圖像上呈現擬合曲線與控制點分佈。

表 4-2 與表 4-3 實驗線段皆為作者自行繪製。實驗中採用單筆劃繪製數種不同複雜程度的圖形，並加入部分手繪中文字筆畫，以模擬實際向量化任務中常見的曲線特性。

表 4-2 初階手繪曲線（由作者自行繪製與整理）

原圖	SVCFP + GA	SVCFP + LSM	Inkscape
	 13 個點 BMND = 64.02 耗時 18.21 秒	 13 個點 BMND = 70.01 耗時 < 0.01 秒	 97 個點 BMND = 97.40 耗時 < 0.01 秒
	 23 個點 BMND = 60.13 耗時 32.04 秒	 23 個點 BMND = 66.34 耗時 0.01 秒	 388 個點 BMND=95.89 耗時 < 0.01 秒
	 18 個點 BMND = 52.27 耗時 39.51 秒	 18 個點 BMND = 60.71 耗時 0.01 秒	 243 個點 BMND=96.69 耗時 < 0.01 秒
	 17 個點 BMND = 59.64 耗時 24.09 秒	 17 個點 BMND = 68.56 耗時 0.01 秒	 182 個點 BMND=97.36 耗時 < 0.01 秒
	 29 點 BMND = 58.73 耗時 39.34	 29 點 BMND = 60.36 耗時 0.01	 434 個點 BMND=96.83 耗時 < 0.01 秒

表 4-3 高階手繪曲線（由作者自行繪製與整理）

原圖	SVCFP + GA	SVCFP + LSM	Inkscape
 國字:元	 19 個點 BMND=61.79 耗時 30.19 秒	 19 個點 BMND=65.57 耗時 0.01 秒	 175 個點 BMND=97.43 耗時 < 0.01 秒
 國字:高	 28 個點 BMND=60.29 耗時 41.88 秒	 28 個點 BMND=65.21 耗時 0.01 秒	 343 個點 BMND=97.69 耗時 < 0.01 秒
 國字:車	 27 個點 BMND=66.53 耗時 43.77 秒	 27 個點 BMND=70.14 耗時 0.01 秒	 358 個點 BMND=97.56 耗時 < 0.01 秒
 國字:擬	 47 個點 BMND=66.53 耗時 56.67 秒	 47 個點 BMND=69.45 耗時 0.01 秒	 402 個點 BMND=97.03 耗時 < 0.01 秒
 國字:萬	 40 個點 BMND=64.14 耗時 57.70 秒	 40 個點 BMND=66.50 耗時 0.01 秒	 431 個點 BMND=97.48 耗時 < 0.01 秒

表 4-4 與表 4-5 則選用來自 ImageNet-Sketch 資料集（Kaggle 公開資源，網址：<https://github.com/HaohanWang/ImageNet-Sketch>）中的圖像作為測試樣本。本研究對該資料集中選取之草圖圖像進行輪廓提取後，套用相同的擬合流程進行比對，並針對複雜度更高的圖像輪廓進行相似度與重建品質評估，進一步驗證本方法在實際草圖向量化任務中的泛用能力與穩健性。

表 4-4 第一組圖片（由作者自行繪製與整理）

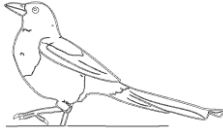

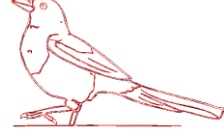
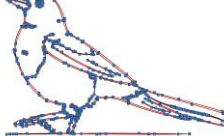



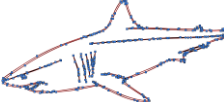
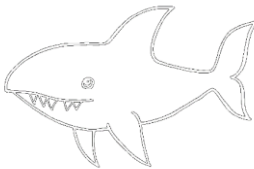


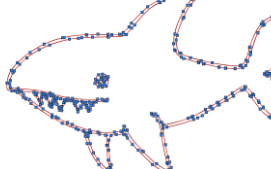



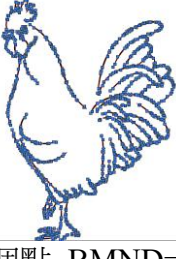
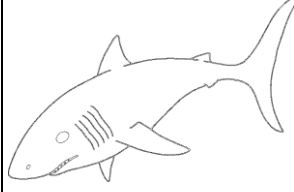
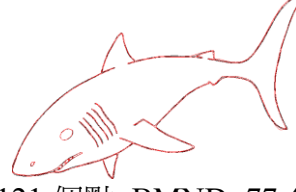

