TR--88--033

一個動畫路徑控制系統之設計

國 立 中 興 大 學

碩 士 論 文

題目： 一個動畫路徑控制系統之設計
(Design of An Animation Path Control System)

所別：應用數學研究所 組別：資訊組

研究生： 陳 石 雄

指導教授： 鄭 國 揚 博 士

中 華 民 國 七十八 年 五 月 日

國立中興大學

應用數學研究所第七屆畢業生畢業論文

題　目：一個動畫路徑控制系統之設計

姓　名：陳石雄　學號：757727

成　績：捌拾伍分

論文考試委員：

楊熙年

石煇然

鄭國揚

中華民國　　　年　　　月　　　日

目　錄

誌　謝

本論文之得以順利完成，應該完全歸功於指導教授鄭國揚博士的諄諄教誨，使我在電腦動畫及計算機圖學的領域裡能有更深一層的瞭解。老師爲人雖然幽默風趣，但治學態度卻非常嚴謹，使我在讀研究所的這段期間，無論是在求學或是爲人處世方面都學得許多，這些將成爲我以後爲社會服務，奉獻自己的最大本能。

感謝所裡資訊組的每一位老師，由於他們的殷勤教誨，使我在資訊課程方面穫益良多。同時也要謝謝中央研究院資訊科學研究所圖書館的職員們，在我查詢資料時給予最大的幫忙。

在此要以最誠摯的敬意感謝撫我、育我的父母親，沒有他們愛的支持與鼓勵，就不可能有今天的我。還有兄姐們全家人的體諒與照顧，才能讓我安心求學，在此同表謝忱。最後，要感謝女友文燕這些日子在精神上的鼓舞與照顧，使我得以專心於論文寫作，謹以此文獻給所有愛護我的師長與親友們。

摘　要

本文提出一種以物體導向(Object-Oriented)爲主之動畫路徑控制系統，簡稱爲A.P.C.S.(Animation Path Control System)。所謂”物體導向”，顧名思義，其設計顯然需針對不同的使用者，而因應出一套不同的東西來。”物體”(Objects)是由”資料”(Data)及”程序”(Procedures)所組合而成，藉著”程序”來做運算，而後將結果存放於”資料”內。其作用完全是由物體間互相”傳遞訊息”(Sending Messages)而達成。

物體導向此名詞首次是由Smalltalk 程式語言致力研究所發表出來的，Smalltalk 並非一般傳統的程式語言，它是一種能充分反應出物體導向設計哲學的程式設計環境。在此要强調的是本系統並非使用Smalltalk 程式語言來設計，而是藉用其物體導向之觀念來發展。在A.P.C.S.裡，所有的物體都能在適當的激勵下有所反應，並採取適當的處理。因此A.P.C.S.提供了一個友善地(friendly)環境給使用者，畫家們能夠很輕易的對螢幕上所顯現的路徑做修改。只要有一個參數變動，路徑便立刻隨之更改而顯示於螢幕，如此反覆下去，直到使用者滿意爲止。

在本篇論文中，我們採用K-B軟樣函數(即Kochanek-Bartels Spline)爲A.P.C.S.系統內控制路徑的數學模式。因爲K-B軟樣函數提供三個參數:(一)張力性(tension)，(二)連續性(continuity)及(三)偏壓性(bias)來調整軟樣本身的幾何形狀及變化狀態，A.P.C.S.本系統以此爲基礎，另外提供一個參數:加速度(accerlation)用以控制顯現中間畫面的速度快慢，畫家藉著這四個參數值的調整，便能調出所要的動畫路徑。

當A.P.C.S.接收到畫家所輸入的路徑資料(即上面所述的四個參數值)後，便利用曲面調適(Surface-Fitting)的方法來決定其它圖形控制點的路徑，最後便可得到各個中間畫面。在本文最後列出A.P.C.S.此系統程式的函數功能及如何使用A.P.C.S.簡介，並舉出一些例子來測試A.P.C.S.系統之可行性。

第一章　緒論

動畫(Animation)，是一門學問，更是一種藝術。其應用範圍非常之廣，無論是商業上或教育訓練上，皆有其價值存在。最著名的例子便是美國華德、狄斯耐公司所創造的一系列卡通影片，更是將動畫技巧發揮得淋漓盡致。其中的人物造型，諸如米老鼠、唐老鴨……等幾乎是無人不知，無人不曉，由此可知動畫的魅力所在！

由於傳統的動畫必須靠畫家將所有的圖案以手工一張一張的畫出，既耗時又費力。爲了改進此缺點，便有電腦動畫(Computer Animation)的產生。一般而言，電腦動畫可括分成兩大類：(一)Key-frame animation及(二)Modelled animation[1]。前者利用電腦輔助計算並產生中間畫面(In-betweens)，並將物體沿著某一路徑(Path)移動，而產生動畫效果。後者則著重畫面於三度立體空間之表達。本文中所提之動畫均只針對Key-frame animation而言，並不考慮Modelled animation。

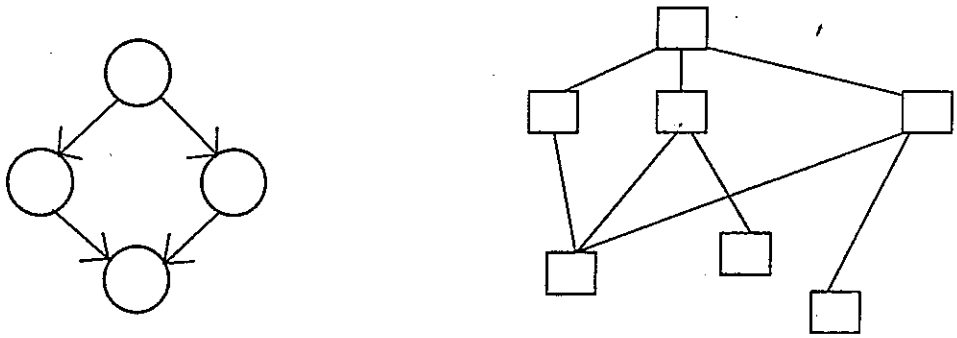
Key-frame animation 的原理非常簡單，由畫家提供首尾兩張主畫面(Key-frame)，並提供物體運動的路徑，藉此路徑之描述而產生許多連續性的中間畫面。將這些中間畫面依照順序存起來，最後以20張/秒到40張/秒(通常是24張/秒)之速度顯示(Display)之。由於人類視覺有約1/24秒的緩衝時間，便有動畫的效果產生。

由上文中可知，控制路徑即相當於控制中間畫面的產生，也就是控制動畫的過程。因此，如何提供一個友善(friendly)及良好的系統供畫家能夠很容易的去調整動畫之路徑，便是本文所欲達成的目標。傳統的結構化程式雖然有許多優點，但將之用於交談式(Interactive)繪圖系統時，使用者必須先將一大堆參數(Parameters)資料丟給電腦，經過運算之後才顯示結果出來。這

對於本身對電腦不是很熟悉的使用者而言，是一件痛苦的差事。因爲他不曉得要記住輸入那一些資料值才能滿足程式所需，也不知道什麼時候電腦才能顯示圖形。有鑒於此，吾人乃考慮利用”物體導向程式(Object－Oriented Programming，以下簡稱爲O.O.P.)”的觀念－物體導向，來設計一套交談式動畫路徑之控制系統，以便改善上述缺點！

所謂”物體導向”，顧名思義，其設計顯然需針對不同的使用者，而因應出一套不同的東西來[8]。依照M. Stefik & D. G. Bobrow[4]所述，所有的O.O.P.中唯一共有的特徵便是”物體”(Objects)。”物體”是由”資料”(Data)及”程序”(Procedures)所組合而成，藉著”程序”來處理某些計算，而後將結果存放於”資料”內。O.O.P.的作用完全是由物體間互相”傳遞訊息”(Sending Messages)而達成的。訊息之傳遞證明了程式中的一個重要原則：資料抽象化(Data Abstraction)。資料抽象化是指一種機構，它簡化了一個資料型態的表現方式及操作運算，它定義了一個封閉領域[6]。此原則說明呼叫者程式(Calling Programs)不需要知道它們所使用的資料型態(Data Types)其內部是如何表達或是如何產生的。它的目的便是程式獨立性的養成，不希望程式的任何一組件需要視其它部份內部的細節而調整。因爲O.O.P.乃藉著訊息的傳遞來溝通各個物體，而每個物體的內部根本就不公開，所以更無法參考到它的內部了。

接著要探討的觀念是O.O.P.內繼承性(Inheritance)的性質。當一些物體有共同的屬性時，它們就被分到同一個等級(Class)。其資料結構屬於格狀結構(Lattice Structure)，如圖(1. 1)所示。”上級”(Superclass)相當於”父輩”，”次級”(Subclass)相當於”子輩”，它們能經由繼承性的性質享有共同的屬性。格子結構可以同時擁有好幾個”上級”或”次級”，也就是”多對多”的關係。



圖(1. 1) 格狀結構

有了這種好處，更能加强對程式的維護。即使要做某種程度的修改時，只要著眼於修改之處就可以了，而不必擔心那些剩餘不動的部份會造成什麼影響。再者，使用O.O.P.對程式本身來說，由於許多物體可同時處理，所以可以"並行處理"(Concurrent)。若用於晶片(chip)上，速度會更快，這也是我們採用物體導向觀念的主要原因所在。

本文以下章節之安排如下：第二章敘述動畫路徑相關之數學基礎，第三章描述路徑之控制方法，第四章爲A.P.C.S.系統架構，第五章爲實例驗證，最後一章是結論。

第二章　路徑之數學基礎

由前文可知，路徑在動畫中扮演了極重要的角色。因此本章將專門討論採用何種曲線爲動畫路徑，期使達成最佳效果。一般而言，數學上的軟樸(Spline)函數是一分段的 k次多項式，在兩分段的共同連接點上，其導數的k-1 階具連續性。因此三次軟樸函數在連接點上有二階的連續性。對形成通過一系列點的曲線言，低次多項式的分段軟樸函數通常較有用，因爲可以簡化計算也可以減少數值的不穩定性(該情形在高階曲線時會產生)。然而，因爲低次多項式不能跨越任意系列的點，需要鄰接較多的多項式片段。基於這些考慮與物理上軟樸的情況，一般的方法是用一系列的三次軟樸函數，每一段函數只跨越兩點。此外，三次軟樸函數是允許有反曲點與在空間扭曲的最低次空間曲線 [9]。因此我們決定採用三次曲線。

本章有關章節部份的安排如下：第一節介紹Hermite Spline及B-Spline的數學模式，第二節介紹Kochanek-Bartels Spline的數學模式，第三節則介紹三維電腦繪圖的數學基礎。

第一節　Hermite Spline 及 B-Spline的數學模式

(A)Hermite Spline

Hermite Spline是一種內插(Interpolation)方法的三次軟樸函數，在每一個曲線線段(Segment)部分，其兩端點的一次導數(即切線向量)已知，加上兩端點的座標植，由這些邊界條件(Boundary condition)，便可求得曲線方程式。以下爲

Hermite Spline的數學定義：

$$P(u)=au^3+bu^2+cu+d$$

$$=[u^3\ u^2\ u\ 1][a\ b\ c\ d]^T \quad 0\leqq u\leqq 1 \quad \cdots\cdots(i)$$

$$P'(u)=[3u^2\ 2u\ 1\ 0][a\ b\ c\ d]^T$$

則其邊界條件爲

$$P(0) = d$$

$$P(1) = a+b+c+d$$

$$P'(0) = c$$

$$P'(1) = 3a+2b+c$$

以矩陣型式表示，則爲

$$\begin{bmatrix} P(0) \\ P(1) \\ P'(0) \\ P'(1) \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 3 & 2 & 1 & 0 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix}$$

$$\Rightarrow \quad [a\ b\ c\ d]^T = M\ [P(0)\ P(1)\ P'(0)\ P'(1)]^T$$

式中 $M = \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$ ……(ii)

∴(i)式中 $P(u)$可以矩陣表示爲：

$$P(u)= [u^3 \ u^2 \ u \ 1] \ M \ [P(0) \ P(1) \ P'(0) \ P'(1)]^T \quad \cdots\cdots(iii).$$

