

第四章

以三角網格為基礎的圖像變形機制

在本章節中，我們將介紹經由圖像變形產生卡通化肖像的步驟細節。圖像變形的技術基礎是來源於影像座標軸的彎曲（Image Warping），只要尋找一個特定的彎曲函數就可以對原先影像的像素座標軸在向量空間中進行轉換，從簡單的放大縮小及旋轉乃至於漣波效果都是常見的應用結果。我們將應用這項基礎，從預先建立的手繪影像資料庫中用網格定義的卡通材質的位置，再尋找對應到真實影像中的彎曲函數，將真實影像中的五官特徵變形成卡通化的肖像。

4.1 三角網格產生方法（Triangular Mesh Creation）

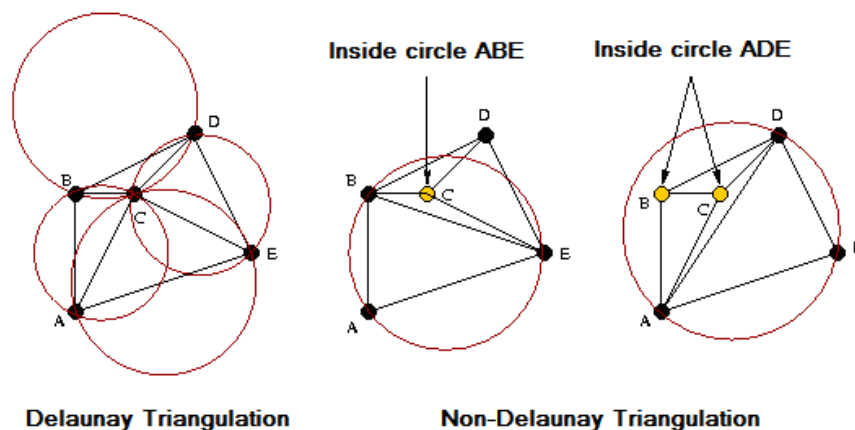


圖 4.1、Delaunay 三角分割

三角網格的自動產生是許多三維立體模型最常使用的基礎，在電腦圖學的相關研究中已列舉出相當穩定且妥為分割的演算法，最著名的莫過於在 1934 年由 B. Delaunay 所提出的 Delaunay triangulation (簡稱 DT) 演算法，Delaunay 三角分割的定義為在一個平面中的點集合的三角分割中，不存在有任一點被包含在 DT 分割後的所有三角形的外接圓內，見圖 4.1 所示。

之後又有 Guibas 及 Stolfi[22]提出運用 “Divide and Conquer”演算法實作應用將相關的電腦領域之中，之後又因應許多不同需求（例如：應用在三維空間與二維空間中的三角網格自動生成）而有許多改進的版本的演算法，在本論文中，我們採用的是 Shewchuk[24]所提出適用於二維空間下的三角分割方法，Shewchuk 提出了共有三種特色和適用性不同的演算法：

1. **拘泥式三角分割 (constrained DT)**: 依照原始點集合生成三角網格，但所生成的分割不一定滿足 DT 的定義條件，是最簡單迅速的分割方式。
2. **約束式的三角分割 (conforming DT, 簡稱 CDT)**: 完全符合 DT 的原始定義，會自動地在邊上插入適當中點 (Steiner points)，以生成三角網格，這些三角形會被加入的點限制較容易取得在角度及面積上一致性。
3. **拘泥約束三角分割 (constrained conforming DT, 簡稱 CCDT)**: 前兩者混合的 DT 自動網路生成，既滿足 DT 定義，且使用較少 Steiner points 的並且儘量讓三角分割取得在角度及面積上的一致性。

由於我們是使用三角分割來產生網格，並使用在圖像變形之中，因此我們必須變形的參考網格和目標網格擁有同樣的分割組成，同時在實作後，我們發現 CDT 生成效果並不如預期，而又為了能讓臉部材質保持較佳的連續性，所以對於三角網格的角度和面積也要求盡可能地保持一致性，最後我們選擇使用 CCDT 分割演算法且不

