

研究報告

TR-83-001

畫名稱：電腦動畫之數學模式

畫主持人：鄭國揚

畫編號：NSC70-0404-E001-03

行期限：中華民國70年4月1日至71年6月30日

期：中華民國七十二年一月十五日

中研院資訊所圖書室



3 0330 03 00034 8

書 考 參
—
借 外 不

0034

緒論	1
1. 電腦動態畫之系統模型	3
2 B 軟楔函數為主之數學模式	5
2.1 靜畫之理論基礎	5
2.2 輸入方法	9
3 自然的靜畫輸入法	12
3.1 問題之敘述	12
3.2 求解的因素考慮	14
3.3 主要結果	17
3.3.1 Δ 處理法	18
3.3.2 頂點投影法	18
3.3.3 特殊線段分類處理	19
3.3.4 第三類曲線段之極少數據點法	20
3.3.5 第二類曲線段之極少數據點法	21
3.3.6 求 B 軟楔曲線及反求控制點法	21
3.3.7 第四類曲線段之極少數據點法	25
3.3.8 圖形數據壓縮法	26
3.4 實例示範	27
4. 動畫處理方法	36
4.1 電腦動畫的定義	36
4.2 B 軟楔曲線間之朦朧相似性	37
4.3 動態畫產生法	40
5. 語言處理	47
5.1 POCA軟體系統結構	47
5.2 圖形資料管理	54
5.3 POCAL 語言的描述	58
6. 以電腦輔助卡通製作的實驗系統 (POCA 系統) 爲實例示範 ..	70
6.1 POCA系統的硬體結構	71
6.2 開機使用程序	73
6.3 製作實況及結果	74
6.4 着色方法	77
6.4.1 着色使用法	77
6.4.2 着色演算法	77

電腦動畫 (Computer Animation) 是圖形模擬 (Graphic Simulation) 的一項科目。其處理的對象是電腦圖形。圖形經過電腦動畫系統的處理後，會自動產生有動態感覺的一連串圖形，以模擬某些人爲藝術化的動作，或者自然界某些物理規則的運動。由於人們的世界視覺，是充滿著動態的彩色圖形，而電腦的發展是爲處理人類的資訊，所以電腦動畫的研究在近代電腦圖形學的領域裏，也就愈來愈顯得重要了。

電腦動畫的問題十幾年來一直被探討 [1,2,3]，但是一直沒有很好的結果，我想最主要的原因在於以數學處理動畫的方法尚未被深入研究之故。目前，我們所看到及所瞭解的動畫系統，仍停留在利用數學轉換的技巧產生圖形，例如用轉換因素 (Transformation factors): 放縮、位移及旋轉，或者用點群決定的內插近似函數，作用到圖形以產生動態變化的圖形序列。用這種方法看起來似乎是最直接也最自然的方式，因爲這種方式和人爲輸入圖形的方式很類似，可以給繪圖的人很大的便利及易于掌握圖形。但是，從系統設計的觀點言，這種便捷的輸入方式及簡易的數學處理法，却大大地限制了電腦動畫的功能及彈性。因爲電腦動畫的處理涉及了人爲藝術的觀點及物理規則的模式，一種較具規模的電腦動畫系統通常需要具有人工智慧及模式辨別的能力。在這種要求下，一套能配合人工智慧論的數學模式就愈顯其重要性了。

近代電腦圖形的技術進展非常快速，不少人提出了創新的圖形處理法，特別是圖形的近似方法有了更廣闊的觀念。其中，我個人認爲 Bézier 函數及 B 軟楔函數對建立電腦動畫的數學模式很有幫助。這些函數固然是從傳統的近似理論推演得到，然其觀點却非常新穎，容易達成電腦圖形的許多要求。簡單地說，這些函數在決定一條曲線函數 f 時，是給一個多邊形（稱爲控制多邊形 C ），使得近似 C 的曲線形狀 $P(C)$ 非常接近 f 的形狀。傳統的內插近似方法，是在 f 上取內插點群 S ，使得近似曲線 $P(S)$ 的形狀非常接近 f 的形狀。這種觀念的改變，產生了新的理論及新的演算法，它們的最主要貢獻在於提供了電腦圖形設計上很多的優點，譬如，B 軟楔曲線容易修改局部的形狀，用很少的控制點可產生複雜的曲線形狀等。它們帶給了近代電腦圖形設計上的方法論很大的衝擊，然而應用這些函數（B 軟楔曲線）來探討電腦動畫的數

學模式，在國外的很多雜誌裏却未曾看到。我個人曾在去年利用這些函數探討它們應用在電腦動畫演算法裏的可行性，目前已獲致滿意的結果，然而，也發掘出了不少連帶性的困難問題極待解決。

其中，最困難的問題是採用了B軟楔函數表現圖形，會使得圖形輸入的方法成為不自然的方式。因為，此時圖形的輸入是用控制多邊形，它經過B軟楔公式運算後才產生真正的圖形，使用人必須經過訓練，對B軟楔函數之特性和圖形之產生者間的關係和規則瞭解後才會駕輕就熟。即使是使用者已熟練B軟楔曲線輸入的方法，其輸入的方式則仍是取直線段的端點，而不是連續的曲線段，這對於原畫家而言，可能是一種無法忍受的方式，對於一般描圖者而言，亦可能是一件費時的工作。解決這項困難問題的唯一途徑是致力探討“B軟楔函數反求控制多邊形的方法何在？”。換言之，如果取B軟楔曲線上的許多樣本點數據，經過“反求過程”(Inversion Process)，可以決定邊數足夠少的控制多邊形。那麼，輸入圖形的方法就可直接描繪，讓反求過程去決定控制多邊形，供系統自動去處理以產生動畫。然而，從目前的數學理論去証明“反求過程的極佳化”(決定極少邊數的控制多邊形)是極困難的，甚至於是不可能的。基于這項理由，反求過程亦僅能是“經驗的演算法”(Heuristic Algorithm)。

一旦圖形的數學表示法確定，以及相關的宜人因素(自然的輸入法)皆被考慮後，則在規畫整個電腦動畫的系統時，將首重模型之建立。本文採用以B軟楔函數為圖形的數學式，則會如上所述，會造成不自然的圖形輸入法，其改變辦法為重新探討B軟楔函數近似理論，以自然的方式輸入圖形，再經過數學轉換成B軟楔函數表示式。經轉換後的B軟楔函數為動畫處理的對象，而且能接受命令語言的語言處理，以產生各種動態圖形編輯的效果，才能達到交互作用系統的要求，有關這方面的詳情，本文擬以電腦協助卡通製作之系統為例說明。故本文裏將提出(1)電腦動畫設計之模型，(2)以B軟楔函數為主之數學模式，(3)自然的靜畫輸入法，(4)動畫處理方法，(5)語言處理，(6)以電腦輔助卡通製作的實驗系統為實例示範，在以下各章節中敘述。

1 電腦動畫之系統模型

動畫設計本身是一種創作的過程，此過程需和許多的情況相配合，才能創造出美好的結果。同樣地，欲實現動畫電腦化，則需要一種互動式的設計系統，它能具有層次分明結構，而且其流程是遞迴的組織。因此，電腦動畫設計系統的功能等級就和其系統模型的建立方式有關，本文所提出的動畫系統模型是屬於具有人工智慧型之互動式電腦輔助設計系統。

爲了瞭解一個電腦輔助動畫設計系統可能的功能，其設計的元素及過程必須層次分明。對於任何軟體系統的情形而言，創造的過程是根據概念及需求而定，同時亦受到系統的某些限制而修飾了其創造的方式，圖1-1所示爲動畫設計過程的概念。任何動畫的設計有三個基本的要件：審美評鑑、動態模擬及應付處理。此三要件是電腦運用時在計算的或互作的程序中表現，以達成一個完整的設計。作用於它們的處理方式有四個修飾的程序，分別是簡化、模擬、最佳化，和模式化。最後輸出的圖形可在螢幕上顯像，必要時可回饋，以影響概念及需求，做創造過程上修飾的工作。

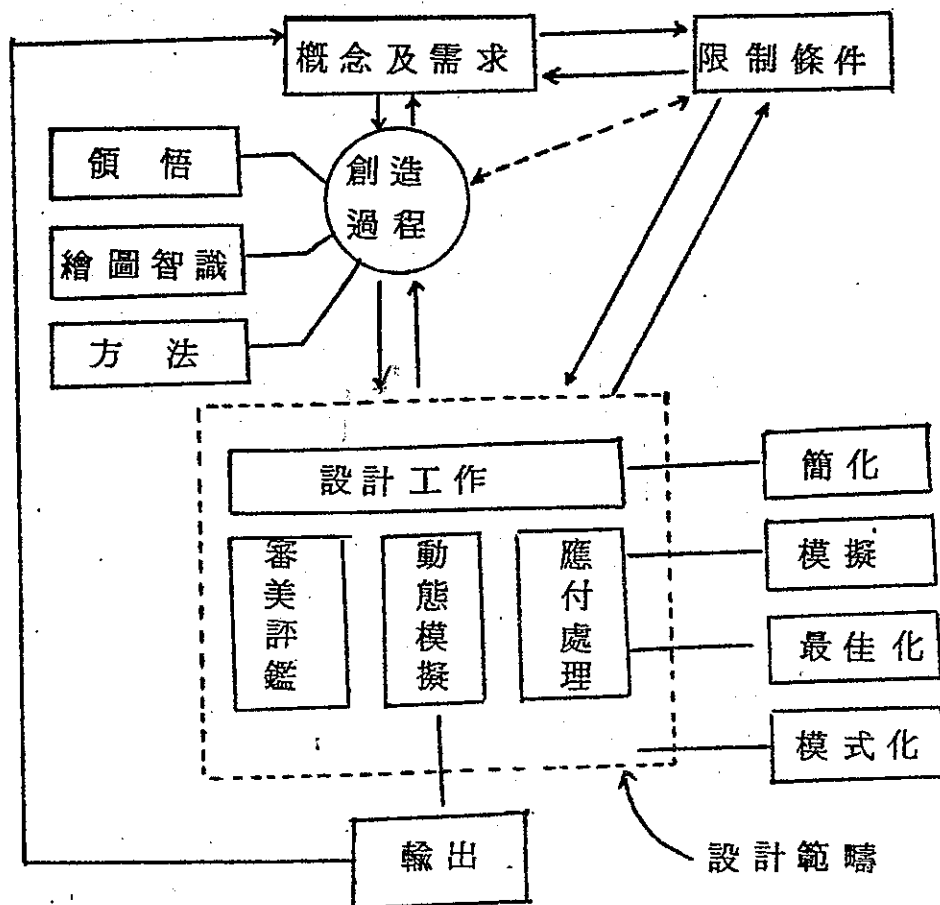


圖 1-1 動畫設計過程

在設計範疇中之動畫設計工作，審美評鑑包括了靜態畫（原畫）的繪製方法，是否容易表現圖形美和產生滿意的動畫，是否容易使設計者發揮其藝術創作等。動態模擬包括了動畫演算法，是否生動或滿足自然界的物理規則。應付處理包括了系統的效能，是否具有完整性，一致性，和易用性等特質。當然，這三項設計工作是互相牽連的，某種模型的選擇會影響到整個設計的觀念及方法。本研究的重點在提出一種具有人工智慧的動畫設計系統模型，以及在電腦上實現此種模型的各種數學模式。

圖1-2 示出一種電腦輔助動畫設計的模型，其中，概念範疇等於圖1-1的概念和需求，設計人代表了創造過程，而電腦範疇中的“程式”代表了設計工作的三要件在電腦裏執行的過程，命令語言溝通了設計者和電腦範疇的工作，然而這種溝通亦可加入人工智慧的觀念及學理，譬如修改靜畫和動畫的美感性和逼真性時，可採用朦朧集合論（Fuzzy Set Theory）。

從原畫產生動畫的概念及需求，會受到選用那種數學模式的表現之影響，不同的數學模式會產生不同的規則。本文採用了B軟楔近似函數論的方法來表現圖形，並配合人工智慧論的觀點，來建造供使用者繪製原畫及產生動畫的規則。因此，這種模型是一種人機互動式的系統，它必須涉及人機間的對話，這種形式常會遇到用自然語言（或朦朧語言）可否被機器接受的問題，本文亦將對這方面提出解決的方法。

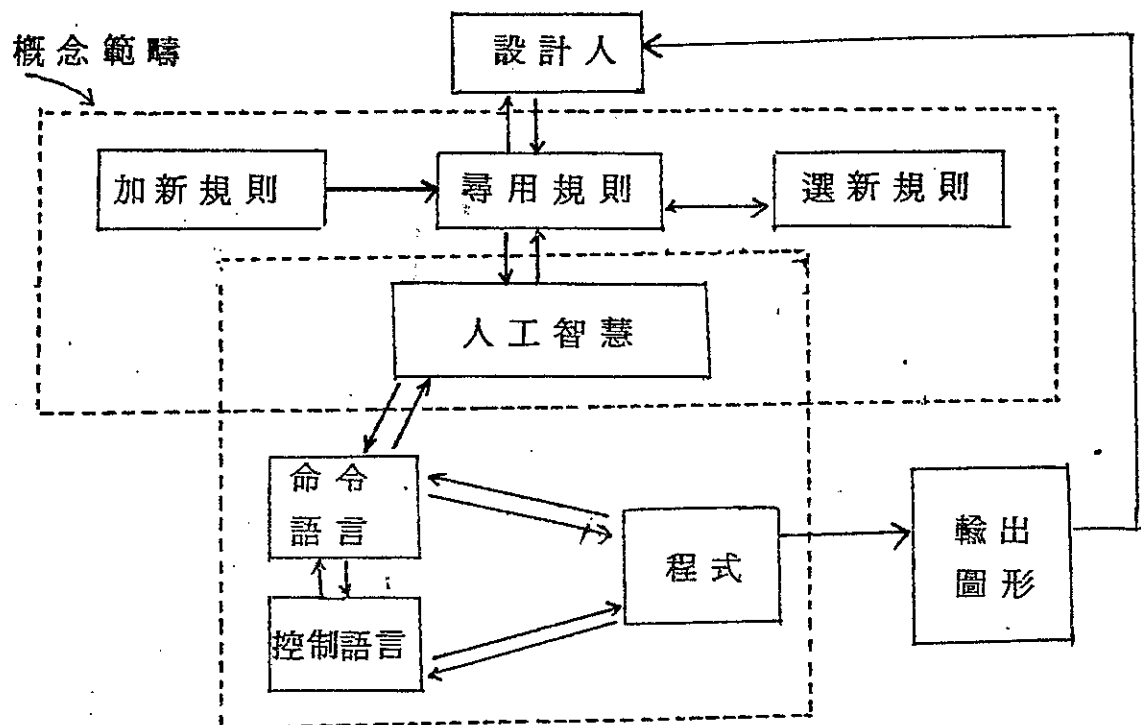


圖1-2 人工智慧的電腦動畫模型

2.1 靜畫之理論基礎

在電腦動畫的應用上，古典的內插多項式逼近，例如 Lagrange, Hermite, ... 等皆是考慮近似曲線通過已知數據點的情形，這些方法有二項主要的缺點：(1)任何一個數據點的更動，會使原先的曲線形狀產生大幅度的變化，缺乏可調整性，不適宜用在人機交談式的作業上，(2)曲線上的數據點應選那些點，使用人無法預知，即使他心中有一曲線形狀的圖樣，他也不知怎樣輸入電腦，尤其在很多實際應用裏，曲線上的點座標不是已知的。典型的例子是寫生畫的畫家在勾劃一條曲線時，他心中所想的是它的形狀，而不是它所通過的數據點，因此，若只要電腦產生圖樣的形狀時，古典的曲線逼近法就不適用。

最早提出利用控制點產生曲線形狀比較成功的方法是 Bézier 多項式 [4]，任一 Bézier 曲線的形狀是由一個素描多邊形所控制，圖形可經由調整其素描控制點的位置而獲得。Bézier 多項式的理論基礎是採用 Beinstein 多項式近似法，它只是將 Beinstein 多項式 [5] 展開式的各項係數改用素描控制點的位置，因為 Beinstein 近似式的基底函數 (Basis Function) 是屬於凸集組合 (Convex Combination)，所以 Bézier 曲線被限囿於由素描控制點所形成的凸集殼 (Convex Hull) 裏，而且逼近素描多邊形的形狀。這種方法的最大優點是圖形可經由素描多邊形產生，而且可隨意更改圖形（經由增添、移動、或刪除素描控制點，改變素描多邊形中獲得）。使用人可以完全不必瞭解數學的理論，也可以操縱自如地將任意形狀的曲線輸入電腦。

Bézier 多項式逼近素描多邊形的缺點是，它為高次多項式近似，近似多項式會隨著素描控制點數增加而增高，這樣會使得修改圖形比較困難，因為移動（或增添、或刪除）一素描控制點會影響到整個曲線的形狀，它缺乏局部修正的能力。若欲改進這項缺點，唯一的方法就是摒棄 Beinstein 基底函數，而改用低次多項式，但必須保持整段曲線的平滑性。在這種條件下，最適宜的方法可能是採用軟楔函數 (Spline Functions) 近似。同樣的道理軟楔函數亦需改頭換面，其基底函數用參數來表示，並用素描控制點當做展開式中的係數，讓我們稱這樣所產生的曲線為 B 軟楔曲線 [5]，B 軟楔曲線是由低次多項式所連接而成（但保持整條曲線是某種程度的連續），所以當移動一素描點時，只會造成局部

的影響，而不會影響到更遠軟楔線段的形狀。

靜畫製作之原理係採用均勻三次 B 軟楔曲線之特性，有關 B 軟楔函數近似的理論，請讀者詳閱[6]，本文僅將需要的公式及其幾何特性提出來討論如下。令 $x=(x_0, x_1, \dots, x_k)$ 為一實數的向量，其中 $x_i \leq x_{i+1}$ ，若它滿足下面的二個條件

(1) 在每個區間 (x_i, x_{i+1}) 是 $m-1$ 次的多項式，

(2) $P \in C^{m-2}$ 為 $m-2$ 次微分，為連續函數。

則一個函數 P 稱為 $m-1$ 次軟楔函數，令 $S(m, P)$ 表示所有 $m-1$ 次軟楔函數所存在的 $m+k-1$ 次元空間，則在所有可能當做的基底中，我們有興趣的是 B 軟楔基底。令向量點 $\{v_i\}$ 為一素描多邊形 C 的端點（或稱為素描控制點）以及 $\phi_{i,m}(s)$ 為在區間 $(x_i, x_{(i+m \bmod n)})$ $i = 0, 1, \dots, n-1$ 的 $m-1$ 次 B 軟楔基底函數，則對應素描多邊形 C 的 $m-1$ 次 B 軟楔曲線是由下式決定

$$P_{m-1}(C) = [x(s), y(s)] = \sum_{i=0}^{n-1} v_i \phi_{i,m}(s) \quad (2.1)$$

其中基底函數 $\phi_{i,m}(s)$ 可由下面的遞迴函數求得

$$\phi_{i,1}(s) = \begin{cases} 1 & , \text{ if } x_i \leq s \leq x_{i+1} \\ 0 & , \text{ others} \end{cases}$$

$$\phi_{i,m}(s) = \frac{s - x_i}{x_{i+m-1} - x_i} \cdot \phi_{i,m-1}(s) + \frac{x_{i-m} - s}{x_{i+m} - x_{i+1}} \cdot \phi_{i+1,m-1}(s),$$

if $m > 1$:

(2.2)

吾人不難証明(2.2)式具有凸集組合的特性，亦即

$$\sum \phi_{i,m}(s) \equiv 1$$

而且它是 $m-1$ 次的低次多項式，換言之，這樣的選取滿足了改善 Bezier 多項式缺陷的要求。為了便於計算，我們選取一組特殊的區間：令 $(x_{i+k} - x_i)/(x_{i+1} - x_i) = k$ ，以及取參數 $u = (s - x_i)/(x_{i+1} - x_i)$

使得 $u \in [0, 1]$ 而且各區間的點數是均勻分佈的。若取 $m=4$ 則上式可化簡成下面的矩陣表示式

$$P(u) = [t^3 \ t^2 \ t \ 1] \frac{1}{6} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix} \begin{bmatrix} v_i \\ v_{i+1} \\ v_{i+2} \\ v_{i+3} \end{bmatrix} \quad (2.3)$$

其中 $t = \text{FR}(ku) \equiv ku$ 的分數部份

$i = \text{INT}(ku) = ku$ 的整數部份

$k =$ 總共的軟楔段 $= n - 3$

$u \in [0, 1]$

上式稱為“均勻三次B軟楔函數”，它的向量值是由素描控制點決定，它的形狀和素描控制點間的關係可解釋如下：
當上式取 $t=0$ 時，就得下式

$$P = \frac{1}{6} (v_i + 4v_{i+1} + v_{i+2}) \quad (2.4)$$

其微分式則得

$$P' = \frac{1}{2} (v_{i+2} - v_i)$$

以及 $P'' = v_i - 2v_{i+1} + v_{i+2}$

從向量分析的幾何表示法知以上三式可用圖 2-1 表示它們的關係，我們稱 P 點為“1/3點”，因為它在三角形 $\overline{V_i V_{i+1} V_{i+2}}$ 裏的位置，是在靠近頂點 V_{i+1} 所引出中線之 1/3 長度的地方。所以，吾人很容易從已知的素描多邊形想像出它所對應之B軟楔曲線的形狀。例如從圖 2-2 的素描控制點知B軟楔曲線的起點會通過 1 點（因為 012 的三角形退化成一條直線，故 1 點通過此線段的中點），從 123 三角形可決定其 1/3 點，而使得 1 點和此 1/3 點間是一條三次曲線，而且其首尾點的斜率和彎曲方向分別可從對應的三

角形獲知，如此繼續觀察每個連續的三角形，則可將整條 B 軟楔曲線的形狀勾畫出來。

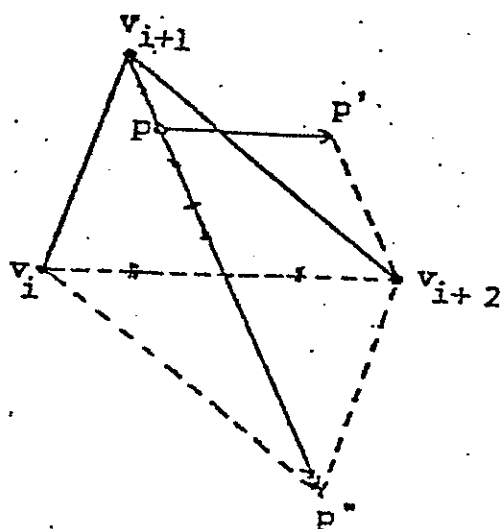


圖 2-1



圖 2-2

均勻 B 軟楔曲線除了容易從素描多邊形獲知其形狀外，最重要的特點是其表示法非常適合於“交互式電腦圖形”的作業，圖形交互式作業的目的乃欲稍微修改局部圖的形狀。由於均勻三次 B 軟楔的基底函數是低次的多項式，稍微移動（或刪除或增添）一個素描控制點，所影響者僅是局部的形狀。它的另一項重要的特點是它可以產生任何形狀的圖形，即使是有尖點的圖形，雖從直觀上言在尖點上曲線是非 C^1 連續，但從均勻 B 軟楔表示式中，只要吾人尖點上連續給三個相同的素描控制點（亦即三角形退化成一點），則在計算上通過該點仍然是連續。因此，有尖點的圖形亦是使用相同的計算式，這種情形可用圖 2-3 顯示。

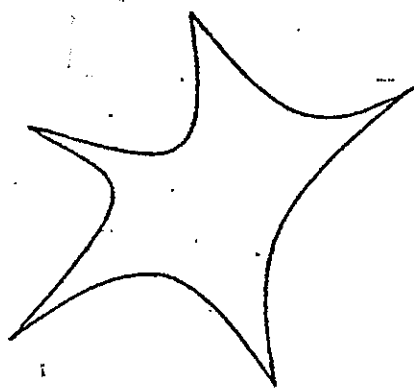


圖 2-3

2.2 輸入方法

雖然圖形在最後顯示出時為影像資料 (Image Data), 但原畫若以影像資料輸入, 則有下列幾點的不適

- (1) 不易輸入
- (2) 不易修改
- (3) 影響資料過於龐大, 浪費極大的貯存空間 (若一個圖形元素 (Pixel) 為 12 bit, 則一幅 640×512 的影像約需 1000 block 貯存才夠)。
- (4) 原畫圖形的修改不容易。

由於上述理由, 原畫的輸入是採用 B 軟楔函數來計算原畫的近似曲線。採用 B 軟楔函數的優點如下:

