

國立暨南國際大學資訊工程學系

碩士論文

指導教授：陳履恆博士

可調變式的 L-System

An Adaptive L-System

碩士生：石凌霖

中華民國九十三年七月二十九日

Abstract

在近年來植物成長及建構植物模型的研究中, L-system 一直是相當主流的手法。自從 1972 年 L-system 被提出後, 愈來愈多的改良讓 L-system 能夠有能力更彈性的能夠描述更複雜的植物結構, 但是隨著 L-system 的描述能力上昇, 相對來說, 撰寫 L-system 的文法也愈來愈複雜, 然而即使是使用了愈來愈複雜的文法, 在描述植物與環境間的互動還是十分的困難。這個問題嚴重的限制了使用者在使用 L-system 去模擬植物的生長。

在這篇論文中, 我們提出了一套可以解決上述問題的植物模擬系統, 名為 SimEco。基本的想法是希望能由 L-system 來保有植物學上的正確性, 再經由 particle system 把環境對植物的影響反應在植物器官的形變上, 如此, L-system 文法的複雜度相對來說就會大大的降低。經由 particle system 的使用, 也可以讓 SimEco system 產生更具有彈性和多樣性的器官形變。SimEco system 透過上述由 L-system 與 particle system 所整合成的 two layer structure 不僅簡化了 L-system 的複雜語法, 更進而對像是向光性, 背地性以及重力等等問題提出解決的方法。

Abstract

In the research field of computer graphics, L-System is the most well-known method to simulate the growth of plants and construct the models of plants. Since it was proposed in 1972, L-system has evolved into several different versions for the capability and flexibility to simulate much more complicated 3D structures of plants. However, the more powerful the L-System evolves the more complicated and difficult its grammar grows. Furthermore, it is still very difficult to simulate the interactions between the plant organs and environment. These problems seriously restrict users to simulate the growth of plants with L-System.

In this paper, we propose a simulation system named SimEco to solve the above-mentioned problems by integrating the Particle System into the L-system. The basic concepts of SimEco are to describe the hierarchical botanic structure of plants by L-System, and to perform the physical effects which act on the plant organs by particle system. The complexity of L-System grammar is therefore reduced substantially. In the meanwhile, the 3D models of organs are deformed by the trajectory of particles which represent the growth path of organs, to provide the variety and flexibility of plants modeling. Through its original two-layered mechanism, SimEco also provides a solution to perform the botanical properties of phototropism and negative geotropism, and the physical effects of collision and gravity during simulating the growth of plants. Moreover, the feedback interface between the L-system layer and particle system layer enables further interactions between plants and environment to be proceeded. Therefore, we categorize SimEco system as an adaptive L-system.

致謝

首先要感謝我的指導教授陳履恆老師，感謝老師從大學到研究所這段時間不斷的對我教導，感謝老師的包容、教誨和督促才讓我在研究的過程中走的如此順利，才讓這篇論文有見天日的一天。感謝我家人的容忍，在我最忙的時候，即使一個月才回家一次，即使一個星期才有一通電話報平安，都能夠忍受，我以後會當孝順的兒子。也感謝學校各位老師多年來的教導，尤其是李家同老師對我的關心和幫忙，也讓我能夠順利的在畢業後考上理想的學校繼續深造。更感謝實驗室裏蔡宗志學長、小包學長和 fishman 學長對我的照顧。

感謝 Taco 學長對我的照顧，在研究的過程中給我很多觀念和想法，也指引我研究的方向、相關的研究知識以及程式技巧，讓我在大學及研究所的生活中，能夠不斷的學習成長。也感謝實驗室裏的翁爺、蕭大俠、小黃、Winner、永富、沛宇、獼猴、笑雲、zard 等等實驗室同僚，讓我在研究生活之餘可以有人陪著聊天打屁解悶，給予我精神上的支持，也感謝詠智在百忙之中還常常抽空陪我玩踩地雷，最重要的是每次都被我電，讓我提振精神再出發。

好酒沉甕底，感謝善良體貼的智雯，在最後的這一段時間認識她之後，天天在深夜時都有你陪著我聊天、替我解悶、陪我說笑、陪我渡過入學口試的壓力、陪我渡過寫程式的煩悶以及趕論文的無聊期間，讓我每天等若比別人多了一段減壓的時間，讓我在這一陣子煩躁和壓力大的時間，不至於壓力過大而崩潰，讓我有寫完這篇論文的動力，不然或許走到一半我可能就雙手舉白棋投降了。

目錄

目錄	5
圖例	7
第一章 Introduction	10
1.1 研究目的與目標	10
1.2 系統簡介	10
1.3 論文編排方式	11
第二章 Previous Work	12
2.1 Particle System	12
2.2 Fractal	15
2.3 Image Based Rendering	17
2.4 Lindermayer System	19
第三章 System Architecture and Implementation	24
3.1 想要解決的問題	24
3.2 系統的流程	25
3.2.1 Plant Database	26
3.2.2 Grammar Interpreter	27
3.2.3 Adaptive L-System	29

3.3 SimEco System 加入 Particle System Layer 的優點	32
3.3.1 Particle System 的基本想法	32
3.3.2 植物器官的形變	35
3.3.3 植物的生長曲線	36
3.3.4 植物的向光性	38
3.3.5 植物的碰撞策略	39
3.4 SimEco System 的優點	42
第四章 Result and Conclusion	43
4.1 玫瑰	43
4.2 馬櫻丹(茜草科植物)	47
第五章 Future Work	50
第六章 Reference	53
第七章 附錄	56
7.1 Appendix A : SimEco L-Script BNF 文法定義	56
7.2 Appendix B : 馬櫻丹的 SimEco L-Script	59
7.3 Appendix C : 玫瑰的 SimEco L-Script	68

圖例

圖 1	預先建立好的植物運動區域.....	12
圖 2	粒子在運動區域的散布運動情形.....	13
圖 3	由 Particle System 所產生的植物影像.....	13
圖 4	簡單規則的 Fractal 取代法經過數次疊代後的結果圖.....	15
圖 5	由 Fractal 所產生的植物影像.....	15
圖 6	由 Image Based Rendering 所形成的植物影像.....	17
圖 7	Light-sensitive production.....	19
圖 8	Right-sensitive production.....	20
圖 9	1988 年 L-System 所 Simulation 的植物 Image.....	21
圖 10	Parametric L-grammar.....	22
圖 11	由 Parametric L-System 所產生的植物影像.....	23
圖 12	流程圖.....	25
圖 13	Plant Database 的主要來源.....	26
圖 14	Grammar Interpreter 的功能和架構.....	27
圖 15	L-Script 的架構.....	28
圖 16	Adaptive L-System 的架構.....	29
圖 17	SimEco system 在各階段時的資訊.....	30

圖 18	L-String 和 Particle 結合.....	32
圖 19	Particle and Particle 之間的關係.....	32
圖 20	Particle and Particle 之間的關係.....	33
圖 21	植物背地性的表現.....	34
圖 22	有 Random Factor 的影響植物背地性.....	34
圖 23	Particle System 對系統形變所產生的影響.....	35
圖 24	Rendering Engine 使用簡單模型貼合植物器官.....	35
圖 25	植物生長曲線示意圖.....	36
圖 26	沒有加入植物生長曲線的玫瑰.....	37
圖 27	有加入植物生長曲線的玫瑰.....	37
圖 28	植物向光性的呈現.....	38
圖 29	碰撞策略公式.....	39
圖 30	單一植物器官的碰撞策略.....	40
圖 31	植物多器官對障礙物的碰撞策略.....	40
圖 32	連續植物碰撞策略示意圖.....	41
圖 33	系統模擬玫瑰生長.....	43
圖 34	玫瑰各視野圖 view(1).....	44
圖 35	玫瑰各視野圖 view(2).....	44
圖 36	玫瑰各視野圖 view(3).....	45

圖 37	玫瑰各視野圖 view(4)	45
圖 38	複雜結構的玫瑰	46
圖 39	系統模擬馬櫻丹(茜草科)的成長	47
圖 40	馬櫻丹各視野圖 view(1)	48
圖 41	馬櫻丹各視野圖 view(2)	49
圖 42	馬櫻丹各視野圖 view(3)	49
圖 43	馬櫻丹各視野圖 view(4)	49
圖 44	馬櫻丹葉子所展現的透光性	50
圖 45	提供給 Database 進行 Training Data 的架構	51

第一章：Introduction

1.1 研究目的與目標

近幾年來，由於電玩形態不變的在轉變，從早期的直線捲軸，到二維平面，一直到現在的三度空間；而在畫面的處理技巧，則由早期的簡單的多邊形加上相對應物體的貼圖到現在建立複雜的三度空間和 3D models，讓遊戲畫面的立體性和真實感大大的提升，讓使用者可以依各自的需求任意旋轉，但是相對來說，如何建造一個三度空間以及 3D models 的取得也就相對變的十分的重要。

在三度空間的建立中，植物一直是場景中不可或缺的部分，所以植物模型在三度空間中的生成一直都是十分重要的，目前植物的模型取得大都偏向使用者利用 3D 模型產生輔助軟體所搭建出來，不但費時費力，所得到的植物模型只是長的像植物的模型，就植物模型的辨認度也是十分的缺乏，並沒有植物學上的正確性可以提供給予做為判斷。

本研究的目的是希望能夠藉由電腦的模擬，經過簡單的文法描述，自動的產生三度空間中的植物模型，並希望所產生的結果可以具有以下五種特性：

1. 具有植物學上的正確性，如葉序、花序等資訊以提供做為辨別；
2. 產生的植物模型是連續的生長過程，而不再只是單一的模型產生，以提供給使用者做觀查；
3. 希望所產生的植物模型要夠像真實世界的植物；
4. 能夠把環境對植物的影響反應在植物的形變上；
5. 所提供的文法描述不能太過於複雜，太難以撰寫；

1.2 系統簡介

從 1972 年 L-system 被提出後，在電腦圖學的領域上，L-system 在模擬植物生長一直有著相當好的成果，主要在於 L-system 可以把植物的特徵、成長特性寫成文法，以便透過 formal language 型的文法，經由不斷疊代的過程，使得不斷由 L-system 所疊代產生的 L-String 中，除了得到植物結構上的關係

外，也會保有該植物寫入文法中的植物特性，也使得所 rendering engine 所繪出的植物影像，得以保有植物學上的正確性。

本研究更希望透過 particle system 提升植物對環境的反應和控制，希望能夠把環境對影響反應在植物的形變，產生更真實的植物影像。

本研究繼續開發一套植物長成模擬系統，名為 SimEco system，主要採用的方式是 two layer structure，包含 L-system layer + particle system layer，希望能由 L-system layer 提供正確的植物 3D structure，再透過 particle system layer 把環境對植物的影響反應在植物的形變上，產生植物的 skeleton，最後再由 rendering engine 依據植物的骨架利用簡單的模型做貼合，得到最後的植物架構。

1.3 論文編排方式

本論文的其它章節編排如下：

- 第一章節是 Introduction。
- 第二章節是 Previous work，則是會探討前人的研究成果，並介紹和了解本篇論文的基本觀念和想法，並說明本研究之特色。
- 第三章 System architecture and implementation，主要是介紹系統的大架構，和使用 particle system 後所得到的好處。
- 第四章 Result and conclusion 則是主要來看數種由 SimEco system 所產生的植物模型和影像。
- 第五章 Future work，討論一些現在還沒有做完的問題。
- 第六章 Reference
- 第七章 Appendix 中，除了附上 SimEco system 文法描述外，也附上了每一個在論文中出現過的植物文法。

第二章：Previous Work

在植物的影像的生成上，主要可以大略分成四大部分：

- (1) Particle System (粒子系統)
- (2) Fractal (碎型法)
- (3) Image Based Rendering (簡稱 IBR)
- (4) Lindermayer System (簡稱 L-System)

2.1 Particle System(粒子系統)

Particle system 一般的做法，是在一個空間之中，由一個或數個 particle generator 射出眾多的 particles，而每一個獨立的 particle 依照運動規則對自我的運動軌跡做計算，最後再由系統把 particle 運動的軌跡記錄下來，根據運動軌跡形成影像。最初使用在植物模擬上的手法，是採用預先限制運動區域的方法，如圖一圖二所示：

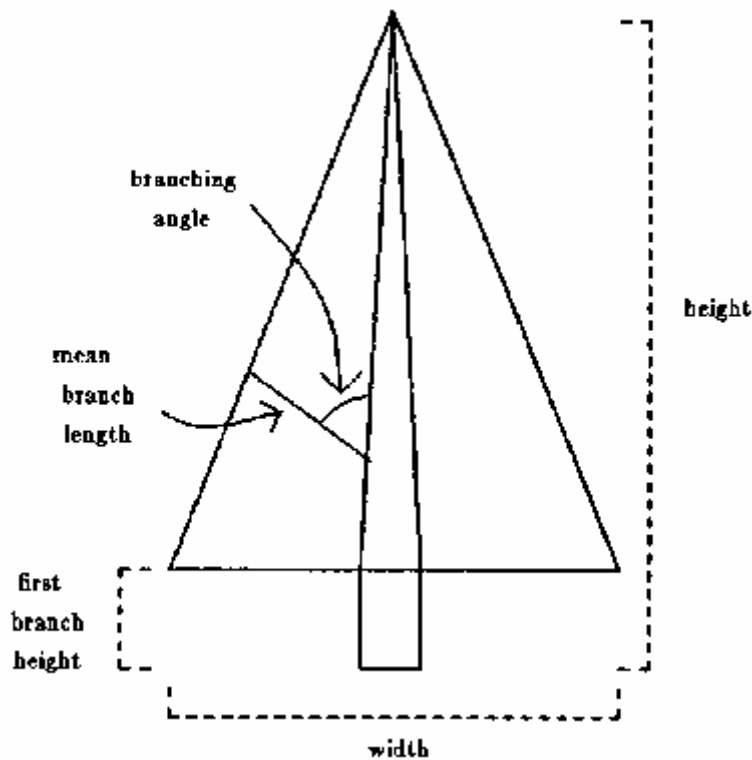


圖 1 預先建立好的植物運動區域

預先建立好的運動區域就像是局限粒子的房間，並在區域中放置 particle

generator，所散出的粒子則在區域中做 random movement，如下圖所示：

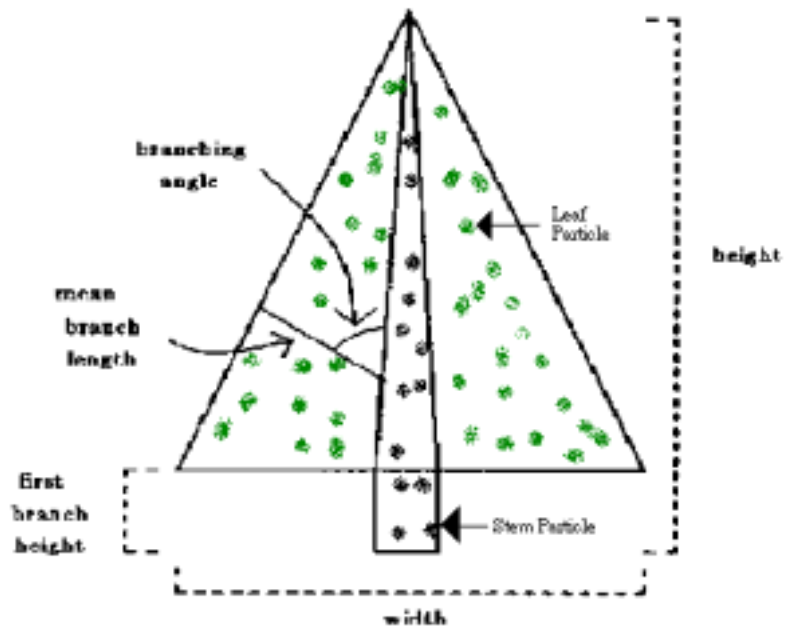


圖 2 粒子在運動區域的散布運動情形

在相對應的地方放置相對應的顏色的 particle generator，再由系統把所有粒子的運動軌跡收集起來，就可以產生如下的影像(如圖三)：

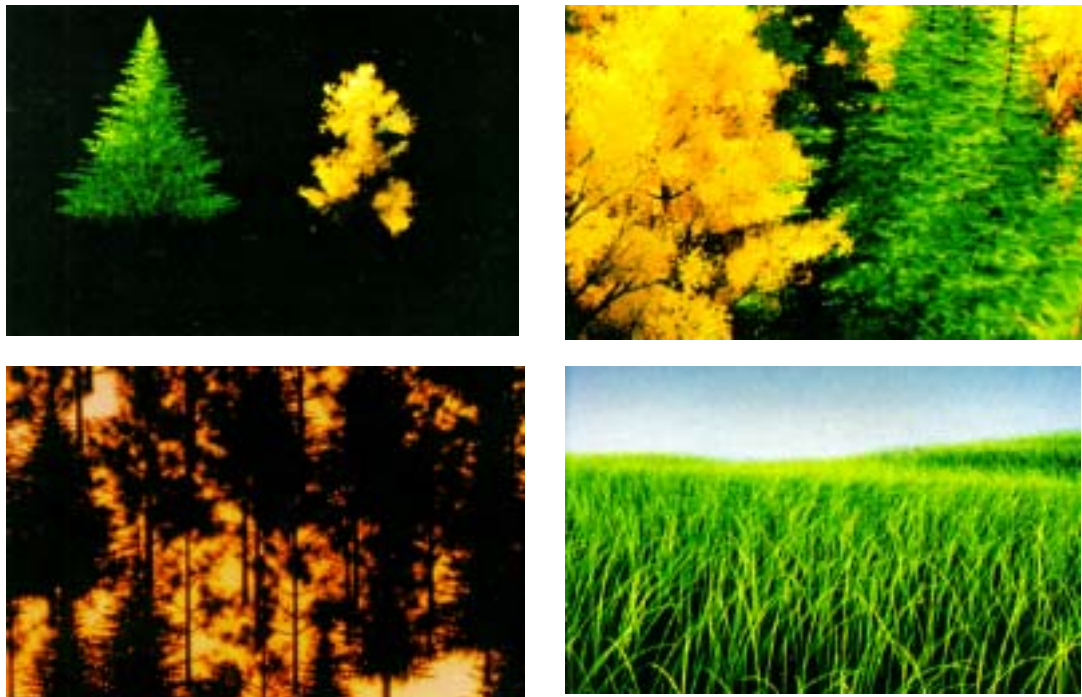


圖 3 由 Particle System 所產生的植物影像

在使用 particle system 所產生的影像中，我們可以明顯的觀察到以下的優

點：

(1) 在產生的植物影像具有相當的真實性。

() 所產生的植物影像也發現具有一定程度的朦朧感，就像是在看著類似油畫般的圖片。

但是在模擬植物的成長中，我們卻發現把透過 particle system 手法所形成的植物影像並無法達成希望模擬植物成長的部分需求：

() 所形成的只是單張沒有連續性的植物圖片。

() 形成影像所需的執行時間過於冗長。

() 需要花費相當的時間預先規劃 particle 在空間中的限制區域。

() 在產生的植物影像中，並無法明顯發現出植物的一些基本特性，如葉序、葉子的形狀和貼圖等等。

2.2 Fractal (碎型法)

在 fractal 所使用的方法則是利用著一些簡單的取代規則，經過不斷重複的制式的不斷疊代，所形成的 3D structure，並在三度空間中使用 rendering 的手法把植物的影像在 3D 空間中描繪出來。下圖是經過數次的 fractal 疊代後所形的植物 structure(如圖四)。

