

TR-91-005

模擬氣球吹氣變形之動畫研究

參 考 書
不 外 借

中 央 研 究 院
資 訊 科 學 研 究 所
79.8.-1
圖 書 室

R5936

中研院資訊所圖書室



3 0330 03 000339 1

私立淡江大學資訊工程研究所

指導教授：鄭國揚 博士

模擬氣球吹氣變形之動畫研究
On Animating Balloon Deformation

嚴漢偉 撰

中華民國七十九年五月三十日

謝 誌

教憲吳君憲益，感其時，深感其煩，不厭其口，試以學生來論，也。一年半以來，利完意見，博士順寶，揚得提供，國研究所得，鄭本所，教授得，導使成，指與胡，謝正與，感與明，導明匪。

龍中，均為，文、均為，及、均為，以、均為，忠、均為，不、均為，許、均為，沛、均為，德、均為，曾、均為，龍、均為，首、均為，朱、均為，真、均為，攻、均為，揚、均為，學、均為，同、均為，在、均為，央、均為，黃、均為，促、均為，使、均為，感、均為，也、均為，專、均為，心、均為，最、均為，上、均為。

龍中，均為，文、均為，及、均為，以、均為，忠、均為，不、均為，許、均為，沛、均為，德、均為，曾、均為，龍、均為，首、均為，朱、均為，真、均為，攻、均為，揚、均為，學、均為，同、均為，在、均為，央、均為，黃、均為，促、均為，使、均為，感、均為，也、均為，專、均為，心、均為，最、均為，上、均為。

龍中，均為，文、均為，及、均為，以、均為，忠、均為，不、均為，許、均為，沛、均為，德、均為，曾、均為，龍、均為，首、均為，朱、均為，真、均為，攻、均為，揚、均為，學、均為，同、均為，在、均為，央、均為，黃、均為，促、均為，使、均為，感、均為，也、均為，專、均為，心、均為，最、均為，上、均為。

論文名稱：模擬氣球吹氣變形之動畫研究

頁數：102

校(院)所組別：淡江大學(學院)資訊工程研究所

組

畢業時間及提要別：七十八學年度第二學期碩士學位論文提要。

研究生：嚴漢偉

指導教授：鄭國揚 博士

論文提要內容：

電腦動畫研究的主要目的之一，在於如何建立一個理想的動畫環境，以使得縱使不懂得其背後所蘊涵理論的使用者，依然能夠以最自然的思考方式，很方便、很容易的表現出所希望的動畫效果。因為在現實世界中，許多物體的運動模式分析往往都必須引用繁複的物理學及數學原理。所以，要建立一個理想的動畫環境、以及應用範圍廣泛的動畫系統，是一項相當龐大而且複雜的工作，並非少數人在短時間內即可完成的。

本研究以氣球為例，嘗試建立一個能夠模擬產生氣球變形效果的動畫環境。從氣球的塑形、吹氣膨脹和受外力作用變形的處理、到真實影像的產生，做一完整的實驗。我們採用二維空間動畫中廣為使用的主架構圖法 (Keyframing) 做氣球的塑形，再以軟楔內插 (Spline interpolation) 的方式產生中間圖像 (Inbetweens)，以使得所塑造出的形狀更為平滑，符合氣球的實際情況。至於變形的處理，則是利用動力學的觀念，根據實際的現象，以基本的物理學原理模擬而得。同時，也考慮其顏色、透明度、背景景物、表面紋理、和光影等真實性因素的變化與影響，以使所表現的動畫影像更為逼真。

在本論文中，除了對所採用的方法及所引用的理論加以詳細分析介紹外，同時，也對系統的架構、使用者介面，實驗結果做一描述與評估。

Title of Thesis: On Animating Balloon Deformation

Name of Institute: Graduate Institute of Information Engineering
Tamkang University

Graduate Date: 06/90

Degree Conferred: Master

Name of Student: Han-Wei Yen

Advisor: Kuo-Yang Cheng

嚴 漢 偉

Abstract:

One of the main purposes for the research of 3D computer animation is to construct an environment - aiding users for designing microworld and populating them with interesting characters. Since objects differ in behavior and physical appearance, an environment has to be composed of many sub-environments. Each sub-environment deals with only one kind of objects, and may be even made up of many smaller sub-environments. On the other hand, for future composition and extension, the sub-environments have to be as modular and abstract as possible.

This research is trying to create an environment for emulating balloons' deformation. Shape modeling, deformation, rendering, and the user interfaces are considered. The shape is modeled by using sweeping representations from keyframing method, and inbetweens are generated via spline interpolations. Balloon's swelling is emulated to make visual illusions instead of physically-based simulation of elastic mechanics. During the deformation, the environmental parameters are changed to make it look more realistic. All these methods are discussed and assessed in detail.

目 錄

第一章 緒論	1
1.1 前言	1
1.2 研究動機與目的	2
1.3 研究內容	3
第二章 電腦動畫概述與相關研究介紹	7
2.1 電腦動畫概述	7
2.2 物體的分類與運動控制	9
2.3 變形效果研究介紹	21
第三章 系統模式與方法	23
3.1 物體塑形	25
3.1.1 軟狀物體的塑形	25
3.1.2 氣球的塑形	31
3.2 變形	36
3.2.1 氣球吹氣變形現象分析	37

3.2.2	模擬方法	38
3.3	影像的真實性處理	49
3.3.1	光影模式	54
3.3.2	Texture Mapping	58
第四章	數學方法	62
4.1	Bezier 曲線	62
4.2	產生通過給定點的曲線	66
4.2.1	曲線的相連	67
4.2.2	產生封閉曲線	70
4.2.3	產生開放曲線	71
4.3	彈性理論基礎	72
第五章	系統架構	77
5.1	系統架構與流程	77
5.2	系統參數	80
第六章	實驗結果與評估	84
6.1	實驗結果	84
6.2	評估	84

第七章 結論與展望 92

參考文獻 94

第一章 緒論

1.1 前言

” 動畫 (Animation) 其實並無任何神秘可言 . . . ; 然而, 就如同世界上其他看來簡單的事一般, 待真正實行起來才會發覺其困難之所在。” 此語出自一位華德迪斯奈動畫工程師的口中〔1〕, 直接明白的道出了動畫發展的哲學。

雖然僅僅一小段幾分鐘的動畫, 往往即需耗費數月的時間才能完成, 遑論所花的人力、物力。但不可諱言的, 動畫帶給人的感覺卻是最直接、有效, 也是最吸引人的。隨著資訊科學的快速蓬勃發展, 結合電腦設備與技術的電腦動畫 (Computer animation), 不但大幅降低了動畫的發展成本, 更使得設計者能夠充份的發揮想像力, 很方便的表現出心中的構想, 而將動畫的發展帶入了一個新的紀元。

最初, 三維空間電腦動畫的技術主要是將傳統的, 廣泛應用於二維空間動畫中之主架構圖法 (Keyframing) 予以擴充至三維空間, 利用插分 (Interpolation) 的方式, 產生中間的圖像 (Inbetweens)。至今仍有許多的動畫系統是採用這種方法〔2,3,4〕。近幾年來, 許

多的動畫研究則著重於真實性的表現方面。除了物體本身的真實效果，例如，光影的影響（Lighting）、背景景物的表現（Background scenery）、和物體表面性質等的處理，以及運動的控制之外，物體運動時的各種現象也希望能以最接近真實情況的方式表現出來。因此，則必須針對物體的許多物理特性加以考慮，諸如，物體的運動模式、運動時各部分隨時間變化的關係、環境對物體的影響等等。當然，所考量的特性愈多愈精確，呈現出的動畫效果也就愈逼真。但相對的，系統所需的計算也就愈多、複雜度也愈高。而這種模擬（Simulation）式的動畫則成為一種綜合了電腦動畫、物理學、數學、乃至於藝術等學門的研究領域。

此外，動畫研究的另一個重點則在於創造一個理想的動畫環境。在此環境下，不但任何使用者都能夠很容易的表現出所要求的動畫效果，而且動畫系統本身也能夠很方便的擴充、管理、與維護。

1.2 研究動機與目的

本研究主要係配合行政院國科會一項電腦繪圖研究計劃中關於電腦動畫方面的子計劃。而為了探討電腦繪圖及電腦動畫的相關理論，並研究動畫產生的實際過程，我們以氣球為例，從氣球的塑形，真實影像表現的處理，到動畫的產生（亦即氣球的吹氣變形過程），實際

做一實驗。一則以驗證所學的理论，瞭解其於實際應用上可能發生的困難，同時研究可能的解決方法，藉以培養研究的能力；另一方面則以建立一個簡單的、特定用途的（氣球吹氣變形）的動畫環境為目標，讓即使不明瞭相關理論的使用者，也能夠輕易的完成所欲表現的動畫。

因為在現實世界中，許多物體的運動模式分析往往都必須引用到深奧的物理學以及數學方面的理論，甚至有些還是尚未為人所知的。所以，要建立一個應用範圍廣泛的動畫系統、以及理想的動畫環境，是一項相當龐大而且複雜的工作，並非少數人在短時間內即可完成的。本研究僅以某些特定物體為對象，研究其真實影像的產生，並模擬其動畫效果，希望經由模組化（Modularization）、以及抽象化（Abstraction）的設計架構，讓以後的研究者能夠很方便的予以擴充或採用，同時也提供相關的研究者作為參考。當然，更盼望能有更多的研究者一同投入此一領域的研究，以使得電腦動畫的發展能夠更完備、更迅速。

1.3 研究內容

本研究以氣球為例，嚐試建立一個能夠模擬產生氣球吹氣變形效果的動畫環境。同時，從氣球的塑形、吹氣膨脹以及受外力作用變形

的處理、到真實影像的產生，做一完整的實驗。

物體的塑形和所建立的動畫模型是一個動畫系統的核心（當然，其中還包括了畫面的真實性處理）。往上發展則為使用者介面等高階系統介面的建立，以使得使用者能夠以最自然、最簡便的方式，完成所要求的動畫。往下游發展則為動畫系統的應用。電腦動畫的應用，主要有影片製作與模擬兩大領域。在影片製作方面，包括了卡通、電影、廣告等等，所著重的則是精緻美觀的影像以及充滿豐富想像力的動作。而在模擬方面，重點則在於如何真實的將所發生的任何變化表現出來。當然，最重要的還是在於模擬所得的數值結果。

氣球的塑形基本上是屬於幾何塑形 (Geometric modeling) 的問題。因為氣球的外形並沒有一定的規則，甚至於每個氣球的形狀都各不相同。而為了能夠滿足此種要求，同時也兼顧使用者的方便性，所以，本系統採用得自於傳統電腦動畫中的掃掠法做氣球的塑形。亦即，由使用者在一給定的路徑上，自行於適當的位置定義出氣球在該處的橫斷面形狀，即動畫中所稱之主圖像 (Keyframes)。系統就根據這些主圖像，於縱向的方向做 Spline Interpolation，產生主圖像間的中間圖像 (Inbetweens)，建構出平滑的氣球外形，以符合氣球實際的狀況。

在本系統中，動畫的效果是以氣球的吹氣變形，以及氣球受外力作用的變形過程表現的。一般而言，於動畫模型中引用運動學、動力

學、或是幾何學上的約束 (Kinematic ,dynamic or geometric constraints) 是最常用的三種方法。藉著已定義好的主圖像，然後以內插的方式自動產生中間圖像，就是一種運動學的動作控制法。然而，要模擬產生最真實的自然現象，則必須借助物理學原理或是動力學的理论，考慮物體受外力或力矩的作用後所發生的反應。此外，若以物體的組成架構為根據，針對組成物體的各個幾何元素 (Geometric primitives) 做動作的處理，有時更能真實的將物體的反應，以及物體的架構表現出來。

氣球吹氣或受力後所產生的外形變化，理論上而言，是屬於材料力學和彈性力學方面的問題。以此來分析，則必須牽涉到非常複雜的彈性力學原理與數學運算。對我們而言，將有實質上的困難。所以，我們將問題簡化，以能夠表現出近似真實情況的效果為主。在實際的做法上，本系統將氣球視為由所有圖像上相鄰的控制點間彼此以彈簧相連而成，而氣球的吹氣變形則相當於，每一個質點在法向量方向上，同時都受到一個向外的彈簧的拉伸，而使得氣球膨脹變形。氣球內部所承受的力（壓力），以及氣球表面上質點的彈性係數，則為系統計算變形量大小的主要依據。雖然此種做法並未實際模擬氣球的動力學原理，而且也沒有很精確的考慮變形過程中的時間因素，但因變形的效果係考量外力對物體的作用而產生，所以，廣義而言，仍屬於一種動力學的動畫模型。另一方面，系統則利用反向動力學 (Inverse

dynamics) 的觀念，先計算物體受擠壓後應有的形狀，然後，根據變形量的大小，求出物體所承受的外力，再算出整個物體變形後的形狀，以模仿氣球受擠壓變形的情况。

當然，物體的真實性表現，除了外形之外，還必須包括光線的考慮 (Lighting)、光影的影響 (Shading)、顏色及透明度的變化、背景景物的效果、以及物體表面紋理的處理等等。所幸，在這方面我們已有軟體工具的輔助，使我們得以儘可能的依需要產生所希望的效果，做更高階部分的處理，不必再從頭發展。

在系統的整體架構上，我們的目標是儘可能的將之模組化與抽象化，以便將來的研究者易於組合與擴充。所以，在設計上系統將整個環境中和使用者有關的參數獨立出來，讓使用者可以最接近實際情況、最簡便的方式，調整相關的參數，產生所欲表現的動畫效果。

本論文共分為七章。在第二章中，將分別針對電腦動畫以及和我們的研究主題相關的研究做一介紹與說明。第三章中則詳細分析討論本研究所採用的系統模型、所考慮的因素、以及處理的方法。而關於所引用到相關的數學與物理原理，則於第四章中說明。第五章則為系統架構、系統流程、系統的需求、以及使用者介面等部分的介紹。第六章則為研究的結果與評估。最後，則為結論與展望。

第二章 電腦動畫概述與相關研究介紹

2.1 電腦動畫概述

電腦動畫，簡單來說就是利用電腦製作一連串的畫面，並將這些畫面以一定的速度連續顯示，藉由人類視覺暫留的特質而產生動感。而依據電腦在動畫過程中所扮演角色的不同，電腦動畫系統一般可分為電腦輔助動畫與模型動畫兩大類〔5〕。

1. 電腦輔助動畫 (Computer-assisted animation)

顧名思義，就是以電腦來輔助如卡通等的動畫製作，包括圖像的輸入、中間圖像的產生、物體運動路徑和動作的描述、著色、背景的处理、以及音效的配合等等。

2. 模型動畫 (Modeled animation)

主要是用以塑造物體的立體圖形，同時處理物體於三維空間中的運動。因三維空間物體影像的處理，諸如光影 (Shading)、遮線和遮面的消除 (Hidden line and hidden surface removal)、陰影 (Shadow)、及表面紋理的處理 (Texture) 等等都非常複雜、費時。所以，這些過程若沒有電腦的輔助將難以完成。

一般而言，三維空間電腦立體動畫的產生有物體塑形、動作的描述與同步處理、及影像的表現三個主要步驟。

1. 物體塑形 (Object modeling)

就是要產生所欲處理物體的立體圖形。通常有兩種模式可供採用：

(1) 由三維空間線段連接而成的線架構模式 (Wireframes) 模式。或是，

(2) 由二維或三維的曲面組合而成的實體 (Solids) 模式。

這兩種模式的物體均可利用數位掃描 (Digital scan)、圖形編輯 (Graphics editing)、或是程式產生 (Programming) 等方法來建造。

2. 動作的描述與同步處理 (Motion specification and synchronization)

動畫效果的產生是因為物體在不同的時間會有不同的位置、大小、或形狀。所以，如何將物體的這些變化隨時間的推移表現出來，往往也就成為電腦動畫中最重要的一環。而當有許多物體同時都在運動時，則還必須考慮到物體間同步的問題。此外，若以程式的方式預先設定 (Programmed) 觀測點和時間的關係，用以模擬虛擬攝影機 (Virtual camera) 的功能，隨著物體的運動而改變位置 (做 Viewing transformation)，也可使得動畫的效果更為生動。

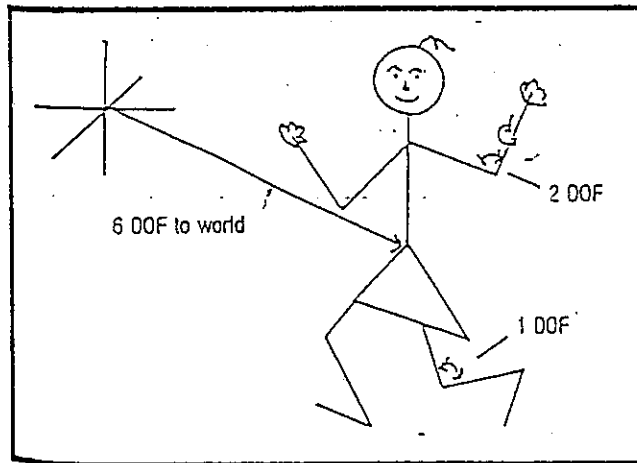
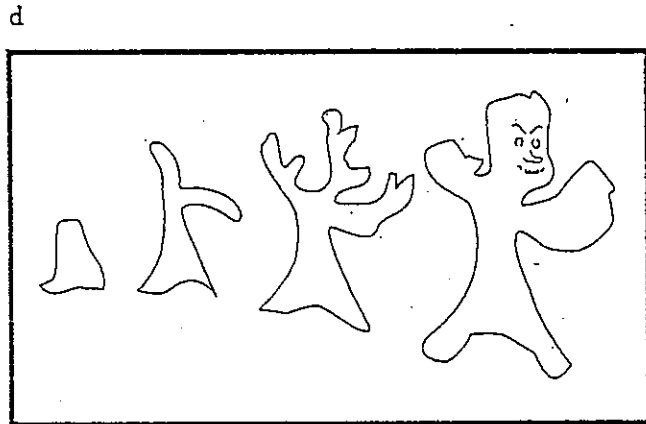
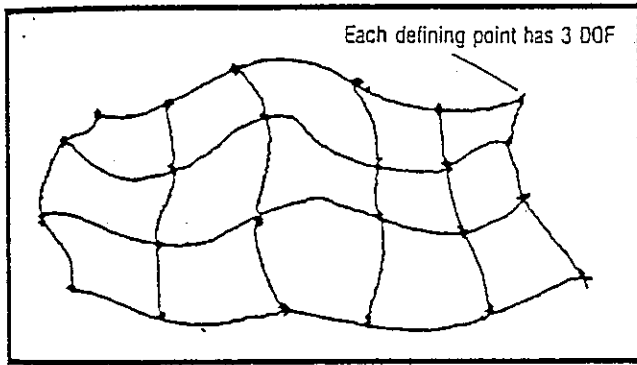
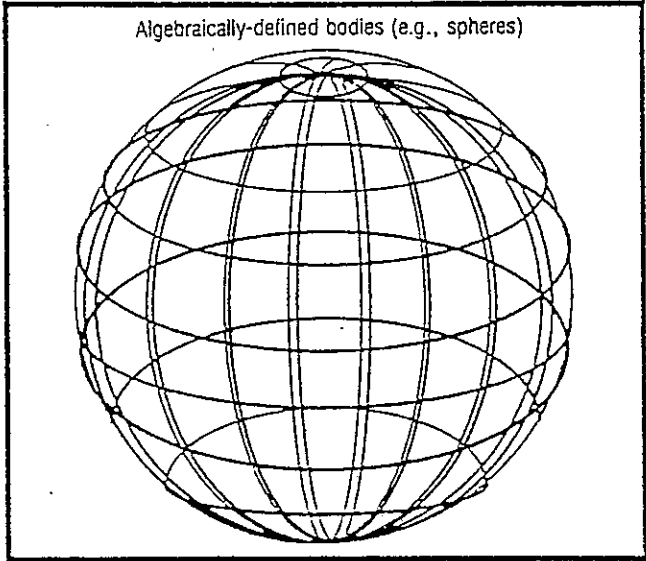
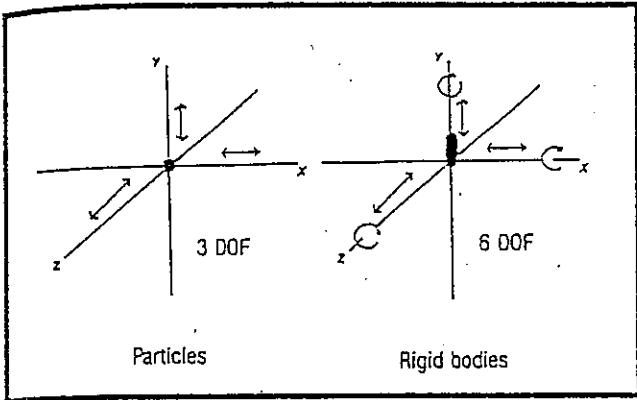
3. 影像的表現 (Image rendering)

此步驟的主要目的就是要產生接近真實的影像。藉由繪影 (Shading)、遮線遮面的消除、透明度的表現、陰影、表面紋理等的處理，以使得所表現的影像看起來更逼真好看。

以上的三個步驟中，尤以動作的描述與同步處理最為重要。因為若是只有塑形和真實性的表現則將只是電腦合成的影像 (Computer synthesized images) 而不成為動畫了。物體的塑形及影像表現方面，在先進國家中已有相當長時間的研究發展，而且在電腦動畫中應用起來也並無太多的困難。但在物體的運動控制 (Motion control) 方面則不然。隨著電腦合成影像真實性程度的要求愈來愈高，在立體動畫中，物體的運動也希望能愈真實愈好。因此，如何提供更自動化及更高階 (more automatic and high level) 的運動控制方法，就顯得益發的重要了。

2.2 物體的分類與運動控制

物體依其運動的方式可分為質點、剛體、柔軟物體、關節相連物體、和變形物體及其他等幾類〔6〕。通常物體的運動都是以自由度 (Degree of freedom) 來描述。自由度就是描述物體的各部分位置時所需設定的獨立變數的數目。如圖 2.1。



e

2.1

Degrees of freedom of motion: (a) particles and rigid bodies, (b) flexible bodies, (c) articulated bodies, (d) miscellaneous bodies, (e) metamorphosing bodies.