- (1) 易於輸入
- (2) 易於修改 (適合局部修改)
- (3) 貯存的資料非常少。

因圖形的幾何轉變或動畫的轉變時, 所需處理的資料量非常少。

唯一的缺點是在輸出圖形顯示時, 計算 B 軟楔函數需耗費較長的時間。但這項缺點, 可經由一種加上快速收斂之 B 軟楔函數的計算方法來舒解。

從 B 軟楔曲線的特性, 我們可以歸出簡單的靜畫輸入原則:
令控制點輸入依次為 $v_0, v_1, v_2, v_3, v_4, \dots, v_n$

- (1) 若 $v_0, v_1, v_2, \dots$ 連成的多邊形稱為控制多邊形, 則 B 軟楔函數的近似曲線在凸多邊形內部如圖 2-4(a)(b)。

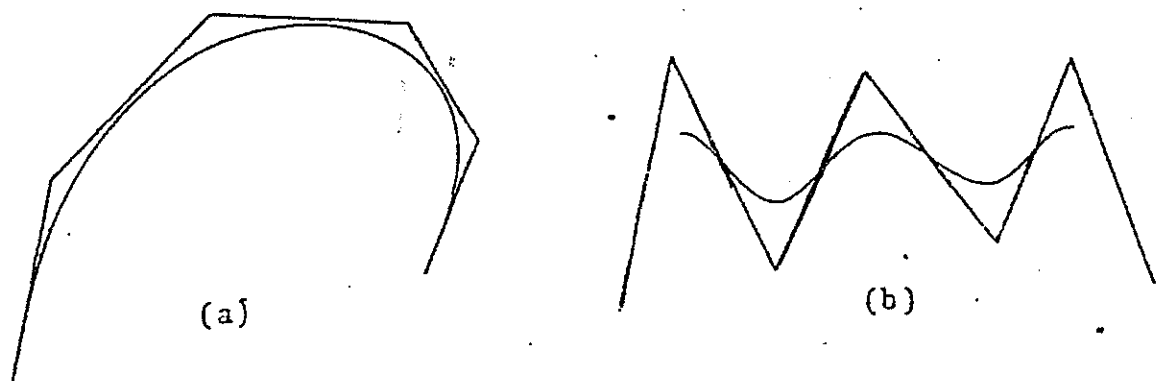


圖 2-4 B 軟楔近似曲線在凸多邊形內部

- (2) 若三個連續的控制點重合, 則近似曲線會通過此重合點。所以欲畫一直線, 則需點入二組三個重合的控制點即可, 如圖 2-5。

(3) 除三點以上重合的控制點，近似曲線不通過控制點。

(4) 若二控制點重合則近似曲線“非常”接近重合點但不通過重合點，如圖2-6。



圖2-5 近似曲線通過三個重合控制點的重合點

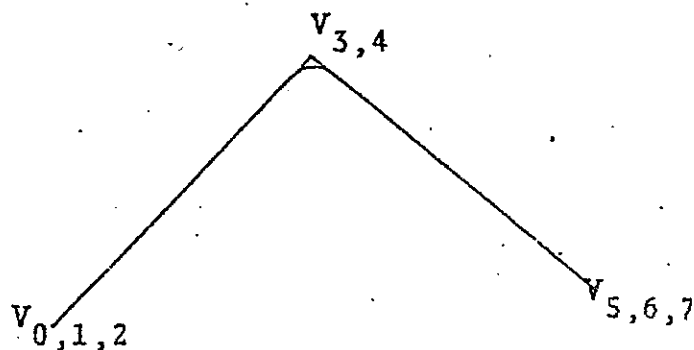


圖2-6 近似曲線非常接近二個重合控制點的重合點

(5) 任三個不重合的控制點，近似曲線會通過由其三個控制點所形成的三角形的中線，且由中線的交點到頂點的距離，恰為到底邊中點之半，如圖2-7。

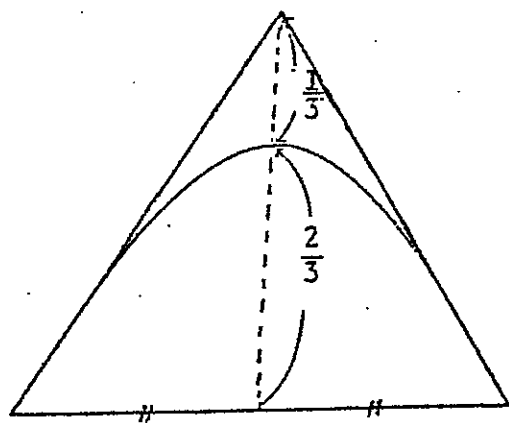


圖2-7 通過三分之一點

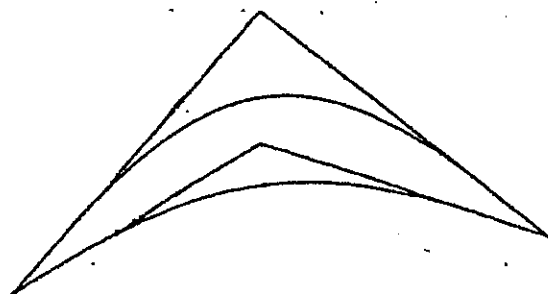


圖2-8 適合局部修改

(6) 欲改變一控制點的位置只需改變局部的近似曲線如圖2-8。從以上6點我們就可以很容易畫出任意的圖形，達成原畫的輸入，如圖2-9。

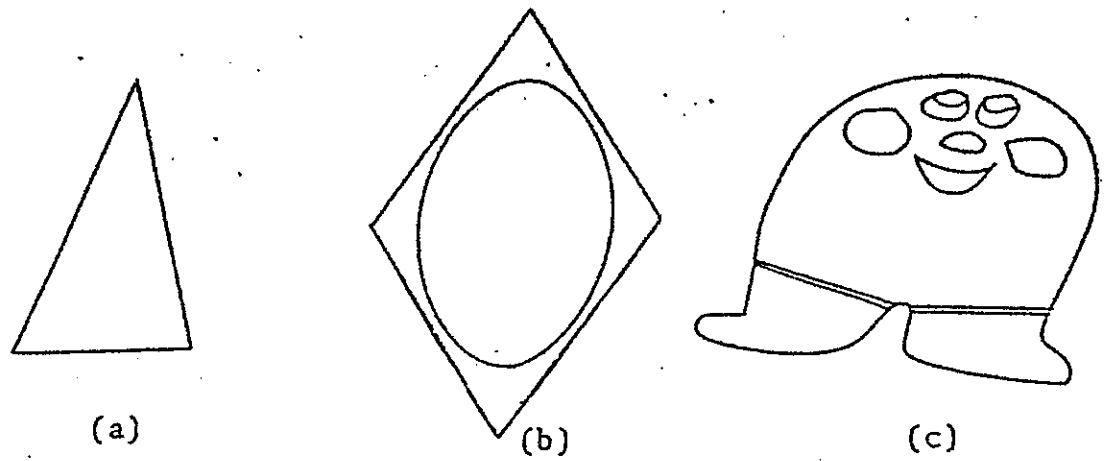
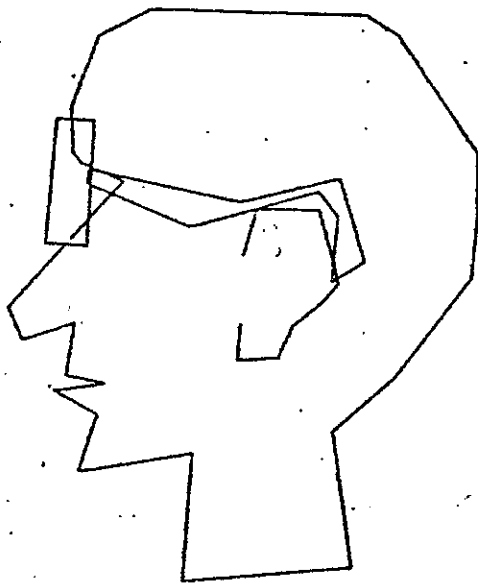


圖 2-9 根據上述五原則極易輸入圖形

雖然根據上述五原則極易輸入原畫的圖形，但是對於實際作畫的人（或者製訂自然規則的造物者）而言，是非常不合自然的作畫法，因為要產生一個圖形，需要畫控制多邊形才能產生，為什麼不能直接畫那個圖形輸入電腦？此問題及解決的辦法將在下一節中討論。

3.1 問題之敘述

目前一般應用 B 軟楔函數理論在電腦圖形設計上的作業方式是給了一組素描控制點（或多邊形），經 B 軟楔公式的計算後例如圖 3-1，給了 58 個控制點，代入均勻 B 軟楔三次曲線公式就得到如圖 3-2 的形狀。本節所要探討的問題，剛好是它的反向作業：給了一條曲線形狀（例如圖 3-2），而曲線形狀是由其上密集的取樣點所表示，那怎樣反求到它的 B 軟楔素描點（例如圖 3-1）？這些素描點是否構成最少的控制點？



控制點數 = $30+4+16+8 = 58$

圖 3-1

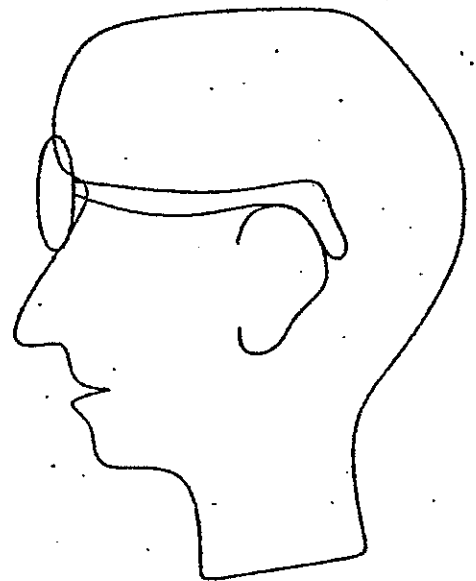


圖 3-2

令 $\{f_i\}$, $i = 0, 1, \dots, n$ 是曲線上的取樣點群，它們定義了此曲線實際的形狀。令 $B(t)$ 是一條軟楔曲線， t 是參數變數，其值在 $[0, 1]$ 間，則我們說 $B(t)$ 和 $\{f_i\}$ 所表示的曲線形狀是“完全一樣”的若它們滿足：

$$\max_i |B(t_i) - f_i| < \epsilon \quad (\text{設定誤差值}) \quad (3.1)$$

其中 $t_0 = 0$, $t_n = 1$, $t_i = t_{i-1} + \frac{1}{n}$, $i = 1, \dots, n-1$ 。換言之，

只要我們事先給了設定的誤差值，則任一條曲線的形狀是可以用 B 軟楔曲線的公式來表示的，這種誤差值的測度是取 B 軟楔曲線和取樣點間的距離偏差，所以這種測度值的大小是由使用人自訂

，可依應用上實際需要而定。

令素描多邊形 C_1 和 C_2 分別產生了 $B_1(t)$ 和 $B_2(t)$ 的二條 B 軟楔曲線，則我們說素描多邊形 C_1 和 C_2 產生了相同形狀的 B 軟楔曲線，若下面的條件成立：

$$\max_i |B_1(t_i) - B_2(t_i)| < \epsilon \text{ (設定誤差值)} \quad (3.2)$$

換言之，雖然一個素描多邊形只能對應（或說產生）一條 B 軟楔曲線，但是反過來說，則不成立。亦即，一條 B 軟楔曲線可以由許多不同的（事實上是無窮多）素描多邊形分別產生。為什麼呢？從 (3.2) 式可知一條 B 軟楔曲線可以由許多近似的 B 軟楔曲線逼近得到，這是一般近似理論裏最基本的觀念。例如圖 3-3 的曲線形狀，可以由圖 3-4 裏二個不同的素描多邊形分別產生。

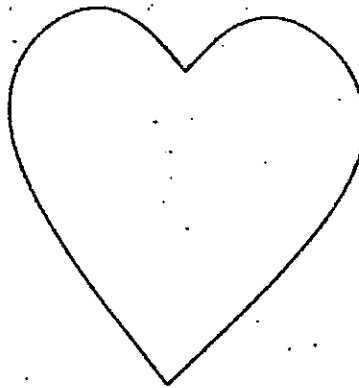


圖 3-3

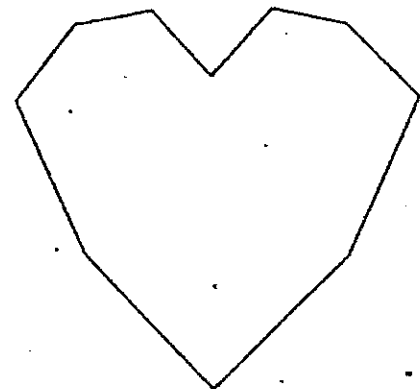
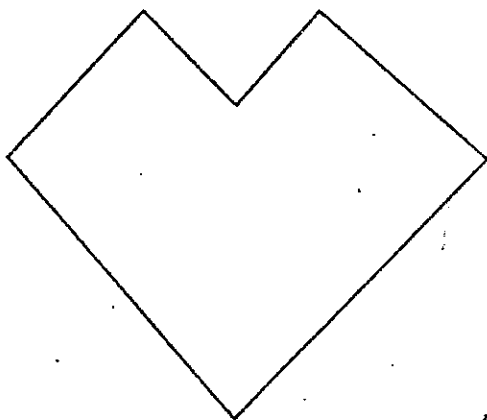


圖 3-4

既然一條 B 軟楔曲線是對應於許多素描多邊形，乃使問題複雜化了，因為我們要從衆多可能的素描多邊形中去挑選具有最少控制點數的多邊形。令 $C_p = \{\xi_i\}$ 是個素描多邊形， ξ_i 是其控制點，令 $o(\xi_i)$ 表示控制點 ξ_i 在 B 軟楔曲線公式中被計算到的次數，

則我們說素描多邊形 C_p 總共的控制點數是

$$|C_p| \equiv \sum_i \sigma(\xi_i) \quad (3.3)$$

令 $\{C_p\}$ 是對應一條 B 軟楔曲線 B 的素描多邊形群，則我們的問題是從 $\{C_p\}$ 中找一個素描多邊形 C_d ，它具有下面的特性：

$$|C_d| = \min_p |C_p| \quad (3.4)$$

但是 C_p 是從近似 B 軟楔曲線反求所得，也就是說二條 B 軟楔曲線雖然有點差異，但是只要它們的最大偏離值小於設定的誤差，則我們又說這二條曲線的形狀是完全一樣，如此反求到的二個素描多邊形皆是屬於 C_p ，因此從各種不同情況（但保持在設定誤差值內），要找到一個 C_d 滿足 (3.4) 式，只能用排列組合去搜尋，其複雜度高到簡直是不可能求到解。我們只期望得到一種反求 B 軟楔演算法。它具有和人為的解互相抗衡的能力。換句話說，給了一任意的圖形，使用人能夠用眼觀的方法去猜測到一個素描多邊形，這個素描多邊形經 B 軟楔函數運算後，所得到的曲線形狀和已知形狀是一樣的（在誤差範圍內），假設這個素描多邊形的控制點數是 σ_1 ，而用演算法 A 反求得到的素描多邊形具有 σ_2 個控制點，則只要 σ_1 和 σ_2 很接近，那麼我們就說演算法 A 是具有 (3.4) 式特性的一個解。

3.2 求解的因素考慮

從問題的敘述中，我們得知反求 B 軟楔演算法可以有很多種，但是一種演算法 A 必然只是得到一種結果，也就是說一條曲線經演算 A 運算後，只能得到一個素描多邊形，從欲達到最佳壓縮的觀點言，若 A_1 決定的素描控制點少於 A_2 所決定者，我們說演算法 A_1 優於 A_2 。

無論如何，當演算法一旦決定後，任意二條完全一樣（在誤差範圍內）的 B 軟楔經此演算法處理後，只能得到形狀完全一致的素描多邊形，因為唯有這樣的演算法才有意義。

一種反求 B 軟楔演算法處理一條曲線只會得到一個素描多邊形，則我們說此條曲線具有的圖形不變值 (GRAPHIC INVARIANT) 是其控制點數，不同的控制點數必然決定了不同形狀的圖形，這裡所謂的不同形狀是指多邊形經幾何運算（旋轉、平移、放大縮

小) 也不能滿足 (3.2) 式。

圖形不變值是個非常重要的值，它可以是將曲線形狀定量的首要值，所以在以處理 B 軟楔為主之圖形資料庫 (PICTORIAL DATA BASE) 的設計上，它可以當做主鍵 (PRIMARY KEY) 或副鍵 (SECONDARY)，如此，在搜尋一條曲線時，可以先找到其圖形不變值的範圍，而大大地減少了搜尋的時間，因此，本節所探討的演算法在圖形資料庫的設計上亦提供了具體的貢獻。

圖 3-5 示出二個不同圖形不變值 (素描多邊形總共的控制點數)，所對應之二條不同形狀的曲線，亦即不同的圖形不變值一定決定不同形狀的曲線，但是相同的圖形不變值是否一定對應相同形狀的曲線？答案是否定的，例如圖 3-6 所示的二條不同形狀的曲線却具有相同的圖形不變值，所以我們知道控制點數只是曲線形狀定量上的一個不變值，應還有更多的圖形不變值存在，研究怎樣去尋找到其他的圖形不變值是個相當重要的課題。

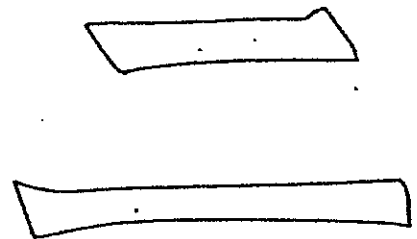
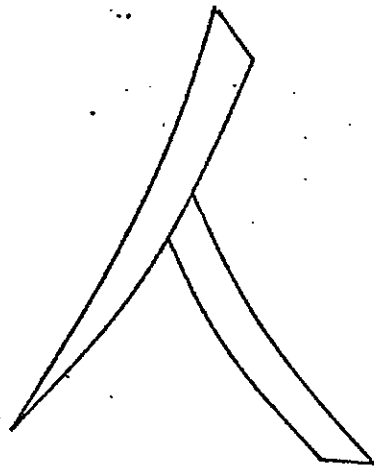
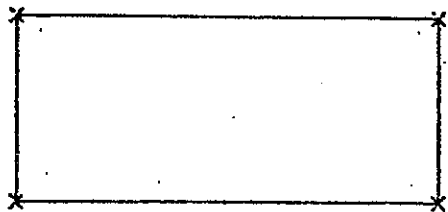
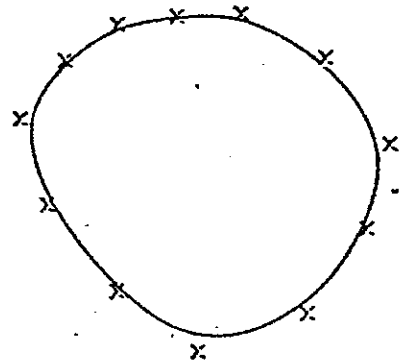


圖 3-5

尖點含有 3 點控制點



共 12 點



共 12 點

圖 3-6

從圖 3-5 的例子可以看出，本文的主要結果和將來的中文字形研究有密切的關連，因為中文字形的每一個筆畫也可以有定量了，這對於中文字的加碼 (ENCODING) 有很大的助益。但是怎樣的演算法才能得到唯一的圖形不變值？從觀察可知，不管多複雜的曲線形狀，它皆是簡單形狀所接合而成，也正式這項理由，B 軟楔曲線公式才能表示任意的曲線形狀，所以吾人所要做的事情是把曲線分段，使每段皆是簡單的曲線形狀，這種工作需由演算法自動完成，因此我們首先需將曲線的形狀，依 B 軟楔曲線表示式的特性分門別類。在本節裏，我們將曲線段分成四類，它們分別是 (1) 直線段，(2) 尖點段，(3) 反曲段，(4) 平滑段。而且決定每段的首尾點是固定的程序，這點保證了：若二條曲線形狀完全一樣，只是一條是由另一條經幾何變換所得者，則它們的類別一樣多，而且每類皆保持了二者間形狀完全一樣的特性。

其次是對不同類的曲線段分別用不同的處理法，每個處理法的重點之一是在保持形狀的完全一致性，因此經處理法運算後，每段的形狀是在設定誤差值的範圍內。另外一項重要的工作，是各類的處理法必需是尋求對應其類之曲線最少點的表示法。最少點的意義是從曲線上的取樣點中，抽取對曲線形狀有重要貢獻者，有了這些重要的點，則僅從他們的數據，也可以決定一條和它完全一樣的曲線。

對曲線形狀有重要貢獻者是那些點？判別的方法就是：若在已選出的一組點中，將它去除，則用剩下的點來決定曲線的形狀時，所得的結果會產生和原曲線不一樣的形狀，因此依照數值分析的技巧，演算法中可以採用選代的方法，反覆測試這種判別的結果，即可將這些具有對決定曲線形狀有影響力的點找出，本節將採用 Bernstein-Bézier 逼近多項式的反覆測試，來找出最少的形狀貢獻點，而且經這種選代法所得到的點是唯一的（假設誤差設定是一個定值）。

綜上所述，一種反求 B 軟楔演算法的主要步驟是分成三部份，第一部份將已知的曲線形狀分段，決定每段的類別，第二部份是對每類的曲線段分別求最少的形狀貢獻點，第三部份是對這些貢獻點依其性質反求 B 軟楔素描控制點，決定曲線的圖形不變值，這種演算法的過程可用圖 3-7 的方塊圖表示。

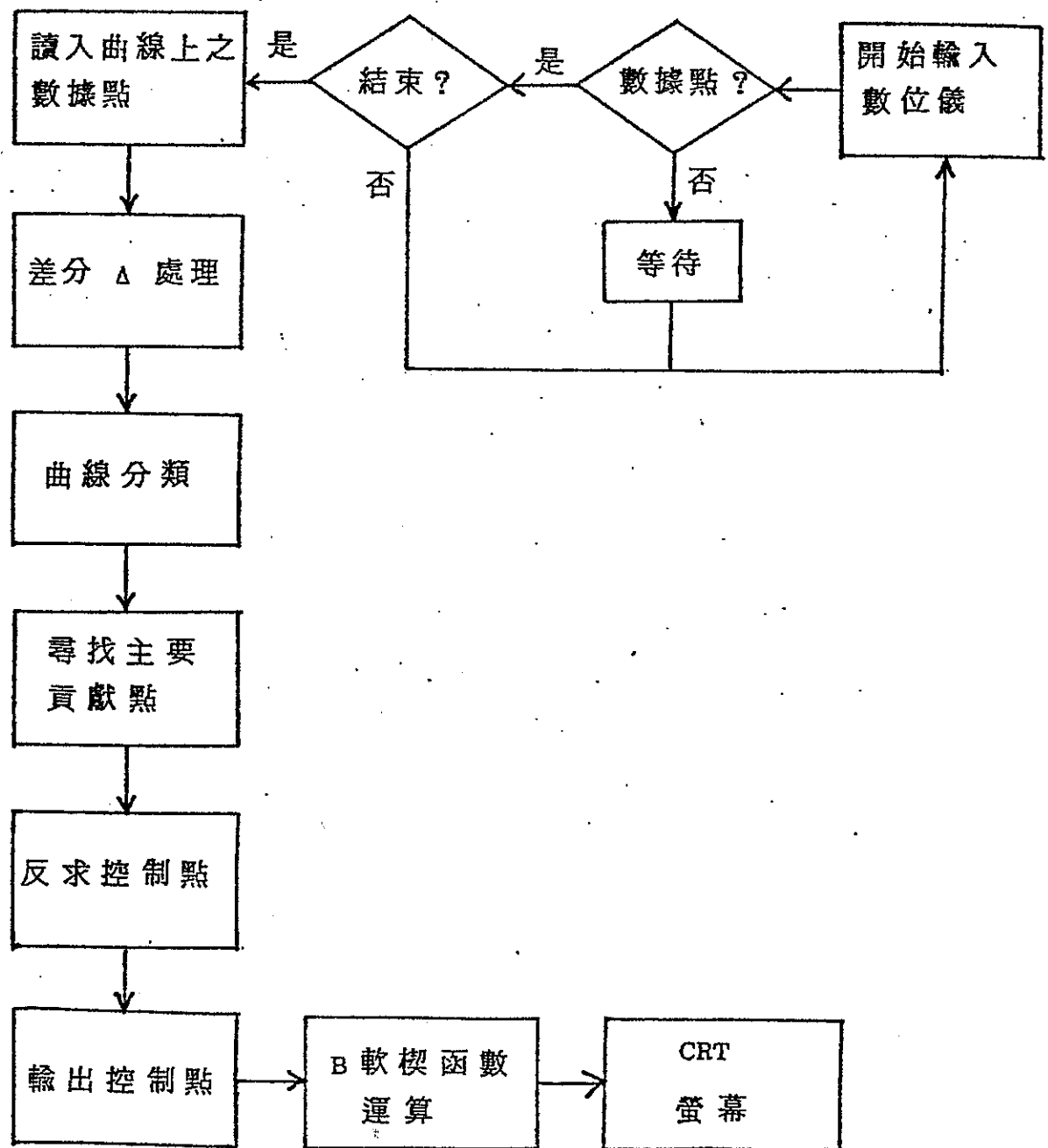


圖 3-7

3.3 主要結果

一種反求 B 軟楔演算法包括了：(1) Δ 處理法，(2) 頂點投影法，(3) 特殊線段分類處理法，(4) 第三類曲線段之極少數據點法，(5) 第二類曲線段之極少數據點法，(6) 求 B 軟楔曲線及反求控制點，(7) 第四類曲線段之極少數據點法，(8) 圖形數據壓縮法，它們的方法分述如下：

3.3.1 Δ 處理法

令 $[x_i, y_i]$, $i = 0, 1, 2, \dots, n-1$, $x_0 = x_n, y_0 = y_n$ 為數位儀(Digitizer)之輸入數據點，則算子 Δ 的定義是

$$\begin{aligned}\Delta x_i &= x_{i+1} - x_i & \Delta y_i &= y_{i+1} - y_i \\ \Delta^2 x_i &= \Delta(\Delta x_i) & \Delta^2 y_i &= \Delta(\Delta y_i) \\ &= \Delta x_{i+1} - \Delta x_i & &= \Delta y_{i+1} - \Delta y_i \\ &= x_{i+2} - 2x_{i+1} + x_i & &= y_{i+2} - 2y_{i+1} + y_i\end{aligned}$$

若 $[x_{i+1}, y_{i+1}] \equiv [x_i, y_i]$ ，則 $\Delta x_i \equiv \Delta y_i \equiv 0$ ，表示數位儀的光筆停止不動，此時 $[x_{i+1}, y_{i+1}]$ 點不讀入，數據點讀入的情況在“等待環”(WAITING LOOP)，否則讀入 $[x_{i+1}, y_{i+1}]$ 點。

令 $\text{VAR}(\Delta)$ 表示在 Δ 處理中有符號變化，例如邏輯函數 $\text{VAR}(\Delta x_i)$ 表示“若 $\Delta x_i < 0$ 則 $\Delta x_{i-1} > 0$ ”，反之亦然，令 $\text{ANG}(\Delta)$ 為在產生 $\text{VAR}(\Delta)$ 處的夾角值，則若 $\text{ANG}(\Delta)$ 小於某一設定值 σ ，我們稱每個這樣的數據點為“頂點”(CLIMAX POINT)。若曲線在頂點是凸出(亦即法線向外)，則稱“正頂點”，反之則稱“負頂點”。若頂點 $[x_i, y_i]$ 的鄰近點 $[x_{i-1}, y_{i-1}]$ 和 $[x_{i+1}, y_{i+1}]$ 間的距離小於某一設定值 ϵ ，則我們稱此頂點為“尖點”(CUSP POINT)。

綜上所述，當數位儀數據點經 Δ 處理後可將曲線上的幾個重要點挑出，這些點分別是：

- 1 直線段的二個端點，因為直線段的 $\Delta y / \Delta x$ 是一常數。
- 2 各個正或負頂點。
- 3 各個正或負尖點。

3.3.2 頂點投影法

頂點分成凸點及凹點兩種情況

- 1 對於凸點，先求頂點的分角線為一準線，將數據點以頂點為準分成二部份，兩部份分別由頂點依次計算數據點與準線的距離，再用 Histogram 求出與準線距離最長的點，此兩點我們稱“轉點”。
- 2 對於凹點，此時的準線為通過頂點、垂直分角線的直線，再以同樣的方法找出兩轉點。

