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

摘要

本研究提出一套基於貝茲曲線的手寫圖形向量化系統，透過自動節點擷取與分段擬合機制，在降低點位數的同時維持高度幾何保真，本研究整合了自創的線段向量與曲率特徵處理演算法（SVCFP），能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，整體流程建置於網頁互動介面，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，本研究在控制點數上平均減少約 84.6%，最高可達 90.8% 壓縮比，同時僅犧牲較少精準度情況下（BMND分數平均變動約28~36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，如數位手寫輸入、字型設計與圖形分析等，為圖形向量化提供一種高效率、低冗餘、視覺保真且實用價值高的解決方法。

研究動機

在數位繪圖中，將手繪圖形轉換為精確的數位格式是一項挑戰，目前手繪的筆劃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響，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。傳統方法常使用鋼筆工具和貝茲曲線，但存在貝茲曲線節點過多導致檔案過大、運算效率降低或人工調整節點耗時高的問題。因此，本研究希望能夠開發出一套具備高效率、精確且自動化的手繪向量化方法，進而提升數位手繪技術的實用性與其應用。

研究目的

(一) 改善傳統貝茲曲線擬合節點過多問題，用更少的節點達到相同或更高的擬合精度，從而減少計算量與降低儲存的成本。

(二) 結合筆劃長度與曲率動態的分析，**利用自創演算法於自動調整貝茲曲線的節點數量與位置**，能減少大量點位下繼續保留手繪筆劃的細微變化與特徵完整性。

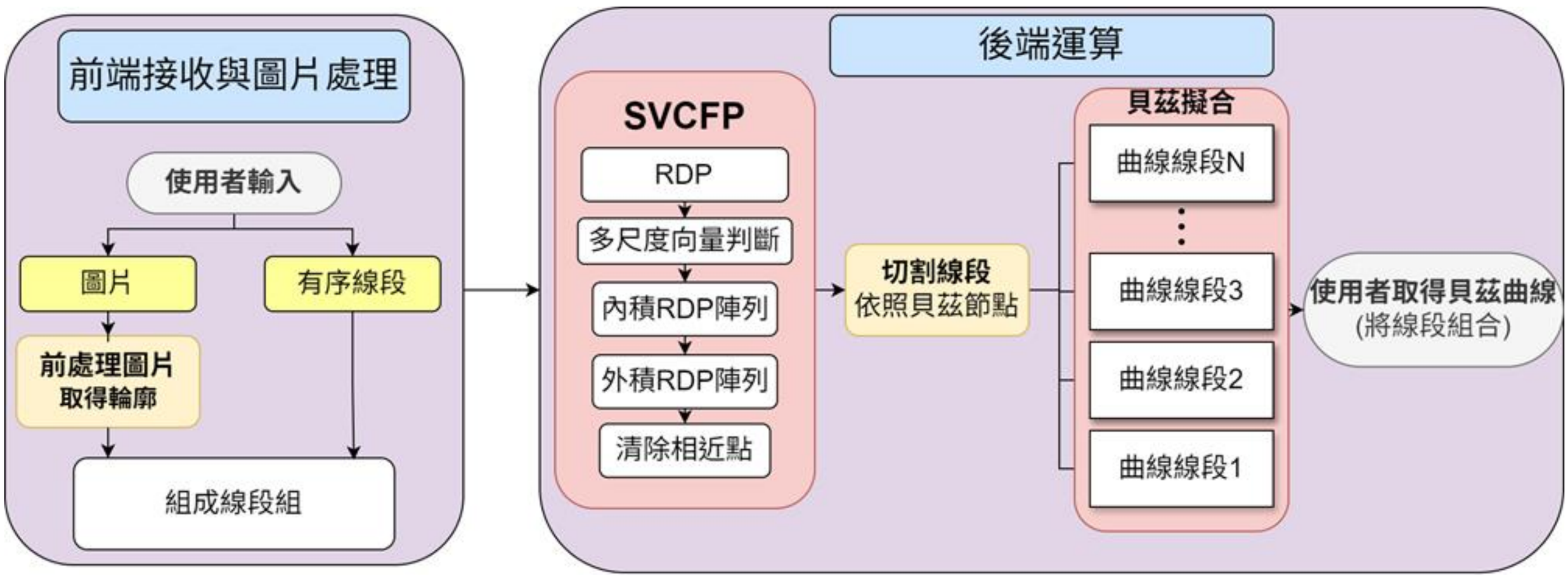
(三) 透過演算法自動化選擇貝茲節點與擬合曲線，使用者無需進行手動調整，即可獲得高品質的向量化結果。

