

國立暨南國際大學資訊工程學系

碩士論文

以影像為基礎的植物生長過程推演手法

**A Deduction Method for the Growing Process of  
Plants based on Images**

指導教授： 陳履恆 博士

研究生： 黃襄安

中華民國 九十五年 八月

# 致謝

很快的研究所兩年的日子過去了，感謝陳履恆老師在這研所兩年跟大學四年中，對我在課業及為人處事的指導與協助，有老師的指點與建議，我才能完成這邊研究論文。也要感謝這幾年來404實驗室的學長與同學們，感謝 bird 學長在我研究遇到瓶頸時給我建議幫我打氣，感謝常常被我麻煩但還是會幫我忙的沛宇學長，感謝小黃跟 winner 兩位學長對我的照顧與幫忙，感謝小黑在程式方面的協助，感謝Taco學長、翁爺、fishman、蛋蛋、阿富、Jimmy、大頭、叮嚕、阿桐、馬達、baca及很多人，要不是有大家的陪伴與協助，我研所的生活就不會這麼快樂與充實。

除了實驗室的同學跟老師之外，系上的許多師長及系辦也在我的求學過程中給予許多協助。感謝那些陪我打球的球友們，讓我有運動減壓的管道。當然也要感謝我的家人，可以容忍我這麼長的時間都待在埔里不回家，還常常打電話關心我的生活跟課業，提供資源讓我的生活不虞匱乏。

最後，感謝我在埔里認識的所有同學跟朋友，因為有大家，埔里才是一個這麼可愛的地方，謝謝大家。

黃襄安 誌於

暨南國際大學科技三館404室

二零零六年八月十三號

論文名稱：以影像為基礎的植物生長過程推演手法

校院系：暨南國際大學科技學院資訊工程系

頁數：41

畢業時間：九十五年七月

學位別：碩士

研究生：黃襄安

指導教授：陳履恆 博士

## 中文摘要

在目前的 CG 領域中產生擬真的植物影像是極為重要的議題。至於要如何產生擬真的植物影像則應該著重在 rendering 的技巧和植物形態特徵的真實呈現。我們所模擬出來的植物影像除了希望要有真實正確的光影效果之外，其形態與特徵在植物學上的正確性也是一個很重要的考量。在模擬植物的成長和建立植物的結構上 L-System 是最為人所熟知的方法，L-System 可以藉由使用自我疊代的生長規則來模擬植物生長過程。但是要產生符合真實植物的生長文法卻因為要將大量的資訊參數化而是件複雜及耗時的工作，為了方便一般使用者，我們開發了文法產生器來協助使用者根據文字及影像資訊來產生生長文法。有了生長文法後，再利用我們開發的一個叫做 SimEco 的 proto-type 系統用來產生擬真的 CG 植物影像。除了要求最終影像的正確性之外，在本研究中我們也會根據一些植物形態學的規則跟限制，來模擬從幼苗開始生長的生長過程，以產生一連串的植物生長的動態影像。另外本研究也將一套物理模型加入 SimEco 系統中，諸如向光性、風力等外在因素對植物形態所造成的永久與暫時性的影響。因此可以模擬植物跟環境互動的結果，像是受風力吹拂後枝葉擺動的動畫效果。

關鍵詞：L-system，植物生成，物理動畫

Title of Thesis : A Deduction Method for the Growing Process of Plants based on Images

Name of Institute : Department of Computer Science and Information Engineering,

National Chi Nan University

Pages : 41

Graduation Time : 2006/07

Degree Conferred : Master

Student Name : Hsiang-An Huang

Advisor Name : Dr. Lieu-Hen Chen

## Abstract

Above all, it is a very consequential issue to produce the photorealistic plant image in CG field. When the issue comes to how to make the image real, the rendering technique and the way of showing the shape and trait of plants takes a great part of it. As we simulate the plant image, we cannot stay away from the botanic accuracy, in addition to the accurate rendering effect. L-system is the most well-known technique that used to simulating the process of plant growing. As we know, L-system could simulate the process of plant growing by using the self-iterative production rules. Building the production rule that exactly like the real plant is a complicate and time-consuming job, because we need to convert lots of information into parameters. For the sake of common users' convenience, we create the L-grammar Generator to help the users create the L-grammar based on the plant description and plant photo. After the L-grammar coming out, then, we use a proto-type system, so-called "SimEco" to create plant image. Beside the result image accuracy, we also simulate the process of growing, from the very beginning, the bud, due to the bounds and rules in morphological botany. Furthermore, our research adds a physical model to simulate the influence of heliotropism, windage and etc. Then we can simulate the interaction between the plant and the environment.

Keywords: L-system, generate plants, physical animation

# 內容目錄

致謝 .....	I
中文摘要 .....	II
Abstract.....	III
內容目錄 .....	IV
圖目錄 .....	VI
表目錄 .....	VII
第一章 緒論 .....	1
1.1 研究背景 .....	1
1.2 研究動機與目的 .....	2
1.3 研究方法 .....	3
1.4 論文架構 .....	4
第二章 文獻探討和相關研究 .....	5
2.1 L-SYSTEM的研究 .....	5
2.2 物理模型之使用 .....	6
第三章 理論基礎 .....	7
3.1 L-SYSTEM理論基礎 .....	7
3.2 植物運動物理模型的理論基礎 .....	9
第四章 系統設計 .....	13
4.1 系統架構 .....	13

4.2 系統流程和操作流程 .....	14
第五章 系統實作 .....	18
5.1 所需要的檔案 .....	18
5.1.1 植物生長敘述檔 .....	18
5.1.2 三視點植物照片檔 .....	19
5.2 生長文法的生成 .....	20
5.2.1 文字資料的使用 .....	20
5.2.2 量化數據的使用 .....	21
5.2.3 綜合資料生成文法 .....	23
5.3 物理模型的實作 .....	24
5.3.1 植物向光性模擬 .....	25
5.3.2 風力效果模擬 .....	26
5.4 研究成果 .....	30
第六章 結論 .....	34
參考文獻 .....	36
附錄 玫瑰的生長規則 .....	38

# 圖目錄

圖 3.1	與上下文無關的文法，(a)為文法生產規則，(b)為文法前三次疊代之結果	8
圖 3.2	與上下文相關的文法，範例一	8
圖 3.3	與上下文相關的文法，範例二	9
圖 3.4	骨架結構示意圖	10
圖 3.5	雙單擺	12
圖 4.1	系統的基本架構	14
圖 4.2	parser操作介面	15
圖 4.3	SimEco CAD操作介面	16
圖 4.4	系統的操作流程	17
圖 5.1	拍攝角度示意圖	19
圖 5.2	玫瑰的照片，(a)是正視圖，(b)是右視圖，(c)是頂視圖	19
圖 5.3	植物結構模型	21
圖 5.4	莖的兩個向量	22
圖 5.7	main stem and branch stem	27
圖 5.8	倒過來的雙單擺	28
圖 5.9	多次計算雙單擺過程	28
圖 5.10	branch stem的模擬	29
圖 5.11	結果對照圖	31
圖 5.12	生長過程，(a)為原圖，(b)為限定光源位置於植物的右方	32
圖 5.13	生長結束，(a)為原圖，(b)為限定光源位置於植物的右方	33

# 表目錄

表 5.1 生長關鍵字的種類.....	20
---------------------	----



# 第一章 緒論

## 1.1 研究背景

縱觀目前不管是在遊戲製作、動畫電影或虛擬實境的運用上，凡是使用電腦圖學來製作擬真的影像上皆免不了要面對戶外大型的場景，而植物的出現更是這些場景中無可避免的因素，即便是室內的場景也可能會有植物影像的出現，因此植物的擬真成像可以說是電腦圖學中不可或缺的一環，至於目前電腦圖學植物影像生成的研究上，大致可分為兩種主要的領域，其一是植物影像的擬真度，另一是如何產生符合植物學形態的植物。

在早期的植物影像生成的研究中，由於硬體的限制，無法做太複雜的計算以模擬真實的植物，而用了許多投機的手法來產生”看起來像是真的”的植物影像。就算很多效果不是經過真實運算所得到的，只要讓使用者乍看之下覺得這株植物看起來很真實，就會被認為是一張好的植物影像，所以影像的真實性其實在植物影像的生成中是十分重要的。近幾年因為硬體方面有了大幅的進步，大量縮短了計算的時間，所以也越來越多研究將重點放在產生更擬真及更正確的植物影像。例如：植物的組織是非對稱的材質、葉子和植物表皮的半透明以及物體的半影區，那麼如何將這些特性表現出來呢，像這樣的問題開始在電腦圖學植物的研究慢慢突顯出來，所以近幾年的植物研究範疇幾乎離不開這個議題，即如何改善植物的rendering技術[1][2][16]。

一張真實的植物影像中，除了光影及材質看起來要真實之外，植物結構上的真實性也是很重要的，所謂結構上的真實，就是植物結構要符合現實中的植物形態。對於植物形態的探討上，主要的方法就是Lindenmayer在1972所提出的L-System[7]，藉由考慮符號定義(alphabet)經由文法生產規則(production rule)的不斷