1. 質點 (Particles)

一個質點就是空間中位於座標 (x,y,z) 上的一點。所以，一個質點最多可有三個自由度。而利用許許多多質點的組合，則能夠塑造出更多樣化的形狀。Reeves [10,11] 等就曾採用質點系統 (Particle System) 來模擬諸如火燄、爆炸、森林等等的自然景觀 [6,7]。

2. 剛體 (Rigid bodies)

如果我們將物體視為由許許多多的點所構成，則剛體就是這些構成物體的點彼此間的相對位置固定，縱使物體運動也不會改變。剛體的運動可以6個自由度來描述：(以Cartesian座標系統來看)分別為X，Y，Z軸方向上的平移和旋轉。因為描述剛體的運動所需的變數較少，所以很多動畫系統都以剛體做為主要的處理對象 [12]。

3. 柔軟物體 (Flexible bodies)

相對於剛體，形成柔軟物體表面的點，其間的相對位置則是可能隨著時間而變的。事實上，若物體外形是由許多的多邊形組合而成，這些點即可視為是多邊形的端點；但若將這些點視為形成物體表面之曲面的控制點，則更能表現出物體的柔軟程度。如果一個柔軟物體是由P個控制點定義而成的，則其具有3P個自由度。

4. 關節相連物體 (Articulated bodies)

關節相連物體是由許多的小段 (Segments) 在關節處相連接而成。例如，人體的骨架就是許多的骨頭組合成的。因為相連的各段必須維持這種相接的關係，所以物體運動時都會受到彼此間在某種程度上的限制，而使得自由度減少。整個物體的自由度即為所有關節點的自由度總和。目前為止也有許多針對關節相連物體的塑形與動畫方面的研究〔13,14,15,16〕。

此外，關節相連物體也可能是柔軟的，亦即是由許多的柔軟段 (Flexible segments) 相連而成的。在這種情況下，物體有兩類可能的運動方式：一是柔軟段的運動，也就是定義此柔軟段的控制點的運動；一是關節部份的運動。而整個物體的自由度則為所有關節部份和柔軟段部份自由度的總和。

5. 變形物體及其他 (Miscellaneous and metamorphosing bodies)

物體的外形也可能是由代數式 (Algebraic equation) 定義而成的。例如，給定球心和半徑即可定義出一個球體。此類物體的自由度則須視其代數式的性質而定。

若物體在運動過程中，不但外形發生變化，自由度也隨之改變，則情況就複雜多了，此類的物體即稱為變形物體。例如，以碎形 (Fractal) 來塑造山的形狀〔17〕；成長中的植物〔18〕；以及由質點系統構成的物體〔11〕等。

在動畫系統中，物體動作的控制方法若依使用者對系統的控制方式來分，可分為交談式 (Interactive)、劇本式 (Scripted)、以及兩種混合三種方式；若依控制的層次來分，則可分為低階 (Low level)、高階 (High level)、以及中間層次三種〔6〕。

1. 交談式控制法

使用者定義了物體的動作或路徑之後，系統就立刻將所產生的效果顯示出來。使用者可以諸如旋鈕、滑鼠等的交談式裝置更改所定義的動作，直到達到要求的效果為止。最常用的交談式控制法有三種：

(1) 主架構法 (Keyframing)

使用者將一連串物體運動時的位置、形狀和時間關係定義好，系統即據此以插分 (Interpolation) 的方式，計算出中間的圖像。其中又有影像主架構動畫 (Image-based keyframe animation) 〔7〕及參數主架構動畫 (Parametric keyframe animation) 〔8〕兩種基本的處理方式。

在三維空間電腦動畫中，主架構法是一種非常方便而且效果不錯的動作控制法。其最大的優點為使用者可對物體的運動獲得完全的控制，對於畫面中所有物體的動作也能有全盤的瞭解，但因此使用者也必須定義物體每個自由度的變化 (資料量龐大)。是屬於一種低階的控制方法。

(2) 路徑指定法 (Path specification)

使用者只定義物體的運動路徑。此種方法較不易掌握所有物體在某一時間的狀態，仍屬低階的控制方法。對於單一物體而言，在主架構圖法中，使用者定義的是物體在某一時間上的位置及狀態，對連續的時間而言，這些主圖像是離散的 (Discrete)；而路徑指定法所定義的則是一條連貫的運動路徑 (Coherent path) [62]。

(3) 函數控制法 (Control functions)

物體的每一個自由度均以一個時間—位置的函數來控制。雖然是較高階的控制方式，但仍然可以輕易的控制到某個自由度的狀態。而且其他的控制方法也可轉換為此種方式。不過，對使用者而言，較不易通盤明瞭所有物體的狀況。

2. 劇本式動畫 (Scripted animation)

系統提供一套動畫語言 (Animation language) 讓使用者用以控制物體的動作；使用者則以程式的方式描述整個動畫過程中各物體的運動及表現方式，然後由系統直譯而產生動畫。這種劇本式的方法提供了使用者和系統間一種高階的溝通方式，而且使得物體動作的控制更為簡便。但相對的也降低了使用者對物體的控制程度。

3. 高階控制法 (High-level methods)

在某些情況下，使用者並不希望也不需要控制到物體動作的每一個細節 (自由度)，因此就需要有較高階的控制方式。常用的高

階控制法有參數控制、有限狀態機、以及指令庫等幾種。

(1) 參數控制法 (Parametric motion control)

使用者藉著參數來改變物體的狀態和動作。這種方法讓使用者可很方便的以少數的指令控制複雜的動作；另一方面，此種方法則可很容易的以程序的方式來完成。因此，參數的選擇往往就成為首要的考慮因素。

(2) 有限狀態機 (Finite state machines)

這是一種適合用以描述和控制具有重複性及協調性動作的方法。例如，Zeltzer [9] 即採用四個狀態的有限狀態機控制關節相連物體的行走 (Walking) 動作。

(3) 指令庫 (Command libraries)

系統提供特殊的指令供使用者做動畫的控制，而這些高階的指令則是透過指令庫中許多低階的控制動作完成的。

以上所介紹的方法都是由使用者主導 (User-directed)，控制動畫過程中的動作變化。但當物體的動作很複雜，或是物體間有很複雜的相互關係時，若仍依靠使用者來處理，不僅費時、費力，而且還容易出錯。因此，就需要更自動化甚至具有智能的控制方法。而要達到這個目的，除了動畫本身的知識與方法外，尚需引用諸如機器人學 (Robotics)、人工智慧 (Artificial intelligence)、和物理學等方面的專門知識。

此外，Zeltzer〔8〕則根據控制模式將動畫系統區分為導引（Guiding）、動畫者（Animator）、以及任務（Task）三個階層。

1. 在導引階層的系統中，物體所有行為的細節都必須由使用者定義，系統並不提供任何動作控制上的輔助。如前所述，此類系統所能產生的動畫效果將受到限制。使用主架構法的系統即屬此類。

2. 而在動畫者階層的系統中，物體的動作則是採用演算法（Algorithm）的方式來描述。因為這種系統可以非常結構化及抽象化（Abstraction）的方式完成，所以對使用者而言，不但使用上更方便，同時也能表現出更豐富的動畫效果。例如，GRAMPS〔20〕。

3. 至於在任務階層的系統中，物體的運動則是以物體的動作（Events）以及物體和周遭物體或環境間的關係（Relationships）來描述。系統必須自行安排物體運動的時程、運動時的變化，同時將物體應有的特徵表現出來，產生真實的動畫效果。這些則必須仰賴一個包含物體的特性、不同環境對物體的影響等資料的知識庫（Knowledge base）的輔助才能完成〔21〕。

此種方法是屬於最高階層的動作控制方式。任何使用者均可輕易的產生所要求的動畫。但是，目前並沒有一個真正一般用途（General purpose）的任務階層動畫系統。

在處理物體的運動時，某些物體特性上，或是動作上約束（Constraints）的考量是不可或缺的。這些約束可能限制了物體的動作，

甚至可能會減少物體在某處的自由度。(如圖 2.2)。隨著物體型態和動作的不同，物體所受到的約束也不盡相同。而且，這些限制也可能以不同的形式存在。譬如，速度、加速度的約束；能量的約束 (Energy constraints)〔22〕；運動學上的約束 (Kinematic constraints)〔23〕；動力學上的約束 (Dynamic constraints)〔24〕；時間空間的約束 (Spacetime constraints)〔25〕等等。

對動畫系統的使用者而言，系統中運動控制的自動化程度愈高，負擔也就愈輕，動畫的製作也就愈容易。但連帶的系統本身的複雜度也將增加，同時，使用者對運動細節的控制能力也會減少。不過，這些考慮乃屬系統製作上及應用上的權衡取捨。接下來我們介紹幾種常

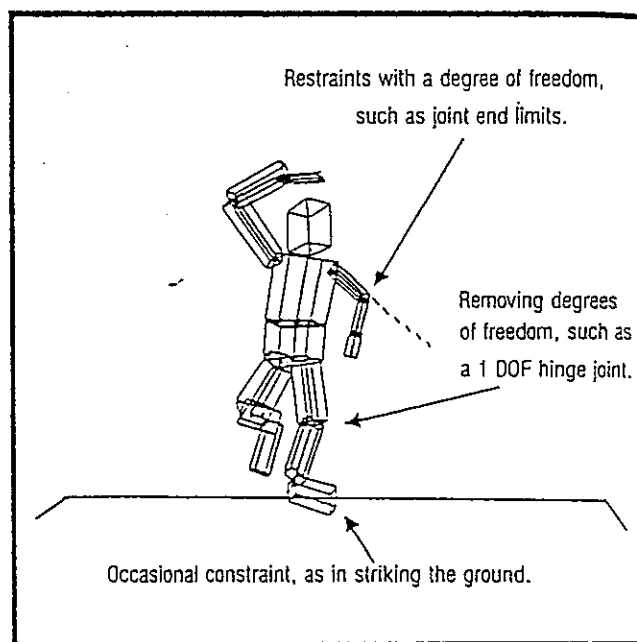


圖 2.2 Constraints on motion.

見的物體運動自動控制的方法。

1. 動力學分析 (Dynamic analysis)

一般的運動控制方法多屬於運動學 (Kinematic) 上的，亦即只考慮物體在每個自由度上的位置和時間的關係；而以動力學 (Dynamic) 分析來控制物體的運動，則必須考量造成物體運動的原因。諸如，物體所受的外力、力矩、物體的質量、以及物體本身質料等等。最簡單的動力學控制方法，就是利用牛頓第二運動定律：

$$F = m a$$

其中， F 為作用於質點上的力； m 為該質點的質量； a 則為質點受力後所產生的加速度。

若是已知物體所受的外力 F ，及物體質量 m ，則可以求出加速度，計算出物體運動過程中在每一個時間的位置。這種問題即稱為正向動力學問題 (Forward dynamics problem)。但若物體的運動必須要滿足某些約束，則需在已知物體的行為下，反算出物體所應承受的外力，這類的問題則稱之為反向動力學問題 (Inverse dynamics problem) [24]。

動力學的運動控制方法能夠真實的反應出物體實際的運動狀況，因而得以產生近乎真實的動畫效果。但其最主要的問題則在於，此種分析往往需要複雜的物理學與數學的演算方法，計算也相當費時。更重要的則是，應如何將這些動力學上的參數和實際的動作結合起

來。然而，由於所獲致的逼真效果，這種模擬式的動畫近年來已廣為動畫研究者所採用。Barzel 和 Barr 即建立了一套剛體的動力分析系統〔24〕；而 Wilhelms〔13〕，Armstrong〔26〕，Isaacs〔23〕則分別採用動力分析方法來控制關節相連物體的運動；Terzopoulos〔27〕則以此方式產生物體彈性變形的動畫效果。

2. 運動學分析 (Kinematic analysis)

以運動學法則做物體運動的控制，通常有兩種方法：一為正向運動學法則 (Forward or direct kinematics)，就是在已知物體主要部份位置的情況下，欲求出整個物體在空間中的位置和形狀；另一為種則為反向運動學法則 (Inverse kinematics)，亦即已知物體的位置，而要決定物體每一個自由度的狀態。

利用反向運動學法則並不需要如動力學分析般繁複的運算，但仍能將物體和周遭環境的關係表現出來，而得到近似真實的動畫效果。其主要的困難則在於如何求出滿足所有條件的解，以及如何找到一個最適當的解。若能根據一些真實性 (realistic) 及自然性 (Natural) 的規範 (Criteria)，則將有助於此種問題的解決〔6〕。

傳統的動畫模型幾乎均採用運動學分析的方式控制物體的動作。通常是利用主架構法定義出物體於某些時間上的狀態，以表現出物體的動畫效果。此種方式雖然較為簡單，但若要產生十分自然、真實的效果，就必須要具備專門的知識了〔1〕。Girard, Maciejewski

〔28〕即採用反向運動學方法控制關節相連物體的運動。

3. 路徑設定及碰撞處理 (Path planning and collision detection)

使用路徑設定演算法做運動控制，就是希望能夠只定義物體的起始和最終位置，系統即自動依據物體所處的環境及所受的約束而產生整個運動過程。此種方法最重要的部份在於碰撞及障礙物的處理。如何找到能夠避開所有障礙物的路徑、及碰撞後物體間的反應等，都是影響動畫效果真實與否的主要因素。

Moore, Wilhelms〔29〕詳細討論了電腦動畫中物體碰撞的問題，同時提出了兩個解決的方法。

4. 演算法控制法 (Algorithmic control)

演算法控制法係指，使用者將系統所提供的指令以程式的方式輸入，系統即根據這些指令將動畫產生出來。使用者可透過參數改變動畫效果，而無須實際去改變物體每個自由度的狀態。有許多的電腦動畫系統都提供了簡單的演算法運動控制法。如立方體的旋轉、行星的運行等。

此外，Stochastic control 也是相當常被採用的一種演算法控制法。這種方法是將物體中某部分自由度以Stochastic process 來控制，使其產生近似亂數 (Pseudo-random) 的效果，以模擬真實景物或效果。例如，以碎形做山脈、樹葉等自然景的塑形〔30〕；或是火燄、爆炸等效果的模擬〔10〕等等。

2.3 變形效果研究介紹

物體變形的處理常常也是決定動畫品質的重要因素之一，因為自然界中，除了剛體之外，幾乎所有的物體運動時（除了移動和轉動）都會發生或多或少的形狀變化。而物體變形的原因、變形的方式、所產生的影響等，則是和物體本身的特性息息相關，各不相同。

Barr〔31〕利用 Jacobian 矩陣的運算，改變物體表面的切線向量和法線向量，使物體產生扭轉、彎曲、尖細等的局部（Local）或是整體（Global）的變形。Sederberg〔32〕則提出了一個利用 Bernstein 多項式控制物體表面控制點位置，而能夠將基本的實體和曲面元素任意變形的 FFD（Free-Form Deformation）模式。同時，使用者尚可自行決定變化後物體體積的改變與否以及物體外形的連續程度。這些方法主要都是用於物體的塑形，配合上其他的塑形方法，則能夠增加幾何塑形的能力以及範圍。

至於在動畫的應用上，Wyvill等〔33〕採用了力場（Field）的觀念，藉著控制點間距離的變化，改變控制點力場的影響範圍，而造成物體形狀的變化，模擬軟狀物體的動畫。Weil〔34〕則根據布料的物理性質，以近似的方式表現布料的懸垂、皺摺等效果。而為了能夠產生出較為真實的動畫效果，近年來電腦動畫的研究有許多都是以依據物體實際物理性質的動畫模型（Physically-based model）為主。

Terzopoulos 等〔27〕即根據彈性理論，以橡皮、海棉、彈簧、紙等材料為例，針對彈性材料，建立了一套處理曲線、曲面、以及實體元素彈性變形的動畫模型。隨後，Terzopoulos 等〔35,36,37〕又以較為簡化的線性彈性 (Linear elasticity) 理論，將之擴充至諸如塑性 (Plasticity)、黏彈性 (Viscoelasticity)、斷裂 (Fracture) 等非彈性變形現象的處理。

這種依物體實際物理性質而定的動畫模型，採用動力學分析的方法，計算出所描繪的物體在受到外力、外在環境、約束、乃至於其他物體碰撞等情況的影響下，物體應有的形狀變化。而且，在變化的過程中間，甚至可完全不需要使用者的介入。此種動畫模型，同時處理了物體的運動以及運動過程中的形狀變化。雖然所採用的演算法都十分複雜，但所獲致的動畫效果卻非常真實，在模擬上尤其有用。

關於 Terzopoulos 的彈性模式，因和本研究有直接的關係，將於第三章中詳細介紹。

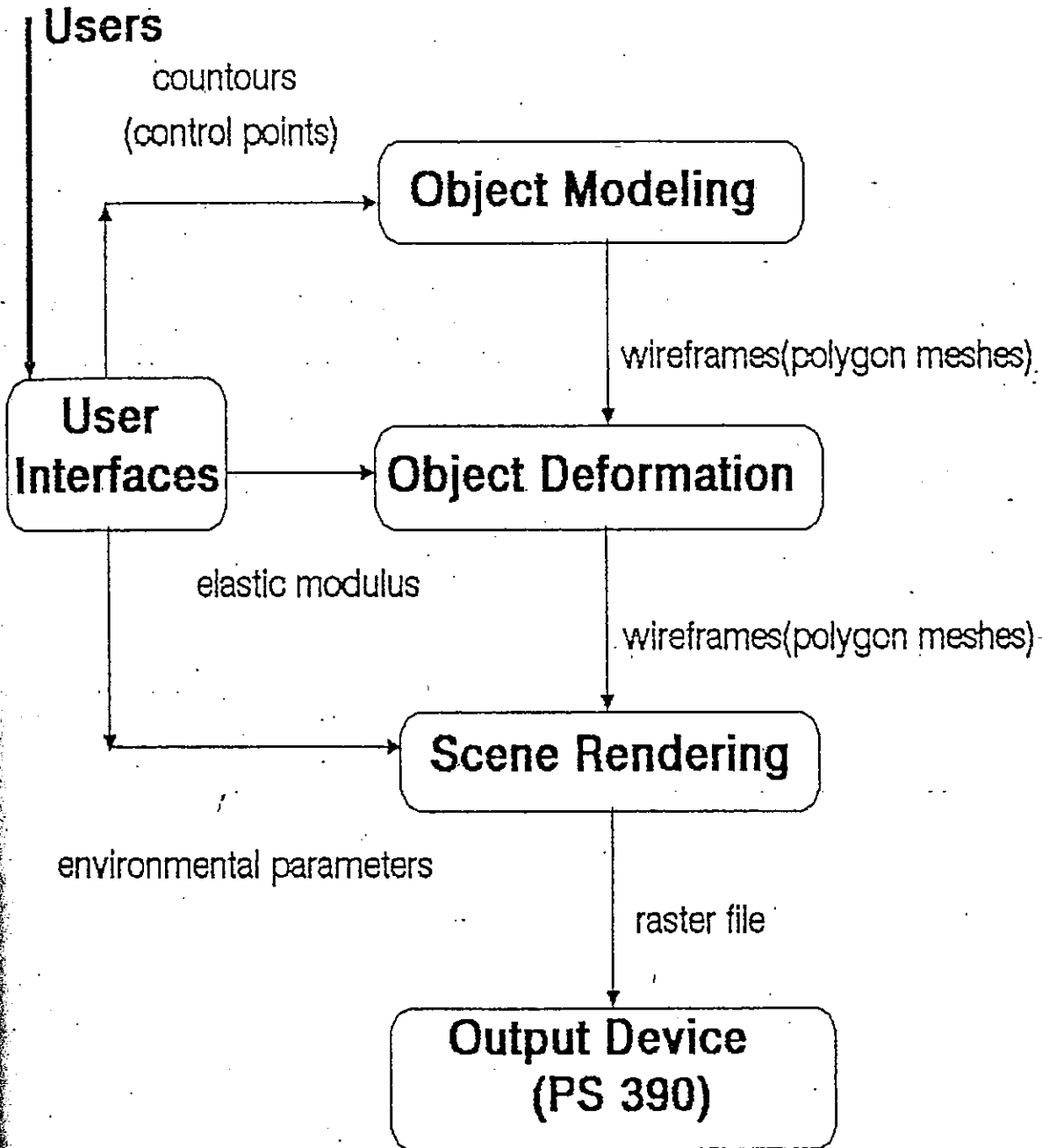
第三章 系統模式與方法

根據第二章中所述之立體動畫的產生步驟，同樣的，我們也將氣球吹氣變形的動畫分為塑形、變形、以及真實性表現三個主要步驟。如圖 3.1 所示。

使用者透過使用者介面，依需要設定系統所提供（包括塑形、變形、及真實性表現三部份）的可調參數。首先，系統由塑形參數所定義之橫斷面，產生氣球初始形狀的線架構（Wireframe）圖形。然後，根據演變法則（Laws of evolution）依序產生出吹氣變形後的氣球外形（此時仍為線架構圖形）。最後，再將所有的圖形做真實效果的處理。將這些變化過程以一定的速度依序連續顯示，即為氣球吹氣變形的動畫。