新增 Steiner points 的方式來生成最後的網格，圖 4.2 為我們分別使用兩種分割所產生的三角形網格。

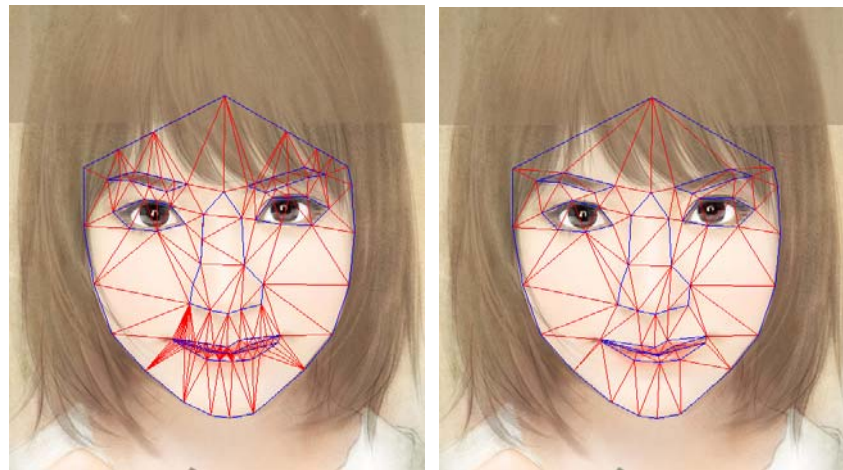


圖 4.2、左圖為 CDT 分割，右圖為 CCDT 分割結果

4.2 質心座標轉換 (Barycentric Coordinates)

經過上一節中產生的臉部三角形網格，在本節中我們將說明於圖像變形中如何運用質心座標系統 (Barycentric Coordinates) 的轉換機制。相對於絕對座標系統，所謂的質心座標系統，係指在包含三角形 ABC 的同一個平面中，平面上任一點均可以利用 ABC 三個頂點與其質心的向量作為基底，所定義的座標系統。

而由於質心的絕對座標，在幾何學中我們已知為三個頂點之絕對座標所決定 (如圖 4.3 所示)，因此，任一點在質心座標系統上亦可相當簡單地經由三個頂點的絕對座標轉換而成。

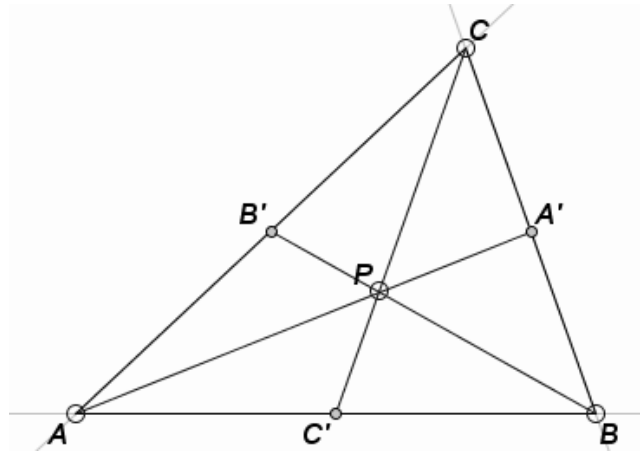


圖 4.3、三角形頂點與質心位置圖

如圖 4.3 所示，與 $\triangle ABC$ 同一平面下的 P 點，其質心座標 (α, β, γ) ，根據三角形質心定義的可以寫成以下通用式：

$$P = \frac{\alpha A + \beta B + \gamma C}{\alpha + \beta + \gamma} \quad (4.1)$$

其中， α, β, γ 可視為三個頂點的權重，當 $\alpha = \beta = \gamma$ 時則定義 P 為 $\triangle ABC$ 之質心。將 $\alpha + \beta + \gamma = 1$ 假設代入，則 (4.1) 可以通過化簡得知：

$$P = \alpha A + \beta B + \gamma C = \alpha A + (\beta + \gamma) \left(\frac{\beta B}{\beta + \gamma} + \frac{\gamma C}{\beta + \gamma} \right) \quad (4.2)$$