研究設備與器材

硬體部分： 筆記型電腦、桌上型電腦。配備：Intel Core i7。
軟體部分： Python 3.X函數、計算與繪圖等套件庫、HTML5、CSS3、JavaScript、Canvas API。

研究過程或方法

一、研究架構圖



▲研究流程圖 (作者自行繪製)

二、前端接收與圖片處理

(一) 使用者輸入：利用Python的Flask套件作為後端網頁框架，搭建了伺服器端運算平台，並結合 HTML 介面實現前後端互動。導入 Canva 套件以追蹤滑鼠活動，捕捉其座標與操作狀態。

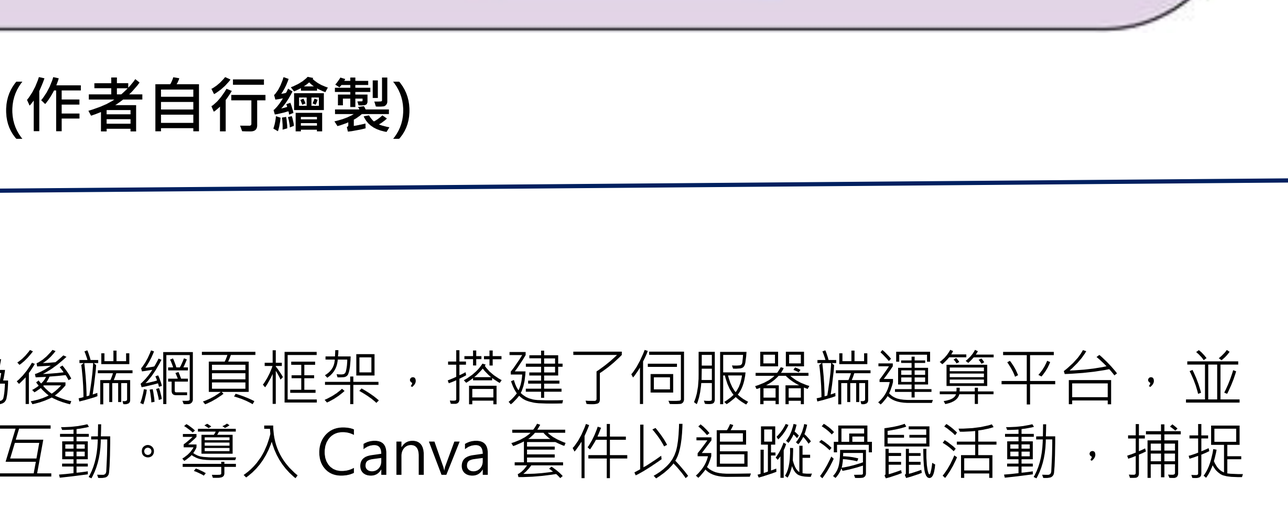
(二) 前處理圖片：圖片會經過一系列前處理，依序為放大圖像、高斯模糊與二值化。

(三) 取得輪廓圖：最後進行輪廓偵測演算法，提取出清晰的邊緣結構。以利後續擬合過程。

三、後端運算 – 線段點位預測

(一) CNN線段切割

- 修改VGG-16模型得新模型VGG-16 (ours)用來預測出線段切割點，將手繪線條切割成數個貝茲三次曲線。
- 手繪曲線輸入模型預測後，得到的輸出標於曲線上，並產生 4 個切割點供後續使用。
- **缺點**：處理複雜的線條時，輸出會限制於四個點，無法計算出更多點支持運算線條。



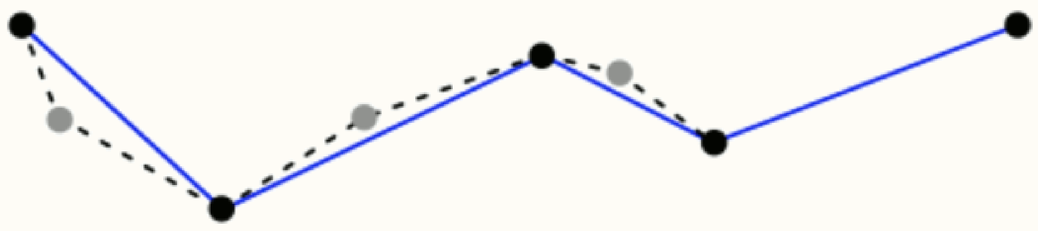
▲VGG-16（ours）模型架構圖(作者自行繪製)