在以下各節中將分別詳細的討論每一個步驟所採用的方法，至於使用者介面的部份則於第五章中說明。

SYSTEM MODEL



3.1 物體塑形

3.1.1 軟狀物體的塑形

廣義的來說，氣球是屬於軟狀物體 (Soft objects) [33] 的一種。若是以第二章中所述，以物體運動觀點的分類而言，氣球則是一種柔軟物體。因為軟狀物體會隨著所處之外在環境的變化而有不同的外形表現，所以在塑形時，尚須考量物體運動 (變形) 後的塑形表現、以及控制變形的難易程度。此種塑形的方式稱為 Phenomena modelling [38]。程序模式 (Procedural model) [39] 則是一種非常適合用以履行 Phenomena modelling 的方法。此種模式主要係由程序及參數二部分所構成；只要藉著參數的改變，即可得到不同的物體外形。若加入時間因素，則能夠很容易的將物體連續的變化表現出來。

一般而言，適合用以做軟狀物體塑形的程序模式有：質點系統、場力函數法 (Field functions)、曲面的組合 (Composite surface patches)、以及掃掠法 (Sweeping representations) 等幾種。

1. 質點系統

質點系統就是以一群質點的集合來表現物體外形的方法 [10]。每個質點都被賦與一些基本的屬性，例如，位置、速度、形狀、

大小、顏色、生命時間等等。每經一段時間，就會不斷的有質點重新產生、移動位置、改變形狀、顏色、或死亡，因而能夠塑造出不規則形狀的物體。根據 Reeves [10] 的描述，物體變化過程中的任何一個圖像 (Frames) 可以以下的步驟產生：

- (1) 產生新的質點
- (2) 賦與這些新質點屬性
- (3) 去除生命時間已結束的質點
- (4) 剩餘的質點依其屬性改變狀態
- (5) 將該圖像做真實性的處理

質點系統是一種適合形狀不規則，且不易予以定義，但又會隨時間而改變的塑形方法。而以程序模式控制物體的變化，許多屬性均可以亂數控制，所以非常簡單方便。但相反的，對於形狀規則的物體，就需要更多的處理。而且因為構成物體的質點數目非常多，無論在形狀的控制、屬性的設定、以及真實性處理上都十分複雜、費時。所以，並不適合用以作氣球的塑形。

2. 場力函數法

所謂場力函數 (Field function) 即為定義某個質點 (Key point) 在空間中影響力範圍的數學函數。Blinn [40] 最早採用場力函數的觀念來做物質分子模型 (Molecular) 的塑形。隨後，Wyvill 等人 [33,41] 將此觀念擴充，同時引進了等位能表面 (Iso-surface

) 的概念，專以軟狀物體為對象，做軟狀物體的塑形及動畫。Wyvill 以 Superquadrics, Superellipsoids 一類函數所產生的連續封閉曲面做為基本元素 (primitives)，然後藉著質點間的位置關係所造成的影響，將之結合而形成不同的形狀 (等位面)，以產生軟狀物體的外形。理論上來說，這種塑形方法能夠產生任何外形的物體。但因物體的外形主要是由質點間的位置決定，通常又都需要由很多的質點組合而成。所以，要隨心所欲的產生要求的形狀並不容易，甚至必需靠不斷的嚐試才行。

3. 曲面合成法 (Composite surface patches)

利用許多任意形狀的曲面 (Free-form surfaces) 來合成物體外形，是 Solid modeling 中常用的一種塑形方式，特別是在機械元件的塑形上。大體上而言，曲面合成的塑形有三種方式〔42〕：

(1) 布林運算法 (Boolean operations on free-form surfaces)

這種方法是先以曲面產生一些簡單而常用的形狀作為基本元素，再將這些基本元素利用布林運算，組合產生複雜的形狀。但因此種合成方式必須做曲面交接的處理 (Surfaces intersections test)，所以，處理上較為複雜、費時。

(2) 曲面的混合 (Blending surfaces)

亦即將二個不同的曲面接合在一起，並使得交接處能夠

滿足所指定的平滑程度要求。這種方法通常用於機械元件的邊緣或角的平滑化處理。

(3) 以特徵多面體產生曲面 (Generating surfaces from a characteristic polyhedron)

使用者依所要求的曲面形狀，於適當的位置給定控制點，設定該曲面之特徵多面體，系統即根據這些控制點將該曲面產生出來。這種塑形方法較為簡單，而且能夠很容易的修改所定義曲面的形狀，也不需要處理曲面的交接情況。其缺點則為不易很精確的塑造所需的曲面形狀，所能產生的形狀也會受到限制。

一個曲面的形成至少都需要12個以上的向量決定其形狀，但是要經由設定這12個向量以獲得所要求的形狀，往往也不是件容易的事，更何況在曲面交接處尚須滿足平滑程度的要求，遑論隨心所欲的產生變形後的形狀。所以，並不適合用以做為氣球的塑形。

4. 掃掠法 (Sweeping representation from animatin)

在幾何塑形中，掃掠法是一種簡單、方便而且應用範圍相當廣的一種塑形方式。其做法為將一個已定義好的曲線或曲面（稱為 Generator，通常為物體的橫斷面），沿著一條指定的路徑（稱為 Director，即為物體的軸或 Spine）掃掠，而產生物體外形。曲線的掃掠將產生一個曲面；而曲面的掃掠則產生三維的物體。掃掠法依其 Generator 及 Director 的不同可分為以下幾種〔43〕：

(1) 平移掃掠 (Translational sweeping)

一個位於 Base plane P_0 上的二維區域，沿著平移向量 T ，平移掃掠至 Cap plane P_1 ，即可產生一個三維的物體〔44〕。其中，Generator 可為任意形狀，而路徑則為一直線。例如，一個圓經平移掃掠則形成一個圓柱體；一個四方形經平移掃掠則形成一個四面體。(圖 3.2)

(2) 旋轉掃掠 (Rotational sweeping)

Generator 可為任意形狀，而 Director 則為一個圓。亦即物體是由二維的區域，沿著一個圓形的旋轉軸掃掠而得〔44〕。例如，圓形的 Generator 經旋轉掃掠將形成一個環。(圖 3.3)

(3) 圓形掃掠 (Circle Sweeping)

物體是由一個圓形的區域，沿著空間中的一條路徑掃掠而產生。其中，Generator 為一個圓，而 Director 則可為空間中的任意曲線〔45〕。(圖 3.4)

(4) General Sweeping

這種掃掠法為前述幾種方法的擴充，物體的橫斷面不再侷限於二維空間中，而和掃掠路徑一般可為三維空間中的任意曲線或形狀。同時，在掃掠過程中，Generator 尚可依需要改變形狀〔46〕。因為 種掃掠法的變化很多，所以能夠產生非常多樣化的物體外形。但在理論上仍有一些未知的部份，所以在實際的處理上可能非常複雜

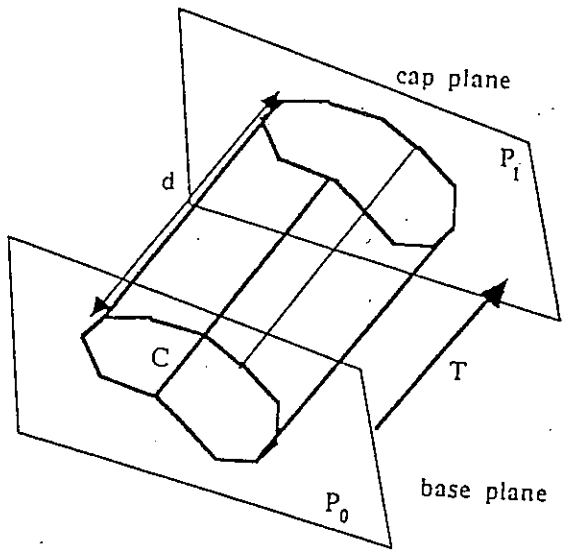


圖 3.2 Translational sweeping

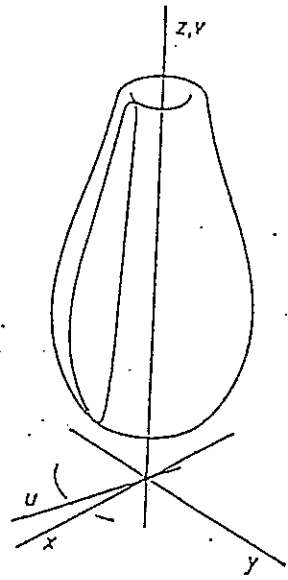


圖 3.3 旋轉掃掠

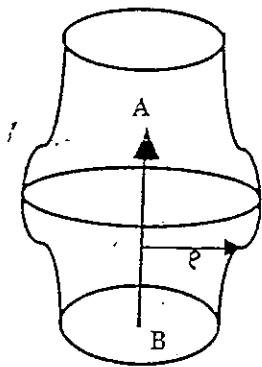


圖 3.4

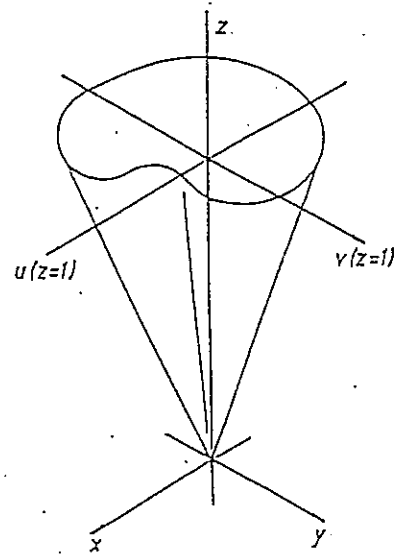


圖 3.5 卡尼克掃掠

。相關研究可參考〔43,46,47〕。(圖 3.5)

事實上，General sweeping 非常類似於二維空間動畫中廣為採用的主架構圖法。在這種動畫方法中，設計者先定義物體的運動路徑，同時，也設定好物體起始、最後的形狀以及運動過程中在某些時間上的形狀和位置，（這些定義好的形狀即為主圖像），然後再依此以內插的方式自動造出中間的形狀變化，而產生動畫。所不同的只是前者是用以做物體的塑形，而後者則是用之於動畫的處理。當然這種方法也可用以做物體的塑形，因為主圖像的位置、大小、方向、和形狀等都已固定，將使掃掠的處理較為單純。而且對於使用者而言，藉著定義橫斷面的方式來設計物體的外形也非常方便。此外，在塑形的處理上，使用者所決定的主圖像（通常是由曲線所構成）可以數學式表示，主圖像間的中間圖像也可以內插的方式產生。所以，所需處理的資料量並不大，而且處理起來複雜度也不高，對氣球的塑形而言，是一個相當理想的方法。

3.1.2 氣球的塑形

根據以上對軟狀物體塑形方法的討論，得自於動畫中的主圖像式掃掠法，是最適合用以做為氣球塑形的的方法。在實際的應用上，我們採取以下兩個步驟：

1. 橫斷面的形狀設計 (Contour design)

每一個橫斷面係為由使用者定義之控制點 (Control points) 所決定的一條軟楔 (Spline) 曲線。使用者可以交談式的方式，輸入適當數量的橫斷面形狀，做為主圖像。

2. 中間圖像的產生及物體表面的四邊形化 (Generating in-betweens and polygonization)

系統根據使用者所定義的主圖像，以內插的方式自動產生中間圖像。同時，為配合真實性處理時的需要，也將物體的外形以多邊形的組合 (Polygon mesh) 表之。

3.1.2.1 橫斷面的設計

為了便於使用者設計及修改橫斷面的形狀，我們採用 Spline Bezier 曲線來描繪氣球的橫斷面。亦即，使用者定義了橫斷面的控制點之後，系統就依據這些點，在相鄰的兩控制點間產生一條通過這兩點的 Bezier 曲線，然後，連接所有的這些 Bezier 曲線，而形成一條通過所有控制點的封閉軟楔曲線 (A closed spline curve with Bezier curve segments)。 (相關的數學原理將於第四章中詳細介紹)。同時，為了使得所產生橫斷面外形的平滑程度，能夠符合氣球的實際狀況，曲線在控制點上的連接必須滿足 C2 Continuous。

如此，對使用者而言，只要知道橫斷面的形狀，就能夠利用系統

所提供的交談式介面，很方便的在適當的位置定義控制點，以決定或修改橫段面的形狀。

3.1.2.2 中間圖像的產生及物體表面的四邊形化

要在二個相鄰的主圖像之間產生中間圖像，則該二個主圖像上的控制點數目（或是線段數目）必須相同，如此方能帶入適當的數學模式，以內插的方式求出中間圖像。至於應該如何使得兩個圖像上具有相同的控制點數目，過去已有許多的研究者提出幾種不同的作法。

Burtnyk 和 Wein [48] 最早提出了一個解決的方法。

對於分別具有 NP1 及 NP2 個控制點數目的兩個圖像，KF1 和 KF2，可經由以下的演算法，使得 KF1 和 KF2 上的控制點數目相同。

If $NP1 < NP2$ Then

 If $NP1 > NP2$ Then

$RT := (NP1-1) \text{ Div } (NP2-1)$

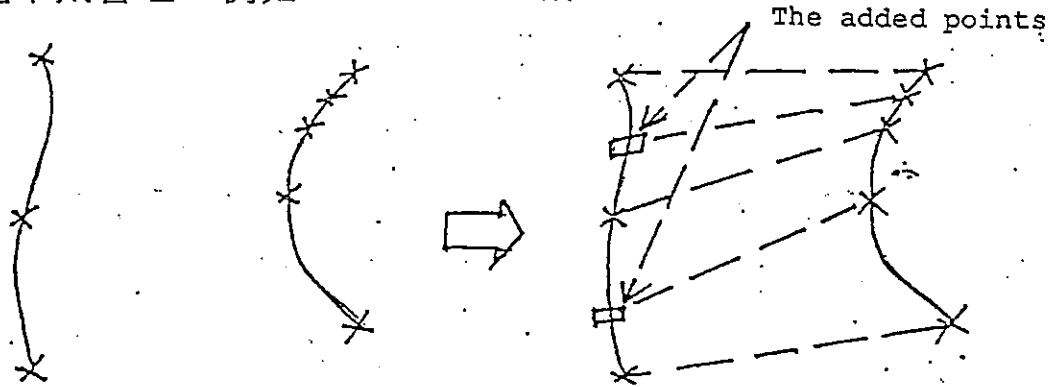
$RS := (NP1-1) \text{ Mod } (NP2-1)$

 RT points are added to the first RS segments, and
 (RT-1) to the others.

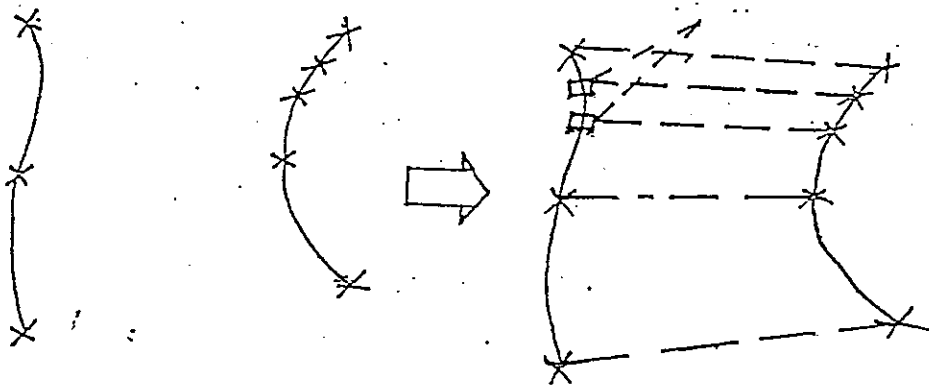
 If $NP2 > NP1$ Then

 the algorithm is the same with NP1 and NP2 exchanged

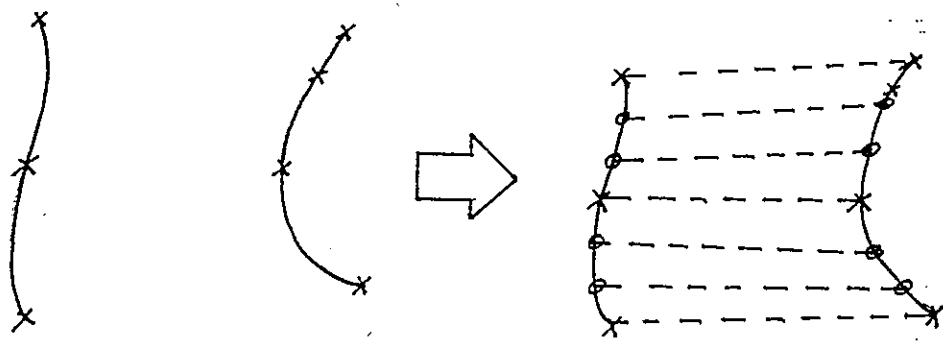
這個演算法的目的，就是將必須要增加的點，平均的加到各段之中，不僅簡單而且快速。但在某些情況下，這種作法所產生的結果，就可能不太合理。例如：



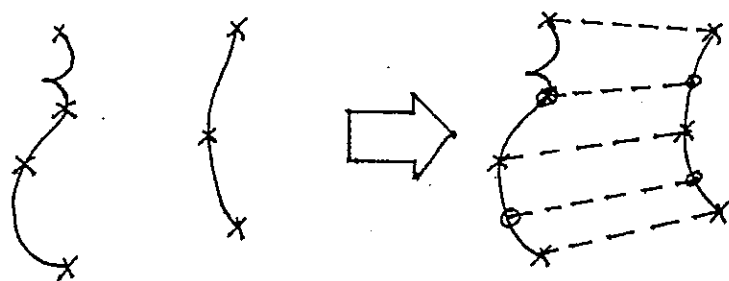
為了改善這種不均勻的現象，陳茂盛〔49〕根據各個控制點相對所在圖像上某一端點的長度比例，做這兩個圖像的對應位置比較，在對應的位置上加入控制點。



而在本研究中，為了方便將來氣球變形時的處理，所以採用較為簡單的方法——不論原來每個圖像上的控制點個數有多少，均以某一控制點為端點，等比例的將每個橫斷面分成相同的段數。



這個方法實際應用起來非常簡單，但可能發生一個問題，就是橫斷面上某一個小段上的變化，經過這種平均取法後，這個變化可能會被忽略了，而無法忠實的將使用者所定義的形狀表現出來。如：



為彌補這種缺點，我們將主圖像上的平均細分段數增加。雖然不能完全改善此缺點，但只要細分的段數夠多，這種錯誤的發生機會就會降低很多，而且在運算時間上也不致增加太多。此外，這種作法也會使得構成物體表面的多邊形更多更小，在做真實性處理時，將使物體的外形更平滑，更真實。

待每個橫斷面上的控制點數目都相等後，我們便可依序作一對一的對應，以內插的方式求出中插畫。

因為氣球的表面十分平滑，所以，線性內插 (Linear interpolation) 的方式並不能滿足我們的要求。我們改以 Spline interpolation 的方式來產生中插畫，亦即類似橫斷面設計時所採用的方法，將所有橫斷面上相對應的控制點視為一組控制點。根據這組控制點，產生一條由 Bezier 曲線段所連接而成的開放軟楔曲線 (An opened spline curve with Bezier segments)。此外，在每個控制點上都滿足 C2 continuous 的連續程度要求，如此方能使得所塑造出的形狀能夠符合氣球的實際情況，而主圖像間中插畫的位置與形狀，即可透過該段 Bezier 曲線的參數求得。

最後，連接所有主圖像和中插畫上相鄰的控制點，即可得到一個由許多的多邊形所構成的氣球外形。

3.2 變形

若物體受外力影響而發生變形，一但此外力除去，物體將會恢復至原來的形狀，則稱此物體具有彈性 (elasticity)。因此，氣球的受力變形 (例如，吹氣) 是屬於彈性變形的一種。所以，理論上來說，氣球吹氣變形的分析是屬於材料力學 (material mechanics) 和彈性力學 (elastic mechanics) 的範疇。在第二章中我們也介紹了幾種在電腦動畫中處理物體變形的的方法。但無論是利用牛頓力學原理

或是用有限元素 (finite element) 法來分析，都涉及專門的物理學原理與繁複的數學運算。所以，在本研究中並不從學理上來分析氣球吹氣變形的原理，而藉著觀察氣球吹氣變形的實際現象，利用基本的物理觀念來模仿真實氣球的變形情況。

3.2.1 氣球吹氣變形現象分析

一般而言，氣球在沒有氣體吹入或是將原有的氣體全部放空的情況下，外表看起來是乾扁的，類似一個曲曲折折的曲面，我們稱此時的氣球為處於起始狀態。而為了說明方便起見，我們將氣球的吹氣膨脹過程分為二個階段：

第一階段：由起始狀態 → 膨脹狀態

自起始狀態開始緩緩吹入氣體，直至氣球各部份的表皮均被撐開，成為一個外表平滑的形狀，則氣球進入了膨脹狀態。在此階段的變化過程中，氣球的顏色、透明度並未發生明顯變化。