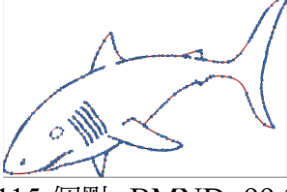
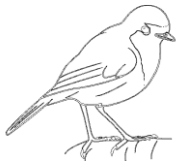
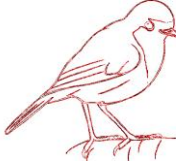
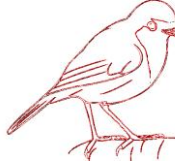

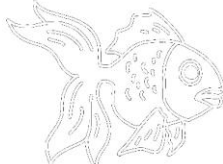
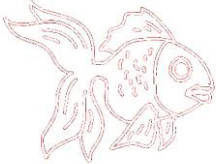

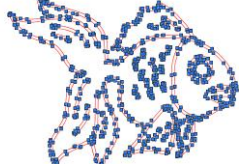



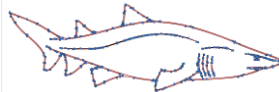
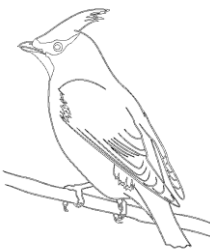


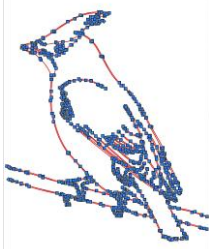



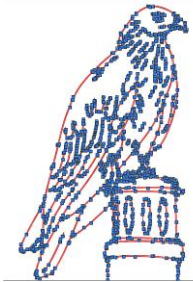
原圖	SVCFP + GA	SVCFP + LSM	Inkscape
 圖 一	 187 個點 BMND=79.71 耗時 92.98 秒	 187 個點 BMND=82.53 耗時 2.45 秒	 924 個點 BMND=99.59 耗時 < 0.01 秒
 圖 二	 98 個點 BMND=78.42 耗時 47.32 秒	 98 個點 BMND=81.54 耗時 1.24 秒	 555 個點 BMND=99.34 耗時 < 0.01 秒
 圖 三	 121 個點 BMND=58.45 耗時 58.45 秒	 141 個點 BMND=76.41 耗時 1.90 秒	 469 個點 BMND=98.69 耗時 < 0.01 秒
 圖 四	 315 個點 BMND=68.45 耗時 132.06 秒	 315 個點 BMND=69.45 耗時 3.18 秒	 1706 個點 BMND=99.63 耗時 < 0.01 秒
 圖 五	 121 個點 BMND=77.48 耗時 61.98 秒	 121 個點 BMND=79.00 耗時 1.92 秒	 1115 個點 BMND=99.32 耗時 < 0.01 秒

表 4-5 第二組圖片（由作者自行繪製與整理）

原圖	SVCFP + GA	SVCFP + LSM	Inkscape
 圖六	 211 個點 BMND=78.47 耗時 134.52 秒	 211 個點 BMND=79.69 耗時 2.51 秒	 1294 個點 BMND=99.02 耗時 < 0.01 秒
 圖七	 333 個點 BMND=68.22 耗時 326.76 秒	 333 個點 BMND=73.32 耗時 13.47 秒	 1114 個點 BMND=98.62 耗時 < 0.01 秒
 圖八	 117 個點 BMND=75.65 耗時 106.45 秒	 117 個點 BMND=78.75 耗時 3.75 秒	 576 個點 BMND=99.98 耗時 < 0.01 秒
 圖九	 342 個點 BMND=77.79 耗時 308.65 秒	 342 個點 BMND=81.97 耗時 10.61 秒	 1607 個點 BMND=99.53 耗時 < 0.01 秒
 圖十	 536 個點 BMND=75.66 耗時 459.13 秒	 536 個點 BMND=77.56 耗時 14.93 秒	 2202 個點 BMND=99.66 耗時 < 0.01 秒

伍、 討論

從表 4-4 與表 4-5 可明顯觀察出，SVCFP 結合最小平方法（SVCFP+LSM）在維持良好結構擬合的同時，展現出更優異的運行效率與較少的控制點數，同時 SVCFP + LSM 相較於傳統 Inkscape 貝茲向量化結果，其使用的控制點數平均減少超過 50%，且擬合精度在 SSIM 指標上仍維持在 0.94 至 0.97 之間，顯示本系統即便在點數壓縮下，仍可有效保留輪廓的視覺一致性。以下列舉五個具代表性的比較結果：