疊代後以符合植物的基本架構，然而為了使植物的 3D 結構可以更具體的展現，所以此後的學者開始經由觀察植物的生長過程，以及探討植物器官彼此的相應關係來強化文法的深度和廣度，同時不斷的加強文法生產規則(production rule)的表現能力，藉此產生出更複雜且真實的單一植物影像[6][10][11][12]。但是在 L-System 強化的過程中有些學者漸漸的著眼於不單只是想要發展出只適用於植物本身的文法，還想將植物在環境中與環境互動的行為也同時加入到影像中，使得植物由原本單一植株的呈現到能展現真實植被的分佈情形[3][4]。

然而在以文法的方式來表達植物學的特性為主流的研究上，有部份的學者是以快速和符合人類對植物意向的表達來探索植物影像的呈現[5][9]，這些的研究內容皆以手動或半手動的方法，經由使用者本身對於植物的概念來製作植物的模型，而不是以文法的方式來表現，雖然這樣的研究欠缺客觀的數據作為依據，不過這樣的方式確實加速了植物模型的產生，並且所產生出來的植物影像也未背離人們常識中對於植物所應具備的特性，藉此在取得自然植物的模型和簡易的產生植物的影像中獲得了適度的平衡。

## 1.2 研究動機與目的

L-System 是藉由文法的疊代來改變符號定義以模擬植物的生長，但是之前的很多研究為了加強 L-System 的表現能力，而在 L-System 中加入了相當多的控制參數及語法定義，導致文法過度的複雜，複雜的文法不但可讀性低，在使用上及學習上都會造成相當的困擾。這樣的文法若要以人工來撰寫，必須要花費相當多的時間來不斷修正文法及參數，才能產生逼真且正確的植物結構及影像。況且對於一般使用者而言，他們必須先學習 formal language 的相關知識，才有能力去撰寫文法。要一般使用者能在短時間內寫出令人滿意的生長文法，可以說是相當的困難。

在這幾年內有關 L-System 跟植物生長模擬的研究論文中，我們看到有許多研究是針對改進 L-System 文法本身的研究成果。但是對於要如何產生 L-System 的文法，

並沒有一個好的解答。對於產生連續的生長模擬影像，甚至於進一步地去預測植物的生長方式，相關的研究更是少之又少。

我們的研究目的除了開發一些輔助工具來自動產生植物生長文法，並且整合影像中以及資料庫中的植物生長資訊，使得我們所模擬出的植物影像，不但具有植物學上的正確性，更可以動畫的方式來呈現植物的生長過程。這樣的一套系統，除了能讓使用者可以不需先經過複雜的撰寫文法訓練，就能透過此系統方便地、簡單地產生出逼真且正確的植物生長動畫。

除了要快速地產生正確的植物生長文法之外，我們也希望可以模擬植物與環境間的互動。在現實生活中，植物無時無刻都在與環境做互動，包括生長時會因為重力、照光時間與方向、風力大小與方向、水源位置等因素的影響，而造成植物生長形態會有所不同。除了上述因素會永久影響植物的結構與形態外，另外還有些因素會暫時改變植物的形態，譬如風力、人為外力等...。這些影響所造成的形變是暫時的，當外力消失後植物就會回復原來的形態。我們也希望能將這些效果加進我們的系統當中，使得我們所產生的動畫更加真實。

### 1.3 研究方法

依據上一節本研究所述的目的，因此本論文首要的目標就是簡化文法的撰寫過程並保留有用的功能，也就是要開發一套全新並符合 L-System 的文法格式，並且可以根據各種的植物生長資訊，包括文字的資訊跟影像的資訊，來產生包含植物生長過程的生長文法。為了降低文法的複雜度及加強可讀性，我們簡化了之前需要在文法中定義的屬性資料，也在文法中加入了植物階層的資訊，讓使用者可以直接從文法中得到更多訊息。另外，為了使文法的疊代可以用來表示植物的生長過程，我們也參考了植物學的相關書籍，找出了植物生長的規則與限制，然後將這些概念加入我們的生長文法中，如此一來文法疊代的過程就可以用來表示植物真實的生長過程。

在模擬受外力影響後，植物形態改變的部分。我們將套用一些物理上的概念，

來模擬結構受力後的改變。在開始計算外力影響之前，我們先將植物的 3D 模型轉換成骨架結構(skeleton structure)。其實如果要做到真實的模擬，應該使用有限元素法(finite element)來計算結構上各點受力後對整個結構的影響。不過由於此方法太過於複雜且計算量相當龐大，如果採用此方法將會大幅降低系統的效能。經過考慮後，我們決定捨棄有限元素法這個精確的方法，而改採用雙單擺系統(double pendulum)來模擬植物受風力吹拂後枝葉擺盪的效果。另外將彈簧(spring)跟阻尼(damper)的概念帶進我們的植物骨架結構，讓模擬結果更真實正確。

## 1.4 論文架構

本論文共分成五個章節，茲分述如下：

第一章為緒論，主要說明研究背景、研究動機與目的、研究方法和論文架構。

第二章為相關文獻探討，主要說明 L-System 和 L-grammar 生成的歷史，並說明最近關於 L-system 和植物擬真成像的相關研究。

第三章為我們系統中所使用到的理論基礎，主要說明 L-System 的理論基礎並舉出一些疊代的操作規則和操作方法的實例，以及介紹我們所使用的物理模型。

第四章為系統架構，說明我們的系統包括那些模組以及各模組的功能，並說明系統的主要操作過程以及和使用者界面的設計。

第五章是系統實作，用來闡述系統實際運作時的所需要資料的取得以及如何生成文法及產生生長動畫的方法，最後在說明骨架節構的使用和成果。

第六章為結論，說明目前我們系統中待解決的一些問題。

## 第二章 文獻探討和相關研究

### 2.1 L-System 的研究

早期植物影像的成像手法並沒有固定的方式，除了請模型師以直觀的方式手動建立植物模型後生成影像外，一般植物影像的呈現手法還是以碎形(fractals)[15]、粒子系統(particle system)[13]和正規化語言(formal language)為主。以碎形手法和粒子系統所產生的影像只是在某一個時間點產生近似於植物形態的影像，此影像不但缺乏植物學上的正確性，更無法確定每張影像看起來會符合我們認知中的植物影像，因此這類的方法只有少部份的研究學者在使用。而真正可以說是植物影像生成手法的主流應該是正規化語言了，而在這裡所指的正規化語言其實就是L-System，這個方法在前面的章節已經有大致的提到，它乃是由德國人Lindenmayer於 1972 年在 ACM symposium on Theory of computing 中所提出的，往後的L-System的相關研究就是架構在這個基本上再繼續延伸下去的。

Prusinkiewicz 等人以L-System為基礎並加入觀察植物實際生長時的一些特性，來增強文法的表達能力，如：控制生長點的位置、主幹分枝夾角的角度等...，並且為了增加植物結構的正確性跟影像說服力，更將實際植物的生長數據加以統計，如花或果實的生長比例或生長速率，當作L-system的參數[10][12]。

另外Deussen等人以植被作為研究主軸，探討如何在廣大的場景中經過空間競爭後植物生長的分布情形，並且由於在這樣的場景中如果同時置入大量的植物將使得系統整體的效能大幅下降，因此他們在系統中引進以點和線段為基礎的LOD(level of detail)來增加系統的效能[3][4]。

到目前為止所提的研究都是著重在植物外觀的表現和空間分佈上，那麼現在我們介紹另外一種討論植物酵素對植物生長之影響的研究，Renton等人提出以植物內

在激素為依據來模擬植物的生長，並且由於內在的激素會因為生長的環境和四季的變化而有所改變所以一組文法將可以有不同之影像[14]。

## 2.2 物理模型之使用

有很多研究有討論到如何定義樹狀結構的物理模型，在討論物理模型之前，如何定義樹狀結構在空間中的關係就有各種的定義方式，一般最直觀的想法就是用 skeleton node 來定義植物的支幹，所以每根 skeleton 可以定義出植物的 topology，skeleton 的長度以及分枝的夾角。有了結構的幾何定義後，接下來是討論如何使該結構產生運動，以及運動的依據。

Granieri 等人於 1995 年所發表的”Production and Playback of Human Figure Motion for Visual Simulation”中，就有提到如何以運動學模擬動作與關節角度改變之對應關係，並可從此延伸，來處理植物骨架結構的運動效果[18]。

像是 Hutchinson 等人在 1996 年於 Eurographics Workshop 所提出的”Adaptative Refinement for Mass-Spring Simulations”中有提到如何將質點與彈簧的概念用在樹狀結構上[19]。