第二階段：由膨脹狀態 → 變形狀態

自膨脹狀態開始再有氣體吹入，則氣球開始變形（進入變形狀態）。此時因吹入之氣體使氣球內部的壓力增加，而將氣球表皮向外擴張。同時，在表皮上的每一個點都會受

到上下及左右二對相互垂直的薄膜張力的作用，使表皮在這些方向上發生拉伸以平衡氣球內部的壓力。

又因氣球各部份的質料並不均勻（厚度不同），所以在發生變形時並非各點均勻變形，而是自最薄的地方開始變形。變形將使得氣球體積增大，表皮變薄，顏色變淡，透明度增加。

若再有氣體吹入，則氣球持續發生變形，直至某處所必須承受的壓力超過了該處薄膜張力的極限而爆破。

3.2.2 模擬方法

本節中我們將對關於彈性變形問題在動畫上的處理方法，以及本研究中所採用的方法作詳細的介紹。就理論上而言，此類問題應以材料力學或彈性力學原理予以分析，但因本研究的重點並不在模擬（Simulation），而且這些物理分析過於繁複，所以我們僅以模仿的方式產生近似的效果。至於相關的基本理論則將於第四章中說明。

Terzopoulos 和 Fleischer [35] 提出了一個根據物理學及幾何學理論模擬產生曲線、曲面及實體元素彈性變形的實際模式（Physically-based model）。這個模式不但能表現出物體的純彈性行為，而且也能將諸如黏彈性（Viscoelasticity）、塑性（Plasticity）

及斷裂情況 (Fracture) 等的非彈性 (Inelastic) 現象。表現出來。

此模式是採用能量 (Energy) 的觀點來考慮物體的變形 —— 物體受力變形後會產生一回復力與之平衡，變形程度愈大則回復力愈大；而在外力尚未去除的情形下，此回復力即是以位能 (Potential energy) 的形式儲存在於物體中。例如，彈簧受外力作用壓縮後，外力所作之功即轉變為彈簧的彈性位能。而變形的程度則是以物體的 Metric tensor 來衡量。

物體上任何一點在時間 t 時的位置可表示為

$$x(u, t) = [X(u, t), Y(u, t), Z(u, t)]$$

其中，若該物體為一實體，則 $u = (u_1, u_2, u_3)$

若為一曲面，則 $u = (u_1, u_2)$

若為一曲線，則 $u = (u_1)$

在不失其一般性的情況下，可令 $0 \leq u_1, u_2, u_3 \leq 1$ 。

同理，物體在未變形時的位置則可表示為

$$x^0(u) = [x_1^0(u), x_2^0(u), x_3^0(u)]'$$

根據 Lagrange 運動方程式及力的平衡原理，可得到下列的式子

$$\mu \frac{\partial^2 x}{\partial t^2} + \gamma \frac{\partial x}{\partial t} + \delta_x \mathcal{E} = f. \quad (3-1)$$

也就是在變形過程中，物體所受的淨外力 $f(u, t)$ 必須和物體受力後所產生的以下三種力平衡：

1. 物體本身受力後產生加速度所形成的力， $\mu \frac{\partial^2 x}{\partial t^2}$ ，其中 $\mu(u)$ 為質量密度 (Mass density) 函數。

2. 物體在運動過程中所散失的力， $\gamma \frac{\partial x}{\partial t}$ ， $\gamma(u)$ 為阻尼密度 (Damping density)，通常是以矩陣來表式，但在此則為一純量 (Scalar)。

3. 物體抵抗變形所產生的彈力， $\delta_x \varepsilon$ ，在此是以位能函數子 (Energy functional)，相對於位置函數 x 的 Variational derivative 來表式。

當物體處於原始未變形狀態時，其位能為 0，變形程度愈大則位能愈大。(圖 3.10)。而且，物體經剛體運動 (如移動、旋轉) 後並不會發生變形，所以不會產生位能變化。

至於物體變形程度的量度，則是採用微分幾何學 (Differential geometry) 中的基本定理來計算 [50]。對於空間中的實體而言，其外形係由物體表面的許許多多點彼此間的距離所決定的，而這些距離 (及角度) 便可以 Metric tensor G 來量度 [51]。

$$G_{ij}(x(u, t)) = \frac{\partial x}{\partial u_i} \cdot \frac{\partial x}{\partial u_j}$$

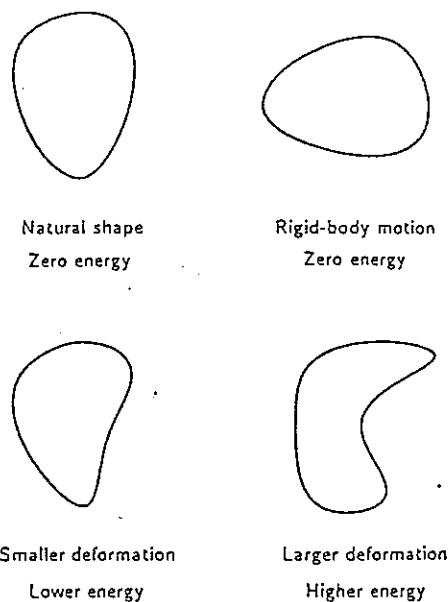


圖 3.10 Energy of deformation

因為根據實體的基本定理：空間中的兩個實體，若在時間 t 時，其 Metric tensor 為 u 的相同函數，其中 $u = (u_1, u_2, u_3)$ ，則這二個實體便具有相同的形狀〔51〕。但對曲面而言，因曲面可能在外形上會有顯著的改變，但表面點間的距離變化卻不大。所以，若是僅僅根據 Metric tensor，並不足以計量曲面的形狀變化。故曲面的基本定理告訴我們：若兩個曲面之 Metric tensor 以及 Curvature tensor B （均為 2×2 的矩陣）均為相同的 u 的函數，其中， $u = (u_1, u_2)$ ，則這兩個曲面具有相同的形狀〔51〕。而 Curvature tensor 的定義則為〔51〕。

$$B_{ij}(x(u, t)) = n \cdot \frac{\partial^2 x}{\partial u_i \partial u_j}$$

其中， $n(u, t) = \frac{\partial x / \partial u_1 \times \partial x / \partial u_2}{|\partial x / \partial u_1 \times \partial x / \partial u_2|}$ 為曲面在時間 t 時的單位法向量。

對於空間中的曲線而言， $u = (u_1)$ ，Metric tensor 和 Curvature tensor 即分別為純量的弧長 (arc length) 函數 $S(x(u, t))$ ，和曲率函數 $\kappa(x(u, t))$ 。但僅以這二個量度仍不足以決定曲線的形狀。因為，雖然這兩個函數相同，但曲線仍可能發生扭曲 (Twist)，以致形狀不同。所以，曲線的基本定理說明：若二曲線之弧長、曲率以及 Torsion 函數 $\tau(x(u, t))$ 均為相同的 u 函數，則這兩個曲線具有相同的形狀 [51]。

由以上的基本定理，彈力位能可由物體變形後與變形前基本的 Tensor 之差的 Euclidean norm (L2) 來量度。所以，對於變形的曲線而言，其彈力位能可表為

$$\mathcal{E}(x) = \int_{\Omega} w^1 (s - s^0)^2 + w^2 (\kappa - \kappa^0)^2 + w^3 (\tau - \tau^0)^2 du$$

其中， $w^1(u)$ ， $w^2(u)$ ，及 $w^3(u)$ 分別為決定曲線的拉伸 (Stretching)、彎曲 (Bending)、以及扭轉 (Torsion) 程度的權重函數 (Weighting function)。

對於空間中變形的曲面而言，則

$$\mathcal{E}(x) = \int_{\Omega} |G - G^0|_{w_1}^2 + |B - B^0|_{w_2}^2 du_1 du_2$$

其中， $|\cdot|_{w_1}$ 及 $|\cdot|_{w_2}$ 分別為和權重函數 $w_{ij}^1(u_1, u_2)$ 及 $w_{ij}^2(u_1, u_2)$ 相關的 Weighted matrix norm。

同理，對於變形的實體而言，

$$\mathcal{E}(x) = \int_{\Omega} |G - G^0|_{w_1}^2 du_1 du_2 du_3$$

其中， $|\cdot|_{w_1}$ 為和權重函數 $w_{ij}^1(u_1, u_2, u_3)$ 相關的 Weighted matrix norm。而 w_{ij}^1 則決定此物體之拉伸及切變 (Shearing) 的程度。

Terzopoulos 即據此選擇了不同的權重函數，以曲面的方式模擬出橡皮、紙張等的彈性變形情況。此外，Terzopoulos 也根據物理學原理，討論了幾種外力的情況。諸如重力、氣體動力、碰撞後的反作用力等等，以模擬出更為真實的效果。

而要利用此變形模式來產生動畫，則需依所處理物體的情況來解以上所討論的平衡方程式 (3-1)。作者則是採用數值分析的方法，分二個步驟來處理：

首先，以有限元素法或是有限差分近似法 (finite-difference approximations) 將 (3-1) 式簡化，同時予以離散化 (discretization)，結果可以得到一組一般微分方程組 (Ordinary differential equations)，(亦即將偏微分方程組簡化為一般微分方程組

)。第二個步驟則是將這些微分方程式分別對時間 t 積分，如此即可根據動力學原理計算出物體表面上每一點在任一時間時的位置，模擬出真實的變形動畫。

由以上的介紹，我們可以知道，這整個過程非常的繁複，從分析到實作都和專門的物理學原理以及數學方法息息相關，而且相當複雜，當然所獲得的動化效果是近乎真實、相當令人滿意的。但以目前的種種條件來看，這種模擬式的動畫對我們而言並不適合。所以，本研究僅以簡化的方式，在不違背真實的情況下，模仿氣球吹氣的變形過程。

此外，根據 Platt 和 Barr [52] 以有限元素法對柔軟物體受力後所產生彈性變形的分析結果——物體表面的這些元素，可以視為是許許多多個別的質點，而相鄰的質點間則是以非線性彈簧互相連接。利用這個模式，假設氣球吹氣的過程中除大氣壓力和所吹入的氣體壓力外，並無其他任何外力的作用，則氣球吹氣時可視為在各個質點的法線向量上同時都受到一個彈簧向外拉伸，而使得原本質點間的彈簧伸長，因而導致氣球發生膨脹變形。而變形的大小則和彈簧的彈性係數 (Elastic modulus) 有關。

這種模式，若以實際的物理原理來分析，設若氣球是由 NKF 個主圖像所構成，每個主圖像上有 NP 個控制點 (質點)，則至少必須知道

$$(NKF-2) \times 3 \times NP + 2 \times 2 \times NP = (3 \times NKF - 2) \times NP$$

個彈簧的彈性係數。同時，這些非線性彈簧的組合，在分析上也需非常繁複的計算。另一方面，對使用者而言，這麼多的參數，雖然得以完全控制氣球各部份的變形情況，但過多的自由度不但將成為使用者的一大負擔，更何況對於不瞭解這些物理性質的使用者，將不易產生所需的動畫效果。基於這些考慮，而且在不違背真實情況的原則下，我們採用以下的方法：

使用者僅需輸入每個質點上的一個彈性係數，做為該質點在法線方向上伸長量的控制參數，亦即僅需 $NKF \times NP$ 個參數。事實上，因為這些主圖像係為由使用者所設定的 CP 個控制點而決定的軟楔曲線，所以，使用者只需決定每個控制點上的彈性係數即可，而所有 NP 個質點上的彈性係數則可由這 CP 個控制點以內插的方式算出。因此，實際上只有 $NKF \times CP$ 個彈性係數必需決定。雖然控制的參數減少，但我們仍可控制到每個質點上的變形程度；而且，當 CP 的值愈接近 NP 時，在每個質點上變形程度的控制就愈精細愈準確。當然，相對的所需的參數也就愈多，計算的時間也愈長。

至於氣球所承受的氣體壓力，因量度不易，且一般人也不清楚要多少的壓力才會有多大的變形，所以，為方便起見，我們採用單位化的量度，也就是將其壓力值定為從 0 到 1 的範圍。當壓力為 0 時，代表氣球處於最開始的膨脹狀態；壓力為 1 時，氣球則達到最大的

變形狀態。使用者僅需決定欲產生若干個圖像 (frames) 的變形狀態，以及氣球變形的演變法則 (evolution law)，系統即自動計算對應之壓力值，而產生變形的過程。演變法則為決定動畫過程中動作變化的快慢、大小等效果的法則。在本系統中則根據變形程度變化的快慢提供以下幾種法則：

$$1. \text{Liner: } P\text{-fract} = \text{fract}$$

$$2. \text{Acceleration: } P\text{-fract} = 1 - \cos(\pi \times \text{fract}/2)$$

$$3. \text{Deceleration: } P\text{-fract} = \sin(\pi \times \text{fract}/2)$$

其中 fract 係根據圖像數目的多寡而定，P-fract 則為對應於該圖像之壓力值 ($0 \leq P\text{-fract} \leq 1$)。

一旦氣球內部所受的壓力決定之後，就可用來計算形狀的變化了。根據彈性力學中的虎克定律 (Hooke's Law) —— 彈性物體受外力作用後所發生的應變 (Strain)，和所受的應力 (Stress) 成正比 [53]。所以，對於氣球表面上的某一個質點 (X_0, Y_0, Z_0) ，當受到內部壓力 P 的作用時，發生變形而移動至位置 (x, y, z) ，則

$$x = P/M \times N_x + x_0$$

$$y = P/M \times N_y + y_0$$

$$z = P/M \times N_z + z_0$$

其中 M 為質點 Q 上之彈性係數， (N_x, N_y, N_z) 則為 Q 點上之單位法向量。

而為了使氣球的變形後不致於背離原來的形狀，同時保持光滑均勻的外形，所以法向量

$$N = (N_x, N_y, N_z) = \frac{(Q_A \times Q_B + Q_B \times Q_C + Q_C \times Q_D + Q_D \times Q_A)}{|Q_A \times Q_B + Q_B \times Q_C + Q_C \times Q_D + Q_D \times Q_A|}$$

但是彈性體都有彈性限度，不可能無限制的發生應變；再者，氣球受力變形愈大表面的薄膜張力也就愈大。所以，當變形到某種程度之後，應變將不再隨壓力的增加而正比增加。而且，氣球本身的橡皮材料也並非滿足虎克定律的正比關係，（參考第四章 4-3節彈性理論的說明）。所以我們改以 \tan 函數來模擬氣球變形時的應力應變關係。

$$x = (\tan^{-1}(P/M)) \times N_x + x_0$$

$$y = (\tan^{-1}(P/M)) \times N_y + y_0$$

$$z = (\tan^{-1}(P/M)) \times N_z + z_0$$

以解決上述的問題。

經實驗的結果，以此方法所產生的氣球吹氣變化過程，雖不若依物體的實際物理性質而定的動畫模式精確，但仍能達到頗為真實的效果，而且計算簡單，使用者對於物體的變化也可擁有較大的控制能力。

除了氣球的吹氣變形現象之外，在本研究中也引用了反向動力學的觀念，對於氣球受擠壓變形的情況做了實驗。

Gourret 等人〔63〕曾以有限元素法，模擬人類手指擠壓彈性物體時，物體以及手指的反應效果。當然，這也是一個根據物體實際物理性質的動力學動畫模型，所呈現出的效果自然是十分的真實。

在本研究中，我們也嘗試以所建立的物體模式，模仿氣球受擠壓變形的效果。根據對於實際現象的觀察，我們可以發現，當氣球（已進入膨脹狀態）一類的飽滿而富彈性的物體，受到手指等物體（外力）的擠壓時，氣球表面受外力的部分將發生凹陷。而因氣球本身具有彈性的緣故，所以凹陷的部分呈現近似半圓球的形狀。氣球表面直接承受外力的部分發生變形的程度最大，陷入最深；隨著距此受力中心點的距離愈大，所產生的變形也就愈小。

由以上的分析，在本系統中就以球體來模擬擠壓的外力。若對氣球擠壓的範圍較大，則以半徑較大的球體表之；反之，擠壓的範圍小則以小半徑的球體表示。而擠壓力道的強弱，則可以此一球體距離氣球表面的遠近表現出來。當氣球表面和該代表外力的球體有接觸時，表示氣球已受到外力的擠壓，則氣球表面和球體接觸的部分，必須凹陷成為所接觸到球體表面的形狀。而凹陷的程度愈深，即代表所承受的外力愈大；反之亦然。

所以，當擠壓的範圍（球體的半徑）以及擠壓力道（球體距氣

球的距離) 決定了之後, 系統就能夠計算出氣球表面的形狀變化。而根據氣球表面這些受到擠壓的點的位置變化, 即可反算出在這些點上分別所應承受的外力值。然後, 再計算這些外力對整個氣球所造成的影響, 受到擠壓的氣球變形效果就表現出來了。此外, 根據力的平衡原理, 氣球必會受到另一相等的外力作用, 以與擠壓的力平衡, 否則氣球將產生加速度, 而不會有明顯的形狀變化。為了方便起見, 系統則在氣球的另一個半球上的對應位置, 做相同的變形處理, 以使得所產生的影像, 不致於違反自然原則。

而若是隨著時間改變球體和氣球間的距離, 即能夠表現出氣球受外力擠壓的動畫過程。

3.3 影像的真實性處理

在本研究中動畫影像的真實性處理工作是利用 ANIMATOR 軟體系統完成的。ANIMATOR 是一套由美國科羅拉多州立大學發展出來, 專門用以處理物體真實性表現的軟體系統, 其輸出入的部分則均以檔案的方式完成。使用者利用 ANIMATOR 所提供的指令, 建構好所需的物體檔 (Object description files), 動畫檔 (Scene description file)、以及由多邊形組合而成的物體外形檔當作輸入, 系統即依據這些檔案中所定義的參數對物體做繪影、表面紋理、透明度等效果的

處理，然後產生 Raster 檔，供該畫面的顯示之用〔54〕。

ANIMATOR的處理主要有畫面分析和物體真實性處理二個步驟〔55〕（圖 3.11）

1. 畫面分析 (Scene Analysis)

此步驟的目的是將畫面中所有的物體，依其大小、距觀測點的遠近，以掃描線演算法 (Scan-line algorithm) 賦予每一個物體優先權 (Priority)，同時也根據觀測點的位置及視界的大小，將物體做適當的轉換 (Viewing transformation) 以及 Clipping。（圖 3.12）

在物體的排序上，則是根據二項測試：

(1) An overlap test, 及

(2) A priority and separability test

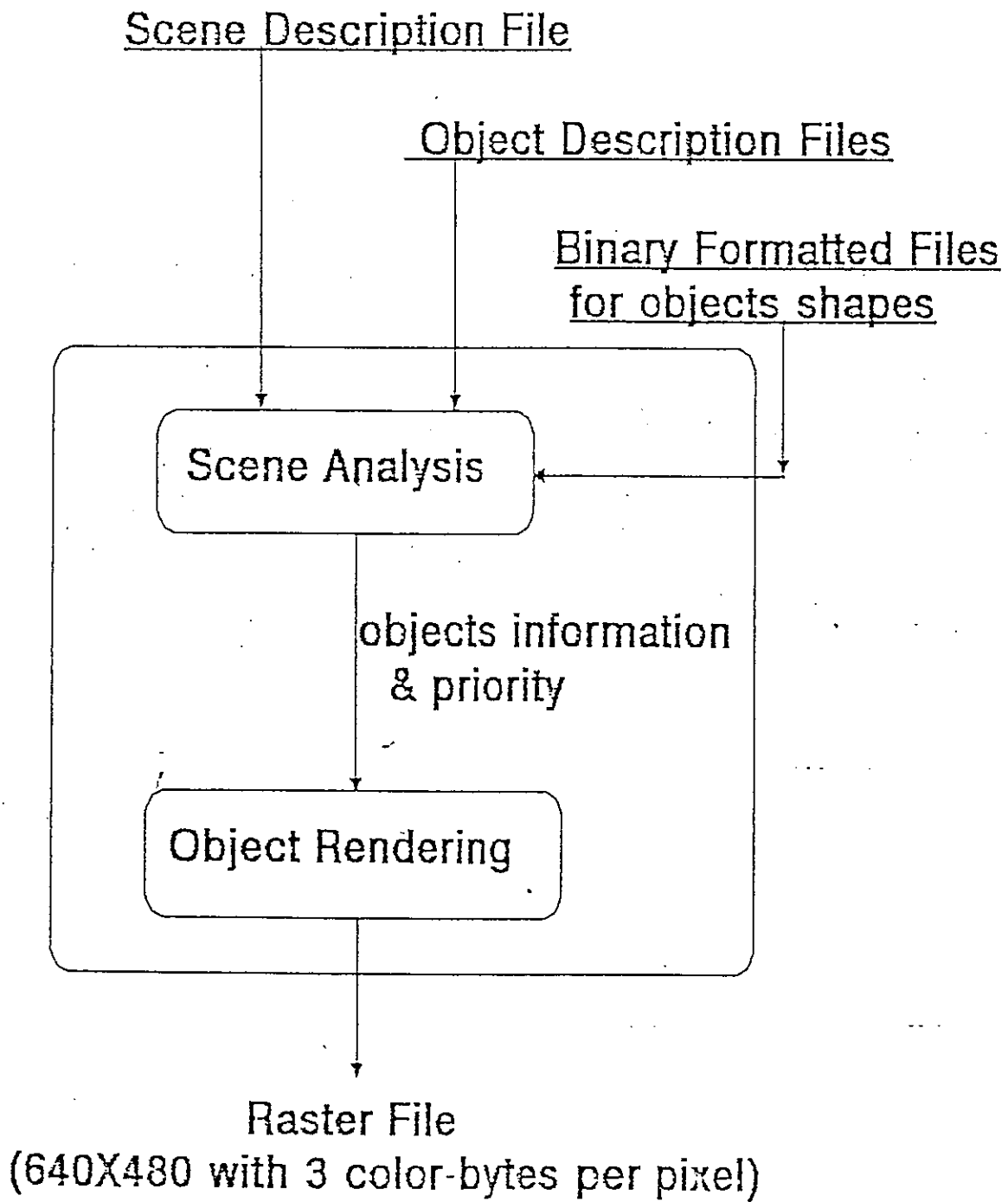
來決定物體間的先後順序關係，同時據此定出物體的優先權，以供下一個步驟之用。

2. 物體真實性處理 (Object rendering)