M = Hermite matrix

將(ii)中之 M值代入(iii)，可得

$$P(u) = P(0)F_1(u)+P(1)F_2(u)+P'(0)F_3(u)+P'(1)F_4(u)$$

其中 $F_1(u) = 2u^3-3u^2+1$

$F_2(u) = -2u^3+3u^2$

$F_3(u) = u^3-2u^2+u$

$F_4(u) = u^3-u^2$

$F_i(i=1,2,3,4)$ 稱爲Hermite加權函數(Blending functions)

Hermite Spline由於內插的特性，保證能通過每個控制點(Control Point)。

(B)B-Spline

不同於內插方法的Hermite Spline，B-Spline是一種近似方法(Approximation method)。該法是以定義多邊形之頂點來產生曲線，需選擇一組基底或加權函數。B-Spline基底在不改變定義多邊形頂點數目之情況下，可改變曲線之階數，進而改變曲線之形狀[5]。

令P(t)爲沿著曲線的位置向量，以參數t之函數表示，則B-Spline之數學定義爲：

$$P(t) = \sum_{i=0}^{n} P_i \, N_{i,k}(t)$$

P_i是n+1個定義多邊形(Control Polygon)之頂點，

$N_{i,k}(t)$是第i個B-Spline基底，k 階曲線之加權函數。

$N_{i,k}(t)$由下列遞迴公式定義：

$$N_{i,1}(t) = \begin{cases} 1, & \text{若 } x_i \leqq t \leqq x_{i+1} \\ 0, & \text{其它情況} \end{cases}$$

$$N_{i,k}(t) = \frac{(t-x_i)N_{i,k-1}(t)}{x_{i+k-1}-x_i} + \frac{(x_{i+k}-t)N_{i+1,k-1}(t)}{x_{i+k}-x_{i+1}}$$

x_i是一節向量 (knot vector) 之元素，參數t沿著曲線P(t)從0變動至tmax。

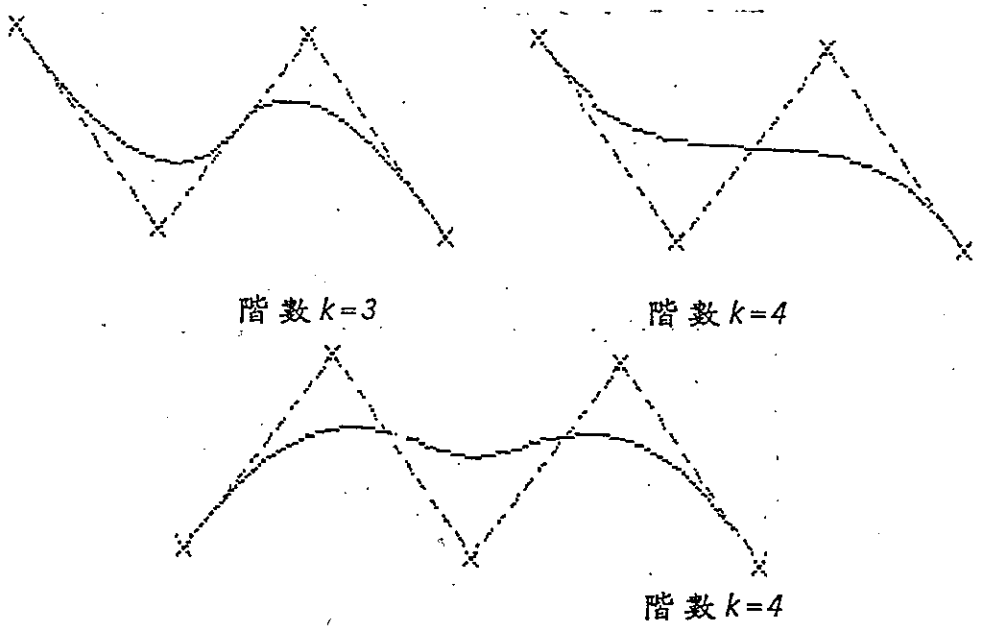
一節向量只是由一系列的實數x_i所組成，對所有的x_i，$x_i \leqq x_{i+1}$。這些x_i之值視爲參數的節點。產生B-Spline線段$(0 \leqq t \leqq tmax)$時，它們可用來指示參數 t之範圍。

B-Spline函數在數學上定義爲一k階(k-1)次的多項式函數，它滿足下列兩條件：

(1)在每一區間$(x_i \leqq t \leqq x_{i+1})$，P(t)是一(k-1)次的多項式。

(2)在整條曲線上，P(t)與其1,2,.....,k-2次導數皆連續。

有了上述特性，B-Spline可以不同的控制方式改變曲線的形狀。這些控制爲，改變整數階數$k(2 \leqq k \leqq n+1)$，用重複的頂點，改變不重複頂點的數目與位置。這些效果顯示於圖(2. 1. 1)。



圖(2. 1. 1) 不同階數與頂點之B-軟樑

由於Hermite Spline無法局部控制曲線形狀，而B-Spline雖

可局部控制曲線形狀，卻不一定通過每一個控制點。爲改進上述缺點，我們採用將於下節所介紹的Kochanek-Bartels Spline。

第二節 Kochanek-Brtels Spline的數學模式

在介紹Kochanek-Bartels Spline之前，必須先介紹其前身——Cardinal Spline。考慮四個連續的控制點P_{k-1}，P_k，P_{k+1}，P_{k+2}，令通過此四點的曲線方程式爲

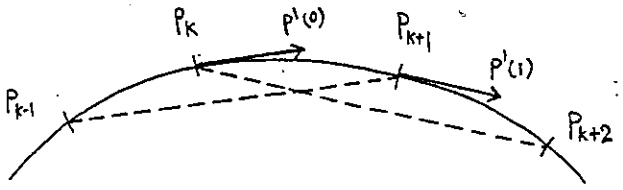
$$P(u)=au^3+bu^2+cu+d \quad , \quad 0 \leqq u \leqq 1 \ 。$$

若定義兩端點的一次導數如下：

$$P'(0)= (1-t)(P_{k+1}-P_{k-1})/2$$

$$P'(1)= (1-t)(P_{k+2}-P_k)/2$$

即曲線端點的斜率平行弦$\overline{P_{k-1}P_{k+1}}$及弦$\overline{P_kP_{k+2}}$，如圖(2. 2. 1)所示：



圖(2. 2. 1)

加上兩個端點條件： $P(0)=P_k$，$P(1)=P_{k+1}$

我們可將$P(u)$以矩陣型式表示之：

$$P(u)= U\ M_C\ P^T$$

式中 $U = [u^3\ u^2\ u\ 1]$

$$P = [P_{k-1}\ P_k\ P_{k+1}\ P_{k+2}]$$

$$M_C = \begin{bmatrix} -s & 2-s & s-2 & s \\ 2s & s-3 & 3-2s & -s \\ -s & 0 & s & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

其中 $s=(1-t)/2$

則$P(u)$稱爲Cardinal Spline。當$t=0$時，稱爲Catmull-Rom Spline。

參數t稱爲"tension"，用以控制曲線在控制點上的彎曲程度。當$t<0$，曲線較有弧度；當$t>0$，曲線趨近於"蹦緊"，即漸趨成爲直線。

D. H. U. Kochanek & R. H. Bartels[2]利用Cardinal Spline爲基礎，加上Continuity及Bias兩個控制變數於其中，而發展出Kochanek-Bartels Spline(以下簡稱K-B Spline)。它提供三種參數用以調整曲線形狀，其邊界條件則如下定義：

$$P(0) = P_k \quad , \quad p(1) = P_{k+1}$$

$$DS_k=(1-t)[(1+b)(1-c)(P_k-P_{k-1})+(1-b)(1+c)(P_{k+1}-P_k)]/2$$

$$DD_k=(1-t)[(1+b)(1+c)(P_k-P_{k-1})+(1-b)(1-c)(P_{k+1}-P_k)]/2$$

式中參數之意義爲

t = tension

b = bias (調整曲線偏左或偏右)

c = continuity (調整曲線邊界的連續性)

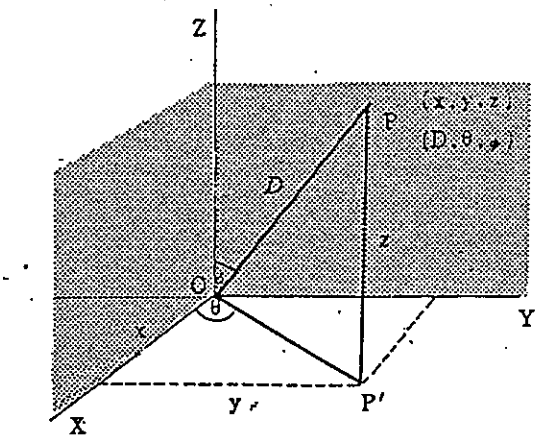
將每個控制點上的切線向量分成進入(incoming)及離開(outgoing)兩部份。若以第 k個控制點來說，其進入部分(又稱Source derivative)以DS_k表示，而離開部分(又稱Destination derivative)則以DD_k表示，其數學定義如上。如此一來，從P_k到P_{k+1}這一段的K-B Spline便可由P_k，P_{k+1}，DD_k，DS_{k+1}這四個邊界條件，代入求Hermite Spline的方法中，再將參數u由0 變化到1 ，即可產生在該區間(interval)內所有中間畫面(In-betweens)的位置。利用對t ，c ，b 三者的控制，吾人可調出所想要的路徑，並據以產生中間畫面。既可局部控制，又可通過每一個控制點(也就是Key-frame處)，這就是我們採用K-B Spline的主要原因。

第三節　三維電腦繪圖的數學基礎

因爲電腦顯像螢幕是二維空間，故我們必需導出在二維空間上顯示三維物體的機構作用，這可利用透視投影(Perspective projection)來完成。

在三維空間上，有直角座標系統(Rectangular coordinate system)及球面座標系統(Spherical coordinate system)兩種系統來表示點的座標。前者以具X，Y，Z座標的有序三維(x，y，z)數字，來表示點到YZ，XZ，XY平面的距離。而後者中每一點均以一有序三維數字(D，θ，Φ)表示，D爲點P到原點的距離，OP爲原點與P相連之線段，OP'爲OP在XY平面上的投影。OP與正Z軸夾角爲Φ，與正X軸夾θ角度，(θ依X軸之逆時針方向測得)。以圖(2. 3. 1)爲例，我們可建立上述兩種座標系統之間的轉換關係：

$$x = D \cdot \sin\Phi \cdot \cos\theta \qquad D = (x^2+y^2+z^2)^{1/2}$$
$$y = D \cdot \sin\Phi \cdot \sin\theta \quad 及 \quad \theta = \tan^{-1}(y/x)$$
$$z = D \cdot \cos\Phi \qquad \Phi = \cos^{-1}(z/D)$$