# 第三章 理論基礎

## 3.1 L-System 理論基礎

在電腦圖學植物的研究上，L-System是首先被提出來以平行覆寫系統(parallel rewriting system)的方式，來模組化植物的形態和生長過程的一套正規化語言(Formal language)[15]，大體說來符合L-System定義的系統必需包含三個基本的架構：有限狀態集合(a finite set of states)、轉換函數(next-states functions)和初始狀態(initial arrays)，而這三種結構皆是由符號定義(alphabet)、文法生產規則(production rule)和起始字串(axiom)所組成，符號定義就是在L-System中用來疊代字串的所有字元符號，而文法生產規則就是每一個字元符號的疊代規則，至於起始字串即是L-System中最初用來定義操作的字串，文法中所有的字串其實都是由起始字串或初始狀態經由疊代文法生產規則而來。

L-grammar和一般的正規化語言基本上是非常相似的，除了L-grammar的疊代過程是同時發生和無法區分非終斷點符號和終斷點符號之外，此外Lindenmayer為了使文法中可以表示植物的分枝結構，因此額外引進了{[, ]}來表示，所以凡是在文法中被[ ]所括住，就表示括號中的文法是一組植物分枝的架構，如果括號中還有括號就表示是分枝的分枝，一般說來若以符號的上下文來區的話，可以將L-System分成兩種主要形式，其一為與上下文無關的文法(context-free grammar)，另一為與上下文相關的文法(context-sensitive grammar)，與上下文無關的文法又稱為 0L-System[15]，意即在疊代文法時和它相鄰的符號定義並不會影響疊代的結果，現在假設我們有一個簡單的例子，並且令符號定義為{0, 1, [, ]}，起始字串為1，而文法生產規則為{ $0 \rightarrow 00$ ,  $1 \rightarrow 0[1][1]$ ,  $[- \rightarrow [, ] \rightarrow ]$ }，那麼前三次疊代的結果應為 0[1][1]，00[0[1][1]0[1][1]] 和 0000[00[0[1][1]][0[1][1]][00[0[1][1]][0[1][1]]]，圖 3-1 中即為字串

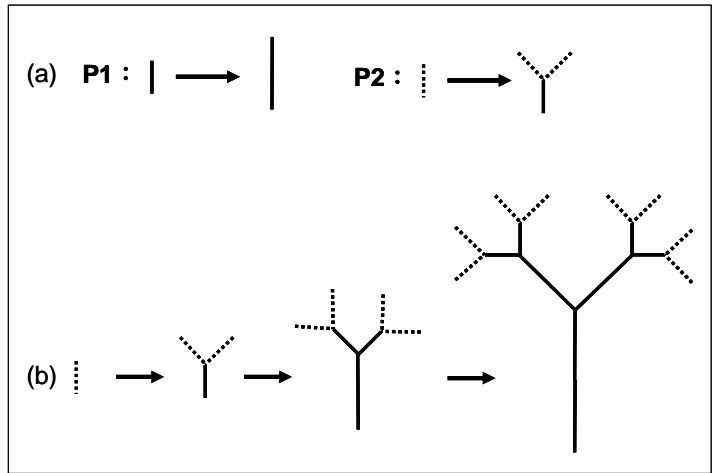


圖 3.1 與上下文無關的文法，(a)為文法生產規則，(b)為文法前三次疊代之結果

的圖形化結果，與上下文相關的文法又可被稱作  $\langle k, l \rangle$ -L-Systems，其中  $k$  表示左邊相鄰的符號定義字母， $l$  表示右邊相鄰的符號定義字母，現假設有兩個簡單的文法，其符號定義皆為  $\{0, 1, [, ]\}$ ，起始字串也都為  $l[0]0[0]0$ ，而文法生產規則各為  $\{0 > 1 \rightarrow 1, [ \rightarrow [, ] \rightarrow ]\}$  和  $\{1 < 0 \rightarrow 1, [ \rightarrow [, ] \rightarrow ]\}$ ，那麼經過兩次疊代的結果應為  $l[0]0[0]0$  和  $l[1]l[1]1$ ，圖 3-2、3-3 為其圖形表示法之結果，而對於與上下文相關的文

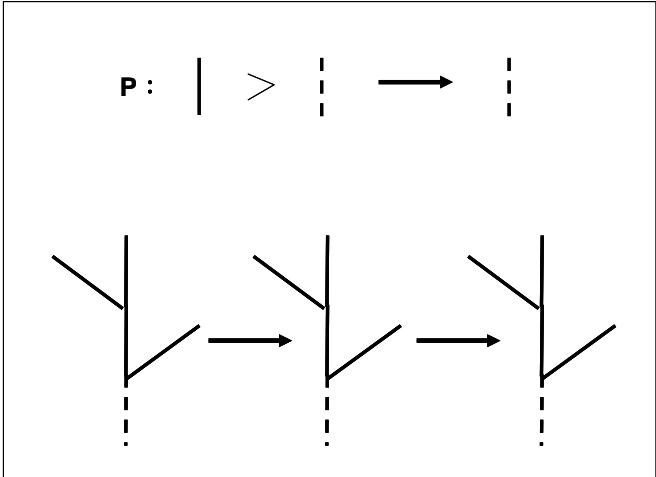


圖 3.2 與上下文相關的文法，範例一



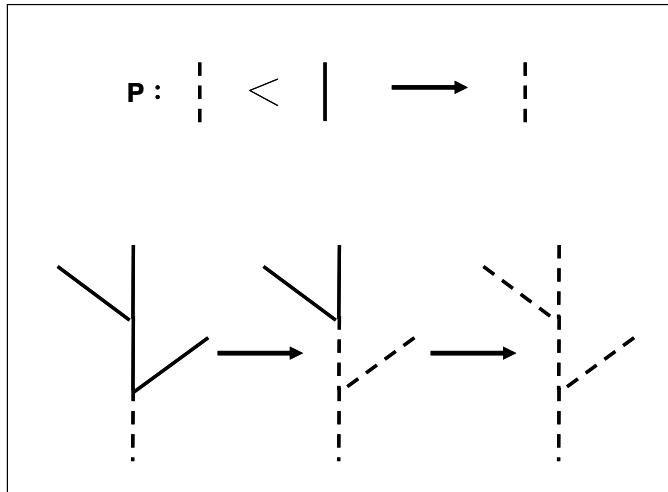


圖 3.3 與上下文相關的文法，範例二

法又可以被分為 1L-System 和 2L-System 兩種，主要的差別在於一個符號定義字母會被兩個符號定義字母所夾，那麼如果產生下一次字串時字母需參照左、右中的其中之一，那就是屬於 1L-System，假如需同時參照左右兩邊，那就是 2L-System；不確定性(non-deterministic)也就是說，同樣的起始字串且套用同樣的文法產生規則卻可以產生多組不同的文法字串，也就是具有不一樣的結果，而非傳播性(non-propagating)則是文法疊代的過程中會有死亡或者消逝的狀況發生，就是會有符號定義字母消失的情形發生。

目前我們所引進的 L-System 是具有不確定性和非傳播性之  $\langle k, \ell \rangle$ -L-systems，它具備高度的文法彈性度、植物特性的展現力和文法撰寫直覺，因此只需要簡單的文法就可以增加我們系統影像的擬真度。

## 3.2 植物運動物理模型的理論基礎

為了模擬植物結構受力後所造成的改變，我們必須將經由疊代產生的 L-string 轉換成一骨架結構 (skeleton structure)，其轉換方式為從生長規則的起始字元開始追蹤，去建立每一段莖與分枝之間的從屬關係與連接關係，以及每段莖相對於其父

節點之夾角角度，每一根 skeleton 都有自己的向量  $V_s$  以及相對於其 parent 的一組夾角角度(rotr, rotu)，關於這組角度的詳細定義，會在第五章系統實作部分做介紹。有了這個骨架結構的定義之後，可以更容易地將物理模型套用至三維植物模型。關於骨架結構的示意圖請見圖 3.4。

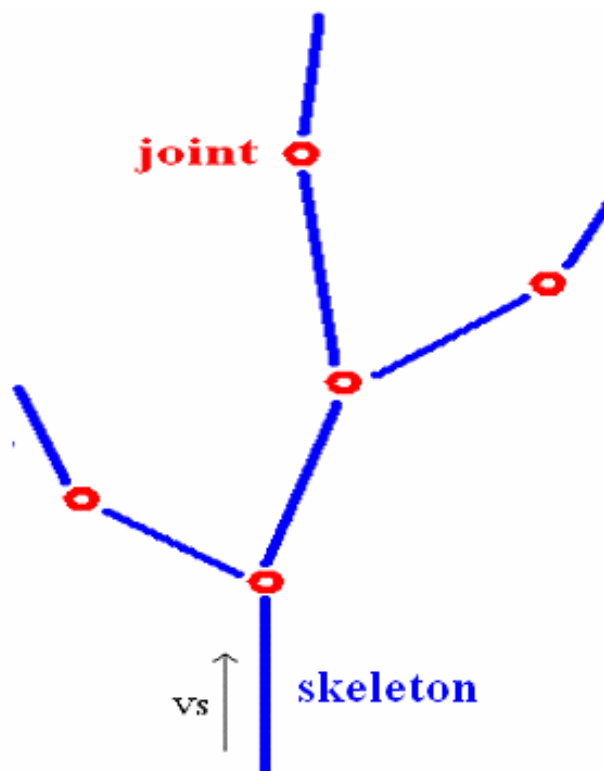


圖 3.4 骨架結構示意圖

接下來要討論我們所使用在骨架結構上的物理模型，本研究使用了雙單擺 (double pendulum) 作為主幹的物理模型，並且在骨架相接處加進彈簧(spring)跟阻尼(damper)的概念，以下會逐一介紹其物理定義。

