

TR-82-016

電腦輔助卡通設計系統

鄭國揚

陳原森

中華民國七十一年五月

中研院資訊所圖書室



3 0330 03 000029 8

書	考	參
借	外	不

0029

摘 要

本文提出一種電腦協助卡通製作的系統，它是以程序爲主的卡通製作，故以(POCA系統 (Procedure Oriented Cartoon Animation System))稱之，它的命令語言稱爲 POCAL, (POCAL敘述句經前處理器翻譯成福傳的副程式呼叫，供執行圖形的處理，以達成用電腦協助設計卡通製作靜畫、動畫、著色等工作)。

POCA系統之設計原理：翻譯器是屬於語法直接編譯的方式，靜畫的產生是用B軟楔曲線建造，動畫的產生是以考慮圖形相似性的演算法爲骨幹，著色的過程亦以動畫的方式自動塗上大部份中間過程的動畫，其中僅小部份是用交互式修整。

本文除了說明POCA系統之設計原理及實務設計外，並以實例示範POCA系統的實用性，其中實例示範除了用照片顯示外，更以拍攝一部約5分鐘的卡通影片實驗測試電腦卡通製作的經濟效益及時間效益，其結果相當滿意，和人工製作相比，約可縮短100倍的時效，經濟效益可能也在這樣的倍數範圍。

第一章 緒論

一般卡通的製作過程大致上包括：(1)主題的企劃，(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)剪接編集，(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，發展成功了一套程序化之電腦卡通製作系統簡稱 POCA 系統：Procedure Oriented Computer-Assisted Animation.

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交互式 (Man-machine Interacting) 的程序處理，從事原畫之運送，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。

POCA 系統係採用我們在中央研究院資訊科學研究所發展的方法 [1]，當做動畫系統的演算處理核心，此種方法主要是採用 B 軟線 (B-spline) 曲線為基礎，並輔以模糊集合論 (Fuzzy Set Theory)，人造智慧論 (Artificial Intelligence Theory)，互作電腦圖形理

論 (Interactive Computer Graphics Theory) 以及近似函數論 (Approximation Theory) 等學理所演變成的演算法。

N. Burtnyk and M. Wein [2] 曾提出 Keyframe Animation 產生動畫的方法,最近 W.T. Reeves [3] 亦提出一產生動畫的方法,不過他們的方法皆係採用內插的近似函數,甚難加入瞬態集合論及人造智慧論的技巧,因此比較困難產生有效的演算法。由於我們的方法和他們的方法大相逕異,若以電腦的術語來講他們的方法是屬於低階的,而我們的方法是屬於高階的,因為我們所用到的理論較為深厚,從系統的效能上做個比較的話我們的方法似乎較具備有自動調整的能力,因此比較有效且具彈性,易於擴充,為使讀者更容易瞭解上述的說明,在本文後附錄了以上三篇論文供讀者參考。

在電腦系統裏,製作原畫是屬於互作電腦圖形的範疇,如何將一張複雜的圖畫輸入電腦,又能適合於各種數學上的演算處理是其最重要的考慮點。由於卡通的圖畫並不考慮圖形的精確性,主要還是考慮怎樣的使互作式較容易設計原畫。基於這樣的考慮我們認為 B 軟標函數 [4], 似乎優於其他的近似函數表示法,因為 B 軟標函數有許多互作式上的優點,主要的優點為圖形是由控制點所決定,修飾圖形僅需調整變化部份的控制點。其缺點是,不自然的輸入方式需輸入素描控制點而不是圖形曲線上的點,但這項缺點並不嚴重,因為只需將素描控制點決定曲線形狀的一些規則

訂出,根據我們的經驗一般人(沒有數學基礎者)很快就學會,事實上本文所舉的例子,以及我們製作了幾分鐘的卡通影片,其中的原畫就是經訓練一天的人員繪畫的,另一項缺點是,B軟探曲線顯像時的計算時間長,以及計算相當多,使中央處理機非常繁忙,因此共時的電腦系統(Time-sharing System)不適用於產生即時的卡通顯像,不過這項缺點可以稍微獲得改善其法是在顯像B軟探曲線的演算上,做更快速的收斂測試,有關這方面的理論請參閱資料[5]。

當從原畫產生動畫時很明顯地,可知原畫藉着形狀慢慢地改變而產生動作,從這個觀點看,在動畫裏每相鄰二張的圖形仍然還是相當相似的,唯漸離的二張圖形則愈不相似,所以原畫經過動畫系統處理時,隨着動作的進行,這種相似性若產生振盪現象,則會有變形的現象。出現這種現象是不自然的消除的,方法就是需分成好幾個動作處理,因此我們所談及的動畫僅限於一個單動作的變化,在一個單動作裏第一張到最後一張圖形間的變化是漸進的,不會產生相似性的振盪的,譬如圖1-1的腳步,從①到②其相似性是漸進的,愈來愈不像第一張圖畫,但是從①到③其相似性就產生了振盪現象,先漸不像後又漸像第一張圖畫,所以我們知道①到②(或②到③)的動畫構成一個單動作,在以下的敘述裏,本文假設一個動畫是針對一個單動作做處理,換言之每次呼叫動畫演算法,

僅是處理從單動作的第一張圖畫到最後一張圖畫的中間過程,而單動作的第一張和最後一張圖畫可看成是一個動作的二張原畫。

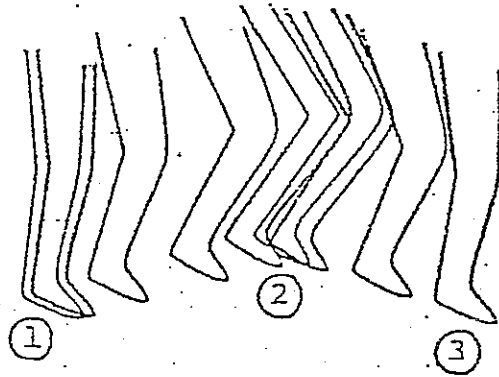


圖 1-1

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 Nested Block,若一動畫有一個單動作,則對應此動畫的程序有 n 個 Begin...End 的 Nested Blok 其中

End(i th Nested Block) = Begin($i+1$ th Nested Block), $i < n$

所以電腦動畫可以經由 Structured Programming 來製造,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。

從以上的敘述可知,動畫演算法的基本要素以能夠計算圖形間的相似性為主,而探討圖形相

似性的問題是圖形認知領域裏的一項重要研究,事實上已經有很多經驗法則方法(Heuristic Algorithm)存在[6],不過這些方法似乎尚未涉及用B軟標函數理論的探討,而POCA系統係採用B軟標曲線當作圖形的數學表示式,所以我們必需創造一套基于B軟標函數之圖形相似性的測度法,才能滿足動畫系統設計的要求,我們所採用的方法是利用模糊集合論[7]處理在集合系圖形間之模糊相似性的測度,採用模糊相似性測度的理由是:(1)B軟標曲線係對應控制多邊形,利用多邊形的邊長和夾角為變數不難訂出模糊集之會員函數(Membership Function),供測度圖形的相似性。(2)模糊集合可以處理模糊的語句,可以規劃成模糊語言,俾供藝術家容易於描述修整動畫的過程。(3)模糊集合論的數學簡單,容易規劃出有效的演算法,而且和人類直覺感官的判斷類似,可以設計出比較自然好用的互作圖形處理系統。

動畫系統主要以B軟標函數為基礎,再配以模糊集合論為基礎之相似性測度及模糊指令處理等演算法所構成,在觀念上,整個動畫系統的規劃和人造智慧論是息息相關的,人造智慧論[8]在圖形上的應用上涉及人們對圖形表徵的解釋,這種解釋可以利用系統的方法電腦化,這樣的工作對於圖形的認知有很大的助益,而且對於規劃文互式電腦圖形系統亦有幫助,譬如當輸入原畫時,如何利用圖形的表徵規劃文談式的圖形描述

語言,又如在動畫產生畫面的過程,如何利用動畫圖形的表徵規則時鐘語言等,就是根據對人造智慧的認識從事各項副系統的設計。

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) [9]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 軟探曲線間的文點進行,將是一項巨大的計算工程,因此採用求 B 軟探曲線間文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免了事先求文點的工作,

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其被遮住的黑線,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?但是動畫系統尚需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,讓塗色可以隨着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方法是很困難找到,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

來產生中間過程各個封閉區域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。藉着這樣的方法自動塗色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。

本篇論文將分成數個章節,第二章:描述 POCA 系統的硬體結構,開機使用的程序以及示範卡通製作的實況與結果,第三章提出原畫繪製的方法及其理論基礎,第四章說明動畫產生的原理,第五章說明着色方法的原理,第六章說明系統軟體結構及 POCA 語言的描述,第七章提出未來研究動畫理論有待解決的一些問題以及擴充目前系統的一些注意事項。

第二章 POCA 系統硬體結構與使用程序 反製作實況與結果

2-1 POCA 系統的硬體結構

POCA 系統的硬體結構如圖 2-1 所示,採用的儀器如下:

· 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 軟標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。

· TEKTRONIX 4054 高折度圖形終端機 ——

為一 1024×780 高折度的螢光顯示幕 (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, 不滿意的圖形, 亦在此機上經由互作式修改 (見靜態畫修改法), 直到滿意為止。

· 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。

· 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) 之互作式着色 (見着色方法), 並在此機的螢

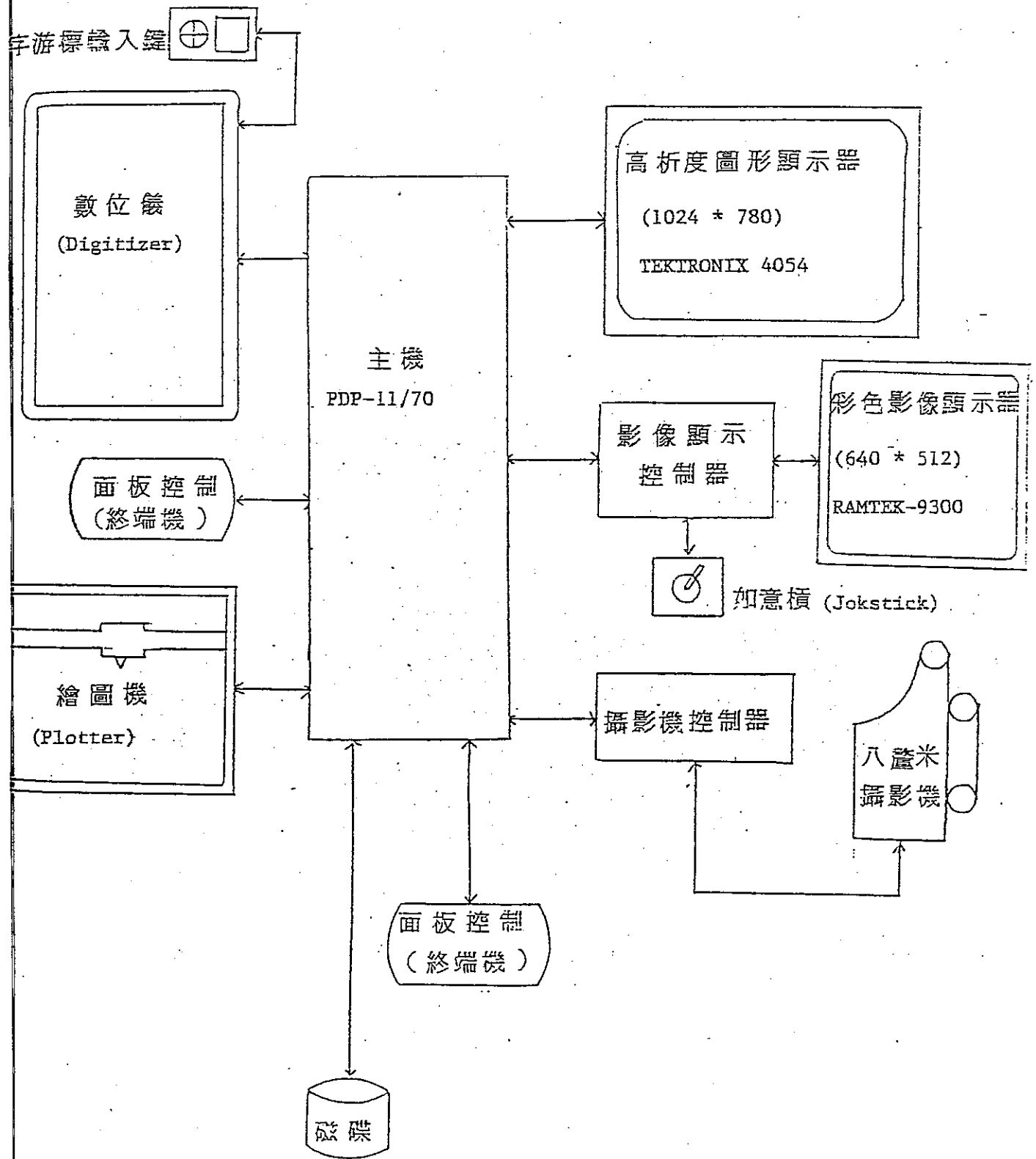


圖 2-1 POCA 系統硬體結構

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。

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,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2-2。