將 (4.2) 簡寫為 $\alpha A + (\beta + \gamma)K$ ，其中由向量定義及上圖相對位置可知 K 點落於 \overline{BC} 上，且 P 點位於 \overline{AK} 上，意即 K 為 \overline{AP} 於 \overline{BC} 上的切點 A' 。由 (4.2) 移項： $(\beta + \gamma)A' = \beta B + \gamma C \Rightarrow \frac{\overline{BA'}}{\overline{A'C}} = \frac{\gamma}{\beta}$ ，同理可得出：

$$\lambda = \frac{\overline{BA'}}{\overline{A'C}} = \frac{\gamma}{\beta} \quad , \quad \mu = \frac{\overline{CB'}}{\overline{B'A}} = \frac{\alpha}{\gamma} \quad , \quad \nu = \frac{\overline{AC'}}{\overline{C'B}} = \frac{\beta}{\alpha} \quad (4.3)$$

從塞瓦定理 (Ceva's Theorem) 及公式 (3) 亦可知，三線共點的 P 點的在質心座標中亦可寫為： $(1, \nu, \lambda \nu)$ 或 $(\lambda \mu, 1, \lambda)$ 或 $(\mu, \mu \nu, 1)$ ，且 $\lambda \mu \nu$ 積等於 1。

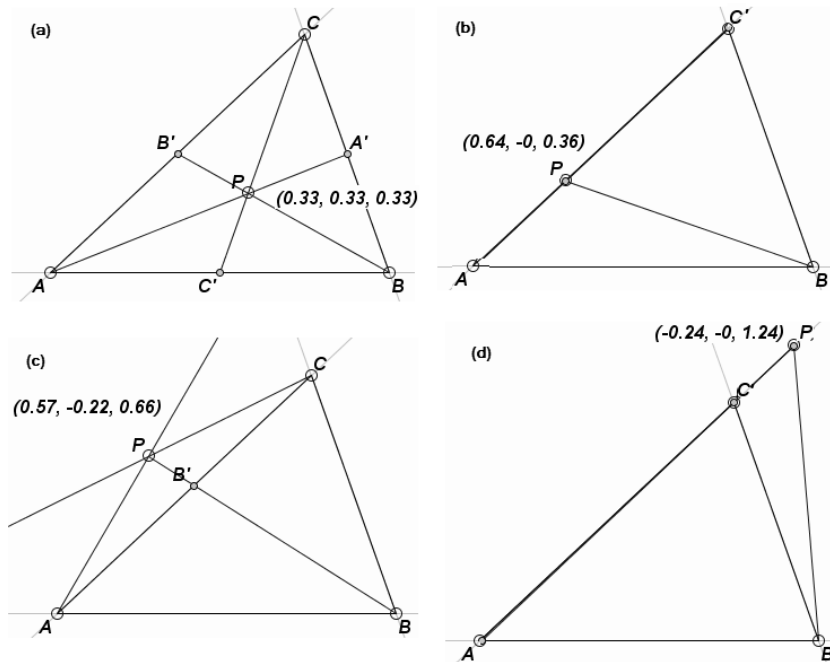


圖 4.4、質心座標 (α, β, γ) 與 $\triangle ABC$ 相對關係

此外，質心座標所擁有的幾何特性有以下幾點：

1. $\triangle ABC$ 頂點的質心座標為 $A(1, 0, 0)$ 、 $B(0, 1, 0)$ 、 $C(0, 0, 1)$ ，三角形內部任一點 P 質心座標 (α, β, γ) ，再經過正規化後， $0 < \alpha, \beta, \gamma < 1$ 。(見圖 4.4a)
2. P 點質心座標， α, β, γ 任一為 0 且其他兩者大於 0 時，則 P 落在其對邊之上。(見圖 4.4b)
3. α, β, γ 任一小於 0 且其他兩者大於 0， P 則落在對邊之外。(見圖 4.4c)
4. α, β, γ 任一為 0，其他兩者有一小於 0 時則落在與其端點反向的延申線之上。(見圖 4.4d)
5. α, β, γ 三者不同為負數。

在圖像變形中，由於我們的輸入輸出影像，都是由同一組特徵點定義所分割出的三角網格，因此在個別的三角網格中的每一個像素材質，就可以運用質心座標系統的對應函數得出目標網格之中的對應點（如圖 4.5 所示）：