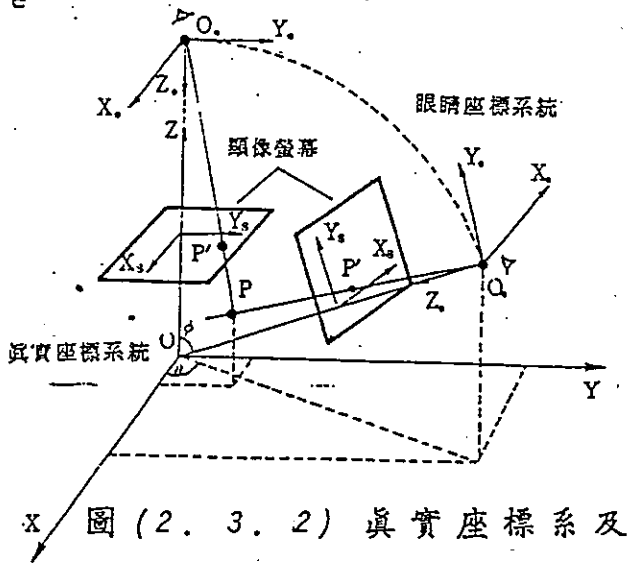


圖(2. 3. 1) 直角座標系及球面座標系的轉換關係

我們將以球面座標爲例，來確定觀視點(觀視者眼睛之位置)，此種方法可以簡化觀視點(Viewpoint)位置的處理[7]。

如上所述，透視投影可用以產生一個二維螢幕影像來模擬眞實的三維物體。有兩種透視投影的方式——中心投影(Central projection)及任意投影(Arbitrary projection)。由於中心投影的觀視點(投影中心)永遠位在眞實座標軸之一軸上，而眼睛座標軸則建立在投影中心，無法表達出一般狀況的投影，所以我們將只介紹任意投影的方法。一旦我們在二維空間上得到一三維物體的投影時，就需要一個從眞實座標系到眼睛座標系的轉換。此一轉換的目的，是要計算三維物體影像中每一點在顯像螢幕上的位置。任意投影的方法即是將觀視點置於三度空間的任意點上，將顯像螢幕定位於距離觀視點d 處，顯像螢幕則假定爲垂直於觀視者眼睛及原點的連線，如圖(2. 3. 2)所示。依照[7]所導出公式，若已知一點的眞實座標爲(x, y, z)，則眼睛座標(x_e, y_e, z_e)可由下列式子得之：

$$x_e = -x \cdot \sin\theta + y \cdot \cos\theta$$
$$y_e = -x \cdot \cos\theta \cdot \cos\Phi - y \cdot \sin\theta \cdot \cos\Phi + z \cdot \sin\Phi$$
$$z_e = -x \cdot \cos\theta \cdot \sin\Phi - y \cdot \sin\theta \cdot \sin\Phi - z \cdot \cos\Phi + D$$

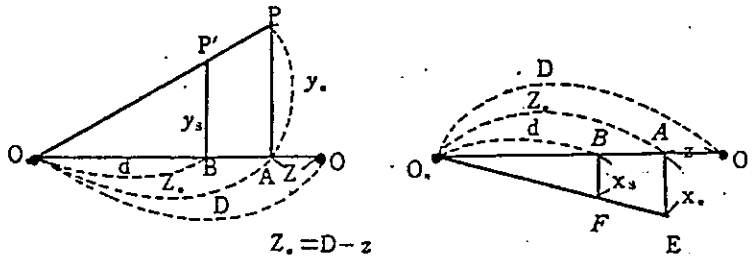


圖(2. 3. 2) 眞實座標系及眼睛座標系的關係

假設投影後影像在螢幕的座標爲(x_s, y_s)，由$Y_e Z_e$平面上

投參數間的關係(圖(2. 3. 3))可得螢幕座標如下：

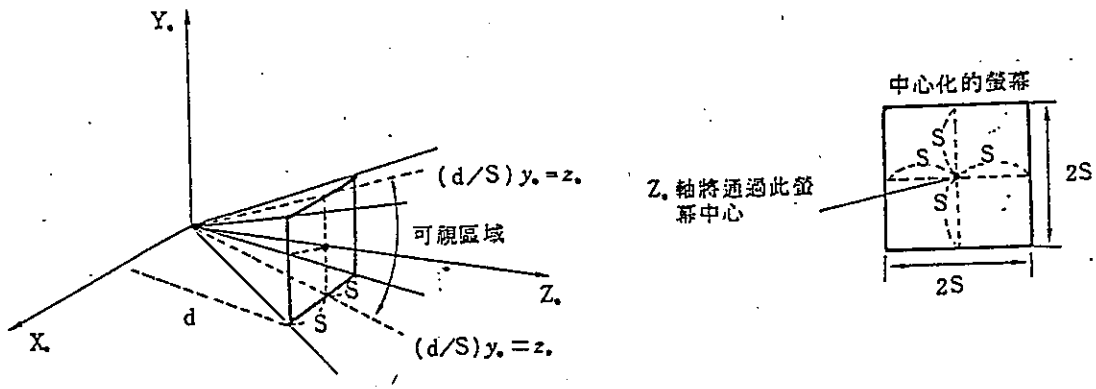
$$x_s = d \cdot (x_e/z_e)$$
$$y_s = d \cdot (y_e/z_e)$$



圖(2. 3. 3)

瞭解每一觀視參數(d ，θ ，Φ及D)如何影響顯像螢幕上所產生的影像是很重要的。改變θ及Φ的值可使我們從不同的角度去觀察一物體，D值改變可使觀測者移近或遠離一物體。若是(d ，θ ，Φ)固定時，改變d 可影響顯像螢幕上之影像大小。除此之外，尚可用另一種方法來控制影像大小。我們可將x_s及y_s用中心螢幕大小S(當原點位於螢幕中心時)加以分割，而轉換爲無維度的分數。由此可產生一個新的螢幕座標系，分別爲：

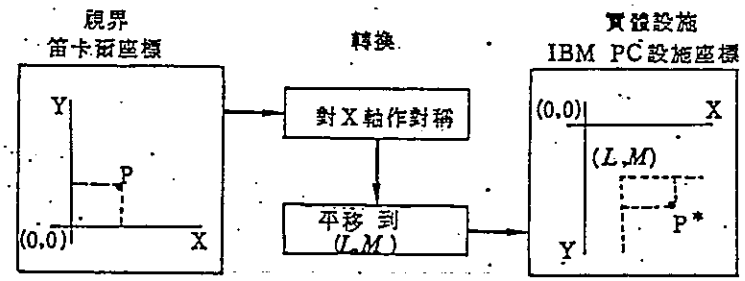
$$x_s = (d/s) \cdot (x_e/z_e)$$
$$y_s = (d/s) \cdot (y_e/z_e)$$



圖(2. 3. 4)

如圖(2. 3. 4)所示三角錐指定眼睛座標系中的一區域，當物體爲可見時，必須在此區域中。若觀測點保持不動，比率(d/S)可控制螢幕上的影像大小，其效果類似於照相機的對焦(Focus)功能。

大部份微電腦所提供的顯像螢幕，其座標系統的位置及方向與傳統的笛卡爾座標系統有些不同。此可由圖(2. 3. 5)得知。



圖(2. 3. 5)

以IBM PC及其相容性機種而言，其座標原點(0，0)位於螢幕的左上角，正X軸朝右，正Y軸朝下。爲能充分表達笛卡爾座標系中的某一點，我們必須做個轉換。非常簡單，首先將該點對X軸作對稱，再將原點平移(L，M)到某一螢幕位置即可。但是由於微電腦繪圖中，電腦螢幕的水平尺度及垂直尺度是不同的，隨著繪圖模式的解析度而變化。因此必須考慮螢幕尺度調整因素SCF進去。以本文程式中採用繪圖模式爲Hercules mode 而言，其解析度爲(720 x348)，所以採用SCF 等於1.5 。若視界以(L，M)爲中心，且寬度爲2Vx單位，長度爲2Vy單位，則可求出該點在微電腦顯像螢幕上的顯像座標爲：

$$x_d = SCF\ [(d/s)(x_e/y_e)Vx + L)$$

$$y_d = -(d/s)(y_e/z_e)Vy + M$$

經由上述轉換公式，即可順利地將一三維物體顯示於二維的電腦螢幕上。

第三章　路徑之控制方法

雖然在前一章已對本文中所採用的K-B Spline簡單介紹過，本章仍將以較多篇幅來對整個動畫路徑的控制做更詳細的說明。第一節介紹如何利用四線譜來對圖形中的某一點做路徑控制，第二節敘述如何控制單一物體(Single object)的路徑，最後一節則舉例說明整個圖形的路徑控制。

第一節　由四線譜來控制一點的路徑

在動畫的產生過程中，每一個控制點的路徑即該點的運動軌跡。該路徑必然通過每一個主畫面的各該點，所以路徑的行走方式一定屬於內插式曲線(Interpolation curve)之一種，這也正是本文採用K-B Spline的主要理由。

讓我們再複習一下K-B Spline的公式定義。首先我們分別討論只有(1)tension控制變數 (2)continuity控制變數 (3)bias控制變數等獨立情形的方程式，接著討論將三種變數整合後的方程式，也就是K-B Spline公式。

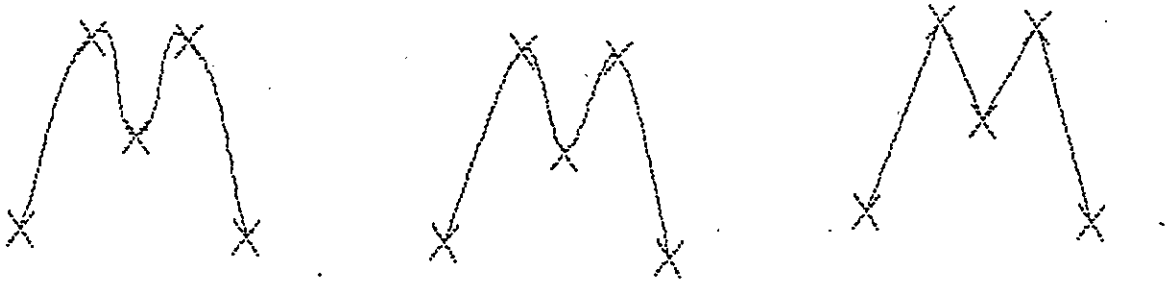
(1)Tension

此tension變數t控制了曲線在某一關鍵位置(Key position)彎曲的尖銳(Sharply)程度，其內定值(Default value)爲0。該tension方程式如下所示：

$$DS_k = DD_k = (1-t)[(P_{k+1}-P_k) + (P_k-P_{k-1})]/2$$

Eq. (3.1.1)

當t值爲1，則$DS_k=DD_k=0$。切線向量爲0，造成曲線在該處有蹦緊(tighten)的感覺。當t值爲-1，曲線相反地便有鬆弛(slack)的感覺。圖(3. 1. 1)便爲t=-1 ，0 ，1 時所產生的曲線形狀。



圖(3. 1. 1)

(2)Continuity

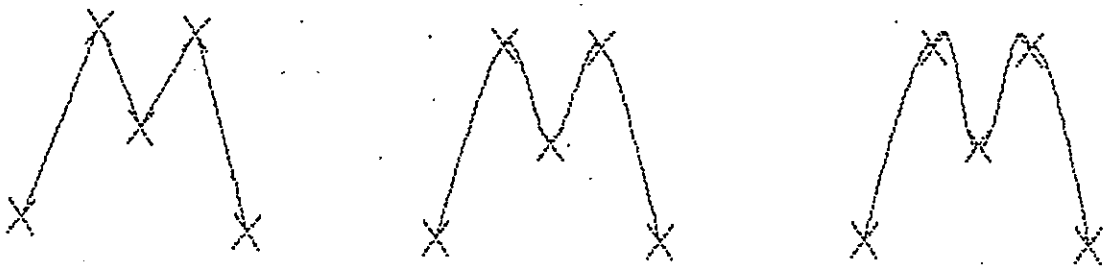
此continuity變數c可控制曲線的不連續性，以求得特殊效果。諸如球的跳躍運動等，其內定值亦爲0。該continuity方程式如下所示：

$$DS_k = [(1-c)(P_k-P_{k-1}) + (1+c)(P_{k+1}-P_k)]/2$$

$$DD_k = [(1+c)(P_k-P_{k-1}) + (1-c)(P_{k+1}-P_k)]/2$$

Eq. (3.1.2)

當$c=0$，則$DS_k=DD_k$。表示曲線在該處的切線向量連續，也就是正常常見的情行。當$|c|$值增大時，DS_k與DD_k的差距會愈來愈大，會在該處產生明顯的角度，這正是不連續性的表示。圖(3. 1. 2)乃爲$c=-1$，0，1時所產生的曲線形狀。



t=0,c=-1,b=0　　t=0,c=0,b=0　　t=0,c=1,b=0

圖(3. 1. 2)