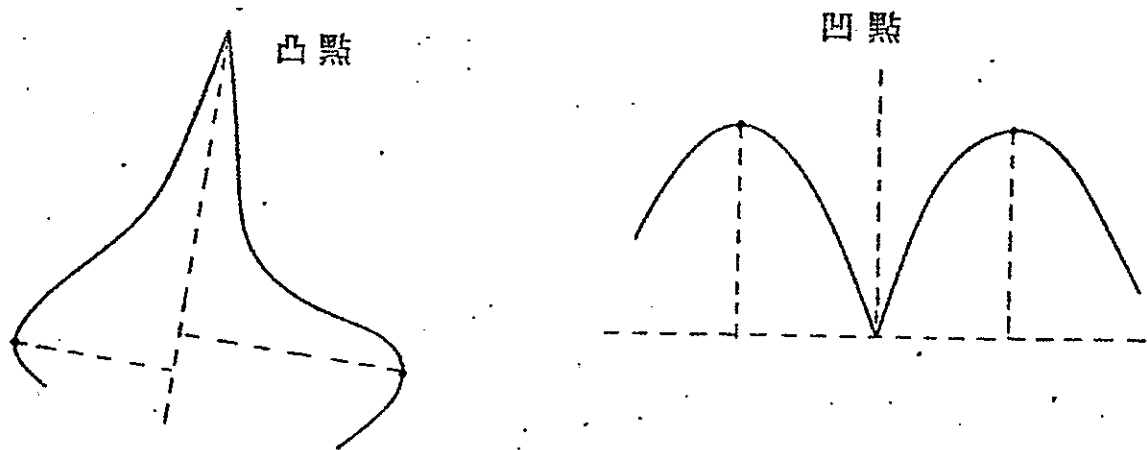


圖 3-8

頂點投影法如圖 3-8 是爲了處理尖點附近線段的分類兩轉點，分別爲“第四類”線段的始、終點。

另一功能是找出隱藏的頂點，如圖 3-9, A 爲一頂點，B 爲一隱藏的頂點，用如前凹點的求法可找到疏忽的 B 點。

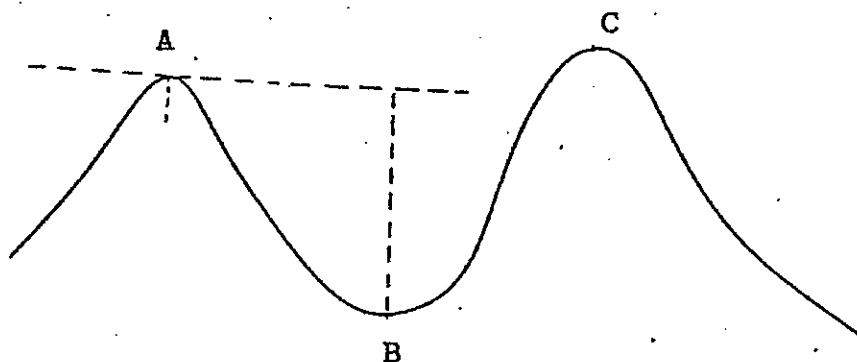


圖 3-9

3.3.3 特殊線段分類處理

選擇從 Δ 處理中首次獲得的尖點當做曲線段分類處理的起點，並依筆順的方向查詢曲線上的重要點，其情況可分述如下：

- 1 若相鄰二個重要點是直線段的端點，則此線段是歸屬於“第一類”(TYPE 1)。
- 2 若相鄰二個重要點皆是頂點，其中一個是正頂點，另一個是負頂點，則用“頂點投影法”查它們之間是否尚有隱藏的頂點（亦即大於設定值 σ 者）。若無，則此線段歸屬於“第二類”(TYPE 2)，否則第一個頂點和隱藏頂點間的線段才是第二類，而把隱藏頂點顯現納入重要點序列中繼續查詢。

3 若相鄰二個重要點是相同符號的頂點，例如皆為正頂點，則用“頂點投影法”查它們之間是否尚有隱藏的負頂點。若有，則此線段歸屬於第二類，並將隱藏頂點納入重要點序列，否則歸屬於“第三類”(TYPE 3)。

4 用“頂點投影法”處理尖點所得到之線段歸屬於“第四類”(TYPE 4)

5 依上述處理後剩下之線段皆歸屬於“第三類”。

綜上所述，經特殊線段分類處理後，將曲線 S 分切成 $\{S_i\}$, $i = 1, 2, \dots, m$, 其中 S_i 只能歸屬於四類中的一類。

3.3.4 第三類曲線段之極少數據點法

令 S_i 是第三類曲線段，則在此線段中沒有反曲點，它是“平滑”的曲線段，從振變消失 (Variation Diminishing Property) 的觀點看，它和任一條直線的交點頂多有數點。從近似理論知，吾人可採用 Bernstein 多項式近似這樣平滑的曲線段，而且在設定誤差的範圍內，所求得最低次的 Bernstein 多項式近似，其次數就決定了極少數據點。

令 $P_j = [x_{ij}, y_{ij}]$, $j = 0, 1, \dots, n$ 是 S_i 上之數位儀數據點，則 n 次 Bernstein-Bezier 多項式近似 S_i 的公式為

$$B_n(S_i) = \sum_{k=0}^n P_k \phi_{k,n}(t) \quad (3.5)$$

其中 $\phi_{k,n}(t) = \binom{n}{k} t^k (1-t)^{n-k}$, $k \in [0, n]$, $t \in [0, 1]$ 它有下列數個性質：

- 1 $B_n(S_i)$ 會通過 S_i 的起點和終點，而一般言是不通過中間的點。
- 2 $B_n(S_i)$ 在首尾點的斜率分別等於起點和第二點連線以及終點前一點和終點連線的斜率。

3 $B_n(S_i)$ 的範圍被限圍在 P_j 點所連成的凸集合內。

從這些性質可知 $B_n(S_i)$ 和 P_j 點所連成多邊形相比較，它具有振變消失的特性，所以若 S_i 是相當平滑時，吾人不難找到甚少的點 C_k (稱為控制點)，除首尾點外皆不在 S_i 上，在 (3.5) 式中置換 P_k 求 $B_n(S_i)$ ，則 $B_n(S_i)$ 的形狀是非常近似 S_i 的，這種求法的過程如下：

- 1 令 $n \doteq n$ ($\doteq$ 表示被置換， ρ 表示 ρ 的最小整數)，並以對應的等分點當做 $B_n(S_i)$ 上的點代入 (5)，反求控制點 C_k 。

2 若用 C_k 所求得的 $B_n(S_i)$ 具有

$$|B_n(S_i) - P_j| < \epsilon \quad (\text{設定的誤差}), j = 1, \dots, n-1 \text{ 則做 3,}$$

否則做 4。

3 反覆做此步驟 (DO FOREVER), 儘量去除控制點 $\{C_{k_p}\}$, 用 SORT $\{d(C_{k_p}, P_{k_p})\}$, d 表示距離, 亦即反求過程中控制點和對應 $B(S_i)$ 上點間的距離, 求遞增序列, 用二分切除法嘗試去除一半控制點後, 是否 Bernstein-Bézier 曲線仍在設定的誤差, 當做到不能再減少時, 則停止。

4 令 $n=2n$ 再回 1。

3.3.5 第二類曲線段之極少數據點法

令 S_i 是第二類曲線段, 則在此線段中有反曲點的存在, 它是 "彎曲" 的曲線段, 若可以找出反曲點, 則由反曲點分別到其相鄰的二個頂點間的二條線段是平滑的, 所以處理此類曲線段的首要工作是找出反曲點, 然後再以第三類曲線段之極少數據點法分別求其前後二線段的極少點。

令符號相反的二個相鄰頂點是 S_i 的首尾點, $P_j = [x_{ij}, y_{ij}]$, $j = 0, 1, \dots, q, \dots$, λ 是 S_i 上的點, 其中 P_q 是反曲點, 則第二類曲線段之極少數據點法如下:

1 用 HISTOGRAM 法從 Δ 處理中找出 P_q 點, 因為 $\Delta^2 P_q$ 和 $\{\Delta^2 P_j\}$ 比較是最小的。

2 以 P_q 為分段點, 將 S_i 分成二段 S_{i1} 和 S_{i2} 。

3 分別以第三類曲線段之極少數據點法處理 S_{i1} 和 S_{i2} 。

3.3.6 求 B 軟楔曲線及反求控制點法

在數學上 B 軟楔曲線是用參數方程式來表示, 假設向量點 $\{V_i\}$ 為一素描多邊形的端點 (或稱素描控制點), 則此多邊形所決定 $m-1$ 次 B 軟楔曲線為

$$P(u) \equiv [x(u), y(u)] = \sum_{i=0}^n V_i \phi_{i,m}(u) \quad (3.6)$$

其中 u 為參數, $\phi_{i,m}$ 為 $m-1$ 次 B 軟楔第 i 個基底函數, $\phi_{i,m}$ 是一 m 次多項式, 因此我們可以知道 B 軟楔函數屬於 C^{m-2} , 若我們有一素描控制點 $\{V_0, V_1, \dots, V_n\}$, 為簡單起見我們取 $m=4$, 令 $t=(u-u_i)/(u_{i+1}-u_i)$, $u \in [u_i, u_{i+1}]$, 則 $t \in [0, 1]$, (3.6) 式可簡化為

$$P_i(t) = [t^3, t^2, t, 1] \frac{1}{6} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix} \begin{bmatrix} V_{i-1} \\ V_i \\ V_{i+1} \\ V_{i+2} \end{bmatrix} \quad (3.7)$$

$$i = 0, 1, \dots, n$$

$P_i(t)$ 是由 $V_{i-1}, V_i, V_{i+1}, V_{i+2}$ 所決定的 B 軟楔曲線段，(3.7) 式稱為 " 均勻三次 B 軟楔函數 "。

若我們已知一曲線上的點 $\{f_i\}_{i=0(1)n}$ ，如何利用 (3.7) 式來反求 $\{V_i\}_{i=0(1)n}$ ？可分為 ① 開放曲線 (open curve) 及 ② 封閉曲線 (closed curve) 二種情況來討論。如 (3.7) 式我們令此曲線由 n 個 B 軟楔曲線段 $\{P_i(t)\}_{i=0(1)(n-1)}$ 所組成，因此 $f_0 = P_0(0)$ ， $f_1 = P_0(1) = P_1(0)$ ， $\dots$ ， $f_{n-1} = P_{n-2}(1) = P_{n-1}(0)$ ， $f_n = P_{n-1}(1)$ ，代入 (3.7) 式，有如下結果：

$$\begin{aligned} f_0 &= P_0(0) = \frac{1}{6}(V_{-1} + 4V_0 + V_1) \\ f_1 &= P_1(0) = \frac{1}{6}(V_0 + 4V_1 + V_2) \\ f_2 &= P_2(0) = \frac{1}{6}(V_1 + 4V_2 + V_3) \\ &\vdots \\ f_{n-1} &= P_{n-1}(0) = \frac{1}{6}(V_{n-2} + 4V_{n-1} + V_n) \\ f_n &= P_{n-1}(1) = \frac{1}{6}(V_{n-1} + 4V_n + V_{n+1}) \end{aligned} \quad (3.8)$$

(3.8) 中共有 $n+1$ 組方程式，欲解出 $\{V_{-1}, V_0, \dots, V_n, V_{n+1}\}$ 共 $n+3$ 個解條件是不夠的，問題發生在端點 f_0, f_n 。因每個軟楔曲線段的端點是由附近 3 個控制點所決定，所以我們先考慮端點的 3 個控制點間之關係，也就是說我們再加入兩個條件，使 (3.8) 式可以解。此處是採用 S.C.Wu, John F. Abel, & Donald P. Greenberg [7] 所提出的方法，我們稱 V_{-1}, V_{n+1} 為虛點，其他 $V_0, V_1, \dots, V_n$ 稱為實點，如圖 3-10，假設此開放曲線兩端曲率為 0，即 $\{V_{-1}, V_0, V_1\}$ 和 $\{V_{n-1}, V_n, V_{n+1}\}$ 分別共線，那麼 V_{-1}, V_0, V_1 所決定的點必為 V_0 ，而 V_{n-1}, V_n, V_{n+1} 所決定的點必為 V_n ，演算過程如下：

由 (3.8) 式知：

$$\begin{aligned} P_0(0) &= \frac{1}{6} (V_{-1} + 4V_0 + V_1) \\ P_{n-1}(1) &= \frac{1}{6} (V_{n-1} + 4V_n + V_{n+1}) \end{aligned} \quad (3.9)$$

因爲端點曲率爲 0，所以

$$\begin{aligned} P_0''(0) &= V_{-1} - 2V_0 + V_1 = 0 \\ P_{n-1}''(1) &= V_{n-1} - 2V_n + V_{n+1} = 0 \end{aligned} \quad (3.10)$$

由 (3.10) 式知：

$$\begin{aligned} V_{-1} &= 2V_0 - V_1 \\ V_{n+1} &= 2V_n - V_{n-1} \end{aligned} \quad (3.11)$$

因此我們已將兩虛點用鄰近它們的實點來表示，將 (3.11) 式代入 (3.9) 式，則得：

$$\begin{aligned} P_0(0) &= V_0 \\ P_{n-1}(1) &= V_n \end{aligned} \quad (3.12)$$

(3.12) 式代入 (3.8) 式：

$$\begin{aligned} f_0 &= V_0 \\ f_1 &= \frac{1}{6} (V_0 + 4V_1 + V_2) \\ &\vdots \\ f_{n-1} &= \frac{1}{6} (V_{n-2} + 4V_{n-1} + V_n) \\ f_n &= V_n \end{aligned} \quad (3.13)$$

此時 (3.13) 式爲一可解的線性方程組，所以我們得到：

$$\begin{aligned} \frac{1}{6} M_0 [V_0, V_1, \dots, V_n]^T &= [f_0, f_1, \dots, f_n]^T \\ M_0 &= \begin{bmatrix} 6 & & & & \\ 1 & 4 & 1 & & \\ & 1 & 4 & 1 & \\ & & \ddots & \ddots & \ddots \\ & & & 1 & 4 & 1 \\ & & & & & 6 \end{bmatrix} \end{aligned} \quad (3.14)$$

所以

$$\begin{pmatrix} V_0 \\ V_1 \\ \vdots \\ V_n \end{pmatrix} = \frac{1}{6} M_0^{-1} \begin{pmatrix} f_0 \\ f_1 \\ \vdots \\ f_n \end{pmatrix} \quad (3.15)$$

以上為開放曲線的情形，封閉曲線就不必再造兩個虛點，事實上它的起點是由 $\{V_n, V_1, V_2\}$ 所決定，終點事實上等於起點，所以我們令 $V_{-1} = V_n$, $V_{n+1} = V_0$, $V_{n+2} = V_1$ 代入 (3.8) 式得

$$\begin{aligned} f_0 &= \frac{1}{6} (4V_0 + V_1 + V_n) \\ f_1 &= \frac{1}{6} (V_0 + 4V_1 + V_2) \\ f_2 &= \frac{1}{6} (V_1 + 4V_2 + V_3) \\ &\vdots \\ f_{n-1} &= \frac{1}{6} (V_{n-2} + 4V_{n-1} + V_n) \\ f_n &= \frac{1}{6} (V_1 + V_{n-1} + 4V_n) \end{aligned} \quad (3.16)$$

即

$$\frac{1}{6} M_c [V_0, V_1, \dots, V_n]^T = [f_0, f_1, \dots, f_n]^T$$

$$M_c = \begin{pmatrix} 4 & 1 & & & & 1 \\ 1 & 4 & 1 & & & \\ & 1 & 4 & 1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & 1 & 4 & 1 \\ 1 & & & & 4 & 1 \end{pmatrix} \quad (3.17)$$

因此得到

$$\begin{pmatrix} V_0 \\ V_1 \\ \vdots \\ V_n \end{pmatrix} = \frac{1}{6} M_c^{-1} \begin{pmatrix} f_0 \\ f_1 \\ \vdots \\ f_n \end{pmatrix} \quad (3.18)$$



圖 3-10

任何曲線我們只要知道圖形上的數據點，則可反求出代表此曲線的某一素描多邊形。若這些點 $\{f_i\}$ 足夠密，則我們可由所求出的素描控制點得到正確的圖形，但這樣很不實際，事實上這圖形不必用如此多的點來表示，所以本文為使這種方法效果增強，於反求控制點前，先做一連串的先前處理，為的是能找出最具代表性的點，只要用這些代表點，所反求的素描多邊形即能最有效的代表此圖形，也就是說，這素描多邊形所決定的B軟楔曲線與原來圖形誤差將在某一可容範圍內。

3.3.7. 第四類曲線段之極少數據點法

這方法用於第四類曲線段——含有尖點的曲線段，此種曲線段變化無窮如圖(3-11)，我們不能僅由尖點及兩端點來描述此曲線段，而應於尖點兩邊各尋找一至二個重要點，將來才不致於求出一誤差甚大的曲線。要找這些重要點，尖點兩邊應以相同的方法來做。對於一邊之重要點的尋找方法步驟如下：

- 1 先找出尖點與端點之切線，此二線必交於一點或重合然後進行2或3。
- 2 若二線重合，表示這一邊為直線，因此中間不必給貢獻點，但端點視為二貢獻點（圖3-11），則可結束。
- 3 若二線相交於一點，找出此點C，如圖3-12，我們欲知 $\overline{CE}$ 的長度，其中D為 $\overline{AB}$ 之中點，E為 $\overline{CD}$ 與 $\widehat{AB}$ 交點，然後進行4或5。
- 4 若 $\overline{CE} \geq \frac{1}{3} \overline{CD}$ ，則直接加E為“貢獻點”。
- 5 若 $\overline{CE} < \frac{1}{3} \overline{CD}$ ，依 $a = \overline{CE} / \overline{CD}$ 的比率，加二“貢獻點”於 $a * \widehat{EB}$ 及 $a * \widehat{EA}$ 處。

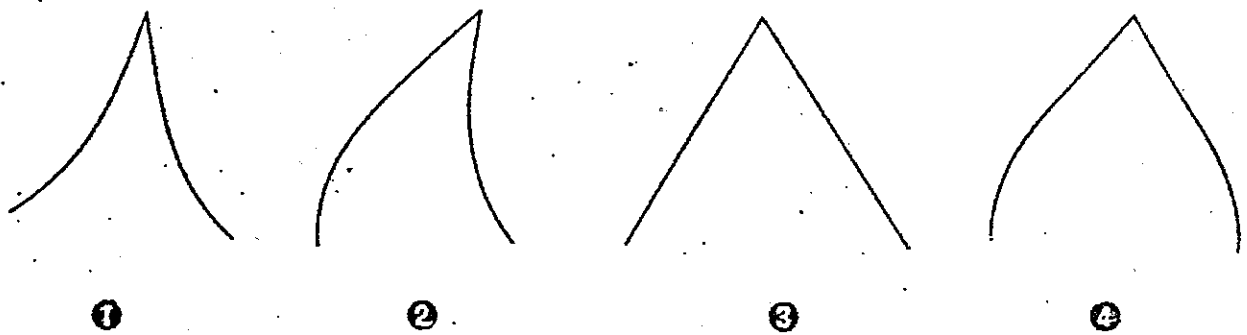


圖 3-11

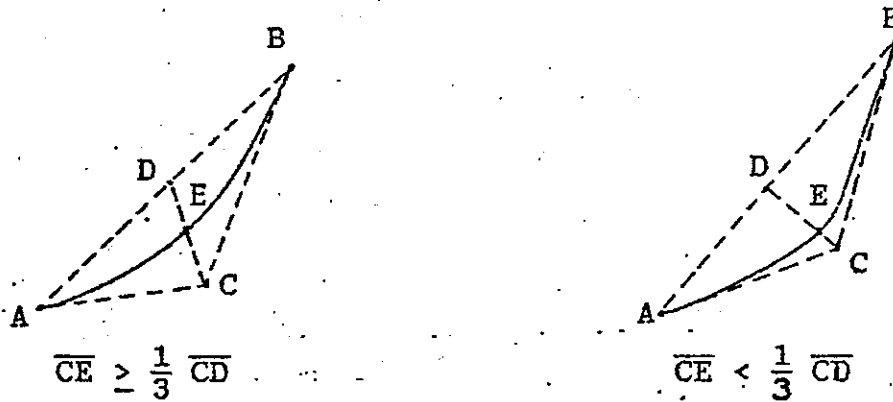


圖 3-12

STEP 4,5 的做法及原因在此說明一下，當 $\overline{CE} \geq \frac{1}{3} \overline{CD}$ 時，曲線必較平滑，可用較少的點來代表，但 $\overline{CE} < \frac{1}{3} \overline{CD}$ 時，曲線彎曲大，因此於彎曲處應多一些點來表示，如果 $\overline{CE} = \frac{1}{3} \overline{CD}$ ，則由 B 軟楔曲線的性质，C 必為一控制點。

3.3.8 圖形數據壓縮法

我們所處理的對象是以有筆畫順序所繪制的圖形為主，在此處所提的壓縮法亦是以處理這類圖形為主要對象，令這類圖形的集合為 O ，任何 O 裏的元素是由一個有序數據點的集合代表其圖形，令圖形是 S ，則 $S = \{P_k\}$ ， $P_k = [x_k, y_k]$ 。

圖形數據壓縮法可以將 O 裏的任一圖形 S 用最少的記憶單元儲存 S 的資訊，當這些資訊被抽取出來計算後，可以得到一個圖形，它和 S 的圖形是非常相似的，或者說在設定的誤差容忍值內它們的形狀是一致的。

表現圖形的資訊可以有許多種形態，每種形態的表現法可能需要的記憶單元是不一樣的，因此若用最少的記憶單元的圖形資訊當做壓縮法輸出的條件，就比較困難。因為要採用何種形態的表現法和所應用的對象有關，所以此處我們以 B 軟楔函數的控制點數是最少作為其條件，而輸出是這些控制點的座標值。採用這種形態的表現法雖然不一定得到最少的記憶單元，但它的好處是其資訊的完整性，若欲轉換成其他形態的表現，則可以根據這些控制點當做轉換式的輸入數據。

圖形數據壓縮法

輸入：任意圖形 S 的數據點

輸出：最少 B 軟楔控制點

方法：1 從 Δ 處理算出曲線 S 上顯著的重要點。

2 從特殊曲線段分類處理算出各種類別的線段 $\{l_i\}$

3 將各個 l_i 可能隱藏的重要點算出。

4 將各個 l_i 的主要點數據 (KEY DATA) 算出，這些主要點數據就構成曲線上最少應有的座標點。

5 將尖點個別處理並反求 B 軟楔控制點。

3.4 實例示範

為證明此反求 B 軟楔演算法的成果，現以一實例來說明，(圖 3-1) 是人為輸入的控制點，這些控制點經 B 軟楔函數運算則得 (圖 3-2)，(圖 3-2) 事實上是分成四個部份——頭 (圖 3-13)、鏡片 (圖 3-14)、鏡柄 (圖 3-15)、耳朵 (圖 3-16)，每一部份人為輸入的控制點數分別為：頭 30 點、鏡片 4 點、鏡柄 16 點、耳朵 8 點，現吾人欲輸入 (圖 3-2) 圖形上的數據點，希望可求得最少的控制點。(表 3-1) 為所輸入的數據點，這些點數相當多，而且我們也希望它愈密愈好，如此經分析起來，才不會失去圖形的原形，表一的數據點，四部份分別經圖形數據壓縮演算法所得到的控制點數分別為：(表 3-2)——頭 50 點、鏡片 15 點、鏡柄 34 點、耳朵 14 點，圖 3-17、3-18、3-19、3-20 分別為它們的素描多邊形，這些控制點數雖比人為的控制點數多，但能由那多的輸入數據點而減少到如此少的控制點來儲存圖形，效果已算不錯

了。圖 3-21、3-22、3-23、3-24 是圖 3-17、3-18、3-19、3-20 分別經 B 軟楔函數計算而得的圖形，圖 3-25、3-26、3-27、3-28 爲輸出的圖形與輸入的圖形重合，由此可見，它們的誤差並不大，圖 3-17 ~ 3-20 確實可視爲代表圖 3-2 的素描多邊形。圖 3-29 爲四部份的素描多邊形（即圖 3-17 ~ 3-20）之組合，讀者可與圖 3-1 比較之，圖 3-30 爲圖 3-21 ~ 3-24 之組合，可與圖 3-2 比較之。

原始數據

頭	次序 座標	1	2	3	• • • • •	317	318
	X	486.9	489.9	492.7	• • • • •	476.9	480.5
	Y	644.9	643.1	641.0	• • • • •	649.0	647.8

鏡片	次序 座標	1	2	3	• • • • •	82	83
	X	283.5	284.0	284.4	• • • • •	282.1	282.7
	Y	557.5	555.7	553.8	• • • • •	516.4	560.0

鏡柄	次序 座標	1	2	3	• • • • •	184	185
	X	328.3	330.9	333.5	• • • • •	321.0	323.4
	Y	531.0	530.6	530.3	• • • • •	532.0	531.6

耳朵	次序 座標	1	2	3	• • • • •	96	97
	X	399.3	399.7	400.3	• • • • •	399.4	399.5
	Y	490.5	492.0	494.3	• • • • •	428.3	430.0

表 3-1

經演算壓縮後之數據（控制點）

頭	次序 座標	1	2	3	• • • • •	49	50
	X	480.5	486.9	550.8	• • • • •	486.9	550.8
	Y	647.8	644.9	579.7	• • • • •	644.9	579.7

鏡片	次序 座標	1	2	3	• • • • •	14	15
	X	284.9	282.4	286.4	• • • • •	282.4	286.4
	Y	555.9	561.6	542.9	• • • • •	561.6	542.9

鏡柄	次序 座標	1	2	3	• • • • •	33	34
	X	323.6	328.3	374.6	• • • • •	328.3	374.6
	Y	531.6	531.0	525.6	• • • • •	531.0	525.6

耳朵	次序 座標	1	2	3	• • • • •	13	14
	X	392.1	399.3	406.6	• • • • •	399.5	393.6
	Y	469.7	490.5	511.3	• • • • •	430.0	452.3