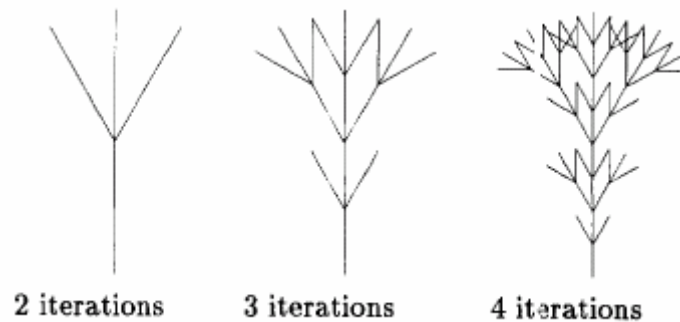


圖 4 簡單規則的 Fractal 取代法經過數次疊代後的結果圖

使用用 fractal 所產生出來的影像如下(如圖五)：



圖 5 由 Fractal 所產生的植物影像

在使用了 fractal 後所產生的影像中，我們可以明顯的感受到：

- () 由於取代的規則數量少而簡單，所以在產生植物的影像時，並不會花費太多時間。
- () 所使用的是 structure-oriented 的架構，所以可以產生的不只有單一

張植物影像，而是可以產生連續性的植物影像。

但是我們也發現了，經 fractal 所產生具有 structure-oriented 的植物影像中，無法達成我們部分需求：

- () 由於取代規則的簡單化，所產生的植物影像過於規則化，使的所產生的植物影像和真實世界的植物影像有段差距。
- () 在產生的植物影像中，並無法明顯發現出植物的一些基本特性，如葉序、葉子的形狀和貼圖等等。
- () 由於規則是 deterministic-ruler，使的產生的影像缺乏變化。
- () 缺乏和 environment 之間的互動。

2.3 Image Based Rendering

使用 image based rendering 的手法，所造成的植物影像如下（如圖六）：



圖 6 由 Image Based Rendering 所形成的植物影像

在使用 image based rendering 中，我們使用了大量的真實植物影像做插圖，利用夾角和影像內插做中間過渡角度的影像，我們可以明顯感受到：

- () 在大量植物的生成和繪畫時，影像的生成幾乎可達到 real time。
- () 在遠距離觀看時，植物具有相當高的擬真程度。
- () 一般的情形下，可以很直覺的判斷出植物的種類。

不過由於是由植物的影像所形成的感覺，所以以下有部分的效果是比較難以達成的：

- () 由 image based rendering 所形成的 3D 空間中，每個植物虛擬地點處並沒有真實的 3D 植物生長，在空間分佈中，並不能提供準確的資料給系統做些判斷，如 collision detection 等等。
- () 在前置處理時，就必須先取得該植物的固定角度的 image，才可以透過內插合成的做法，得到中間過渡影像。

- ()無法以任意近距離觀察所形成的植物影像，很容易出現失去立體感的情形。
- ()由於在 3D 空間中並沒有該植物的 3D 架構，所以在影子的處理方面也是採用貼圖的方式，對於植物影子的顯現並不容易得到正確的植物影像。
- ()在做連續植物的生成模擬時，需要具備連續的植物影像，而且所得到的植物影像變化量也降低不少。

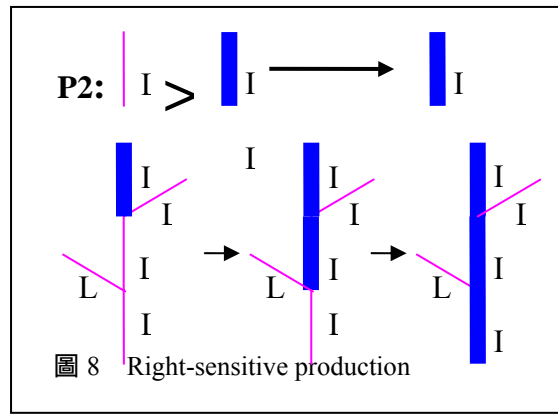
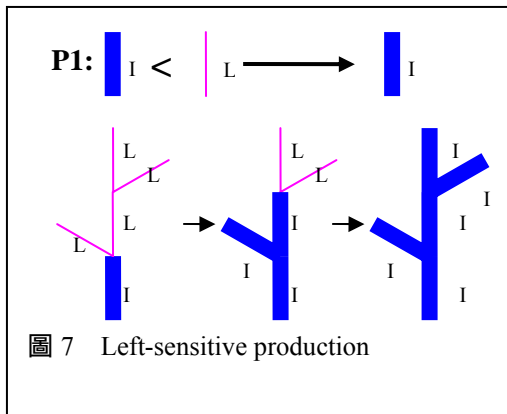
2.4 L-System

從 1972 年 L-System 被提出後，在電腦圖學的領域上，L-system 在模擬植物生長一直有著相當好的成果，主要在於 L-system 可以把植物的特徵、成長特性寫成文法，以便透過 formal language 型的文法，經由不斷疊代的過程，使得不斷由 L-system 所疊代產生的 L-string 中，除了得到植物結構上的關係外，也會保有該植物寫入文法中的植物特性，也使得所 rendering engine 所繪出的植物影像，得以保有植物學上的正確性。近年來，關於 L-system 的研究中，也不斷的在文法上作改善，除了希望可以加強對植物的各種成長特性的模擬正確性，也希望可以加強由環境變數對 L-string 的影響。

從對植物的觀察中，我們發現植物本身的多樣性、環境對植物的影響、和植物內部的訊息傳遞，為了能充分表現諸多特性，在本研究中決定採用 L-system，並加入下列 L-system 特性：

1. $\langle k, l \rangle$ L-system properties

在植物的生長過程中，常常可以發現單一的植物器官，除了本身的成長考量外，也常常會受到相鄰的器官影響，如引力、日照、養份傳遞、空間競爭等等，進而改變自我成長的狀態，而在 $\langle k, l \rangle$ L-system 中，則可以充分的表現出這個特性。在 $\langle k, l \rangle$ L-system 中， k 代表 left-sensitive 而 l 代表的是 right-sensitive，代表著一個在 L-system 中的 alphabet 會受到 string 中繼承關係相近的左邊 k 個相近的 alphabets 的影響，和繼承關係相近的右邊 l 個相近 alphabets 的影響，而做出不同的 grammar selection。圖七和圖八則分別是 left-sensitive and right-sensitive 的示意圖。



2. Non-deterministic system properties

走在校園裏，常常會發現在校園的各個角落中，到處都有數棵相同種類的植物生長在一起，明顯可以感受到植物具有相同的特徵；在仔細靠近時，則會發現在植物的細微處，則有些 random factor 在裏面，甚至於植物的生長規則都有大同小異之處，而在 non-deterministic system 中，則是強調以 multi-rulers 來突顯植物學上的這個特性，可以提供系統在相似的環境因素影響下，借由 Random factor 的影響來得到生長的細微差異，甚至於是植物生長規則上的大同小異之處；存在 L-grammars database 裏的植物文法，一開始擁有相近被選取的機率，而在 SimEco system 不斷的做成長模擬後，所產生的 feedback 則會不斷的去 training L-grammars database，進而得到較符合植物學上正確性的選取機率。

3. Non-propagating system properties

夕陽斜掛在地平線的一端，金黃色的光芒灑落滿地，男女主角相擁，一起站在葉子紛飛的樹林中，看著屬於他們的未來。在一般影片中常看到的情景，正突顯了植物的另一個特性，一棵植物產生的器官都有可能消失，比如像是樹葉的掉落、樹幹受損等等，而在 L-system 植物的字串表示中，代表植物 strings 的每一個 alphabet 也都有機會消失，而 non-propagating system properties 就是允許 L-system 可以

消去植物 strings 中的 alphabet，使得植物 organs 的死去和掉落可以被直覺的 implementation。

在早期 L-System 所得到模擬的植物影像就已經有不錯的影像效果，圖 9 是 1988 年所提出論文由 L-system 所產生的植物影像：



圖 9 1988 年 L-system 所 Simulation 的植物 Image

在上圖我們可以明顯的感受到所產生的植物影像過於規則化，所產生的植物架構並不像現實生活中所看到的植物，所以之後 L-system 上的研究，主要是透過 L-grammars 上的不斷改進和強化，希望能夠提供使用者更強大的 L-grammar，讓使用者可以用來描述更複雜更真實的植物結構。

Parametric L-system 則是其中最重要也是最具代表性的，主要的作法是使用參數傳遞的方式，進而在各 grammars 之間傳遞時間訊息和其他資訊，讓文法描述功能性更為健全，以下是部分 parametric L-system 的文法描述：

```

#de_ne m 3 = _ plastochron of the main axis _ =
#de_ne n 4 = _ plastochron of the branch _ =
#de_ne u 4 = _ signal propagation rate in the main axis _ =
#de_ne v 2 = _ signal propagation rate in the branch _ =
ignore : + _=
! : S(0)F(1; 0)A(0)
p1 : A(i) : i < m_1 _!A(i + 1)
p2 : A(i) : i == m _ 1 _! [(60)F(1; 1)B(0)]F(1; 0)=(180)A(0)
p3 : B(i) : i < n_1 _!B(i + 1)
p4 : B(i) : i == n _ 1 _! F(1; 1)B(0)
p5 : S(i) : i < u+v _!S(i + 1)
p6 : S(i) : i == u + v _! "
p7 : S(i) < F(l; o) : (o == 0)&&(i == u _ 1) _! #F(l; o)!S(0)
p8 : S(i) < F(l; o) : (o == 1)&&(i == v _ 1) _! #F(l; o)!S(0)
p9 : S(i) < B(j) _!"

```

圖 10 Parametric L-grammar

而使用 Parametric L-System 所產生的植物結構如下：

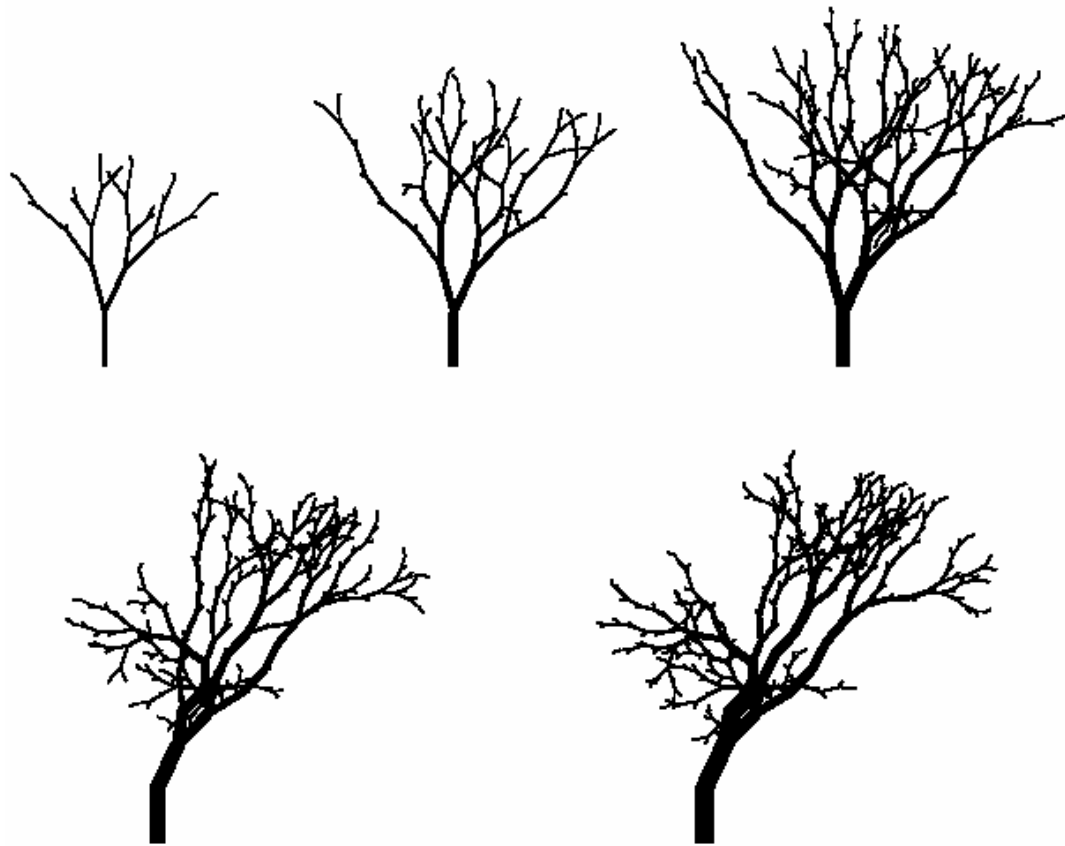


圖 11 由 Parametric L-System 所產生的植物影像

我們可以明顯的發現由 parametric L-system 所產生的植物影像已經明顯和之前 L-system 所產生的植物影像有明顯多樣化和真實性上的差別，由 parametric L-system 所產生的植物影像不但貼近真實世界所看到的植物結構，更在植物多樣性上遠遠超過之前的 L-system。

不過由於 parametric L-system 在導入時間訊息和參數傳遞進入後，也使的 L-grammar 變的十分的複雜，不僅僅是可讀性大大的降低，也讓 L-grammar 的撰寫變的艱難耗時，使的 parametric L-system 的實用性相對來說也大幅的降低。

第三章 System Architecture and Implementation

3.1 想要解決的問題

在發展架構在 L-system 上的植物模擬系統時，發現有些部分問題需要克服，在 previous work 一連串的討論之中，我們發現了數個需要克服和解決的問題：

1. 量化數據的缺乏，使的在撰寫植物文法時，大都是使用者不斷的經過 try and error 的方式，經由費時費力的不斷修正，最後得到有較接近真實植物的影像結果。
2. 雖然 parametric L-system 的文法十分的強大，而且能夠表現植物的多樣化，但文法太過複雜難以撰寫，使的使用者在使用上諸多不變和難以學習，我們則試著去把參數化的部分自文法部分抽離，使用較原本較簡單文法結構的 L-system，再使用 particle system 來補足 parametric L-system 靠參數化所卻傳遞的資訊部分。
3. 雖然使用了像 parametric L-system 這麼複雜的植物文法，在描述和表達植物和環境之間的互動時，仍然十分的困難，我們試著藉由整合進 particle system，希望能把植物和環境之間的互動做更明顯和直接的表達。

因為 1.2.3 所提到的，我們希望能夠提出一套較完整的想法來改善這些問題，最終目標希望能夠提供使用者方便以及較容易上手使用的 L-system，讓使用者可以更真實的模擬植物生長。

3.2 系統的流程

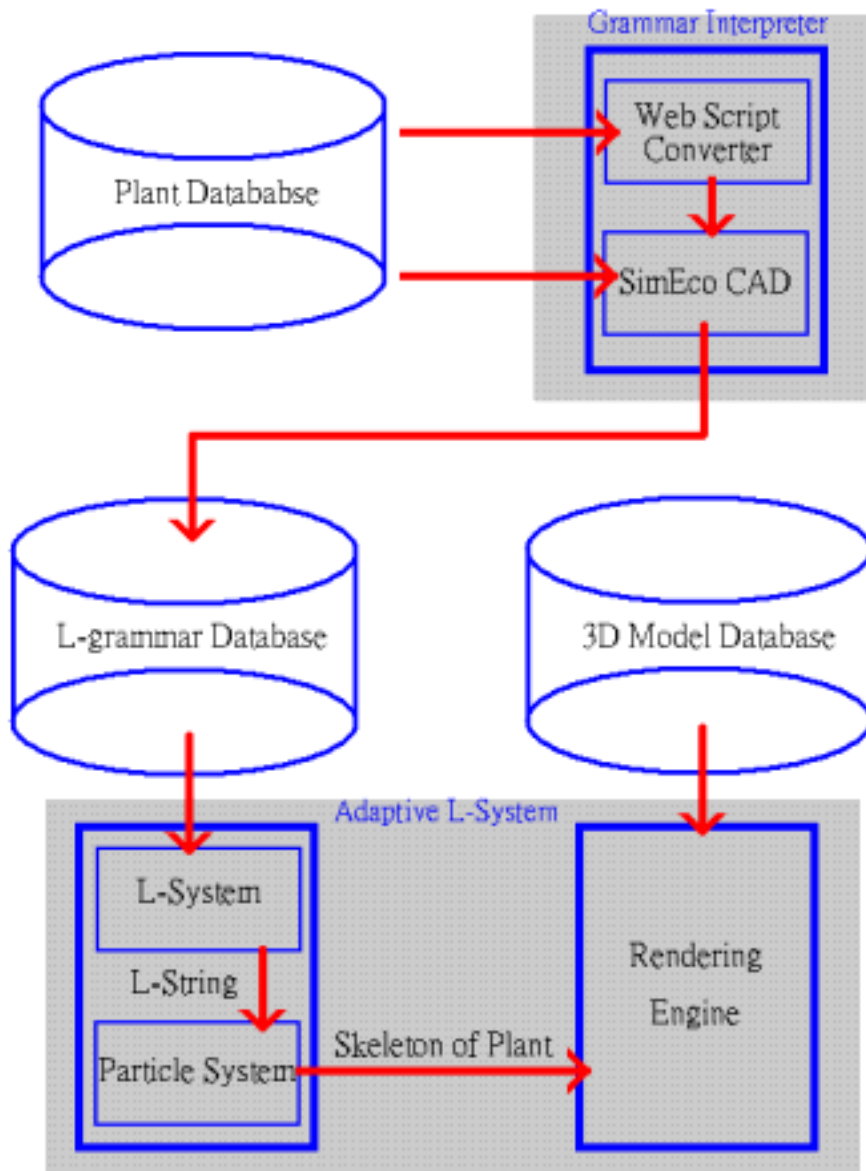


圖 12 流程圖

上圖是我們計劃主要的流程圖，我們主要是希望能夠透過 grammar interpreter，從 plant database 抽取出撰寫 L-grammar 所需要的資訊，進而產生 L-Script。

而 adaptive L-System 則會從 L-grammars database 中，選出該植物的描述，經過 rewriting system，所產生的 L-string 則會保有該植物特性，在經過 particle system 產生植物 skeleton 之後，再由 rendering engine 依照植物骨架貼合形成植物模型。