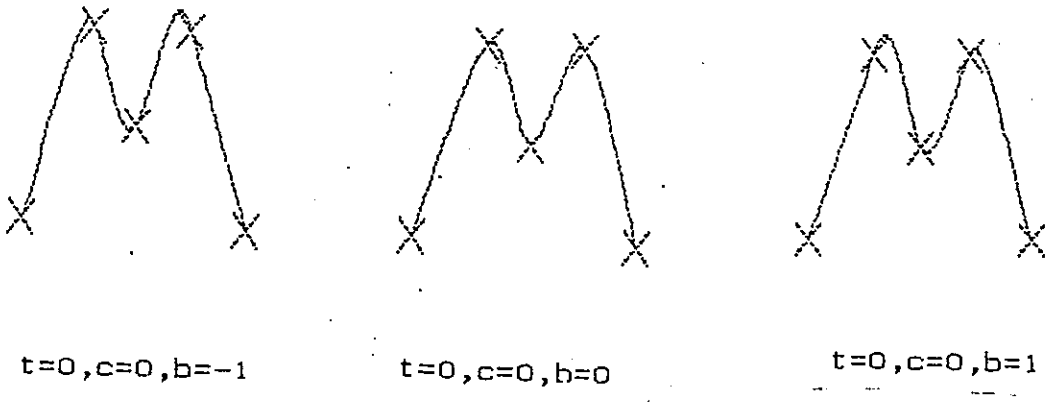
(3)Bias

此bias變數b控制曲線在某一關鍵位置的偏向。該bias方程式如下所示：

$$DS_k = DD_k = [(1+b)(P_k - P_{k-1}) + (1-b)(P_{k+1} - P_k)]/2$$

Eq. (3.1.3)

b之內定值爲0，亦即曲線既不偏左，也不偏右。當b值爲1，曲線將偏向於右邊；當b值爲-1，曲線將偏向於左邊。圖(3. 1. 3)便爲b=-1，0 ，1 時所產生的曲線形狀。



圖(3. 1. 3)

(4)Tension，Continuity及Bias結合

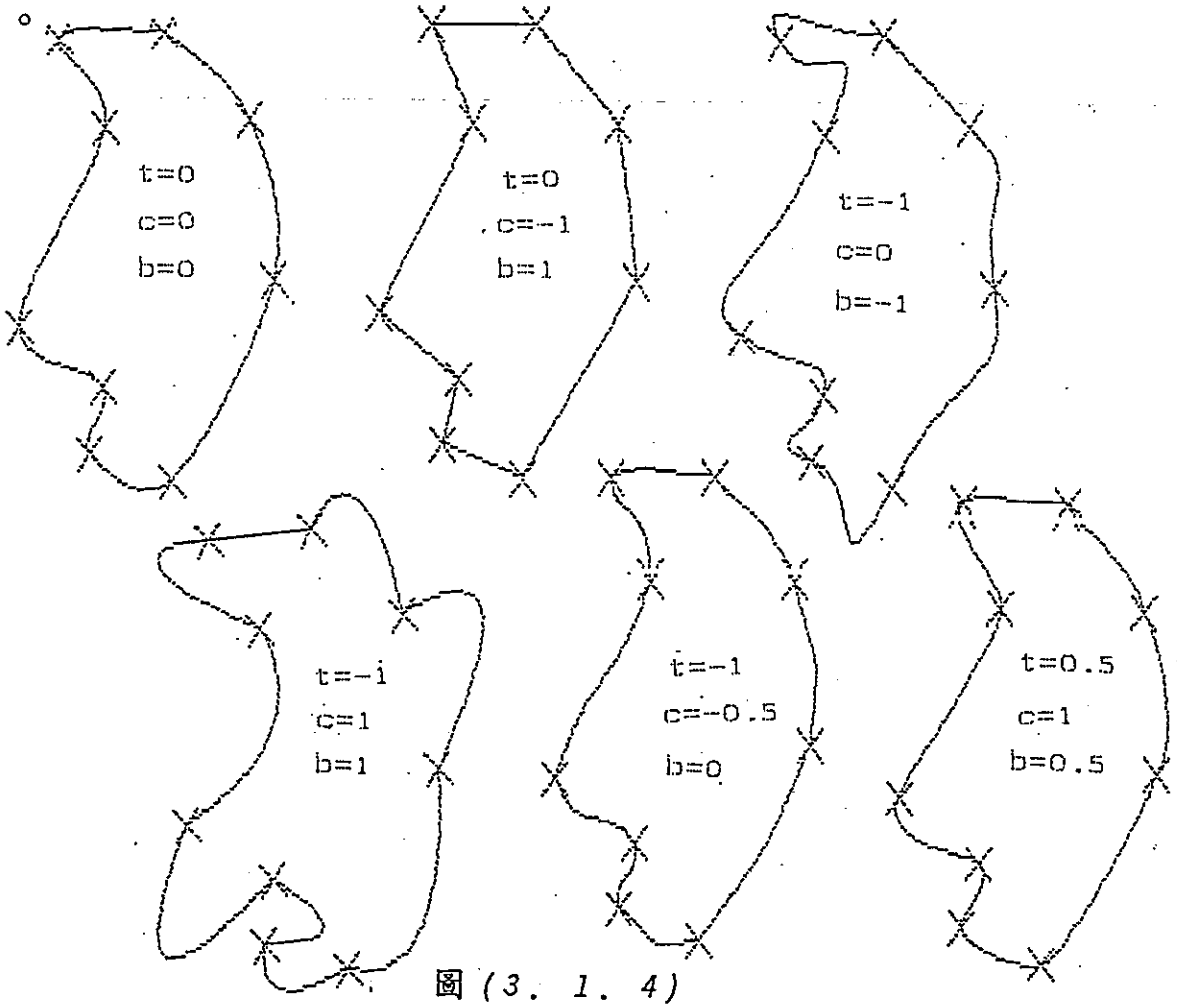
將上面三種變數同時考慮進去，藉著調整t，c，b三個變數，便可產生許多滿足吾人所需的路徑形狀，也就是K-B Spline的產生。其公式如下：

$$DS_k=[(1-t)(1-c)(1+b)(P_k-P_{k-1})+(1-t)(1+c)(1-b)(P_{k+1}-P_k)]/2$$

$$DD_k=[(1-t)(1+c)(1+b)(P_k-P_{k-1})+(1-t)(1-c)(1-b)(P_{k+1}-P_k)]/2$$

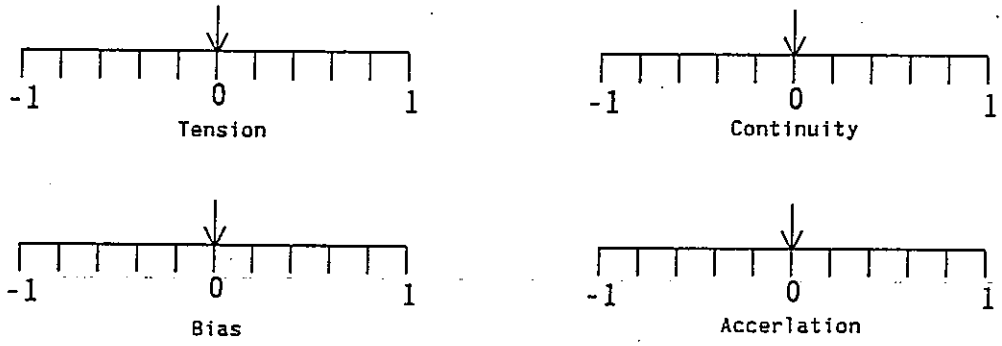
Eq. (3.1.4)

圖(3. 1. 4)乃不同的t，c，b值所調出來的K-B Spline圖形



圖(3. 1. 4)

由上所知，用t，c，b三者可調整路徑的幾何形狀。但爲了使動畫能產生生動的效果，我們必須加上一個加速度(accerlation) a的參數進去，供使用者(畫家)調整，於是便有四線譜的產生。所謂四線譜，就是分別代表tension，continuity，bias及accerlation四種參數的四條線。它們提供畫家能夠很輕易的由電腦鍵盤(Keyboard)來調整這四項參數值而得到所需要的路徑，如圖(3．1．5)所示。



圖(3．1．5) 四線譜

Accerlation此參數a之主要作用在於調整物體在路徑上顯現的速度快慢。若a=0，表示等速度運動，物體將以近似等間隔(interval)的方式顯現出來。若a>0，物體顯現的間隔將愈來愈大，而讓使用者有加速度的感覺。同理，若a<0，物體顯現的間隔會愈來愈小，亦即減速度的感覺。

假設K-B Spline公式所產生的中間畫面數目爲N(相當於所拍照片之張數)，但實際上所採用之中間畫面數目爲M，則必須M<=N，否則極可能某一個中間畫面重複採用多次，而產生”畫面停頓”之效果，致使動畫有不連續的感覺，這是吾人所需注意的事項。除非是動畫本身需强調該效果，那又另當別論。

至於如何由N個中間畫面去採取M個來使用，有很多的方法可以實施。以本文所採用的方法爲例，我們可算出通過前後兩個主畫面的路徑總長度L。若a=0，則將L平均等分成(M+1)份而有M個

分割點。由前一主畫面起算，從N個中間畫面選取M個最靠近分割點的中間畫面即可。若爲減速度運動，則使用餘弦(cosine)函數來分割L。其變化範圍爲π／2到0，分別對應到前後兩個主畫面(因爲cos(π／2)=0，cos 0=1)。將π／2均等分成(M+1)份，每一個分割點之餘弦值兩兩之間的變化量愈來愈小。從這些分割點所選擇的中間畫面便有減速度的感覺。至於加速度部份，只要用1減去減速度部份公式所得的餘弦值即可。

利用四線譜對某一控制點做路徑控制後，下一節我們將討論單一物體的路徑控制。

第二節　單一物體的路徑控制

一個單一物體是由許多控制點所組合而成的圖形。基本上來說，這些圖形可括分成兩大類：(1)開放型曲線(Open curve)及(2)封閉型曲線(Close curve)。所謂開放型曲線是指圖形中起點位置與終點位置不相同而造成圖形無法閉合，此類圖形稱之。若是圖形中起點與終點爲同一點，則稱爲封閉型曲線。

以單一物體來說，由於每個圖形的控制點數目少則數十個，多則數以萬計。我們在探討物體的路徑控制時，不可能針對物體中的每一個控制點做路徑控制。如此不但浪費時間，而且可能無法達成在動畫過程中所要求的”單調化”(monotonic)。所謂”單調化”即指於前後主畫面間產生的中間畫面其圖形變化呈現遞增或遞減的動作，而無一會兒增加，一會兒卻又突然縮減的情形發生。爲了解決此問題，我們可利用一種稱爲”曲面調適”

(Surface-Fitting)的方法來完成。

在介紹曲面調適方法之前，有兩個問題需先加以討論。

由於中間畫面完全由前後兩個主畫面及其路徑來決定，所以主畫面與中間畫面圖樣中的點及部位必須爲一對一對應。以一個階層式樹狀結構而言，除了根部節點(root node)以外，其下的各個分支樹均可稱之爲部位(part)。以機器人(Robot)爲例，它由頭、手、足、軀幹所組成，而這些手、足等分支即可說是組成機器人的各個不同部位。在邏輯上的觀點，每一個主畫面與中間畫面的點數目及部位數目是相等的。實際上由於有時候需要特殊的畫面來表達誇張的效果，通常前後兩個主畫面的點數目或是部位數目是不會一致的。爲了能夠順利的產生中間畫面，我們必須做預先處理(Pre-Process)，使得前後主畫面的點數或部位數目相等。依據[1]的方法，可得下列兩個演算法(Algorithms)來解決此問題。

假設前後兩個主畫面(Key1及Key2)其部位數目分別爲N1及N2，點數目分別爲NP1及NP2。若N1=N2且NP1=NP2，則不需任何的預先處理即可直接產生中間畫面，否則便由下述二演算法預先處理之。演算法1是針對不同的部位數目之解法，演算法2則是在同樣部位數目下，卻有不同點數目的解法。

演算法一：

設N1>N2，(若N2>N1，此演算法相同，只是N1與N2之角色互換而已。)

則 N1-N2 個部位數必須加到Key2上面。

至於如何加法，則由下列原則訂之：

令 $A = N1 \bmod N2$

$B = N1 \operatorname{div} N2 + 1$

$C = N1 \operatorname{div} N2$

從Key2中取出A個部位，將每個部位分解成B個部位；剩下的(N2-A)個部位則每個部位分解成c個部位，如此Key2中的部位數目即會與Key1相同。

[$\because$ $A \cdot B + (N2-A) \cdot c = N1$]