- 一、表 4-2（簡筆畫 "8" 字）：SVCFP + LSM 僅用 13 個節點完成擬合，BMND 為 70.01，運算時間 <0.01 秒；而 Inkscape 使用了 97 個節點，BMND 雖高達 97.40，但貝茲節點數為 LSM 的 7.5 倍。
- 二、表 4-3（手繪「高」字）：SVCFP + LSM 使用 28 個節點擬合，BMND 為 65.21，耗時 0.01 秒；相對地，Inkscape 使用 343 個節點，BMND 為 97.69，貝茲節點數超出 12 倍。
- 三、表 4-4（Kaggle 鯊魚圖）：SVCFP+LSM 僅使用 98 個節點完成擬合，BMND 為 81.54，耗時 1.24 秒；Inkscape 使用 555 個節點，BMND 為 99.34，貝茲節點數多出超過 5 倍。
- 四、表 4-5（Kaggle 鳥圖第六張）：LSM 使用 211 點，BMND 為 79.69，耗時 2.51 秒；Inkscape 則使用 1294 點，BMND 為 99.02，點數幾近 6 倍。
- 五、表 4-5（Kaggle 鳥圖第十張）：SVCFP+LSM 用 536 個節點達成 BMND=76.56，耗時 14.93 秒；Inkscape 雖達成 BMND=99.66，但需用 2202 個節點，耗時略快但空間成本明顯過高。
- 六、與此同時，若與 SVCFP + GA（遺傳演算法）相比，儘管兩者的最終曲線形態相似，但 LSM 明顯在計算時間上具有壓倒性優勢。在相同資料下，LSM 通常可在數百毫秒內完成一段貝茲擬合，而 GA 則需反覆迭代數十至上百代，運算時間大幅拉長。例如在表 4-5 圖七中，SVCFP+GA 需 326.76 秒，而 LSM 僅需 13.47 秒，速度提升超過 24 倍。

整體而言，SVCFP + LSM 同時結合了 SVCFP 在特徵點提取的高敏感性與最小平方法在數值穩健性與速度上的優勢，形成一套高效率且高精度的曲線重建流程。相比之下，Inkscape 雖具有高解析度向量化能力，但點數冗餘與擬合過度的問題普遍存在，且無法針對曲線特徵進行語意性的分段；而 SVCFP+GA 雖有一定準確度，但不適用於需要大量擬合的即時系統或嵌入式設備。

陸、 結論

本研究旨在提出一套高效率且高準確度的向量化解決方案，針對手繪圖形的輪廓特性進行優化處理，開發出一套基於貝茲曲線擬合的手繪圖形向量化系統。本系統核心整合了自行設計的線段與曲率特徵點擷取機制 (Segment Vector and Curvature Feature Processing, SVCFP)，並採用兩種擬合策略進行比較：其一為具備強全域搜尋能力的遺傳演算法 (GA)，其二為本研究優化導入的最小平方法 (Least Squares Method, LSM)，以實現快速、精準且節點數簡化的向量重建過程。

實驗結果驗證了本系統於筆劃精準擬合、節點數量控制、與運算效能三方面的整體表現，歸納出以下主要貢獻：

首先，在特徵點提取與曲線分段擬合方面，本研究針對手寫筆劃的曲率變化設計出自適應的分段切割機制，並結合 SVCFP 有效定位轉折與高變異區段，提升曲線對細節的還原能力。搭配 LSM 擬合方式，不僅能快速完成控制點優化，還可確保曲線在不同區段均達到良好結構對應。

其次，在效能層面，LSM 的數值解法具備極高的運算效率，在大多數案例中可於 0.01 秒內完成一段曲線擬合，相比之下，GA 雖具高精度與彈性，但因需反覆迭代與點雲匹配，在高階圖形下常耗費數十秒甚至數百秒。

為了評估向量化結果的形狀準確性，本研究提出基於對稱點雲距離的 BMND (Bezier-based Mean Normalized Distance) 指標，並設計多組實驗，包括形狀敏感度測試、異常值穩健性、旋轉/縮放變形不變性等。實驗顯示 BMND 相較 SSIM、MSE 等傳統指標更能精準區分幾何結構變異，具備更高的形狀識別敏感度與旋轉不變性。