(二) 自創演算法 Segment Vector and Curvature Feature Processing (SVCFP)

為了**解決 CNN 點位不足等諸多限制**，本研究發明了一套嶄新的演算法並透過幾何特徵分析，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，以下依序為此演算法的架構流程。

（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，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，並保留圖像轉折點和極值點。

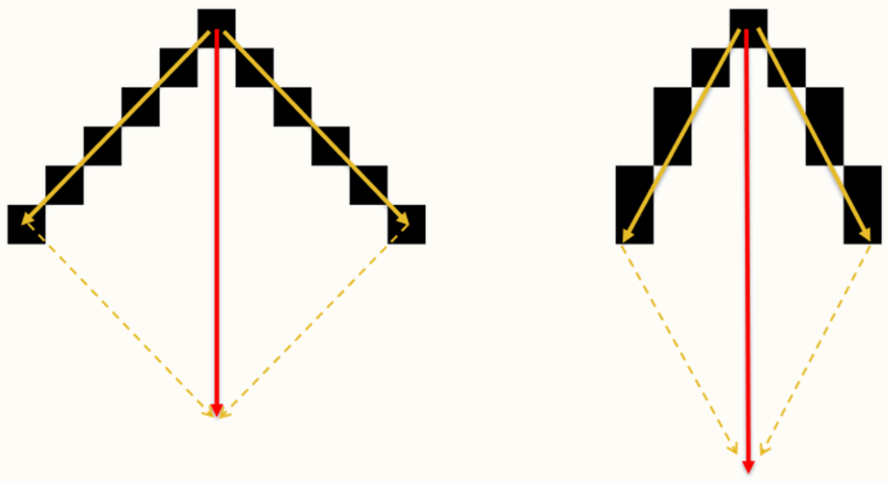


▲ RDP演算法示意 (圖片來源 : https://en.wikipedia.org/wiki/Ramer%E2%80%93Douglas%E2%80%93Peucker_algorithm)

向量判斷

此法透過分析簡化點在不同鄰域尺度下的向量大小，結合統計與閾值判斷，更準確地識別出顯著簡化點。

▲ 特殊曲線向量和縮短示意圖 (作者自行繪製)



▲向量判斷原理 (作者自行繪製)

角度變化

- 透過分析每個簡化點前後向量的內積來偵測角度變化，並利用外積判斷方向是否轉變，當外積符號改變時，該點即被認定為重要的轉向點。
- 分數超過預設閾值的點將被標記為特徵點，為了彌補RDP簡化可能遺漏的關鍵曲線形態，引入了中點插入與相近點融合機制。
- 避免節點過於密集，會檢查新點與周圍點的距離，若過於接近則執行消融策略，自動剔除冗餘點，確保最終特徵點的分布均衡與準確性。