演算法二：

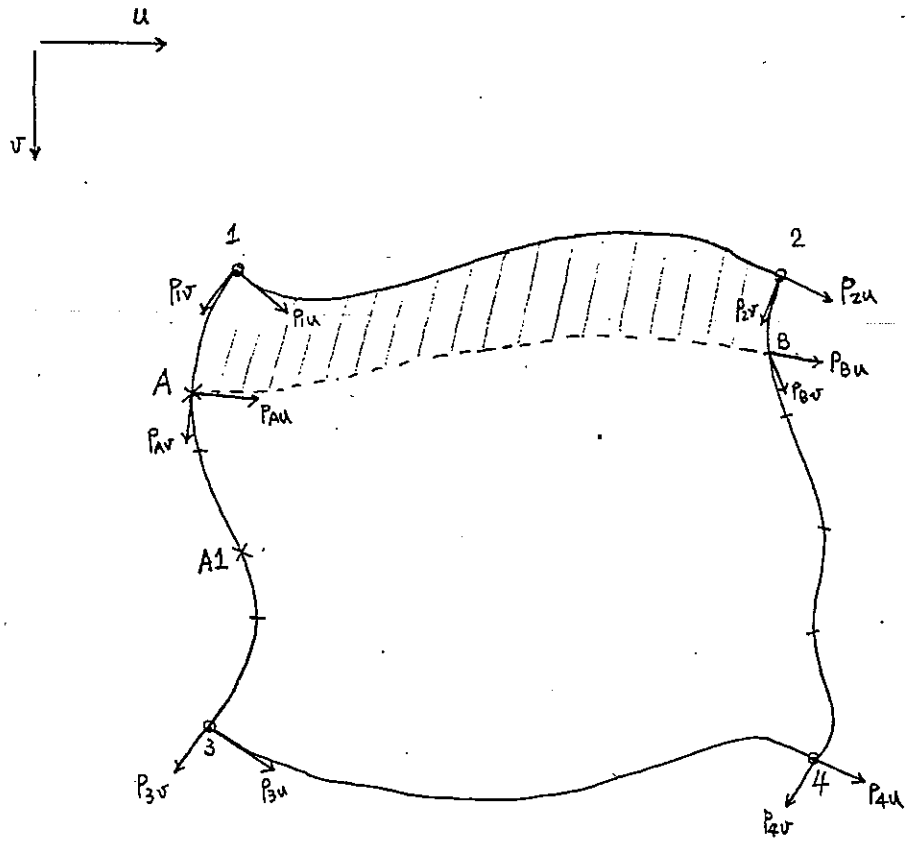
設NP1>NP2，

令 $RT = (NP1-1) \operatorname{div} (NP2-1)$

$RS = (NP1-1) \bmod (NP2-1)$

從Key2取前面RS個線段區域，每個區域增加RT個點；剩餘部份每個區域增加(RT-1)個點即可。

接著討論利用曲面調適方法來對單一物體做路徑控制。以圖(3. 2. 1)爲例，假設圖(3. 2. 1)中Key1及Key2均爲開放型曲線。



圖(3. 2. 1) 曲面調適

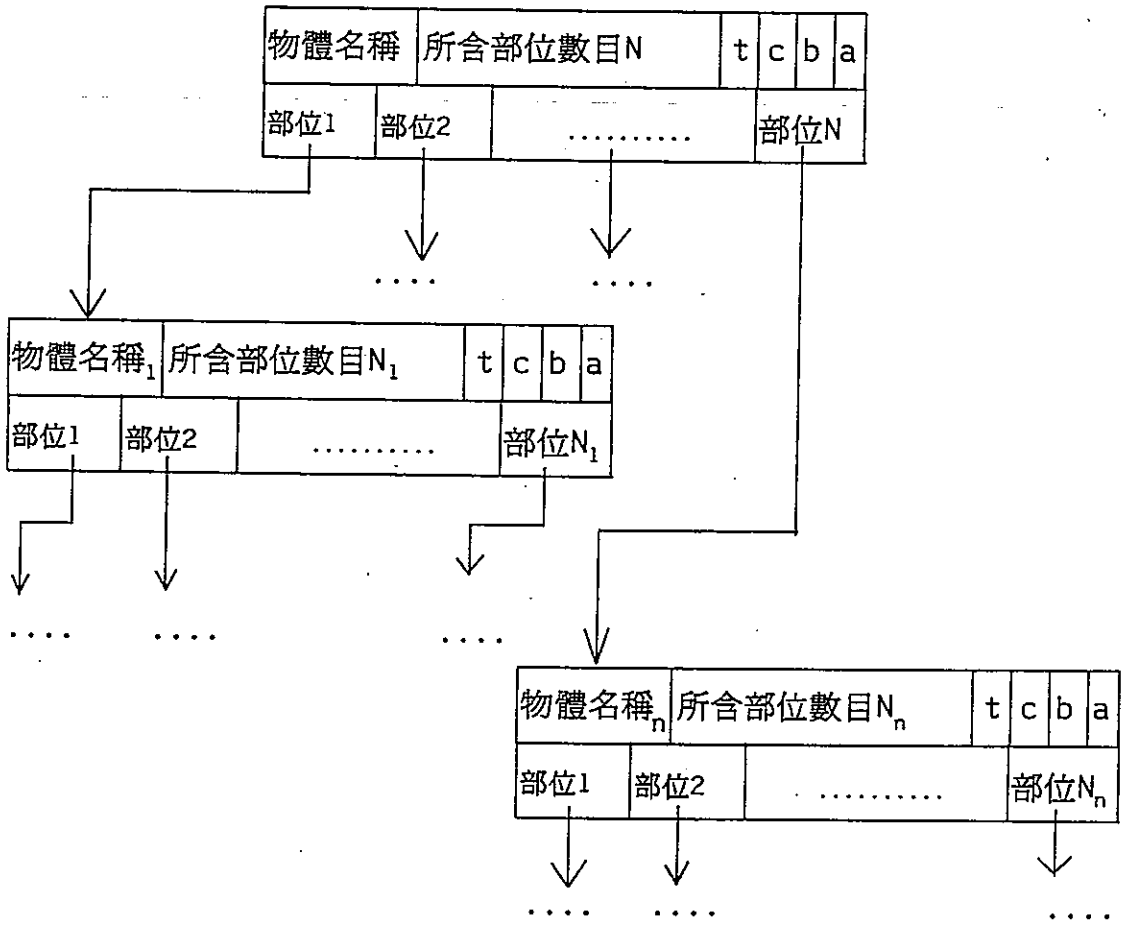
Key1圖形中有四個控制點，Key2有六個控制點。吾人可利用上面曾提過的演算法二在Key1圖形中另外增加兩點A及A1(如"x"號所標示者)。對於開放型曲線，我們必須提供首尾兩端點之路徑控制。u代表路徑變化方向，v代表圖形本身變化方向。Pu表示u方向之切線向量，Pv爲v方向之切線向量。由於Key1及Key2的端點條件：點1，2，3，4的位置向量，Pu及Pv均已知，再假設該四端點上的扭轉向量爲零。由這些條件便可建立一個Ferguson曲面[3]，它屬於雙三次(bi-cubic)曲面之一種。從該Ferguson曲面上，我們可得知點A及點B在u方向的切線向量P_{Au}及P_{Bu}。再利用點1，2，A，B的條件，用相同方法可建立圖(3. 2. 1)中陰影部份的曲面。此時將曲面內通過A，B兩點的邊界當成點A移至點B的路徑，其餘各點的路徑求法相同，最後加以整合便是一個物體的路徑控制。

上述曲面調適方法基於曲面特性，我們可認爲整個物體的路徑控制是平滑的(smooth)。至於封閉型曲線，只要將它視爲開放型曲線經捲曲(wrapped)後的結果即可。因爲首尾均爲同一點，所以曲線只需給定起點的一條路徑控制即可。

第三節　畫面的路徑控制

通常一個畫面是由許多物體(*Multi-objects*)所構成，如果這些物體相互之間無任何關係，則可以用上節所述單一物體之路徑控制方法分別調整，最後再加以整合即可。若是物體之間互相有所關連，則需詳加注意。

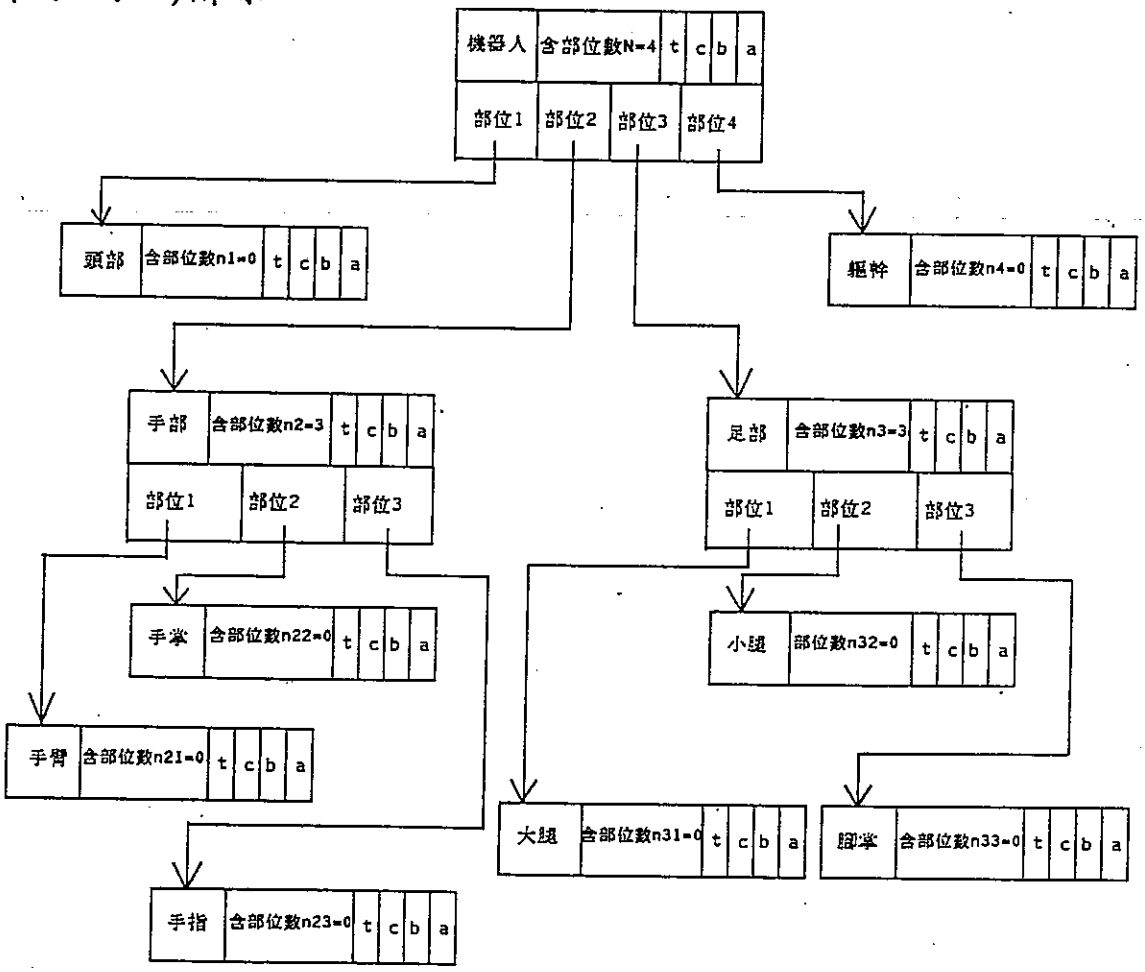
首先我們定義一個畫面的資料結構，如圖(*3. 3. 1*)所示。



圖(*3. 3. 1*) 畫面的資料結構

在圖(3. 3. 1) 的資料結構中，t,c,b,a 即前面所提及的四線譜之值，所含部位數目N 決定了串列(Link)數目，若N=0 ，表示該物體爲最底層的樹葉節點(leaf node) ，其下不再包含任何部位，均可視成爲單一物體。