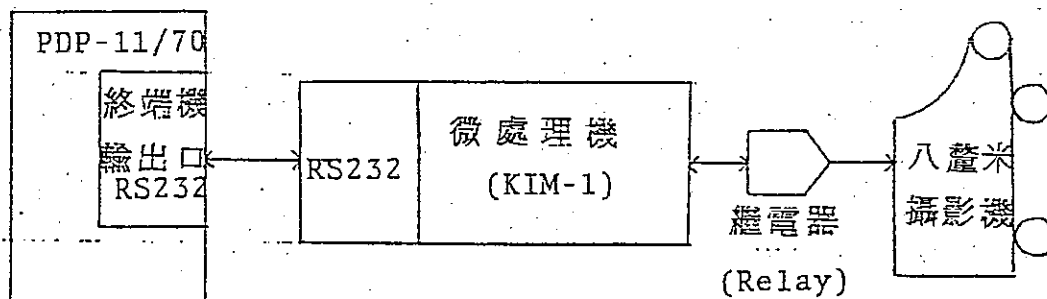


圖2-2 攝影機控制器

2-2 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系統的所有程式亦為六個工

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遞用，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, RAMTEK9300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 RETURN 鍵。

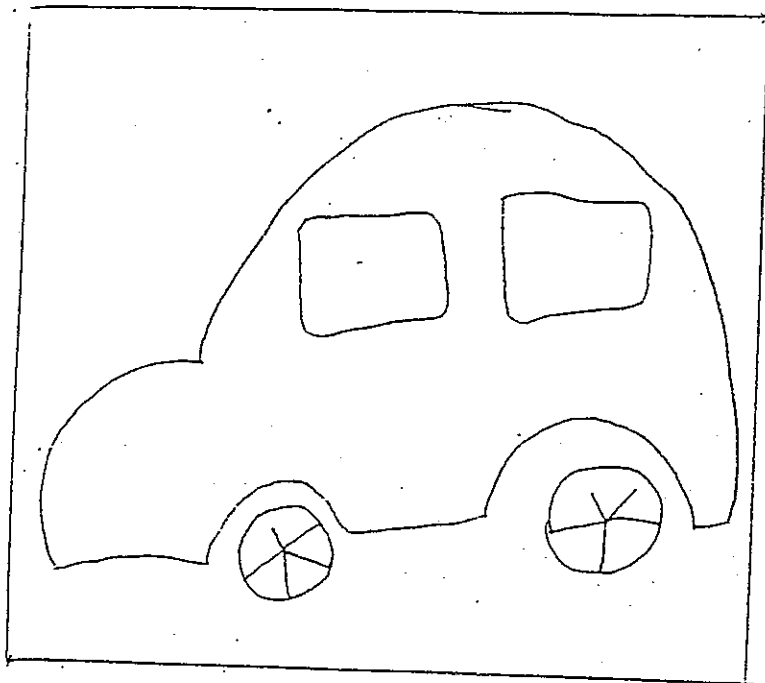
* 表 POCA 系統開始接受 POCAL 命令 (見 POCAL 命令語言的描述),此時鍵入 POCAL 命令,即可開始製作卡通圖畫。

2-3 製作實況及結果

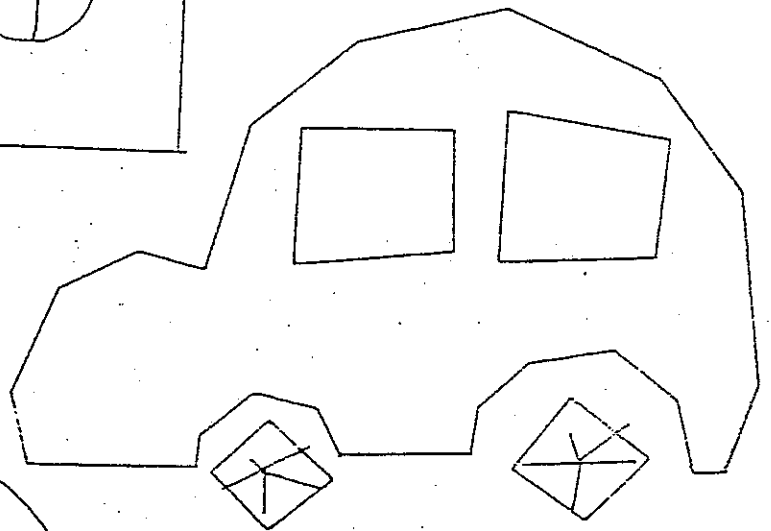
本節中示範從手稿到圖形着色後的靜畫

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

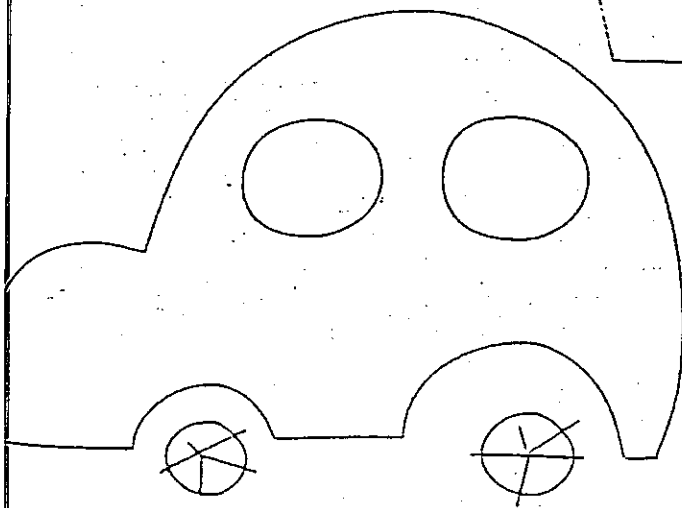
為測試 POCA 系統在電腦輔助卡通設計上之實用性,我們嘗試製作並拍攝了一部 4 分 20 秒的實驗影片,從原畫輸入到影片拍攝完成共耗 120 時/人,計有 15 幕 (見附錄) 每幕約為 10 幅左右基本圖形組成,輸入的基本圖形大約 150 幅,拍攝使用的圖形大約 550 幅,除輸入外餘為自動產生出的圖形。



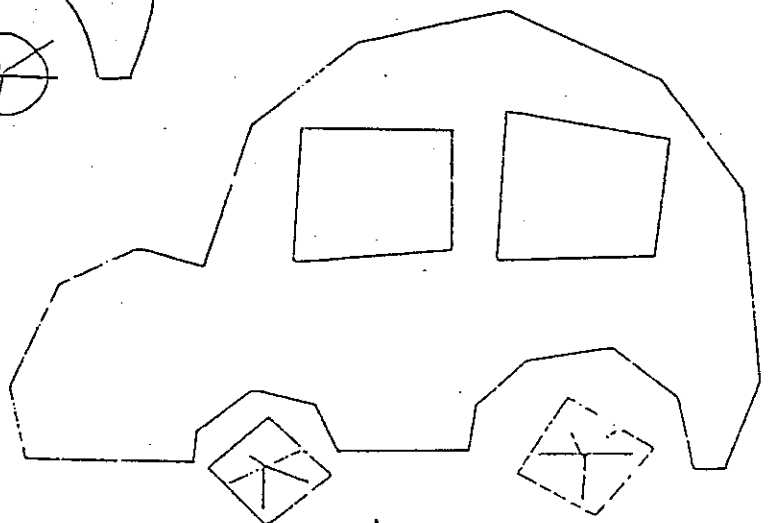
(a) 手稿



(b) 輸入圖形之控制多邊形

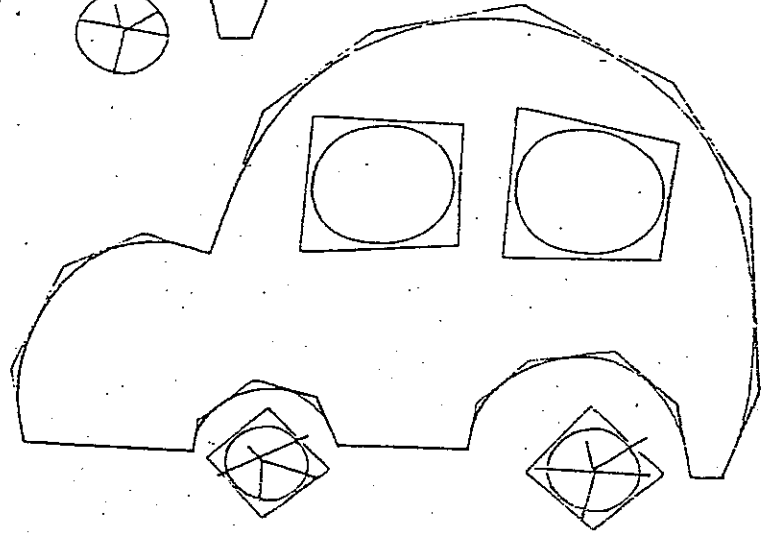
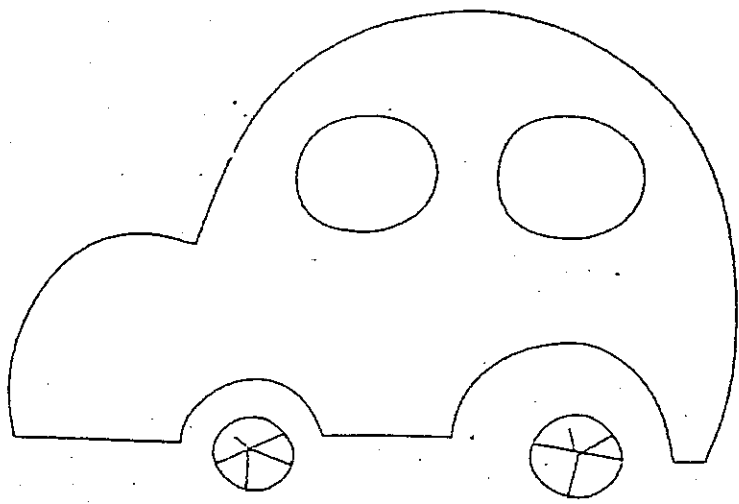