四、貝茲擬合

選擇最佳的損失函數-Hausdorff距離

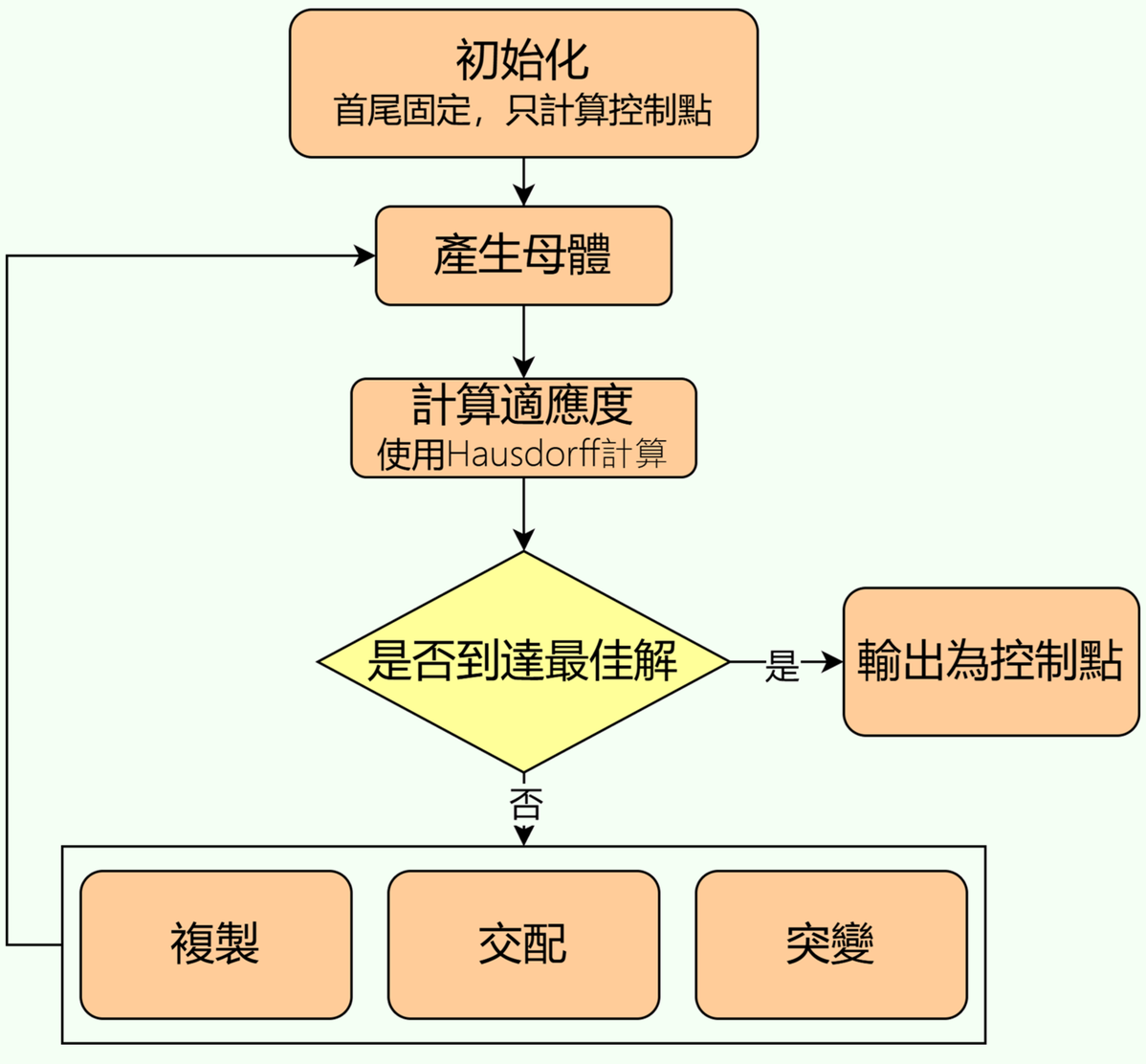
Hausdorff 距離是一種用來衡量兩個點集之間相似度的指標，**應用於手繪線段與擬合線段的座標比對**，透過計算兩者間的最大距離並標準化。

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

▲ Hausdorff 距離公式

擬合方法

(一) 遺傳演算法 (Genetic Algorithms, GA)



▲應用GA演化流程於貝茲擬合

初期採用遺傳演算法（GA）**優化貝茲曲線控制點，固定起終點、僅調整中間點**，透過選擇、交配與突變等演化策略，搭配 Hausdorff 距離評估誤差，不僅能跳脫區域最佳解，亦可藉由調整種群大小與世代數在擬合準確度與計算效率間取得良好平衡。

(二) 最小平方方法 (Least Squares Method, LSM)

步驟1 設定條件與擬合目標

已知始點P_0、終點P_3，假設待優化中間控制點P_1、P_2擬合點集 $\{Q_i\}_{i=1}^{n}$

步驟2 三次貝茲曲線的參數化數學模型

$$B(t) = (1-t)^3P_0 + 3(1-t)^2tP_1 + 3(1-t)t^2P_2 + t^3P_3\ , t \in [0,1]$$

步驟3 控制點的線性化表示與向量分解

重新表示曲線點 $B(t_i)$ 為中間控制點的線性組合如下

$$B(t_i) = A_1(t_i)P_1 + A_2(t_i)P_2 + R(t_i)$$

其中 $A_1(t_i) = 3(1-t_i)^2t_i$、$A_2(t_i) = 3(1-t_i)t_i^2$、

$$R(t_i) = (1-t_i)^3P_0 + t_i^3P_3$$

步驟4 建立最小誤差平方和目標函數

以擬合點誤差最小為目標，建構損失函數如下

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

步驟5 矩陣化建模與應用最小平方法解出最適控制點

解出 $\begin{bmatrix}P_1\\P_2\end{bmatrix} = (A^TA)^{-1}A^Tb$ ，$A = \begin{bmatrix}A_1(t_1) & A_2(t_1)\\ \vdots & \vdots \\ A_1(t_n) & A_2(t_n)\end{bmatrix}$，$b = Q - R\cdot A$