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = F^{-1}(\alpha, \beta, \lambda) = [\alpha, \beta, \lambda] \times \begin{bmatrix} A'(x_1, y_1) \\ B'(x_2, y_2) \\ C'(x_3, y_3) \end{bmatrix} \quad (4.4)$$

因此理論上在計算過程裡，相對於一般運用 Single Line 或 Multiple Lines 的彎曲對應座標，不同三角形中的質心座標系統轉換幾乎不需耗費太多時間，同時不需要額外提供權重參數。

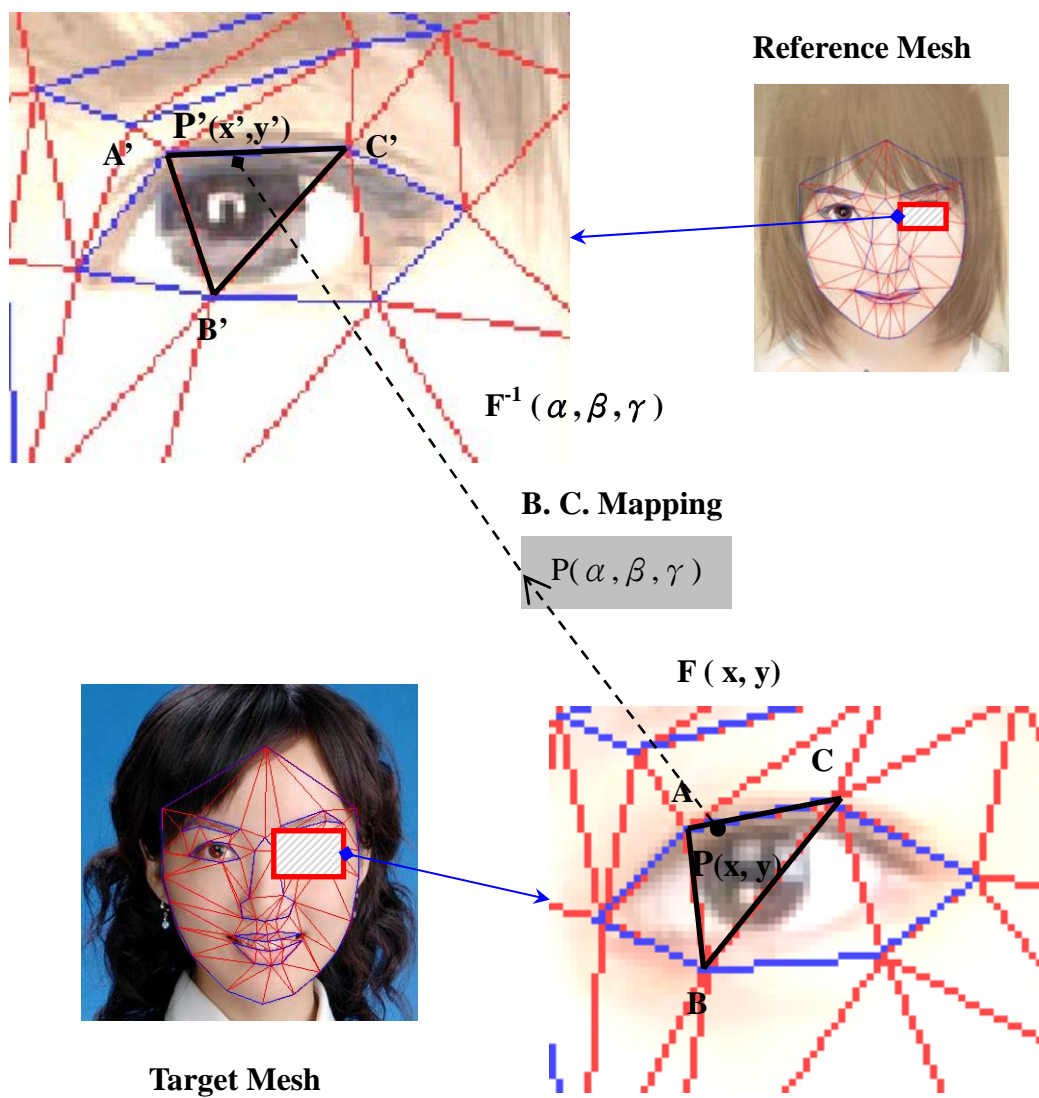


圖 4.5、利用質心座標對應進行網格變形

在前一章的系統架構中，我們針對每一張輸入的真實影像與資料庫中的卡通肖像材質，皆定義了同樣數目與順序的臉部特徵點，在 66 個臉部特徵點中，共產生 114 個符合 CCDT 的三角形，在接下來的變形機制中，就是運用這些位於不同影像中的三角形區域來尋找質心座標的轉換及對應。