(c) 輸入圖形

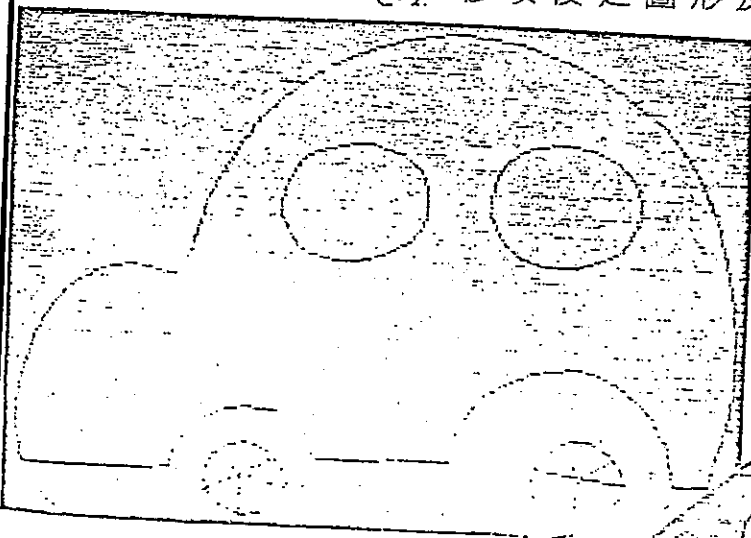


(d) 修改後之控制多邊形

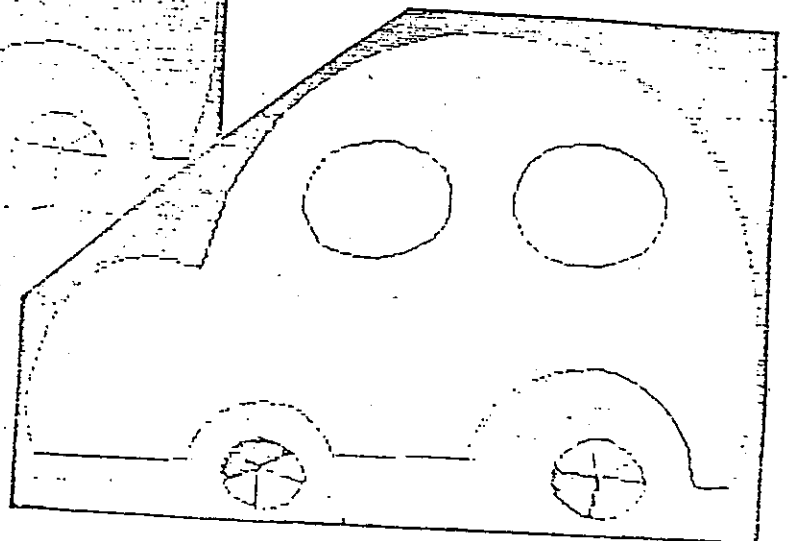
圖 2-3 製作實例(1)



(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 2-3 製作實例(2)

3.1 靜畫之理論基礎

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,這些方法在電腦動畫的應用上有二項主要的缺點:(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,特別在許多實際應用裏,曲線上的點座標不是已知的,典型的例子是寫生畫,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 [11],任一 Bézier 曲線的形狀是由一個素描多邊形所控制,因此,圖形可經由調整其素描控制點的位置獲得。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 [12] 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 曲線被限圍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,因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,唯一的方法就是採系 Beinstein 基底函數,而改用低次多項式,但必需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標函數(Spline Functions) 近似,同樣的道理,吾人亦需將軟標函數改頭換面,用參數表示的基底函數,用素描控制點當做展開式中的係數,讓我們稱這樣的曲線為 B 軟標曲線 [13], B 軟標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標線段的形狀。

靜畫製作之原理係採用均勻三次 B 軟標曲線之特性,有關 B 軟標函數近似的理論,請讀者詳閱 [4],本文僅將需要的公式及其幾何特性提出來討論,令

$x = (x_0, x_1, \dots, x_k)$ 為一實數的向量其中 $x_i \leq x_{i+1}$, 則

一個函數 P 稱為 $m-1$ 次軟標函數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.

令 $S(m, P)$ 表示所有 $m-1$ 次軟標函數所存在的 $m+k-1$ 次元空間, 則在所有可能當做的基底中, 我們有興趣的是 B 軟標基底. 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 軟標基底. 函數, 則對應素描多邊形 C 的 $m-1$ 次 B 軟標曲線是由下式決定

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

其中基底函數 $\phi_{i,m}(s)$ 可由下面的遞迴函數 (2) 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)

吾人不難證明 (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 Bézier 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

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

其中 $t = FR(ku) = ku$ 的分數部份

$i = INT(ku) = ku$ 的整數部份

$k =$ 總共的軟標段 $= n-3$

$u \in [0, 1]$

上式稱為"均勻三次 B 軟標函數", 它的向量值是由素描控制點 v_i 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

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

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 3-1 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 軟標曲線的形狀。例如從圖3-2 的素描控制點知 B 軟標曲線的起點會通過 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,如此繼續觀察各個連續的三角形,則可將整條 B 軟標曲線的形狀勾劃出來。

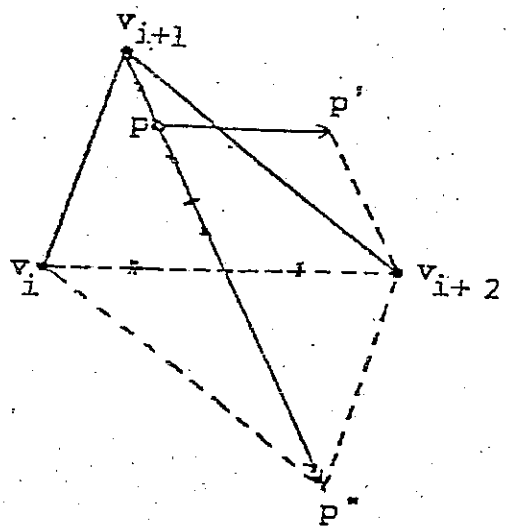


圖 3-1



圖 3-2

均勻 B 軟標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 軟標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,雖然從直

觀上言在尖點上曲線是非 C^1 連續,但從均勻 B 軟標表示式中,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 3-3 顯示。

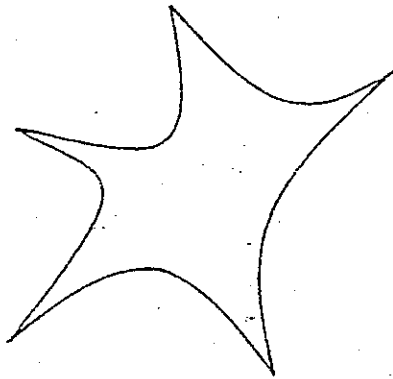


圖 3-3

3-2 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) 影像資料過於龐大,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 軟標函數來

計算原畫的近似曲線

係用 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

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

在顯示時計算 B 軟楔函數耗費較長的時間,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評解。

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 3-4(a)(b)。

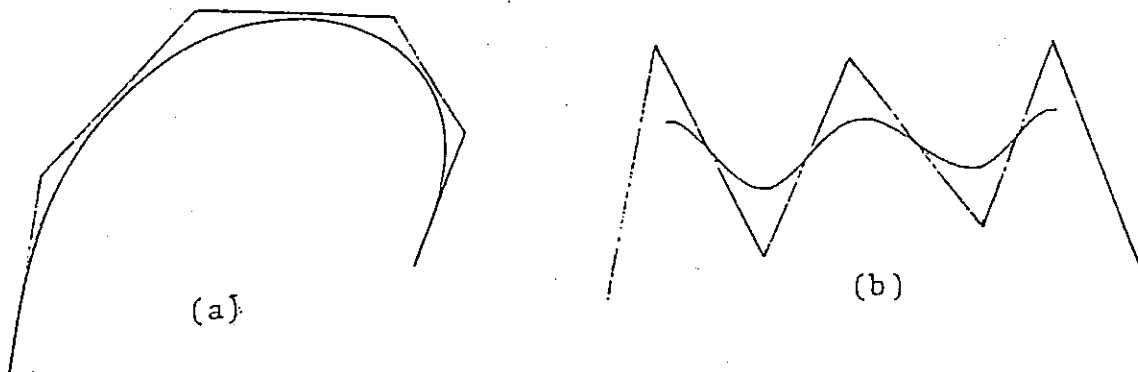


圖 3-4 B 軟楔近似曲線在凸多邊形內部

(2) 若三個連續的控制點重合,則近似曲線通過此重合點,所以欲劃一直線,則點入二組三個重合的控制點即可,如圖3-5。

(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3-6。



圖3-5 近似曲線通過三個重合控制點的重合點

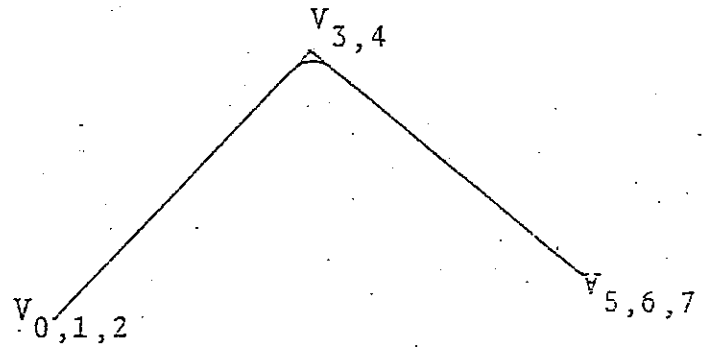


圖3-6 近似曲線非常接近二個重合控制點的重合點

(5) 任三個不重合的控制點,近似曲線通過其三控制點所形成的三角形的中線,且和中線的文點

到頂點的距離,恰為到底邊中點之半,如圖 3-7。

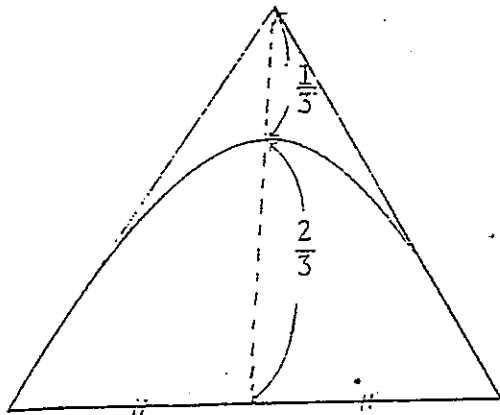


圖 3-7 通過三分之一點

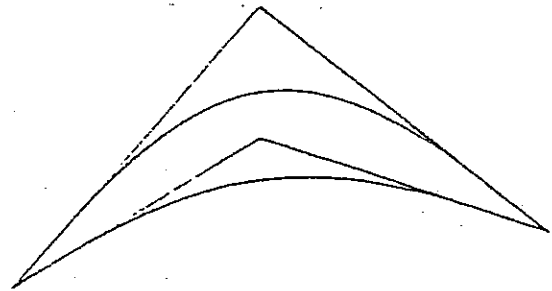


圖 3-8 適合局部修改

(6) 改變一控制點的位置只改變局部的近似曲線如圖 3-8。

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 3-9。

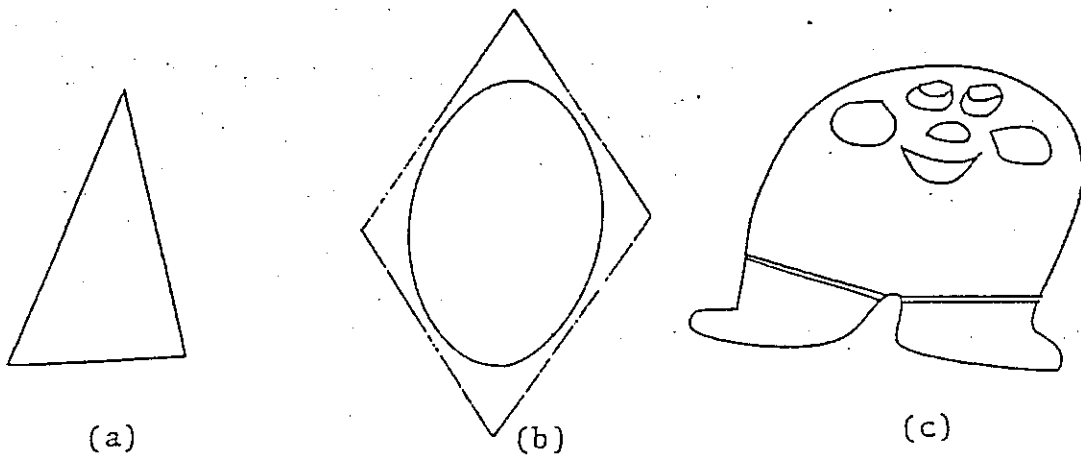


圖 3-9 根據上述五原則極易輸入圖形

為配合着色方法,在 POCA 系統中所有的原畫的每一部份均必須為封閉的,此限制並不對原畫影響極大(見附錄)。

第四章 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 $|c(s^i)| \neq |c(s^f)|$ ，因此為了決定它們之間的“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。為方便討論計令