步驟6 完成控制點組合$\{P_1, P_2\}$與曲線擬合

研究結果與討論

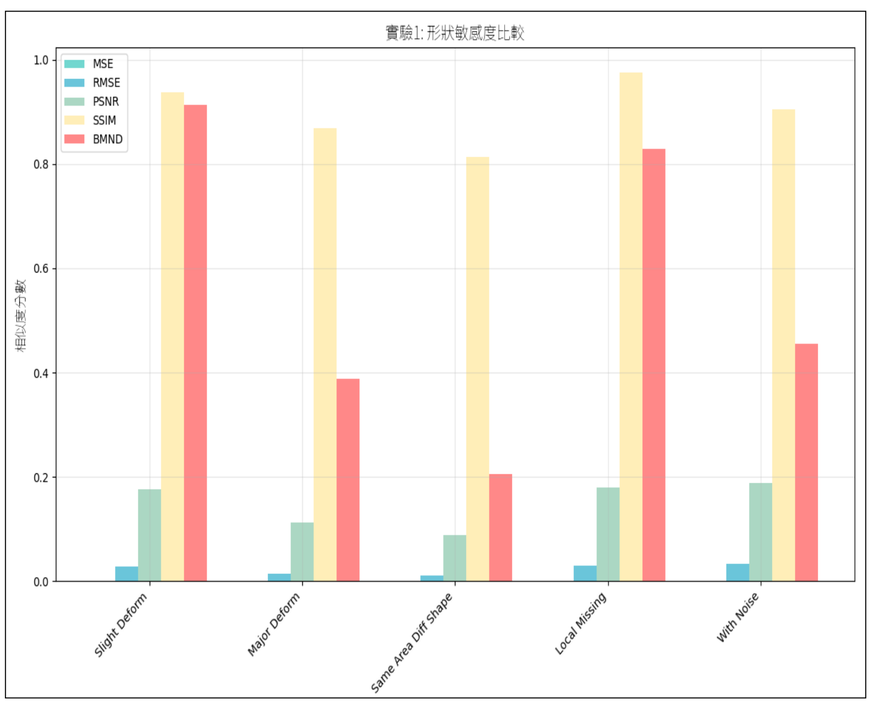
(一)自創圖像相似指標 (Bidirectional Mean Nearest Distance Similarity , BMND)

定義

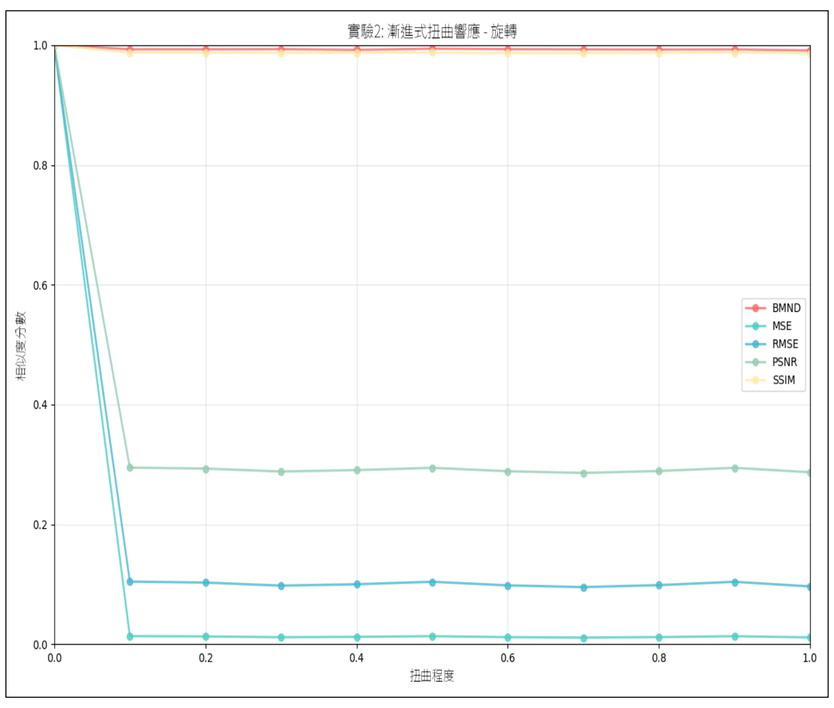
- 給定擬合曲線集 $A = \{a_1, a_2, \ldots, a_n\}$ 與原始輪廓集 $B = \{b_1, b_2, \ldots, b_m\}$
- 圖像集相似指標：$\text{Similarity}(A, B) = \dfrac{1}{1 + \frac{1}{2}\left(\frac{1}{m}\sum_{j=1}^{m} \min_i \|b_j - a_i\| + \frac{1}{n}\sum_{i=1}^{n} \min_j \|a_i - b_j\|\right)}$