表 3-2

原 始 輪 入 圖 形

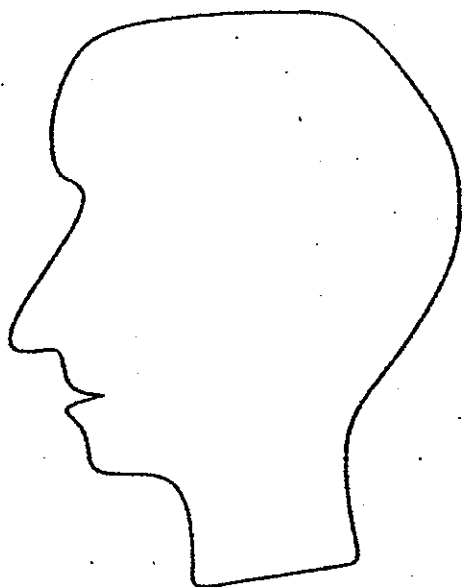


圖 3-13



圖 3-14

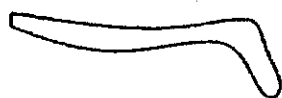


圖 3-15



圖 3-16

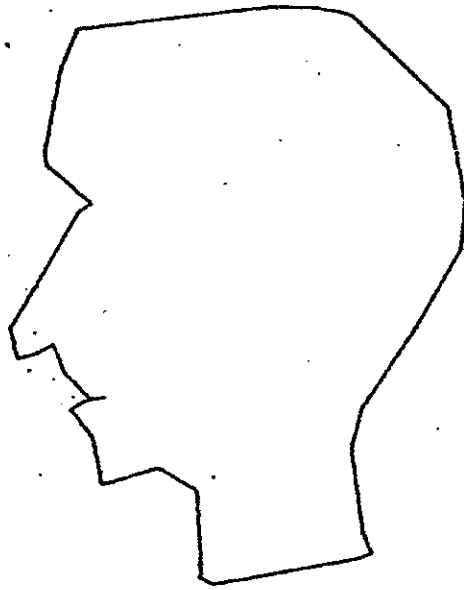


圖 3-17



圖 3-18

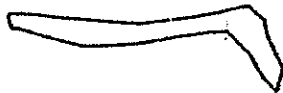


圖 3-19



圖 3-20

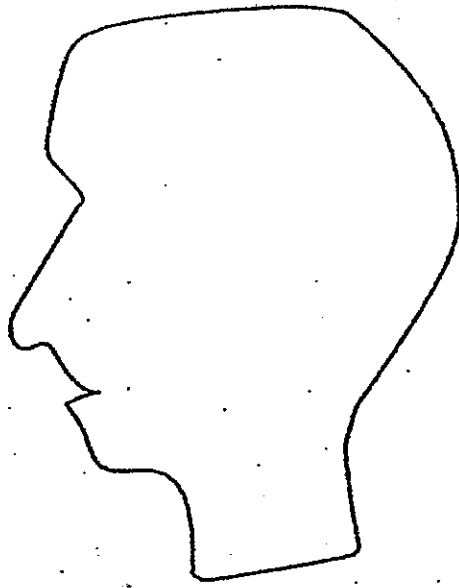


圖 3-21

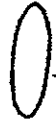


圖 3-22

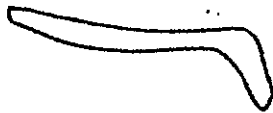


圖 3-23



圖 3-24

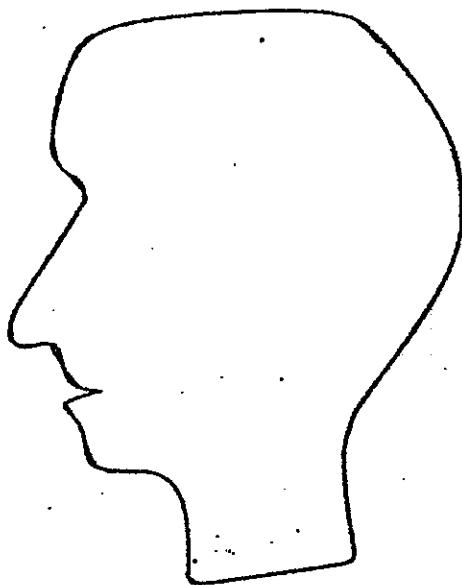


圖 3-25



圖 3-26

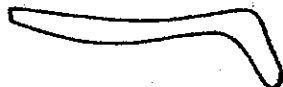


圖 3-27



圖 3-28

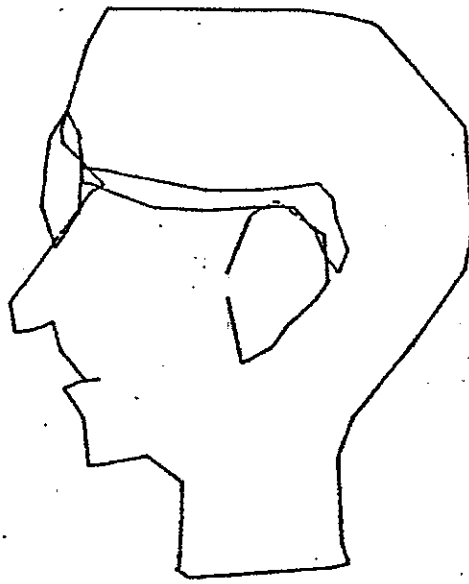


圖 3-29

輸出圖形之組合

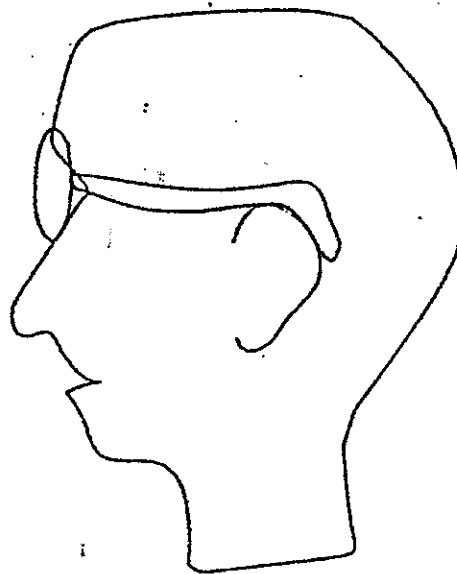


圖 3-30

4.1 電腦動畫的定義

爲了便於說明動畫產生的過程，讓我們參看圖一（腳步的移動），很明顯地，動畫可以分解成許多細部的動作，每個細部的動作構成單方向的變化，譬如圖一的動畫可細分成①至②及②至③的單向圖形變化，換言之，如果將圖上一點的時間變化繪出曲線，則單向漸增（漸減）的部份就構成該動畫的一個細部動作，爲簡便計，我們稱一個細部動作爲“單動作”依照電腦的術語來講，一個動畫就是一個“程序”（PROCEDURE），其中的任一單動作就是該程序裏的一個 Begin... End 的 Block，若一動畫有 n 個單動作，則對應此動畫的程序有 n 個 Begin...End 的 Nested Blocks，其中

$$\text{End}(i \text{ the Nested Block}) = \text{Begin}(i+1 \text{ th Nested Block}), i > n$$

所以電腦動畫可以經由 Structured Programming 來建造，只要它的每個單動作可以經由“動畫演算法”產生，換言之，只要我們給每個單動作的第一張（Begin）圖畫及最後一張（End）圖畫，若“動畫演算法”可以自動產生中間過程的圖畫，那麼動畫就可以用 Structured Program 產生。

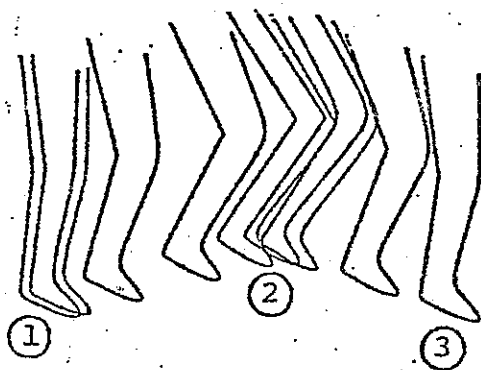


圖 4-1

問題是：動畫演算法所產生之中間過程的圖畫，它們所構成圖形變化的逼真性滿意嗎？令一個單動作經某一個動畫演算法處理後所產生的圖畫爲 $\{S_k\}$, $k=1, \dots, \ell$, 其中 S_1 和 S_ℓ 爲已知的第一張和最後一張的圖畫，這些圖畫所構成的動畫用 $\int_{\alpha} S_k$ 表示，用 $\xi \in [0, 1]$ 表示逼真性的滿意程度， $\xi = 1$ 表示絕對滿意， $\xi = 0$ 表示絕對不滿意， $0 < \xi < 1$ 表示具有程度上的滿意，若演算法是可調

的（經人機交談的作業方式），亦即單動作的圖畫有很多的方式產生，故對一組圖畫言，它們是經某個演算法（或方式）所產生，則我們可以說某些動畫演算法所產生之連續畫面所構成逼真性是可接受的，只要它們是屬於下面的集合

$$A = \{ \alpha_i : \xi_i / \int_{\alpha_i} S_k \mid \xi_i \geq \sigma, 0 < \sigma \leq 1 \}$$

其中 σ 是設定的 Threshold Value，或者是指藝術家的認可值，他不滿意時，即指該 $\int_{\alpha_i} S_k$ 不合乎他的要求，此時 $\sigma > \xi_i$ 。

從以上的討論，吾人可以正式給電腦動畫下個定義：令動畫演算法是一個 Mapping

$$\alpha : S_i \longrightarrow S_f$$

它將單動作的第一張圖片 S_i 投影至第後一張圖片 S_f （產生中間的圖片 $\{S_k\}, k=1, \dots, f$ ），而且

$$\alpha \in A$$

電腦動畫的問題即是要找到這樣的一個 α 。

對於某些單動作，動畫演算法 α 是較容易求得的，譬如動畫是經由幾何變換（旋轉、放大縮小、平移）所產生者：飛機（或無敵鐵金剛）的飛行，錶針的轉動，……等，因為它們有一定規律（或數學運轉）可遵循，其他屬於同樣的範疇者是工程上的動畫，例如電荷在已知磁力場（或電力場）運動所產生的磁力場（或電力場）變化，是經由解微分方程式獲得。在這些應用上，通常可以找到 α ，其 $\xi=1$ 。比較困難的動畫是屬於具有“表情”的連環畫（或卡通），因為它幾乎沒有規律可遵行，完全是一種欣賞力的昇華，可找到的 α 在大部份的情況下，其 $\xi < 1$ ，本文的重點則是針對這種具有表情之動態畫做一番探討。

4.2 B 軟楔曲線間之朦朧相似性

令 S^i 表示對應素描多邊形 C^i 的 B 軟楔曲線， S^f 為對應 C^f 的 B 軟楔曲線，由電腦動畫的定義知，從靜態畫 S^i 變化至 S^f 的單動作是一個映像 $M(t) : S^i \rightarrow S^f, t \in [t_i, t_f]$ ，其中 $M(t_i)S^i \equiv S^i, M(t_f)S^i \equiv S^f$ ，則任何二個相鄰的 $M(t)S^i$ 若“非常”相似，那怎我們可以保證此單動作是“相當”平穩的變化（參閱圖 4-2）。在這裏我們用了朦朧的描述句，二個圖形非常相似是什麼意思？電腦怎樣瞭解它

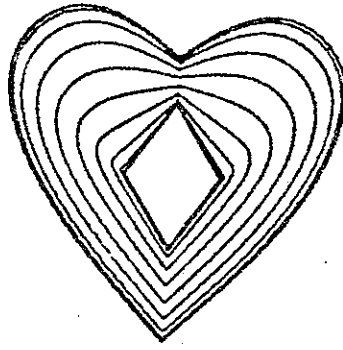


圖 4-2

令 U 表示一個單動作裏所有可能產生之 B 軟楔曲線的集合，則一個 U 的「模糊子集合」(Fuzzy Subset)，是由一個「會員函數」(Membership Function) $\mu_Z : U \rightarrow [0,1]$ 來訂其特性，亦即對於 U 的每個元素 ζ (任一可能產生的 B 軟楔曲線) 有一個數值 $\mu_Z(\zeta) \in [0,1]$ 來表示 ζ 在 Z 的會員等級。我們說 $S_U(\zeta)$ 是模糊子集合 Z 的「支持」(Support)，如果它可用下式表示

$$S_U(\zeta) = \{\zeta | \zeta \in U \text{ 以及 } \mu_Z(\zeta) > 0\}$$

上式意味著模糊子集合 Z 可表示為

$$Z = \{ \mu_Z(\zeta)/\zeta \mid \zeta \in S_U(\zeta) \} \quad (4.1)$$

對於任二個在 U 的 B 軟楔曲線 ζ 和 η 間之「模糊關係」(Fuzzy Relation) R 亦是用一會員函數 $\mu_R(\zeta, \eta)$ 訂其特性，所以它是 Cartesian 乘積 $U \times U$ 的模糊子集合，可表示如下

$$R = \{ \mu_R(\zeta, \eta)/(\zeta, \eta) \mid \zeta, \eta \in U \} \quad (4.2)$$

以下我們討論一種在特殊狀況下建造之 B 軟楔曲線群的模糊關係，這項觀察對於發展一套動態畫演算法有甚大的助益。令 $C = \{v_0, v_1, \dots, v_n\}$ 其中 $v_0 = v_n$ ，爲了方便用數學式表示，假設我們採用 Beirstein 基底函數 (亦即 B 軟楔曲線是由一條 Bézier 高次多項式所形成)，則對應 C 的曲線是由下式決定：

$$\sum_{k=0}^n v_i \binom{n}{k} t^k (1-t)^{n-k}, \quad t \in [0,1]$$

從集合論知 $\bar{C} \subseteq C$ 可表示為

$$\bar{C} = \{v_i \mid n_i\}, \quad i = 0, 1, \dots, n$$

$n_i=1$ 表示 v_i 在 $\bar{C}$ 內, $n_i=0$ 表示 v_i 不在 $\bar{C}$ 裏, 假設素描多邊形 $\hat{C}$ 是比 $\bar{C}$ 多取一個 C 的素描點所構成, 則對於每個這樣的 $\hat{C}$ 吾人可用 $\bar{C}$ 置換 $\hat{C}$ 繼續建造新的 $\hat{C}$, 直至 $\hat{C} \equiv \bar{C}$, 換言之, 從一個 $\bar{C}$ 開始到 C 間有許多的 $\bar{C}$ 和 $\hat{C}$, 我們想探討的問題是: 它們和對應 C 之曲線形狀間的朦朧關係是怎樣呢? 以及它們彼此之間的朦朧關係又是怎樣呢?

從朦朧集合論的觀點看, 吾人可將 C 所決定的曲線形狀看成 "標準形狀" (Standard Shape), 則每個 $\bar{C}$ 所決定的曲線形狀是分別由其會員函數 μ_i , $\mu_i \in [0,1]$, 訂定其特性, 而 $\mu=1$ 是對應 C 所決定之曲線形狀的會員函數, 則令

$$\mu_k = \max_t \binom{n}{k} t^k (1-t)^{n-k}, \quad t \in [0,1]$$

因為當 $k=n$ 時, $\mu_n=1$, 而對於 $k < n$ 者, 則 $0 < \mu_k < 1$, 所以我們可用下式表示朦朧子集合 "標準形狀"

$$\text{"Standard Shape"} = \{\mu_k / \bar{C} \mid \mu_k=1 \text{ if } \bar{C} \equiv C \text{ and}$$

$$\mu_k = \max_t \left\{ \binom{n}{k} t^k (1-t)^{n-k} \text{ if } \bar{C} \subset C \right\} \quad (4.3)$$

其中 k 表示 $\bar{C}$ 總共的素描點 (亦即 $|\bar{C}|$)。換言之, 當 C 的所有素描點皆被取時, 則它所決定的曲線形狀 "一定" 是最標準的, 若僅有的部份素描點被取時, 則它們所決定的曲線形狀到底 "標準" 到何種程度是由其會員函數所決定, 藉著這樣的表示法, 吾人可以將一個抽象的形狀用數值表示, 而讓電腦可以處理。

令 $\bar{C}$ 和 $\hat{C}$ 所決定之曲線形狀的標準性是分別由它們的會員函數 $\bar{\mu}$ 和 $\hat{\mu}$ 訂定, 則二條曲線形狀的 "相似" (Similar to) 是一種

朦朧關係，這種相似的朦朧關係可由一個會員函數 $\mu(C, \hat{C})$ 訂定其特性，從直覺的觀點看，它可以表示如下

$$\text{"Similar to"} = \{ \mu(\bar{C}, \hat{C}) / (\bar{C}, \hat{C}) \mid \mu(\bar{C}, \hat{C}) = \min(\frac{\bar{u}}{\bar{u}}, \frac{\hat{u}}{\hat{u}}) \} \quad (4.4)$$

由於增加素描點會使 $\bar{u}$ 和 $\hat{u}$ 值的差距拉大，而使得二者間所決定的曲線形狀愈不像，這種相似程度的降低可從圖 4.3 看出。從此例中可以發現一項重要的現象：相同形狀的二個素描多邊形，若它們的素描點數不一致，則會產生二個形狀不像的 B 軟楔曲線，反之，二個形狀不一樣的素描多邊形會不會產生相同形狀的 B 軟楔曲線呢？答案是肯定的，因為從近似理論，我們知道在某種誤差的限度下，一條曲線的形狀可用其上不同組合的點群表示，二個不同組合的點群，經過均勻 B 軟楔反函數的過程，就得二個不同形狀的素描多邊形，而它們所決定的 B 軟楔曲線形狀却是一致的，這種情形可從圖 4.4 看出。

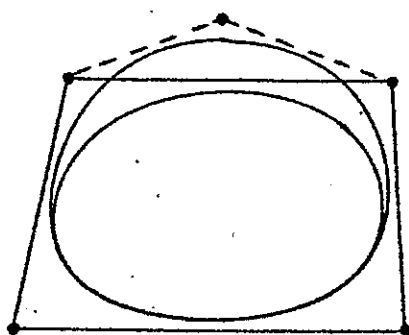


圖 4-3

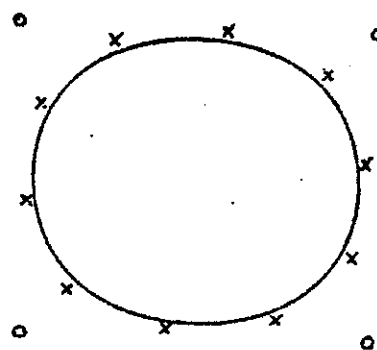


圖 4-4

4.3 動態畫產生法

令 $C(s^i)$ 和 $C(s^f)$ 分別表示單動作的第一張和最後一張圖形 s^i 和 s^f 的素描多邊形，則動態畫產生的演算法乃欲在 $C(s^i)$ 和 $C(s^f)$ 間產生素描多邊形移動的軌跡，通常二個素描多邊形的素描點數並不一樣多，亦即 $|C(s^i)| \neq |C(s^f)|$ ，因此，爲了決定它們之間的“移動軌跡”(Moving Paths)，第一步的工作就是要轉換單動作的首尾二個素描多邊形，使它們具有相同的素描點數。爲方便討論計，令 $|C(s^i)| < |C(s^f)|$ ，則從上節的討論知，吾人不能直接在素描多邊形 $C(s^i)$ 的邊上增加素描點，因爲那樣會使 s^i 完全變形，唯一的方法是透過均勻 B 軟楔反函數的過程來決定一個新的素描

令 $|\tilde{C}| = \max(|C(S^i)|, |C(S^f)|)$ ，則均勻 B 軟楔反函數的過程包括二個步驟（以下我們僅討論封閉 B 軟楔曲線的情形）：(一) 在曲線 S^i 上尋找數據點群，它們將 S^i 均分成 $|\tilde{C}|$ 段，亦即找出 $S^i(t_k)$ ， $t_k=0, 1/|\tilde{C}|, 2/|\tilde{C}|, \dots, 1-1/|\tilde{C}|$ ，假設這些數據點分別是 $P_0(0)$ ， $P_1(0), \dots, P_{q-1}(0)$ ，其中 $q = |\tilde{C}|$ ，而 $P_0(0)=P_{q-1}(1)$ ，則從 (3.8) 式知，這些點滿足以下的線性系統方程式

(二) 從(4.5)式解得新的素描點群 $\{v_i\}$, $i = 0, 1, \dots, q-1$, 令此素描多邊形為 $C(S^i)$, 則在直觀上 $\tilde{S}^i$ 的曲線形狀和 S^i 是 "非常" 相似, 因為它是從近似法 [9] (逼近 S^i) 中求得的。

以下讓我們假設 $c(s^i)$ 和 $c(s^f)$ 已經具有相同數目的素描點，則動畫演算法的第二件工作就是要將二者的素描點互相匹配，以產生素描點移動的軌跡，由於我們所要處理的是多邊形的端點，所以這件工作並不困難。首先，我們將二個多邊形的重心分別求出，將面積較小者套入面積較大者，並使二者的重心在同一點，為方便討論，我們假設 $c(s^i)$ 的面積大於 $c(s^f)$ ，因此， $c(s^f)$ 被套入 $c(s^i)$ 內，我們用 $c(\tilde{s}^f)$ 表示平移 $c(s^f)$ 後的素描多邊形，於是 $c(s^i)$ 和 $c(s^f)$ 素描點的配對是藉著匹配 $c(s^i)$ 和 $c(\tilde{s}^f)$ 的素描點來完成（注意： $c(s^i)$ 和 $c(\tilde{s}^f)$ 不一定要相互包含）。

-41-

則媒公 $\bar{v}_l$ 滿足

$$T(\bar{v}_k, \bar{v}_l) = \min_{0 \leq k \leq m} \{ T(v_h, \bar{v}_k) \} \quad (4.6)$$

其中 T 是一種測度值，我們所採用的測度法是這樣的：先在 $\{\bar{v}_k\}$ 中找距離較靠近 v_h 者（譬如說取一半 $C(\bar{s}^f)$ 的素描點），再用 $\{\bar{v}_h\}$ 的平分角線和這些素描點的平分角線分別求得的夾角，當做測度值 $T(v_h, \bar{v}_k)$ 。

一旦媒婆和媒公決定後，則可按順序成批匹配，依順時針的方向配對成 $(v_{h+1}, \bar{v}_{l+1}), (v_{h+2}, \bar{v}_{l+2}), \dots, (v_{h-1}, \bar{v}_{l-1})$ ，就完成了匹配的工作，圖 4.5 示出紅心和鑽石的匹配。

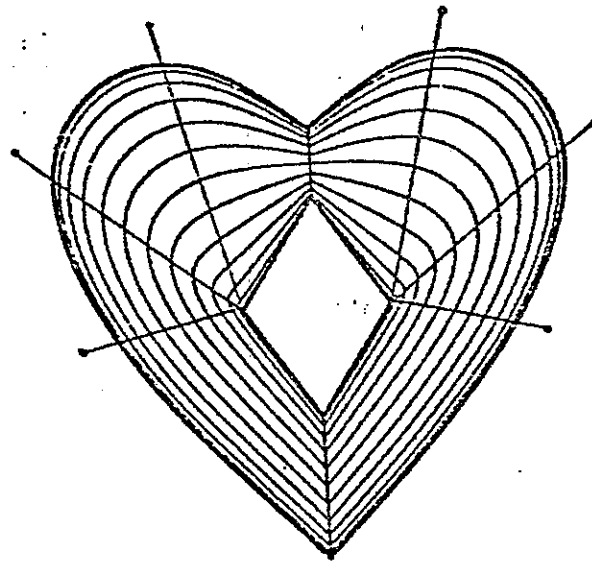


圖 4-5

素描點匹配完成後，動畫演算法的第三項工作就是要決定它們彼此之間的行進軌跡，為方便說明計，讓我們參看圖 4.6。令 $O(s^i)$ 和 $O(s^f)$ 分別表示多邊形 $C(s^i)$ 和 $C(s^f)$ 的重心，則行進軌跡是從二個多邊形對它們的重心“稍微”(A Bit)收縮或放大來完成，這樣就決定了行進軌跡首尾點的位置和其上的斜率，有了這些條件就可決定一條三次的曲線，但是為了將來可容許朦朧指令調整行進軌跡的形狀，我們採用了如圖上所示的均勻軟楔曲線，每條這樣的曲線是由七個控制點定其形狀，這七點的選法是：第 2 點（或第 4 點）是稍微收縮（或放大）第 1 點（或第 5 點）所決定，而第 0 點（或第 6 點）是取曲線起點（或終點），是第 1 點（或第 5 點）的 B 軟楔控制點，它的長度和稍微收縮（或放大）等長，第 3 點是用來控制行進軌跡的形狀，它的選取需經