圖 4.6、運用三角網格質心座標系統變形產生臉部卡通肖像

在影像處理的細節中，像素座標均是以直角矩陣的方式來反覆處理，因此，即使取得將是包含所欲處理的三角形的矩形區域，在實作中為了避免約有 50% 的像素是不必要

的重覆運算，我們妥善運用質心座標的三角形內部判斷特性，在每一次的 x 軸像素橫移時，先計算出位於三角形內部 y 軸像素所在的區段，如此再為真實影像中的每一個像素點，貼上資料庫影像中對應點上的卡通材質像素。

相較於其他多邊形網格變形，質心座標的運用帶來許多的便利和迅捷的特性，首先就理論系統上已為網格變形的機制保證了：位於三角網格內的每一點皆可以找到位於另一個影像中的對應點，因此原始影像及參考影像在影像維度的大小上不需要對稱，及不需額外作加權上的運算，此外藉由與絕對座標間的正逆轉換函數也提供的運算速度上的保證，在同一時間，直接判斷該點是否位於三角網格之內的特質，也為相關的圖學運算中亦帶來了實作上的便利性，更節省下許多額外的運算時間。我們實作的以變形為基礎的卡通肖像轉換系統介面見圖 4.6 所示。

4.3 以形狀比對為基礎的排序與混合藝術風格生成 (Shape Matching Driven Polymorph and Dataset ordering)

我們在建立手繪風格的卡通肖像影像資料庫時，是選用了許多不同風格且不同人物的卡通影像進行手動標記，因此同樣的輸入可以依使用者喜好選擇不同風格的參考樣版來產生不同的卡通化肖像結果，由真實影像運用不同的手繪卡通樣版轉換而成。系統亦可依據輸入臉型，自動挑選合適的卡通樣版，本論文中，對於某些臉部特徵色明顯的手繪影像或輸入影像，我們使用了以五官特徵點所構成的形狀比對 (Shape Matching) 方法[13]於互動介面中來建議使用者挑選較適合的資料庫樣版，詳細形狀比對演算法步驟說明於下：

Step1. 對齊形狀：

形狀比對的第一步，我們先載入已在前處理中定義的臉部特徵點定義資料 F ，這些資料是以特徵點在影像中的絕對位置，在比對之前除了個別截取對應臉部元件（Facial Components）比對之外，我們仍需對個別元件進行簡略的對齊及相對位置的計算，先以原件形狀的重心（Gravity Center）對齊的基準，再將以影像為基準的絕對座標轉為以重心為原點的相對座標。

Step2. 一維複數向量傅利葉轉換：

我們將這些座標組標記為： $x_i(n)$ 及 $y_i(n)$ ，分別表示為第 i 的臉部構成元件的第 n 的特徵點，臉部元件的總數為 M 個，每一個元件內含 N_i 個特徵點座標組。形狀比對的第二步我們參考臉部輪廓誇張參數化[13]標準中，將特徵點的座標， $u_i(n)=x_i(n)+jy_i(n)$ 轉換至複數空間（ $j=\sqrt{-1}$ ），再套用一維空間的傅利葉轉換（Fourier Transform），產生我們定義的一個臉部形狀描述子（Facial Shape Descriptors, 簡稱 **FSD**）：

$$a_i(k) = F\{u_i(n)\} = \sum_{n=0}^{N_i-1} u_i(n) \cdot e^{-j2\pi kn / N_i} \quad (4.5)$$

$$\tilde{u}_i(k) = F^{-1}\{a_i(k)\} = \sum_{k=0}^{L_i-1} a_i(k) \cdot e^{j2\pi kn / N_i} \quad (4.6)$$