彈簧：是一種結構上的單元，當它連接兩個物體時，會有量值相同但方向相反的兩個力作用在兩個物體上，根據虎克定律，彈力為彈簧由原先靜止的長度，因受力而伸長或縮短的長度與彈力係數的函數。彈力係數是與彈簧受力而形變有關的量值：

$$F_s = k_s (L - r)$$

其中是 $F_s$ 是彈力係數， $L$ 是彈簧伸長或壓縮後的長度，而 $r$ 則是彈簧靜止時的長度。

阻尼：在數值模擬中常與彈簧合併使用，阻尼的作用與黏滯阻力相同，皆為抵消速度。在這種情況下，如果阻尼連接兩個正在靠近或遠離的物體時，阻尼會試著減慢兩物體的相對速度。由阻尼所發出的力，與兩連接物體之間的相對速度與阻尼係數 $k_d$ 成正比，相對速度與阻尼力(damping force)的關係式如下：

$$F_d = k_d (v_1 - v_2)$$

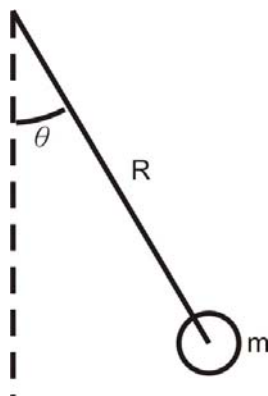
當我們同時在系統內使用彈簧跟阻尼時，我們可以將它們組合成為單一的彈簧/阻尼單元，因此用一個公式就能計算合併之後的力。用向量表示時，連接兩物體的彈簧/阻尼單元的公式如下：

$$\mathbf{F}_1 = -\{ k_s (|\mathbf{L}| - r) + k_d [ (\mathbf{v}_1 - \mathbf{v}_2) \cdot \mathbf{L} ] / |\mathbf{L}| \} \mathbf{L} / |\mathbf{L}|$$

$$\mathbf{F}_2 = -\mathbf{F}_1$$

$\mathbf{F}_1$ 是物體 1 所受的力，而 $\mathbf{F}_2$ 是物體 2 所受的力， $\mathbf{L}$ 為彈簧/阻尼單元的長度向量( $|\mathbf{L}|$ 則表示向量 $\mathbf{L}$ 的量值)。根據此一彈簧/阻尼單元，我們可以模擬植物枝幹受到外力影響後，在空間中的位移量以及因為阻尼力的作用，而使得擺蕩幅度漸漸變小的情況。

接下來我們要介紹雙單擺，在介紹雙單擺之前要先提一下單擺這個簡單的物理運動。單擺是由一個質點跟一段沒有重量擺長所組成的



$\theta$  = 單擺的角度 (0 度為垂直)

$R$  = 擺繩長度

$T$  = 擺繩張力

$m$  = 單擺質量

$g$  = 重力加速度

我們先定義水平方向的單位向量  $\mathbf{i}$ ，及垂直方向的單位向量  $\mathbf{j}$ 。然後我們可以定義出單擺的位置、速度、與加速度如下：

位置 =  $R \sin \theta \mathbf{i} - R \cos \theta \mathbf{j}$ 。

速度= $R \theta' \cos \theta \mathbf{i} + R \theta' \sin \theta \mathbf{j}$ 。

加速度= $R(\theta'' \cos \theta \mathbf{i} - \theta'^2 \sin \theta \mathbf{i} + \theta'' \sin \theta \mathbf{j} + \theta'^2 \cos \theta \mathbf{j})$ 。

單擺會受到兩個力的影響，這兩個力分別為擺繩的張力 $T$ 跟重力 $g$ ，公式如下

$$\mathbf{F} = T \cos \theta \mathbf{j} - T \sin \theta \mathbf{i} - m g \mathbf{j}。$$

至於雙單擺，其實就是串接兩組單擺而成的，如圖 3.5。

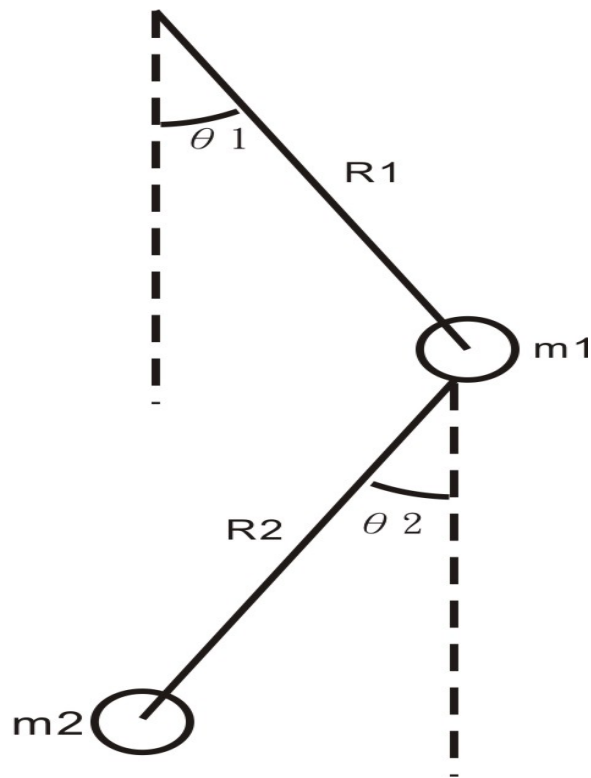


圖 3.5 雙單擺

雙單擺為兩個單擺之串接，所以有兩個質點跟兩條擺繩。在同樣給定單位水平向量

$\mathbf{i}$  跟垂直向量  $\mathbf{j}$  的情況下，我們可以求出兩質點的位置、速度、加速度如下：

$x_1 = R_1 \sin \theta_1$	$V_{x_1} = \theta_1' L_1 \cos \theta_1$	$A_{x_1} = -\theta_1'^2 L_1 \sin \theta_1 + \theta_1'' L_1 \cos \theta_1$
$y_1 = -R_1 \cos \theta_1$	$V_{y_1} = \theta_1' L_1 \sin \theta_1$	$A_{y_1} = \theta_1'^2 L_1 \cos \theta_1 + \theta_1'' L_1 \sin \theta_1$
$x_2 = x_1 + R_2 \sin \theta_2$	$V_{x_2} = x_1' + \theta_2' L_2 \cos \theta_2$	$A_{x_2} = x_1'' - \theta_2'^2 L_2 \sin \theta_2 + \theta_2'' L_2 \cos \theta_2$
$y_2 = y_1 - R_2 \cos \theta_2$	$V_{y_2} = y_1' + \theta_2' L_2 \sin \theta_2$	$A_{y_2} = y_1'' + \theta_2'^2 L_2 \cos \theta_2 + \theta_2'' L_2 \sin \theta_2$
位置	速度	加速度

# 第四章 系統設計

## 4.1 系統架構

為了能讓使用者簡便地產生生長文法，我們開發了一個名為 L-grammar Generator 的文法生成輔助系統。此系統包含兩個主要的模組，其第一個模組 Parser，第二個模組為 SimEco CAD。為了取得植物的生長資訊及結構資訊，我們需要輸入含有植物生長敘述的文字檔案以及三張經由攝影取得的植物照片。下面圖 4.1 是我們系統的基本架構與大致的流程，其中 Parser 是用來處理生長敘述的文字檔，它可以根據植物名稱去搜尋植物資料庫中有無該植物的生長敘述，然後再從找到的生長敘述中去分析與搜尋出代表植物特性的關鍵字，再利用這些關鍵字來協助我們產生含有生長資訊的植物生長文法。

至於在 SimEco CAD 中，使用者必須輸入三張由固定視點拍攝的植物照片，分別是正視圖、右視圖跟頂視圖。然後再以手動的方式去指定照片中相對應的結構，透過計算後就可以決定該植物的結構，得到主幹與分枝間的從屬關係，以及每段互相有連接的莖之間的相對夾角資訊。除了結構的資訊外，使用者可以輸入該植物的高度與最大直徑，系統就可以依此來決定莖的生長速度。有了結構資訊跟之前由 Parser 所得到的生長關鍵字的訊息後，系統就可以產生出包含生長過程的生長文法。

有了生長文法後，再將文法輸入我們所開發的 SimEco system，SimEco 會分析生長文法，從中載入場景定義、植物各器官的屬性定義，以及植物的生長規則 (production rules)。分析完文法後，就會根據生長規則去做文法的疊代，並將每次疊代後的字串根據植物莖與葉子的模型資料，還有環境設定的參數，render 出擬真的植物影像，其中字串疊代的過程即代表植物的生長過程。並且我們在疊代的過程中，會根據光源方向以及重力的影響，來調整每段莖之間的角度，以模擬出植物的向光性與背地性對植物結構所造成的影響。在疊代完成，也就是植物生長完了後，