這個步驟則是根據畫面分析後所得到的資訊，對物體做繪影、表面紋理、透明度、Antialiasing、和 highlight 等真實性表現的處理。（圖 3.13）。

接下來我們就針對物體真實性處理方面的作法做比較詳細的介紹。

ANIMATOR



Scene Analysis

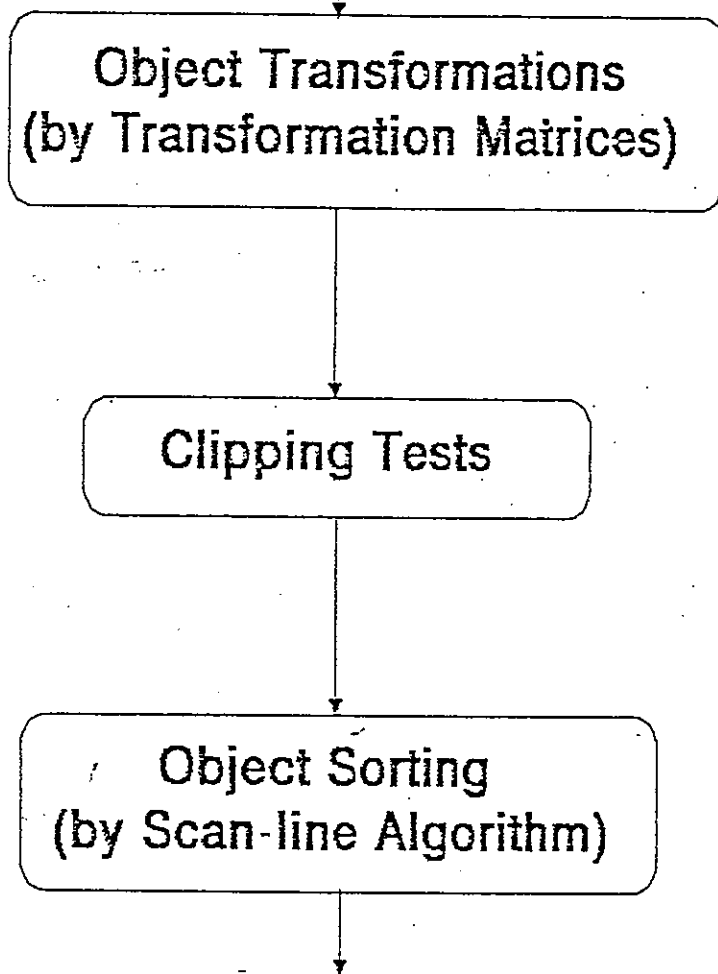


圖 3.12

Object Rendering

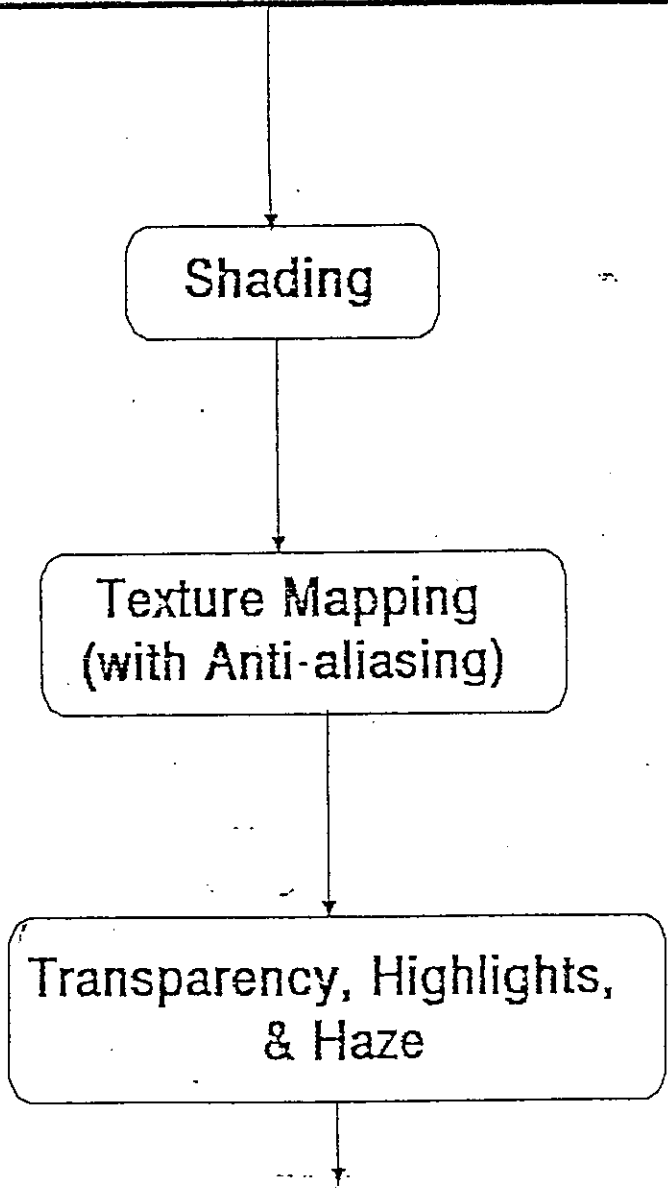


图 3.13

3.3.1 光影模式 (Illumination Model)

ANIMATOR 採用 Gouraud shading [56] 的方式做為畫面光影的處理。其原理為以掃描線的方式決定每條掃描線上各個圖元 (pixel) 的 Intensity。但是因為只計算了物體表面多邊形端點上的法向量。而且掃描線也不一定會通過這些端點。所以，就先計算出端點上的 Intensity，然後以 Bilinear interpolation 的方式求得掃描線上各個圖元的 Intensity [57]。

Intensity 的計算則是使用簡單的光影模式 (Simple illumination model)。

$$\begin{aligned} I &= \text{ambient reflection} + \text{diffuse reflection} \\ &= I_a k_a + I_i k_d \cos \theta \quad 0 \leq \theta \leq \pi/2 \end{aligned}$$

其中，

I_a 為入射的 ambient light intensity

k_a 為 ambient 反射常數 ($0 \leq k_a \leq 1$)

I_i 為點光源之 intensity

k_d 為 diffuse 反射常數 ($0 \leq k_d \leq 1$)

θ 為入射光和物體表面法向量間的夾角

ambient 的部分為物體受背景光源 (Background light) 的影響而產生的。亦即周遭物體受光源照射，多次反射後的光對該物體所產生的影響。diffuse 部分則為物體受光源照射後，沒被吸收部分的光向四面八方反射出去對物體所形成的影響。根據 Lambert's Cosine Law：這種反射光的強度和入射光以及物體表面的法向量間所形成夾角的餘弦函數成正比。

此外，當觀測者位於某些角度時，將能夠感受到較強的反射光線，而在物體表面受光照射的部分，則顯現近似光的顏色的一塊區域，這種現象稱為 Specular reflection，(圖 3.14)。因 Specular reflectance 和入射光的角度、物體表面特性、光的波長等有關，其中又牽涉複雜的物理性質。所以，一般在計算 Specular 反射，多採用 BT Phong 所提出的一種實驗性模式〔58〕。

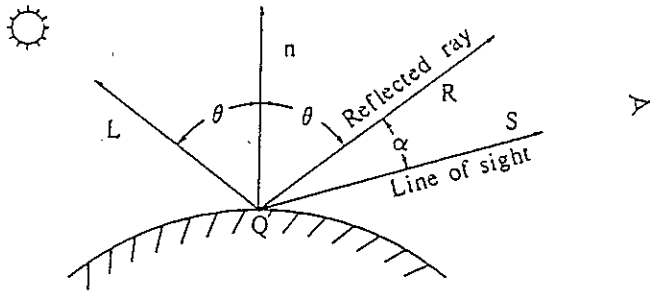
$$I_s = I_l \omega(i, \lambda) \cos^n \alpha$$

其中，

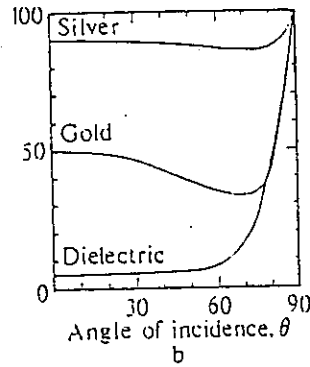
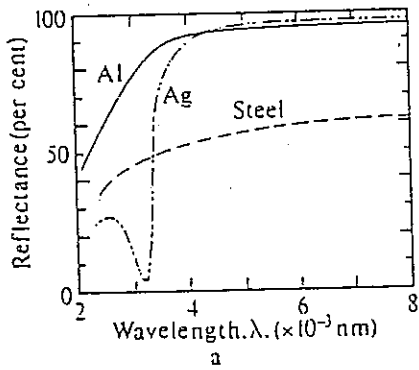
$\omega(i, \lambda)$ 為已知光線入射角 i ，光波長 λ 的情況下所得之反射曲線 (Reflection curve)，(圖 3.15)

n 為模擬控制反射光之空間分布 (Spatiala distribution) 的參數，(圖 3.16)

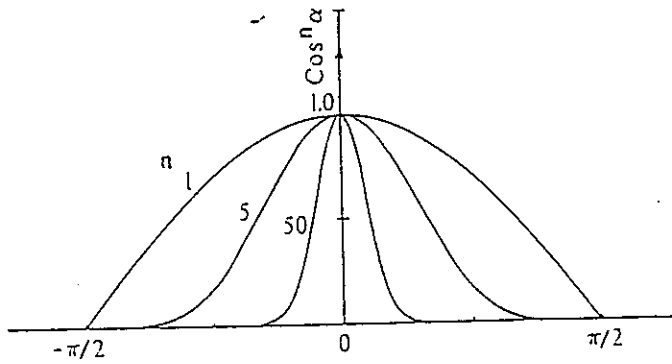
α 為反射光和觀測者間的夾角。



3.14 Specular reflection.



3.15 Reflection curves.



3.16 Approximate spatial distribution function for specularly reflected light.

因 $\omega(i, \lambda)$ 函數太複雜，所以通常將之簡化為

$$I = I_a k_a + \frac{I_l}{d + K} (k_d \cos \theta + k_r \cos^n \alpha)$$

所以，整個光影模式就可寫成（ m 個光源的情況）

$$I = I_a k_a + \sum_{j=1}^m \frac{I_{l_j}}{d + K} (k_d \cos \theta_j + k_r \cos^n \alpha_j)$$

Gouraud shading 最大的缺點，就是會有十分明顯的 Mach banding 現象。亦即，在物體表面，Intensity 變化比較大的相鄰多邊形的邊緣上，會有 Intensity 不連續，而產生不夠平滑、一塊一塊的感覺。主要的原因則在於 Gouraud 是以 Intensity 做內插的緣故。當然，這種方法也比較簡單。後來 Phong [58] 則改採用法向量做內插，先算出各個圖元上的法向量，再由法向量計算 Intensity 的方法加以改良，大大的降低了 Mach banding 現象（無法完全消除），而獲致更為真實的光影效果。

至於物體的透明度表現上，ANIMATOR 則是採用 Kay和Greenberg [59] 所提出的非線性透明度演算法（Non-linear transparency algorithm）：

$$t = T^d$$

其中，

t 為透明度，

T 為單位距離的透明度，

d 為光線穿過介質所行經的單位距離長度。

當觀測透明物體時，可以發現，在我們正視的物體中心部分最透明，而距離觀測中心愈遠，和觀測點所形成的夾角愈大時，就愈不透明。所以，透明度和觀測者與物體表面法向量間夾角的餘弦函數值成正比例關係。

$$t = \frac{n}{n + \cos(\theta + \alpha)}$$

當 n 愈大時，物體透明度的收斂程度愈大，也就愈不透明，感覺也比較厚；反之，n 愈小，物體透明度愈高，看起來也愈薄。

3.3.2 Texture Mapping

Texture 就是物體表面紋理的表現。通常有二種做法，一是將設計好的圖樣“貼”於物體表面上 (Texture mapping)；另一則是改變物體表面的粗糙程度 (Bump mapping)。物體表面粗糙程度的改變基本上是利用擾亂函數來改變物體表面的法向量，使物體看起來有凹凸不平的感覺，而非實際去改變物體的外形。而要在平滑的表面加上特殊圖樣，本質上則是屬於座標系統間的轉換問題。亦即，將定義於

Texture space (s,t) 上的紋理圖樣 (Texture pattern) ，加至定義於 Object space (u,v) 的物體表面上，最後再投影至 Screen space (x,y) 上，而得到最後的影像。(圖 3.17) ，ANIMATOR 所提供的就是這種 Texture mapping 的方法。

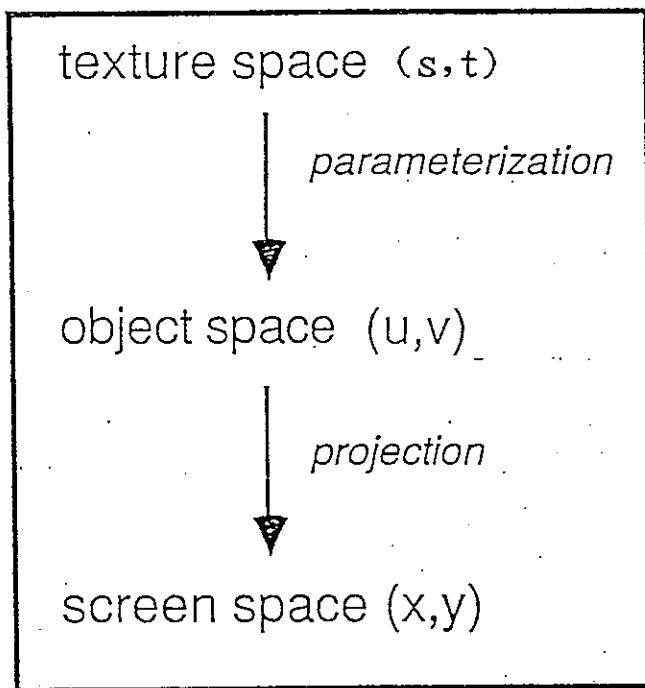


圖 3.17 The overall mapping is composed of a surface parameterization and a viewing projection.

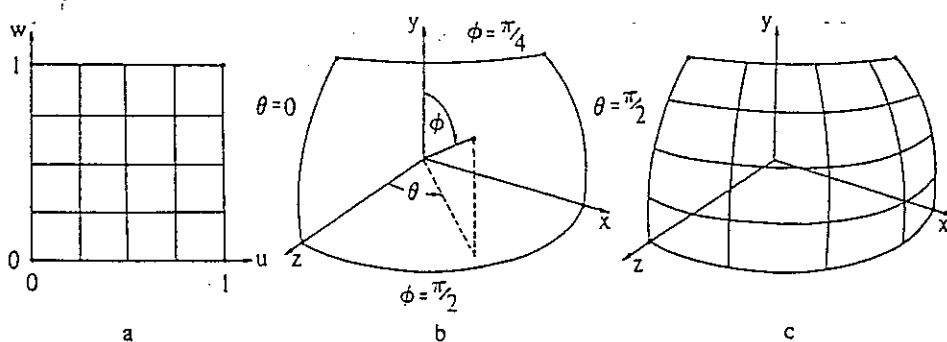


圖 3.18 Mapping.

Texture mapping 一般有三種常用的作法〔60〕：(圖 3.19)

1. Screen Scanning : 是最普遍被採用的方法，也稱為 Inverse mapping。
2. Texture Scanning : 是最簡單的方法，也稱為 Forward mapping
3. Two-pass methods : 最適合於 Multiprocessors 的環境下。

ANIMATOR 則是採用 Forward mapping 的方法來做表面紋理的處理。

已知 Texture space 和 Object space 間的轉換規則 (例如，線性的對應關係)：

$$u = f(s, t), \quad v = g(s, t)$$

再由適當的已知條件 (例如，Corners match)，則可得到每個 (s, t) 平面上的點對應到 (u, v) 平面上的位置，如此即可決定 (u, v) 上各個點的顏色 (亦即表面紋理)。然後，將 (u, v) 上的點投影至 Screen space (x, y) 的每一個圖元上，就可得到表面貼上圖樣的物體影像。

物體表面紋理的處理可大幅提高影像的真實性，但往往也會造成相當程度的 Aliasing 問題，使得物體表面所貼上的圖樣看起來有失真的感覺，(例如，鋸齒狀的現象)。因此，Antialiasing 的處理，在影像的真實性表現上也是不可或缺的。

圖 3.19 .

SCREEN SCANNING :

```
for y
  for x
    compute u(x,y) and v(x,y)
    copy TEX[u,v] to SCR[x,y]
```

TEXTURE SCANNING :

```
for v
  for u
    compute x(u,v) and y(u,v)
    copy TEX[u,v] to SCR[x,y]
```

TWO - PASS :

```
for v
  for u
    compute x(u,v)
    copy TEX[u,v] to TEMP[x,v]
for x
  for v
    compute y(x,v)
    copy TEMP[x,v] to SCR[x,y]
```

第四章 數學方法

4.1 Bezier 曲線

Bezier 曲線是電腦輔助幾何設計 (CAGD Computer Aided Geometric Design) 中用以設計任意形狀曲線 (Free - Form Curve) 的一種極為常用的方法。

一條由 $n+1$ 個控制點 (Control Points), P_i , ($i=0, 1, 2, \dots, n$) 所定義而成的 Bezier 曲線 $R(t)$, 可以參數式表示為

$$R(t) = \sum_{i=0}^n P_i B_i^n(t) \quad \text{for } t \in [0,1],$$

其中, $B_i^n(t)$ 為 Bernstein 多項式, 在此做為 blending function (圖 4.1)。

$$B_i^n(t) = \binom{n}{i} (1-t)^{n-i} t^i, \quad i=0, 1, \dots, n,$$

Bernstein 多項式具有以下的性質 [64,65] :

1. Partition of unity

$$\sum_{i=0}^n B_i^n(t) \equiv 1,$$

2. Positivity

$$B_i^n(t) \geq 0 \quad \text{for } t \in [0,1],$$

3. Recursion

$$B_i^n(t) = (1-t)B_i^{n-1}(t) + tB_{i-1}^{n-1}(t),$$

4. Symmetry

$$B_i^n(t) = B_{n-i}^n(1-t)$$

5. $B_0^0(t) \equiv 1$

6. $B_i^n(t) = 0$ for $t \notin [0,1]$,

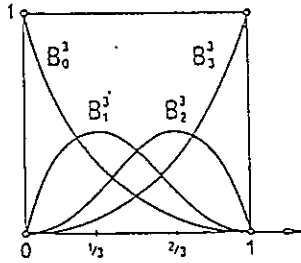
而 Bezier 曲線則具有下列特性 :

1. Affine invariance

因為 $\sum_{i=0}^n B_i^n(t) \equiv 1$, 所以曲線 $R(t)$ 和控制點 P_i 間的關係即使經過 affine transformation 也不會改變。

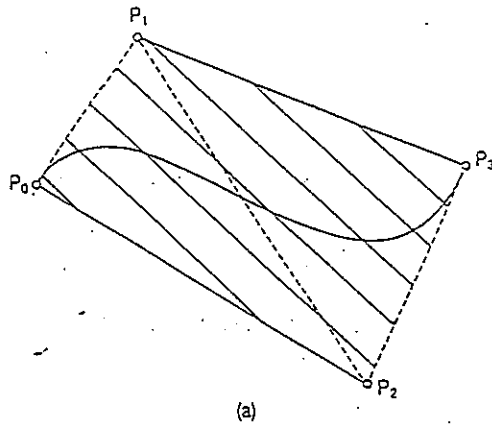
2. Convex hull property

曲線 $R(t)$ 中每個控制點的權重 (weight) 係由 $B_i^n(t)$ 而定, 而 $\sum_{i=0}^n B_i^n(t) \equiv 1$, 且 $B_i^n(t) \geq 0$ 。所以, Bezier

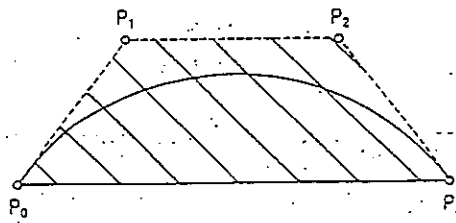


4.1

Cubic Bernstein polynomials.