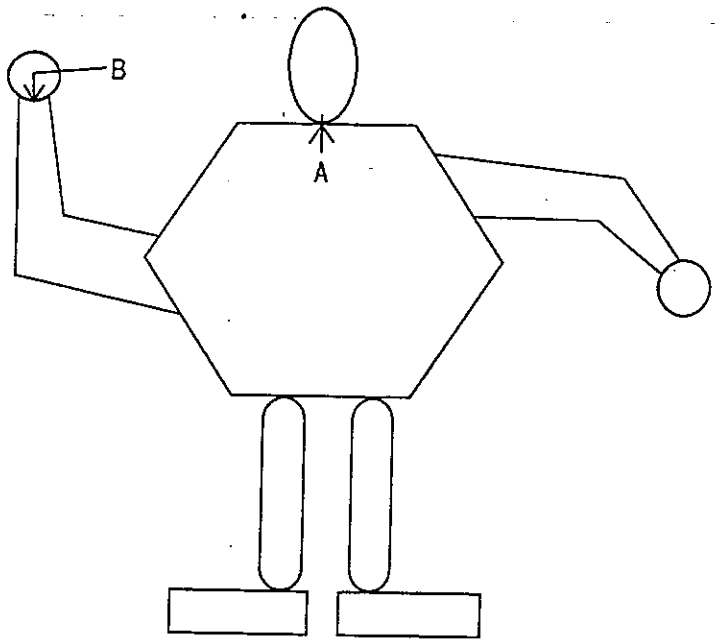
以機器人的動畫爲例，假設它是由頭部、軀幹、手部、足部四部份組成，而手部又可分成手臂、手掌、手指三部份，足部可分成大腿、小腿及脚掌三部份。則機器人的資料結構如圖(3. 3. 2)所示。



圖(3. 3. 2) 機器人的資料結構

在製作整個畫面動畫時，只要某一個部位的四線譜經過調整，則其底下的子部位(Sub-part)應該馬上繼承其值而跟著調整。另外爲了防止在動畫過程中可能發生”頭尾分家”的現象，我們必須限定各個關鍵點(Key-point) 相互之間的距離只能在一個容許的誤差範圍內，才不致因爲某一部位在運動時而其他相關部位卻保持不動的分離現象，如此的動畫便不夠逼眞，而無法達到我們的目標。

在此必須對關鍵點下定義，凡是各部位與其他相關部位間的相接處即爲關鍵點。若相接處爲一個區域，則取該區域的重心爲關鍵點。以圖(3. 3. 3) 來說，機器人的頸關節(點A)或是手腕關節(點B)均是關鍵點之一。



圖(3. 3. 3) 機器人

各個部位均有其自己的區域座標系統(Local coordinates system)及四線譜以供調整各該部位的動畫，而以整個機器人來

說是一個總體座標系統(Global coordinates system)，它們之間的相互關係應把握住。若單一部位只是區域性的運動，且其關鍵點所移動距離在容許誤差內，則其他部位均不受影響，否則與該部位有關連的部位均須以相同的四線譜之值去做路徑調整，如此動畫才不致於失真。

第四章　A.P.C.S.系統架構

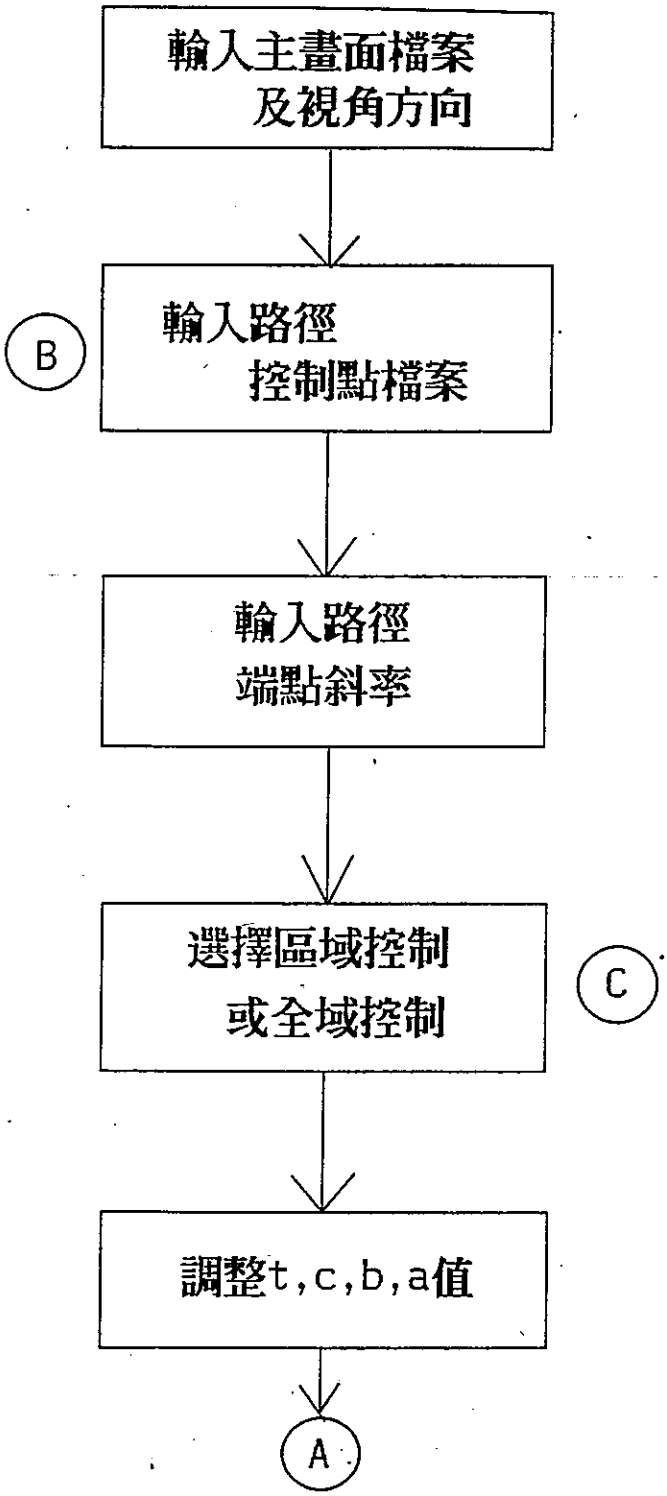
依據前面所述，我們使用C 語言，以Turbo C 1.5版在與IBM PC/AT相容的16位元個人電腦上發展一套動畫路徑控制系統(Animation Path Control System，簡稱A.P.C.S.)來實際測試。

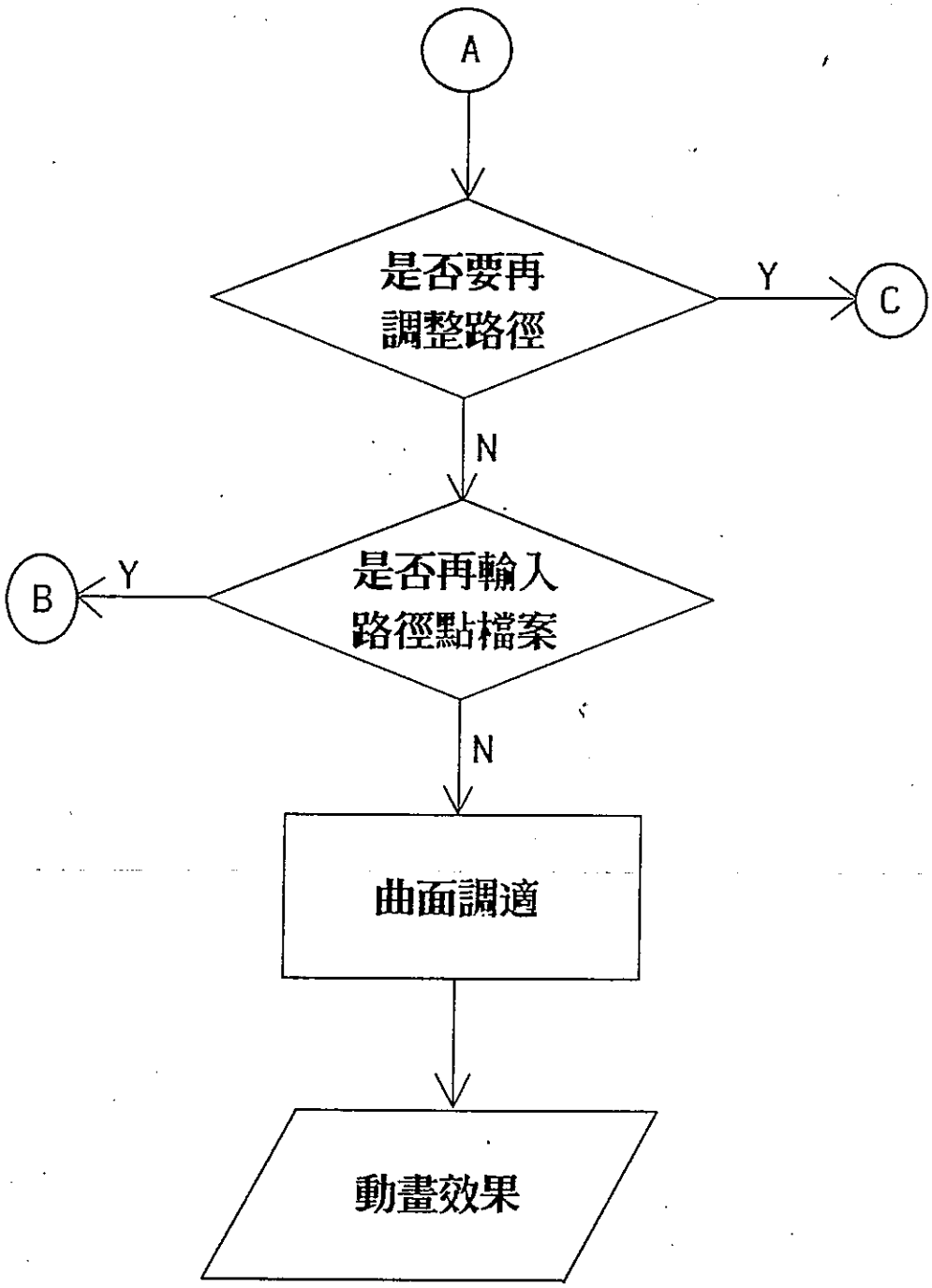
本章第一節將介紹該系統架構、功能及所受限制。第二節則以圖例配合，引導使用者如何使用該系統。

第一節　A.P.C.S.之架構、功能及限制

A.P.C.S.由一主程式animate.c及三個副程式addpoint.c，split.c及s_fiting.c所組成。主程式內有許多函數，其功能稍後再作介紹。副程式addpoint.c之主要功能在測試前後兩個主畫面所含的控制點數是否相同，若數目不同，addpoint.c將以線性內插之方式使其相同。副程式split.c則將前後兩個圖形控制點檔案拆開，再重新組合成只含前後兩圖形各一點之路徑控制點檔案。至於s_fiting.c之主要功能在於使用曲面調適的方法來產生除端點之外，其餘各個中間控制點之動畫路徑。

圖(4. 1. 1) 為A.P.C.S.之系統架構圖。


輸入主畫面檔案
及視角方向
B
輸入路徑
控制點檔案
輸入路徑
端點斜率
選擇區域控制
或全域控制
C
調整t,c,b,a值
A



圖(4. 1. 1) A.P.C.S.系統架構

接著介紹主程式animate.c中各個函數功能及其輸入/輸出。函數功能表介紹如下：

1.Define_View_Parameters()

輸入：由使用者鍵入視線方向角度值及其顯示圖形之間隔時間。

輸出：無。

功能：定義觀視者(Viewer)視線方向及視界轉換係數。

2.Read_data_file()

輸入：鍵入含有路徑控制點座標的檔案名稱，例如anil.dat。

輸出：產生控制點在螢幕上的二維(2D)座標。

功能：讀取一連續控制點的三維(3D)座標資料，經由

VIEW_Transform(a,b,c)，產生這些控制點的螢幕座標。

3.VIEW_Transform(a,b,c)