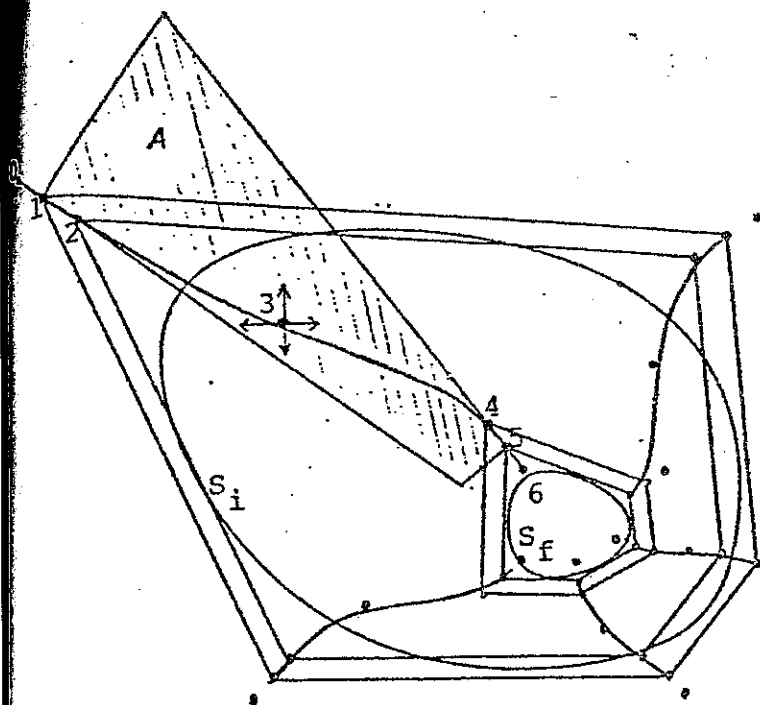


圖 4-6

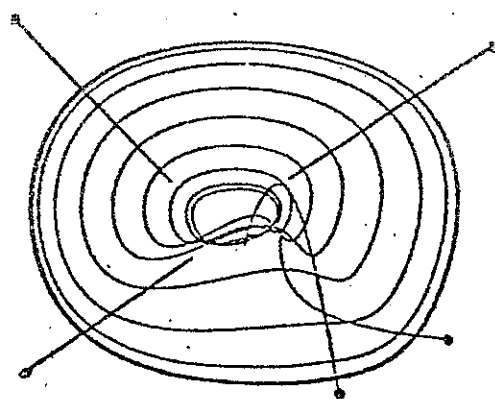


圖 4-7

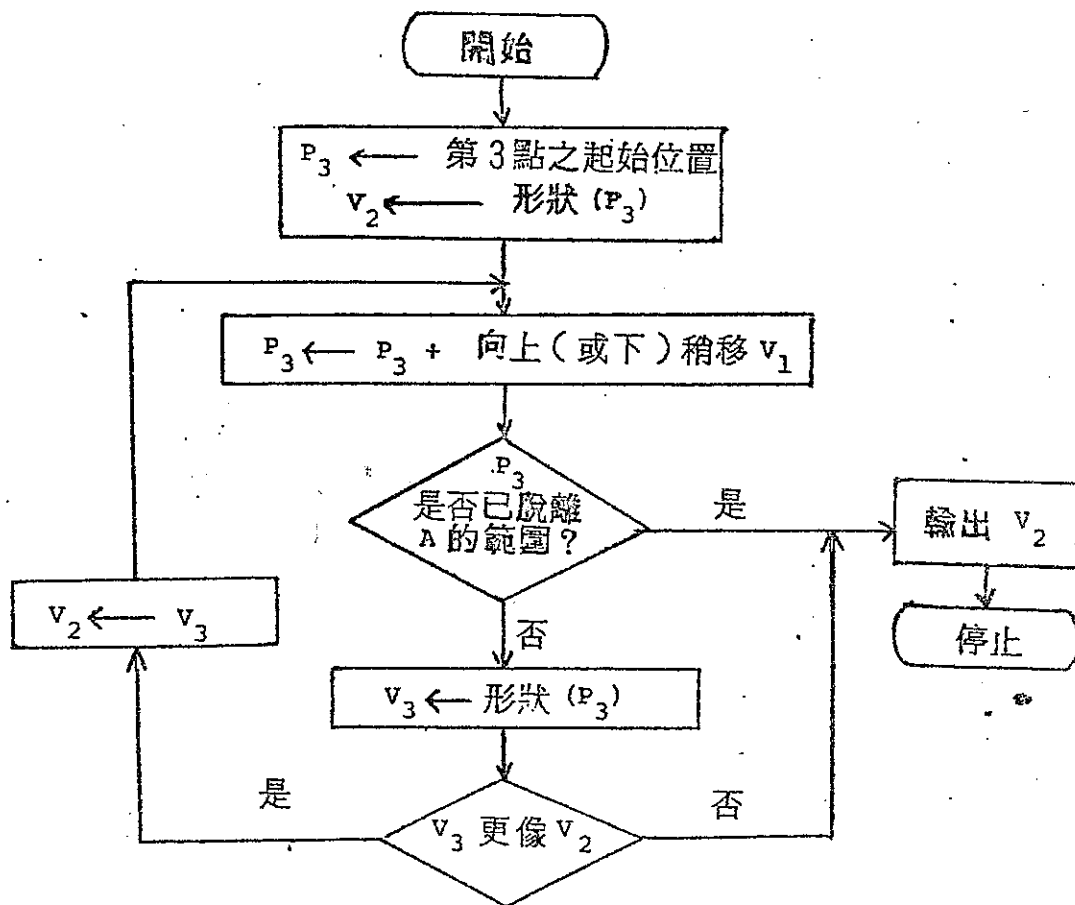


圖 4-8

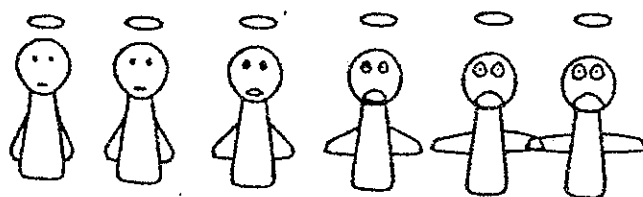


圖 4-9

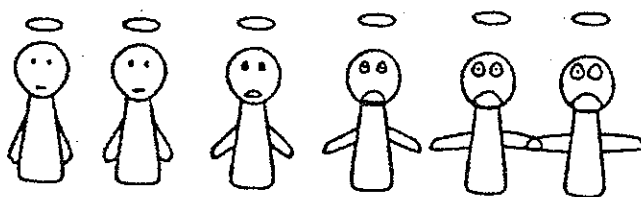


圖 4-10



圖 4-11

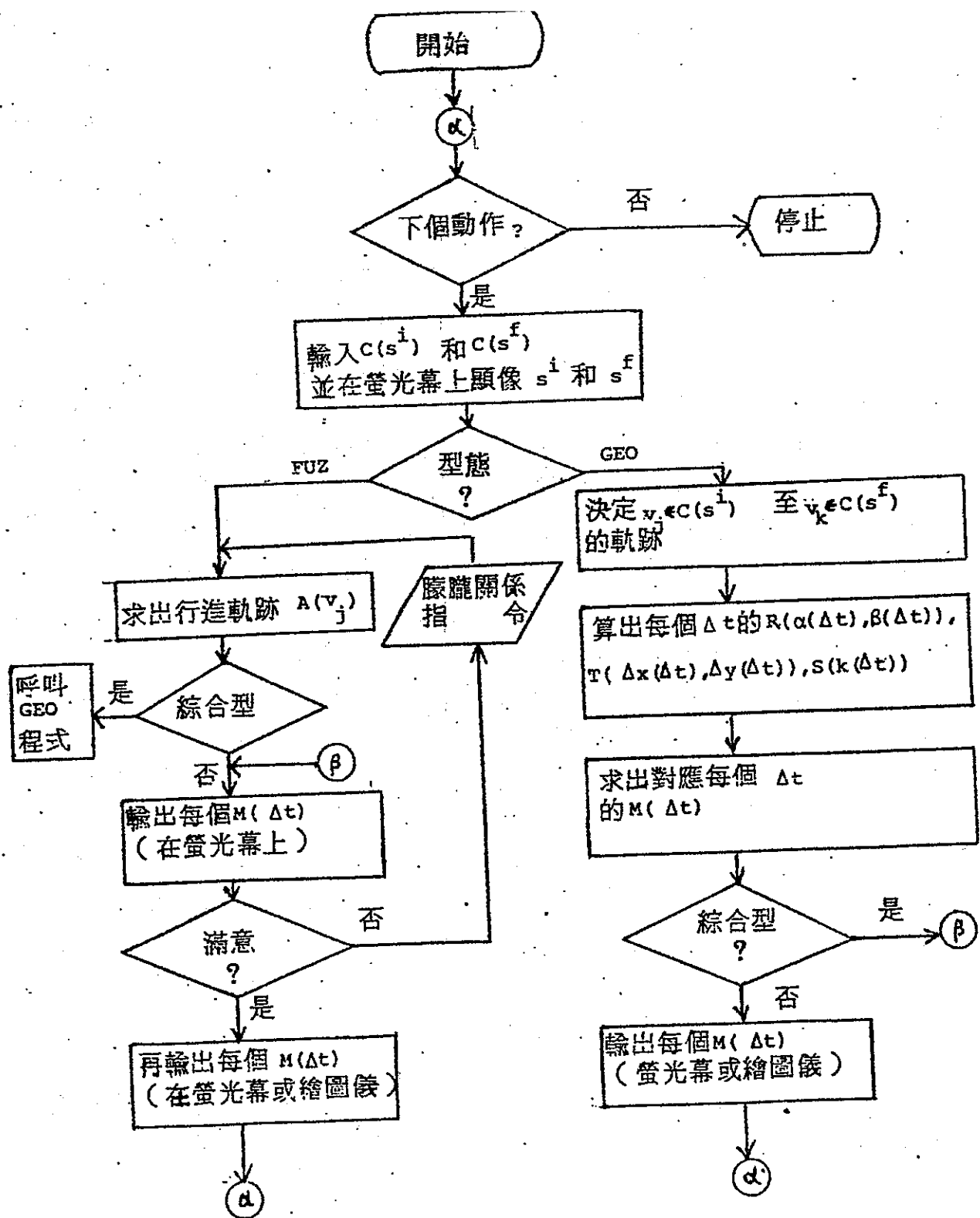


圖 4-12

過特別的考慮。

爲了避免相鄰兩個B軟楔曲線疊交在一起而產生不自然(扭曲)的動畫(如圖4-7)，第3點的選取必需限制在某一個範圍，從B軟楔曲線的特性知，這個範圍就如圖4-6上所示的A，又從觀察知，最不可能產生動畫扭曲的選擇是取三角形 $\overline{123}$ 和 $\overline{345}$ 的面積和爲極小者，所以我們就取具有此項性質的第3點當做起始的選擇，若有必要，再做朦朧程式的微調，因此動畫演算法的第四項工作就是要提供怎樣調整第3點位置的朦朧程式。

從朦朧邏輯論的觀點言，朦朧指令是一種「騎牆語言」(Linguistic Hedge)，在此處我們所用的朦朧指令不外乎：向東(東南、南、西南、西、西北、北、東北)稍移，更進一層的騎牆語言是：大概(或許)(有點)向東等，所以這種朦朧程式的設計並不難。從圖4-4知第3點所存在的範圍是A，「稍移」(Move Bit)第3點是A的一個朦朧子集合，它的會員函數可以用一種類似「費氏搜尋」(Fibonacci Searching)來訂定其每次跨步的長度，圖4-8示出我們所採用的「朦朧程式」(Fuzzy Program)，有關朦朧程式的定義請參閱[11]，此處不再重述。圖4-9示出未經朦朧指令調整的一張動畫，圖4-10示出它經朦朧程式處理後的結果(僅修飾一處)。

以上我們所論及的動畫產生法僅是最基本的部份，事實上，爲了使動畫更爲逼真，動畫系統必需提供更多可供朦朧指令調整的程式，例如提供控制素描點行進速率變化的朦朧指令，就可使動畫產生高階表情顯像的形態，所以在設計交談式的程式時，就可納入類似如下的指令：

SPEED(Located by Cursor) = Bit Fast, ..., etc.

還有一點值得一提的是，上述的動畫產生法並不能產生經幾何變換的動畫，換言之，如圖4-11的動畫必需另外處理，所以動畫系統的主程式應具有如圖4-12所示的結果。

本節將介紹一種我們發展成功之一套程序化之電腦卡通製作系統稱為 POCA (Procedure-Oriented Computer-Assisted Animation) 系統的電腦輔助動畫設計系統，而說明它的命令語言（稱為POCAL），乃供執行圖形的處理，以達成電腦輔助設計動畫在製作靜畫、動畫、著色等工作。

5.1 POCA軟體系統結構

POCA軟體系統結構如圖 5-1 所示，由於圖形處理工作複雜，程式中所需預留的暫存空間極大且在 PDP11 系列機器的操作系統 RSX-11M 下，限制一個工作程式不得超過 64K 數元組 (byte)，以致整個系統不能寫成一個單獨的工作程式 (task)，解決的方法有二，(一)利用 RSX-11M 系統提供的重疊 (Overlay) 技巧，把工作程式建成樹狀，使得執行時在某一瞬間不超過 64K 數元組的目標程式存於主記憶體中。(二)使用多工作程式 (Multitasking) 的技巧，把整個系統寫成數個工作程式，工作程式間靠共用旗幟及通訊緩衝區來完成同步及資料傳送的工作。考慮執行速度的快慢，系統的擴充及維護方便起見，我們採用了多工程程式的拔巧。

整個系統分為 6 個工作程式

工作程式 1 : MIAN

工作程式 2 : DMTSK

工作程式 3 : BSPLNE

工作程式 4 : IMAGE

工作程式 5 : FILL

工作程式 6 : SPL

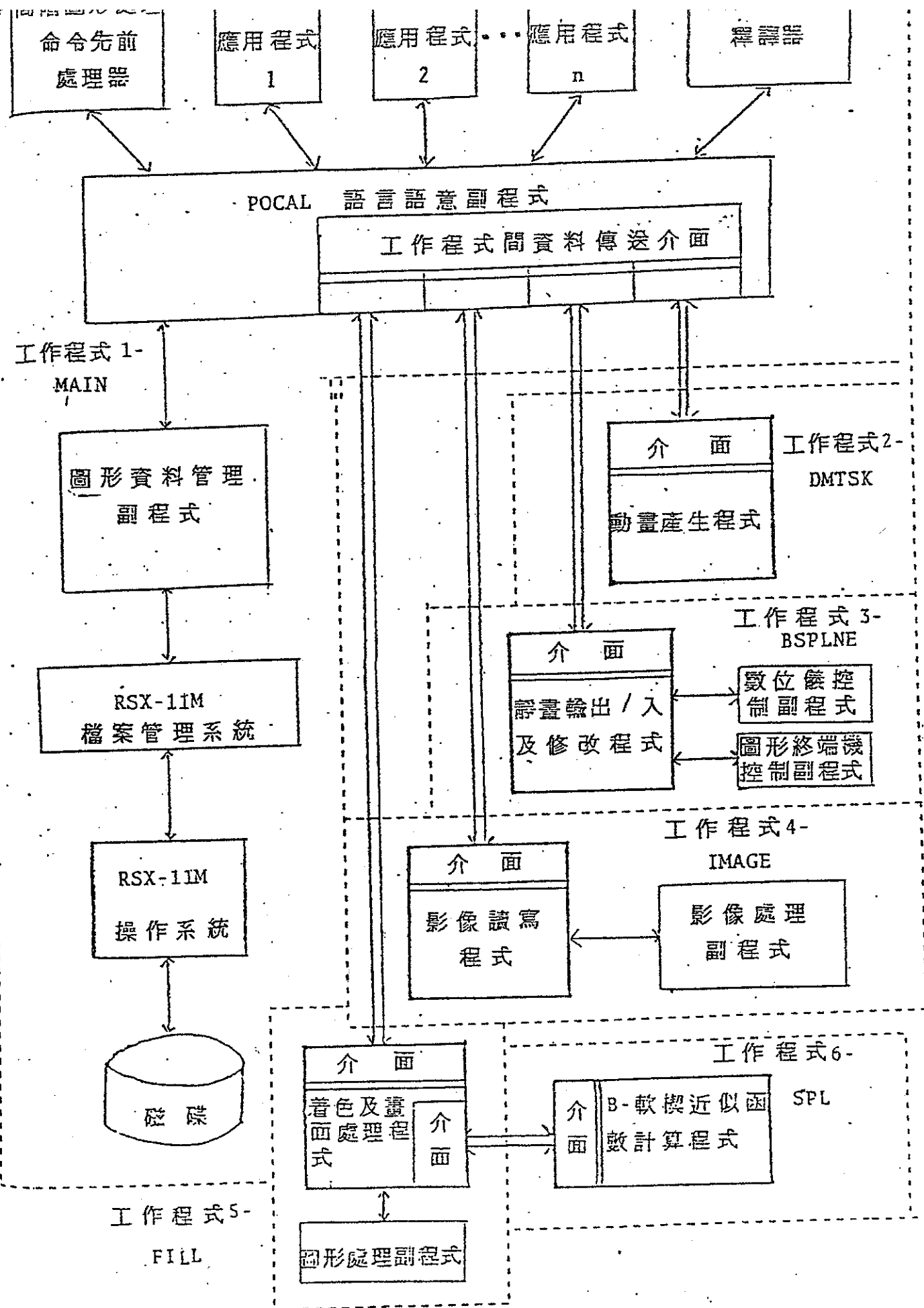


圖 5-1 POCA 軟體系統結構

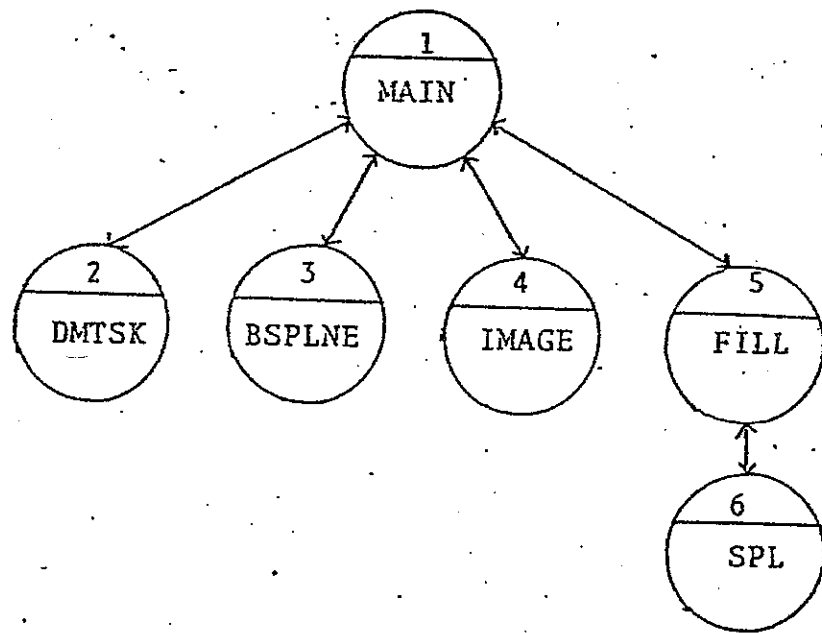


圖 5-2 POCA 系統中六個工作程式之主僕關係

主工作程式 (Master Tasks) MAIN 有四個僕工作程式 (Slave Task) DMTSK BSPLNE IMAGE 及 FILL，此四個僕工作程式之間沒有任何關係，SPL 為 FILL 的僕工作程式其功用是幫助 FILL 工作程式做 B 軟楔近似函數的計算，其餘四個僕工作程式，分別執行一個或數個 POCAL 語言的語意功能，如表 5-1 所示

工作程式	語意副程式
MAIN	EXIT, HELP, CAMERA, CONCATE, ROTPT, ROTCG, MIRROR, ZOOMX, ZOOMY, ZOOMD, ZMCGX, ZMCGY, ZMCGD, COPY, CANCEL, SUBTRACT, TRARL, TRAAB.
DMTSK	DM
BSPLNE	MFIG, IN
IMAGE	IMR, IMW, IMWRL
FILL	RCS, DIS, DISCG, DRAW, COOR, ERASE, COLOR

表 5-1 語意副程式所在的工作程式

DMTSK 工作程式，主要為動畫產生程式，因演算法複雜及預留緩衝區極大，故成一工作程式。

BSPLNE 工作程式，包括靜畫輸出 / 入及修改程式，實質上它是一個素描多邊形控制點的編輯程式 (Editor)，因涉及從數位儀 (Digitizer) 輸入資料及從高析度顯示器 (TEKTRONIX 4054) 輸出 / 入資料及圖形，故亦包含數位儀控制副程式及圖形終端機控制副程式 (Terminal Control System) [12] 為 TEKTRONIX 軟體模組。

IMAGE 工作程式，主要工作為對 RAMTEK 顯示器的影像做讀 (存) 及寫 (重現) 的工作，亦包含影像處理副程式。

FILL 工作程式的工作包括著色、顯像及彩色顯示器的畫面處理亦包括圖形處理副程式。

上述影像處理副程式及圖形處理副程式，均為 RAMTEK 公司的一個稱為福傳影像反畫圖軟體模組 (FIPP-FORTRAN Image & Plot Package) [13] 中之一部份。

MAIN 工作程式，除了前四個工作程式的語意功能外，其他的 POCAL 指令語意功能均在此工作程式中執行，主要為圖形資料管理副程式。此管理副程式除擁有本身的圖形管理法則外，所有檔案資料均透過 RSX-11M 的檔案管理系統 (FCS-File Control System) 來貯存。

MAIN 工作程式的主程式可為：

- (1) 應用程式：因所有的 POCAL 指令均寫成福傳語言的副程式呼叫方式，即使用者可把這些副程式和福傳語言合用，寫成使用者自己的應用程式。
- (2) 高階圖形處理命令先前處理器：除可直接呼叫 POCAL 語言副程式外，專為使用者所寫的應用程式更具可讀性及更好的文件敘述 (Documentation) POCA 系統提供一更高級的高階圖形處理命令語言，可和福傳程式併用來寫應用程式，唯在執行前須先經一先前處理器 (Preprocessor) 翻譯成 POCAL 語意副程式，高階圖形命令語言語法見表5-2。
- (3) 互動式命令釋譯器：為使使用者直接以互動式方式使用 POCA 系統，亦有一互動式命令釋譯器直接接受交談式命令作輔助設計的工作，互動式命令見表5-3。

. INPUT <name>
 . READ-CURSOR (<var>,<var>)
 . MODIFY-PICTURE <name>
 . DISPLAY <name>
 . DISPLAY-CG <name>
 . DRAW <name>
 . COORDINATE
 . ERASE-SCREEN <color>
 . COPY <name> = <name>
 . CANCEL <name>
 . CONCATE <name> = <name> + <name> {+ <name> + ...}
 . SUBTRACT <name> = <name> - <name> {- name - ... }
 . DYNAMIC-MOTION <name>,<name> STEP $\begin{bmatrix} \langle \text{int} \rangle \\ \langle \text{var} \rangle \end{bmatrix}$ INTO <name>
 . COLOR <name>
 . $\begin{bmatrix} \text{TRASLATE-REL} \\ \text{TRASLATE-ABB} \end{bmatrix}$ <name> $\begin{bmatrix} (\langle \text{int} \rangle, \langle \text{int} \rangle) \\ (\langle \text{var} \rangle, \langle \text{var} \rangle) \end{bmatrix}$ INTO <name>
 . ROTATE-CG <name> <degree> {INTO <name>}
 . ROTATE-PT <name> TO $\begin{bmatrix} (\langle \text{int} \rangle, \langle \text{int} \rangle) \\ (\langle \text{var} \rangle, \langle \text{var} \rangle) \end{bmatrix}$ <degree> {INTO <name>}
 . MIRROR <name> $\begin{bmatrix} (\langle \text{int} \rangle, \langle \text{int} \rangle) \\ (\langle \text{var} \rangle, \langle \text{var} \rangle) \end{bmatrix}$, $\begin{bmatrix} (\langle \text{int} \rangle, \langle \text{int} \rangle) \\ (\langle \text{var} \rangle, \langle \text{var} \rangle) \end{bmatrix}$ {INTO <name>}
 . ZOOM-CG <name> BY <x-factor>
 . ZOOM-PT <name> TO $\begin{bmatrix} (\langle \text{int} \rangle, \langle \text{int} \rangle) \\ (\langle \text{var} \rangle, \langle \text{var} \rangle) \end{bmatrix}$ <x-factor>,<y-factor> {INTO<name>}
 . IMAGE-READ FROM $\begin{bmatrix} (\langle \text{int} \rangle, \langle \text{int} \rangle) \\ (\langle \text{var} \rangle, \langle \text{var} \rangle) \end{bmatrix}$, $\begin{bmatrix} (\langle \text{int} \rangle, \langle \text{int} \rangle) \\ (\langle \text{var} \rangle, \langle \text{var} \rangle) \end{bmatrix}$
 . IMAGE-WRITE

. IMAGE-WRITE-RL TO $\left[\begin{array}{l} (<\text{int}>, <\text{int}>) \\ (<\text{var}>, <\text{var}>) \end{array} \right]$

. CAMERA $\left[\begin{array}{l} <\text{int}> \\ <\text{var}> \end{array} \right]$ TIMES

其中 [] 表選擇其中之一，{ } 表可選或不選。

<name> 爲 6 個字母之圖形名稱，字母爲 {A-Z, 0-9} 之一。

<var> 爲變數名稱，依 FORTRAN 語言的定義。

<int> 爲整數。

<degree>, <color> 可爲 <var> 或 <int>

<x-factor>, <y-factor> 爲實數。

表 5-2 高階圖形命令語言語法

```

1111 DEVICE I (D)-DIGITIZER , (T)-TEKTRONIX , (R)-RANTEK
$1H      I INPUT PICTURE (D)
$1K5     I READ CURSOR LOCATION OF JOYSTICK BY ENTER SWITCH 0 (R)
$1F10    I MODIFY PICTURE (T)
$1F15    I DISPLAY PICTURE (R)
$1U3C0   I DISPLAY CENTER OF GRAVITY OF PICTURE (R)
$1R6W    I DRAW PICTURE (R)
$1C0R    I DISPLAY X,Y COORDINATE (R)
$1E8SE   I ERASE SCREEN (R)
$1C0PY   I COPY PICTURE C OBJ = SOURCE I
$1C0NCL  I CANCEL PICTURE
$1C0NCTE I CONCATENATE TWO PICTURE INTO ONE C OBJ = BACKGROUND + FOREGROUND I
$1C0NTR  I SUBTRACT SEGMENT FROM PICTURE C OBJ = PIC - SEG I
$1DH     I DYNAMIC MOTION
$1C0LOR  I COLORING PICTURE (R)
$1E8RL   I TRANSLATE PICTURE RELATIVELY
$1E8AR   I TRANSLATE PICTURE ABSOLUTELY
$1ROTCH  I ROTATE ABOUT CENTER OF GRAVITY
$1ROTPT  I ROTATE ABOUT ONE POINT
$1MIR   I MIRROR PICTURE , MIRROR LINE IS SPECIFIED BY TWO POINTS
$1Z00HX  I ZOOM PICTURE ALONG X-AXIS
$1Z00HY  I ZOOM PICTURE ALONG Y-AXIS
$1Z00HD  I ZOOM PICTURE ALONG (X,Y)-AXIS
$1Z00GX  I SAME AS Z00HX
$1Z00GY  I SAME AS Z00HY
$1Z00GD  I SAME AS Z00HD
$1EXIT   I EXIT
$1HELP   I PRINT THIS DOCUMENT
$1MR     I IMAGE READ (R)
$1MR     I IMAGE WRITE (R)
$1MR     I IMAGE WRITE BY RELOCATION,
$1CAMER  I CAMERA