我們採用傅利葉轉換的優勢，是在於其可內嵌有空間域中的所有關係和運算資訊，其中 L_i 是轉換後所選取的頻率數目，同時，頻域的內容也可在逆轉換後也可以找到連續性的對應關係，因此在頻域中對產生的描述子進行運算，理論上都可以對應回原先的臉部組件的形狀在座標軸之於旋轉（rotation）、位移（translation）、剪切（shearing）等變化。

從式（4.5）及（4.6），當以 F_p 為輸入的臉部特徵點定義，與 F_{db_s} 為手繪卡通資料

庫中的第 s 個樣版的特徵點定義資訊時，我們可以將用以下來逆傅利葉轉換表示當圖像變形的過程：

$$F_p = \bigcup_i F^{-1} \left\{ \frac{1}{N_p} \sum_{k=1}^{N_p} FSD_i^{F_{pk}} \right\} \quad (4.7)$$

Step3. 計算頻域特徵向量的強度誤差

將特徵點形成的形狀資訊轉換至頻域後，我們可以定義相異描述子之間的距離 $Dist$ 為：

$$Dist(FSD_i^{F_p}, FSD_i^{F_{db_s}}) = \left[\frac{1}{L_i} \sum_{k=0}^{L_i} \left(\left| a_i^{F_p}(k) \right| - \left| a_i^{F_{db_s}}(k) \right| \right)^2 \right]^{0.5} \quad (4.8)$$

其中，我們根據每個個別臉部組件中的形狀描述子在頻域之間的距離定義，再利用個別組件的距離總合作為兩者的在形狀比對的分數：

$$S(P, Q_s) = \sum_{i=1}^M Dist(FSD_i^P, FSD_i^{Q_s}) \quad (4.9)$$

此分數即是我們自資料庫中挑選的依據指標，通過此項指標的簡易排序結果將如表 4.1 所示，使用者得以在互動介面中獲知就當前輸入的照片，在參考樣版影像上的建議，選擇較為合適的來產生卡通肖像。

在實作形狀描述子距離計算時，由於我們所使用的各個臉部元件特徵向量長度約在 5~16 個特徵點之間，轉換後的特徵空間小的緣故，因此如再分別計算個別形狀的距離分數所帶出的意義不大，而且還有失去個別五官在空間中的相對位置資訊之虞，因此採取的修正作法是將全部的 66 個臉部定位特徵，視為單一個複合形狀來進行比對，表 4.1 為分別針對不同性別的卡通素材資料庫，依輸入影像不同所進行排序的結果，對應到的參考的距離分數的分布則如圖 4.7 所示。