所生成的植物模型可以根據外力的影響去做改變，我們套用了一些物理模型來模擬枝葉受到外力後搖擺的情形，關於此部分之後會做更詳細的介紹。

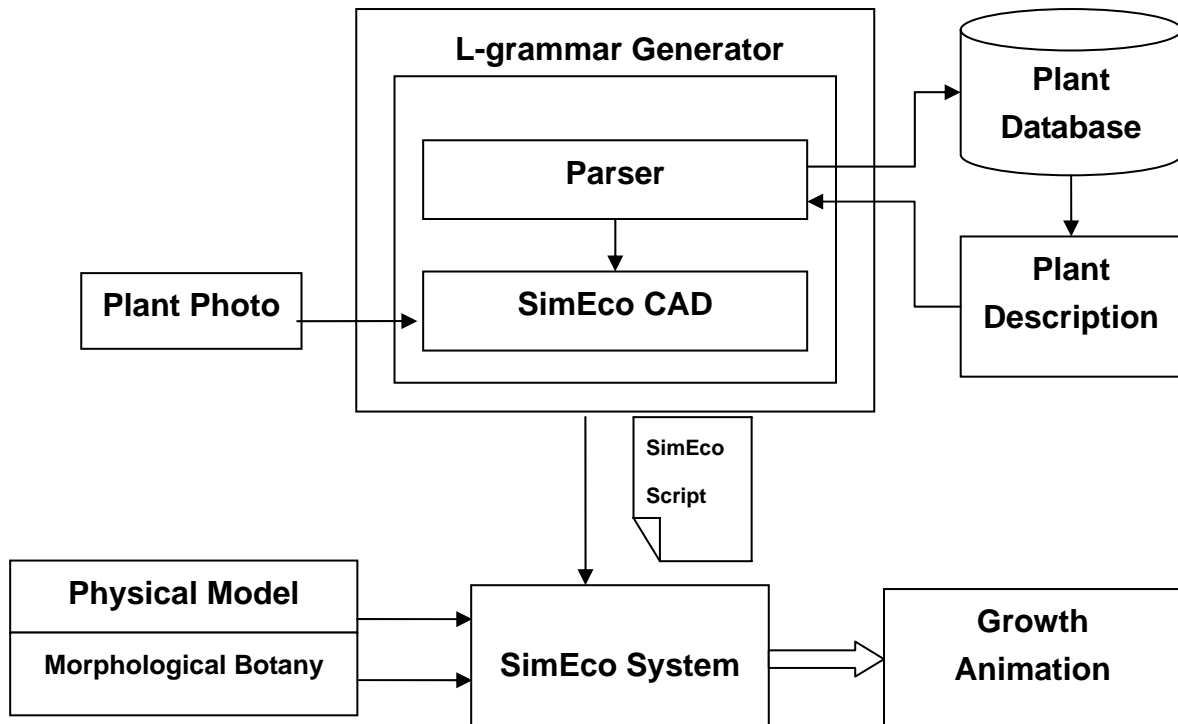


圖 4.1 系統的基本架構

## 4.2 系統流程和操作流程

下面我們會說明 L-grammar Generator 基本的系統流程和操作流程，在系統流程中我們會再次說明各單位元件的基本功能和執行方法以及資料的傳輸，而在操作流程中我們會著重在使用者操作系統時，系統的決策過程。

在執行 L-grammar Generator 時，首先在 parser 模組輸入植物的名稱，然後 parser 會在植物資料庫中搜尋有無該植物的生長敘述，找到敘述後就從中搜尋生長關鍵字 (性狀、葉序等....)，另外再讀入植株高度跟主莖的最大直徑，完成文字資料的輸入步驟。當這些文字資料都輸入系統後，將這些文字資料傳到 SimEco CAD 模組。SimEco CAD 還須再輸入三張植物照片，經由使用者手動拉出植物結構後，根據文

字資料及從影像中取得的量化資訊還有植物學中對植物生長的定義來產生植物的生長文法。有了生長文法後就要模擬植物生長過程跟環境互動的情形。我們是使用自行開發的 SimEco System 來進行文法的疊代及 rendering 的工作，在 SimEco System 中會將表示疊代結果的 L-string 轉換成一骨架結構 (skeleton structure)，更方便地跟環境產生互相效果。生長的過程跟結果會受植物形態學中對植物形態的一些生長限制，還有根據我們給定的物理模型所計算出之結構所影響。系統將換產生連續的動畫來表現這些效果。

系統的使用方式，首先使用者會開啟 parser 模組然後輸入植物名稱，parser 的操作介面請見圖 4.2。



圖 4.2 parser 操作介面

當輸入完後系統會根據植物名搜尋資料庫中有無生長敘述資訊，並從中取得該植物的生長關鍵字。如果沒有找到的話使用者必須手動輸入生長關鍵字，並且輸入植株高度與莖的最大直徑。文字資料都設定好後，按確定鍵進入 SimEco CAD 模組 (圖 4.3)。在 SimEco CAD 模組中，必須載入三張植物照片 (分別是正視圖、右視圖及頂視圖)，由使用者在三張照片中選取出相對應的枝幹，以此產生在三度空間中的植物結構模型。有了枝幹的模型後，再自行指定葉子的排列順序跟分布情況，讓模

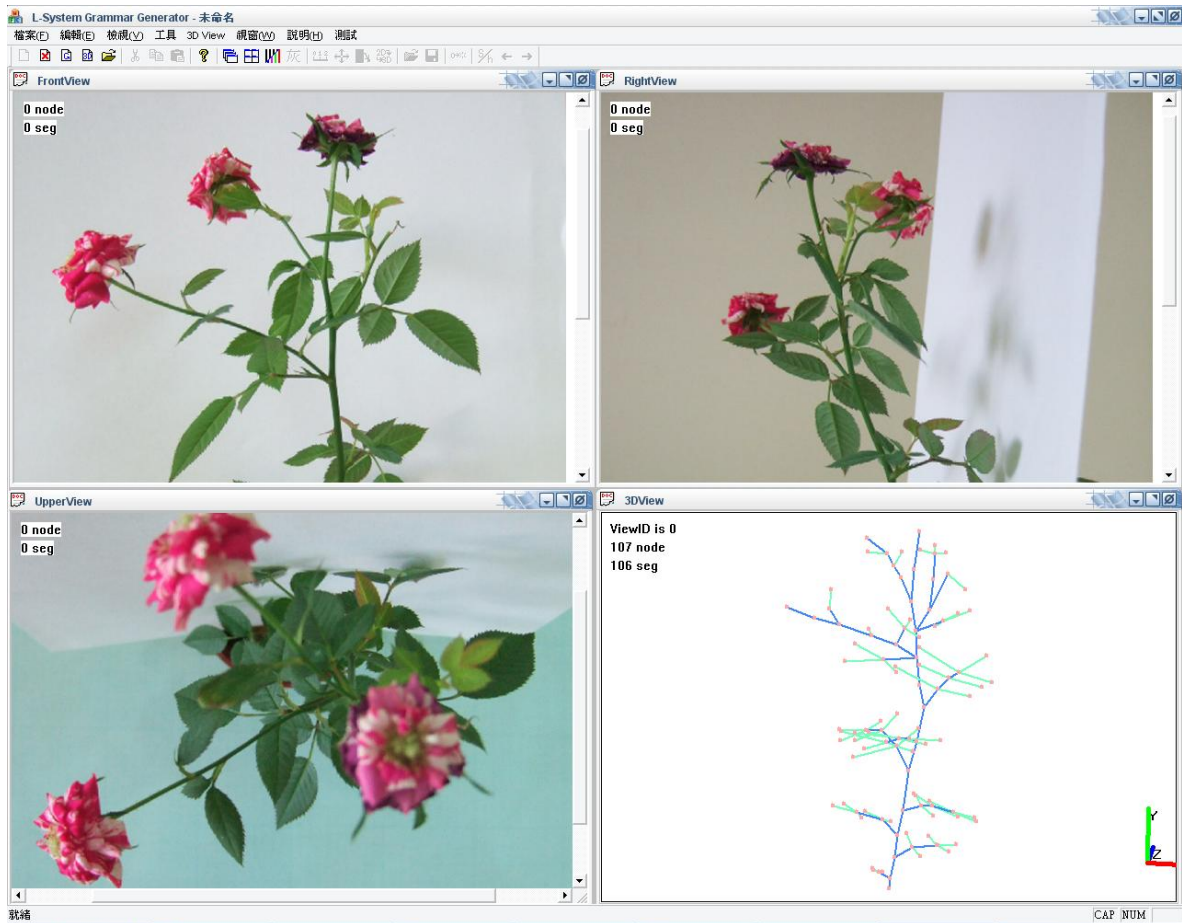


圖 4.3 SimEco CAD 操作介面

型的結構更接近真實的植物架構。結構指定好後，開啟文法生成對話框，由系統自動根據所有的文字及數據資訊產生生長文法，之後再將文法存進文法資料庫。之後再由 SimEco system 讀入文法，進行文法疊代與產生植物影像的工作。開啟 SimEco 後要先使用文法讀入對話框來載入生長文法，在經過 SimEco 對文法的分析與載入各器官的模型與屬性之後，系統會根據起始字元及生長文法展開疊代的步驟。此時系統就會將每次疊代後的結果所表示的植物影像展示在畫面上，視窗下面還有現在表示目前狀態的 L-string。待植物生長完了後，使用者可以設定場景中風的方向與強度，來觀看結構受風力影響所產生搖擺的效果。詳細的系統操作過程如圖 4-3 所示。