$|c(s^i)| < |c(s^f)|$ 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，因為那樣會使 s^i 完全變形，唯一的方法是透過均勻 B 軟探反函數的過程來決定一個新的素描多邊形，它和 $c(s^f)$ 具有相同的素描點數，而且它決定的曲線形狀和 s^i 一致。

令 $|\bar{c}| = \max(|c(s^i)|, |c(s^f)|)$ ，則均勻 B 軟探反函數的過程包括二個步驟(以下我們僅討論封閉

B 軟探曲線的情形)：(一)在曲線 s^i 上尋找數據點羣，它們將 s^i 均分成 c 段，亦即找出 $s^i(t_k)$ ， $t_k = 0, 1/|\bar{c}|, 2/|\bar{c}|, \dots, 1 - 1/|\bar{c}|$ ，假設這些數據點分別是 $P_0(0)$ ，

$P_1(0), \dots, P_{q-1}(0)$, 其中 $q = |\bar{c}|$, 而 $P_0(0) = P_{q-1}(1)$, 則從 (4) 式知這些點滿足以下的線性系統方程式

$$\frac{1}{6} \begin{pmatrix} 4 & 1 & & & & 1 \\ & 1 & 4 & & & \\ & & & \ddots & & \\ & & & & \ddots & \\ & & & & & 1 & 4 & 1 \\ 1 & & & & & & & 1 & 4 \end{pmatrix} \begin{pmatrix} v_0 \\ v_1 \\ \vdots \\ \vdots \\ \vdots \\ v_{q-2} \\ v_{q-1} \end{pmatrix} = \begin{pmatrix} P_0(0) \\ P_1(0) \\ \vdots \\ \vdots \\ \vdots \\ P_{q-2}(0) \\ P_{q-1}(0) \end{pmatrix} \quad (5)$$

(二) 從 (5) 式解得新的素描點羣 $\{v_i\}, i = 0, 1, \dots, q-1$, 令此素描多邊形為 $c(s^i)$, 則在直觀上 \bar{s}^i 的曲線形狀和 s^i 是“非常”相似, 因為它是從近似法 14 (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, 將面積如何決定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 $c(s^f)$ 內, 我們 $c(\tilde{s}^f)$ 用表示平移 $c(s^f)$ 後的素描多邊形, 於是 $c(s^i)$ 和 $c(\tilde{s}^f)$ 素描點的配對是藉着匹配 $c(s^i)$ 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)。

令要素描點 v_k 是 $C(S^i)$ 中最具影響形狀的控制點,換言之,刪除它會使曲線形狀產生最大的變化者(或產生最不相似的形狀),我們稱要素描點 v_k 是“媒婆”(Match-Maker),一種尋找媒婆的簡易法是這樣的:若 $C(S^i)$ 有尖點,則取其中一點當媒婆,若無,則取對應反曲點的控制點當媒婆,若仍無反曲點,則任取一點當媒婆,一旦在 $C(S^i)$ 決定那點當媒婆後,就需在 $C(\tilde{S}^f)$ 找到媒公,媒公的找法如下:令 $v_j \in C(S^i), \bar{v}_k \in C(\tilde{S}^f), j, k = 0, 1, 2, \dots, m$, 則媒公 \bar{v}_ℓ 滿足

$$T(\bar{v}_k, \bar{v}_\ell) = \min_{0 < k < m} \{ T(v_h, \bar{v}_k) \}$$

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 $C(\tilde{S}^f)$ 的要素描點),再用 $\{\bar{v}_h\}$ 的平分角線和這些要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 $(v_{h+1}, \bar{v}_{\ell+1}), (v_{h+2}, \bar{v}_{\ell+2}), \dots, (v_{h-1}, \bar{v}_{\ell-1})$, 就完成了匹配的工作圖4-1示出紅心和鑽石

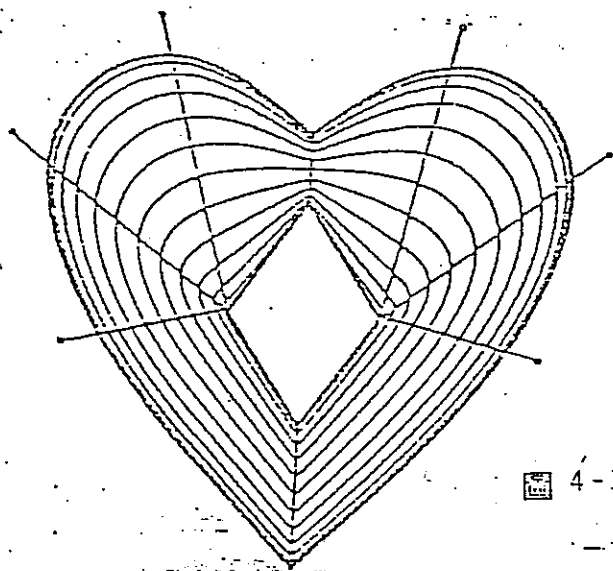


圖 4-1

的匹配。

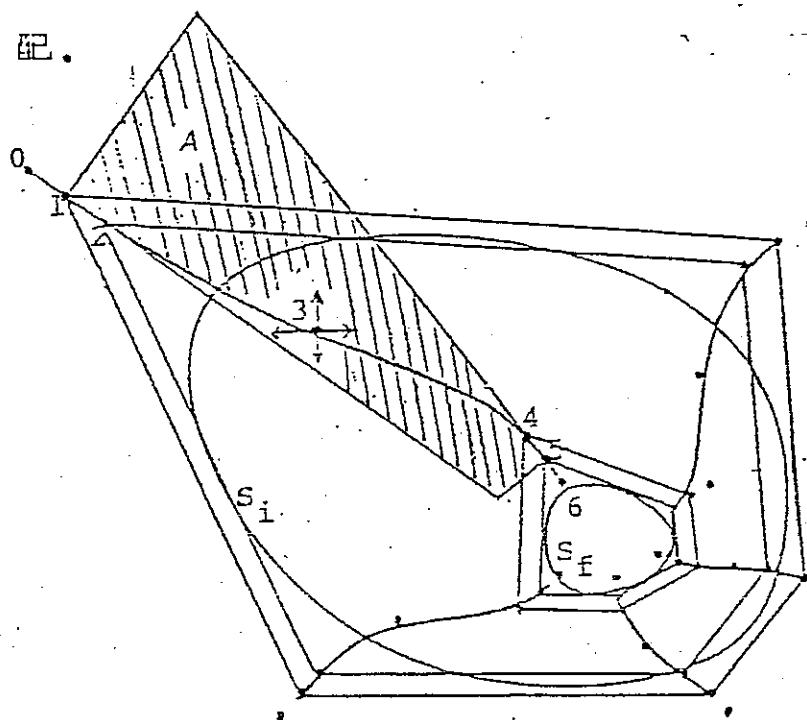


圖 4-2

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4-2。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”(ABit) 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，我們採用了如圖上所示均勻 B 軟標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 軟標控制點，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,它的選取需經過特別的考慮。爲了避免相鄰兩個B軟標曲線疊在一起而產生不自然(扭曲)的動畫(如圖4-3,第3點的選取必需限制在某一個範圍,從B軟標曲線的特性,知,這個範圍就如圖4-2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,再作微調。因此動畫演算法的第四項工作就是要提供怎樣調整第3點位置的步驟程式,如圖4-4所示爲動畫產生的一個實例。

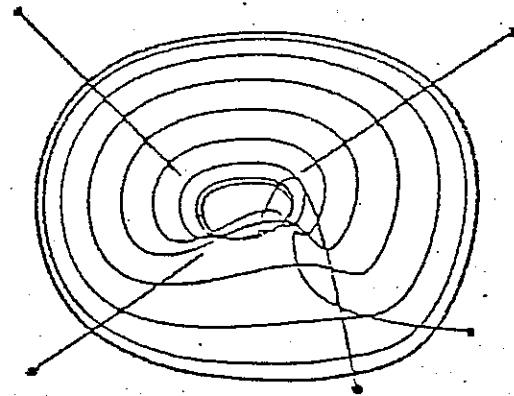
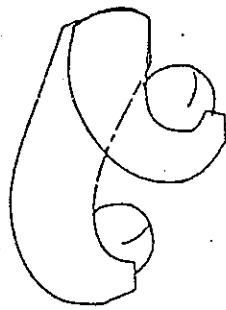
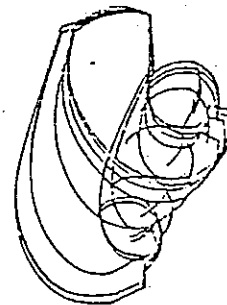


圖 4-3



(a) 第一及最後一張靜畫



(b) 產生的中間動畫

圖 4-4 動畫產生實例

5-1 着色使用法

卡通製作的系統是以能夠提供諸大、活潑、有趣的卡通畫為主旨，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作一張畫紙，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5-1。

(3) 使用POCAL者以互作式方式，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（稱為顏色種子—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。

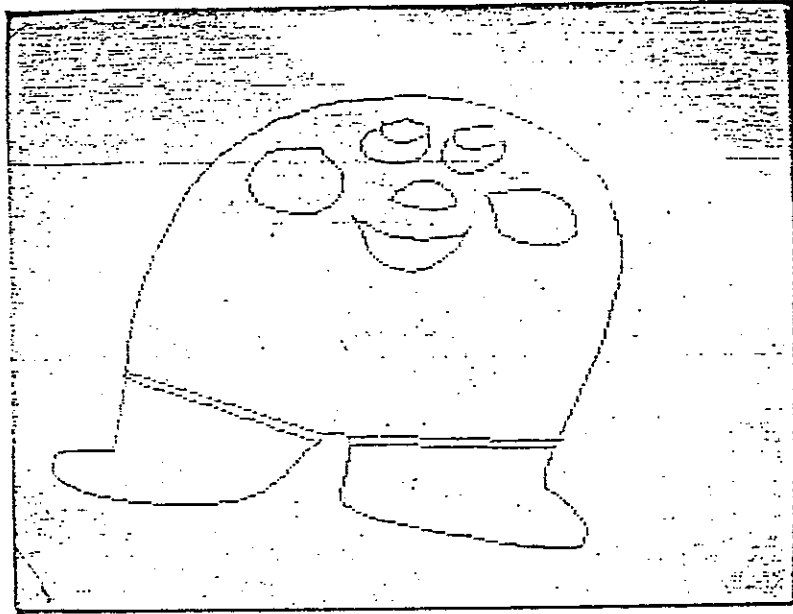


圖 5 — 1

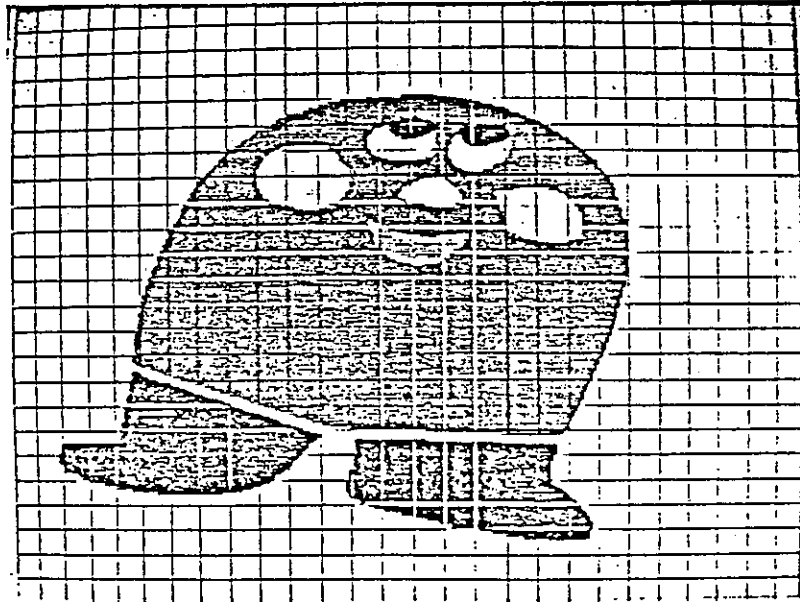


圖 5 — 2

(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 5-2 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堅實且漂亮。