3.2.1 Plant Database



圖 13 Plant Database 的主要來源

在資訊發達的時代中，網路資料一直是很重要的來源，我們的 plant database (如圖 1 3) 主要也是希望可以從網路上抓取我們在撰寫植物文法時所需要的資料，主要分成對植物的文字描述，植物影像二個部分，我們希望可以從文字的描述中找到植物特性相關的描述如葉序、花序等，再從網路上植物的影像中找出部分量化數據，像是莖長多粗，夾角幾度，葉子長多大，比例如何等等。希望能夠降低使用者在撰寫文法時做 try and error 的時間。

3.2.2 Grammar Interpreter

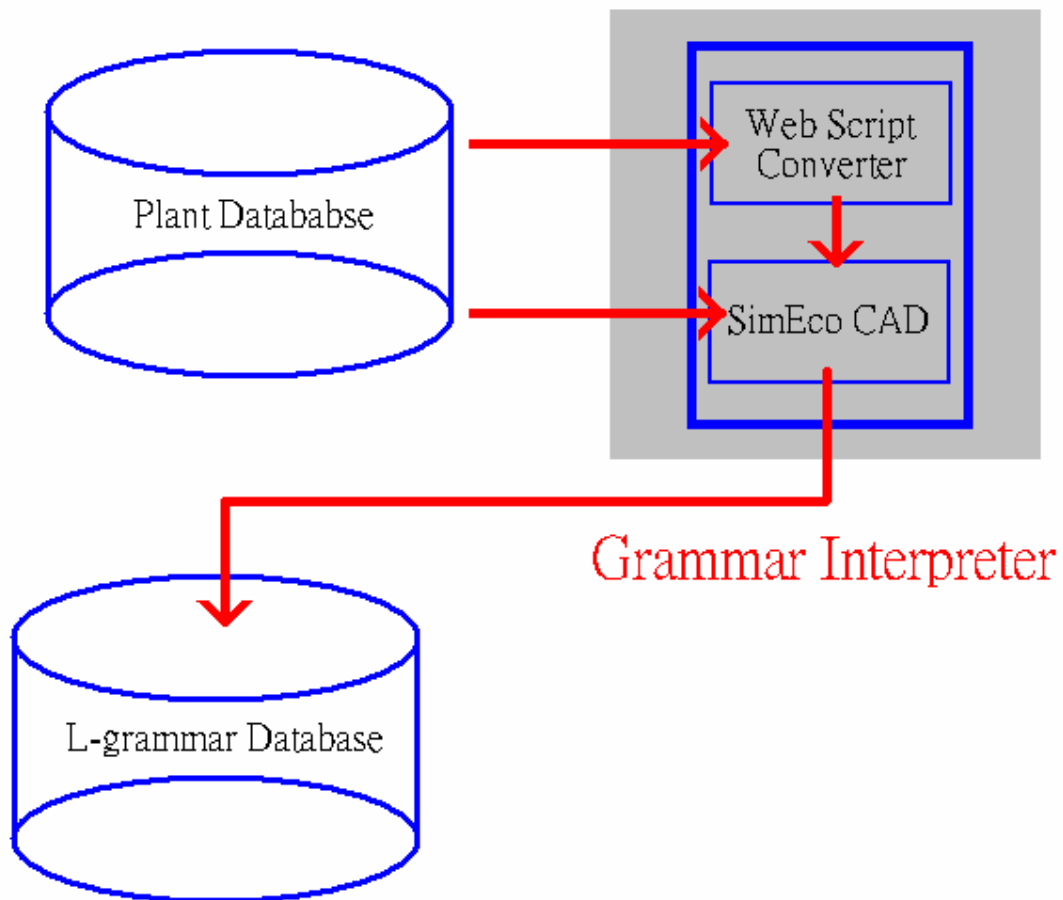


圖 14 Grammar Interpreter 的功能和架構

Grammar interpreter 主要分成二個部分，web script converter 主要是會從 plant database 中，由植物文法描述中做 data mining，取得某些植物重要植物特性，建立 L-script 的雛型；接著再經由 SimEco CAD 對 plant database 中的該植物影像做 image processing 的動作，取得無法由網路文字描述所提供的量化數值，例如：植物的葉序的夾角角度、主幹和分枝粗細比例、葉子的大小比例、植物生長曲線等等資訊，補足由 web script converter 所提供的 L-script 雛型，建立完整的 L-script，並存入 L-grammar database 裏面，希望能夠建立 L-grammar database 提供給做 L-system 相關研究做使用。

而存放在 L-grammar 的 L-script，主要是採用 hierarchical 架構，並提供使用者可以在 L-script 定義各種屬於環境和植物的相關變數和影響，其架構如

下：

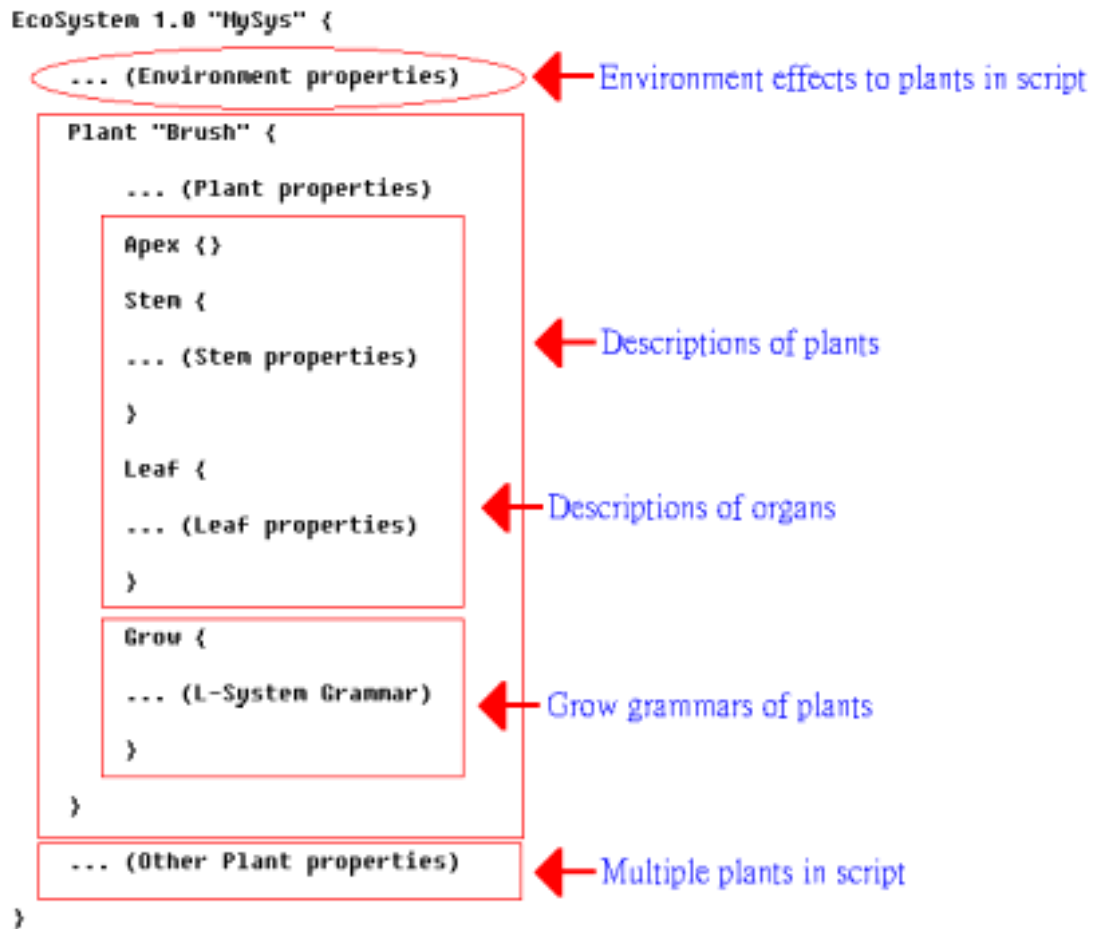


圖 15 L-Script 的架構

SimEco L-Script 允許使用者在一開始定義環境相關的 global effect，像是重力和光源的位置等等，在 SimEco L-script 中可以同時定義多棵植物，每個植物定義主要分成三個部分：

- (1) 植物基本特性定義，例如生長位置、生長方向及生長深度等等。
- (2) 植物各器官和各器官狀態的定義。
- (3) 植物生長文法。

這方面主要是由實驗室黃俊育同學做相關研究，如果有興趣請參閱他的論文研究，在本篇論文中就不再多做冗長的描述。

3.2.3 Adaptive L-System

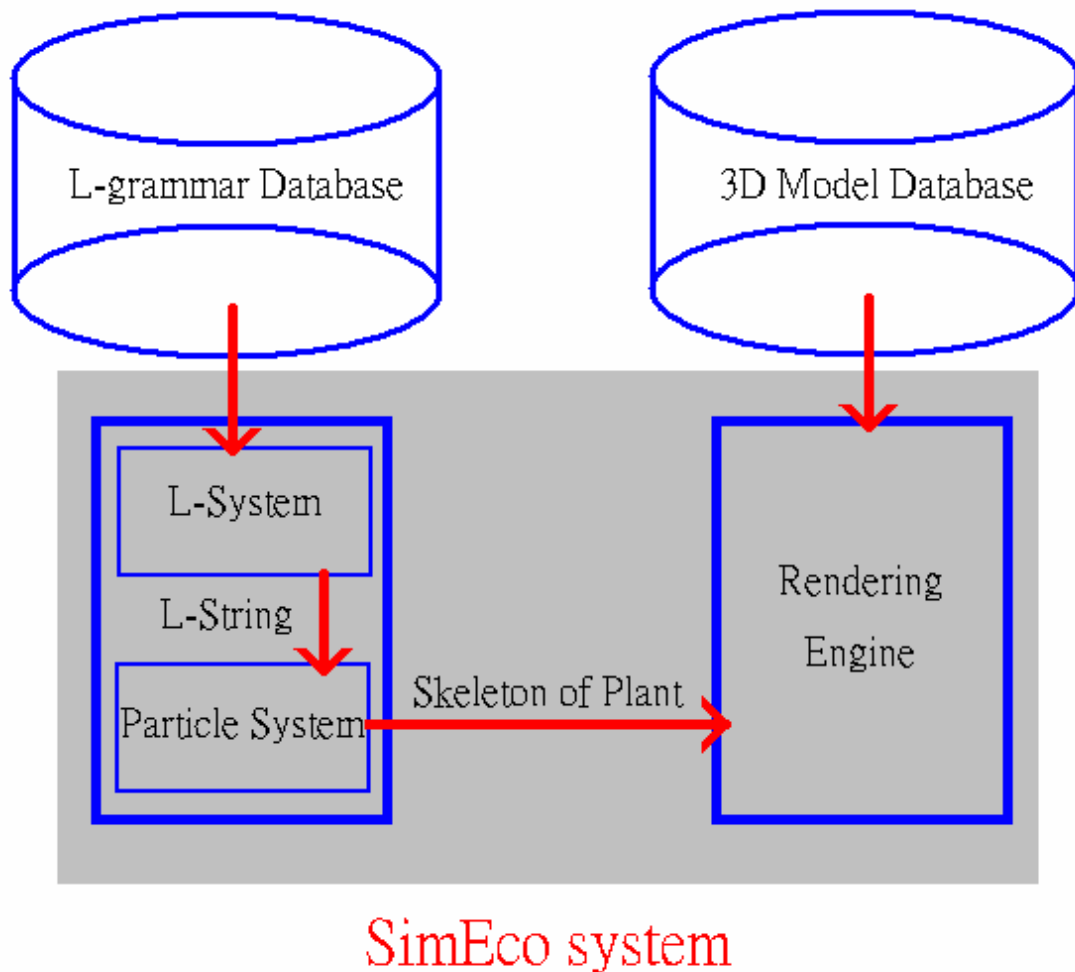


圖 16 Adaptive L-System 的架構

這也是本篇論文最重要的部分，有關 SimEco system 主要是採用 two layer structure，包含：

- (1) L-system layer;
- (2) Particle system layer;

L-system layer 會由 L-grammar Database 中把取得所對應的 L-script，從 L-script 中取得疊代該植物時所需要的 L-grammar，透過 rewriting system 不斷的把植物特性寫入 L-string，使用由 L-system 所產生的 L-string 具有植物學上的正確性。

Particle system layer 則會從 L-system 所產生的 L-string 做處理，把

L-string 中每個 alphabet 視為獨立的 particle generator，每個 particle generator 都會射出 particle，而 particle 在空間中會依照基本的物理模型在 3D 空間中做運動，而 particle system 則會把每個 particle 在空間中運動軌跡記錄下來，產生植物的 skeleton。

Rendering Engine 則會依照由 particle system 所產生植物的 skeleton，使用相當簡單的 3D models 來沿著 skeleton 做模型的貼合，最後得到植物的影像，其示意圖如下：

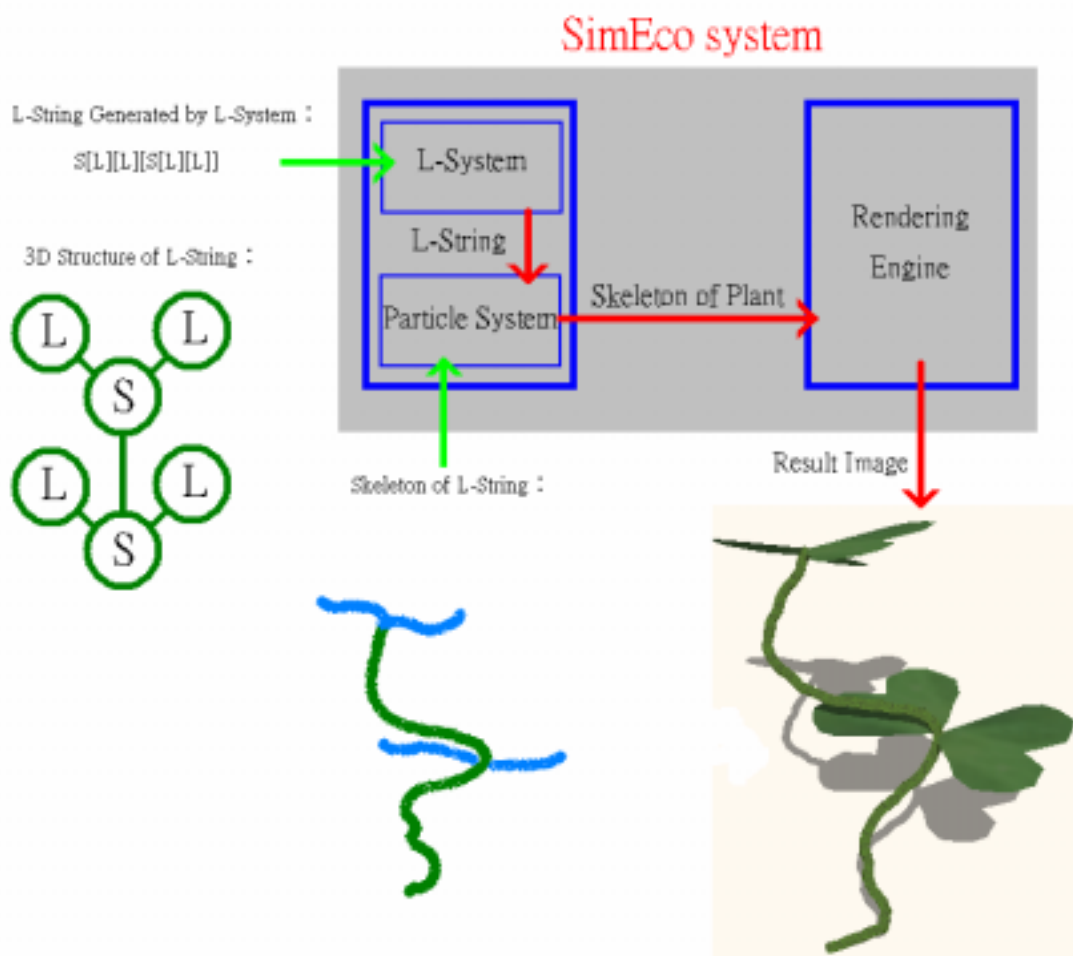


圖 17 SimEco system 在各階段時的資訊

則 L-system 所產生的 L-string 雖然是以 string 的型態存放在 system 中，但是其實 L-string 是具有 3D 結構的，由 L-string 所保有的 3D 資訊，系統可以在 3D 空間中，利用圓型的小球以堆積的方式，堆成一棵三維空間的樹狀結構，而每個圓型小球都會被 particle system 視做獨立的 particle generator，

particle generator 會射出 particle 在三維空間中運動，經由計算加速度和外力的影響，我們可以把環境對植物的影響直覺的反應在 particle 的運動軌跡上，而由 particle generator 所射出的每個 particle 在空間中所形成的運動軌跡都會被 particle system 收集起來當做是植物的 skeleton，由於運動軌跡是植物 organs 外形重要的依據，換句話說，我們也就可以輕易的把環境的影響，輕易的透過 particle system 反應在植物的形變上。

在接下來的部分也是本研究最主要的核心部分，主要是討論如何結合 particle system 以及基本的物理模型和植物學上的觀點來反應來解決 L-system 在模擬植物生長時會遇到的生長形變以及生長問題，主要希望能夠得到以下期望得到的優點：

1. 可以把環境對植物的影響反應在植物器官的形變上。
2. 可以降低 SimEco L-script 的植物文法撰寫上的複雜度。

3.3 SimEco System 加入 Particle System Layer 的優點

3.3.1 Particle System 的基本想法

在 SimEco system 中，是採用 two layer structure，L-system 和 particle system 結合的情形如下：

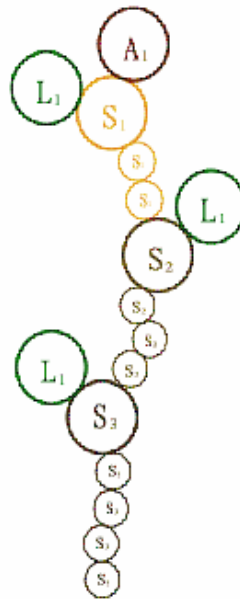


圖 18 L-String 和 Particle 結合

每個 organ 都被視為是個 particle generator，而每個 particle generator 則會在則會在空間中射出 particle，而 particle 和 particle 之間的關係如下圖所示：



圖 19 Particle and Particle 之間的關係

每個 particle generator 都會繼承前一個 parent particle generator 的末速度和終點位置，當做自己的初使位置和初速度，由 particle generator 所射出的 particle，則會由 particle generator 地方繼承初速度和初使位置，在 3D 空間中，會受到各方面的引力如重力或向光性，以及 random factor 影響，在空間中做等速、等加速、變加速等等運動，而在空間中做運動。

