

ANA : 在廣域分散式計算環境之電腦動畫系統  
ANA : Computer Animation Systems upon WAN-based  
Distributed Computing Environments

林永森 李俊賢 王宗銘  
Jeong-Sen Lin, Jiunn-Shyan Lee and Chung-Ming Wang

中興大學資訊科學研究所 圖學、多媒體暨虛擬實境實驗室  
GMVR Laboratory Institute of Computer Science National Chung Hsing University  
jslin, jslee, cmwang@cs.nchu.edu.tw

摘要

我們發展出一個整合性的廣域分散式電腦動畫系統—ANA 系統，結合廣域網路上異質的計算資源來加速電腦動畫的成圖計算時間。系統以 3D Studio 設計模型當作輸入，擁有很方便的設計動畫模型界面；同時，以光線追蹤法作成圖計算提升了動畫圖像的品質；利用廣域分散式計算節省不少成圖計算的時間；最後系統是以標準的 MPEG 動畫檔案格式輸出，提高動畫輸出的相容性。

關鍵字: 電腦動畫, 分散式計算環境, 光跡追蹤法

Abstract

We developed a integral distributed computer animation system, ANA, upon distributed computing environment connected the heterogeneous workstations on the wide area networks. ANA system provides the model transferability constructed from 3D Studio. It saves the time of designing the models. The rendering using ray tracer is upon WAN-based distributed computing environment, not only improves the quality of animation images, but also reduces much time of rendering. The animation output is MPEG-1 that saves many disk spaces and be more compatible.

Keywords: computer animation, distributed computing environment, ray-tracing

一、導論

電腦動畫[1]結合了電腦科技與藝術創作，是一種多變性及有效性的視覺表現技巧，且吸引了很

多人的興趣。隨著電腦科技日益進步，高品質和高技術的電腦動畫創造了我們想像不到的變幻空間，不但提昇了我們在視覺上的享受，更能幫助人類實現不可能的夢想。目前電腦動畫應用在各類商品、廣告、簡報系統、輔助教學系統、電腦遊戲等，可以讓整個產品更加活潑生動，更具說服力。

電腦動畫影片的製作過程，可分為動畫製作和後製作兩部份[1]。單就電腦動畫製作部份而言包括了模型輸入、成圖計算、和動畫輸出三大步驟。其中模型輸入模型還包含了故事腳本設計、模型建造、動畫編輯、材質設定等。在此三大步驟中，成圖計算(Rendering)的品質將影響一部電腦動畫的表現。由傳統之掃描線演算法(Scan-line Algorithms)，在單一計算資源上製作圖像，除了缺乏高度真實感(Photo-realism)外，一部電腦動畫的完成亦需等待長久時間。因此，以單一計算資源利用傳統演算法無論在製作速度與視覺效果方面實已不敷使用。

光線追蹤法(Ray Tracing)[2,5,6,7,12]是另一種成圖計算的演算法，它能產生高品質圖像。光線追蹤法的基本原理是當觀察者透過螢幕(Image plane)觀看物體表面時，物體的呈現就等於在模型中由光源發出的光線經折射、反射等物理光學現象後反應在物體表面的合成效果。因此，光線追蹤法可以模擬出較真實的陰影、反射、折射和透明等光學效果，產生高品質、擬似照像般真實的圖像(Photo-realistic images)。不過，光線追蹤法比傳統計算語法要耗費更多的計算時間，而且模型愈複雜，所需時間愈長。成像單張圖像時間可長達數小時，甚至數天之久。

1970 年代開始，分散式系統[3,8]隨著高速區域電腦網路而發展。近年來，學校和研究單位已普遍使用高性能的個人電腦跟工作站，這些機器並由高速網路連結成為區域網路或廣域網路。善用這些網路上強大的計算資源成一分散式系統，將可以大

大提昇原本在單一計算資源上的應用程式的效率。分散式系統的設計和技術已有非常多的相關研究和發展工作，並有很多研究單位相繼研發出平行分散式系統的發展軟體，減少了在發展分散式系統的困難。

本文目的在研究利用廣域網路的計算資源，使用光線追蹤法，發展一個分散式電腦動畫系統。在第二節中，我們首先介紹背景知識。第三節裡則介紹我們的系統 - ANA 系統設計及實作。在第四節裡列出系統的結果跟分析。最後，在第五節闡述了結論與未來工作。