# 系統操作流程

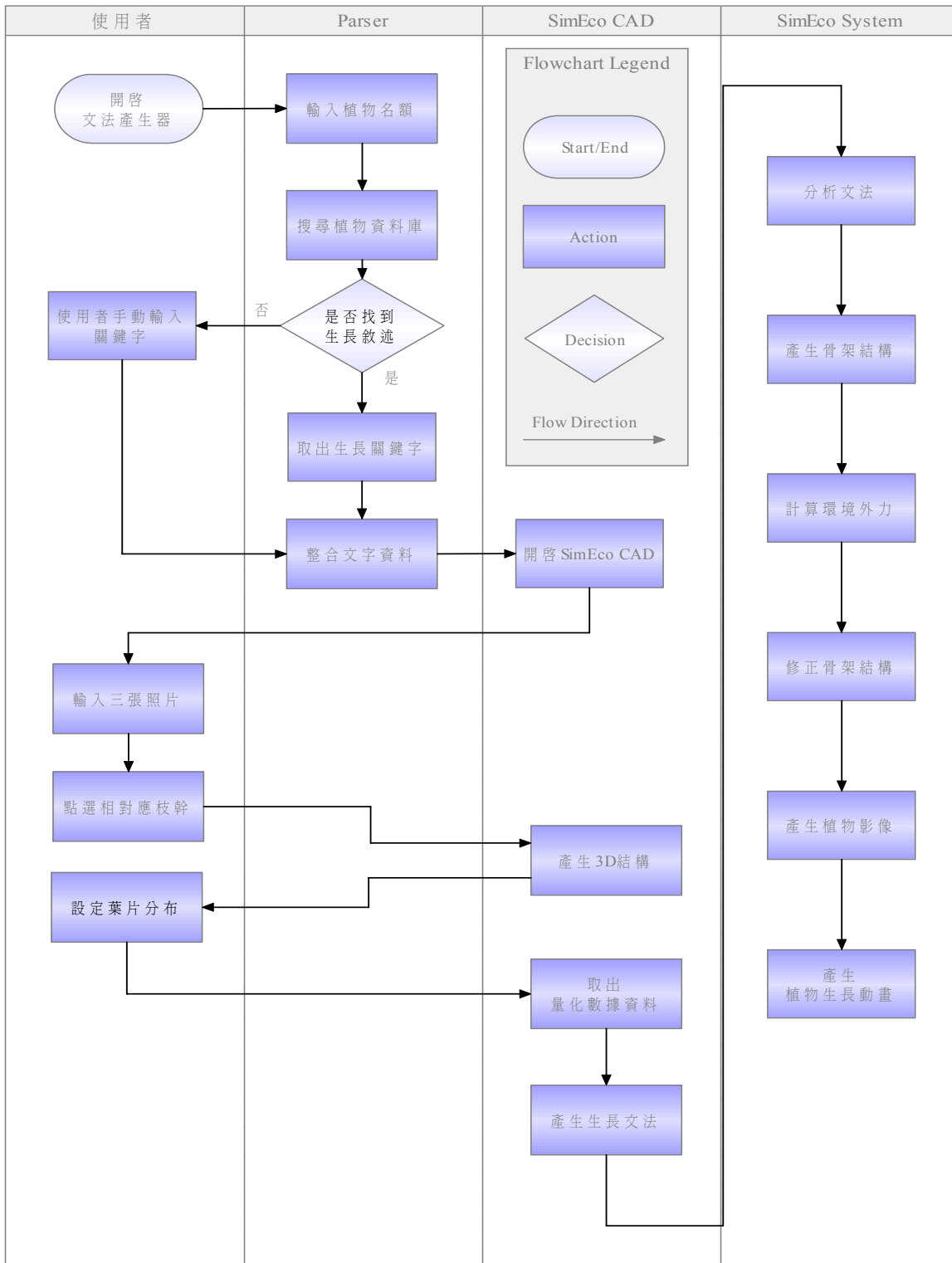


圖 4.4 系統的操作流程

# 第五章 系統實作

## 5.1 所需要的檔案

在我們的系統中需要載入兩種檔案，一種是植物生長敘述的檔案，另一種是三個不同角度所拍攝的植物照片檔，其中生長敘述的檔案可以從植物資料庫中取得，而植物的照片檔案中需要有三個角度所拍攝的影像 — 正視圖、右視圖與頂視圖。

### 5.1.1 植物生長敘述檔

植物生長敘述檔，是由植物學家長期觀察植物後根據植物特徵及生長情形所撰寫出文字敘述，一般都會包含下列資訊：科名、學名、英文名、源地、分佈地、植株類型、葉序、葉形（含葉端、葉基、葉緣、質理、葉脈）、花序、花形等資料，這些資料有的是條列式列出，有的是以敘述的方式寫成，下面是玫瑰生長敘述檔的簡單範例：

中文名：玫瑰
英文名：Rose
科名：薔薇科 Rosaceae
學名：Rosa hybrida Hort.ex. Schleich
源地：北半球之極地帶至北非洲、墨西哥及印度、中國大陸
分佈地：全台栽植
植株類型：常綠灌木
葉序：大互生、小對生
葉形：橢圓形、長卵形
葉端：銳、突尖、
葉基：鈍
葉緣：鋸齒
質理：薄革質
葉脈：羽狀側脈

### 5.1.2 三視點植物照片檔

為了得知植物的結構在空間中的關係，以及取得植物量化的數據，我們需要三張攝影角度必須互相垂直的照片，拍攝角度示意圖請看圖 5.1。以便從互為正交的影像當中重建回樹狀物體的三維模型。圖 5.2 就是依我們的條件所拍攝的植物照片。