把問題簡化成單一 particle 在空間中的運動問題後，就可以輕鬆套用運動模型來計算，把二維的運動模型套用到三維空間中，如下圖所示：

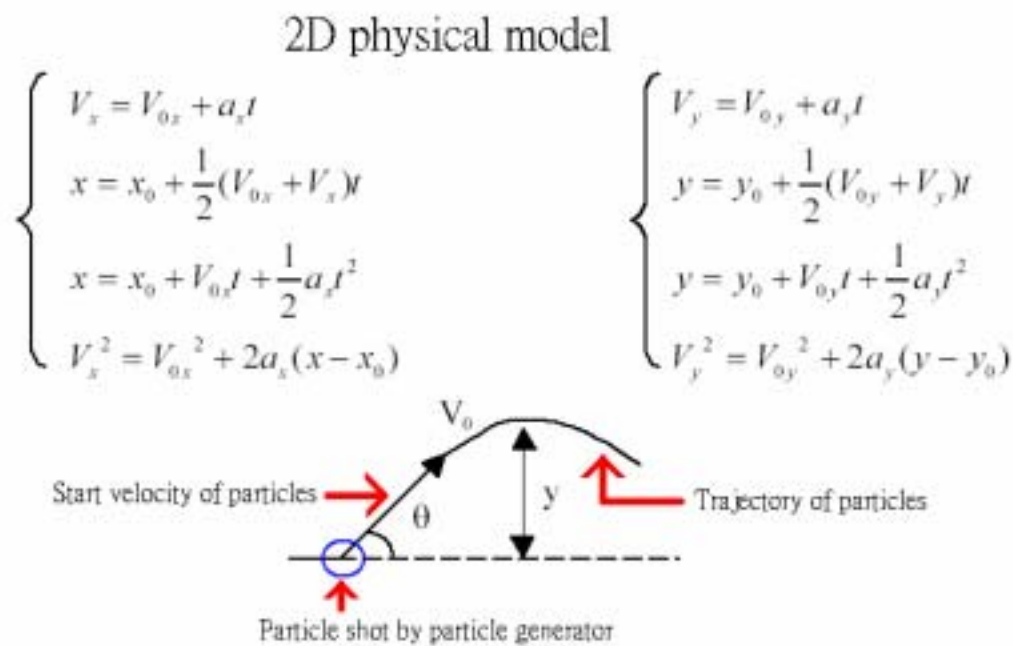


圖 20 二維平面運動軌跡的計算

每個 particle 在 3D 空間中，經由物理模型可以計算得到新的位置和新的向量，而運動過後的軌跡則會被 particle system 記錄下來，在模擬背地性時，則可以使用一虛擬向上加速度，則利用物理模型可以表現植物背地性在植物成長模型上，就如下圖所示，我們可以簡易表現植物的背地性在植物模型上：

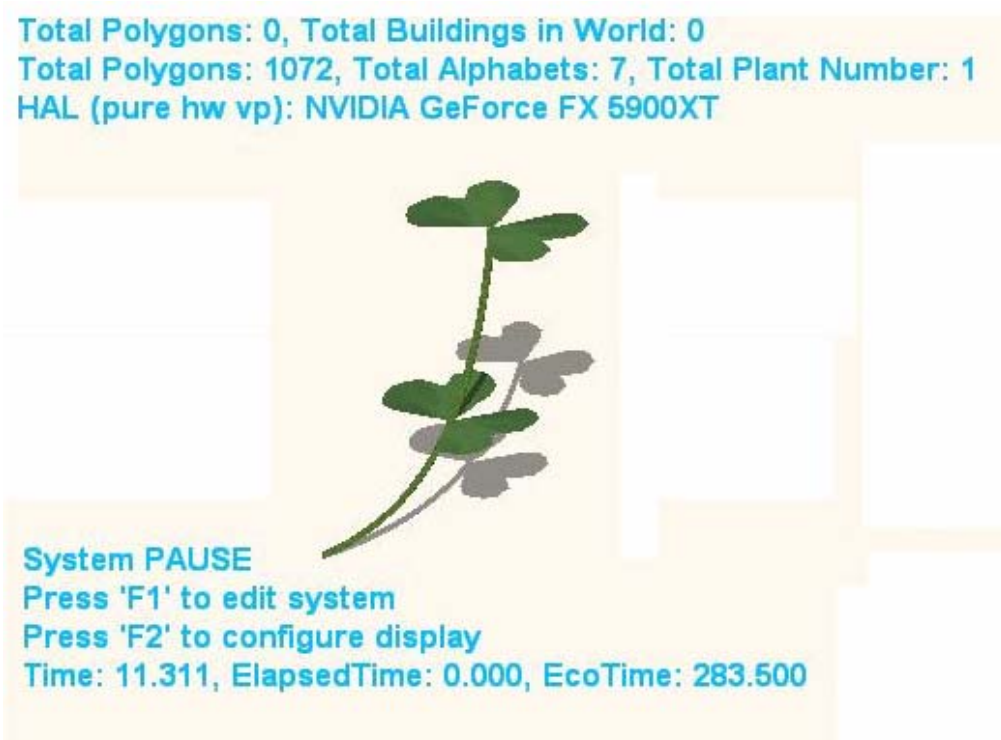


圖 21 植物背地性的表現

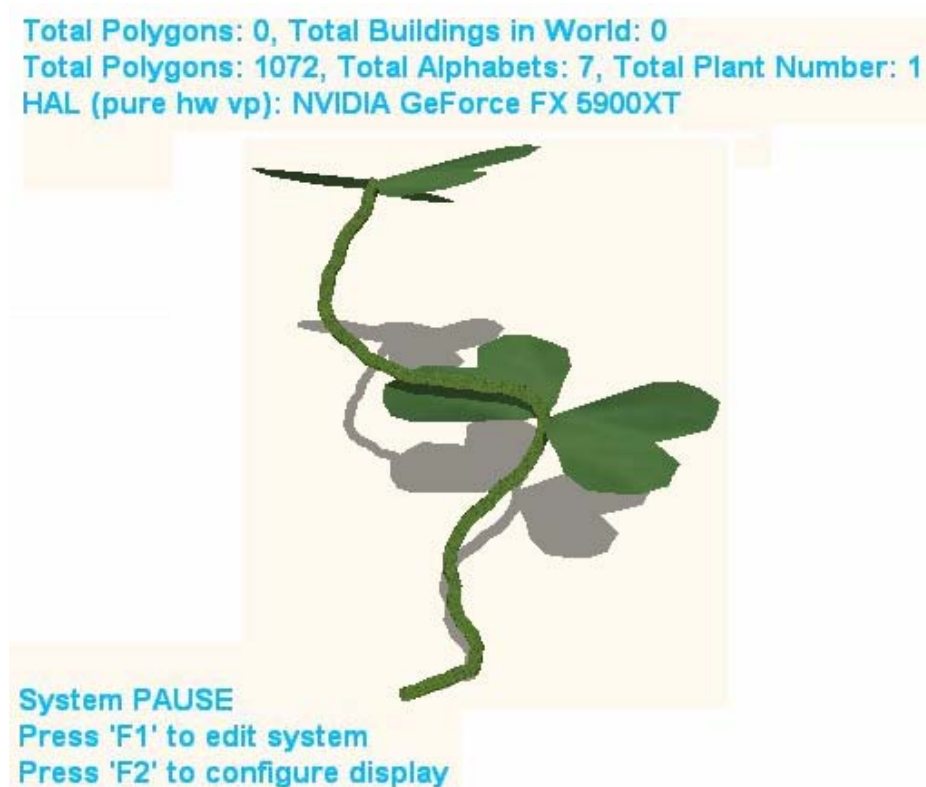


圖 22 有 Random Factor 的影響植物背地性

圖 22 是有著 random factor 的影響下，植物背地性對植物生長的影響，很明顯可以感覺到成長方向還是往上的受背地性所主導。

3.3.2 植物器官的形變

由於植物模型是 rendering engine 依據植物 skeleton 做貼合所得到的模型，所以在 SimEco system 中，植物器官的形變是由植物的 skeleton 所決定。如下圖所示：

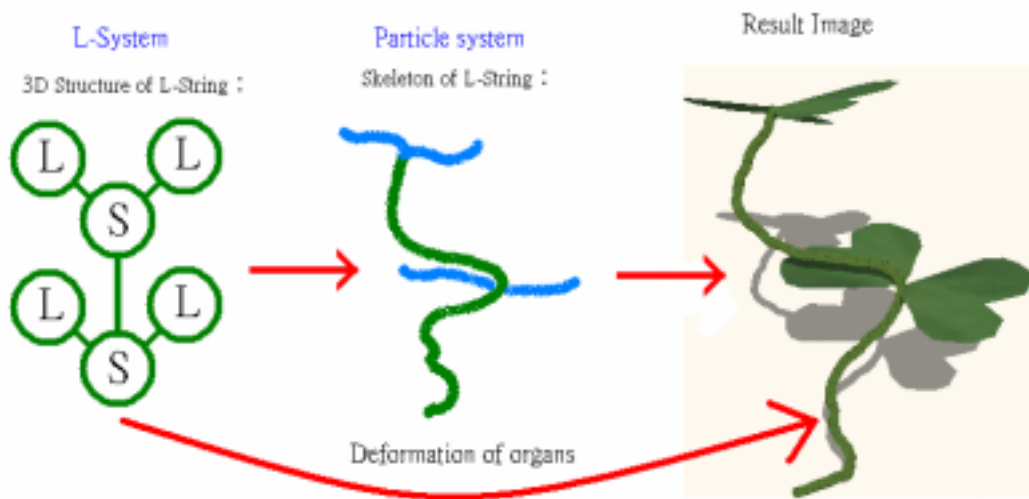


圖 23 Particle System 對系統形變所產生的影響

由上圖可以知道，植物的 skeleton 對 SimEco system 而言是暫存資訊，主要是提供 rendering engine 做植物模型貼合的重要依據，使用這種手法有以下數種好處：

- (1) 植物的 skeleton 是由 particle 在 3D 空間中受到環境的引力影響所產生的，換句話說，環境對植物器官的影響表現在器官形變上。
- (2) 可以使用簡單的幾何模型去貼合植物器官模型，可以大大降低模型取得的複雜，讓 3D model database 中的可以存有簡單幾何模型。

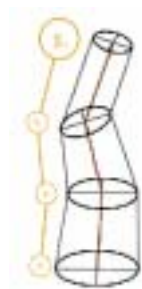


圖 24 Rendering Engine 使用簡單模型貼合植物器官

3.3.3 植物的生長曲線

在以往 L-system 在模擬植物生長時，都把植物生長視為直線生長，其實經由觀察不難發現，植物在生長之初，生長速率會較快，慢慢的在長成之後，植物生長速率會漸緩，二者的生長曲線如下圖所示：

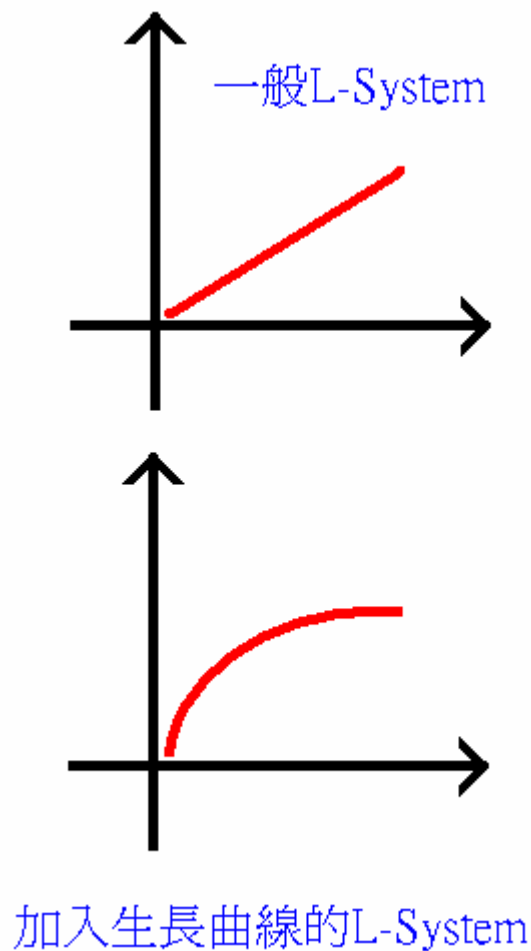


圖 25 植物生長曲線示意圖

加入了植物生長曲線後，我們可以明顯的感受到生長的植物更貼近真實生活中看到的植物一般，圖 26 and 圖 27 分別各是有加入生長曲線和沒有加入生長曲線，由系統所模擬出來的植物影像，上圖沒有生長曲線的玫瑰，我們可以明顯的感覺到每個莖是以線性做生長，而有加入生長曲線下圖的玫瑰，在頂端的部分明顯可以感受到莖的生長漸緩，也使的所產生的植物影像更像真實生活中隨處可見的植物影像。



圖 26 沒有加入植物生長曲線的玫瑰

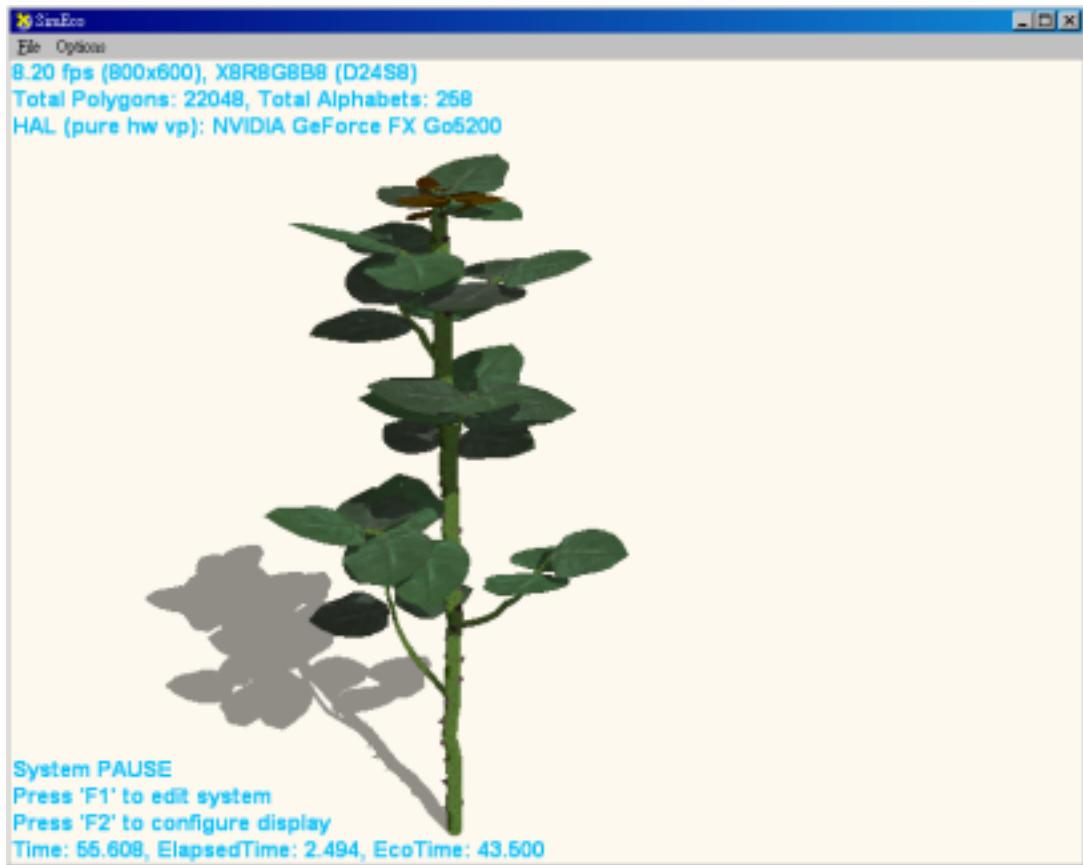


圖 27 有加入植物生長曲線的玫瑰

3.3.4 植物的向光性

SimEco system 在採用 particle system 之後，植物對向光性的反應變的十分的容易呈現，在 SimEco system 的模擬中，向光性對植物的影響就只變成是一個對 particle 產生影響的外力，而向光性的大小則是由當時 particle 所處的位置產生大小的變化和影響，得到的 particle 軌跡就可以反應向光性對植物的影響，下圖就是植物對植物對向光性的反應和影響。

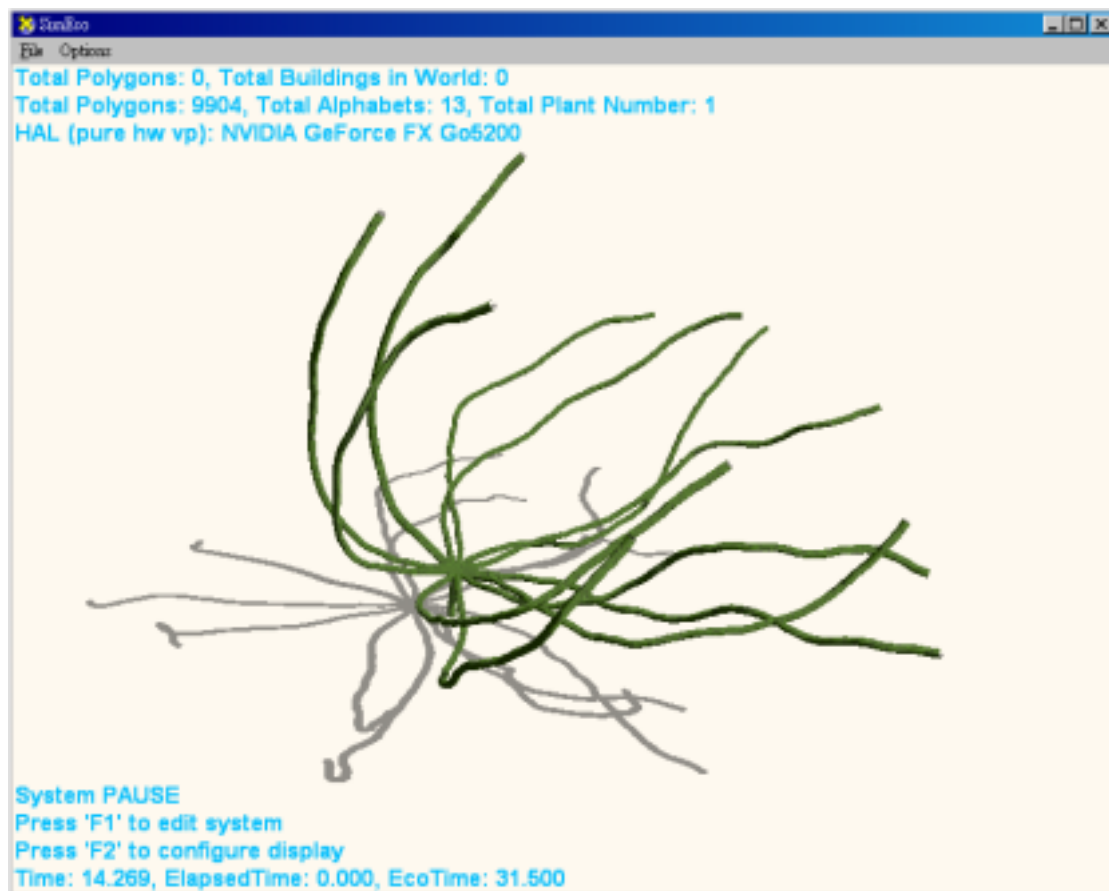


圖 28 植物向光性的呈現

上圖的植物主要受三個部分的影響，分別來自 random factor、向光性以及背地性。可以明顯感受到簡單的植物對三個外力的影響，並把三個影響反應在植物器官的形變上，為了幫助使用者了解和明顯觀測到莖的形變，系統把植物的葉子拿掉，方便使用者觀測植物器官的形變。