表 4.1、運用 FSD 間的距離分數來排序參考樣版影像


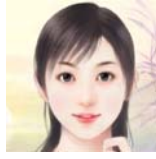



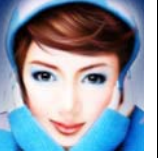





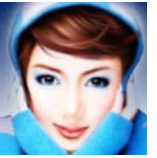












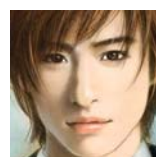



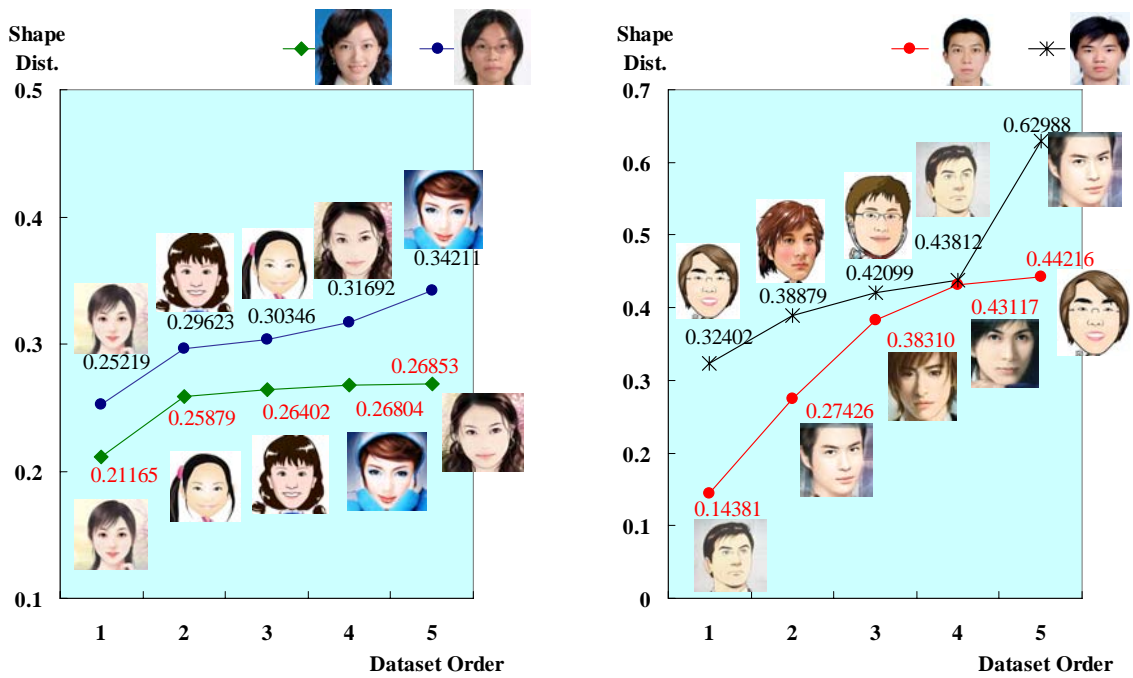
First 5 th Samples with Minimum Shape Dist. Score						
						
						
						
						

圖 4.7、FAD 距離分數分布



在前面的定義中，我們除了運用傅利葉轉換來定義臉部特徵形狀的描述子（Shape Descriptor），通過這些形狀的描述子的距離定義，除了改善資料庫中的影像排序機制，由圖 4.7 的 FSD 分布來看，FSD 之間的距離代表了參考影像與原始影像在五官特徵上的影像變形代價，距離越小的相對代價越低，考慮到應用這些形狀描述子在改善及強化圖像變形的效果上，我們參考了在多重變形（Polymorph）[32]的研究，此種機制是提出一種利用多個影像的參數化，來使得最後的影像變形能具有較多的細部特徵，因此在本論文中嘗試以排序後的次序，將所有列入參考範圍的卡通素材於 FSD 的距離分數，以倒數後所占比重為基礎，對此種變形機制來進行實驗，希望能讓產生的卡通肖像跳脫出原來的素材，同時具有多個藝術繪畫風格，在變形過程中，我們將多個列入參考的影像當作多項式中的代表不同基底的變數，參考單一變形的代價做為新的變數權重來計算混合合成的結果（見圖 4.8）。

運用此法的優點所合成的結果影像，除了將擁有比較多的特徵空間參考外，在自由度上更具下達至個別五官的元件特徵，運用多項式變形方法，可以使合成的卡通化肖像保留更多的相異性質和特徵，從而改善了一對一影像變形中參考特徵單調的特性。

在圖 4.8 中， I_0 代表是輸入的真實影像， $I_1 \sim I_3$ 表示從蒐集的藝術畫風資料庫中，依據形狀描述子的距離函數 D 大小，所排序出的前三名臉部特徵形狀，最為接近的卡通素材。 I_{ij} 代表輸入影像 I_i 參考資料庫的畫風 I_j 運用 W_{ij} 變形函數產生的卡通肖像化結果，在一般的影像變形演算法中，結果影像除了變形函數之外還會經過色彩混合（Color Blending）函數 B_j （公式 4.10）：

$$I_{ij} = B_j W_{ij} \quad (4.10)$$

然而我們在影像變形的實作中是利用尋找像素點在三角質心座標的對應點（公式 4.4），因此產生單一畫風的肖像結果時，因此在此處變形並不需要經由混色的處理，而

權重 \bar{W}_j 則是由距離函數 $D(I_i, I_j)$ ，在所有被參考的所有影像中所占的倒數比來決定（見公式 4.11）。

$$\bar{W}_j = \frac{j}{\sum_j D(I_i, I_j)}, (D(I_i, I_j) > 0) \quad (4.11)$$

最後的多項式影像變形中（式 4.12），這些參考影像權重 \bar{W}_j ，將決定的色彩混合函數 B_j 中，代表單一畫風肖像 I_{ij} 內每一個像素被選用的比例，混色後將生成最後具有多種藝術風格的卡通肖像 I'_0 。