(a)



(b)



4.2 Bézier curves and their convex hulls.

曲線上的任何一點，必須位於由全部控制點所連接而成的多邊形（Control polygon）之內。如圖 4.2。

3. End point interpolation

$$\text{因 } B_i^n(0) = 0 \text{ iff } i = 0$$

$$B_i^n(1) = 0 \text{ iff } i = n$$

所以，曲線 $R(t)$ 必通過第一個和最後一個控制點。

4. Pseudo - local control

Bernstein 多項式 $B_i^n(t)$ ，於 $t = i/n$ 時有唯一的最大值。所以，當改變某個控制點 P_i 的位置時，則整個曲線的形狀在 $t = i/n$ 的附近區域所受的影響最大。雖然整條曲線均會受到影響，但使用者仍可藉此預知改變後的大致形狀。

5. Variation - diminishing property

空間中和 Bezier 曲線有最多交點的平面，即為 control polygon 所在的平面。亦即 Bezier 曲線不會有任意振盪（Oscillation）的現象。

Bezier 曲線最大的缺點，在於其次數係由控制點的數目決定。因此，當我們希望以較多的控制點（10個以上）來塑造形狀較複雜的曲線時，將無法以 Bezier 曲線來處理。此外，Bezier 曲線除了首尾二個控制點外，並不通過其他的控制點。所以，為了讓使用者能夠更容易、更準確的定義出所要求的形狀，我們採用由 Bezier 曲線段

所形成的軟楔曲線 (Spline curves in Bezier form) 供使用者做橫斷面的設計。

4.2 產生通過給定點的曲線

欲產生一通過給定點的封閉曲線，如圖 4.3 所示，一般的做法為在相鄰的二個給定點間做一段曲線，並使其在給定點上依所要求的連續程度 (Level of continuity) 相連而得。

在本研究中，為滿足氣球橫切面的平滑程度要求，我們採用三次 (Cubic) 的 Bezier 曲線相連，同時，在接點上滿足二階連續 (C2 Continuous) 。原理說明如下：

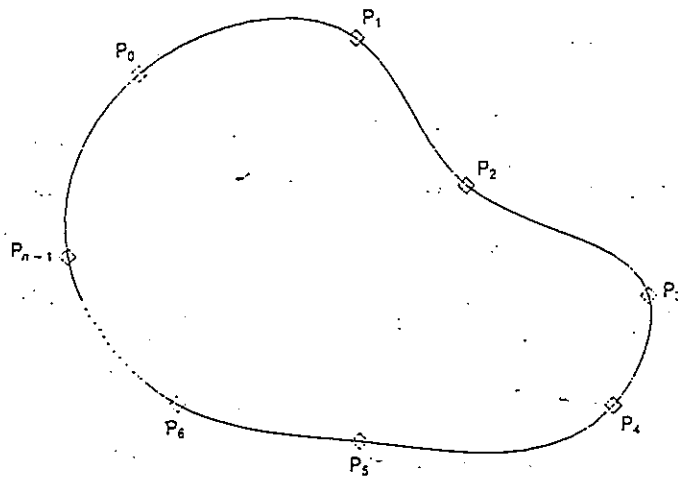


圖 4.3 Closed curve.

4.2.1 曲線的相連：

圖 4.4 顯示一條三階的 Bezier 曲線 $R_i(t)$ ，及其控制點 $P_{i,j}$ ($j=0, \dots, 3$)。定義控制向量 (Control vectors) A_i, B_i, C_i 及 D_i 如下：

$$\begin{aligned} A_i &= P_{i,3} - P_{i,0} \\ B_i &= P_{i,1} - P_{i,0} \\ C_i &= P_{i,3} - P_{i,2} \\ D_i &= P_{i,2} - P_{i,1} \end{aligned}$$

要使所有的曲線能夠平滑的相連起來，則這些控制向量必需滿足一些限制：

1. 切線向量與主法線向量的連續 (Continuity of tangents and principal normals)

如圖 4.5 所示，二條交於 P_i 的 Bezier 曲線 $R_{i-1}(t_{i-1})$ 及 $R_i(t_i)$ ，其中， Q_i 為直線 $\overline{P_{i-1,1}P_{i-1,2}}$ 和 $\overline{P_{i,1}P_{i,2}}$ 的交點。假設控制點間的線段長度有以下的比例關係：

$$\begin{aligned} \overline{P_{i-1,2}P_i} : \overline{P_iP_{i,1}} &= k_i : 1 \\ \overline{P_{i-1,1}P_{i-1,2}} : \overline{P_{i-1,2}Q_i} &= k_i' : 1 \\ \overline{Q_iP_{i,1}} : \overline{P_{i,1}P_{i,2}} &= k_i'' : 1 \end{aligned}$$

考慮 R_{i-1} 及 R_i 在 P_i 上滿足切線及主法向量的連續。對 $R(t)$ 而言，其主法線向量定義為位於切線向量和 $R(t)$ 所

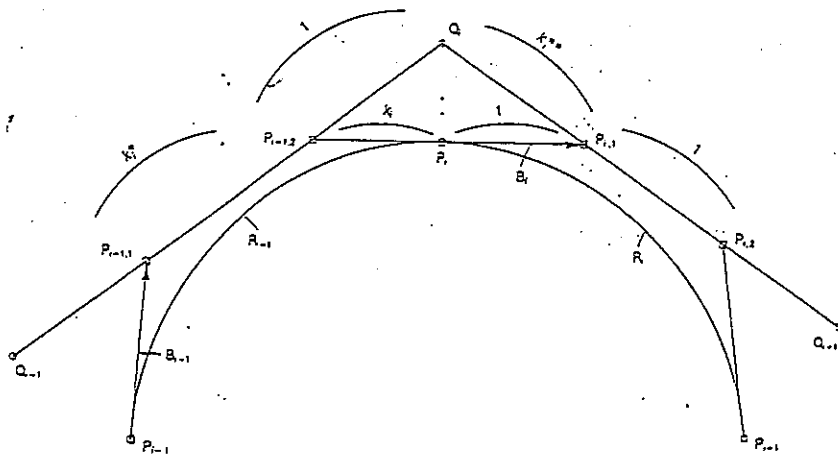
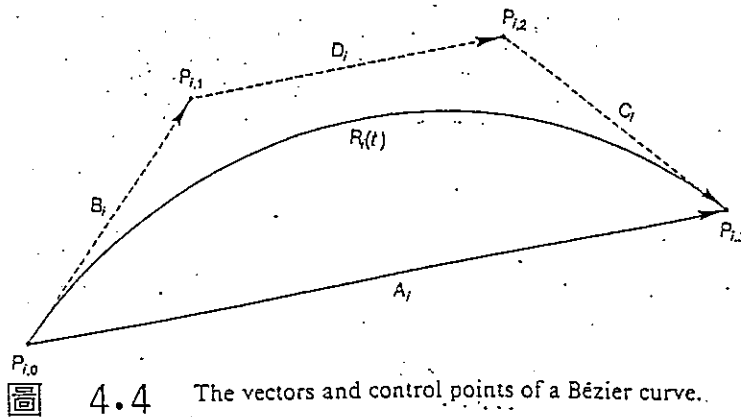
形成平面上，且垂直於切線向量的單位向量。

切線向量

$$t = \frac{\dot{R}(t)}{|\dot{R}(t)|}$$

主法線向量
$$n = \frac{\ddot{R}(t) - (t \cdot \ddot{R}(t))t}{|\ddot{R}(t) - (t \cdot \ddot{R}(t))t|}$$

為滿足此連續條件，則控制點 $P_{i-1,2}$ ， $P_{i,1}$ 和交點 P_i



必需位於同一直線上，而且 Q_i 必存在。因此控制向量間具有以下的關係：

$$\begin{aligned} C_{i-1} &= k_i B_i \\ \frac{D_{i-1}}{k_i} - C_{i-1} &= B_i - k_i^* D_i \end{aligned} \quad (4-1)$$

將 C_{i-1} ， D_{i-1} 及 D_i 分別以 B_{i-1} ， B_i 及 B_{i+1} 取代，則可得

$$\begin{aligned} B_{i-1} + (k_i^* + k_i^* k_i^* + k_i + k_i^* k_i) B_i \\ + k_i^* k_i^* k_{i+1} B_{i+1} &= A_{i-1} + k_i^* k_i^* A_i \end{aligned} \quad (4-2)$$

控制向量滿足 (4-2) 式，則 R_{i-1} 和 R_i 的連接滿足切線向量與主法線向量的連續。

2. 現將曲線 $R_{i-1}(t_{i-1})$ 和 $R_i(t_i)$ 以相同的參數 u 來表示，設若

$$u = t_{i-1} = 1 + \frac{t_i}{k_i}$$

則，

$$\begin{aligned} R_{i-1}(t_{i-1}) &= R_{i-1}(u) \\ R_i(t_i) &= R_i(k_i u - k_i) \end{aligned}$$

兩曲線相連滿足第一階及第二階導函數的連續，亦即

$$\begin{aligned} \dot{R}_{i-1}(1) &= k_i \dot{R}_i(0) \\ \ddot{R}_{i-1}(1) &= k_i^2 \ddot{R}_i(0) \end{aligned}$$

則必須

$$k_i = k_i' = k_i''$$

將此條件代至 (4-2) 式中，可得

$$\begin{aligned} B_{i-1} + 2k_i(k_i + 1)B_i + k_i^2 k_{i+1} B_{i+1} \\ = A_{i-1} + k_i^2 A_i \end{aligned} \quad (4-3)$$

利用 (4-3) 式，則我們可得到通過給定點的曲線。

4.2.2 產生封閉曲線

圖 (4.3) 所示為一通過給定點 P_i ($i=0, 1, \dots, n-1$) 的一條封閉曲線。欲產生此一曲線，則 (4-3) 式必須成立，其中 B_i 為未知數 (事實上為向量)，且

$$k_i = \frac{|A_j|}{|A_i|}$$

where

$$\text{if } i = 0 \quad \text{then } j = n-1 \quad \text{else } j = i-1$$

因為有 n 個點，所以可得到 n 個方程式，故可解得 B_i 代入 (4-1) 式中可求得 C_i ，即可算出每一段 Bezier 曲線上的另二個控制點。如此即可產生由這些 Bezier 曲線所形成的封閉軟楔曲線 (圖 4.3)。

4.2.3 產生開放曲線

欲產生如圖 4.6 所示之通過給定點 P_i ($i=0,1..n-1$) 的一條開放曲線，同樣的 (4-3) 式必須成立。但現在只有 $n-1$ 個連接點，亦即只有 $n-1$ 個方程式，但卻有 $n+1$ 個變數， B_i 。所以，仍需二個條件。

最常用的條件是使曲線的兩端點 P_0 ， P_n 上的第二階導數為 0，則

$$\begin{aligned} 2B_0 + k_1 B_1 &= A_0 \\ B_{n-1} + 2C_{n-1} &= A_{n-1} \end{aligned}$$

但如此將使得靠近曲線端點的部位成為直線，如圖 4.7，並不能滿足我們的平滑需求，所以我們採用其他條件——連接點 P_i 及 P_{n-1} 上的第三階導函數連續，則

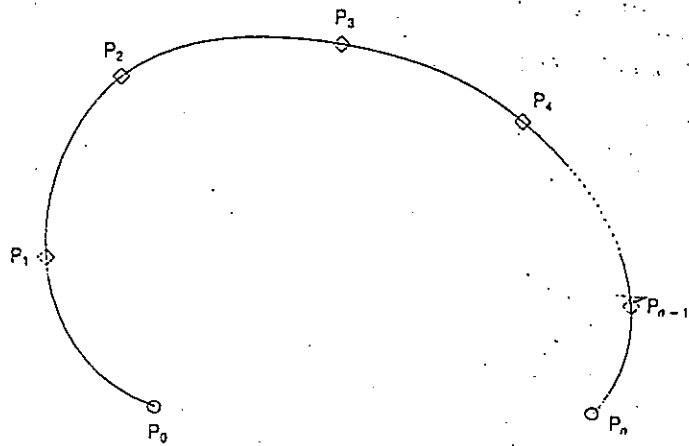
$$\begin{aligned} \ddot{R}_0(1) &= k_1^3 \ddot{R}_1(0) \\ \ddot{R}_{n-2}(1) &= k_{n-1}^3 \ddot{R}_{n-1}(0) \end{aligned} \tag{4-4}$$

對曲線 $R_i(t_i)$ 而言，其第三階導數可表示為

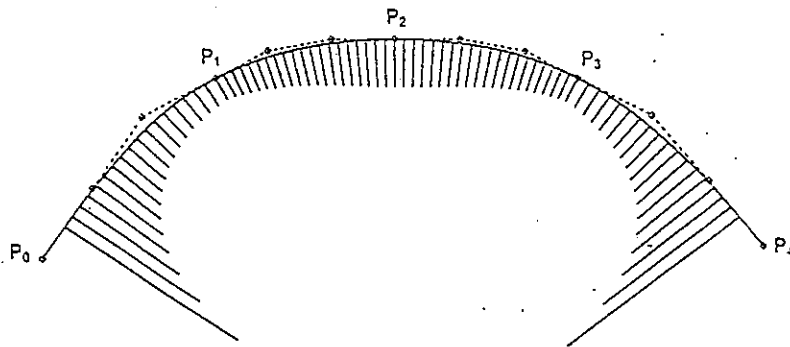
$$\ddot{R}_i(t_i) = 6(3k_{i+1}B_{i+1} - 2A_i + 3B_i)$$

所以，(4-4) 式可改寫為

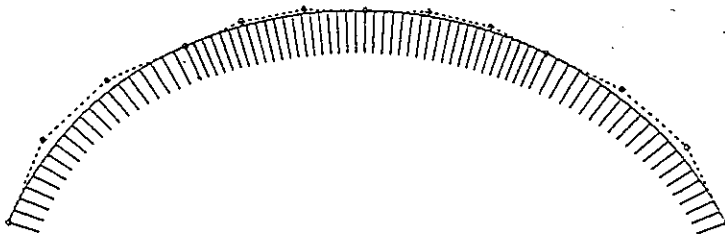
$$\begin{aligned} 3B_0 + 3(k_1 - k_1^3)B_1 - 3k_1^3 k_2 B_2 &= 2A_0 - 2k_1^3 A_1 \\ 3B_{n-2} + 3(k_{n-1} - k_{n-1}^3)B_{n-1} - 3k_{n-1}^3 k_n B_n \\ &= 2A_{n-2} - 2k_{n-1}^3 A_{n-1} \end{aligned}$$



4.6 Open curve.



4.7 A composite curve composed of Bézier curves.



4.8 A composite curve composed of Bézier curves.

有了這二個額外的條件，我們將可產生如圖（4.8）之平滑自然的軟楔曲線。

為了讓使用者在設計橫斷面時，能夠更容易繪出所需要的形狀，所以我們採用 Spline curves in Bezier form，讓此曲線通過使用所給定的控制點。但因此也犧牲了如 B-spline 般的 local control 能力。不過，我們仍可藉著控制曲線連接點的 tangent 及 curvature 連續程度的方式，使移動某個控制點時所影響的範圍縮小，以增加使用者對曲線的 local control 能力。

4.3 彈性理論基礎

任何受力外或外力矩作用後不會產生形狀變化的物體稱之為剛體。這是一種理想固體，實際上並沒有真正的剛體存在。亦即，任何物體受外力或力矩的作用後，均將產生某種程度的變形，而變形的多寡則取決於物體單位面積之受力，即應力（Stress），以及物體本身的性質，其間的關係常以各種彈性係數來表示。研究物體所受的力和力所產生的彈性變形間關係的理論，稱為彈性理論。

物體由於外在因素（受力、溫度變化等）而變形時，在它內部任一截面的兩方即出現相互作用力，稱為內力。單位面積上的內力則稱為應力。

$$p(\text{應力}) = \lim_{\Delta A \rightarrow 0} \frac{\Delta F}{\Delta A} = \frac{dF}{dA}$$

同截面垂直的稱為正應力或法向應力 (Normal stress) ；同截面相切的稱為剪應力或切應力 (Shearing stress) 。

物體在受應力作用的情況下，其形狀大小的變更則稱為應變 (Strain) 。任何應力均有一對應的應變產生。

假設有一質料均勻且自然長度為 l_0 的直棒，當於棒的兩端施以大小相等方向相反的兩拉力 F 時，此棒長度變為 l 。當然，此一伸長現象並非發生於兩端，而是棒中每一部份均以同一比例伸長。由於此棒係受拉伸張力之作用，故稱此棒之應變為拉伸應變 (Tensile strain) 。其定義為伸長量 Δl 與原長度之比。

$$\epsilon_t (\text{張應變}) = \frac{\Delta l}{l_0} = \frac{l - l_0}{l_0}$$

若改為壓力則產生壓應變 (Compressive strain) 。

$$\epsilon_c (\text{壓應變}) = \frac{\Delta l}{l_0} = \frac{l_0 - l}{l_0}$$

同理，切應力則將造成切應變 (Shearing strain) 。

應力與應變之比，或形成每單位應變所需之應力，稱為某材料的彈性係數 (modulus of elasticity) 。彈性係數愈大，則產生一定應變所需之應力就愈大。

$$\text{彈性係數} = \frac{\text{應力}}{\text{應變}}$$

楊氏係數 (Young's modulus) 最常用以表示一般材料的特性，為衡量物體受力時變形量大小的一種指標。其定義為

$$Y = \frac{\text{拉伸應力}}{\text{拉伸應變}} = \frac{\text{壓應力}}{\text{壓應變}} = \frac{\frac{F}{A}}{\frac{\Delta l}{l}} = \frac{lF}{A\Delta l}$$

上式可改寫為

$$F = \frac{YA}{l} \cdot \Delta l = k\Delta l$$

此即著名的虎克定律 (Hooke's law)。其中常數 $K = \frac{YA}{l}$ 。

對於材料的力學性通常常以應力應變圖來表示，圖 4.10 為一韌性固體的應力應變圖。塑性固體應變增加時，應力隨著應變成正比增加。但當達到屈服點 (Yielding point) Y 時，也就是應力達到材料的屈服應力 σ_y 時，材料呈現屈服現象。此時，材料自行伸長而應力保持不變，如圖 4.10 b；或者應變仍然增加，而應力至此突然下降，然後保持常數，如圖 4.10 a。此後應變即隨著應力的增加而增加，一直到強度的極限 σ_b 。這時材料所能承受的應力便達到最大限度，材料從此時就開始逐漸受到破壞。

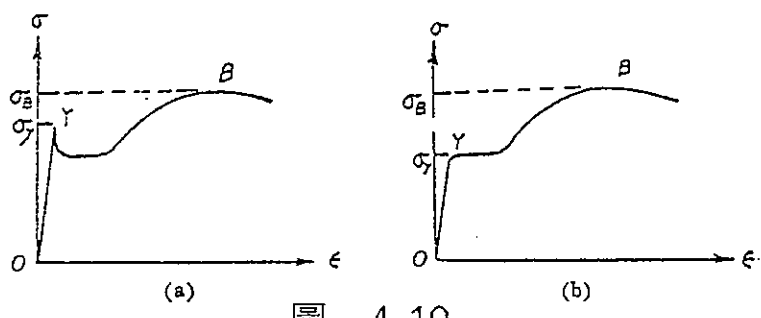


圖 4.10

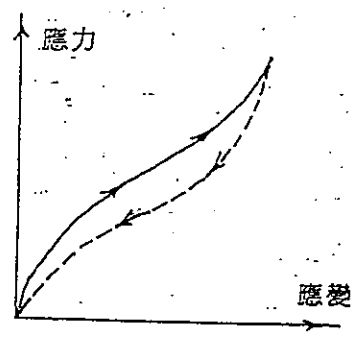


圖 4.11 非晶體之橡皮的應力與應變關係

圖 4.11 為加硫橡皮之應力應變關係圖。由圖中可知，加硫橡皮之應力應變並非正比關係，而且當負荷減少時，應力應變曲線之改變和原曲線不同，而是沿著一種迴路進行，經過迴路一周是要作功的，因此便產生彈性的滯後現象 (Elastic hysteresis) —— 增加或減少應力時兩曲線不能重合的現象。由於此橡皮當外力去除時，尚能回復原狀，因此為一彈性體。但因有能量的消耗，所以此彈力並非保守力〔61〕。

第五章 系統架構

本章中將針對本系統的整體架構，以即所使用的硬體設備、軟體系統、及其流程做詳細的介紹。同時，在第二節中也將對於系統所提供的可調參數及其影響做一說明。

5.1 系統架構與流程

本系統可分為動畫圖像的產生與圖像的顯示二個主要的部分。在動畫圖像的產生方面是由軟體程式完成的。為了配合所使用的硬體設備，以及輔助影像表現的軟體工具，系統中的所有程式均於 VAX 11/750 的 Ultrix 系統，或是 SUN Workstation 的 UNIX 作業系統下發展完成的。而圖像的顯示則是經由 Evans & Sutherland 的 PS300 系列 (PS340 或 PS390) 繪圖工作站顯示出來 (圖 5.1)。系統的主要流程，則如圖 5.2 所示。