3.3.5 植物的碰撞策略

加入 particle system 後，植物的碰撞問題就簡化成為單一 particle 對障礙物時所應該採取的碰撞策略，我們採取以下的公式做系統的碰撞策略：

$$\overrightarrow{NewVel} = (C_i - i)^{f(i)} * \overrightarrow{V} + i * \overrightarrow{R}$$

\overrightarrow{NewVel} : New Velocity of particles after taking in count the collision

C_i : The collision segment number of particle

i : Now segment number of particle

$f(i)$: The attraction of collision object

\overrightarrow{V} : Original velocity of particle

\overrightarrow{R} : Directional vector of surface

圖 29 碰撞策略公式

採取上述公式有數個優點：

- (1) 可以透過 $f(i)$ 來調整障礙物對植物的影響，提供更佳的操控性給使用者。
- (2) 利用簡單的向量運算，大大降低碰撞計算時的計算量。

圖 30 主要是希望讓使用者觀測到單一器官利用生長策略，如何去把碰撞策略反應在植物器官的形變上，在其中我們還可以發現，當植物器官感覺到不會再有碰撞的影響時，就又会再改變生長方向，往所欲成長的方向再做生長，圖 30 很明顯可以感覺到上描述的二件事情：

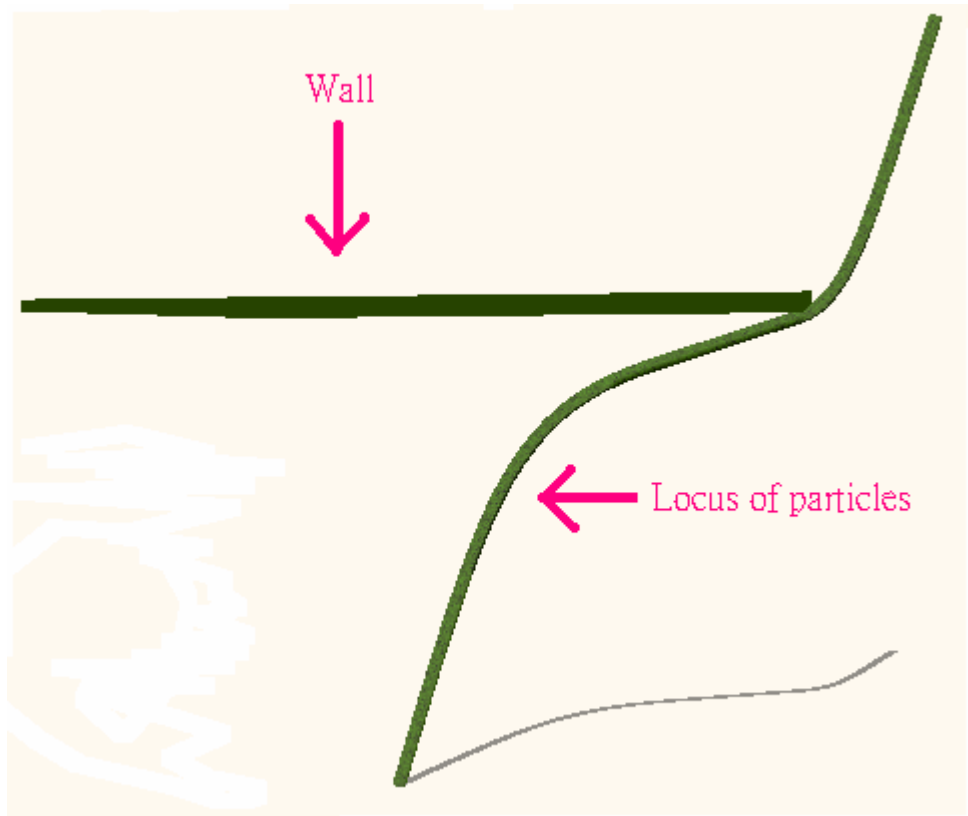


圖 30 單一植物器官的碰撞策略

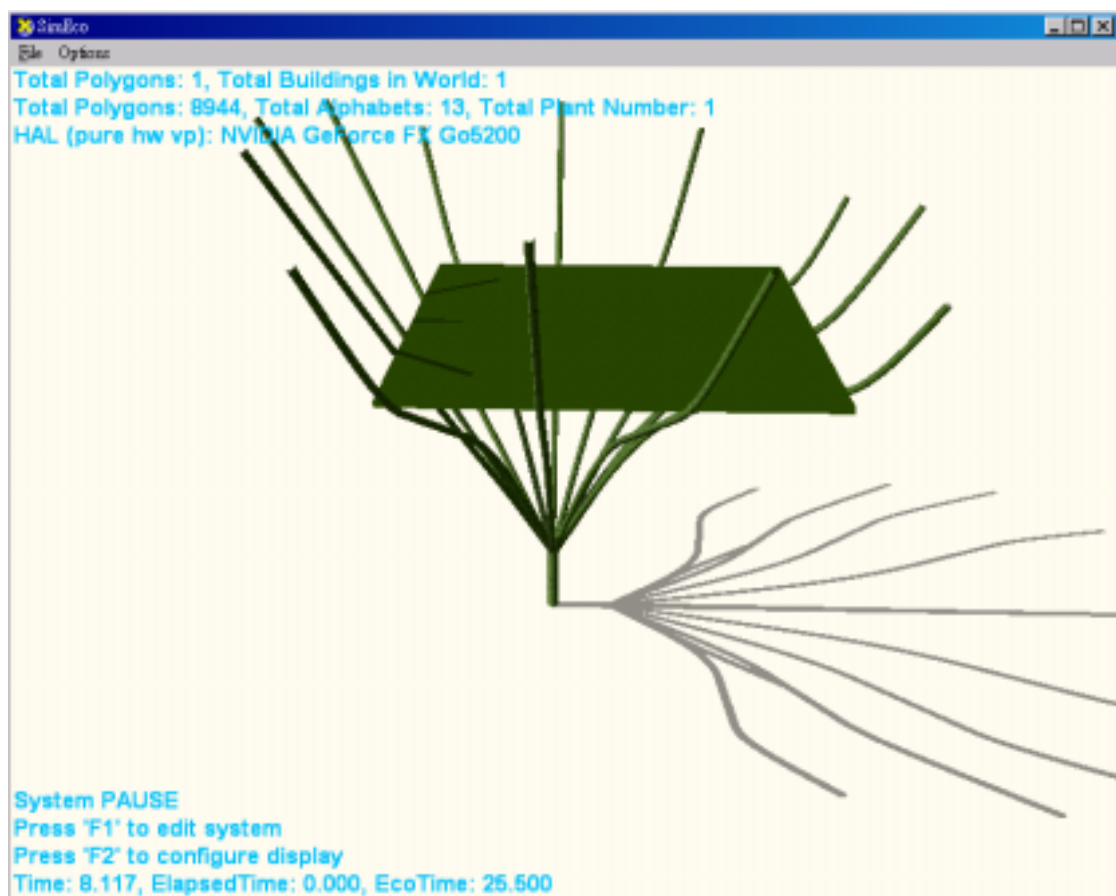


圖 31 植物多器官對障礙物的碰撞策略

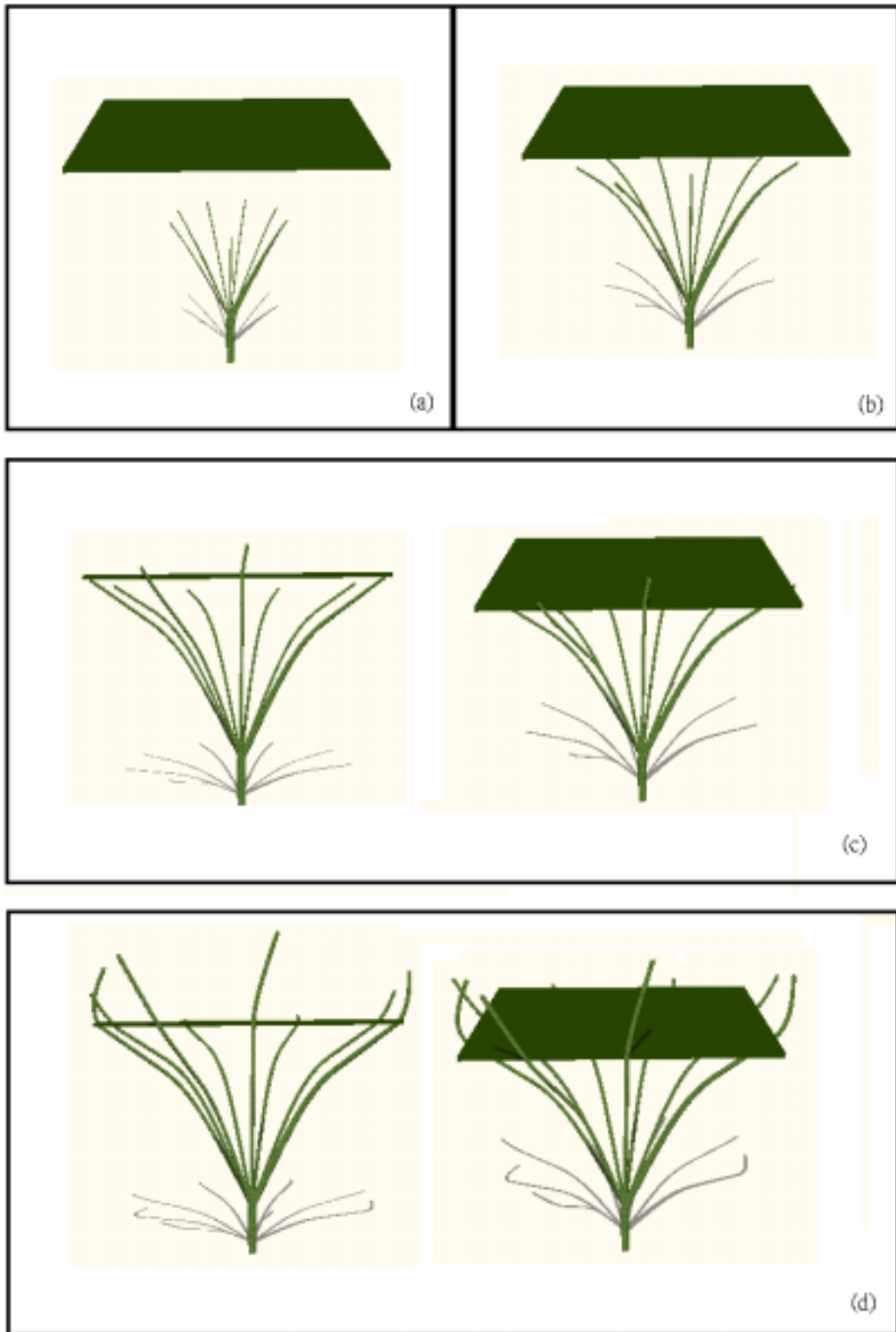


圖 32 連續植物碰撞策略示意圖

3.4 SimEco System 的優點

SimEco system 採用 two layer structure , 包含 L-system layer + particle system layer , 所能夠得到的好處包含以下四項 :

1. 透過使用 L-system , 可以使產生的植物保有植物學上的基本正確性。
2. 在讀入 L-script 後 , 由於記錄下每個由 L-system 所產生的 L-string , 方便系統可以產生連續的植物生長影像 , 以提供給使用者做觀察。
3. 可以把環境對植物的影響 , 透過植物器官的形變 , 反應出環境的影響力給使用者知道。
4. 在撰寫 SimEco L-script , 由於有 particle system 的輔助 , 所以在撰寫植物文法時 , 相對來說學習負擔度就大幅的降低。

第四章 Result and Conclusion

4.1 玫瑰(Rosa rugosa)

科名：薔薇科 Rosaceae，薔薇屬

俗名：刺枚花，徘徊花。

葉序：互生

特徵：雙子葉植物，羽狀複葉，葉脈明顯。莖上有刺，枝幹較粗硬，花朵大，每一花序只有少數幾朵(花小而多者為薔薇)。

分布：歐洲和中國大陸的遼寧、河北、山東等省份，以及北非、墨西哥、印度等皆有分布。

用途：庭園露地栽培，花壇，盆栽，切花，香料，藥用，賦香。

系統影響：Random factor，向光性以及背地性。

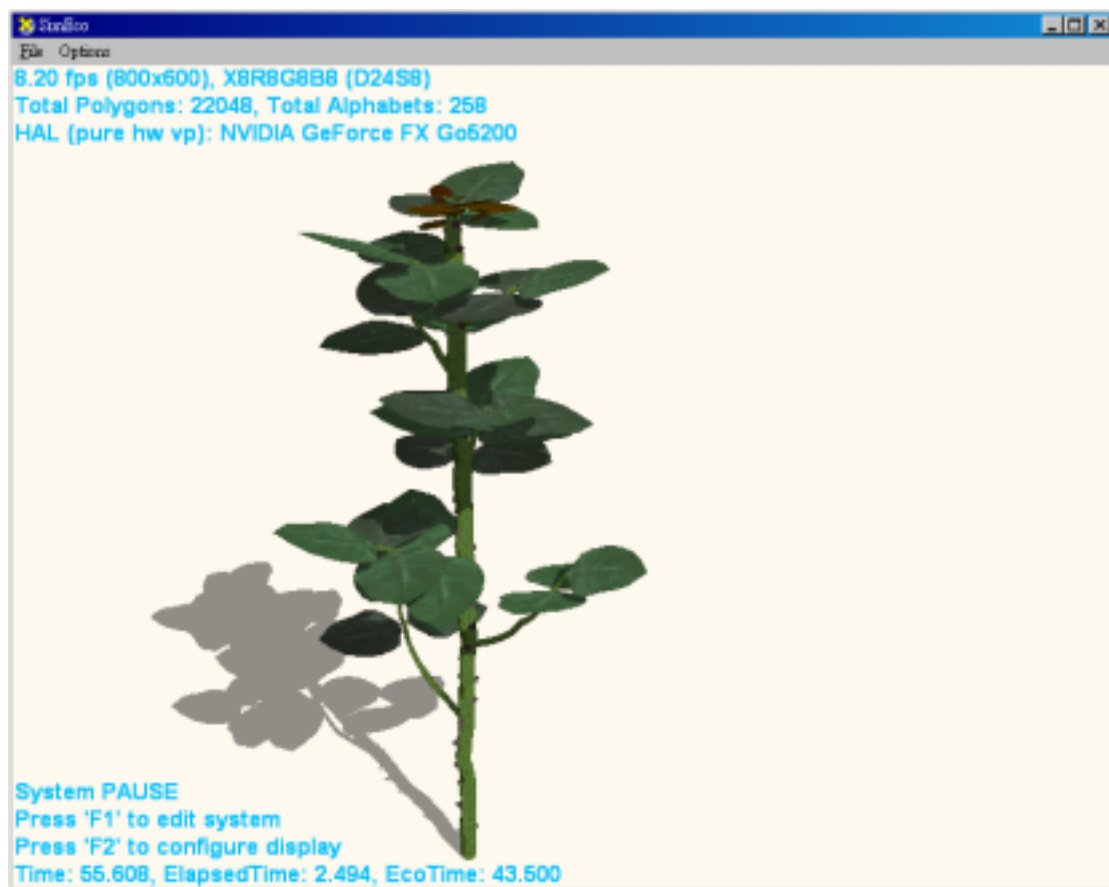


圖 33 系統模擬玫瑰生長

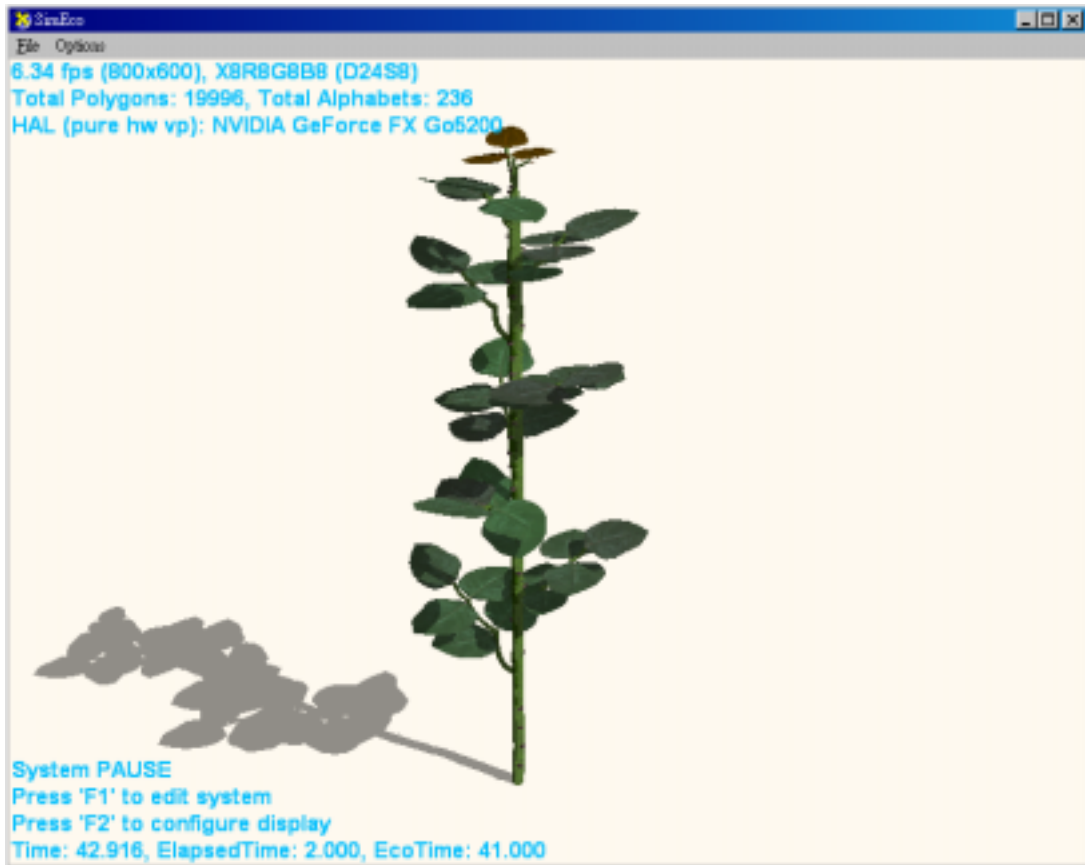


圖 34 玫瑰各視野圖 view(1)