優點 相較傳統像素的誤差法，**BMND的形狀敏感與對稱辨識更強，有效捕捉輪廓與結構變化**

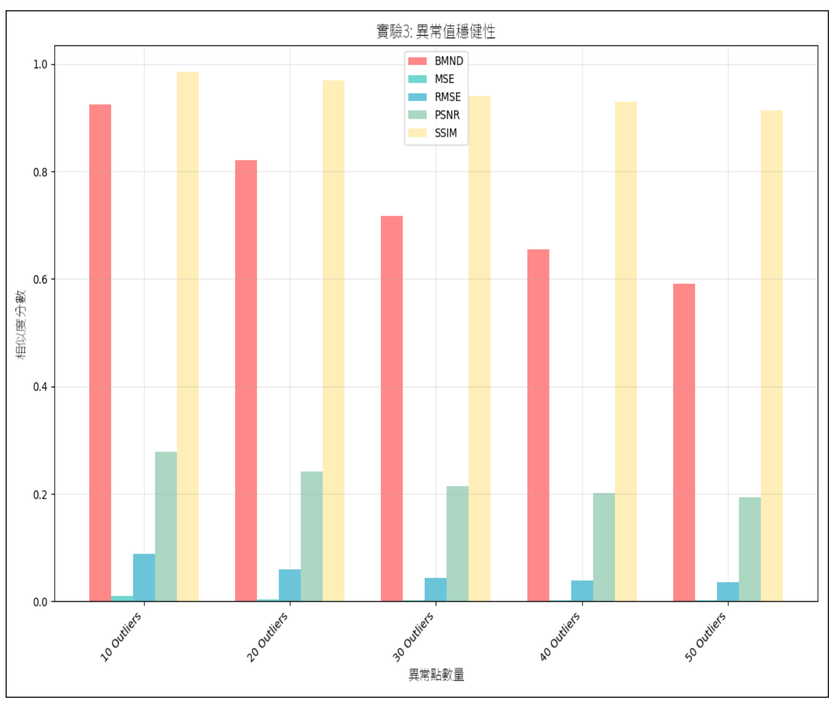
驗證 針對BMND的準確度，比較其他指標，設計**三種**實驗來驗證準確性 (完整數據解析請參閱工作書)



▲形狀敏感度比較 (作者自行繪製)



▲幾何變形穩定比較 (作者自行繪製)



▲雜訊干擾穩健比較 (作者自行繪製)

(二) 輪廓擬合結果 (**所需點數大幅減少且維持良好結構擬合**，更多筆擬合的結果請參閱工作書)

手繪曲線圖	SVCFP + GA	SVCFP + LSM	Inkscape
數字 : 8	所需點數 : 13個 BMND = 64.02 耗時 18.21 秒	所需點數 : 13個 BMND = 70.01 耗時 < 0.01秒	所需點數 : 97個 BMND = 97.40 耗時 < 0.01秒
國字 : 擬	所需點數 : 47個 BMND = 66.53 耗時 56.67 秒	所需點數 : 47個 BMND = 69.45 耗時 0.01 秒	所需點數 : 402個 BMND = 97.03 耗時 < 0.01秒
國字 : 高	所需點數 : 28個 BMND = 60.29 耗時 41.88 秒	所需點數 : 28個 BMND = 65.21 耗時 0.01 秒	所需點數 : 343個 BMND = 97.69 耗時 < 0.01 秒
國字 : 萬	所需點數 : 40個 BMND = 64.14 耗時 57.70秒	所需點數 : 40個 BMND = 66.50 耗時 0.01 秒	所需點數 : 431個 BMND = 97.48 耗時 < 0.01秒