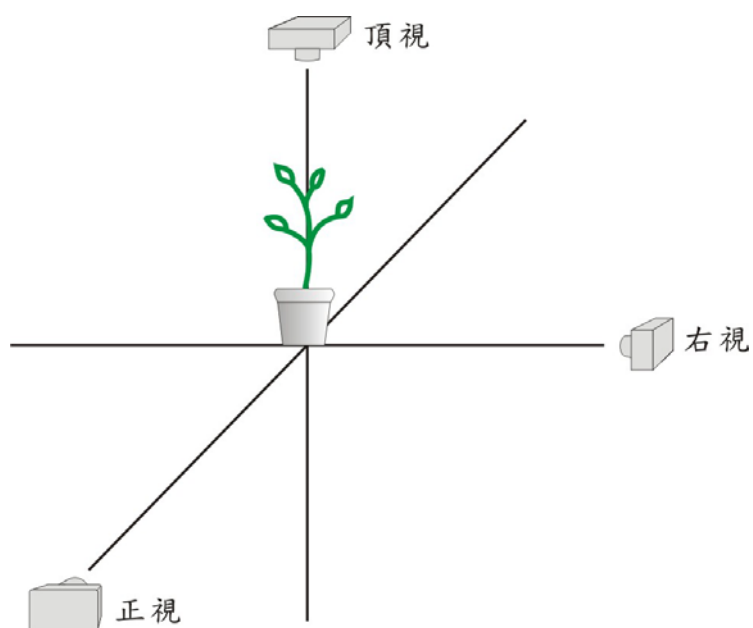


圖 5.1 拍攝角度示意圖

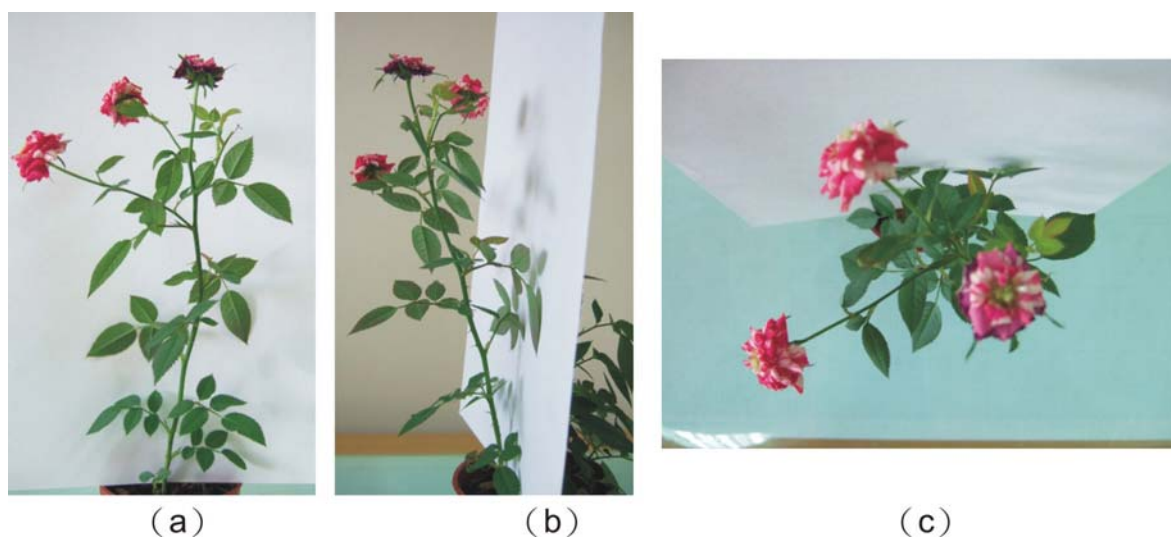


圖 5.2 玫瑰的照片，(a) 是正視圖，(b) 是右視圖，(c) 是頂視圖

## 5.2 生長文法的生成

這一節主要說明的是 L-grammar Generator 如何自動產生出植物的生長文法 (L-grammar)，而且要讓文法疊代的過程能符合植物合理的生長過程。系統必須根據描述植物外觀及特性的文字資料，還有從照片中取得的植物結構量化數據資料來生成生長文法，如果就光只有這些資訊其實是無法得之其生長過程的。在前一個版本的 L-grammar Generator 中，系統所產生的生長文法只有最後一步的疊代結果符合植物的形態，其疊代的過程並不符合植物生長的原則，也不具有任何的意義。所以我們將在系統中加入植物學的觀點，設法將植物生長的過程用生長規則 (production rules) 表現出來。

### 5.2.1 文字資料的使用

這邊所指的文字資料主要是從植物生長敘述中經由字串比對所擷取出來的生長關鍵字，關鍵字的種類整理在表 5.1：

關鍵字種類	說明
葉序	對生、十字對生、互生、輪生、叢生
植株類型	喬木、灌木、草本
生存週期	一年生、一至二年生、多年生
葉形	橢圓形、圓形、倒心形、心形、倒卵形、卵形、菱形、盾形

表 5.1 生長關鍵字的種類

有了生長關鍵字後，我們要將關鍵字所代表的意義表現在生長文法裡面，其中葉序關鍵字可以決定植物的基本結構，性狀關鍵字可以決定莖的模型跟貼圖，葉形關鍵字則可以決定葉子的模型及貼圖，至於生存週期關鍵字是使用在長時間模擬時所使用。

## 5.2.2 量化數據的使用

在上一小節中我們說明了如何將從文字中所取得的資訊用在文法生成上，不過也可以看出其實能從文字中所取得的訊息相當少，我們只能得知該植物大約的生長方式，就算有了葉序關鍵字，我們也無法從生長敘述中得知每組葉序結構相間隔的距離或是分枝結構與主幹結構間所夾的角度。另外對於葉片生長的分佈情形，以及葉片與植株的大小比例，都無法從生長敘述中來取得。所以我們從照片中取出我們所需要的量化資訊，並用這些數據資料來建構植物的模型。在圖 5.3 中可以看到我們從照片中所取出的結構，這個結構是由一堆線段跟節點所組成的，整組結構就包含了所有的量化資訊。

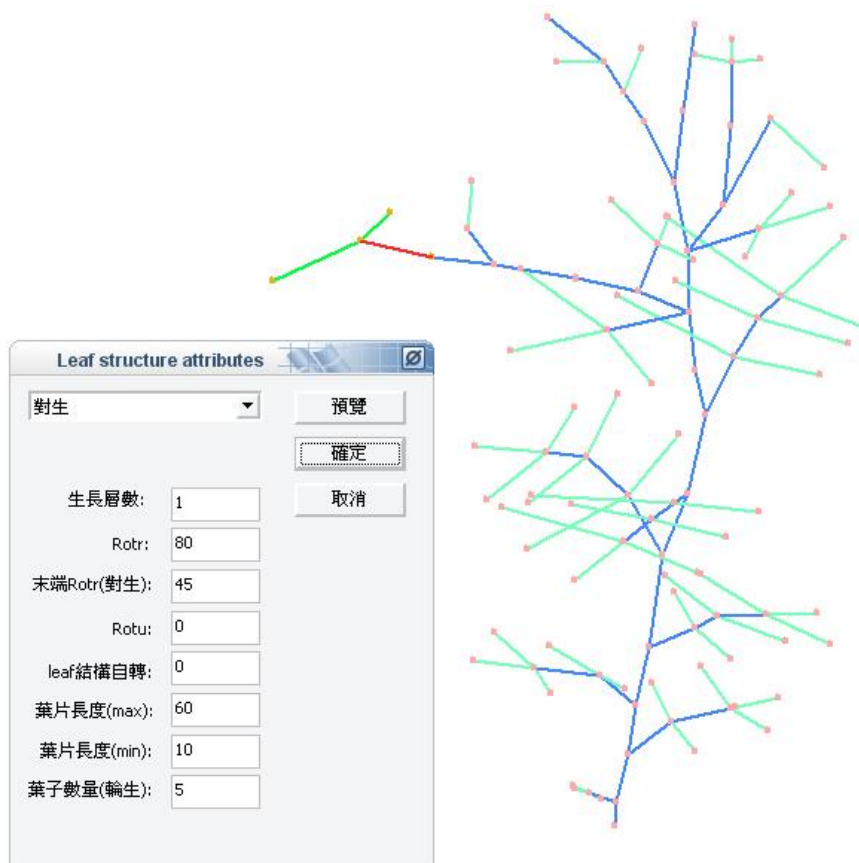


圖 5.3 植物結構模型

除了每段莖的長度外，最重要的資料就是莖與莖之間的相對夾角，我們將相對夾角定義如下。首先每一段莖都有一組互相垂直的單位向量，分別是代表軸心的 $V_{\text{skeleton}}$ 跟與其垂直的 $V_{\text{right}}$ ，如圖 5.4。

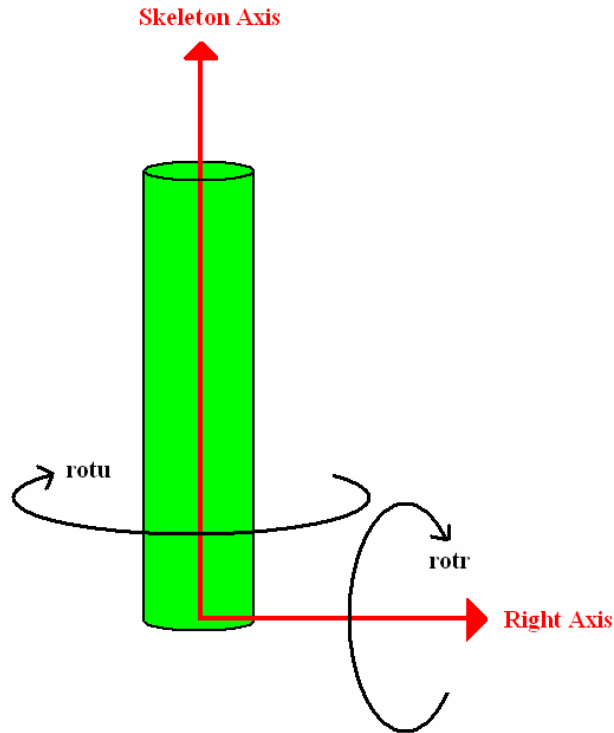
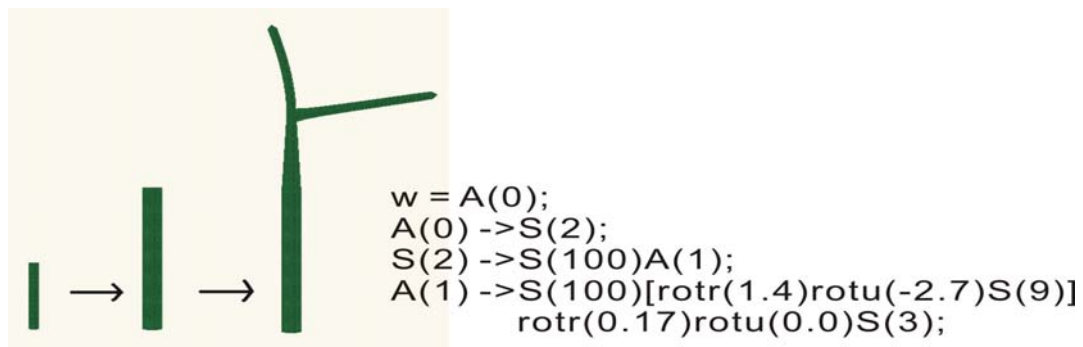