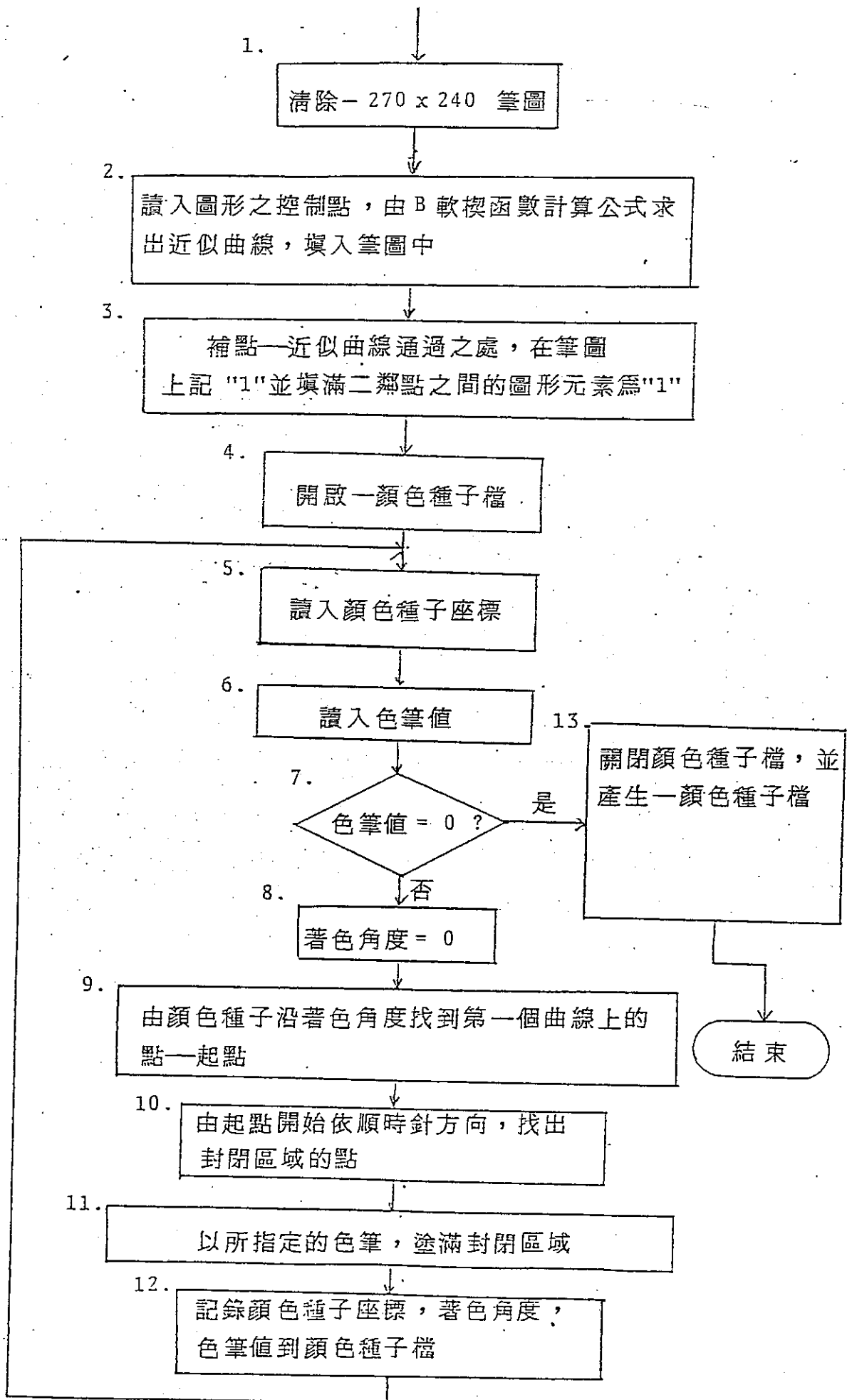
5-2 着色演算法 (Painting Algorithm)

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. 為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為 0。

2.3. 由 B-軟探函數計算公式算出近似曲線的軟探段, 在筆圖中可能為不連接的二點, 所以任二鄰點之間的圖形元素, 也必須記入 "1" 表示近似曲線通過這些圖形元素, 以構成一封閉區域。這項處理



著色法流程图

稱為補點,其情形如圖5-3所示。

			x_4	0	0	x_3	0	0	0	x_6										
		0									0									
		x_5									x_7									
		0									0									
	0										0									
	x_2										x_8									
	0										0									
	0		0	0	x_{11}	0					x_9									
	x_1	0	0				0	0												
								x_{10}												

x : 計算出的近似軟楔段端點
0: 補入點

圖 5-3 補點一在筆圖中記錄近似曲線通過之處

4. 選立一顏色種子檔,記錄顏色種子的座標位置,着色方向的角度及色筆值。
- 5.6.以互動式方式,讀入顏色種子座標及色筆值。
7. 當色筆值等於零,就表示着色已完畢。
8. 因為指定一個封閉區域是靠種子座標的位置來指定的,種子座標在封閉區域之中,若不規定一尋找封閉區域的方向,則會產生混淆不清,不知所指的現象,找到的封閉區域非為想要的封閉區

據,如圖5-4中。若顏色種子位於A點,而A沿正X軸

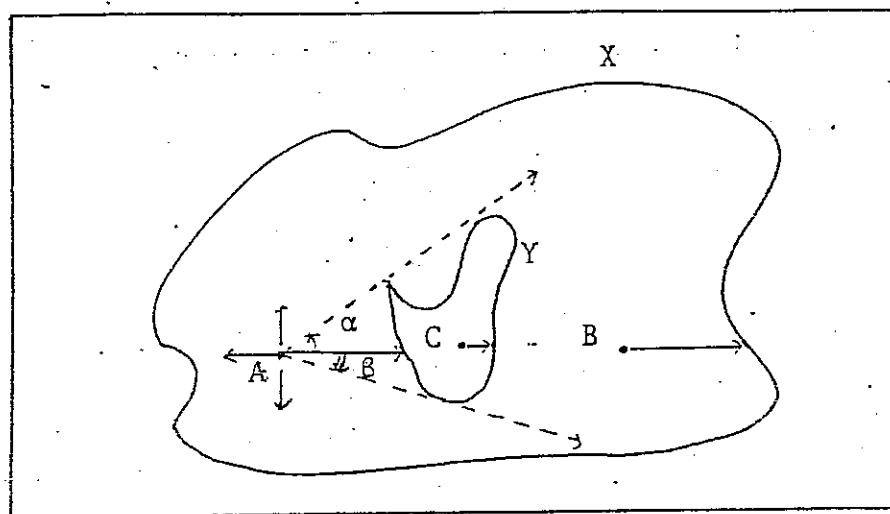


圖 5-4 規定著色角度為零

方向尋找則找到的封閉區域為Y;若沿負X軸或正,負Y軸方向,則找到的區域為X,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 5-5 所示。

0	0	0
\emptyset	x	0
0	0	0

x : 筆圖中任一點
 $\emptyset, 0$: 可能的鄰點

圖 5-5 筆圖中任一點的可能鄰點

又因為着色角度為 0, 所以沿著着色角度找到區域上的一點(3x3筆圖的中心點)後, 從 \emptyset (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 3x3 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 5-6

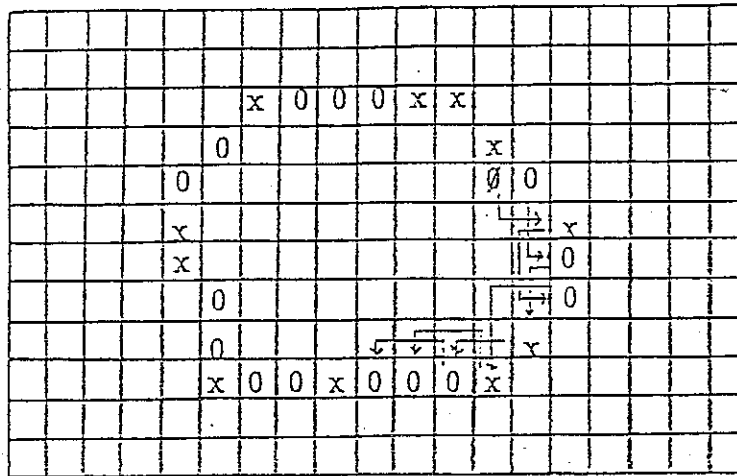


圖 5-6 封閉區域的找法

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 Mix x 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) 依次以色筆劃滿步驟 (6) 中所求得的小區域, (漸增加 y 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 座標值的大小分成幾個區間,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 5-7。 V_1, V_2, V_3, V_4 為多邊形的四個頂

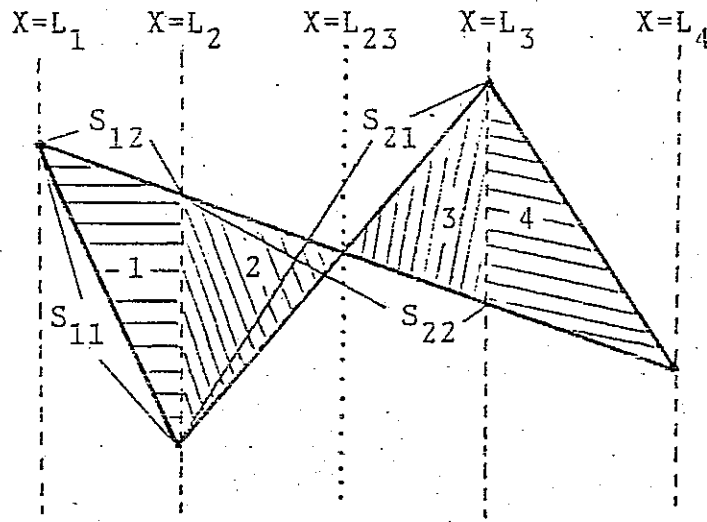


圖 5-7 把一個區間再分割為二

點, V_1 及 V_2 的 X 座標分別為 L_1 及 L_2 , 第一次找到 $\text{Min}X = L_1, N\text{Min}X = L_2$ 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 $\text{Min}X = L_2, N\text{Min}X = L_3$ 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 $N\text{Min}X = L_{23}$ 回到步驟(4)繼續進行演算法的處理依圖 5-7 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。

6-1 POCA軟體系統結構

POCA 軟體系統結構如圖 6-1 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 RSX-11M 操作系統下,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,把工作程式連成樹狀,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,(2)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技巧。