▲擬合實際手繪曲線 (作者自行繪製與整理)

原圖像	SVCFP + GA	SVCFP + LSM	Inkscape
圖 : shark	所需點數 : 117個 BMND = 75.65 耗時 106.45 秒	所需點數 : 117個 BMND = 78.75 耗時 3.75 秒	所需點數 : 576個 BMND = 99.98 耗時 < 0.01 秒
圖 : bird	所需點數 : 211個 BMND = 78.47 耗時 134.52 秒	所需點數 : 211個 BMND = 79.69 耗時 2.51 秒	所需點數 : 1294個 BMND = 99.02 耗時 < 0.01 秒
圖 : eagle	所需點數 : 536個 BMND = 75.66 耗時 459.13 秒	所需點數 : 536個 BMND = 77.56 耗時 14.93 秒	所需點數 : 2202個 BMND = 99.66 耗時 < 0.01 秒
圖 : fish	所需點數 : 333個 BMND = 68.22 耗時 326.76 秒	所需點數 : 333個 BMND = 73.32 耗時 13.47 秒	所需點數 : 1114個 BMND = 98.62 耗時 < 0.01秒

▲擬合 kaggle ImageNet-Sketch 資料集 (圖片來源 : https://www.kaggle.com/wanghaohan/datasets)

結論與未來展望

(一) 曲線分段與特徵點自動擷取
本研究會根據輪廓局部變化程度自動切段，並由 SVCFP 精準擷取轉折與高變異區段。

(二) 貝茲擬合曲線演算法的效率提升
用 SVCFP + LSM 優化控制點，相較SVCFP + GA貝茲擬合最佳的結果，運算速度最高快24倍。

(三) 節點數量的大幅減少
相較Inkscape，向量化後控制點數平均減少約 84.6%降低向量圖的儲存最高達 90.8%壓縮比。

(四) 結構相似度維持穩定的水準
在控制點減少下，擬合結果的 SSIM 仍穩定維持於 0.94~0.97，表示曲線與原圖具高度視覺一致性。

(一) 未來可研究局部動態節點調整
依筆劃速率與區域曲率動態變更控制點密度，提升高變區段細節還原度。

(二) 引入深度學習輔助擬合
日後探討引入 GNN 或 Attention 機制預測節點配置，提升泛化能力與處理速度。。

(三) 採用混合評估機制建構
未來可結合 BMND (幾何) 、SSIM (結構) 、DTW (時序) 進行多維度圖形相似度分析。

(四) 延伸跨平台即時應用
未來延伸系統至行動裝置、嵌入式平台、WebAssembly，並支援電子筆、數位簽名板等輸入設備。

參考文獻資料

[1]Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. Numerische Mathematik, 1, 269–271.
[2]陳品均, & 鄭璧瑩. (2012). 針對動態路徑規劃之 D++ 演算法研究及其應用 (Doctoral dissertation).
[3] 明杰, & 張良正. 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.
[4]Holland, J. H. (1975). Adaptation in natural and artificial systems. University of Michigan Press.
[5]Hausdorff, F. (1914). Grundzüge der Mengenlehre. Veit.
[6]Baydas, S., & Karakas, B. (2019). Defining a curve as a Bezier curve. Journal of Taibah University for Science, 13(1), 522-528.
[7]Pastva, T. A. (1998). Bezier curve fitting (Doctoral dissertation, Monterey, California. Naval Postgraduate School).
[8]Mad, S. A. A. A. S., Zain, M. Y. M., & Miura, K. T. (2023). Curve fitting using generalized fractional Bézier curve.
[9]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 (Doctoral dissertation).
[10]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.
[11]Srinivas, M., & Patnaik, L. M. (1994). Genetic algorithms: A survey. computer, 27(6), 17-26.
[12]Krzeszewska, U., Poniszewska-Marańda, A., & Ochelska-Mierzejewska, J. (2022). Systematic comparison of vectorization methods in classification context. Applied Sciences, 12(10), 5119.
[13]Ramer, U. (1972). An iterative procedure for the polygonal approximation of plane curves. Computer graphics and image processing, 1(3), 244-256.
[14]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.