$$I'_i = \sum_j (\bar{W}_j \cdot B_j W_{ij}) \quad (4.12)$$

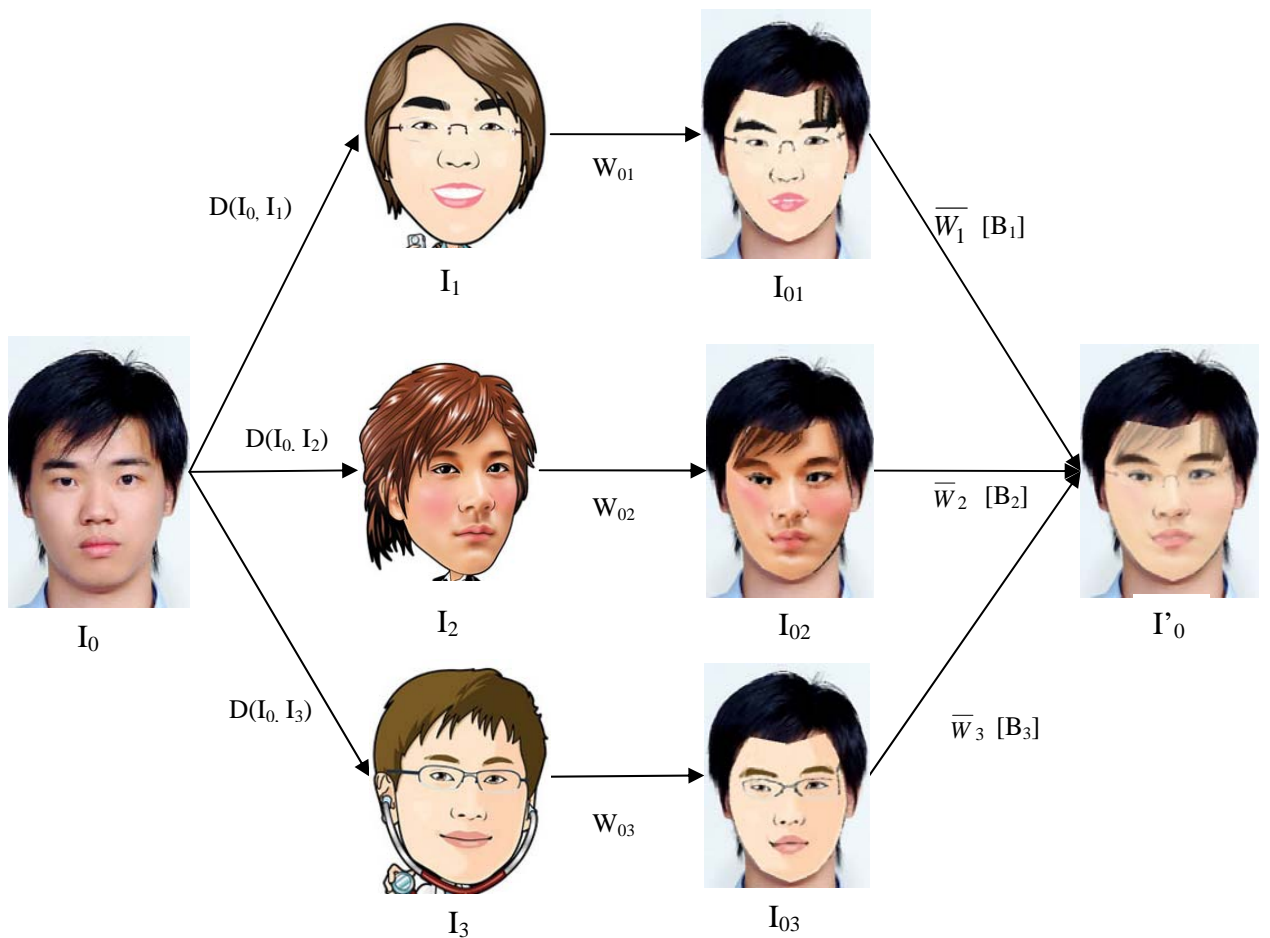


圖 4.8、Polymorph 影像變形方法[32]系統架構及轉換的結果