圖 35 玫瑰各視野圖 view(2)

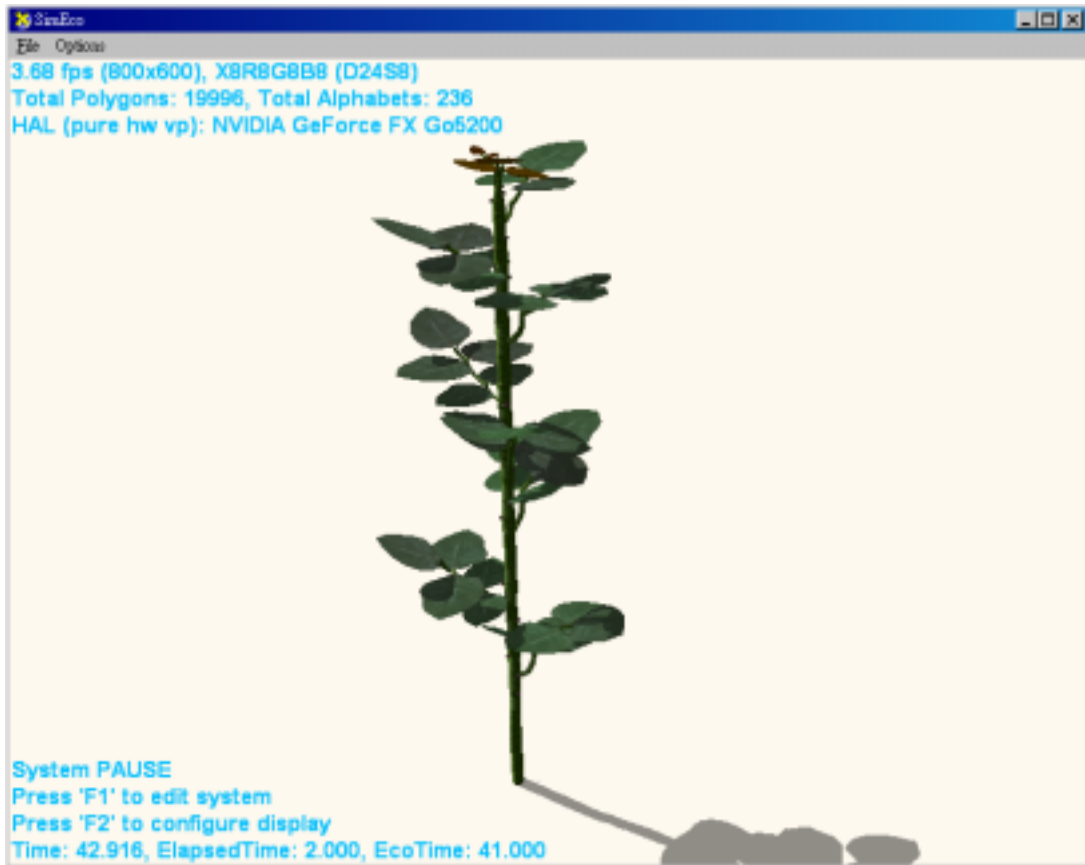


圖 36 玫瑰各視野圖 view(3)



圖 37 玫瑰各視野圖 view(4)

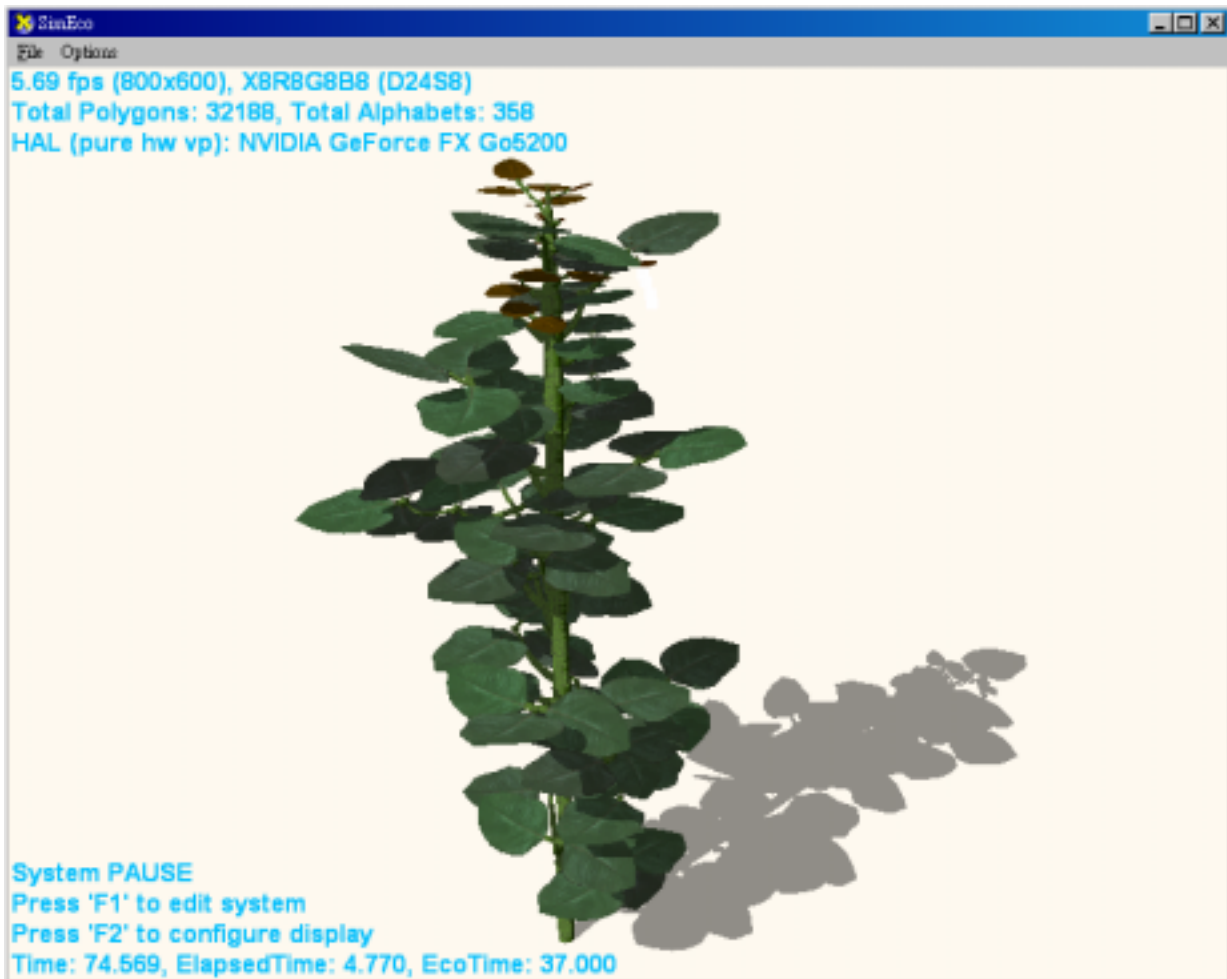


圖 38 複雜結構的玫瑰

4.2 馬櫻丹 (*Lantana camara*)

別 名：五色梅,五龍蘭

原產地：西印度

葉序：十字對生帶托葉

花序：頭狀花序

備註：馬櫻丹原在荷蘭人佔領台灣期間引進。有一種特殊的嗆鼻臭味，
而茜草科植物都有著似類的葉序。

系統影響：Random factor，向光性以及背地性。



圖 39 系統模擬馬櫻丹(茜草科)的成長



圖 40 馬櫻丹各視野圖 view(1)

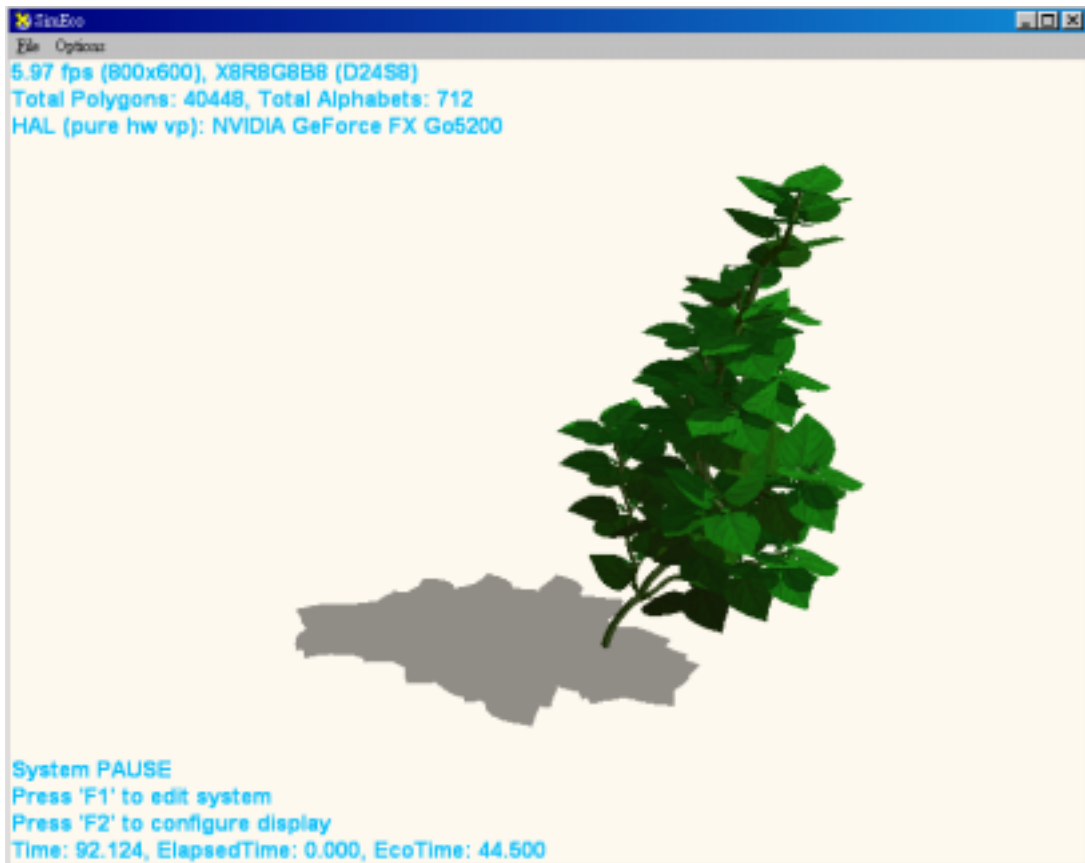


圖 41 馬櫻丹各視野圖 view(2)

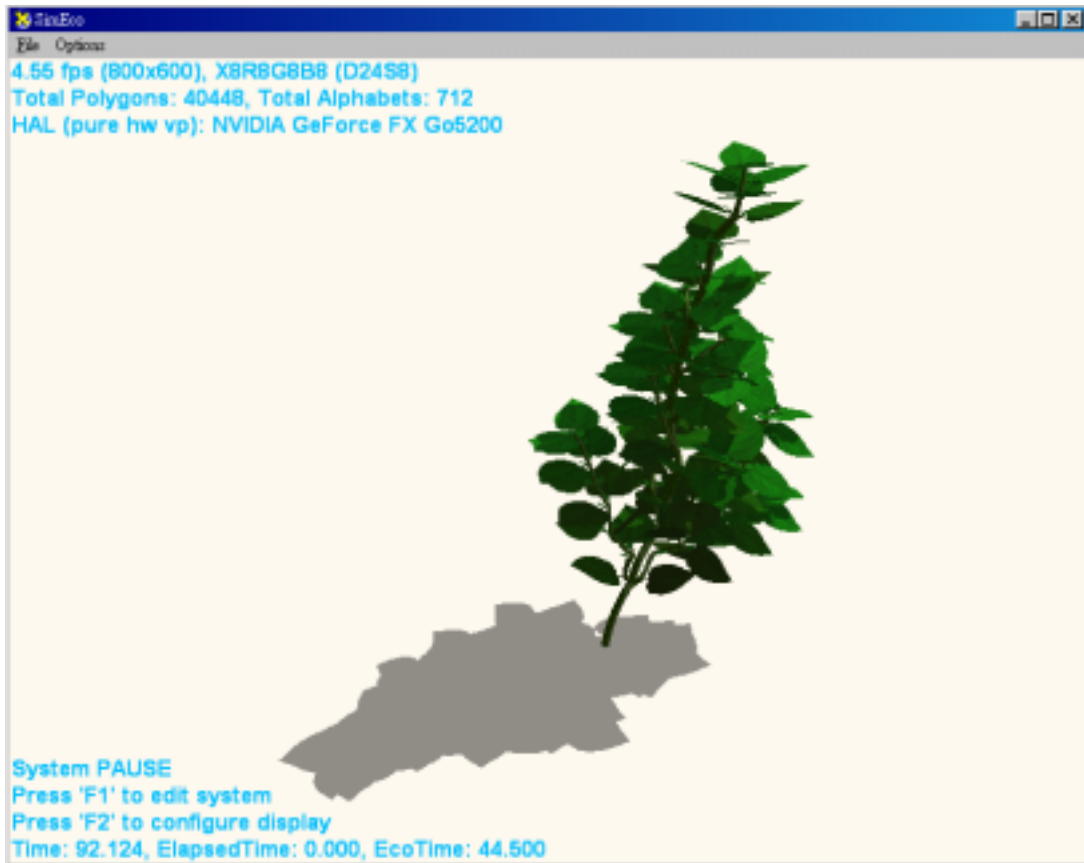


圖 42 馬櫻丹各視野圖 view(3)

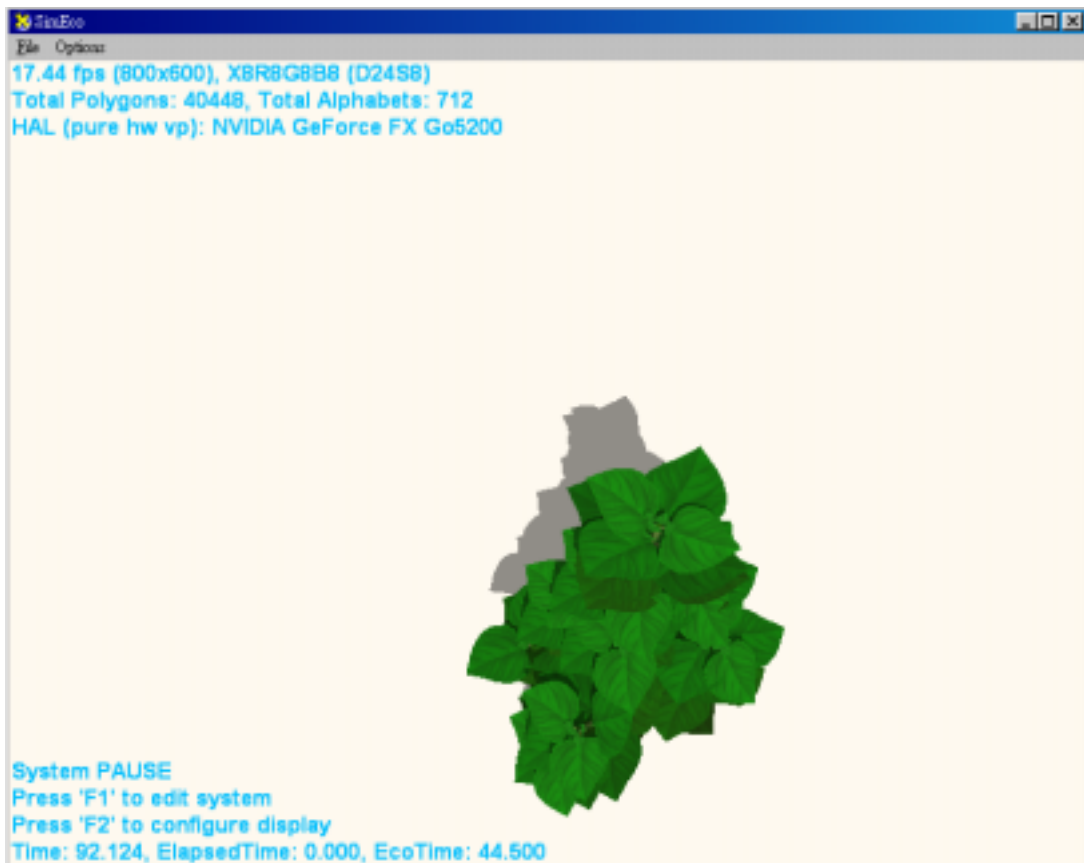


圖 43 馬櫻丹各視野圖 view(4)

第五章 Future Work

SimEco system 採用 two layer structure 的架構，主要包含三個部分：L-system layer、particle system layer and rendering engine。系統在模擬植物生長時，已經得到不錯的結果，但是在系統做模擬時還是有相當多的問題值得好好再去做研究以及探討，以當做我們努力研究的方向：

1. 系統在加入 particle system 後，已經能把向光性、重力以及背地性的影響反應在植物器官的形變上，而在各項引力大小對植物器官的影響上，則一直未有相當具體的量化數據可供參考，而各種植物對於三種引力的反應大小也各不相同，如何能取得所需要的量化數據一直是在撰寫 L-system script 時，很大的問題。
2. particle system layer 在做 collision 的部分還可以再加強，所採用的演算法也並非是最佳的演算法，在處理 collision 時也只局限於單面的障礙物上。
3. Rendering engine 的部分，在植物各器官表面的處理也應該放棄單純只用 texture 處理的手法，採用更真實的展現，如下圖所示，希望能夠表現更多植物在現實生活中所展現的現象反應在系統植物的模擬上。



圖 44 馬櫻丹葉子所展現的透光性

4. 在 particle system 和 L-system 之間應該建立起溝通的管道，應該改變 particle system and L-system 現在單純只有 L-system 提供 L-string 給 particle system 的架構，需再經由 particle system 把植物和環境影響的互動結果，當做 feedback 提供給 L-grammar and L-system，做為修正植物生長文法的重要規則，架構如下：