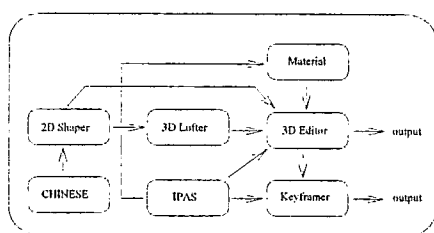
## 二、相關背景

### 1. 3D Studio 動畫軟體

3D Studio[9]是由美國 Autodesk 公司，依照其開發 AutoCAD 軟體的開放性架構，所繼續研發出的一套動畫軟體。它的特色在於我們能在個人電腦上以便宜的價格製作出相對在電腦工作站上使用昂貴的動畫軟體才能達到的效果。3D Studio 擁有人性化的界面，功能強大的 Keyframer 模組，已是目前在個人電腦上很受歡迎的動畫製作軟體。

3D Studio 可分為五個模組，分別為 2D Shaper、3D Loftter、3D Editor、Material Editor 和 Keyframer。製作動畫的流程如圖一所示。

3D Studio 有三大模組來作物體(Object)的設計。第一個便是 3D Editor 模組提供了各種基本的立體幾何模型製作，另外我們還可以利用 2D Shaper 及 3D Loftter 兩模組來製作各種形態的物體。Material Editor 模組則可以製作各種材質並在 3D Editor 模組設定給各個物體。我們所建構出的模型可以在 Keyframer 模組裡進行動畫變化的編輯，即可製作出動畫。



圖一 3DS 各個模組及其相互關係

然而 3D Studio 之成圖計算方式是採用掃描線演算法，無法模擬出真實的光影、折射、反射的效果。

### 2. 光線追蹤法

光線追蹤法[2]主要的概念是由螢幕(Image plane)逆向追蹤光線到光源，由觀察者到螢幕上的每一像素(pixel)射出多條光線(eye ray)，每個物體被光線所打到時，表面的那點顏色就會被計算。這點被

光線打到的數目和強度決定了該點是陰影或是亮點。假若物體表面具有反射或折射性質，則新的光線因此產生。如此反覆計算在三度空間模型中所有可能的光線與物體的交互作用。因為光線追蹤法是模擬真實光線的物理作用，並考慮物體的物理性質（如折射、反射性質等），故可產生很好的光線效果。

光線追蹤法用分散式計算加速已相當普遍，但通常都是用來製作單張靜畫為多，應用在動畫製作成爲一套系統尚不多見。

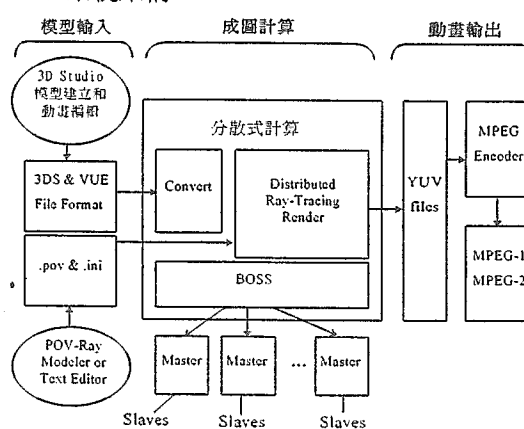
## 三、ANA 系統設計與實作

### 1. ANA 系統之目標

ANA 系統爲一整合性的廣域分散式動畫系統，因此 ANA 系統需要考慮的包含輸入模型建造、成圖計算、以及動畫輸出。系統設計目標如下：

- 採用普遍被使用的動畫模型當作輸入，並且方便編輯製作。
- 成圖計算採用光線追蹤法。
- 使用分散式計算以加快動畫製作時間。
- 分散式計算環境能擴展到廣域網路，並能聯合廣域網路上異質的計算資源。
- 以 MPEG-1 和 MPEG-2 視訊標準當作動畫輸出檔案。
- 讓使用者容易操作本系統。
- 利用外部的參數檔案使系統彈性大。
- 具有快速瀏覽單張功能。
- 具有動態的工作量平衡能力。
- 具有高度的容錯能力。

### 2. ANA 系統架構



圖二 ANA 系統架構圖

ANA 系統架構如上圖二所示，計分爲模型輸入，成圖計算與動畫輸出三大部分。

#### (1). 模型輸入

ANA 系統可接受兩種格式的動畫檔案當作輸

入: .3DS 加 .VUE 或是 .pov 加 .ini。

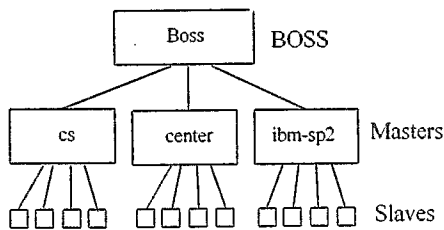
.3DS 是 3D Studio 的模型檔案, .VUE 是記錄動畫資訊的檔案, 在 3D Studio 的 Keyframer 裡可以由 Render/Setup/Make .VUE 產生出來。我們利用 3D Studio 很方便的操作界面來製作動畫模型, 這是系統的一大特色。