整個系統分為 6 個工作程式

工作程式 1 : MAIN

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

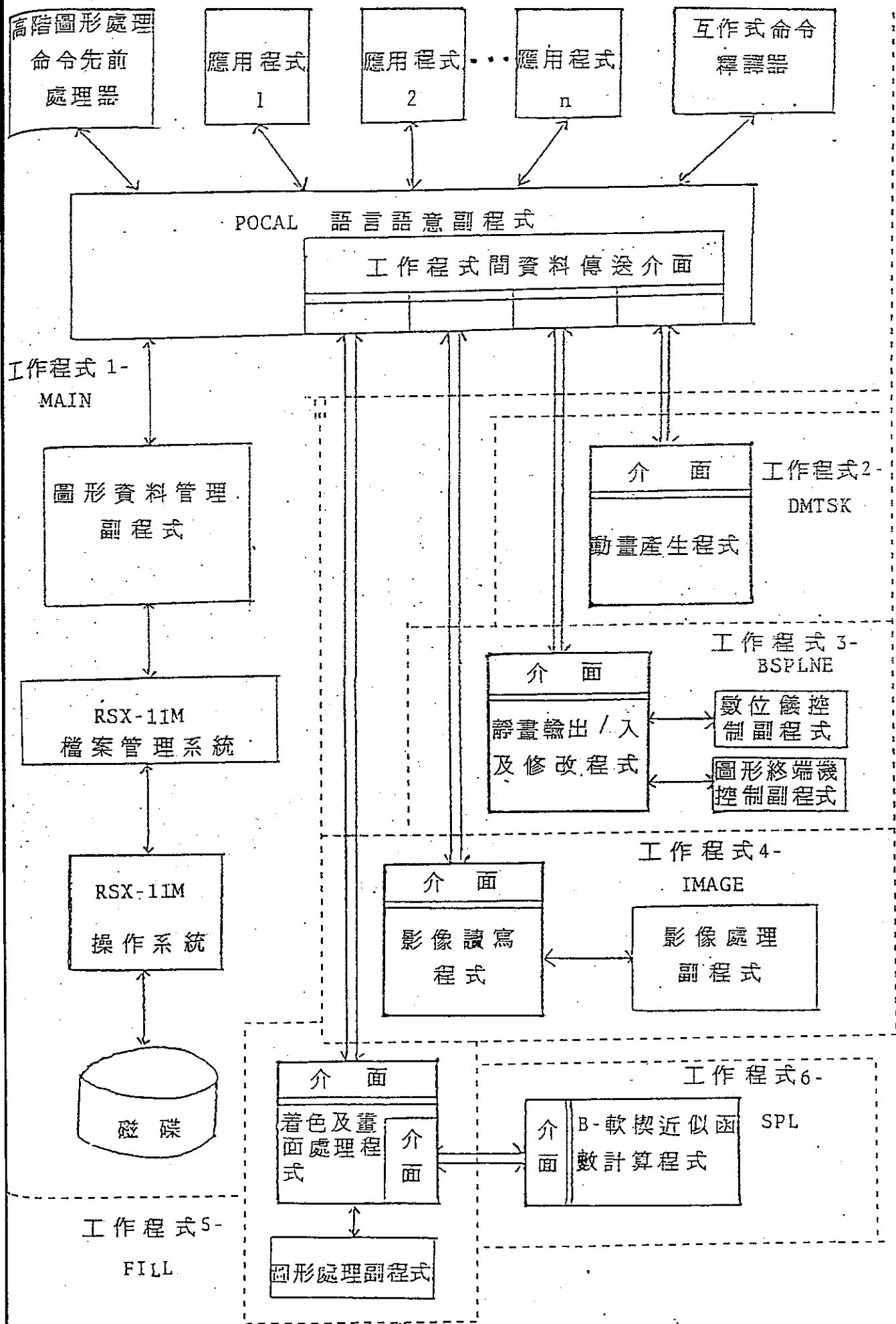


圖 6-1 POCA 軟體系統結構

們的主僕(Master-Slave)關係如圖 6-2

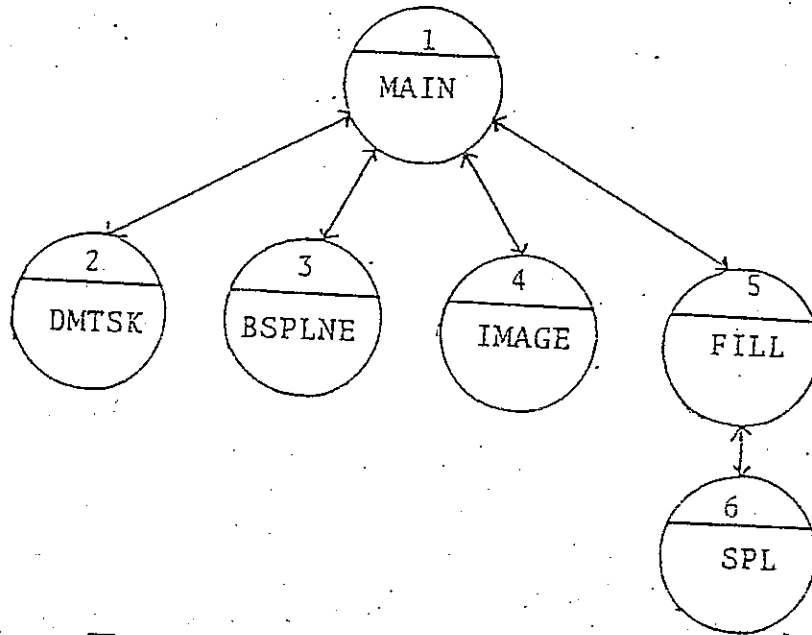


圖 6-2 POCA 系統中六個工作程式之主僕關係

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 軟標近似函數的計算, 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 6-1 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 6-1 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),因涉及從數位儀輸入資料及從高折度顯示器 (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)[15]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(荷)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着色,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及畫圖軟體模組(FIPP-FORTRAN Image & Plot Package) [16] 中之一部分。

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 的檔案管理系统(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,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6-2。

(3) 交互式命令釋譯器: 為使使用者直接以交互式方式使用POCA 系統,亦有一交互式命令釋譯器直接接受文談式命令作輔助設計的工作,交互式命令見表6-3。

- . 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>
- . CONCAT <name> = <name> + <name> { + <name> + ... }
- . SUBTRACT <name> = <name> - <name> { - name - ... }
- . DYNAMIC-MOTION <name>, <name> STEP $\begin{pmatrix} \langle \text{int} \rangle \\ \langle \text{var} \rangle \end{pmatrix}$ INTO <name>
- . COLOR <name>
- . $\begin{pmatrix} \text{TRASLATE-REL} \\ \text{TRASLATE-ABB} \end{pmatrix}$ <name> $\begin{pmatrix} (\langle \text{int} \rangle, \langle \text{int} \rangle) \\ (\langle \text{var} \rangle, \langle \text{var} \rangle) \end{pmatrix}$ INTO <name>
- . ROTATE-CG <name> <degree> { INTO <name> }
- . ROTATE-PT <name> TO $\begin{pmatrix} (\langle \text{int} \rangle, \langle \text{int} \rangle) \\ (\langle \text{var} \rangle, \langle \text{var} \rangle) \end{pmatrix}$ <degree> { INTO <name> }
- . MIRROR <name> $\begin{pmatrix} (\langle \text{int} \rangle, \langle \text{int} \rangle) \\ (\langle \text{var} \rangle, \langle \text{var} \rangle) \end{pmatrix}$, $\begin{pmatrix} (\langle \text{int} \rangle, \langle \text{int} \rangle) \\ (\langle \text{var} \rangle, \langle \text{var} \rangle) \end{pmatrix}$ { INTO <name> }
- . ZOOM-CG <name> BY <x-factor>
- . ZOOM-PT <name> TO $\begin{pmatrix} (\langle \text{int} \rangle, \langle \text{int} \rangle) \\ (\langle \text{var} \rangle, \langle \text{var} \rangle) \end{pmatrix}$ <x-factor>, <y-factor> { INTO <name> }
- . IMAGE-READ FROM $\begin{pmatrix} (\langle \text{int} \rangle, \langle \text{int} \rangle) \\ (\langle \text{var} \rangle, \langle \text{var} \rangle) \end{pmatrix}$, $\begin{pmatrix} (\langle \text{int} \rangle, \langle \text{int} \rangle) \\ (\langle \text{var} \rangle, \langle \text{var} \rangle) \end{pmatrix}$
- . IMAGE-WRITE

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

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

中 [] 表選擇其中之一，{ } 表可選或不選。
<name> 為 6 個字母之圖形名稱，字母為 {A-Z,0-9} 之一。
<var> 為變數名稱，依 FORTRAN 語言的定義。
<int> 為整數。
<degree>, <color> 可為 <var> 或 <int>
<x-factor> , <y-factor> 為實數。

表 6-2 高階圖形命令語言語法


```

### DEVICE I (D)-DIGITIZER * (T)-TERMINIX * (R)-FAHTEK
$IH I INPUT PICTURE (D)
$KCS I READ CURSOR LOCATION OF JOYSTICK BY ENTER SWITCH 0 (R)
$HFIO I MODIFY PICTURE (T)
$DIS I DISPLAY PICTURE (R)
$DISCO I DISPLAY CENTER OF GRAVITY OF PICTURE (R)
$DRAW I DRAW PICTURE (R)
$COORD I DISPLAY X,Y COORDINATE (R)
$ERASE I ERASE SCREEN (R)
$COPY I COPY PICTURE I OBJ = SOURCE J
$CANGC I CANCEL PICTURE
$CONCAT I CONCATENATE TWO PICTURE INTO ONE I OBJ = BACKGROUND + FOREGROUND J
$SUSTR I SUBTRACT SEGMENT FROM PICTURE I OBJ = PIC - SEG J
$DH I DYNAMIC MOTION
$COLOR I COLORING PICTURE (R)
$TRARL I TRANSLATE PICTURE RELATIVELY
$TRAD I TRANSLATE PICTURE ABSOLUTELY
$ROTCD I ROTATE ABOUT CENTER OF GRAVITY
$ROTF I ROTATE ABOUT ONE POINT
$MIRR I MIRROR PICTURE * MIRROR LINE IS SPECIFIED BY TWO POINTS
$ZOOHX I ZOOM PICTURE ALONG X-AXIS J
$ZOOHY I ZOOM PICTURE ALONG Y-AXIS J RELATIVE TO A SPECIFIED POINT
$ZOOHD I ZOOM PICTURE ALONG (X,Y)-AXIS J
$ZOOGX I SAME AS ZOOHX J
$ZOOGY I SAME AS ZOOHY J
$ZOOGD I SAME AS ZOOHD J RELATIVE TO CENTER OF GRAVITY
$EXIT I EXIT
$HELP I PRINT THIS DOCUMENT
$IMR I IMAGE READ (R)
$IMW I IMAGE WRITE (R)
$IMWR I IMAGE WRITE BY RELOCATION.
$CAMER I CAMERA