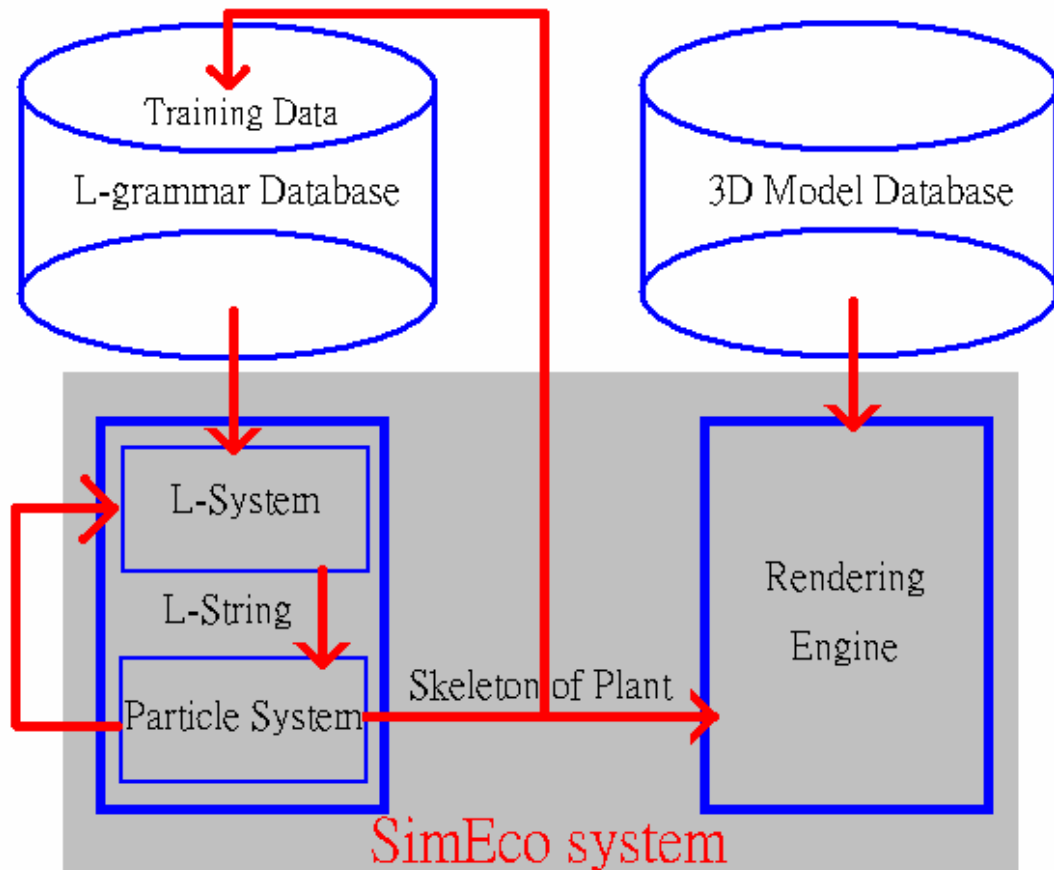


圖 45 提供給 Database 進行 Training Data 的架構

主要提供的 feedback 做修正有二個部分，

- (1) 把環境的影響傳回給 L-system，修正 non-deterministic rulers 之間選取機率的調整，又或是在 growth grammars 中 random factor range 的調整重要參考依據。
- (2) 由 grammar interpreter 所得到的 L-grammar database 中，得到的選取機率由於無法由文字或影像取得，所以一開始會假設成 normal

distribution，但經由不斷的模擬成長過程，所得到的 training data 做訓練後，希望可以得到較為正確的 L-grammar database。

5. 在空間競爭方式，也是植物模擬的一個重要的課題，SimEco system 在加入 particle system layer，也提供植物在做空間競爭時重要的依據，以 particle 在空間中分佈的密度為考量或許也是十分重要的研究方向。

開發一個生態模擬系統是相當龐大而複雜的事，要正確地模擬更是困難，我們試著讓自己朝向正確的方向，希望在將來能夠創造出一個有用的生態模擬系統，使之在教育、研究與自然保育上有所供獻。

第六章 Reference

1. Przemyslaw Prusinkiewicz, Aristid Lindenmayer, James Hanan. Development models of herbaceous plants for computer imagery purposes. ACM SIGGRAPH Computer Graphics, Proceedings of the 15th annual conference on Computer graphics and interactive techniques June 1988 Volume 22 Issue 4, page(s): 141-150
2. Radomír Měch, Przemyslaw Prusinkiewicz. Visual models of plants interacting with their environment. Proceedings of the 23rd annual conference on Computer graphics and interactive techniques August 1996, page(s): 397 - 410
3. Lindenmayer, G. Rozenberg. Developmental systems and languages. Proceedings of the fourth annual ACM symposium on Theory of computing May 1972, page(s): 214 - 221
4. The use of positional information in the modeling of plants. Proceedings of the 28th annual conference on Computer graphics and interactive techniques August 2001. Page(s): 289 – 300
5. William T. Reeves, Ricki Blau. Approximate and probabilistic algorithms for shading and rendering structured particle systems. ACM SIGGRAPH Computer Graphics, Proceedings of the 12th annual conference on Computer graphics and interactive techniques July 1985 Volume 19 Issue 3 Appendix, page(s): 313 – 322
6. Jason Weber, Joseph Penn. Creation and rendering of realistic trees. Proceedings of the 22nd annual conference on Computer graphics and interactive techniques September 1995, page(s): 119 – 128
7. Shlyakhter, I.; Rozenoer, M.; Dorsey, J.; Teller, S. Reconstructing 3D tree models from instrumented photographs. IEEE Computer Graphics and Applications , Volume: 21 Issue: 3 , May/Jun 2001, page(s): 53 -61
8. Przemyslaw Prusinkiewicz. Simulation modeling of plants and plant ecosystems. Communications of the ACM July 2000, Volume 43 Issue 7, page(s): 84 - 93
9. Kokai, G.; Toth, Z.; Vanyi, R. Evolving artificial trees described by parametric L-systems. Electrical and Computer Engineering, 1999 IEEE Canadian Conference on, Volume: 3 , 1999, page(s): 1722 -1727 vol.3
10. Stefanovski, S.; Loskovska, S.; Mihajlov, D. Representation and realistic rendering of objects defined by L-systems. Mediterranean Electrotechnical Conference, 1998. MELECON 98, 9th, Volume: 1, 1998, page(s): 86 -90 vol.1
11. Nagahashi, H.; Niwa, A.; Agui, T.; Competition and mutualism in a simulation of adaptive artificial organisms. Evolutionary Computation, 1995., IEEE International Conference on , Volume: 2 , 29 Nov-1 Dec 1995 , Page(s): 695 -700

vol.2

12. Thimbleby, H.W.; Witten, I.H.; Pullinger, D.J.; Concepts of cooperation in artificial life. *Systems, Man and Cybernetics, IEEE Transactions on* , Volume: 25 Issue: 7 , Jul 1995 , Page(s): 1166 -1171(No)
13. Banzhaf, W.; Artificial selection in a system of self-replicating strings. *Evolutionary Computation, 1994. IEEE World Congress on Computational Intelligence., Proceedings of the First IEEE Conference on* , 27-29 Jun 1994, Page(s): 651 -655 vol.2(No)
14. Shiyong Dai; Hu Cheng; A general procedure of self-propagation. *Evolutionary Computation, 1995., IEEE International Conference on* , Volume: 1 , 29 Nov-1 Dec 1996 , Page(s): 501(No)
15. Fulkerson, B.; Parunak, V.; The living factory: applications of artificial life to manufacturing. *Autonomous Decentralized Systems, 1995. Proceedings. ISADS 95., Second International Symposium on* , 25-27 Apr 1995 , Page(s): 391 -397(No)
16. Przemyslaw Prusinkiewicz, Mark Hammel, Radomír Měch; The Artificial Life of Plants. *Artificial life for graphics, animation, and virtual reality, volume 7 of SIGGRAPH '95 Course Notes, pages 1-1-1-38. ACM, Press, 1995.*(No)
17. Lieu-Hen Chen, Ling Lin Shing, Tzung Jr Tsai; An Adaptive L-System. *CVGIP 2004.*
18. U hao Chan, Lin Lin Shing, and Lieu-Hen Chen; Adaptive L-System. *CG Workshop 2003.*
19. Animation of plant development. Przemyslaw Prusinkiewicz , Mark S. Hammel , Eric Mjolsness. *Proceedings of the 20th annual conference on Computer graphics and interactive techniques September 1993, page(s): 351 – 360*
20. Integration of optimization by genetic algorithms into an L-system-based animation system. Noser, H.; Stucki, P.; Walser, H.-P. *Computer Animation, 2001. The Fourteenth Conference on Computer Animation. Proceedings* , 2001, page(s): 106 -253
21. Modeling and rendering of realistic feathers. Yanyun Chen , Yingqing Xu , Baining Guo , Heung-Yeung Shum. *ACM Transactions on Graphics (TOG) , Proceedings of the 29th annual conference on Computer graphics and interactive techniques July 2002, Volume 21 Issue 3, page(s): 630 – 636*
22. Joanna L. Power , A. J. Bernheim Brush , Przemyslaw , David H. Salesin. *Interactive arrangement of botanical L-system models. SIGGRAPH 1999, 175 – 182.*
23. Durikovic, R., Czanner, S. An efficient technique for ray tracing of a D0L-system. *Computer Graphics, Spring Conference on, 2001. 165 – 172.*

24. Alvy Ray Smith. Plants, fractals, and formal languages. ACM. SIGGRAPH. 1984, 1 – 10.
25. William T. Reeves , Ricki Blau. Approximate and probabilistic algorithms for shading and rendering structured particle systems. ACM SIGGRAPH Computer Graphics , Proceedings of the 12th annual conference on Computer graphics and interactive techniques July 1985, Volume 19 Issue 3 , Pages: 313 – 322.
26. Jeffrey J. McConnell. Three dimensional tree grammars for the modeling of plants. ACM Press New York, NY, USA , 1988, Pages: 494 – 499.
27. Jason Weber , Joseph Penn. Creation and rendering of realistic trees. Proceedings of the 22nd annual conference on Computer graphics and interactive techniques September 1995, pages: 119 – 128.
28. Shlyakhter, I.; Rozenoer, M.; Dorsey, J.; Teller, S. Reconstructing 3D tree models from instrumented photographs. IEEE Computer Graphics and Applications , Volume: 21 Issue: 3 , May/Jun 2001 Page(s): 53 -61.
29. Kokai, G.; Toth, Z.; Vanyi, R. Evolving artificial trees described by parametric L-systems. Electrical and Computer Engineering, 1999 IEEE Canadian Conference on , Volume: 3 , 1999 Page(s): 1722 -1727 vol.3.
30. Stefanovski, S.; Loskovska, S.; Mihajlov, D. Representation and realistic rendering of objects defined by L-systems. Mediterranean Electrotechnical Conference, 1998. MELECON 98., 9th , Volume: 1 , 1998 Page(s): 86 -90 vol.1.

第七章 Appendix

7.1 Appendix A : SimEco L-Script BNF 文法定義

Ecosystem	=	"Ecosystem" VersionNumber Name '{' {SysProperty ';' } {Plant} '}'
VersionNumber	=	RealNumber
Name	=	"" { String } ""
SysProperty	=	SoilPH DayLight RainRate Gravity GravityEffect GravityEnable LightPosition LightEnable LightEffect
Plant	=	"Plant" Name '{' {PlantProperty ';' } {Organ} Grow '}'
Wall	=	"Wall" '{' {WallProperty ';' }
WallProperty		WallPosition WallDirection WallLWH WallDraw Texture
WallPosition	=	"WallPosition" '=' Vector
WallDirection	=	"WallDirection" '=' Vector
WallLWH	=	"WallLWH" '=' Vector
WallDraw	=	"WallDraw" '=' NaturalNumber
RealNumber	=	Digit, {Digit}, '.', Digit, {Digit}
String	=	Character, {Character Digit}
SoilPH	=	"SoilPH" '=' Number
DayLight	=	"DayLight" '=' Number
RainRate	=	"RainRate" '=' Number
Gravity	=	"Gravity" '=' Vector
GravityEffect	=	"GravityEffect" '=' RealNumber
GravityEnable	=	"GravityEnable" '=' NaturalNumber
LightPosition	=	"LightPosition" '=' Vector
LightEnable	=	"LightEnable" '=' NaturalNumber
LightEffect	=	"LightEffect" '=' RealNumber
PlantProperty	=	Position Direction DeriveTime
Organ	=	OrganName '{' {Stage} '}'
Grow	=	"Grow" '{' Axiom {ProductionRule ';' } '}'
Digit	=	'0' '1' '2' '3' '4' '5' '6' '7' '8' '9'
Charater	=	'A' 'B' 'C' 'D' 'E' 'F' 'G' 'H' 'I' 'J' 'K' 'L' 'M' 'N'

		'O' 'P' 'Q' 'R' 'S' 'T' 'U' 'V' 'W' 'X' 'Y' 'Z' 'a' 'b' 'c' 'd' 'e' 'f' 'g' 'h' 'i' 'j' 'k' 'l' 'm' 'n' 'o' 'p' 'q' 'r' 's' 't' 'u' 'v' 'w' 'x' 'y' 'z'
Number	=	RealNumber RandomNumber NaturalNumber
Vector	=	"vector" '(' RealNumber ',' RealNumber ',' RealNumber ')'
Position	=	"Positoin" '=' Vector
Direction	=	"Direction" '=' Vector
DeriveTime	=	"DeriveTime" '=' Number
OrganName	=	"Apex" "Stem" "Leaf" "Root" "Flower" "Fruit"
Stage	=	(("Stage" NaturalNumber) "Default") '{' {StageProperty ';'} '}'
Axiom	=	'w' '=' Sentence ';'
ProductionRule	=	(Condition "->" Production) (StepCondition "->" StepProduction) (Condition "->" "null" ';')
RandomNumber	=	"rand" '(' RealNumber ',' RealNumber ')'
NaturalNumber	=	Digit, {Digit}
StageProperty	=	Texture GrowRate LifeTime BordenRate WeightFactor Color ConeNoise NoiseTime Model Material Strength Geotropism
Sentence	=	{Rotation} Segment {Segment} [Branch {Branch} [[{SelectTerm}] Sentence]]
Condition	=	[LeftSensitive] Predecessor [RightSensitive]
Production	=	[SelectTerm]Successor { ' ' [SelectTerm]Successor }
Stepcondition	=	[LeftSensitive] StepPredecessor [RightSensitive]
StepProduction	=	[SelectTerm]StepSuccessor { ' ' [SelectTerm]Successor }
Texture	=	"Texture" '=' Name
GrowRate	=	"GrowRate" '=' Number
LifeTime	=	"LifeTime" '=' Number
BroadenRate	=	"BroadenRate" '=' Number
WeightFactor	=	"WeightFactor" '=' Number
ConeNoise	=	"ConeNoise" '=' Number
NoiseTime	=	"NoiseTime" '=' Number
Color	=	"Color" '=' RGBA
Model	=	"Model" '=' Name
Material	=	"Material" '=' MTRL

Strength	=	"Strength" '=' Number
Geotropism	=	"Geotropism" '=' Number
Rotation	=	("rotx" "roty" "rotz" "rotr" "rotu") (' Number ')
Segment	=	SegmentType '(' SegmentStage ')'
Section	=	{SelectTerm} {Rotatetion} Segment {Segment}
Branch	=	'[' Section ']'
LeftSensitive	=	{Segment} '<'
Predecessor	=	Segment
RightSensitive	=	'>' {Segment}
SelectTerm	=	ProbTerm EnvironmentTerm
Successor	=	Sentence
StepPredecessor	=	SegementType '(' Pronoun ')'
StepSuccessor	=	SegementType '(' Pronoun '+' NaturalNumber ', Pronoun '<' NaturalNumber ')'
RGBA	=	"RGBA" '(' Number ', Number ', Number ', Number ')'
MTRL	=	"MTRL" '(' Number ', Number ', Number ', Number ', Number ', Number ')'
ProbTerm	=	"prob" '(' RealNumber ')'
EnvironmentTerm	=	--< Oops >--
SegmentType	=	'A' 'S' 'L' 'F' 'T' 'R'
Pronoun	=	'i'
SegmentStage	=	NaturalNumber
Comment	=	"//" {String}

7.2 Appendix B : 馬櫻丹的 SimEco L-Script

```
EcoSystem 2.0 "MySys" {  
    SoilPH = 7.0;  
    RainRate = 0.9;  
    DayLight = 12.0;  
    Gravity = vector(0.0, 1.0, 0.0);  
    GravityEffect = 1.02;  
    GravityEnable = 1;  
    LightPosition = vector(0.0, 0.0, -1.0);  
    LightEffect = 1.03;  
    LightEnable = 1;  
    Plant "Lantan camara" {  
        Position = vector(0.0, 0.0, 0.0);  
        Direction = vector(0.0, 0.1, 0.0);  
        DeriveTime = 3.0;  
        Apex {}  
        Stem {  
            Stage 0 {  
                Model = "DefaultStem.i";  
                Texture = "seafloor.bmp";  
            }  
            Stage 1 {  
                Model = "DefaultStem.i";  
                Texture = "seafloor.bmp";  
            }  
        }  
    }  
}
```

```

Stage 2 {
    Model = "DefaultStem.i";
    Texture = "seafloor.bmp";
    GrowRate = 0.00375;
    GrowTime = 3.0;
    BroadenRate = 0.01;
    Color = RGBA(0.0, 1.0, 0.0, 1.0);
    Material = MTRL(0.5, 1.0, 0.0, 0.1, 0.1, 0.1);
}

Stage 3 {
    Model = "DefaultStem.i";
    Texture = "seafloor.bmp";
    GrowRate = 0.1875;
    GrowTime = 3.0;
    BroadenRate = 0.1;
    ConeNoise = rand(0.00, 0.01);
    NoiseTime = 0.5;
    Color = RGBA(0.0, 1.0, 0.0, 1.0);
    Material = MTRL(0.5, 1.0, 0.0, 0.1, 0.1, 0.1);
}

Stage 4 {
    Model = "DefaultStem.i";
    Texture = "seafloor.bmp";
    Material = MTRL(0.5, 1.0, 0.0, 0.1, 0.1, 0.1);
}

Stage 6 {

```

```

    Model = "DefaultStem.i";
    Texture = "seafloor.bmp";
}
Stage 7 {
    Model = "DefaultStem.i";
    Texture = "seafloor.bmp";
}
Stage 8 {
    Model = "DefaultStem.i";
    Texture = "seafloor.bmp";
    GrowRate = 0.00375;
    GrowTime = 3.0;
    BroadenRate = 0.01;
    ConeNoise = rand(0.00, 0.01);
    NoiseTime = 0.5;
    Color = RGBA(0.0, 1.0, 0.0, 1.0);
    Material = MTRL(0.5, 1.0, 0.0, 0.1, 0.1, 0.1);
}
Stage 9 {
    Model = "DefaultStem.i";
    Texture = "seafloor.bmp";
    GrowRate = 0.1875;
    GrowTime = 3.0;
    BroadenRate = 0.1;
    ConeNoise = rand(0.00, 0.02);
    NoiseTime = 0.5;

```

```

        Color = RGBA(0.0, 1.0, 0.0, 1.0);
        Material = MTRL(0.5, 1.0, 0.0, 0.1, 0.1, 0.1);
    }
    Stage 10 {
        Model = "DefaultStem.i";
        Texture = "seafloor.bmp";
        GrowRate = 0.01875;
        GrowTime = 5.5;
        BroadenRate = 0.015;
        Color = RGBA(0.0, 1.0, 0.0, 1.0);
        Material = MTRL(0.5, 1.0, 0.0, 0.1, 0.1, 0.1);
    }
    Stage 11 {
        Model = "DefaultStem.i";
        Texture = "seafloor.bmp";
        GrowRate = 0.01875;
        GrowTime = 10.5;
        BroadenRate = 0.015;
        Color = RGBA(0.0, 1.0, 0.0, 1.0);
        Material = MTRL(0.5, 1.0, 0.0, 0.1, 0.1, 0.1);
    }
    Stage 99 {
        Model = "DefaultStem.i";
        Texture = "seafloor.bmp";
    }
    Default {