輸入：一點的空間座標，x,y,z座標分別爲a,b,c。

輸出：該點在螢幕上顯示的二維座標。

功能：由一些轉換公式，而達成視界轉換的目的。

4.Predefine_ENDS_slope()

輸入：鍵入K-B Spline前後兩端點的斜率。

輸出：無。

功能：可由使用者定義起點及終點的切線向量。

5.Initial_Graphics()

功能：由Turbo C 所提供的函數，目的爲設定繪圖模式。

6.Cal_Hermite_Basis()

功能：產生Hermite blending function之值。

7.Make_Boundary()

功能：顯示螢幕初始狀態(Initial state)。

8.Check_GorL_Control()

輸入：以鍵盤中的數字鍵8或2選擇顯示於螢幕上的全域控制(Global Control)或是區域控制(Local Control)。

輸出：無。

9.Pre_Draw()

功能：顯示控制點之位置，以”x”代表之。

10.Scan_Control_Point()

輸入：選擇一個區域控制部份的首尾兩個控制點。

輸出：無。

功能：此函數僅存在於區域控制函數(Local Control())中，定出其控制區域之範圍。

11.Scan_Parameters()

輸入：由使用者於鍵盤上數字鍵4或6選擇四種參數值。

輸出：無。

功能：供使用者調整路徑參數之用途。

12.Scan_Tension_Value()

輸入：由使用者於螢幕上調整tension值。

輸出：螢幕上顯示受tension影響後的路徑形狀。

功能：可控制K-B Spline之瑞緊或疏鬆度。

附註：Scan_Continuity_Value()

Scan_Bias_Value()

Scan_Accerlation_Value()

三者I/O與此類似，功能則分別控制K-B Spline之連續性，偏向度及加速度。

13.tnextpixel()

功能：接收由使用者敲擊鍵盤所傳送的訊息。

附註：cnextpixel()，bnextpixel()及anextpixel()均同。

數字鍵 4表示往左移 1單位

數字鍵 6表示往右移 1單位

SHIFT-L 表示往右移 3單位

SHIFT-J 表示往左移 3單位

ENTER 表示接收該參數值

ESC 表示跳出此項函數

14.DRAW(dis，die)

功能：畫出由點dis到點die此段(segment)的K-B Spline。

15.Cal_tangent_vector(dis，die)

功能：計算從點dis到點die之區段中每個控制點

的切線向量，利用K-B Spline之公式求得。

16.Cal_INBETWEENS(dis，die)

功能：計算從點dis到點die之區段內，其中間畫面的值。

17.Draw_Ani_Path(dis，die)

功能：繪出由點dis到點die此區段之動畫路徑。

18.Accer(is，ie)

功能：決定由點*is*到點*ie*此區段之加速度值。

19.Decide_Show_Inbetween()

功能：決定那一個中間畫面要顯式出來。

20.Ask_Adjusting()

輸入：由使用者決定，按’*Y*’可重新調整路徑，

否則即確定該路徑形狀。

輸出：無。

功能：可調整路徑至滿足所需要爲止。

21.Showing_Animation()

功能：於螢幕上繪出動畫的顯現效果。

請注意本系統現階段只做到對單一物體做路徑控制的境界，且僅能針對由前後兩張主畫面之中間路徑做控制，而無法一次輸入多張主畫面做各區域之路徑控制。再者，若爲封閉型曲線，雖然僅需一條路徑，但本系統仍會要求第二條路徑，此時只需將第二條路徑之參數值按第一條路徑所給之參數值指定即可！

第二節　如何使用A.P.C.S.

步驟一：

在DOS狀態下，請鍵入APCS。則系統會要求使用者依序輸入前後兩個主畫面的控制點檔案名稱(例如Key1.dat及Key2.dat)。(如圖(4. 2. 1))。

```
A>apcs
Please keyin the KEY1 Filename!
key1.dat
Please keyin the KEY2 Filename!
key2.dat
```

圖(4. 2. 1)

步驟二：

當系統接收訊息之後，會將此二檔案分割(Split)成數個僅含前後主畫面各一個控制點所組成的檔案，檔案數目即前後主畫面中所含控制點數目較多者。其命名均以ani爲字首，並且會在螢幕上顯示訊息，請使用者輸入視線方向、延時及兩個ani*.dat的檔案去做路徑控制。(如圖(4. 2. 2))。

```
Adding_Points work is done.
You must Input 2 Files to adjust animation path:
  ani1.dat and ani4.dat.
ani1.dat contained the first graphic point to adjust its path.
ani4.dat contained the last graphic point to adjust its path.
Please input the VIEWER'S direction and delay time:
THETA ,PHI & delay-time value.
30 40 200
```

圖(4. 2. 2)

步驟三：

當要調整第一條路徑時，使用者需輸入*ani1.dat*，再輸入曲線之端點斜率後(如圖(4. 2. 3))，螢幕將由文字模式變成繪圖模式(如圖(4. 2. 4))。使用者以數字鍵8或2選擇*Global*或*Local Control*。若爲*Global Control*，螢幕上以"x"表示之所有控制點之間的路徑皆受控制，否則使用者便可利用一些特殊鍵移動螢幕左上角之箭號去選擇所影響的控制區域，(如圖(4. 2. 5))。

```
Please keyin the anix.dat filename
to adjust the point's animating path!
ani1.dat
The Starting and Ending Points' slope ranged from 0 to 1.
Plsase input the slope : DS[0],DD[N]
1
1
```

圖(4. 2. 3)

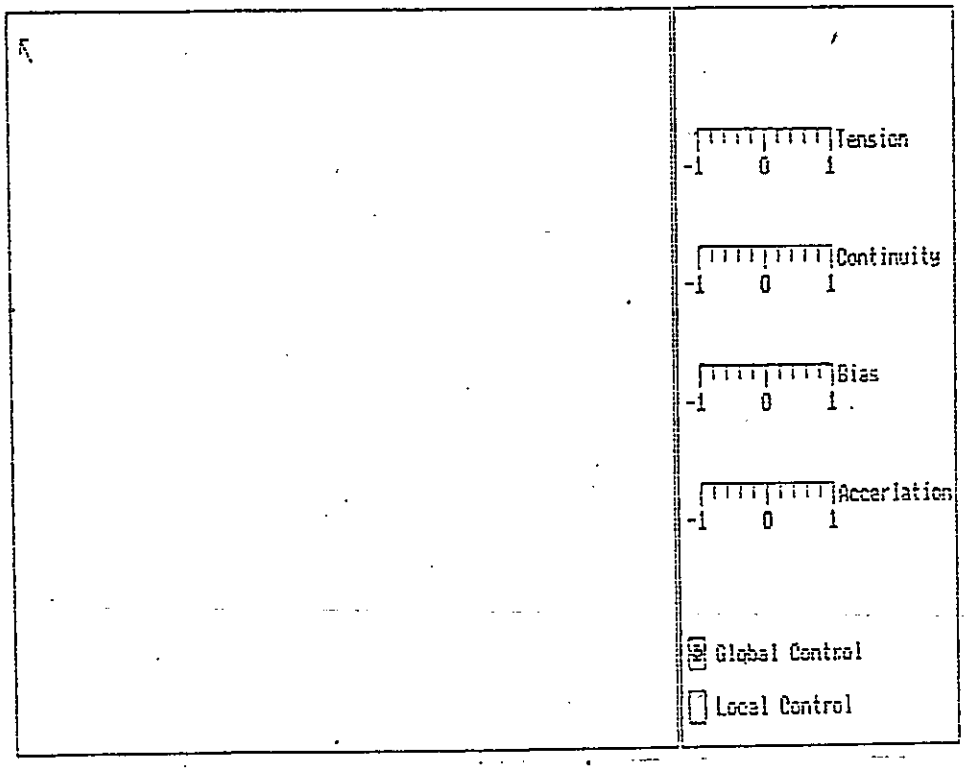

Tension
-1 0 1
Continuity
-1 0 1
Bias
-1 0 1
Acceleration
-1 0 1
Global Control
Local Control

圖 (4. 2. 4)

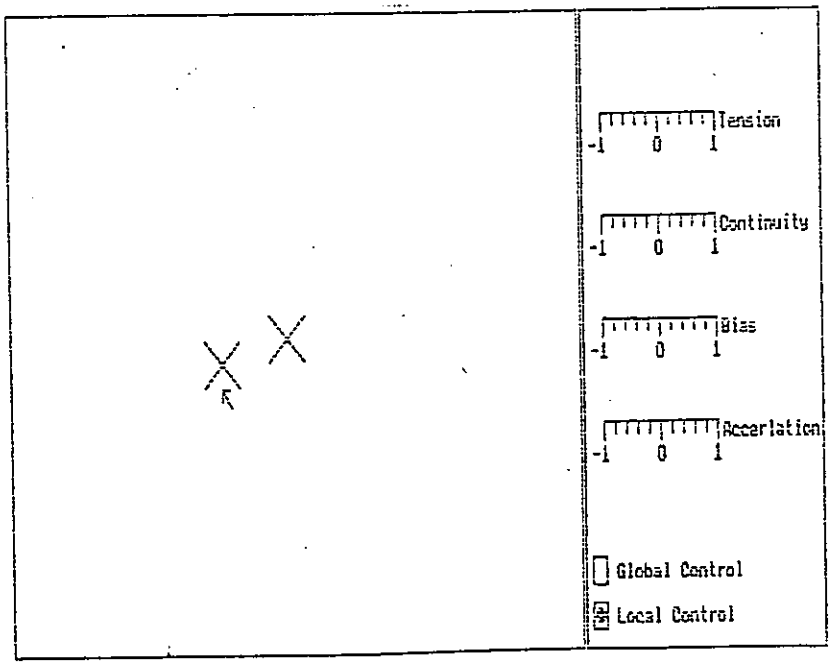
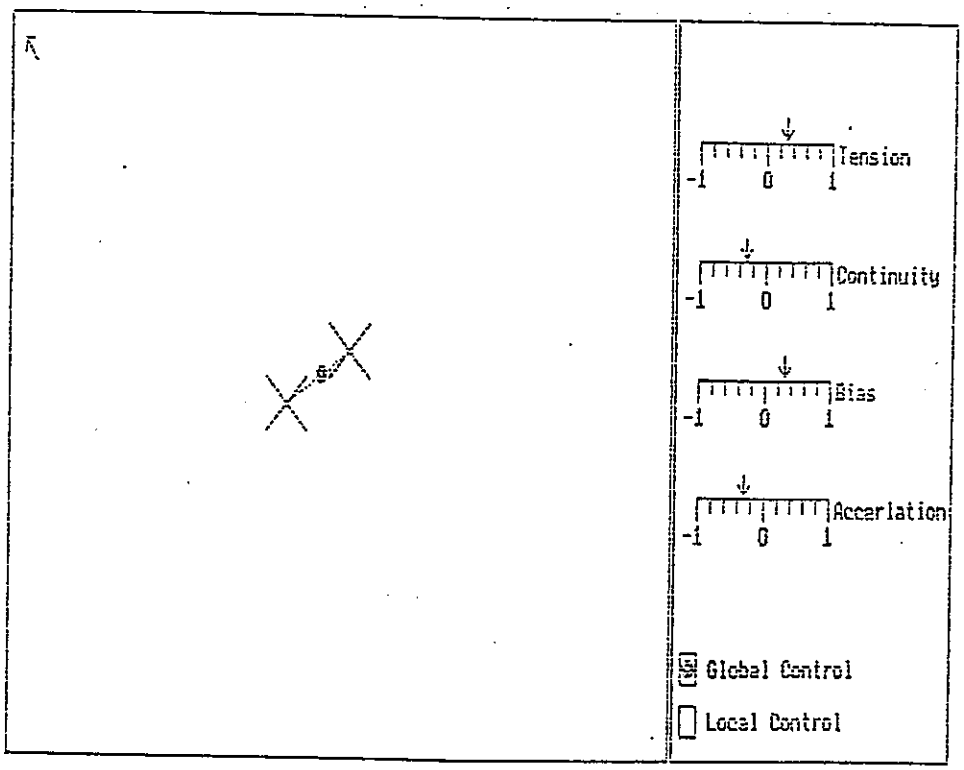