```

表5-3 交互式命令

5.2 圖形資料管理

一個卡通影片，我們可以把它分成五個層次

- 1 影片
- 2 畫面
- 3 動作
- 4 基本圖形
- 5 基本動作

他們的關係如圖 5-3

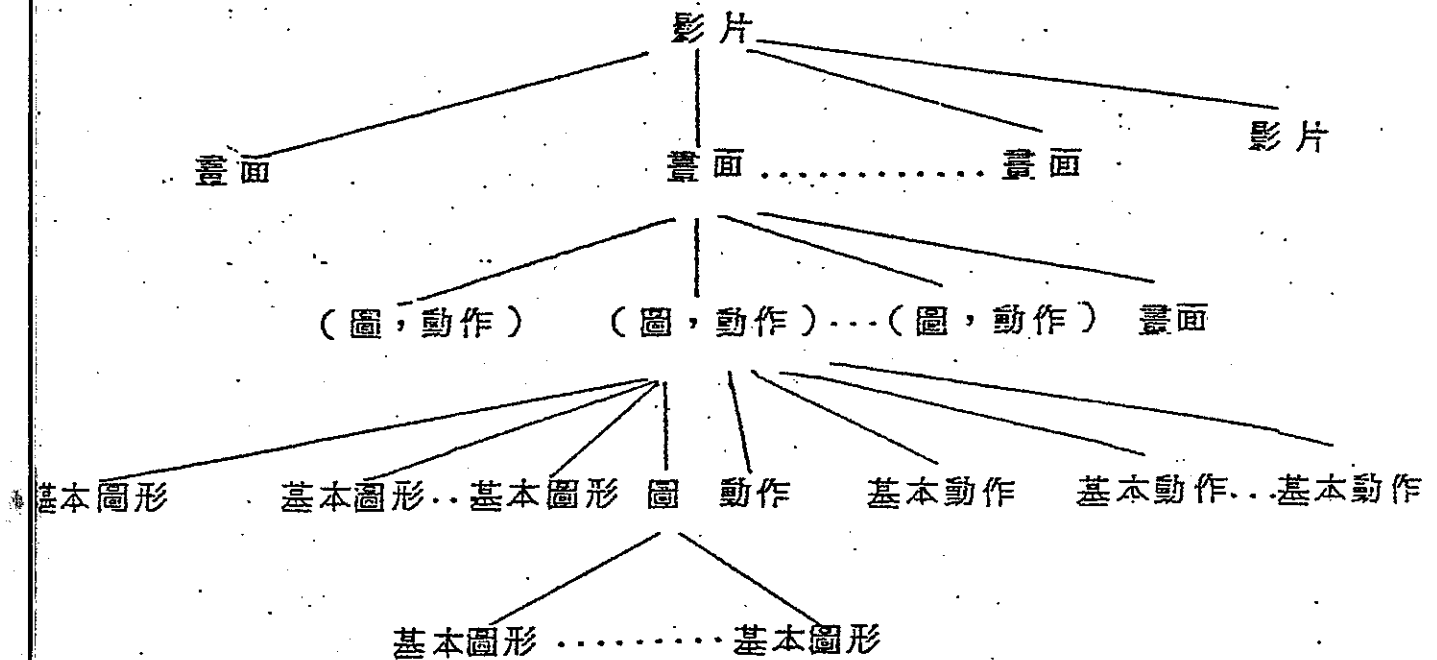


圖 5-3 影片的組成

動作只是一種意念，也就是圖形連續變化所給於人的一種感覺，真正的組成一段影片的就是圖，若把圖 5-3 動作去掉，得到圖形的關係如圖 5-4。

我們可以得到以下的關係式

影片 → 畫面 | 影片

畫面 → 畫 | 畫面

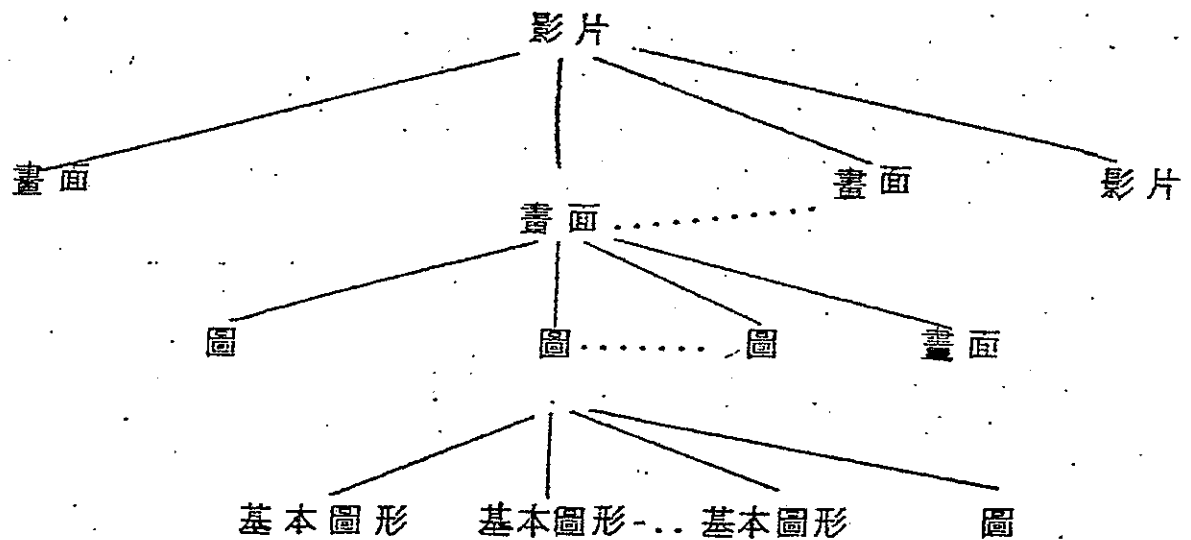


圖 5-4 影片和圖的關係

也就是說它們的關係均是遞迴的 (Recursive)，影片和畫面的差別在於是否以 24/18 張 / 秒連續放映。畫面和畫的差別在於畫面是由幾個畫組成，畫和基本圖形的差別，在於基本圖形是最基本的元素，不能再分開。即為輸入時的基本單位，基於這個因素我們可再把圖 5-4 化簡成圖 5-5 就得到

圖 → 圖 | 基本圖形

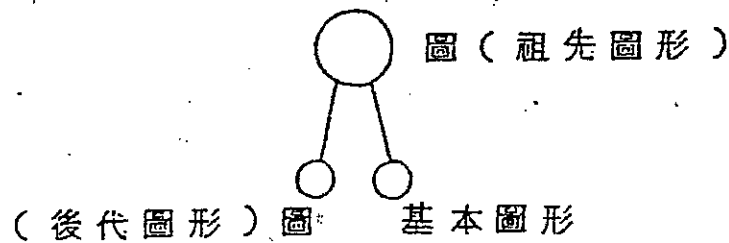


圖 5-5

在 POCA 系統中，基本圖形就是我們處理的最小單元，而任何圖可由任意基本圖形或任意圖組成，所以在系統中我們要記錄的就只有二件事了。(1) 基本圖形為何？(2) 圖是由那些圖或基本圖形組成的？同時我們根據樹狀定義，組成圖形的基本圖形，為所組成圖形的後代圖形，或組成的圖形為它們的祖先圖形。對於每一個圖形或基本圖形我們記錄的方式是這樣的：每一個基本圖形有

三個檔，(1)總檔(Head file)，(2)組成此圖的控制多邊形的所有控制點檔，(3)顏色種子檔。總檔中記錄①控制點檔檔案名稱，②所有後代圖形的名稱，③是否已着色，④着色角度，⑤種子顏色檔檔案名稱，⑥此圖形的祖先圖形數目。若以 P_i 表圖形， C_i 表控制點檔， S_i 表種子顏色檔，則一個基本圖形的結構如圖5-6所示。

任一非為基本圖形的圖可能為圖5-7所示(a)三個基本圖形 P_1, P_2, P_3 ，(b) P_4 是由 P_1, P_2, P_3 三個基本圖形組成，(c) P_4 為 P_1, P_2, P_3 的祖先圖形， P_1, P_2, P_3 為 P_4 的後代圖形。

系統中存放圖形資料的表格如下，見表5-4其中筆分類見(-)。

基本圖形名稱 (A6)	控制點檔名稱 (A6)	種子顏色檔名 (A6)	是否著色 (Y/N)	著色角度 (I3)	祖先圖形個 數 (I3)

總 檔

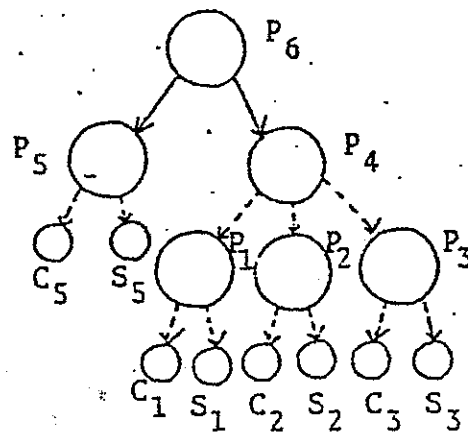
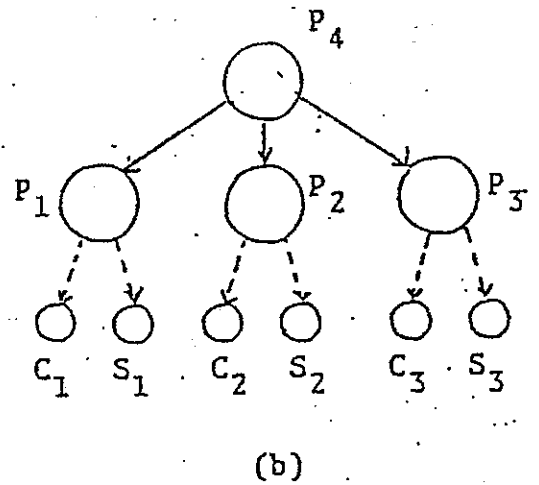
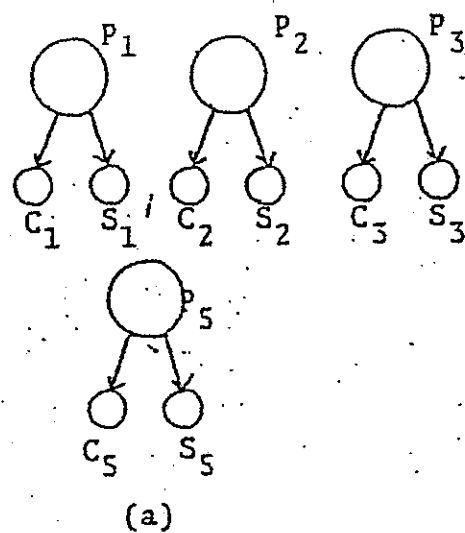
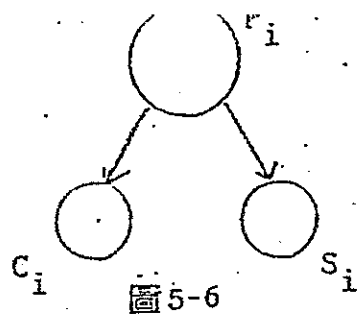
X_i 座標 (I3)	Y 座標 (I3)	筆分類 (I1)

控制點檔

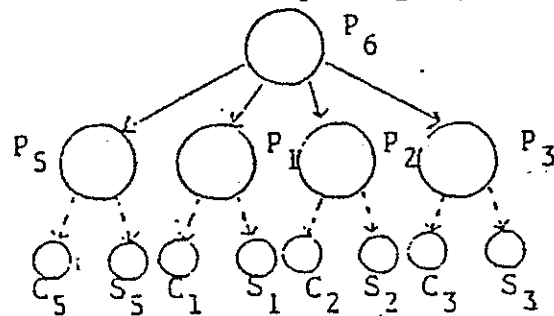
X 座標 (I3)	Y 座標 (I3)	色筆值 (I3)

種子顏色檔

表 5-4 圖形資料表



(c) p_6 由 p_1, p_4 組成



(d) 由 (c) 演變成, p_1, p_2, p_3, p_5 組成 p_6

圖 5-7

5.3 POCAL語言的描述

POCAL 語言語意副程式可分為 8 類分別如下：

- 1 圖形輸入副程式 : IN
- 2 輔助圖形修改副程式 : RCS, MFIG
- 3 圖形輸出副程式 : DIS, CAMER
- 4 輔助圖形輸出副程式 : DISPT, DISCG, DRAW, COOR, ERASE
- 5 圖形處理副程式 : COPY, CANCL, CNCTE, SUBTR, DM
- 6 圖形着色副程式 : COLOR
- 7 圖形幾何轉移副程式 : MIRR, ZOOMX, ZOOMY, ZOOMD, ZMCGX, ZMCGY, ZMCGD.
- 8 影像處理副程式 : IMR, IMW, IMWRL

各副程式所使用的係數 (Arguments) 規格 (Format) 如下：

- 1 PICNAM, SOUR, OBJ: (A6) 為任意 6 個字母的組合，表圖形名稱，字母為 (A-Z), (0-9) 其中之一。
- 2 IX, IY: (I3) 為 3 位數整數，表畫面上的 X, Y 座標。
- 3 FACT: F(3,1) 放大或縮小倍數，為一實數。
- 4 CLR: (I3) 3 位數整數，表色筆值。

以下按分類說明各指令

(一) 名稱：IN(Input Figure) 圖形輸入

功能：

從數位儀輸入一個基本圖形

呼叫格式：

CALL MIN(PICNAM)

說明：

圖形輸入屬互作式，當此副程式一被執行，系統即在終端機上顯示出

```
CREAT B-SPLINE CURVE SYSTEM
1. Input mode .....[I]
2. Add mode .....[A]
3. Delete mode .....[D]
4. Move mode .....[M]
5. Plot mode .....[P]
6. Exit mode .....[E]
```

即進入 B 軟楔函數控制點編輯副系統，此乃因圖形輸入後，可能還須加以修改共有 6 種方式供使用，即輸入(Input)，加點(Add)，除點>Delete)，移點(Move)，顯示(Plot)，及結束(Exit)分別說明於下：

1 輸入

當選擇此指令時系統顯示出

```
.Input control points from digitizer
Pen down .....[1]
Pen up & continue after open control point ....[2]
Pen up & continue after close control point ...[3]
Pen end after open control point .....[4]
Pen end after close control point .....[5]
Input coordinate of left corner & upper right corner
```

使用者，先由數位儀上定出圖形的窗函數 (Window)，即圖形的範圍，亦即圖形的最左下點及最右上點的座標，定好窗函數後系統印出

Input data begins

使用者即可開始以數位儀輸入圖形的素描多邊形之控制點，數位儀之十字游標鍵上有十六個輸入鍵，其中五個(1-5) 用來表示圖形輸入筆的提筆或下筆的狀況，使用者把十字游標的交叉點對準控制點，以下述原則按入 1,2,3,4,5 其中之一鍵：

(1) 按入 "1" 鍵表示筆放下到紙上劃入。

(2) 按入 "2" 鍵表示畫完此點後，欲把筆提起，畫另外一筆，同時亦表示此輸入的點為一筆劃的最後一控制點，此點不用來和首點作連接，即畫一不封閉的一筆劃之最後一控制點，且表示還欲畫下一筆。

(3) 按入 "3" 和按入 "2" 同，但筆畫是封閉的。

(4) 按入 "4" 和按入 "2" 同，但此筆畫完後，即結束圖形輸入。

(5) 按入 "5" 和按入 "3" 同，但此筆畫完後，即結束畫形輸入。

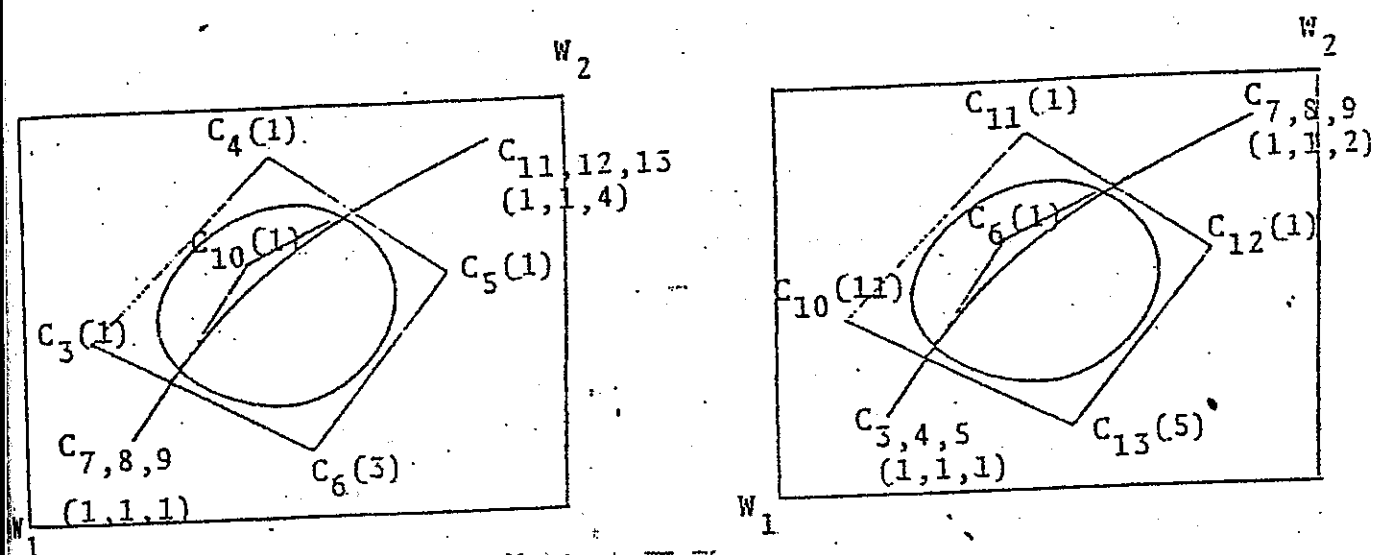
綜合上述五點，我們可以更簡捷的說，若為圖形輸入最後一筆的最後一個控制點按 "4" 或 "5" 此筆，若為封閉則按 "5" 否則按 "4"。

若非為圖形最後一筆，但却為一筆之最後一控制點，則按 "2" 或 "3"，若為封閉則按 "3"，若非為封閉按 "2"。其他點均按 "1" 如表 5-5、圖 5-8 為一輸入範例。

	非 封 閉	封 閉
非 最 後 一 筆	2	3
最 後 一 筆	4	5

其他：1

表 5-5 提筆 / 下筆的鍵



W 表窗函數

C 表控制點

() 中表鍵號碼

下標表輸入順序

圖 5-8 輸入範例

2 加點

系統先在高析度顯示器 TEKTRONIX 4054 上顯出圖形，及其控制點，再顯示一十字游標，並在終端機上顯出。

```
.Insert control points from 4054  
Move cursor to control point before pt
```

使用者，把十字游標指到所欲加入點之前一點，在鍵盤上打入一鍵（任一字母），系統再顯出。

```
Move cursor to insert point
```

使用者把十字游標移到欲加入點之位置，按入鍵，系統再顯出。

```
# of repeated points :
```

使用者，打入欲加入點的數目。

3 除點

系統先在高析度顯示器上顯出圖形及其控制點，並顯示一十字游標在終端機上顯出。

```
.Delete control points  
Move cursor to point to be deleted
```

使用者，以十字游標指出欲去除的點，系統再顯出。

```
There are      control points  
How many points to be deleted ?
```

系統計算出被指出的位置有幾個控制點，使用者打入欲去除的點數。

4 移點

系統在高析度顯示器上顯出圖形及控制點，並顯出一十字游標，在終端機上顯示。

```
.MOVE CONTROL POINT TO SOME WHERE  
Move cursor to control point to be moved
```

使用者以十字游標指出欲移動的點，系統再顯出。

```
Move cursor to desire control point
```

使用者以十字游標指出欲移往之位置。

5. 顯圖

在繪圖機上或高析度顯示器上，系統打出

```
.Plot picture  
Plot to Textronix 4054/Plotter [T/P] :
```

若為T，則在高析度顯示器上畫出圖，若為P則再顯出

```
SCALING FACTOR : ?
```

? 表放大倍數，為一實數。

6. 結束(Exit)

結束此副程式。

系統在每一種方式結束後，即回到程式起頭，直到此指令被執行才結束，且產生或修改圖形的總檔及控制點檔。

(二) 名稱：RCS (Read Cursor) ——讀取十字游標座標

功能：

經由如意桿的移動十字游標位置，讀取在畫面上的座標
呼叫格式：

```
CALL MRCS(IX,IY)
```

說明：

使用者在彩色顯示幕上，以如意桿移動十字游標至欲指出座標的位置，再按入如意桿控制器上的Enter 鍵，則X,Y 座標分別放置於IX及IY，此座標可供編輯畫面時使用。

(三) 名稱：MFIG (Modify Figure) ——修改圖形

功能：

修改已輸入之圖形。

呼叫格式：

```
CALL MFIG(PICNAM)
```

說明：

修改指令和MIN 副程式(見(一))相同，除不具指令功能外。

(四) 名稱：DISPT(Display Some Point)

功能：

在彩色顯示器上，顯示出一點。

呼叫格式：

```
CALL MDISPT(IX,IY)
```


說明：

IX, IY 爲欲顯示點的 X,Y 座標，因爲一點座標在顯示幕看起來太小，所以顯示出的點是以 IX, IY 爲中心的一個正菱形。

(d) 名稱：DISCG (Display Center of Gravity of a Picture) 顯示圖形的重心。

功能：

顯示一個圖形的重心。

呼叫格式：

CALL MDISCG(PICNAM)

說明：

在彩色顯示器上，顯出一個基本圖形的重心，以供編輯畫面使用。

(e) 名稱：DIS (Display a Picture) —— 顯圖。

功能：

顯示出圖形。

呼叫格式：

CALL MDIS(PICNAM)

說明：

在彩色顯示器上顯示出圖形，若圖形未着色，則只顯出輪廓，若圖形超出畫面範圍，則被切除。

(f) 名稱：DRAW (Draw a Picture) —— 描圖。

功能：

畫出圖形的輪廓。

呼叫格式：

CALL MDRAW(PICNAM)

說明：

在彩色顯示器上顯示圖形輪廓，若圖形超出畫面範圍則被切除。

(g) 名稱：COOR (Coordinate) —— 座標。

功能：

顯示出 X,Y 座標。

呼叫格式：

CALL MCOOR

說明：

在彩色顯示器上，每隔 10 格畫出 X,Y 座標，供編輯畫面使用。

(九) 名稱：ERASE(Erase Screen) —— 清除畫面。

功能：

清除畫面並改成指定顏色。

呼叫格式：

CALL MERASE(CLR)

說明：

改變彩色顯示器的畫面成某一指定的顏色。

(十) 名稱：COPY —— 拷貝。

功能：

拷貝一圖形成另一圖形。

呼叫格式：

CALL MCOPY(OBJ, SOUR)

說明：

OBJ 表原圖，SOUR 表新圖。

經拷貝後新圖自成另一個圖，自己擁有總檔、控制點檔及顏色種子檔，修改原圖或新圖均不影響對方。

(十一) 名稱：CANCEL(Cancel a Picture) 去除。

功能：

去除一張圖。

呼叫格式：

CALL MCANCL(PICNAM)

說明：

去除錯誤或不必要的圖，以節省貯存空間。

當一張圖被去除時，先去除此圖的總檔，且把祖先數目減一，若祖先數目為 0，則把關於此圖的控制點檔及顏色種子檔均去除，否則此圖必為其他圖之一部份，仍需保留控制點檔及顏色種子檔。

(十二) 名稱：CONCATE 合併

功能：

合併二個小圖成一個大圖。

呼叫格式：

CALL MCNCTE(OBJ, SOUR1, SOUR2)

說明：

當二個圖形合併時，只產生一個新的總檔，且把二個原圖之祖先數目各加一。

(四) 名稱：SUBTRACT(SUBTRACT A SUB picture from a picture)

減除副圖。

說明：

減除先前由合併而來的副圖。

呼叫格式：

CALL MSUBTR(OBJ, SOUR1, SOUR2)

說明：

僅由 CONCAT 副程式合併幾個副圖而成的一張圖，可用此副程式來減除。減除時先從原圖中的總檔去掉此圖的記錄，再把此副圖的祖先數目減一，若為零，則去除此副圖。

(五) 名稱：DM(Dynamic Motion) —— 動畫。

功能：

由前後二張靜畫，產生中間的動畫。

呼叫格式：

CALL MDM(FNAME, LNAME, GNAME, ISTEP)

說明：

FNAME：表第一張靜畫。

LNAME：表最後一張靜畫。

GNAME：表產生的動畫名稱。

ISTEP：表產生幾張(I2)。

若 GNAME 為 xxxxxx, ISTEP 為 i，

則產生出的動畫名稱為

xxxxxxx001, xxxxxx002, ... xxxxxx*i*

其中 FNAME 和 xxxxxx001 相同。

LNAME 和 xxxxxx*i* 相同。

(六) 名稱：COLOR (着色)。

用途：

對圖形着色。

呼叫格式：

CALL MCOLOR(PICNAM)

說明：

系統先在彩色顯示器畫出圖形輪廓，再從終端機顯出

INPUT XSEED, YSEED FROM CURSOR AND ENTER SWITCH

使用者把十字游標移至欲着色的封閉區域中並按入 'ENTER' 鍵系統再顯出

INPUT COLOR FROM SCREEN AND TYPE IT

使用者鍵入色筆顏色，則系統依著色角度為0，找到封閉區域後塗上顏色，並把被顏色蓋掉的輪廓重描一次，再繼續由使用者着色，直到選擇的顏色為0，系統並在圖形總檔中記入着色角度為0，着色狀態為 'Y' 並產生顏色種子檔。

(C) 名稱：TRANSLATE Relatively (相對平移)。

功能：

相對於原來的位位置平移一個圖形。

呼叫格式：

CALL MTRAKL(OBJ, SOUR, IX, IY)

說明：

- 1 OBJ 可以和 SOUR 相同，表示移後圖形名稱不變。
OBJ 亦可和 SOUR 不同，即平移後產生一個新的圖形。
- 2 IX, IY 為平移相對座標，IX 向右為正，向左為負，IY 向上為正，向下為負。

(C) 名稱：TRANSLATE ABSOULTELY (絕對平移)。

功能：

移動圖形使重心落在指定的一點。

呼叫格式：

CALL MTRAAB (OBJ, SOUR, IX, IY)

說明：

參考(十六)之說明。

(C) 名稱：ROTATE Relative to Center of Gravity (對重心旋轉)。

功能：

對重心旋轉。

呼叫格式：

CALL MROTCG (OBJ, SOUR, ITHETA)

說明：

- 1 見(十六)說明 1。
- 2 ITHETA 為旋轉角度(弧度)，反時針為正，順時針為負。
- 3 改變着色角度旋轉後角度 = (旋轉前角度 + 旋轉角度) 除 360 的除數。

(C) 名稱：ROTATE Relative to a specified point (對指定點旋轉)。

功能：

對指定點旋轉。

呼叫格式：

CALL MROTPT (OBJ, SOUR, IX, IY, ITHETA)

說明：

1 IX, IY 為指定點座標。

2 見 (十八) 之說明。

(四) 名稱：MIRROR (鏡射)。

功能：

對一指定軸鏡射。

呼叫格式：

CALL MMIRR (OBJ, SOUR, IX1, IY1, IX2, IY2)