```

表 6-3 交互式命令

2 圖形管理

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 6-3

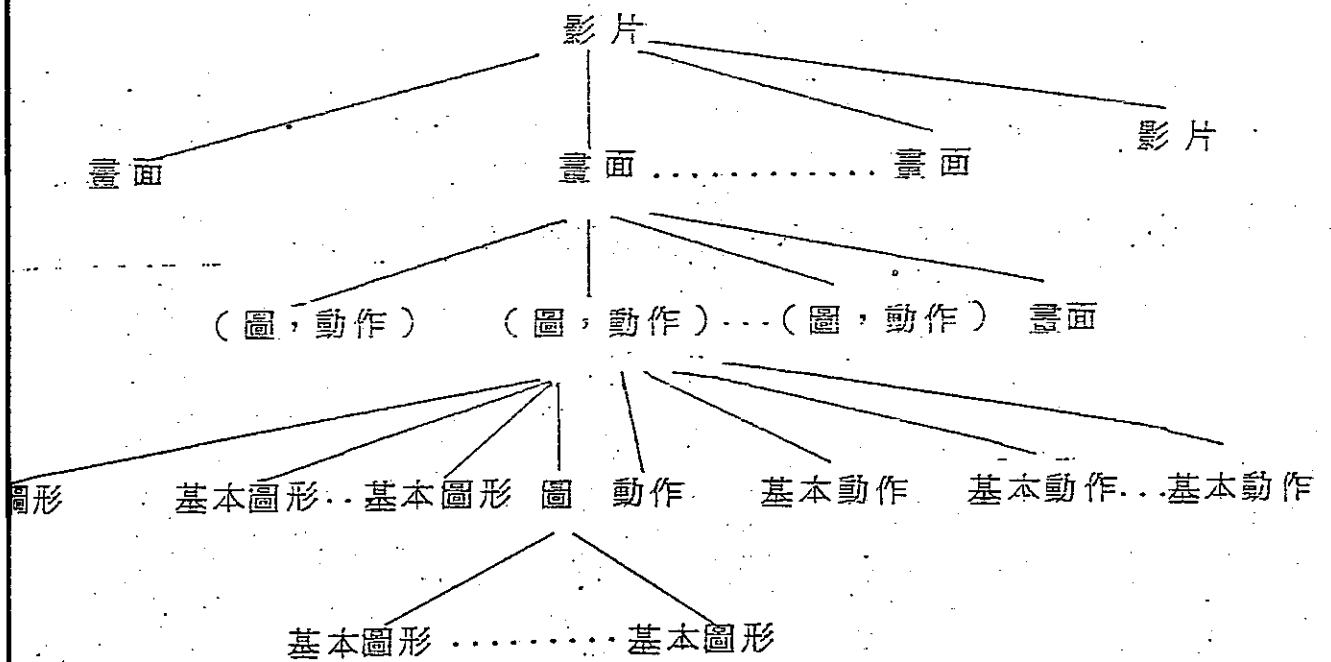


圖 6-3 影片的組成

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 6-3 的動作去掉，得到圖形的關係如圖 6-4。

我們可以得到以下的關係式：

影片 + 畫面 | 影片

畫面 + 畫 | 畫面

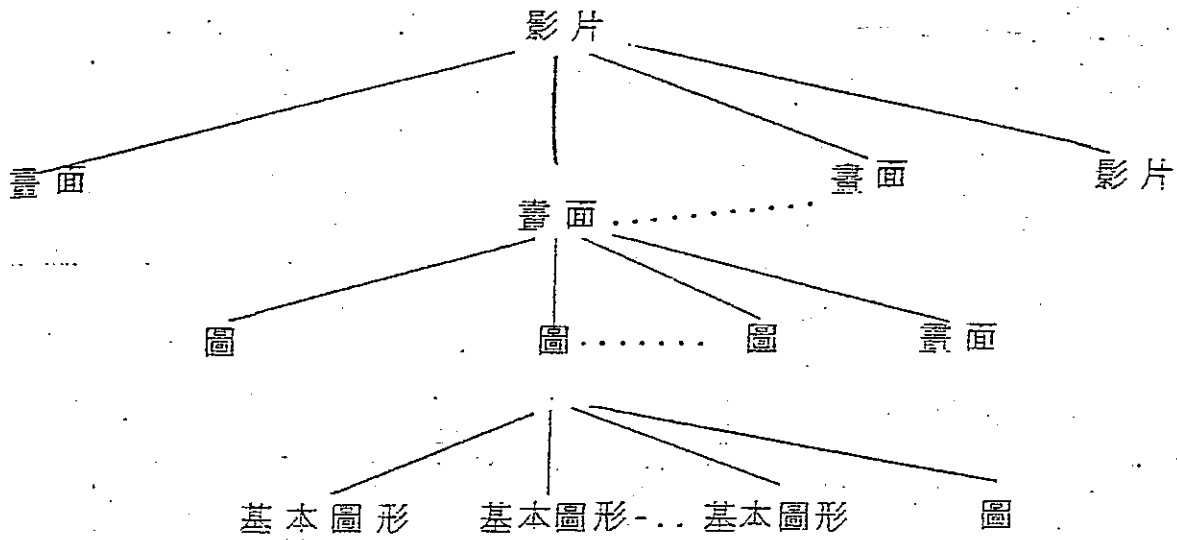


圖 6-4 影片和圖的關係

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6-4化簡成圖6-5就得到

圖 + 圖 | 基本圖形

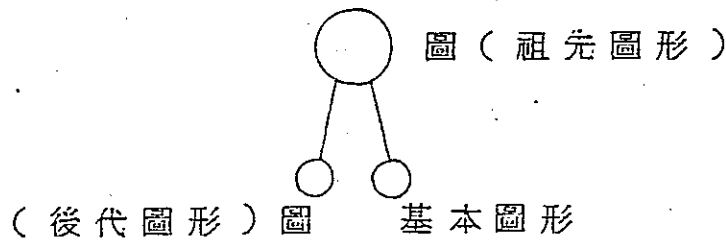
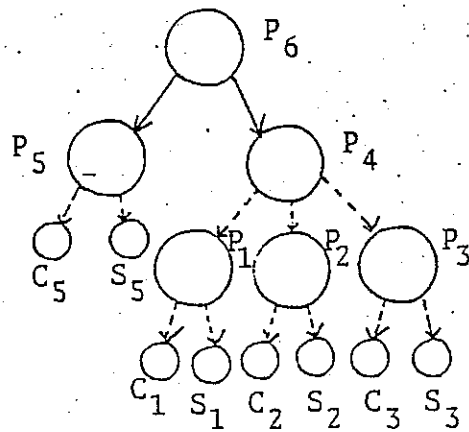
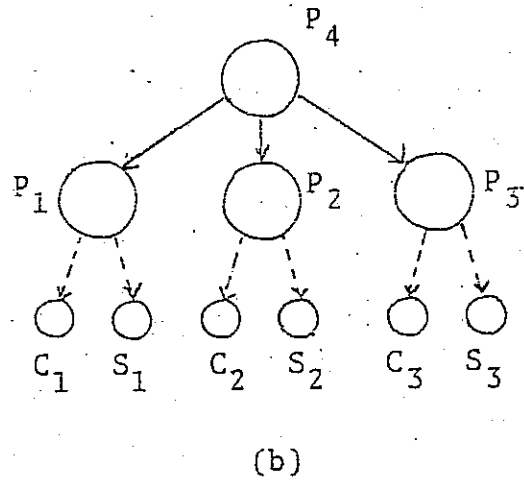
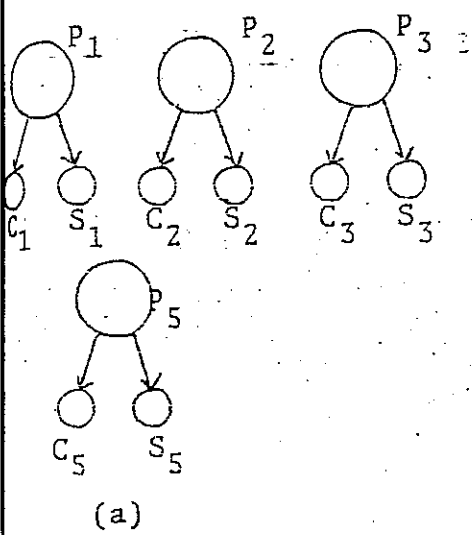
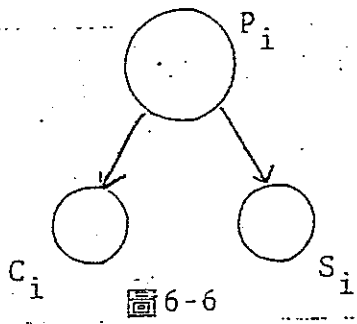


圖 6 - 5

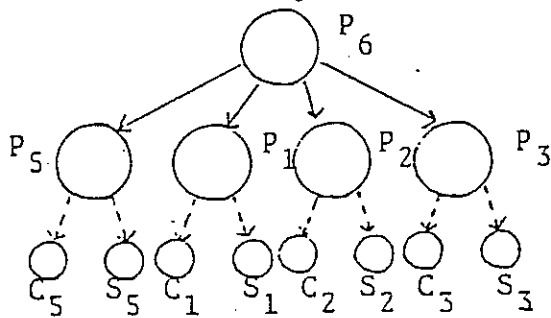
在 POCA 系統中,基本圖形就是我們處理的最小單元,而任何圖可由任意基本圖形或任意圖組成,所以在系統中我們要記錄的就只有二件事了。(1)基本圖形為何(2)圖是由那些圖或基本圖形組成的。同時我們定義,組成一個圖形的圖或基本圖形為所組成圖形的後代圖形,所組成的圖形則稱祖先圖形。每一個圖形或基本圖形我們記錄的方式是這樣的,每一個基本圖形有三個檔分別為(1)總檔(head file),(2)組成此圖的控制多邊形的所有控制點檔,(3)顏色種子檔。總檔中記錄 ① 控制點檔檔宗名稱,② 所有後代圖形的名稱,③ 是否已着色,④ 着色角度,⑤ 種子顏色檔檔宗名稱,⑥ 此圖形的祖先圖形數目。若以 P_i 表圖形, 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 6-6 所示。

任一非為基本圖形的圖可能為圖 6-7 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 6-4,其中筆分類見 6-3.1。



(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

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)	祖先圖形值 數 (I3)

總 檔

X 座標 (I3)	Y 座標 (I3)	筆分類 (I1)

X 座標 (I3)	Y 座標 (I3)	色筆值 (I3)

控制點檔

種子顏色檔

表 6-4 圖形資料表

6-3 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

6-3.1

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:

圖形輸入屬交互式,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 軟探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),點

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文叉點對準控制點,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"。

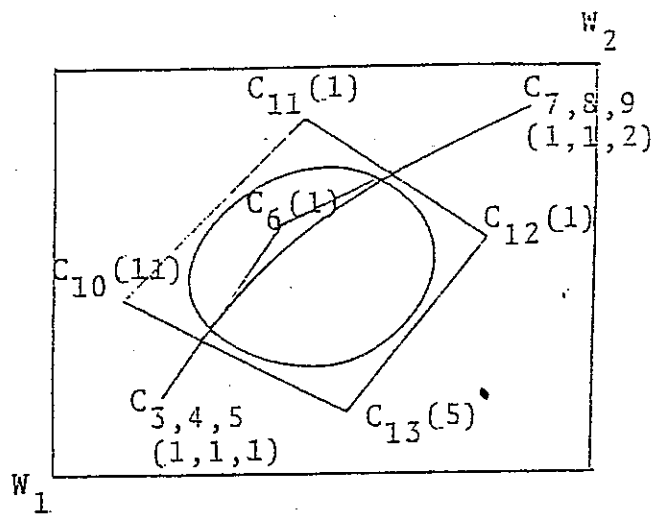
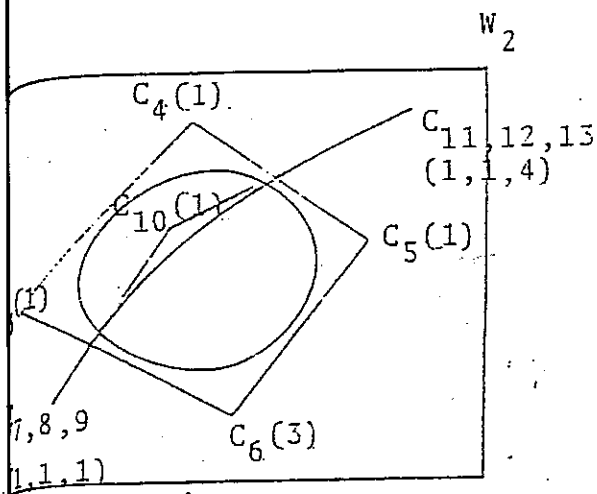
其他點均按"1"。

如表6-5,圖6-8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 6-5 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 6-8 輸入範例

(2) 加點

系統先在高折度顯示器 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

系統先在高折度顯示器上顯出圖形及控制點,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

系統在高折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

在繪圖機上或高折度顯示器上,系統打出

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

若為 T,則在高折度顯示器上畫出圖,若為

P 則再顯出

SCALING FACTOR : ?