對本系統而言，動畫的產生就是要產生氣球變形過程中的圖像。關於其原理以及所採用的方法，前幾章中已有完整的討論說明，在此不再贅述。

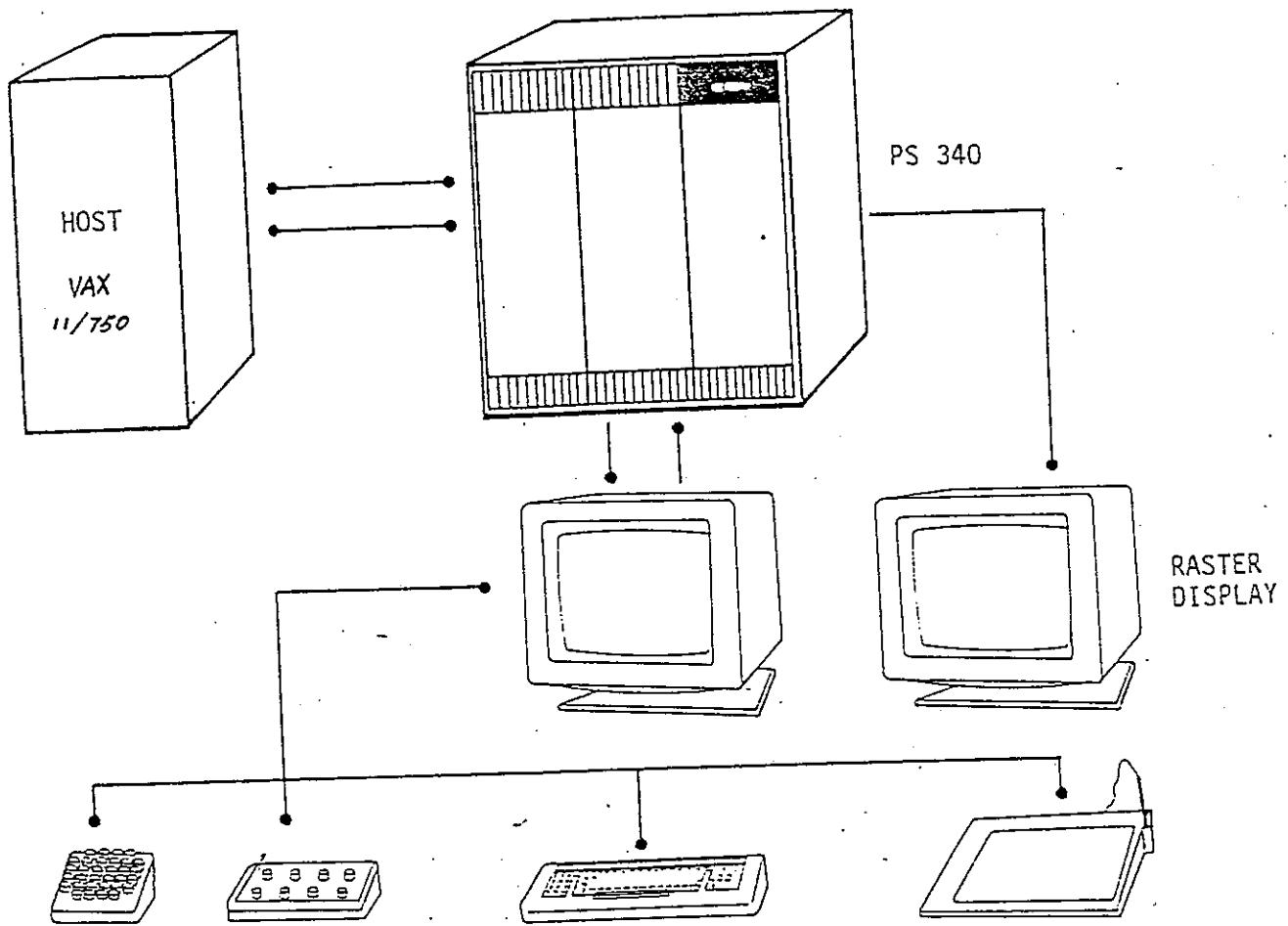


图 5.1 The PS 340

IAS0355

SYSTEM STRUCTURE

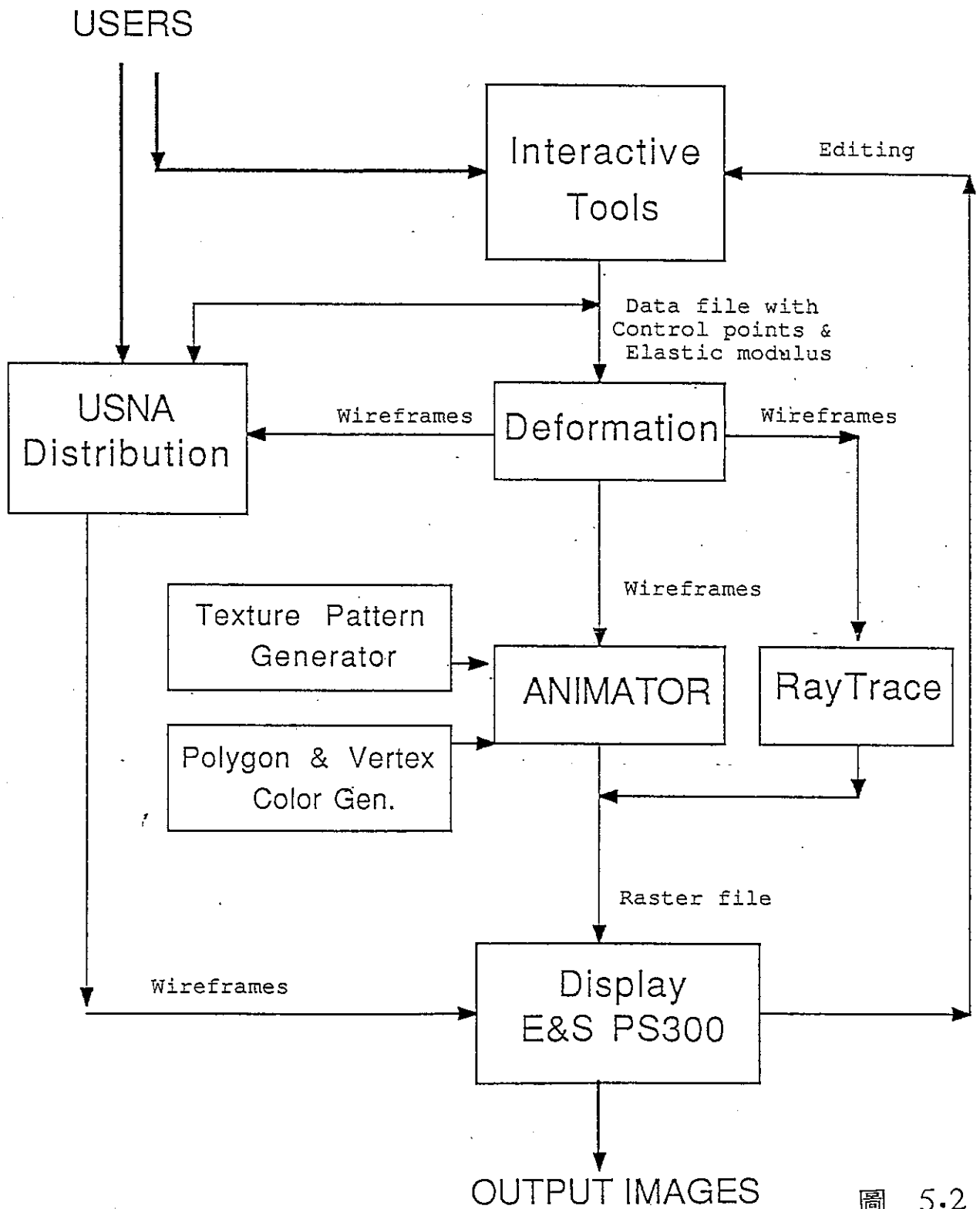


圖 5.2

使用者可以透過 Ultrix 系統下，由美國海軍學院 (USNA) 所發展的線架構塑形系統 (USNA Distribution)，或是系統於 PC 上所設計의交談式工具，輸入建構氣球外形所需的橫斷面 (控制點)，及這些控制點上的彈性係數。系統即根據這些資料，以及所指定的演變法則，依序增加吹入氣體的壓力，逐一的將吹氣變形後的線架構圖形產生出來。系統即利用 ANIMATOR 或是 RayTrace 等的軟體工具，對使用者所安排的畫面做繪影、表面紋理、光線追跡 (Raytracing)、著色、和透明度等真實性效果的處理。系統同時也提供了中文字的紋理製作系統，以及用以設定物體表面多邊形或其端點顏色的輔助工具，以使得所產生的效果能夠更生動、更多樣化。然後，系統即將每個畫面以 Raster 檔的形式產出來，透過 PS390 工作站即可看到每個畫面的結果。

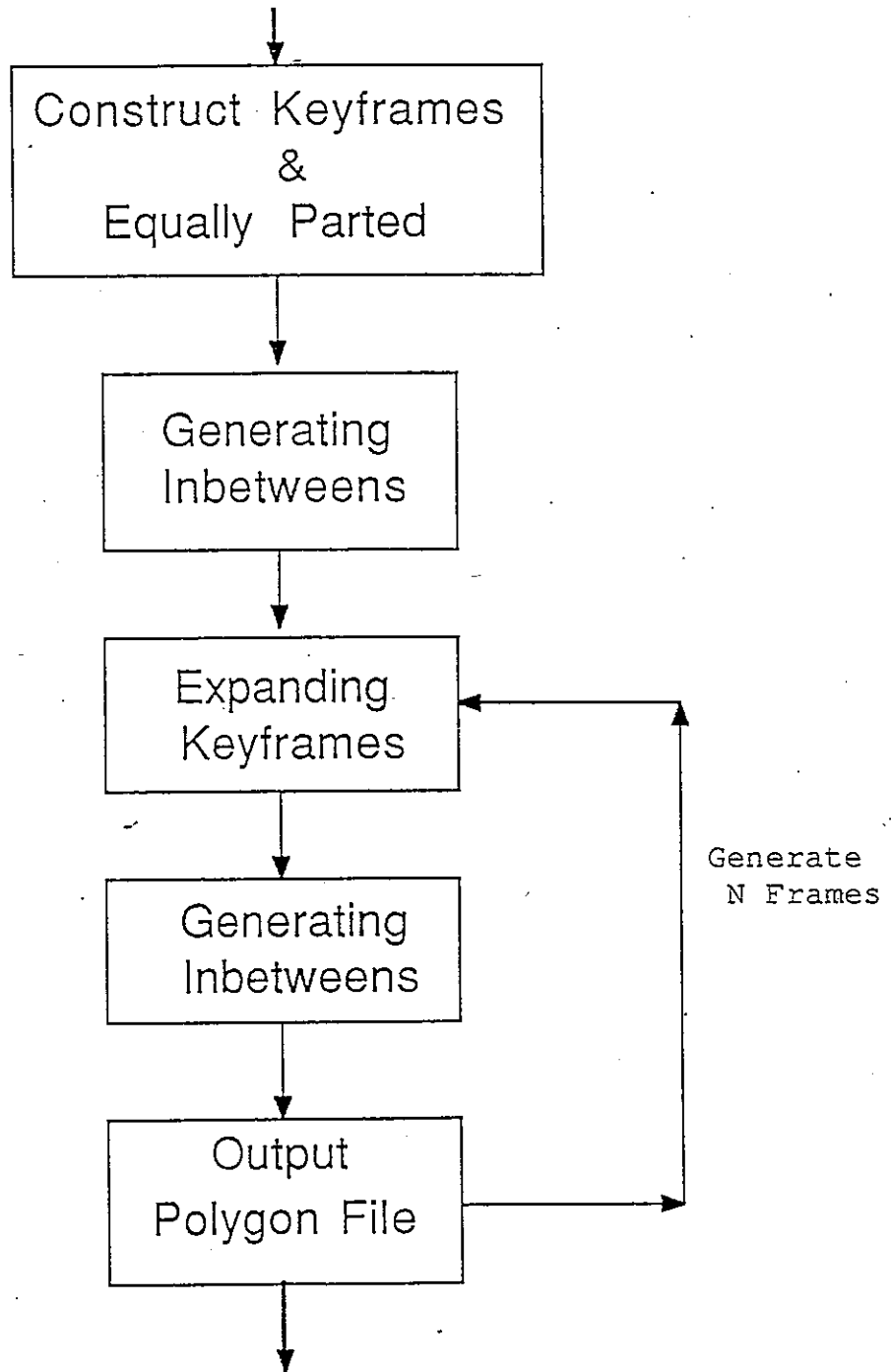
在本系統中，氣球變形的處理有氣球吹氣變形、以及擠壓變形二部分。其流程則分別如圖 5.3 及圖 5.4 所示。相關的原理請參照 3-2 節的說明。關於 ANIMATOR，也請參考 3-3 節中的介紹。

5.2 系統參數

根據系統產生動畫的步驟，使用者所能控制的參數共有以下幾類

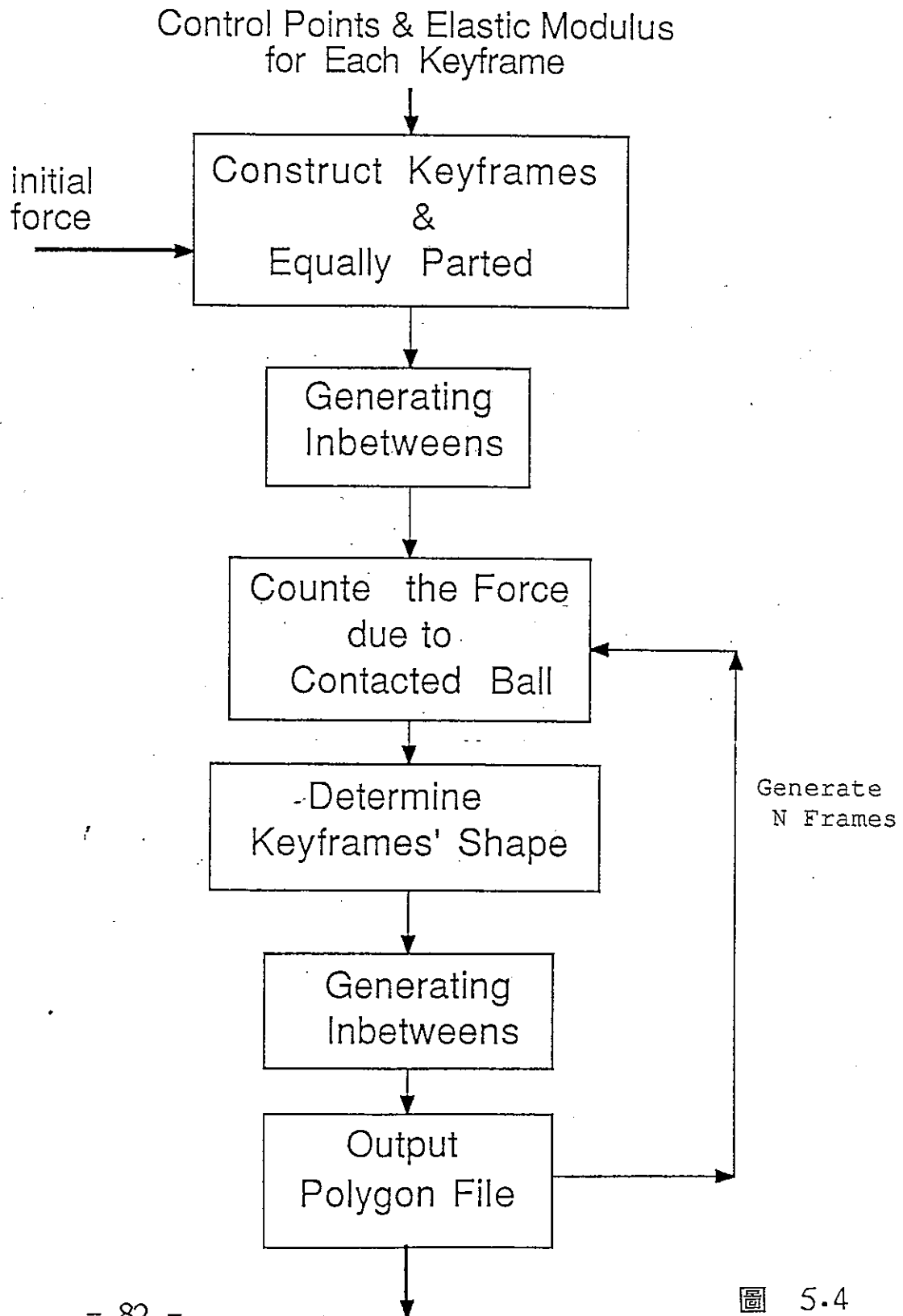
PUFFING

Control Points & Elastic Modulus
for Each Keyframe



■ 5.3

SQUASHING



1. 塑形及變形參數

包括定義每一個橫斷面的控制點，以及每個控制點上的彈性係數。彈性係數愈大，則產生相同變形所需的應力就愈大；反之亦然。使用者可以藉著這些控制點以及彈性係數的改變，產生出不同形狀、不同變形效果的氣球。

2. 環境參數

此部分的參數係供 ANIMATOR 做影像的真實性表現之用。包含了畫面檔 (Scene description file) 以及物體檔 (Object description file) 兩個部分 (參圖 3.11)。關於每個物體的顏色、反光程度、透明度、及 Bounding box 等則於物體檔中描述。而整個畫面的安排，包括有那些物體、物體的大小、位置、物體間的關係、觀測點、光源的位置、光源的強度、顏色、背景等等，則於畫面檔中定義。

經由這些參數的控制，使用者即可充份加以利用，以產生所要求的動畫效果。

第六章 實驗結果與評估

6.1 實驗結果

本章以一系列的相片，顯示利用前述動畫模型與方法所獲致的結果。動畫過程的表現，本應將一連串的變化過程逐一拍下，然後再連續放映出來。但限於時間上的因素，在本論文中僅以同一畫面中，數個連續變畫的靜止影像來表現其動的感覺。

6.2 評估

根據前幾章中所描述的動畫模型與方法，以及本章中的照片，可以看出，大致上都能夠將所預期的變形效果產生出來。不過，因為並沒有很完全、很精確的考慮氣球本身的物理特性，所以在壓力以及彈性係數的設定上，和真實的數值會有一段差距，以致在使用時，可能會造成氣球形狀控制上的不便。

另一方面，關於氣球的塑形，因為系統只針對掃掠軸為直線的情況處理，所以所能建構的氣球形狀也就比較有限。

再者，因為系統中對於動力學上的分析不夠完整，所以使用上必需由使用者定義較多的參數。有時甚至需要嘗試再三才能獲得所希望的效果。但相對的，使用者對動畫過程也可得到較多的控制。