.pov 是公用軟體 POV-Ray[10] 的模型檔案, .ini 則是該軟體 3.0 版以後所使用的 INI 檔案。INI 檔裡描述了 POV-Ray 所使用的相關參數。例如動畫會用到的時鐘啟動(Initial\_Frame)和結束的畫面(Final\_Frame); 時鐘變數的初值(Initial\_Clock)和終值(Final\_Clock)。

(2) .成圖計算

ANA 系統的成圖計算是在廣域分散式環境上平行計算的。分散式計算是採用機器分群的方式。機器分群以使用同一個檔案系統為限制, 主程式(Boss)傳給每一群一份原始的資料檔, 群內各個機器(Slaves)再以同樣的檔案做計算。若是輸入格式為 3DS 加 VUE 檔, 在進行成圖計算前必須將檔案轉成 POV-Ray 所使用的檔案, 也就是 .pov 和 .inc 檔案。底下就分散式部份作較詳細的說明。

ANA 系統的分散式計算環境是階層式的[8], 如圖三所示。系統共分三層 BOSS, Masters, 和 Slaves。



圖三 ANA 系統的分散式架構

Boss 是主控程式, 負責將動畫以單張(frame)為一個單位, 接受 Masters 來要求工作。每個 Master 所管轄的 Slaves 都是在同一個檔案系統(file system)下, 存取同樣的輸入模型。Masters 的工作是將工作分配給 Slaves, 並做管理。真正作成圖計算工作的是 Slaves。分散式計算的流程如圖四。其中分成四個階段:

第一階段 Initialization Stage 是 Boss 啟動各個 Master, 各 Master 再啟動 Slaves, Masters 向 Boss 報告正常啟動的機器數目, 準備分配工作和計算。

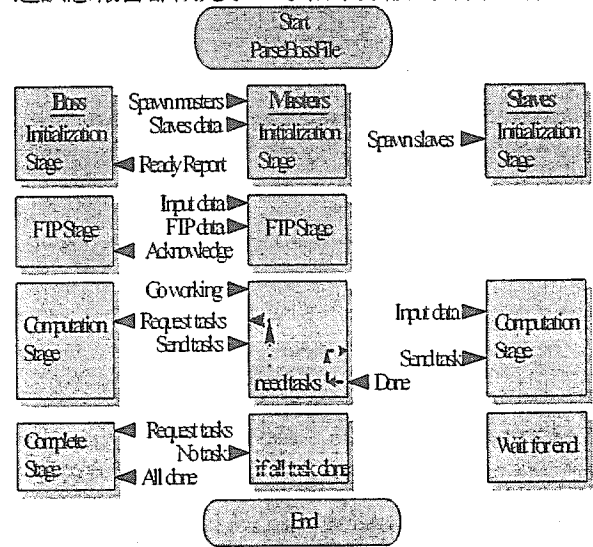
第二個階段 FTP Stage 是 Boss 將輸入的動畫模型資料以 FTP 方式傳給 Masters, 這一個階段是可以獨立出來的, 也就是說可以不在 ANA 系統裡做 FTP, 只要在下一個階段裡 Master 檢查輸入的資料都在區域的硬碟裡就行了。

第三個階段 Computation Stage 先是 Boss 傳給 Masters 輸入的資訊, 由 Masters 檢查區域的硬

碟裡的輸入資料是否正確, 若是正確, Masters 開始向 Boss 要工作, 同時也傳給 Slaves 輸入的資訊, Slaves 開始等待 Masters 分配工作。

Masters 得到 Boss 所分配的的工作後, 開始分配給所屬的 Slaves 做, 並記錄所有 Slaves 工作的情形, 在 Slave 完成一個工作(one frame)後緊接著向 Master 報告並等待下一個工作。當 Master 發現剩下的工作快沒了, 則向 Boss 要求。若 Boss 已經沒有的工作了, Boss 會進入等待結束的狀態, 為第四個階段。

Boss 進入第四階段後, 等待所有的 Masters 送訊息報告都做完了, 才結束分散式計算工作。



圖四 分散式計算流程

(3) .動畫輸出

各群機器做完成圖計算後, 將圖檔壓縮成 MPEG-1 格式, 再傳回給 Boss 同時解壓縮成 .Y .U .V 的檔案, 最後再由 Boss 壓成 MPEG-1 輸出。