說明：

1 (IX1, IY1), (IX2, IY2) 為鏡射軸上的二點。

2 鏡射後着色角度等於 2 倍鏡射軸角度減 3 倍原着色角度除 360 之除數。

3 見 (十六) 說明 1。

(五) 名稱：ZOOM a picture along X Axis Relative to a specified point
(對指定點沿 X 軸放大)。

功能：

對指定點沿 軸放大)。

呼叫格式：

CALL MZOOMX (OBJ, SOUR, IX, IY, XFACT)

說明：

1 IX, IY 為指定點座標。

2 見 (十六) 說明 1。

(六) 名稱：ZOOM a picture along Y Axis Relative to a specified point
(對點沿 Y 軸放大)。

功能：

對指定點沿 Y 軸放大。

呼叫格式：

CALL MZOOMY (OBJ, SOUR, IX, IYYFACT)

說明：

見 (二十一) 說明。

(七) 名稱：ZOOM a picture along X & Y Axis Relative to A specified point
(沿 X, Y 軸對指定點的放大)。

功能：

沿 X,Y 軸對指定點放大。

呼叫格式：

CALL MZOOMD (OBJ,SOUR,IX,IY,XFACT,YFACT)

說明：

見 (二十一) 說明。

(四) 名稱：ZOOM a picture along X Axis Relative to Center of Gravity
(沿 X 軸對重心放大)。

功能：

沿 X 軸對重心放大。

呼叫格式：

CALL ZMCGX (OBJ,SOUR,XFACT)

說明：

見 (二十一) 說明。

(五) 名稱：ZOOM a picture along Y Axis Relative to Center of Gravity
(沿 Y 軸對重心放大)。

功能：

沿 Y 軸對重心放大。

呼叫格式：

CALL ZMCGY (OBJ,SOUR,YFACT)

說明：

見 (二十一) 說明。

(六) 名稱：ZOOM a picture along X & Y Axis Relative to Center of Gravit
(沿 X,Y 軸對重心放大)。

功能：

沿 X,Y 軸對重心放大。

呼叫格式：

CALL MGMCGD (OBJ,SOUR,FACT)

說明：

見 (二十一) 說明。

(七) 名稱：CAMERA (拍攝)。

功能：

拍攝一張畫面。

呼叫格式：CALL MCAMER(TIME)

說明：

TIME 為連續拍攝的次數。

(兎)名稱：IMAGE READ (影像讀取)。

功能：

讀取彩色顯示機畫面上的影像。

呼叫格式：

CALL MIMR (PICNAM)

說明：

系統顯出

INPUT UPPER LEFT CORNER OF WINDOW FROM JOKSTICK

使用者移動如意桿，使十字游標移到欲讀取影像的窗函數之左上角，鍵入 'ENTER' 鍵。

系統再顯出

INPUT RIGHT LOWER CORNER OF WINDOW FROM JOKSTICK

使用者以如意桿移動十字游標至窗函數右下角，並按 'ENTER' 鍵入。

輸入窗函數之範圍後，系統即把彩色顯示器上的畫面，依窗函數的指定範圍讀取影像，影像存放格式見參考資料[13]。

(兎)名稱：IMAGE WRITE (顯出影像)。

功能：

顯示已貯之影像。

呼叫格式：

CALL MIMW (PICNAM)

說明：

依照原影像位置(存放時所定的窗函式顯示)。

(兎)名稱：IMAGE Write to a specified Location (顯示影像在指定地方)。

功能：

在指定地方顯示已存之影像。

呼叫格式：

CALL MIMWRL (PICNAM,IX,IY)

說明：

依照顯像原貯存之窗函數範圍，在指定位置顯像，IX,IY 表示新的窗函數之左上角座標。

一般卡通的製作過程大致上包括：(1)主題的企畫，(2)改編故事為劇本，(3)分鏡及漫畫的導演，(4)製作原畫及動畫，(5)描線或影印去除遮隱線，(6)着色，(7)拍攝成影片，(8)剪接編輯，(9)配音(10)試片等步驟。我們以上述(4)至(7)步驟的電腦化為主要目標，發展成功了POCA系統。

POCA系統採用自行設計之POCA語言當做卡通製作的命令語言，它是屬於程序化的語言，經先前處理器(Preprocessor)之翻譯後，產生之目標程式(Object Program)，係一些以FORTRAN語言寫成的語意副程式呼叫(Semantic Subroutine Call)，此時進入電腦動畫系統核心處理和顯像，若有需要則可經由人機互動式(Man-machine Interacting)的程序處理，從事原畫之建造、修整以及產生動畫塗色等工作，如此反覆進行直至拍攝到滿意的畫面動作，才算是一個完整的週期。

卡通畫面通常是由相當複雜的圖形所組成，而且通常一張畫面是由很多張畫面重疊而成，在圖形的處理上於是就產生遮隱線(Hidden Line)[14]的問題。當一張背景的圖畫有部份的圖形為另一張圖畫遮住時，則背景圖的遮隱線就必需去除，而其不被遮住的線段，又需與另一張圖相接部份的線段重合成為一個新的封閉區。換言之，當二張卡通圖畫有重疊時，就必需重新描線並擦掉遮隱線。這項工作若以求B軟楔曲線間的交點進行，將是一項巨大的計算工程，因此採用求B軟楔曲線間交點的方法處理遮隱線是不切實際的，將使計算時間長到無法忍受即時作業的要求。另外一種方法是，採用區域重疊的方式，來解決這項問題。我們可以指定最深的背景區域為最優先次序的顯像段，如此我們就可得到一組依照優先次序顯像編排的封閉區域。第一個元素是最優先的區域，首先被呼叫出來顯像，其顯像是依照筆圖(Bit Map)的方法，將其區域用某種指定的顏色塗上，第二個元素再呼叫出來亦以同樣的方式顯像，若它有部份和第一個元素重疊時，則第一個元素被遮住的部份，會被塗成用來指定第二個元素的顏色。如此繼續，一直到整個被依序編排的元素皆已處理後為止。用這種方式的處理避開了事先求交點的工作，雖然多塗些被遮蓋的顏色，但顯像處理時間卻大量地被節省了。

若採用區域重疊的方式顯像，則最後的邊線若要描成黑線時，將是怎樣的處理？事實上這項問題甚易解決，因為只要在每次

一個區域顯像時，同時亦將其邊線描成黑線則在下個元素顯像時，若有遮住此區域的部份圖形則亦會自動去除被遮住的黑線，所以當這種過程一直繼續到整個元素皆被處理過後，剩下來的黑線就是完整的一張卡通畫面的描線。

目前我們所用的彩色終端機是屬於影像處理的機種，稱為 RAMTEK 9300 型顯像終端機，塗色的方法是移動十字游標(Cursor)至一個封閉曲線的附近或內部，若此封閉曲線的區域用筆圖(Bit Map)表示，則我們只需指定顏色的編號，就可將顏色塗上該封閉區域，所以着色方法最主要的考慮是，如何將一個任意形狀之封閉區域的筆圖決定出來？但是動畫系統尚要考慮中間過程所產生圖畫的着色問題，我們不應該對單動作的每一張圖畫逐一着色，因為這樣會花費太多的人力在着色的工作上，正確的方法應該是設計一套自動着色的方法，讓塗色可以隨著動畫圖形的改變而能自動找到新的封閉區域，雖然完全正確的動畫自動塗色法是很困難找到。因為動畫圖形的變化和封閉區域變形的關係是不可捉摸的，但是高準確的方法却是不難找到的。因為我們可以利用動畫演算法來產生中間過程內各個封閉區域用來塗色指示之十字游標的位置，除非相鄰圖形的變化太大，否則十字游標通常會落在新的封閉區域內。藉著這樣的方法自動塗色其本身也是一種決定十字游標的動畫。根據我們實驗的結果顯示，這種方法有百分之九十以上的自動塗色是正確的，而不正確的部份可經由互作用式的修改而達成完全正確塗色的任務，因此，可減少大量的着色工作。

6.1 POCA 系統的硬體結構如圖 6-1 所示，採用的儀器如下：

1 數位儀

作為圖形輸入的工具，每一幅靜畫係用十字游標輸入其圖形，圖形係以 B 軟楔曲線所對應之控制多邊形之控制點表示。當輸入這些控制點在數位儀上的座標後，系統會根據其上所訂定之窗函數(Window)，將決定圖形的實際座標數據存入主電腦處理。

2 TEKTRONIX 4054 高析度圖形終端機

為一 1024 × 780 高析度的螢光顯示幕 (Storage Tube Display)，當圖形由數位儀輸入完畢，圖形即在此終端機的螢幕上顯示出。而不滿意的圖形，亦在此機上經由互作式修改，直到滿意為止。

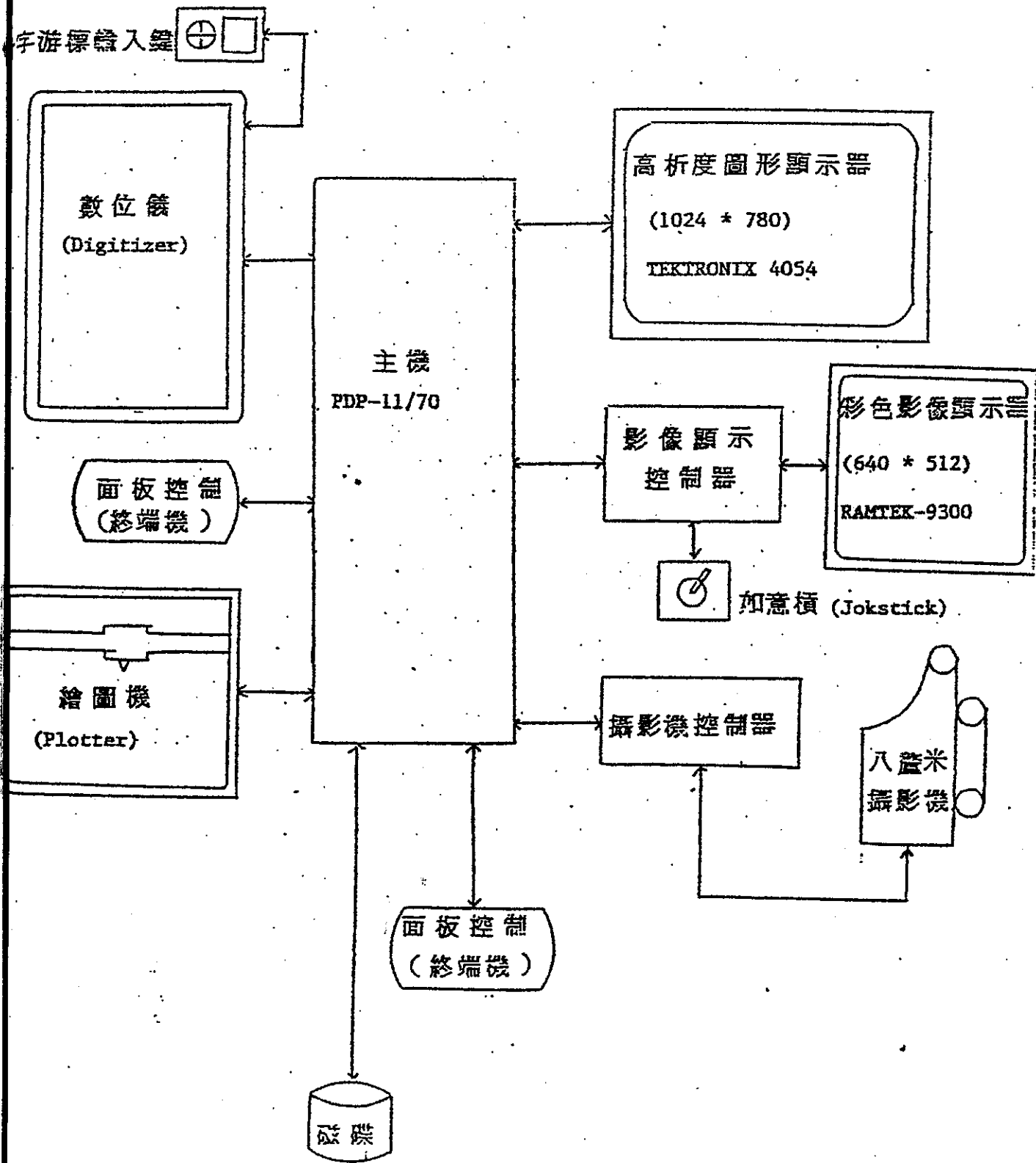


圖 6-1 POCA 系統硬體結構

3 繪圖機

當圖形修改滿意後經由繪圖機畫在紙上以作為靜畫存檔或故事版編輯之用。

4. RAMTEK 9300 彩色影像顯示器

靜畫經修改完成後，在此機上以操作如意桿 (Jokstick) 之互動式着色，並在此機的螢幕上顯示出影片畫面的結果，畫面上的十字游標經由如意桿遙控移動位置。

5 八厘米攝影機及其控制器

八厘米攝影機用以拍攝從 RAMTEK 9300 顯示器上顯示出的影片畫面，為配合製作的程序化，攝影機為連線控制，一簡單的控制器作為攝影機及主機間之介面。此控制器是由主機的一個終端機輸出口，經由 RS232 介面送出連續二數元組 (bytes)，作為拍攝一次的控制訊號，控制器由微處理機 (KIM-1) 經由 RS232 介面收到控制訊號後，作一邏輯判斷，即發出一訊號推動一個繼電器 (Relay) 使攝影機的快門拍攝一次，如圖 6-2。

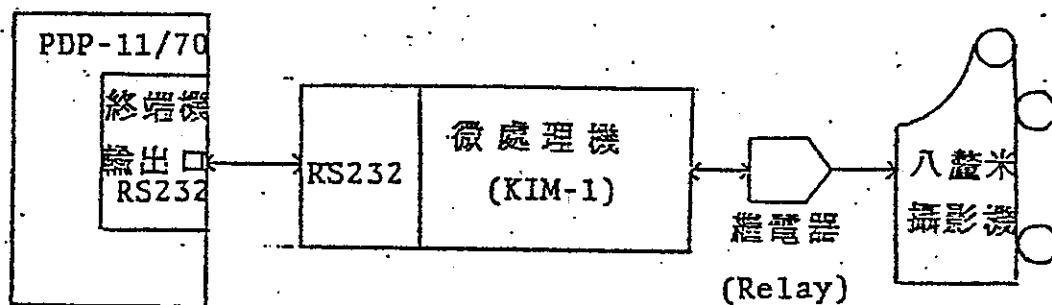


圖 6-2 攝影機控制器

6.2 開機使用程序

因圖形處理的複雜及程式中須預留極大的緩衝區 (Buffer Space)，作為圖形數據的暫存空間，又因在 PDP-11 的 RSX-11M 作業系統下，限制一個可執行的工作程式 (Task)，不能超過 64K 數元組 (Byte)，且為使系統往後易於擴充，POCA 系統的所有程式分為六個工作程式。工作程式之間利用共用旗幟 (Common Flag) 及通訊緩衝器 (Communication Buffer)，作為同步 (Synchronization) 及工作程式間之資料傳送用，所以使用 POCA 系統必須分別執行 (Run) 六個工作程式，其開機使用的程序如下：

- 1 打開數位儀，TEKTRONIX 4054，RAMTEK 9300 的電源。
- 2 因 TEKTRONIX 4054 本身為一獨立處理機，所有繪圖功能均由其本身的獨立軟體程式執行，線上使用需先設定一些初值。

在 TEKTRONIX 4054 打入

```
> CALL "RATE",9600,4,2  
> CALL "GRAFIN", " ",2,0  
> CALL "TERMIN"
```

3 數位儀讀取方式設定為讀點方式 (Point Mode)。

4 在面板控制之終端機上執行 POCA 系統的六個工作程式：

```
>RUN IMARW/TASK=IMAGE + +  
>RUN CURVE/TASK=SPL + +  
>RUN FILL/TASK=FILL + +  
>RUN BSPLNE/TASK=BSPLNE + +  
>RUN DMTSK/TASK=DMTSK + +  
>RUN POCAL/TASK=MAIN + +  
:
```

各工作程式之功用及其間之關係，見系統軟體結構。

其中當 > RUN IMARW/TASK=IMAGE 及 > RUN FILL/TASK=FILL 時，因涉及 RAMTEK 9300 之軟體模組 (Software Package) 的起始設定，需回答 'Y' 表示起始值設定為標準形態。

+ 表鍵盤上的 RETRUN 鍵。

% 表 POCA 系統開始接受 POCAL 命令 (見 POCAL 命令語言的描述)，此時鍵入 POCAL 命令，即可開始製作卡通圖畫。

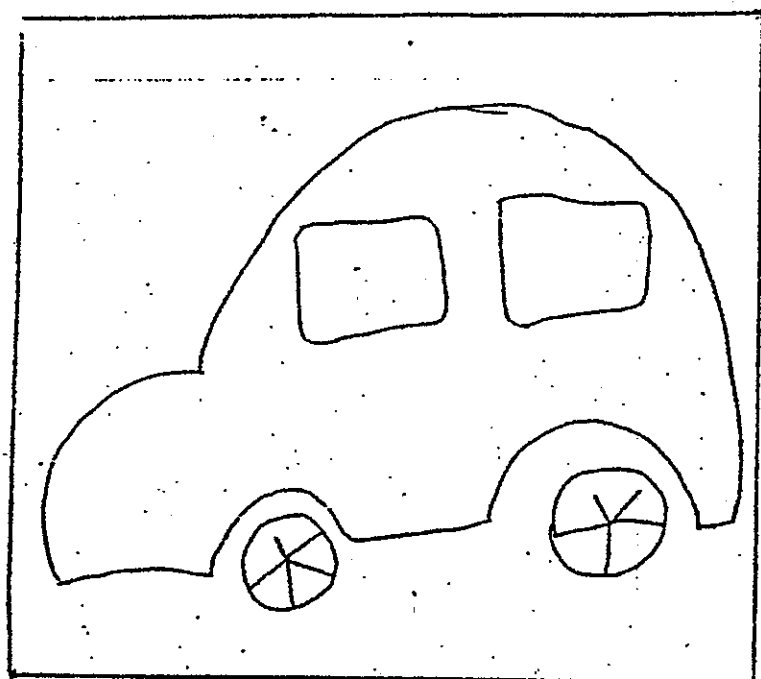
6.3 製作實況及結果

本節中示範從手稿到圖形着色後的靜畫

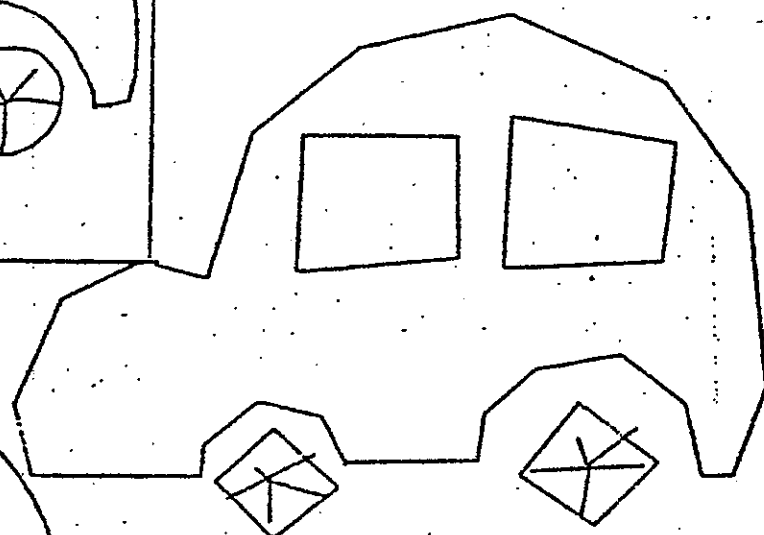
- 1 手稿一圖 6-3(a)
- 2 輸入圖形的控制多邊形一圖 6-3(b)
- 3 輸入圖形一圖 6-3(c)
- 4 加點及移點後之控制多邊形一圖 6-3(d)
- 5 修改後之圖形一圖 6-3(e)
- 6 着色前之圖一圖 6-3(f)
- 7 着色後之圖一圖 6-3(g)

為測試 POCA 系統在電腦輔助卡通設計上之實用性，我們嘗試製作並拍攝了一部 4 分 20 秒的實驗影片，從原畫輸入到影片拍攝

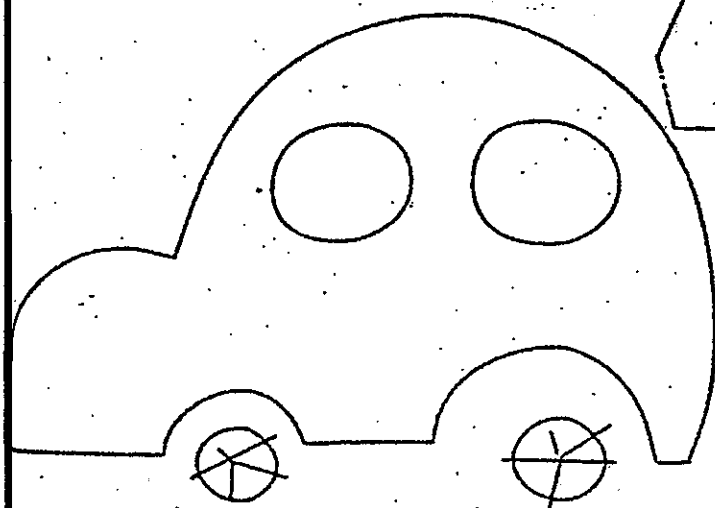
完成共耗 120 時 / 人，訂有 10 套（見附錄）母版約為 10 幅左右基本圖形組成，輸入的基本圖形大約 150 幅，拍攝使用的圖形大約 550 幅，除輸入外餘為自動產生出的圖形。



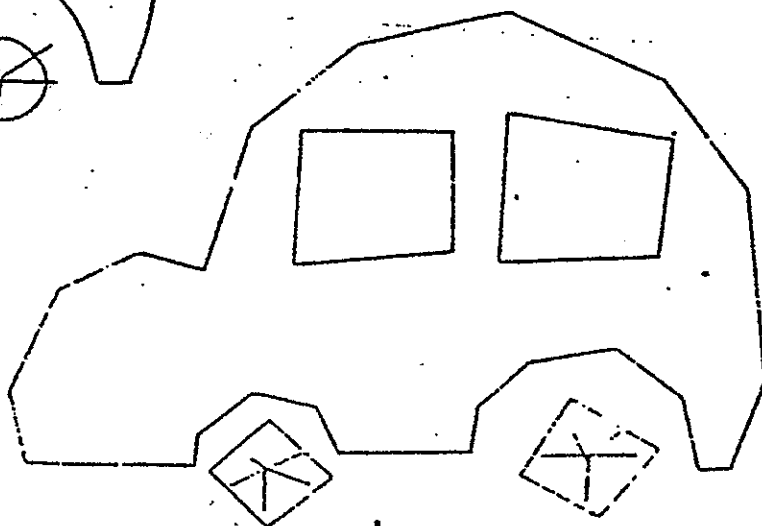
(a) 手稿



(b) 輸入圖形之控制多邊形

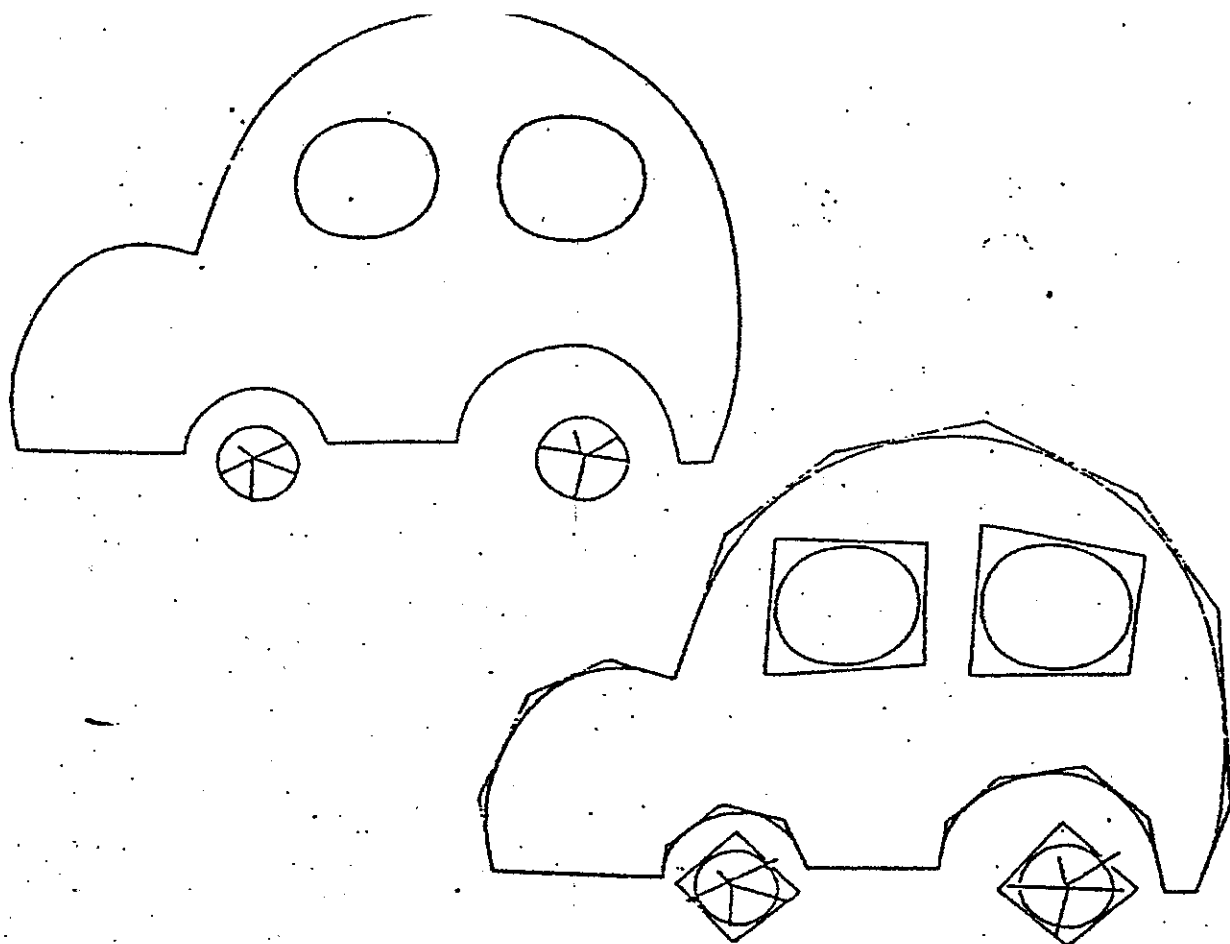


(c) 輸入圖形

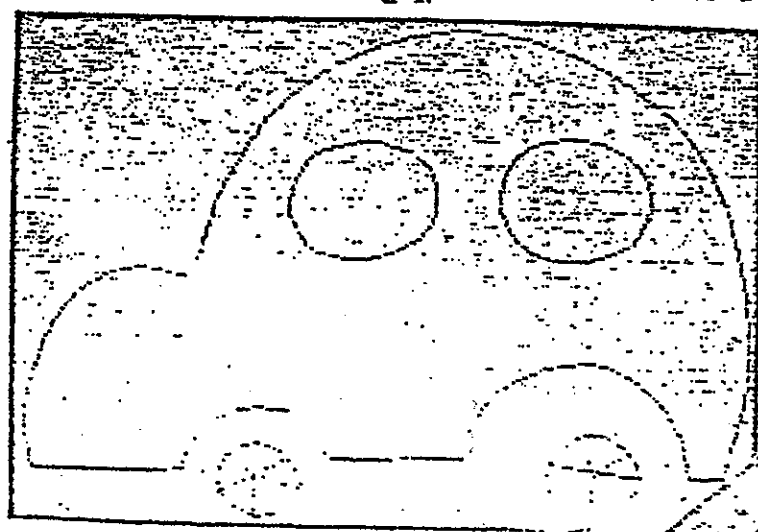


(d) 修改後之控制多邊形

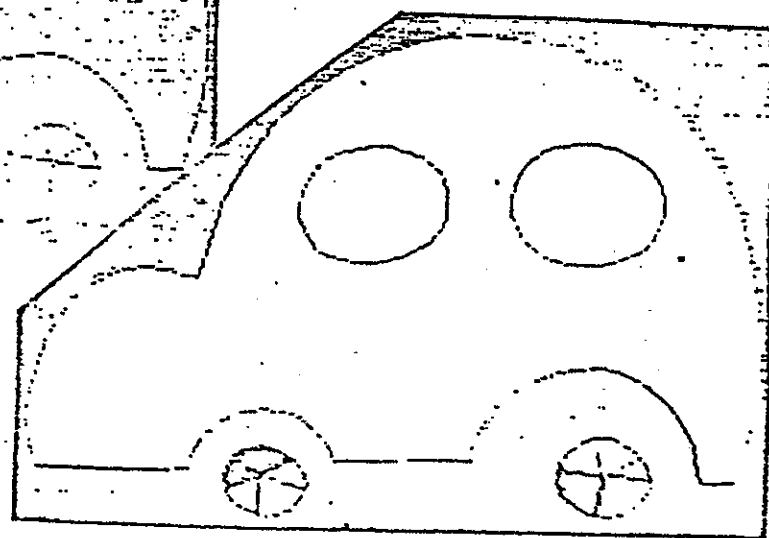
圖 6-3 製作實例 (1)