Tension
-1 0 1
Continuity
-1 0 1
Bias
-1 0 1
Acceleration
-1 0 1
Global Control
Local Control

圖 (4. 2. 5)

步驟四：

上述步驟完成後，使用者便可利用鍵盤去調整*K-B Spline*的參數值。依序分別是*tension*，*continuity*，*bias*及*accerlation*。其範圍均定在*-1*至*1*之間，它們的內定值均爲*0*。每次只要調整一個參數，路徑形狀或是顯現速度便會立刻更動而於螢幕上顯示之，*(*如圖*(4. 2. 6))*。若是不想調整該參數，按*ESC*鍵離開即可。



圖*(4. 2. 6)*

步驟五：

當四項參數值均給定後，系統會詢問是否要再重新調整路徑。若要重新調整，按'Y'鍵，否則系統便接收此一路徑資料。緊接著系統會再次要求使用者按'Y'鍵後再輸入第二條路徑之檔案名稱(例如ani4.dat)，之後步驟與輸入第一條路徑時完全相同。

步驟六：

待第二條路徑調整完畢，系統會要求使用者按ENTER鍵以結束工作。請注意，按ENTER完後，只是告訴系統可以開始做曲面調適，以便求得其餘各控制點之路徑，並非整個工作立刻完成。由於曲面運算繁瑣耗時，估計約需五分鐘之久，請使用者耐心等待，至出現" Surface-Fitting work is done." 此訊息之後才算是眞正的結束，請再按任何鍵離開以便畫出圖形(如圖(4. 2. 7))。

```
Press <ENTER> to Stop the work !
Please wait a moment! Surface Fitting is working now .....
Surface-Fitting work is done.
Press any key to draw the animation effect!
```

圖(4. 2. 7)

至此，*A.P.C.S.*之運作終告一段落，其效果則於下一章中列舉出來。

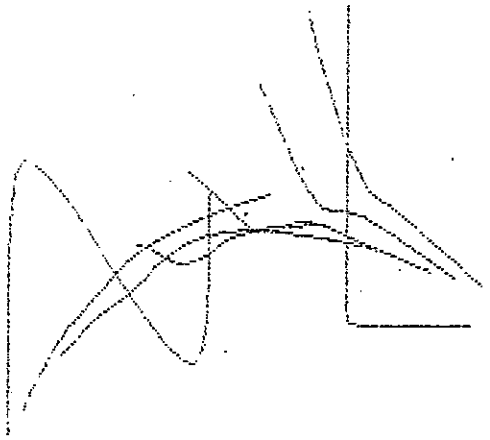
第五章　實例驗證

經過A.P.C.S.調整物體的動畫路徑後，所產生的輸出(Output)爲一些圖形控制點座標的檔案。將這些控制點代入產生三維物體形狀系統--O.S.G.S.[10]，可得到各個中間畫面，如此便完成動畫過程。

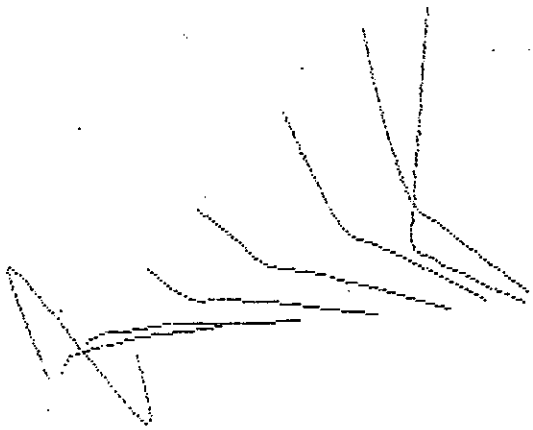
本章將以數個範例來驗證A.P.C.S.之效果。

範例一：

英文字母N變到L的動畫。假設第一張主畫面爲N，最後一張主畫面爲L，經過A.P.C.S.處理後，其結果如圖(5. 1. a)及圖(5. 1. b)。



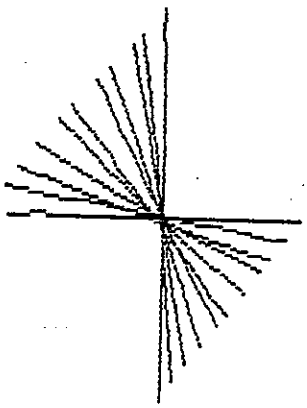
圖(5. 1. a)爲視角方向θ=0，Φ=90時的情形。



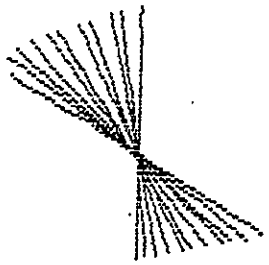
圖(5. 1. b)爲視角方向θ=30，Φ=40時的情形。

範例二：

一根棍子從垂直變水平的運動過程。其結果如圖(5．2．a)及圖(5．2．b)。



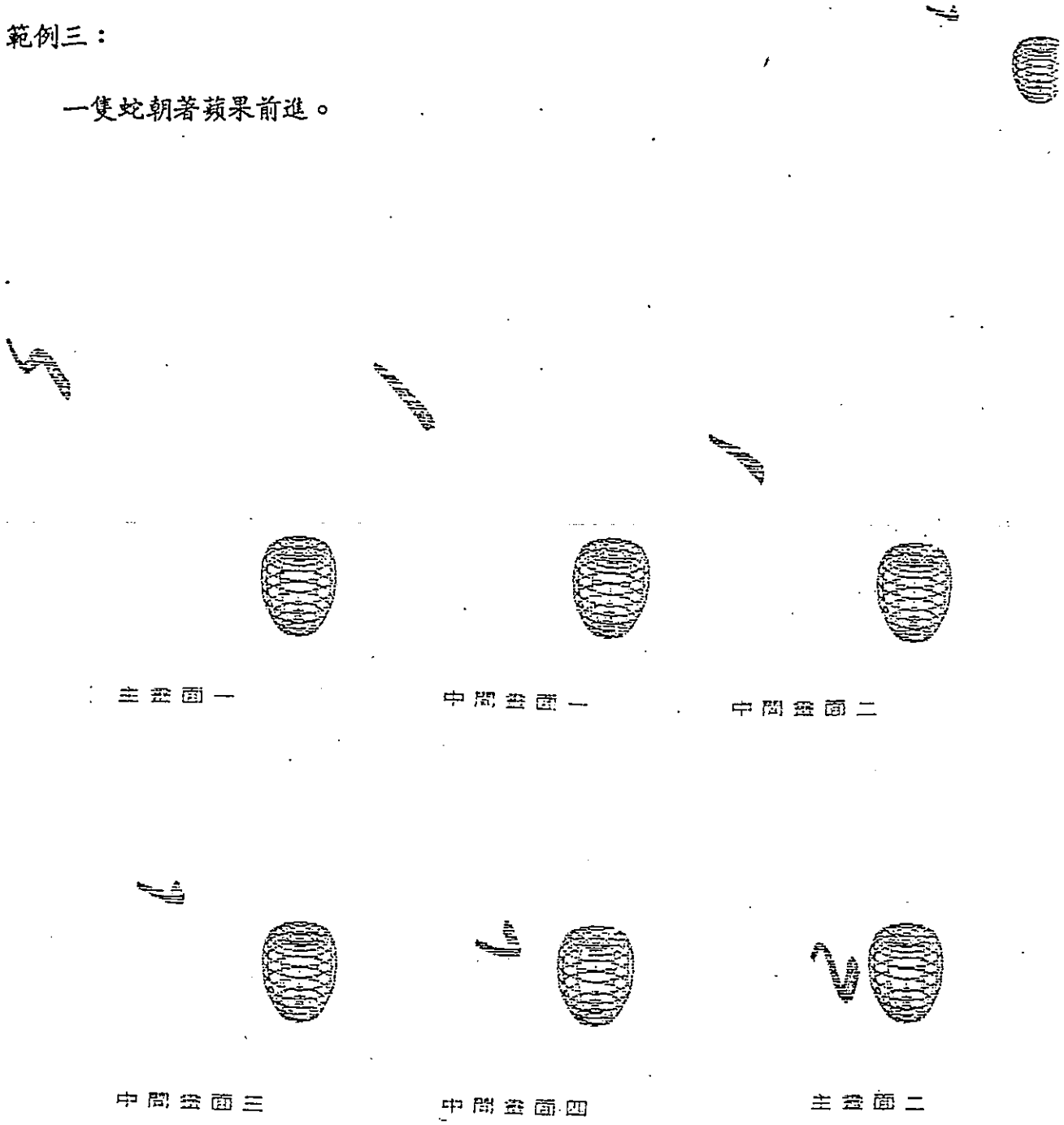
圖(5．2．a)爲視角方向θ=0，Φ=90時的情形。



圖(5．2．b)爲視角方向θ=40，Φ=50時的情形。

範例三：

一隻蛇朝著蘋果前進。



圖(5. 3)

第六章 結論

本研究所提出之A.P.C.S.系統架構其主要構想爲如何藉用電腦之助，來幫助畫家解決傳統動畫中最惱人的問題--必須繪出數以萬計的中間畫面，也許這些中間畫面與其他畫面之間可能只是小部份的差異而已。畫家只要在A.P.C.S.中輸入前後主畫面的資料進去，便可好整以暇的等待A.P.C.S.將中間畫面繪出來。如果這些中間畫面不滿足畫家所需，他隨時可進入A.P.C.S.中利用四線譜去做修改，直到滿意爲止。

根據實驗結果顯示，只要我們將A.P.C.S.中的t，c，b，a四線譜之值調好，所產生的中間畫面效果蠻好的，而且A.P.C.S.在人機介面的溝通方式上，是採取交談式(Interactive)的方法，提供了一個較友善的工作環境給使用者。

A.P.C.S.到目前爲止，僅發展到對單一物體的路徑控制。因爲僅是要驗證物體導向觀念及K-B 軟樑函數應用於動畫路徑控制上其理論的可行性與否，A.P.C.S.的發展證明了它的可行性。

至於未來的研究方面，可由下列幾點來考量：

(1) 爲使動畫能夠有生動、活潑的感覺，可將A.P.C.S.系統功能加以擴充，增加整個畫面的路徑控制功能。

(2) A.P.C.S.在調整路徑時，只顯示了控制點的路徑。可以考慮另外繪出前後主畫面之圖形，讓畫家瞭解該控制點與主畫面圖形間的相對應關係，此舉將有助於畫家對動畫路徑的全盤控制。

(3) A.P.C.S.由於是在PC上執行，速度較慢爲其缺點。我們可考慮利用一些平行處理(Parallel processing)的演算法則於該系統或是把A.P.C.S.置於如VAX/VMS或者如SUN Workstation等

大型電腦上執行，以便改進此缺點。

參 考 文 獻

[1]. Daniel Thalmann，Computer Animation Theory and Practice，Addison-Wesley，1982

[2]. D.H.U.Kochanek & R.H.Bartels，" Interpolating Splines with Local Tension，Continuity，and Bias Control"，ACM. Computer Graphics，Vol.18，no.3，pp. 33-41，July 1984

[3]. I.D.Faux & M.J.Pratt，Computational Geometry for Design and Manufacture，虹橋書局，1985

[4]. M.Stefik & D.G.Bobrow，" Object—Oiented Programming：Themes and Variations"，AI Mag.，Vol.6，no.4，pp. 40-62，1986

[5]. Newman & Sproull，Principles of Interactive Computer Graphics，虹橋書局，1985

[6]. 王凡．蔡國輝 譯，程式語言基礎，儒林圖書公司，1986

[7]. 李更生 編譯，交談式微電腦圖學，松崗圖書公司，1985

[8]. 吳建平 編譯，程式語言結構，松崗圖書公司，1986

[9]. 祝華健 譯，電腦繪圖的數學基礎，儒林圖書公司，1984

[10]. 簡水源，"供動畫使用之物體外形生成系統"，國立中興大學應用數學研究所資訊組碩士論文，1989