四、結果與分析

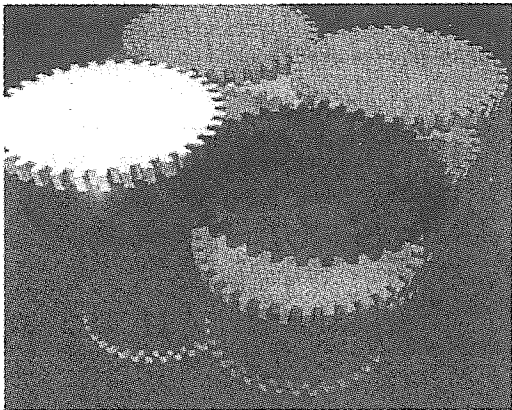
ANA 系統發展環境所使用的語言是 C 語言, 平行分散式計算環境採用 PVM (Parallel Virtual Machine) 工具。成圖計算則採用了 POV-Ray Team 所發展的 POV-Ray 3.0。MPEG-1 程式採用 Berkeley 大學的公用軟體 mpeg\_encode 和 Stanford 大學發展的 PVRG-MPEG CODEC 1.1。

1. 測試模型

Haines 曾提出一些標準模型供光線追蹤系統(ray tracing systems)來測試效率[13], 這些標準模型可由 Standard Procedural Databases (SPD) package 獲得。這些模型並非針對動畫而設計, 但我們將之修改成爲 6 秒也就是 360 個 frames 的動畫。我們是以其中一個標準模型 gears, 參數 size

為 2 當做輸入。原本模型是靜止不動的八個齒輪，我們加了 POV-Ray 3.0 的內部時間變數使八個使輪動起來，每一張畫面轉動一度。動畫畫面的解析度為 352×240，模型內含有五個點光源，畫面有處理鋸齒消除 (Antialiasing)，鋸齒消除深度 (Antialiasing Depth) 為 3。

圖五顯示動畫中的一張畫面。



圖五測試模型中的一張畫面

2. 單一計算資源之成圖計算時間

表一是一些機器成圖計算動畫第一張所需的時間。這裡最快的機器為 UltraSPARC 需要花費 239 秒的時間，而最慢是 SPARC-ELC 需要 2233 秒的時間。

	時間(秒)
UltraSPARC	239
SPARC-10	758
SPARC-5	713
SPARC-2	1878
SPARC-ELC	2233
ibm-sp2	273

表一 各種機器成圖計算所花費時間

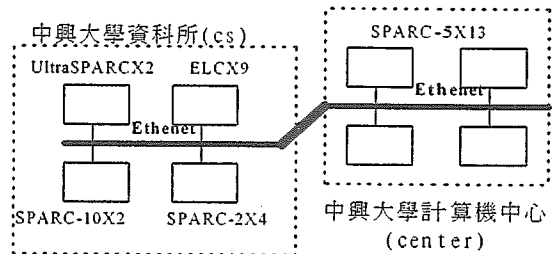
我們再分別以 SPARC-5 和 PC-Pentium-133 計算其中第 1、180、360 張畫面，來預估這兩種機器成圖計算時間需要花費多少時間。SPARC-5 的 CPU 速度是 110 MHz，記憶體為 32M；Pentium-133 的 CPU 速度是 133MHz，記憶體為 32M。表二是預估結果，請注意這些數據只是成圖計算時間，並不包括製作 MPEG 壓縮時間。兩種機器來製作動畫至少都需要花上兩天以上的時間。

單位 秒	SPARC-5	PC-Pentium-133
第1張	713	493
第180張	720	505
第360張	697	510
合計	2130	1508
預估動畫需要	2day23h	2day2h16m

表二 預估動畫所需時間

3. WAN 環境之測試一

我們第一次測試環境如圖六所示。包括中興大學資料所兩台 UltraSPARC，兩台 SPARC-10，四台 SPARC-2，九台 ELC，共 17 台。另外連接中興大學計算機中心 13 台 SPARC-5。總共有 30 台計算資源。資料所和計算機中心是以光纖連接。



圖六 測試環境一

表三是由 ANA 系統所得的數據結果。

Host Set	cs	center
機器數	17台	13台
工作Frames	147張	213張
FTP	0	5
Rendering	19249	18548
Encoder	264	641
FTP MPEGs	0	7
Decoder	86	123
	單位 秒	