(e) 修改後之圖形及其控制多邊形



(f) 著色前之圖



(g) 著色後之圖

圖 6-3 製作實例 (2)

6.4.1 着色使用法

卡通製作的系統是以能夠提供跨大、活潑、有趣的卡通畫爲主旨，甚少以“非常逼真”來引人注意，所以我們所採用的着色方法原則上使處理時間及貯存空間儘量減少，而不失彩色圖畫的功能。最基本的方法，就像幼兒的着色簿一樣在着色簿中先描好塗色的輪廓，再依每一塊（即一封閉區域）塗上一種顏色，在POCA系統裏，我們採用這種塗色方式，使用少數的色彩一塊一塊塗上顏色，已能夠形成一幅色彩豐富又活潑的畫面了。使用的着色方法如下：

- 1 當 POCA 系統開始執行時在 RAMTEK 9300 上，即時顯出一個白色畫面當做一張畫紙，在畫紙四週標有1-64 表 64 種顏色的色筆，以供着色使用（見附錄）。
- 2 當進入着色程式時，POCA 即先描出原畫的輪廓見圖 6-4。
- 3 使用 POCA 者以互動式方式，由背景往前景依次着色，（根據美術學上着色原則）着色時是以十字游標指出欲着色的封閉區域中的一點，（稱爲顏色種子 — COLOR SEED）再指出使用的色筆編號，（色筆號碼加上 100 表示塗滿畫紙，即塗底色，色筆前加上“-”表示所塗區域的邊線要描成黑邊，0：表示着色完畢）系統即以指定的色筆塗上指定的區域。
- 4 系統重新顯示因着色而被蓋掉的輪廓。
- 5 依 3 4 的步驟進行，一直到全部塗完顏色後停止。

圖 6-5 示出一張着色完成的畫面，可以看出這種着色方法所得的畫面是非常豔麗且漂亮。

6.4.2 着色演算法 (Painting Algorithm)

POCA 系統採用的着色演算法，可用以下流程圖表示。
值得說明之處有以下幾點：

- 1 爲了着色方便及易於尋找指定的封閉區域，我們以一個 270×240 的二進位筆圖 (Binary Bit-Map) 來記錄，“1”表示有曲線通過，“0”表示未有曲線通過圖形元素 (Pixel) 的分佈情形，再從筆圖中去找找到欲着色的封閉區域，所以開始先清除筆圖，即令每一圖形元素值均爲 0。
- 2 3 由 B — 軟楔函數計算公式算出近似曲線的軟楔段，在筆圖中可能爲不連接的二點，所以任二鄰點之間的圖形元素，也

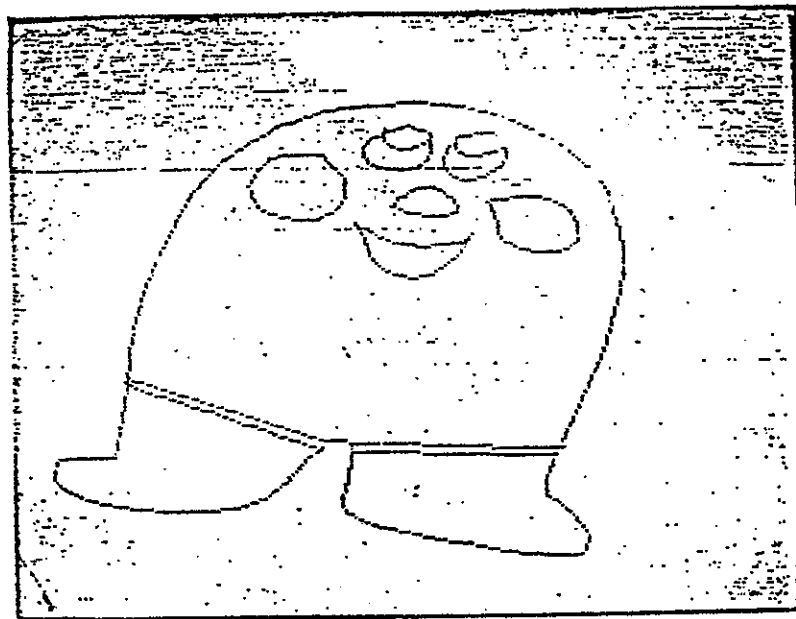


圖 6-4

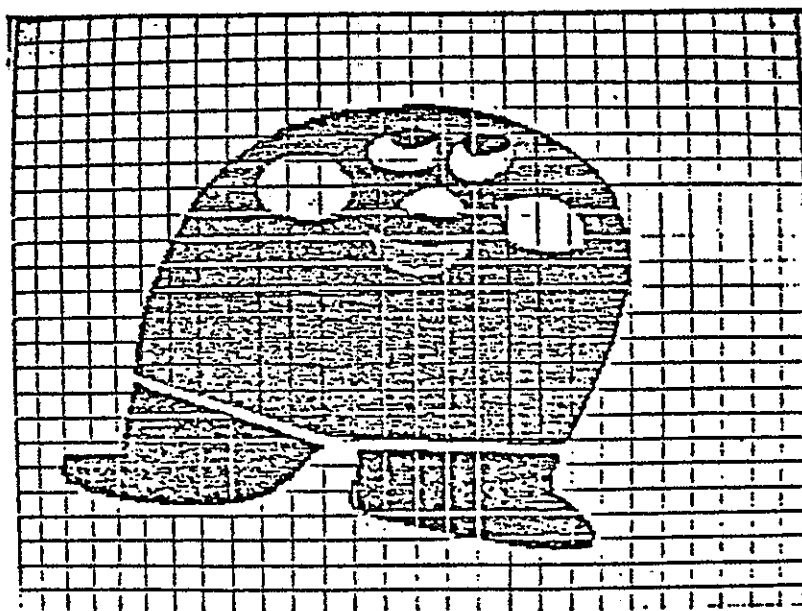
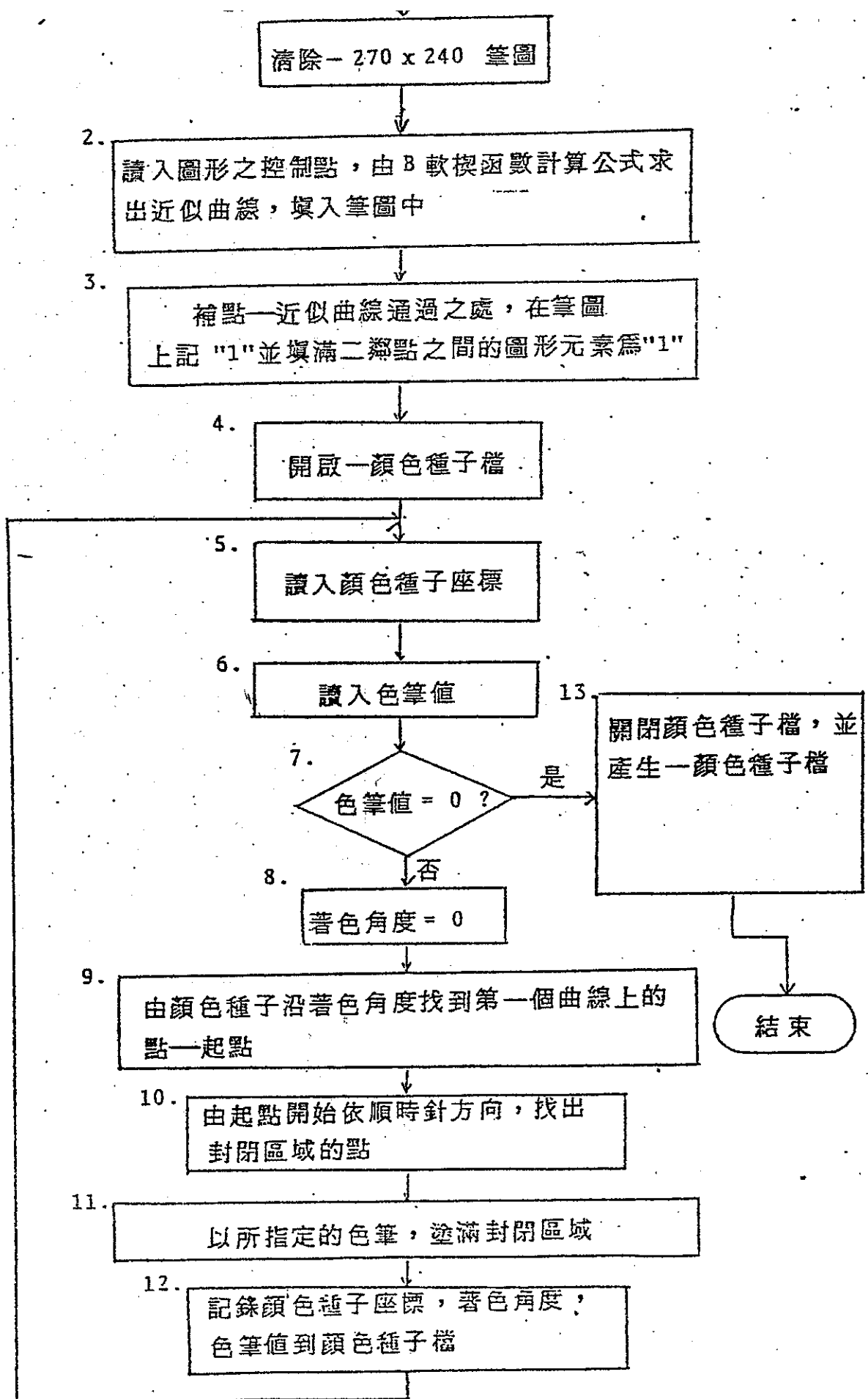


圖 6-5



著色法流程图

[illegible]

圖 6-6 補點一在筆圖中記錄近似曲線通過之處

-

-80-

Y 軸方向，則找到的區域為Y；亦即由 A 沿小於 α 角大於 β 角的方向則找到區域為Y，其他為X。基於此點，我們定義着色角度為顏色種子尋找區域所前進的方向，在塗色中着色角度規定為0，即均沿正 X 軸方向尋找，故如上圖。欲着封閉區域X，則選 B 為顏色種子，若欲着封閉區域Y，則選C，當圖形旋轉或鏡射時此着色角度需加以調整。9. 10. 因為封閉區域內任一點的鄰點，在筆圖中，必定出現，在以此點為中心的 3×3 筆圖中（在外圍八點之一）如圖 6-8 所示。

0	0	0
ϕ	x	0
0	0	0

x : 筆圖中任一點
 $\phi, 0$: 可能的鄰點

圖 6-8 筆圖中任一點的可能鄰點

又因為着色角度為0，所以沿着着色角度找到區域上的一點（ 3×3 筆圖的中心點）後，從 ϕ （起點）開始，依反時針方向繞中心點尋找，必能找到起點的下一鄰點，再以找到的鄰點為 3×3 筆圖的中心點，前一中心點為起點，依反時針方向繞中心點尋找，即可找到下一鄰點。依此步驟反覆處理，即可找到封閉區域的所有點，如圖 6-9。

					x	0	0	0	x	x				
				0						x				
			0							ϕ	0			
			x							4	5	x		
			x							1	2	0		
			0							3	6	0		
			0						5	7	8	x		
			x	0	0	x	0	0	0	0	x			

圖 6-9 封閉區域的找法

11. 填色演算法：

- ① 令所有組成多邊形的點為 P 。
- ② 在 P 中找最小的 x 值及最大 x 值分別稱為 MinX 及 MaxX 。
- ③ 在 P 中找次小的 x 值（大於 MinX 中最小的） NMinX 。
- ④ $X=\text{MinX}$ 和 $X=\text{NMinX}$ 的範圍中找出通過此區域的所有線段 S 。
- ⑤ 若 S 中有任二線段相交於 C ，則令 $\text{NMinX}=C$ 再跳回至步驟 ④。
- ⑥ 求 S 中之所有線段和 $X=\text{MinX}$, $Y=\text{NMinX}$ 交點之 y 座標，按 y 值由小到大排列。
- ⑦ 依次以色筆畫滿步驟 ⑥ 中所求得的小區域，（漸增加 x 值）
令 $\text{MinX} = \text{NMinX}$ 。
- ⑧ 若 $\text{MinX} < \text{MaxX}$ 則跳回步驟 ③，否則結束。

此演算法是依照多邊形端點之 x 座標值的大小分成數個區間，再分別對其中的小區域，以指定的色筆用平行於 y 軸的直線畫滿介於此區間中的小區域。步驟 ⑤ 中，檢查區間中是否有線段相交，若有線段相交則再分割此區間為更小之區間，加以塗色如圖 6-10。 v_1, v_2, v_3, v_4 為多邊形的四個頂點， v_1 及 v_2 的 x 座標分別

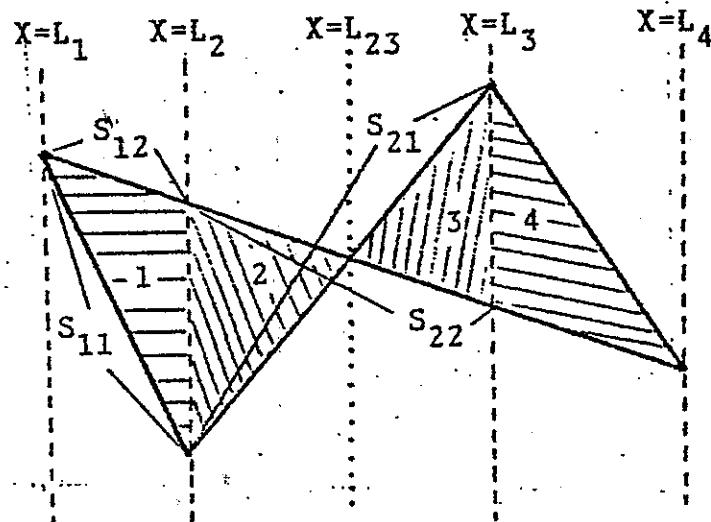


圖 6-10 把一個區間再分割為二

為 L_1 及 L_2 ，第一次找到 $\text{MinX}=L_1$, $\text{NMinX}=L_2$ 形成一介於 L_1 及 L_2 之間之一小區間，找到二線段 S_{11} 及 S_{12} ， S_{11} 及 S_{12} 在 L_1 及 L_2 之間沒有交點，區域 1（介於 L_1 , L_2 及 S_{11} , S_{12} 之間即為一個找到應塗色的小區域，再依次找到 $\text{MinX}=L_2$, $\text{NMinX}=L_3$ 介於其間之二線段 S_{21} , S_{22} 交於 v_{23} 所以此區間再被 L_{23} 分割為 $\text{NMinX}=L_{23}$ 回到步驟 ④ 繼續進行演算法的處理，依圖 6-10 之 1, 2, 3, 4 填滿 4 個小區域即完成以 v_1, v_2, v_3, v_4 形成之多邊形的塗色工作。

本文提出一種電腦動畫設計系統之模型，此模型考慮了人機互動時的「宜人因素」(Human Factors)，讓動畫設計者可以用自然的方式，將原畫直接用光筆繪圖輸入電腦。考慮了朦朧語言，可以用自然的語言和電腦交談，並將電腦動畫問題以朦朧集合表示，使整個系統的設計是根據數學和資訊科學的理論推演。數學上，我們採用了B軟楔的近似函數理論，推演出適合產生動畫的演算法，資訊科學上，我們採用了人工智慧的觀念，使電腦範疇的工作是完全自動的。換言之，使電腦在接受到命令語言後，能做出非數值的判斷，並產生用數值表現的圖形。

電腦動畫設計系統之模型是由幾個數學模式分別表現的子系統匯集而成的。本文中所提之「圖形數據壓縮演算法」對於大量圖形的繪製及儲存提供了很多的方便，因為若以人爲尋找控制點的方式來輸入圖形，是非常困難且費時的。有了此種演算法，使用者可以不必遷就因數學上採用近似函數所產生的繪製限制，所以採用此種「反求過程」的數學模式在繪製原畫工作上，是非常宜人的。

動畫產生的效果如何，人人感受可能不一樣，亦即動畫問題的本身就是朦朧的，所以，動畫系統和朦朧集合理論之關係是非常密切的。今用B軟楔函數表現圖形，就需將B軟楔曲線所表現的圖形形狀和朦朧集合理論互相結合，這種關係的數學模式於是成了整個動畫產生過程的主要依據。我們以卡通製作爲對象，將這些數學模式在電腦實現，其結果是相當令人滿意的。

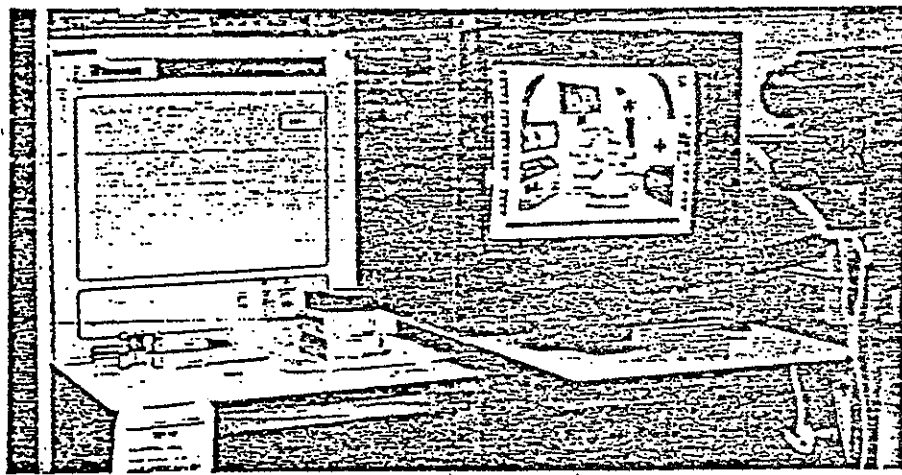
綜上所述，本研究的主要成果，是提出一創新的電腦動畫設計法，它的特色是完全以數學及資訊科學的理論爲基礎，除了具有「先導性」、「創見性」、和「理論性」外，更以實際的實驗來証實其理論與實用的可行性。

8 参 考 文 献

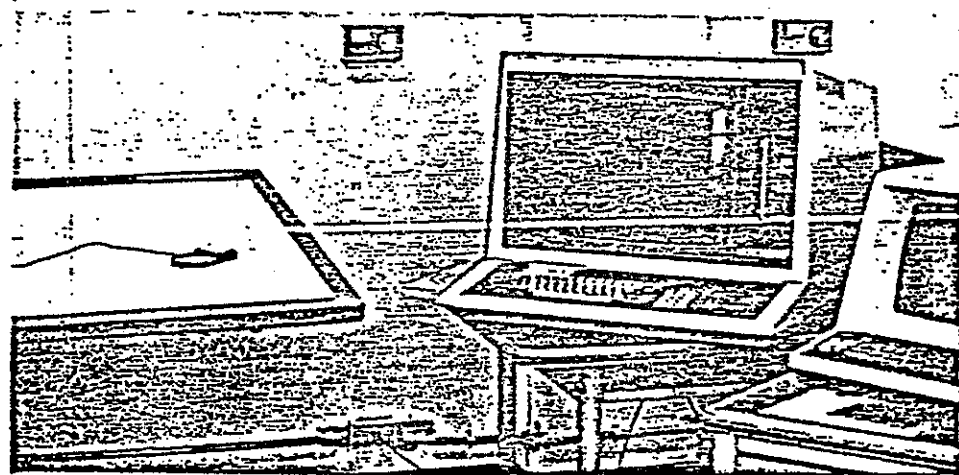
1. Wein, M. "Computer Animation," in Encyclopedia of Computer Science and Technology, Vol.5, Marcel Dekker, New York, pp. 397-436, 1977.
2. Burtnyk, N., and Wein, M., "Interactive Skeleton Techniques for Enhancing Motion Dynamics in Key Frame Animation," Comm. ACM, pp.564-569, Oct. 1976.
3. Reeves, W. T., "Inbetweening for Computer Animation Utilizing Moving Point Constraints," Proc. ACM Siggraph, Vol.15, No.3, pp.263-269, August 1981.
4. Bezier, P. "Numerical Control -- Mathematics and Applications," Wiley, New York, 1972.
5. Forrest, A. R., "Interpolation and Approximation by Bezier Polynomials," Computer J., Vol.15, No.1, pp.71-79, 1972.
6. Coons, S. A., "Surface Patches and B-Spline Curves," in Computer Aided Geometric Design, Barnhill and Riesenfeld (eds.), Academic Press, 1974..
7. Wu, S. C., Abel, J. F., and Greenberg, D. P., "An Interactive Computer Graphics Approach to Surface Representation," Com. ACM, pp.703-712, Oct. 1977.
8. Zadeh, L. A., "Calculus of Fuzzy Restrictions," in Fuzzy Sets and Their Applications to Cognitive and Decision Processes," Edited by Zadeh, L. A., et al., Academic Press, pp.1-39, 1974.
9. Cheng, K. Y., "On Fuzzy Similarity Relation of B-Spline Curves," Proc. of International Computer Symposium, Taipei, pp.1292-1298, 1974.
10. Cheng, K. U., "A B-Spline Based Computer Animation System," Journal of the Chinese Institute of Engineers, Vol.4, No.1, pp.39-45, 1981.
11. Chang, S. K., "On the Execution of Fuzzy Programs Using Finite-State Machines," IEEE Trans. on Computers, Vol.C-21, pp.241-253, 1972.
12. TEKTRONIX Terminal Control System Reference Manual.
13. RAMTEK FORTRAN Image and Plot Package User's Guide.
14. Giloi, K. W., "interactive Computer Graphics -- Data Structures, Algorithms, Languages," Prentice-Hall, Inc., Englewood Cliffs, N. J., pp.154-158, 1978.

.1
.2
.3
.4
.5
.6
.7
.8
.9
.01
.11
.21
.31
.41

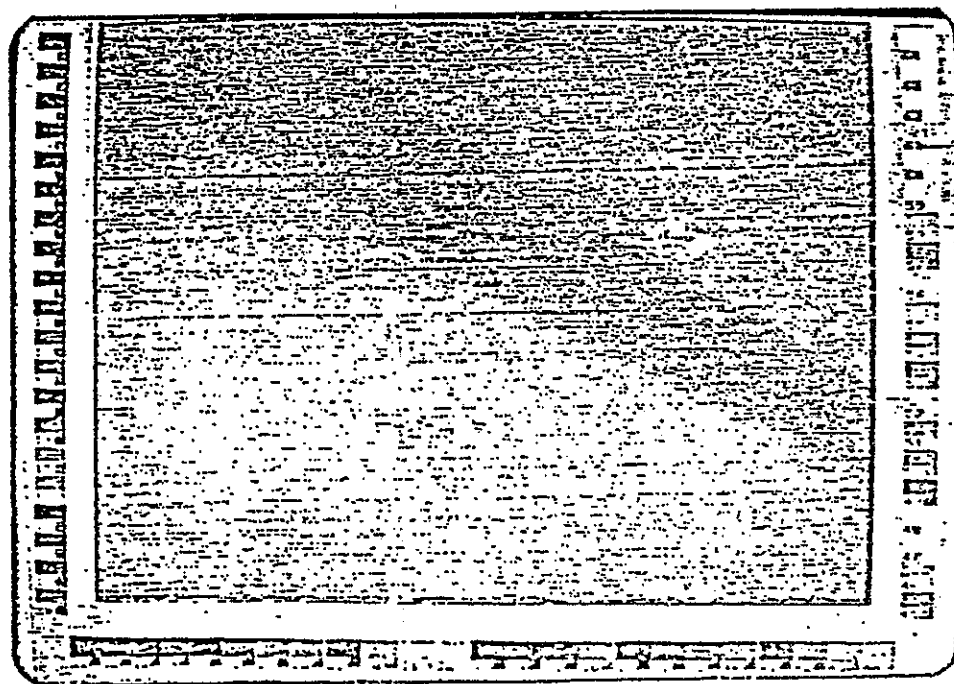
附 錄



圖形顯像在彩色螢幕上



數位儀及高析度顯示幕



64種色筆分佈在畫面四週