```

```

    Model = "DefaultStem.i";
    Texture = "seafloor.bmp";
    GrowRate = 0.1875;
    GrowTime = 3.0;
    BroadenRate = 0.1;
    ConeNoise = rand(0.00, 0.007);
    NoiseTime = 0.5;
    Color = RGBA(0.0, 1.0, 0.0, 1.0);
    Material = MTRL(0.5, 1.0, 0.0, 0.1, 0.1, 0.1);
}
}
Leaf {
    Stage 0 {
        Model = "馬櫻丹.i";
        Texture = "馬櫻丹.bmp";
        GrowRate = 0.005;
        GrowTime = 3.0;
        Color = RGBA(0.6, 1.0, 0.6, 1.0);
        Material = MTRL(1.8, 1.8, 1.3, 0.3, 0.3, 0.2);
    }
    Stage 1 {
        Model = "馬櫻丹.i";
        Texture = "馬櫻丹.bmp";
        GrowRate = 0.005;
        GrowTime = 3.0;
        Color = RGBA(0.6, 1.0, 0.6, 1.0);
    }
}

```

```

        Material = MTRL(1.8, 1.8, 1.3, 0.3, 0.3, 0.2);
    }
    Stage 2 {
        Model = "馬櫻丹.i";
        Texture = "馬櫻丹.bmp";
        GrowRate = 0.025;
        GrowTime = 3.0;
        Color = RGBA(0.6, 1.0, 0.6, 1.0);
        Material = MTRL(1.5, 1.8, 1.0, 0.3, 0.3, 0.2);
    }
    Stage 3 {
        Model = "馬櫻丹.i";
        Texture = "馬櫻丹.bmp";
        Color = RGBA(1.0, 1.0, 1.0, 1.0);
        Material = MTRL(1.2, 1.6, 0.8, 0.1, 0.1, 0.1);
    }
    Stage 4 {
        Model = "馬櫻丹.i";
        Texture = "馬櫻丹.bmp";
        Color = RGBA(1.0, 1.0, 1.0, 1.0);
        Material = MTRL(rand(1.0, 1.5), 1.3, 0.6, 0.3, 0.3, 0.2);
    }
    Stage 5 {
        Model = "馬櫻丹.i";
        Texture = "馬櫻丹.bmp";
        Color = RGBA(1.0, 1.0, 1.0, 1.0);

```



```

        Material = MTRL(rand(1.5, 2.0), 1.3, 0.6, 0.3, 0.3, 0.2);
    }

    Stage 6 {
        Model = "馬櫻丹.i";
        Texture = "馬櫻丹.bmp";
        Color = RGBA(1.0, 1.0, 1.0, 1.0);
        Material = MTRL(rand(1.5, 2.0), rand(0.7, 1.3), 0.6, 0.3,
            0.3, 0.2);
    }

    Stage 7 {
        Model = "馬櫻丹.i";
        Texture = "馬櫻丹.bmp";
    }

    Default {
        Model = "馬櫻丹.i";
        Texture = "馬櫻丹.bmp";
        Color = RGBA(1.6, 1.0, 1.6, 1.0);
        Material = MTRL(rand(2.0, 4.0), rand(0.8, 0.8), 0.6, 0.3,
            0.3, 0.2);
    }
}

Grow {
    w = A(0);
    A(0) -> S(0);
    S(0) -> S(2)[rotr(0.7854)rotu(0.3491)S(10)L(0)]
        [rotr(1.2472)rotu(0.3491)S(11)L(1)]
}

```

$[rotr(0.7854)rotu(3.8397)S(10)L(0)]$
 $[rotr(1.2472)rotu(3.8397)S(11)L(1)]$
 $S(1);$
 $S(1) \rightarrow prob(0.98)$
 $S(2)[rotr(0.7854)rotu(1.9199)S(10)L(0)]$
 $[rotr(1.2472)rotu(1.9199)S(11)L(1)]$
 $[rotr(0.7854)rotu(5.0615)S(10)L(0)]$
 $[rotr(1.2472)rotu(5.0615)S(11)L(1)]$
 $S(0) | S(99);$
 $S(2) \rightarrow S(3);$
 $L(1) \rightarrow L(2);$
 $L(6) \rightarrow prob(0.8)L(7) | L(8);$
 $L(i) \rightarrow L(i + 1, 1 < i < 5);$
 $L(8) \rightarrow E();$

 $S(3) > S(3) \rightarrow S(4);$
 $S(4) < S(10) \rightarrow S(6);$
 $S(4) \rightarrow S(5);$

 $S(6) \rightarrow S(8)[rotr(0.7854)S(10)L(0)]$
 $[rotr(1.2472)S(11)L(1)]$
 $[rotr(0.7854)rotu(3.8397)S(10)L(0)]$
 $[rotr(1.2472)rotu(3.8397)S(11)L(1)]$
 $S(7);$
 $S(7) \rightarrow prob(0.9)$
 $S(8)[rotr(0.7854)rotu(1.9199)S(10)L(0)]$

[rotr(1.2472)rotu(1.9199)S(11)L(1)]

[rotr(0.7854)rotu(5.0615)S(10)L(0)]

[rotr(1.2472)rotu(5.0615)S(11)L(1)]

S(6) | S(99);

S(8) -> S(9);

}

}

}

7.3 Appendix C : 玫瑰的 SimEco L-Script

```
EcoSystem 2.0 "MySys" {  
    SoilPH = 7.0;  
    RainRate = 0.9;  
    DayLight = 12.0;  
    Gravity = vector(0.0, 1.0, 0.0);  
    GravityEffect = 1.02;  
    GravityEnable = 1;  
    LightPosition = vector(0.0, 0.0, -1.0);  
    LightEffect = 1.01;  
    LightEnable = 1;  
    Plant "Rose Rugosa" {  
        Position = vector(0.0, 0.0, 0.0);  
        Direction = vector(0.0, 0.1, 0.0);  
        DeriveTime = 3.0;  
        Apex {}  
        Stem {  
            Stage 0 {  
                Model = "DefaultStem.i";  
                Texture = "seafloor.bmp";  
                GrowRate = 0.012;  
                GrowTime = 3.0;  
                BroadenRate = 0.03;  
                Color = RGBA(0.0, 1.0, 0.0, 1.0);  
                Material = MTRL(0.5, 1.0, 0.0, 0.1, 0.1, 0.1);  
            }  
        }  
    }  
}
```

```

    Strength = 0.9;
    Geotropism = -0.5;
}
Stage 1 {
    Model = "DefaultStem.i";
    Texture = "seafloor.bmp";
    GrowRate = 0.012;
    GrowTime = 3.0;
    BroadenRate = 0.03;
    Color = RGBA(0.0, 1.0, 0.0, 1.0);
    Material = MTRL(0.5, 1.0, 0.0, 0.1, 0.1, 0.1);
    Strength = 0.9;
    Geotropism = -0.5;
}
Stage 2 {
    Model = "DefaultStem.i";
    Texture = "seafloor.bmp";
    GrowRate = 0.012;
    GrowTime = 3.0;
    BroadenRate = 0.03;
    Color = RGBA(0.0, 1.0, 0.0, 1.0);
    Material = MTRL(0.5, 1.0, 0.0, 0.1, 0.1, 0.1);
    Strength = 0.9;
    Geotropism = -0.5;
}
Stage 3 {

```

```

    Model = "DefaultStem.i";
    Texture = "seafloor.bmp";
    GrowRate = 0.012;
    GrowTime = 3.0;
    BroadenRate = 0.03;
    Color = RGBA(0.0, 1.0, 0.0, 1.0);
    Material = MTRL(0.5, 1.0, 0.0, 0.1, 0.1, 0.1);
    Strength = 0.9;
    Geotropism = -0.5;
}
Stage 4 {
    Model = "DefaultStem.i";
    Texture = "seafloor.bmp";
    GrowRate = 0.012;
    GrowTime = 3.0;
    BroadenRate = 0.03;
    Color = RGBA(0.0, 1.0, 0.0, 1.0);
    Material = MTRL(0.5, 1.0, 0.0, 0.1, 0.1, 0.1);
    Strength = 0.9;
    Geotropism = -0.5;
}
Stage 5 {
    Model = "DefaultStem.i";
    Texture = "seafloor.bmp";
    GrowRate = 0.012;
    GrowTime = 3.0;

```

```

    BroadenRate = 0.03;
    Color = RGBA(0.0, 1.0, 0.0, 1.0);
    Material = MTRL(0.5, 1.0, 0.0, 0.1, 0.1, 0.1);
    Strength = 0.9;
    Geotropism = -0.5;
}
Stage 6 {
    Model = "DefaultStem.i";
    Texture = "seafloor.bmp";
    GrowRate = 0.015;
    GrowTime = 3.0;
    BroadenRate = 0.03;
    Color = RGBA(0.0, 1.0, 0.0, 1.0);
    Material = MTRL(0.5, 1.0, 0.0, 0.1, 0.1, 0.1);
    Strength = 0.9;
    Geotropism = -0.5;
}
Stage 7 {
    Model = "DefaultStem.i";
    Texture = "seafloor.bmp";
    GrowRate = 0.00375;
    GrowTime = 3.0;
    BroadenRate = 0.03;
    Color = RGBA(0.0, 1.0, 0.0, 1.0);
    Material = MTRL(0.5, 1.0, 0.0, 0.1, 0.1, 0.1);
    ConeNoise = rand(0.00, 0.008);
}

```

```

    NoiseTime = 1.0;

    Strength = 0.9;

    Geotropism = -0.5;
}

Stage 10 {
    Model = "DefaultStem.i";

    Texture = "seafloor.bmp";

    GrowRate = 0.015;

    GrowTime = 3.0;

    BroadenRate = 0.03;

    Color = RGBA(0.0, 1.0, 0.0, 1.0);

    Material = MTRL(0.5, 1.0, 0.0, 0.1, 0.1, 0.1);

    ConeNoise = rand(0.00, 0.006);

    NoiseTime = 0.5;

    Strength = 0.9;

    Geotropism = -0.5;

"Gravity" '=' Vector      }

Stage 11 {
    Model = "DefaultStem.i";

    Texture = "seafloor.bmp";

    GrowRate = 0.06;

    GrowTime = 3.0;

    BroadenRate = 0.03;

    Color = RGBA(0.0, 1.0, 0.0, 1.0);

    Material = MTRL(0.5, 1.0, 0.0, 0.1, 0.1, 0.1);

    ConeNoise = rand(0.00, 0.006);

```



```

    NoiseTime = 0.5;
    Strength = 0.9;
    Geotropism = -0.5;
}
Stage 12 {
    Model = "DefaultStem.i";
    Texture = "seafloor.bmp";
    GrowRate = 0.09;
    GrowTime = 5.0;
    BroadenRate = 0.03;
    Color = RGBA(0.0, 1.0, 0.0, 1.0);
    Material = MTRL(0.5, 1.0, 0.0, 0.1, 0.1, 0.1);
    ConeNoise = rand(0.00, 0.006);
    NoiseTime = 0.5;
    Strength = 0.9;
    Geotropism = -0.5;
}
Stage 13 {
    Model = "DefaultStem.i";
    Texture = "seafloor.bmp";
    GrowRate = 0.12;
    GrowTime = 3.0;
    BroadenRate = 0.03;
    Color = RGBA(0.0, 1.0, 0.0, 1.0);
    Material = MTRL(0.5, 1.0, 0.0, 0.1, 0.1, 0.1);
    ConeNoise = rand(0.00, 0.006);

```

```

    NoiseTime = 0.5;
    Strength = 0.9;
    Geotropism = -0.5;
}
Stage 14 {
    Model = "DefaultStem.i";
    Texture = "seafloor.bmp";
    GrowRate = 0.09;
    GrowTime = 5.0;
    BroadenRate = 0.03;
    Color = RGBA(0.0, 1.0, 0.0, 1.0);
    Material = MTRL(0.5, 1.0, 0.0, 0.1, 0.1, 0.1);
    ConeNoise = rand(0.00, 0.006);
    NoiseTime = 0.5;
    Strength = 0.9;
    Geotropism = -0.5;
}
Stage 15 {
    Model = "DefaultStem.i";
    Texture = "seafloor.bmp";
    GrowRate = 0.03;
    GrowTime = 3.0;
    BroadenRate = 0.045;
    Color = RGBA(0.0, 1.0, 0.0, 1.0);
    Material = MTRL(0.5, 1.0, 0.0, 0.1, 0.1, 0.1);
    ConeNoise = rand(0.00, 0.007);
}

```

```

    NoiseTime = 0.5;

    Strength = 0.9;

    Geotropism = -0.5;
}

Stage 16 {
    Model = "DefaultStem.i";

    Texture = "seafloor.bmp";

    GrowRate = 0.06;

    GrowTime = 6.0;

    BroadenRate = 0.0225;

    Color = RGBA(0.0, 1.0, 0.0, 1.0);

    Material = MTRL(0.5, 1.0, 0.0, 0.1, 0.1, 0.1);

    Strength = 0.9;

    Geotropism = -0.5;
}

Stage 17 {
    Model = "DefaultStem.i";

    Texture = "seafloor.bmp";

    GrowRate = 0.02;

    GrowTime = 3.0;

    BroadenRate = 0.0625;

    Color = RGBA(0.0, 1.0, 0.0, 1.0);

    Material = MTRL(0.5, 1.0, 0.0, 0.1, 0.1, 0.1);

    Strength = 0.9;

    Geotropism = -0.5;
}

```

```

Stage 18 {
    Model = "DefaultStem.i";
    Texture = "seafloor.bmp";
}

Stage 20 {
    Model = "DefaultStem.i";
    Texture = "seafloor.bmp";
    GrowRate = 0.00;
    GrowTime = 9.0;
    BroadenRate = 0.06;
    Color = RGBA(0.0, 1.0, 0.0, 1.0);
    Material = MTRL(0.9, 0.7, 0.9, 0.1, 0.1, 0.1);
    Strength = 0.9;
    Geotropism = -0.5;
}

Stage 21 {
    Model = "DefaultStem.i";
    Texture = "seafloor.bmp";
}

Stage 22 {
    Model = "DefaultStem.i";
    Texture = "seafloor.bmp";
    GrowRate = rand(0.02, 0.045);
    GrowTime = 3.0;
    BroadenRate = 0.00;
    Color = RGBA(0.0, 1.0, 0.0, 1.0);
}

```

```

    Material = MTRL(0.9, 0.7, 0.9, 0.1, 0.1, 0.1);
    Strength = 0.9;
    Geotropism = -0.5;
}
Default {
    Model = "DefaultStem.i";
    Texture = "seafloor.bmp";
    GrowRate = 0.05625;
    GrowTime = 3.0;
    BroadenRate = 0.12;
    ConeNoise = rand(0.00, 0.008);
    NoiseTime = 1.0;
    Color = RGBA(0.0, 1.0, 0.0, 1.0);
    Material = MTRL(0.5, 1.0, 0.0, 0.1, 0.1, 0.1);
}
}
Leaf {
    Stage 0 {
        Model = "玫瑰.i";
        Texture = "玫瑰.bmp";
        GrowRate = 0.008;
        GrowTime = 3.0;
        Color = RGBA(0.6, 1.0, 0.6, 1.0);
        Material = MTRL(1.0, 0.5, 0.0, 0.3, 0.3, 0.2);
    }
    Stage 1 {

```

```

    Model = "玫瑰.i";
    Texture = "玫瑰.bmp";
    GrowRate = 0.016;
    GrowTime = 3.0;
    Color = RGBA(0.6, 1.0, 0.6, 1.0);
    Material = MTRL(1.2, 0.8, 0.0, 0.3, 0.3, 0.2);
}

Stage 2 {
    Model = "玫瑰.i";
    Texture = "玫瑰.bmp";
    GrowRate = 0.008;
    GrowTime = 3.0;
    Color = RGBA(1.0, 1.0, 1.0, 1.0);
    Material = MTRL(1.0, rand(1.1, 1.5), 1.0, 0.3, 0.3, 0.2);
}

Stage 3 {
    Model = "玫瑰.i";
    Texture = "玫瑰.bmp";
    Color = RGBA(1.0, 1.0, 1.0, 1.0);
    Material = MTRL(1.0, rand(1.1, 1.5), 1.0, 0.3, 0.3, 0.2);
}

Default {
    Model = "玫瑰.i";
    Texture = "玫瑰.bmp";
    Color = RGBA(1.0, 1.0, 1.0, 1.0);
    Material = MTRL(1.0, 1.0, 1.0, 0.3, 0.3, 0.2);
}

```

```

    }
}
Grow {
  w = A(0);
  A(0) -> S(5)S(5)S(0);
  S(0) -> S(6)S(6)S(6)S(6)[rotr(1.2472)A(10)]S(1);
  S(1) -> prob(0.98)S(6)S(6)S(6)S(6)
        [rotr(1.2472)rotu(2.4)A(10)]S(2) | S(21);
  S(2) -> prob(0.98)S(6)S(6)S(6)S(6)
        [rotr(1.2472)rotu(4.8)A(10)]S(3) | S(21);
  S(3) -> prob(0.98)S(6)S(6)S(6)S(6)
        [rotr(1.2472)rotu(7.2)A(10)]S(4) | S(21);
  S(4) -> prob(0.98)S(6)S(6)S(6)S(6)
        [rotr(1.2472)rotu(9.6)A(10)]S(0) | S(21);
  S(5) -> S(6);
  S(6) -> prob(0.8)S(7)[rotr(1.2472)
        rotu(rand(0.0, 6.2832))S(20)S(21)] | S(7);
  S(21) -> S(22);
  S(7) > S(7) -> S(8);
  S(8) > S(8) -> S(9);

  A(10) -> S(10)S(11)S(15)L(0);
  S(10) -> prob(0.5)S(11) | prob(0.5)S(12)
        | S(14)[rotr(1.5708)S(17)L(0)][rotr(1.5708)
        rotu(3.1416)S(17)L(0)];
  S(11) -> S(12)[rotr(1.5708)S(17)L(0)]

```

[rotr(1.5708)rotu(3.1416)S(17)L(0)];

S(12) < S(15) -> S(17);

S(8) < S(12) -> S(14);

S(12) > S(16) -> S(13);

S(11) < S(15) -> S(16);

S(14) < S(17) -> S(18);

S(18) < L(1) -> L(3);

S(16) < L(0) -> L(1);

S(17) < L(0) -> L(1);

L(1) -> L(2);

}

}

}