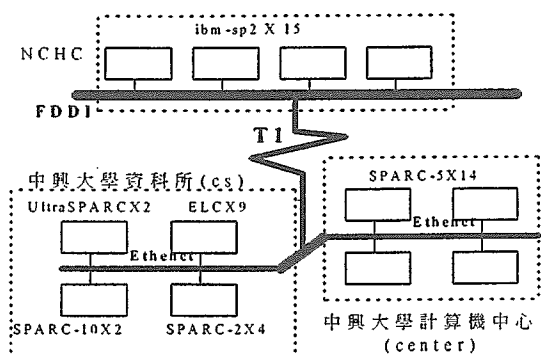
表三 ANA系統測試結果一

以 Boss 為觀點，總共花費的時間為 initialization 花費 1 秒，分散式計算總時間 (包括 PvmInitialize, FTP, Rendering, Encoder 和 Decoder) 為 19346 秒，最後壓成 MPEG-1 檔共耗費 798 秒。

其中成圖計算時間約 SPARC-5 的 13.2 倍，PC-Pentium-133 的 9.4 倍。

4. WAN 環境之測試二

我們第二次測試環境加進了高速電腦中心的 ibm-sp2 十五台，其機器架構如圖七：中興大學和高速電腦中心是以 T1 線連接 (中間距離約 80 公里)。



圖七 測試環境二

表四是由ANA系統所得的第二次測試結果。

Host Set	cs	center	ibm-sp2
機器數	17台	14台	15台
工作Frames	77張	126張	157張
FTP	0	5	28
Rendering	12456	11437	11311
Encoder	104	359	172
FTP MPEGs	0	4	780
Decoder	37	61	77
		單位	秒

表四ANA系統測試結果二

以 Boss 為觀點，總共花費的時間為 initialization 花費 4 秒，分散式計算總時間(包括 PvmInitialize, FTP, Rendering, Encoder 和 Decoder) 為 12502 秒，最後壓成 MPEG-1 檔共耗費 795 秒。

其中成圖計算時間約 SPARC-5 的 20.4 倍，PC-Pentium-133 的 14.5 倍。

### 五、結論與未來工作

藉由 ANA 系統的設計與實作我們成功地發展出一個廣域分散式動畫系統，且我們獲知一些結論：ANA 系統從 3D Studio 設計模型並經轉檔後以光線追蹤法成圖計算使我們節省不少設計模型的時間；同時以光線追蹤法作成圖計算提升了電腦動畫的品質；我們也利用了廣域網路的分散式計算，使工作效率大大提升：在 PC 上原本需要 2 天多的動畫，ANA 系統只需要不到 4 小時的時間便可製作完成。另外輸出我們是以 MPEG-1 視訊檔案輸出，使得動畫檔很容易在多種平台媒體播放。

在未來，除了對系統做更嚴密的測試和分析外，以下是 ANA 系統的未來工作：

- 增加 MPEG-2 視訊檔案輸出。
- 將 PC 納入計算資源，發展 PC 工作平台，並製作 PC 圖形使用者界面。

### 參考文獻：

- [1] M. H. Chang and C. M. Wang, "A Visualization of NCHU Campus Using 3-D Computer Animation Techniques", *Proceedings of 1995 Workshop on Computer Applications*, pp. 1-6, Taiwan, April 1995.
- [2] A. S. Glassner, "An Introduction to Ray Tracing", Academic Press Limited, 1989.
- [3] G. Coulouris, J. Dollimore and T. Kindberg, "Distributed System Concepts and Design", Addison-Wesley, 1994.
- [4] Kevin L. Gong, "Berkeley MPEG-1 Video Encoder User's Guide", Computer Science Division, University of California, 1995.
- [5] S. M. Rubin and T. Whitted, "A three-dimensional representation for test rendering of complex schemes", *Computer Graphics*, 23(4), 1980.
- [6] J.D. Foley, A. Van Dam, S. K. Feiner, J.F. Hughes, "Computer Graphics: Principles and Practice", Addison-Wesley, 1992.
- [7] A. Watt and M. Watt, "Advanced Animation and Rendering Techniques", Addison-Wesley, 1992.
- [8] J. Martin, "Computer Networks and Distributed Processing", Prentice-Hall, 1981.
- [9] Autodesk Inc., "Autodesk 3D Studio Release 4 Reference Manual", Autodesk Inc., 1995.
- [10] POV-Team, "Persistence of Vision Ray-Tracer User's Documentation 3.0", POV-Team, 1997.
- [11] C. A. Liu, "A Study of Distributed Computer Animation", Institute of Computer Science, National Chung-Hsing University, 1996.
- [12] C. M. Wang, C. W. Tseng, and S. M. Wang, "The Development of Parallel Ray Tracing System in Distributed Workstation Environments", *Fundamentals of Computer Graphics*, pp. 167-182, World Scientific, 1994
- [13] E. Haines, "A Proposal for Standard Graphics Environments", *IEEE Computer Graphics and Applications*, 7(11):3-5, November 1987.