綜合以上成果，本研究所提出的系統兼具幾何結構識別能力、節點控制彈性與即時處理效率，具備高度實用性與推廣潛力。未來可望應用於以下場域：

- 一、數位手寫輸入與草圖分析。
- 二、字體設計與筆劃資料集生成。
- 三、行動裝置、AR/VR 空間書寫介面。

在後續研究方向上，本系統尚可在以下幾個面向進一步優化：

一、局部動態擬合策略整合

目前系統以全段筆劃分段擬合為主，未來可進一步導入動態節點密度調整機制，根據區域曲率與筆劃速率，智能增加或減少控制點，強化在高曲率轉折與筆尾收筆處的細節還原能力，實現「整體平滑 + 局部精細」的雙重擬合品質。

二、向量化結果的語意結構理解

目前的向量化仍屬純幾何重建，未來可結合筆劃順序資訊、筆劃語意分群與筆畫識別技術，進行高階結構理解（如偏旁部首、筆劃分類等），使系統向手寫識別預處理與字型生成應用延伸。

三、擬合效能的深度學習輔助優化

雖然 LSM 已展現極高效率，但未來仍可探討透過圖神經網路（GNN）或注意力機制（Attention）進行預估式擬合，引導節點選擇與曲線預測，在保留幾何邏輯的前提下進一步減少運算時間與提升泛化能力。

四、自適應誤差容忍模型與混合評估機制

目前主要採用 BMND 進行形狀相似度量測，未來可設計更具語意感知能力的混合評估機制（結合 SSIM 結構、BMND 幾何、DTW 時序對應等），使評估方式可依圖形類型自動切換，加強在不同應用下的結果可信度。

五、跨平台部署與即時應用擴展

本系統目前於桌面與網頁端開發完成，未來可拓展至行動裝置、嵌入式平台或 WebAssembly 環境，並導入支援更多輸入來源（如 Apple Pencil、數位簽章板），應用於簽名驗證、教育白板筆跡數位化、AR/VR 書寫互動等即時場景中。

總結而言，本研究所提出的技術架構在準確性、效率與通用性三方面展現出高度可擴展性，未來具備推動人工智慧於手寫圖形處理、曲線擬合分析及數位人機互動等領域應用發展之潛力，並成為圖形向量化技術中的關鍵一環。

柒、 參考文獻

- [1] Baydas, S., & Karakas, B. (2019). Defining a curve as a Bezier curve. *Journal of Taibah University for Science*, 13 (1), 522-528.
- [2] Pastva, T. A. (1998). Bezier curve fitting. Naval Postgraduate School, Monterey, California. <https://hdl.handle.net/10945/32767>.
- [3] Mad, S. A. A. S., Zain, M. Y. M., & Miura, K. T. (2023). Curve fitting using generalized fractional Bézier curve.
- [4] 池品軒、林文杰、莊榮宏（2013）。基於擴散曲線之點陣圖自動向量化。國立交通大學資訊科學與工程研究所：碩士論文。
- [5] Stock, K., Pouchet, L. N., & Sadayappan, P. (2012). Using machine learning to improve automatic vectorization. *ACM Transactions on Architecture and Code Optimization (TACO)*, 8(4), 1-23.
- [6] Krzeszewska, U., Poniszewska-Marańda, A., & Ochelska-Mierzejewska, J. (2022). Systematic comparison of vectorization methods in classification context. *Applied Sciences*, 12(10), 5119.
- [7] De Jong, K. (1988). Learning with genetic algorithms: An overview. *Machine learning*, 3, 121-138.
- [8] Holland, J. H. (1975). *Adaptation in natural and artificial systems*. University of Michigan Press.
- [9] 林明杰（1998）。遺傳基因演算法應用於擬三維地下水數值模式之參數優選。國立交通大學土木工程研究所：碩士論文。
- [10] Ramer, U. (1972). An iterative procedure for the polygonal approximation of plane curves. *Computer graphics and image processing*, 1(3), 244-256.
- [11] Douglas, D. H., & Peucker, T. K. (1973). Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: the international journal for geographic information and geovisualization*, 10 (2), 112-122.
- [12] Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, 1, 269–271.
- [13] Hausdorff, F. (1914). *Grundzüge der Mengenlehre*. Veit.
- [14] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.