? 表示放大倍數, 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。

6-3.2

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。

6-3.3

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6-3.1)相同,除不具I指令功能外。

6-3.4

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 為欲顯示點的X,Y座標,因為一點座標在顯示幕看起來太小,所以顯示出的點是以IX,IY為中心的一個正菱形。

6-3.5

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。

6-3.6

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 (Clipping)。

6-3.7

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。

6-3.8

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。

6-3.9

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。

色。

6-3.10

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。

6-3.11

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。

6-3.12

名稱：CONCATE 合併

(Concatenate Two Picture Into ONE)

功能:合併二個小圖成一個大圖。

呼叫格式：

CALL MCNCTE(OBJ,SOURL, 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。

6-3.13

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,SOURL,SOUR2)

說明：

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

6.3-14

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,
則產生的動畫名稱為
xxxxxx001, xxxxxx002, ... xxxxxx*i*
其中 FNAME 和 xxxxxx001 相同
LNAME 和 xxxxxx*i* 相同。

6-3-15

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要着色的封閉區
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,則系統依着色角度
為 0, 按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, 系統並在圖形總擋中記入着色角
度為 0, 着色狀態為 'Y' 並產生顏色種子擋。

6-3.16

名稱: TRANSLATE Relatively (相對平移)。

功能:

相對於原來的位置平移一個圖形。

呼叫格式:

說明：

1. OBJ可以與 SOUR 相同,表示移後圖形名稱不變。

OBJ亦可與 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。

6-3.17

名稱：TRANSLATE ABSOLUTELY(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 6-3.16之說明。

6-3.18

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 6-3.16 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 的餘數)。

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 6.3-18 之說明。

6.3-20

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 之餘數。
3. 見 6-3.16 說明 1。

6.3-21

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 X 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6-3.16 說明 1.

6-3.22

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

功能：

對指定點沿 Y 軸放大。

呼叫格式：

CALL MZOOMY (OBJ,SOUR,IX,IYYFACT)

說明：

見 6-3.21 說明。

6-3.23

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

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 6.3-21 說明。

6-3.24

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

功能：

沿 X 軸對重心放大。

呼叫格式：

CALL ZMCGX (OBJ,SOUR,XFACT)

說明：

見6-3.21 說明。

6-3.25

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6.3-21 說明。

6-3.26

名稱：ZOOM a picture along X & Y Axis Relative to Center of Gravity
(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6.3-21 說明。

6-3.27

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。

6-3.28

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 JOYSTICK

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,鏗入'ENTER'鏗。

系統再顯出

INPUT RIGHT LOWER CORNER OF WINDOW FROM JOYSTICK

使用者以如意桿移動十字游標至窗函數右下角,並按'ENTER'鏗入。

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[16]。

6-3.29

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)。

6-3.30

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：

依照影像原貯存之圖函數範圍,在指定位置顯像, IX, IY表示新的圖函數之左上角座標。

本篇論文討論了設計一套電腦輔助設計系統所遇到的一連串問題,這些問題大致上有:

- (1)圖形的輸入及表示法
- (2)圖形的動畫產生法
- (3)圖形的着色法
- (4)圖形的貯存法
- (5)文談式的卡通設計方法

本文提出上述各項問題的解決方法,雖然現在的系統程式已相當龐大,需用多工作程式技巧建立起整個系統,但爲了提高系統工作效能,以下幾項功能應再予加入系統裏:

(1)在動畫產生法中,第四項工作"提供怎樣調整第三點位置"的方法,或許可以勝駭"邏輯論的觀點來加以調整。

(2)圖形輸入經由使用者輸入圖形的控制多邊形而完成,太多的交互式降低圖形輸入效率。若能以圖形掃描器,讀入原圖再利利用影像處理的方法加以處理,再反求圖形的控制多邊形,則可提高圖形輸入效率且交互式着色方法亦可借助圖形掃描器,讀入位於封閉區域中的種子顏色而提高着色速度。

(3)目前所用的彩色顯示器為RAMTEK 9300 原為影像顯示用,其能顯示的圖形元素極高(640×512),且顏色的層次(16×16×16=4096)亦極多,使顯像的工作太過於複雜而且降低了顯像速度。但就卡通圖形而言,顏色約只須80種左右及要求的圖形元素不需如此的高,同時在系統中着色及顯像均以封閉區域為單位,若此封閉區域的着色工作以硬體線路來操作更能提高顯像速度,故重新設計一適合於卡通圖形顯像器是提高系統工作效率之一方法。

(4)由於卡通製作的過程是順序的(Sequential)且對一張圖的處理是有一定的次序,依輸入着色,圖形變化或產生,及着色,對於同一張圖形的處理有一定的順序,又因為微處理機的價格越來越低所以若能把系統的各種功能分佈到一個微處理機上執行,而只把複雜的運算工作交由主機來處理則可提高卡通製作的效率。

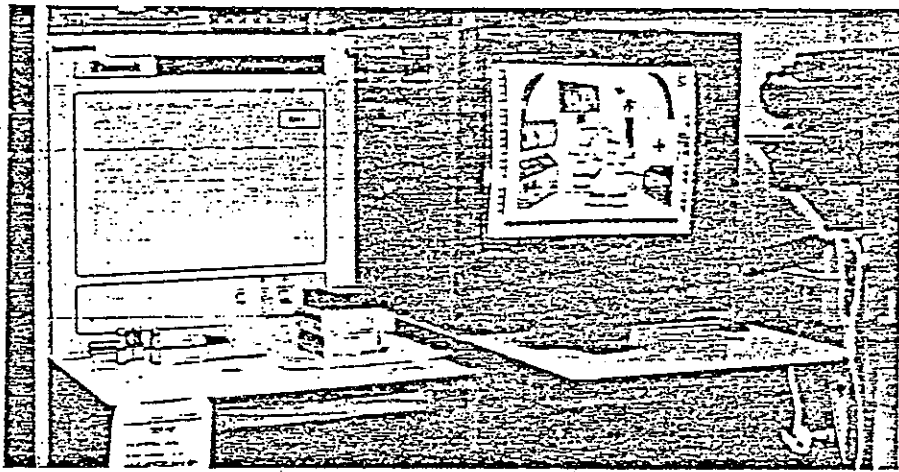
参 考 文 献

- ng, K. Y., "A B-spline based computer animation system," of the Chinese Inst. of Eng., pp.37-45, Jan. 1981.
- nyk, N., and Wein, M., "Interactive skeleton techniques enhancing motion dynamics in key frame animation," J. ACM, pp.564-569, Oct. 1976.
- ves, W. T., "Inbetweening for computer animation utilizing key point constraints," Proc. ACM Siggraph, Vol.15, No.3, pp.263-269, August 1981.
- ns, S. A., "Surface patches and B-spline curves," in Computer Aided Geometric Design, Barnhill and Riesenfeld (Eds.), Academic Press, (1974).
- , J. M. and Risenfeld, R. F., "A theoretical development of the computer generation and display of piecewise polynomial surfaces," IEEE Trans. PAMI pp.35-46, Jan. 1980.
- iesca, E. and Guzman, E. "how to describe pure form and measure differences in shapes using shape numbers," Pattern Recognition, Vol.12, pp.101-112, 1980.
- h, 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.
- g, K. Y. and Chen, C. N., "Computer Animation with Artificial Intelligence," Proc. of National Computer Symposium, pp.327, 1981.
- i, K.W., "Interactive Computer Graphics — Data Structures, Algorithms, Languages," Prentice-Hall, Inc., Englewood Cliffs, N.J., pp.154-158, 1978.
- 11 RSX-11M/M-PLUS EXECUTIVE Reference Manual.
- er, P. "Numerical Control -- Mathematics and Applications," Wiley, New York, 1972.
- est, A. R. "Interpolation and approximation by Bézier Polynomials," Computer J., vol. 15, No. 1, pp.71-79, 1972.
- er, C. "On calculating with B-splines," J. Approx. Theory, pp.50-62, 1972.
- , K. Y. "On Fuzzy similarity relation of B-spline curve," Proc. of International Computer Symposium, Taipei, pp.1292 - 1293, 1980.

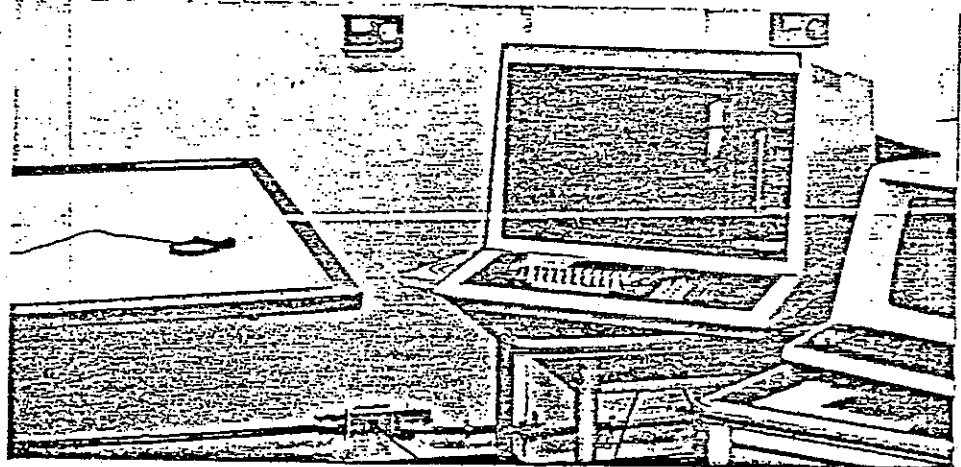
15. TEKTRONIX Terminal Control System Reference Manual.

16. RAMTEK FORTRAN Image and Plot Package User's Guide.

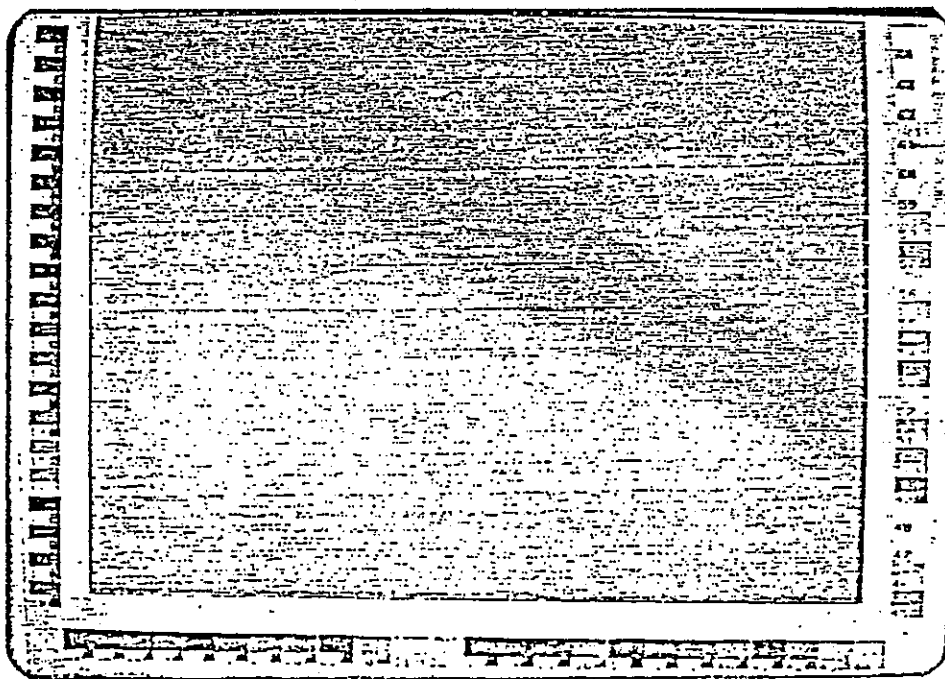
附 錄



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