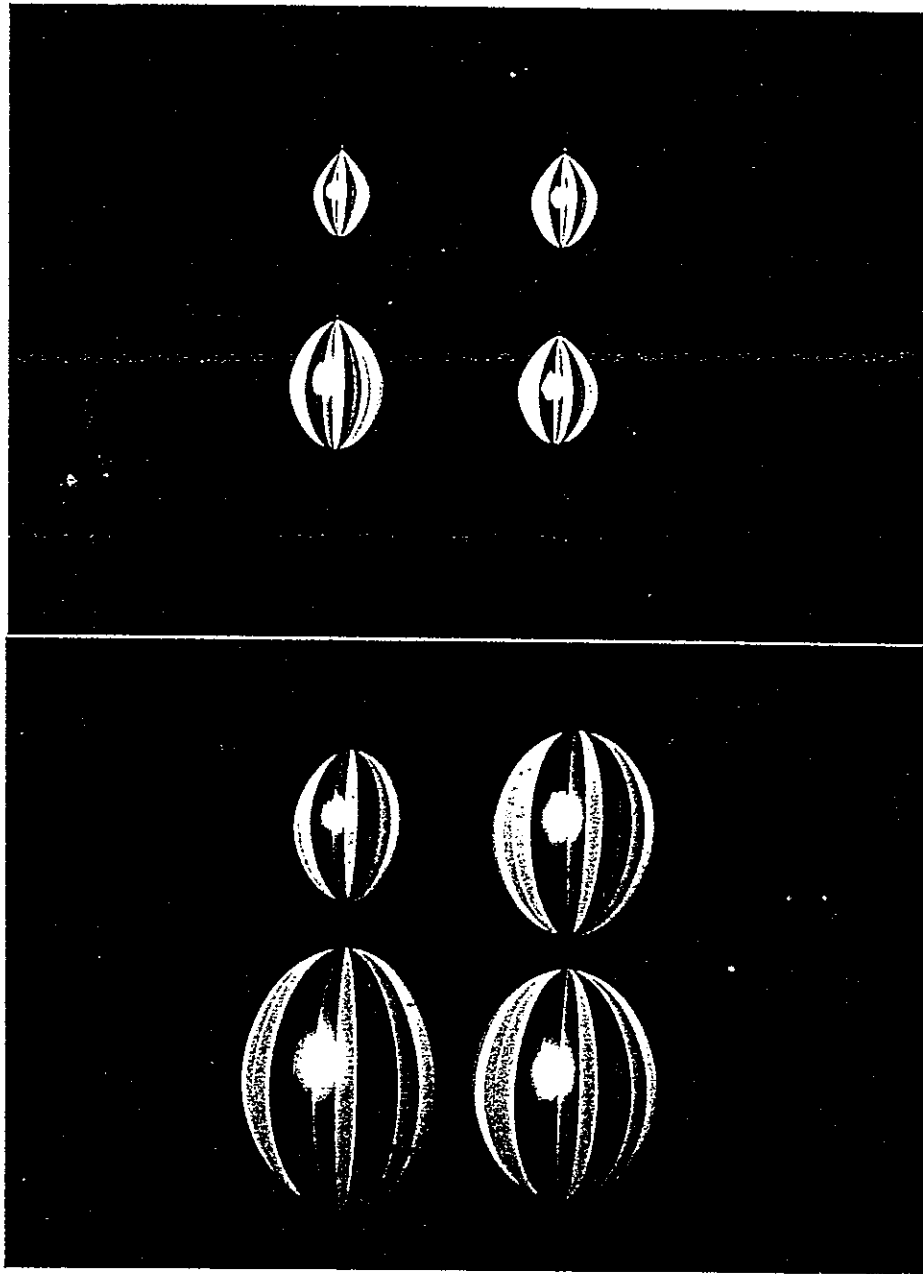


圖 6.1 氣球吹氣變形的動畫過程

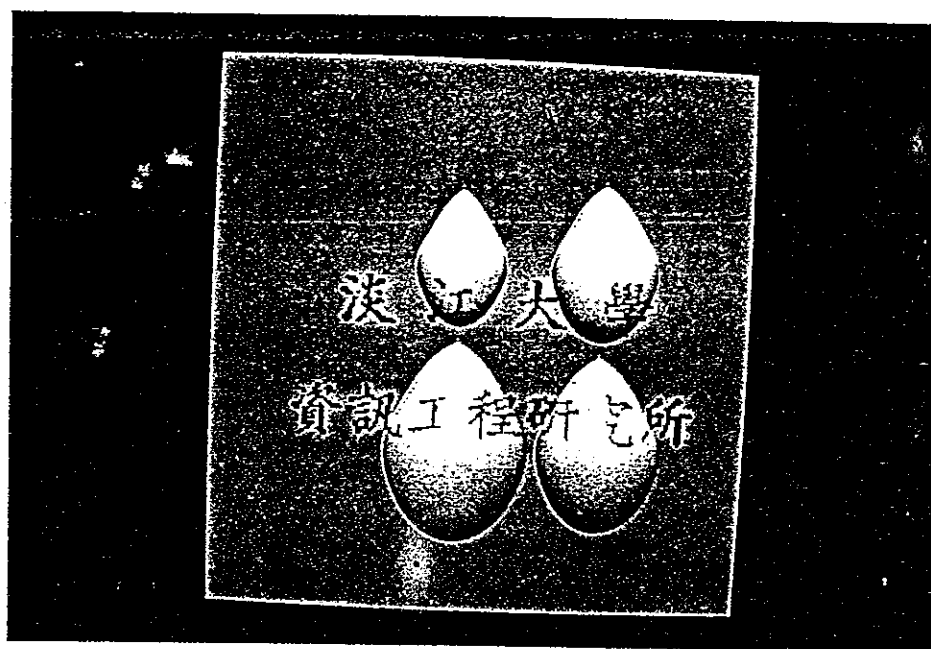


圖 6.2 氣球吹氣變形的動畫過程，加上中文字幕為背景

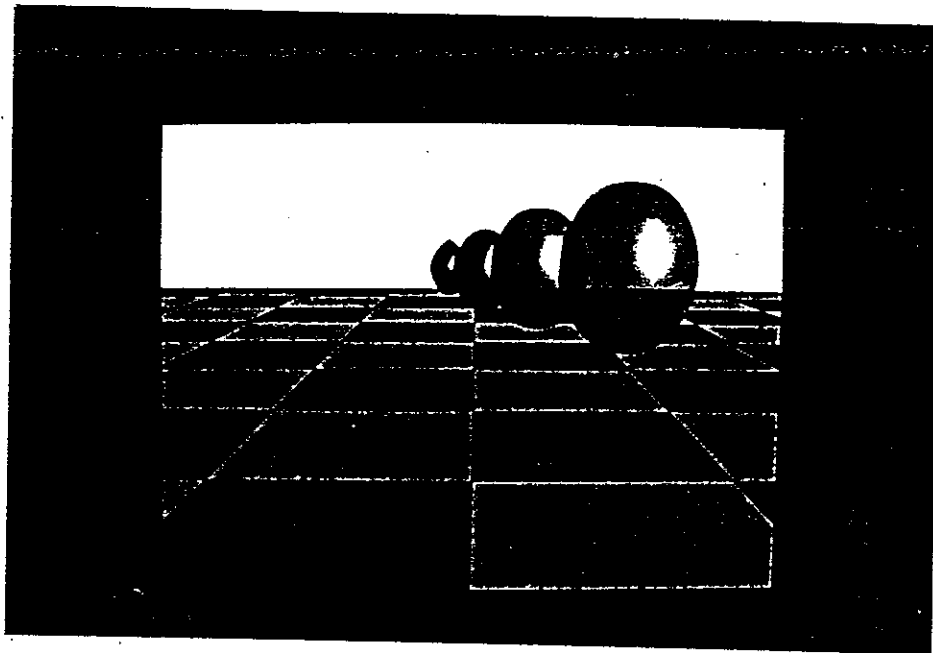


图 6.3

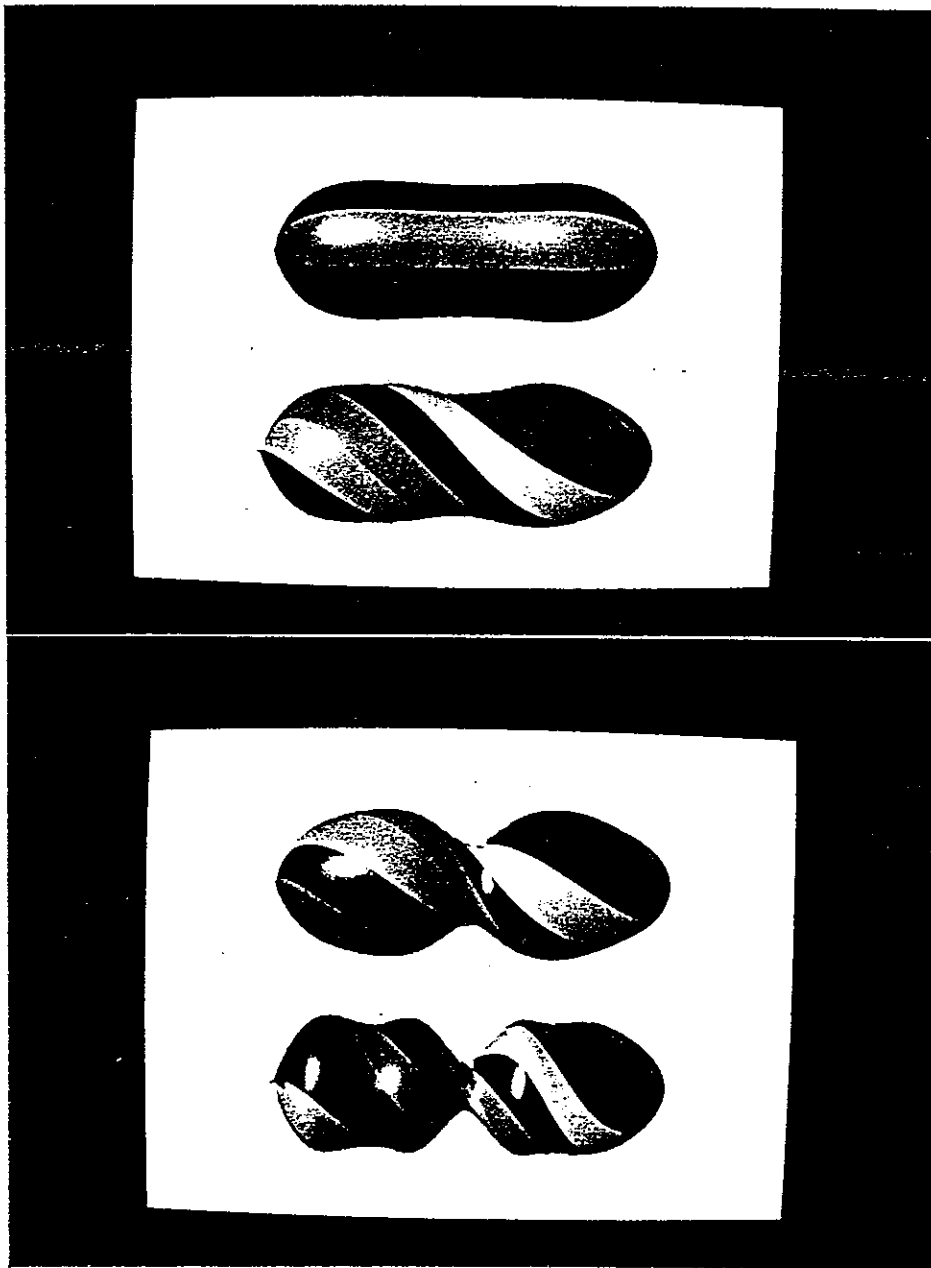


圖 6.4 扭曲變形的氣球

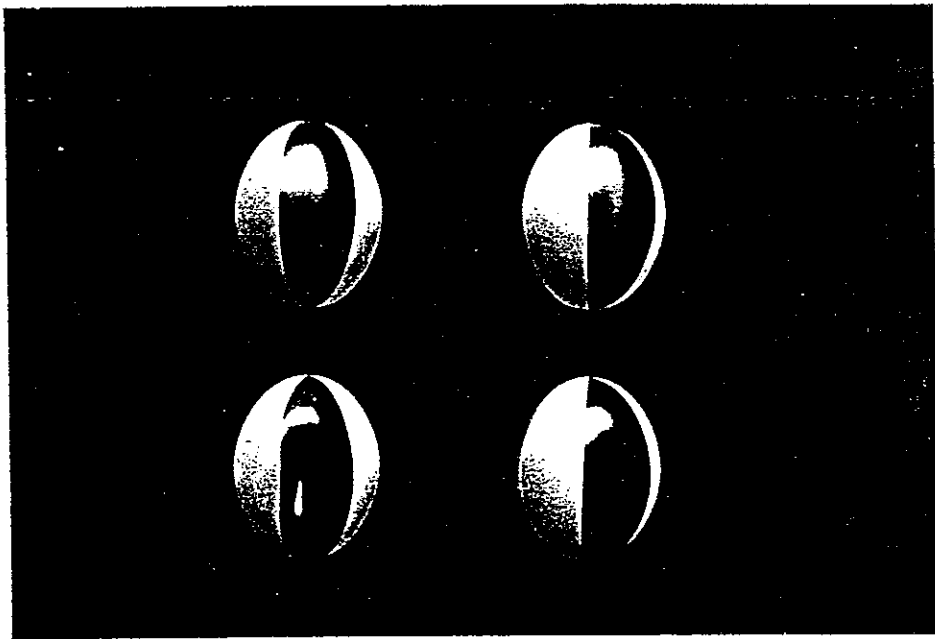


圖 6.5 氣球受擠壓變形的變化

第七章 結論與展望

經過多年來的發展，以及物理學等方面知識的輔助，電腦合成影像所能呈現的真實性與複雜度均已大幅的提高了。同樣的，這也正是三維空間電腦動畫的發展趨勢——所要求物體動作的表現，也希望能愈真實、愈複雜愈好。因此，就不得不引入物理學、數學、自動控制、人工智慧、乃至於藝術等方面的學理。在電腦動畫中，利用運動學以及動力學上的原理，一方面能夠精確的控制物體的動作；另一方面，則提供了動畫設計者模擬物體受力作用後所產生效果的能力。而且設計者並不需要對這些現象有深入的瞭解或描述。所以，這種模式也就成為廣被採用的動畫模型。

本研究的主要目的，是希望能模擬產生氣球吹氣變形的動畫效果。但是這種變形在動力學上的分析，對我們而言，過於繁複，而且無法將這些原理轉化為可處理的演算法。所以，本研究僅藉著氣球變形所發生的現象、以及基本的物理學知識，予以簡化，以模仿實際的變形效果。此外，在本研究中也以同樣的動畫模型，利用反向動力學的觀念，對於氣球受外力擠壓產生變形的情況做了模擬。所得到的效果大致上也都能夠符合現實的狀況。

當然，未來還是必須根據氣球本身的物理性質，建立一套較精確、完整的動畫模型，才能夠表現出更真實、更廣泛的實際現象。另一方面，如何將塑形、運動控制、以及真實性處理等圖形演算之可調變數加以分離，同時，也賦與這些參數適當的幾何意義與物理特性。經由程式的控制，讓使用者能以最自然的思考方式，透過電腦控制這些參數，產生所希望的動畫效果，也是將來努力的目標。

參 考 文 獻

- [1] Lasseter J (1987) Principles of traditional animation applied to 3D computer animation. Computer Graphics (Proc. SIGGRAPH'87) 21(4): 35-44
- [2] Chuang R, Entis G (1983) 3-D computer animation - step-by-step. IEEE Computer Graphics and Applications, 3:18-25
- [3] Gomez JE (1984) Twixt: A 3-D animation system. Proc. Eurographics '84 North-Holland
- [4] Magnenat-Thalmann N, Thalmann D (1983) The use of high-level 3-D graphical types in the Mira animation system. IEEE Computer Graphics and Applications, 3:9-16
- [5] Magnenat-Thalmann N, Thalmann D (1985) Computer animation : theory and practice. Springer-Verlag Tokyo
- [6] Wilhelms JP (1987) Toward automatic motion control. IEEE Computer Graphics and Applications, Apr. 11-22
- [7] Steketee SN, Badler NI (1985) Parametric keyframe

- interpolation incorporating kinetic adjustment and phrasing control. Computer Graphics (Proc. SIGGRAPH'85) 19(3):255-262
- [8] Zeltzer D (1985) Towards an integrated view of 3-D computer animation. The Visual Computer 1:249-259
- [9] Zeltzer D (1982) Motor control techniques for figure animation. IEEE Computer Graphics and Applications, Nov. 53-60
- [10] Reeves WT (1983) Particle systems - a technique for modeling a class of fuzzy objects. Computer Graphics (Proc. SIGGRAPH'83) 17(3):359-376
- [11] Reeves WT, Blau R (1985) Approximate and probabilistic algorithms for shading and rendering structured particle systems. Computer Graphics (Proc. SIGGRAPH'85) 19(3):313-322
- [12] Hahn, JK (1988) Realistic animation of rigid bodies. Computer Graphics (Proc. SIGGRAPH'88) 22(4):299-308
- [13] Wilhelms JP, Barsky BA (1985) Using dynamic analysis to animate articulated bodies such as humans and robots. Proc. Graphics Interface'85:209-229

- [14] Korein V, Badler NI (1983) Techniques for generating the goal-directed notions of articulated pictures. IEEE Computer Graphics and Applications, Nov.
- [15] Badler NI, Morris MA (1982) Modeling flexible articulated objects. Proc. Computer Graphics'82, 305-314
- [16] Calvert TW, Chapman J, Patla A (1982) Aspects of the kinematic simulation of human movement. IEEE Computer Graphics and Applications, 2(9):41-50
- [17] Fournier A, Fussel D, Carpenter L (1982) Computer rendering of stochastic models. Comm. ACM, Jun. 371-384
- [18] Smith AR (1984) Plants, fractals, and formal languages. Computer Graphics (Proc. SIGGRAPH'84) 18(3):1-10
- [19] Isaacs PM, Cohen MF (1987) Controlling dynamic simulation with kinematic constraints, behavior functions and inverse dynamics. Computer Graphics (Proc. SIGGRAPH'87) 21(4):215-224
- [20] O'Donnel TJ, Arthur JO (1981) GRAMPS—a graphics language interpreter for real-time, interactive, three-dimensional picture editing and animation. Computer Graphics (Proc. SIGGRAPH'81) 15(3):133-142

- [21] Zeltzer D (1983) Knowledge-based animation. Proc. ACM SIGGRAPH/SIGART Interdisciplinary Workshop on Motion, Toronto, 187-192
- [22] Witkin A, Fleischer K, Barr A (1987) Energy constraints on parametrized models. Computer Graphics (Proc. SIGGRAPH'87) 21(4):225-232
- [23] Isaac PM, Cohen MF (1987) Controlling dynamic simulation with kinematic constraints, behavior functions and inverse dynamics. Computer Graphics (Proc. SIGGRAPH'87) 21(4):215-224
- [24] Barzel R, Barr AH (1988) A modeling system based on dynamic constraints. Computer Graphics (Proc. SIGGRAPH'88) 22(4):179-188
- [25] Witkin A, Kass M (1988) Spacetime constraints. Computer Graphics (Proc. SIGGRAPH'88) 22(4):159-168
- [26] Armstrong WW, Green MW (1985) Dynamics for animation of characters with deformable surfaces. Proc. Graphics Interface'85:203-208
- [27] Terzopoulos D, Platt J, Barr A, Fleischer K (1987) Elastically deformable models. Computer Graphics (Proc.

SIGGRAPH'87) 21(4):205-214

- [28] Girard M, Maciejewski AA (1985) Computational modeling for the computer animation of legged figures. Computer Graphics (Proc. SIGGRAPH'85) 19(4):263-270
- [29] Moore M, Wilhelms J (1988) Collision detection and response for computer animation. Computer Graphics (Proc. SIGGRAPH'88) 22(4):289-298
- [30] Michael Barnsley (1988) Fractals Everywhere. Academic Press, Inc.
- [31] Barr AH (1984) Global and local deformations of solid primitives. Computer Graphics (Proc. SIGGRAPH'84) 18(3):21-30
- [32] Sederberg TW, Parry SR (1986) Free-form deformation of solid geometric models. Computer Graphics (Proc. SIGGRAPH'86) 20(4):151-160
- [33] Wyvill G, McPheeters G, Wyvill B (1986) Data structure for soft objects. The Visual Computer 2(4):227-234
- [34] Weil J (1986) The synthesis of cloth objects. Computer Graphics (Proc. SIGGRAPH'86) 20(4):49-54
- [35] Terzopoulos D, Fleischer K (1988) Deformable models.

The Visual Computer 4:306-331

- [36] Terzopoulos D, Fleischer K (1988) Modeling inelastic deformable viscoelasticity, plasticity, fracture. Computer Graphics (Proc. SIGGRAPH'88) 22(4):269-278
- [37] Terzopoulos D, Witkin A (1988) Physically based models with rigid and deformable components. IEEE Computer Graphics and Applications, Nov. 41-51
- [38] Thalmann NM, Thalmann D (1987) Image synthesis : theory and practice. Springer-Verlag Tokyo 1987
- [39] Newell ME (1975) The utilization of procedure models in computer synthesized images. Ph D Dissertation, Univ. of Utah
- [40] Blinn J (1982) A generalization of algebraic surface drawing. ACM Trans. Graphics 1(3):235-256
- [41] Wyvill B, McPheeters C, Wyvill G (1986) Animating soft objects. The Visual Computer 2(4):235-242
- [42] Hiroaki Chiyokura (1988) Solid modeling with DESIGNBASE. Addison-Wesley
- [43] Bronsvoort WF, Nieuwenhuizen PR, Post FH (1989) Display of profiled sweep objects. The Visual Computer,

5:147-157

- [44] van Wijk JJ (1984) Ray tracing objects defined by sweeping planar cubic splines. ACM Trans. Graphics 3(3):223-237
- [45] van Wijk JJ (1984) Ray tracing objects defined by sweeping a sphere. Proc. Eurographics'84, 73-83
- [46] Bronsvort WF, Klok F (1985) Ray tracing generalized cylinders. ACM Trans. Graphics 4(4):291-303
- [47] Tanaka T, Naito S, Takahashi T (1989) Generalized symmetry and its application to 3D shape generation. The Visual Computer 5:83-94
- [48] Burtnyk N, Wein M (1971) Computer generated keyframe animation. Journal of Society for Motion Picture and Television Engineering 80(3):149-153
- [49] 陳茂盛 (1989) 變形物體運動模糊影像之產生. 碩士論文, 淡江大學
- [50] Do Carmo MP (1974) Differential geometry of curves and surfaces. Prentice-Hall, Englewood Cliffs, NJ
- [51] Faux JD, Pratt MJ (1981) Computational geometry for design and manufacture. Halstead Press, Horwood NY

- [52] Platt JC, Barr AH (1988) Constraint methods for flexible models. Computer Graphics (Proc. SIGGRAPH'88) 22(4):279-288
- [53] 錢偉長，彈性力學，亞東書局
- [54] Fitzhorn P, Johnson G (1987) User's manual for the ANIMATOR. Colorado State Univ., Colorado
- [55] Crow FC (1982) A more flexible image generation environment. Computer Graphics (Proc. SIGGRAPH'82) 16(3):9-18
- [56] Gouraud H (1971) Computer display of curved surfaces. Doctoral Dissertation, Univ. of Utah, Salt Lake City
- [57] Rogers DF (1985) Procedural elements for computer graphics. McGraw-Hill, Inc.
- [58] Phong BT (1975) Illumination for computer-generated pictures. Comm. ACM 18(6):311-317
- [59] Kay DS, Greenberg D (1979) Transparency for computer synthesized images. Computer Graphics (Proc. SIGGRAPH '79) 13(2):158-164
- [60] Heckbert PS (1986) Survey of texture mapping. IEEE Computer Graphics and Applications 6(11):56-67

- [61] 葉阿真 (1984) 普通物理學 . 達靈頓出版有限公司
- [62] Shelley K, Greenberg D (1982) Path specification and path coherence. Computer Graphics (Proc. SIGGRAPH'82). 16(3):157-166
- [63] Gourret JP, Thalmann NM, Thalmann D (1989) Simulation of object and human skin deformations in a grasping task. Computer Graphics (Proc. SIGGRAPH'89) 23(3):21-29
- [64] Farin G (1988) Curves and surfaces for computer aided geometric design : a practical guide. Academic Press, Inc.
- [65] Bohm W, Farin G, Kahmann J (1984) A survey of curve and surface methods in CAGD. Computer Aided Geometric Design 1:1-61