圖 5.4 莖的兩個向量

我們將root stem的skeleton axis定為 $(0, 1, 0)$ ，right axis定義為 $(1, 0, 0)$ 。root stem由於沒有parent，所以並沒有相對夾角的資料。除了root stem外，所有stem的角度都是相對於其parent的。相對夾角有兩組數據，分別是rotr跟rotu，單位是弧度。其定義為child stem的兩組向量的計算方式為先求得parent的 $V_{\text{skeleton}}$ 先以其right axis為軸旋轉rotr度後再以skeleton axis旋轉rotu度所得到旋轉矩陣 $M_{\text{rotation}}$ ，然後再將parent的 $V_{\text{skeleton}}$ 跟 $V_{\text{right}}$ 乘上 $M_{\text{rotation}}$ 求得child的兩組向量。所以我們只要有兩段相連stem的 $V_{\text{skeleton}}$ 和parent的 $V_{\text{right}}$ ，就可以推算出child stem相對於parent stem的兩組角度資訊。

### 5.2.3 綜合資料生成文法

有了文字資料跟量化數據後，接下來就是產生生長文法的工作。為了能使我們產生的生長文法的疊代過程可以符合植物的生長過程，我們要先得知植物的生長規則。首先，字典上對生長的定義是：「植物或動物攝取食物後體積增加之過程」，體積如要增加，它必須是永遠的。換句話說，應該是不可逆的(irreversible)。且植物的生長包括細胞數量的增加以及單個細胞體積的增加。一般來說，只有在頂芽部分的分生組織會持續經由細胞分裂產生新的細胞，其他部分的細胞只是漸漸增加本身的體積。所以莖的末端生長速率最大，且莖會先生長至一定長度即停止變長，之後才開始增加它的直徑。所以我們要將上述這些植物生長的現象用 L-grammar 來表現，以此來產生模擬植物生長的動畫。首先針對頂芽分生速度最快這點，我們設計了下面的文法來表示：



上面這段生長規則表示三步疊代過程，一開始的生長點  $A(0)$  會變成代表莖的  $S(2)$ ，第二條生長規則會將  $S(2)$  變成  $S(100)A(1)$ ，即一段莖跟頂端的一個生長點  $A(1)$ ，然後在第三條生長規則中，再由  $A(1)$  開始產生分枝結構。如何一來便可符合植物末端為生長及分生的起點此一規則。至於另一個生長現象：莖會先生長至一定長度即停止變長，之後才開始增加它的直徑。我們用下列的生長規則來模擬：

```

W = A(0);
S(2) -> S(100) A(1) ;
A(1) -> S(100) [rotr(1.4)rotu(-2.7)S(9)] rotr(0.17)rotu(0.0)S(3) ;
S(9) -> S(200) A(14) ;
//控制 stem 的生長次數
S(100) -> S(101);
S(101) -> S(102);
S(102) -> S(103);
S(200) -> S(201);

```

從上面這段生長規則中，前半部是定義出植物的結構，後半部則是用來控制莖的疊代次數，並以此來決定莖的粗細。其中 S(100)會經過三次疊代後變成 S(103)，每經過一次疊代系統就會根據該 Stem 的生長速率重新計算模型的大小，所以經過越多次的疊代就表示該 Stem 可以長得越粗大。另外根據起始狀態的不同，S(100)跟 S(200)可疊代的次數不同，所以最後長成的大小也不同，因為可以用來代表不同粗細的莖，也符合生存時間越久的器官其體積越大這一點。

綜合以上所有的資訊跟規則，系統可以自動產生生長文法，並且會將生長規模依其階層產生不同的縮排，方便使用者理解生長規則，以更進一步地去對生長規則做微調的動作。

### 5.3 物理模型的實作

在 SimEco 中，為了加強動畫的擬真度所以我們引進了一套物理模型來模擬植物結構受外力影響後所產生的改變。我們在系統中實作兩種環境因素的影響，一個是在生長過程中根據光源位置去將結構做調整，藉此來模擬出植物的向光性。另一個則是在生長過程完了後，根據變動的風力向量去計算植物結構，藉此來模擬被風吹動的效果。因此在本節我們將說明我們如何去實作這兩個動態效果，SimEco System 處理此部分的流程請參照圖 5.5。



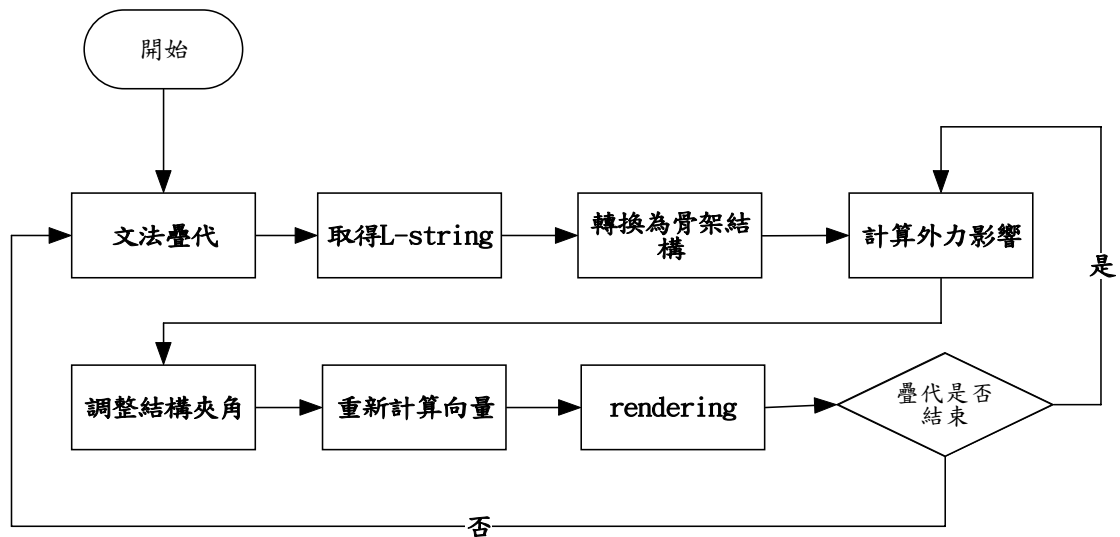


圖 5.5 物理模擬的流程圖

### 5.3.1 植物向光性模擬

在生長的過程中，也就是系統還在做生長規則的疊代時，系統會根據光源的位置，不斷的去調整骨架結構中每根skeleton的角度，調整的方式為直接將skeleton的向量 $V_{skeleton}$ 加上一指向光源位置的向量 $V_{toLight}$ ，得到一個新的 $V_{skeleton}'$ ，我們根據 $V_{skeleton}'$ 重新計算該skeleton相對於其parent的夾角角度後再重繪整個結構。因此，根據不同大小的 $V_{toLight}$ ，可以產生不同程度的偏向效果。

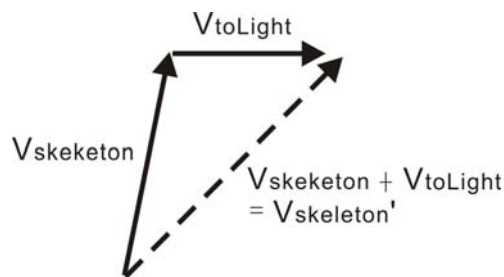


圖 5.6 計算 skeleton 偏向後的向量

當我們重新設定每根skeleton的角度後，系統要重新計算出整株植物每根skeleton的向量 $V_{skeleton}$ ，之後再進行模型資料的更新及rendering的動作。

### 5.3.2 風力效果模擬

首先我們會將經由生長文法疊代後所產生的 L-string 轉換成骨架結構，其方法為從根部開始記錄每段 skeleton 的從屬關係以建立一棵樹狀結構。然後根據目前各段 skeleton 的長度 L 和半徑 r 還有密度 d 計算出質量 M 以及彈性係數 ks 及阻尼係數 kd，計算方式如下式：

$$M = L \cdot (\pi r^2) \cdot d。$$

ks 和 kd 則與質量成正比。

在 SimEco System 每次更新畫面前根據外力及骨架結構中每一根 skeleton 重新計算骨架結構中各 skeleton 之間的夾角度數。在系統中會將骨架結構分成 main stem 跟 branch stem 兩組來考慮(見圖 5.7)，只有 main stem 最末端的 skeleton 會受外力影響，外力由向量 F 代表，向量 F 的長度即代表外力的大小。系統中每根 skeleton 都有一貫穿軸心的向量  $V_{skeleton}$ ，設主幹最末端的莖的向量為  $Terminal\_V_{skeleton}$ ，當受到外力 F 作用時，系統會根據 terminal skeleton 的質量及彈性係數 ks 計算出其偏向的角度及方向，當偏向的角度小於一個與其彈性係數成正比的 threshold 值時，則該 skeleton 的動作就不會影響到其 parent skeleton。在外力消失後則會根據其彈性係數 ks 及阻尼係數 kd 來回擺動將能量給消散掉後歸於靜止。

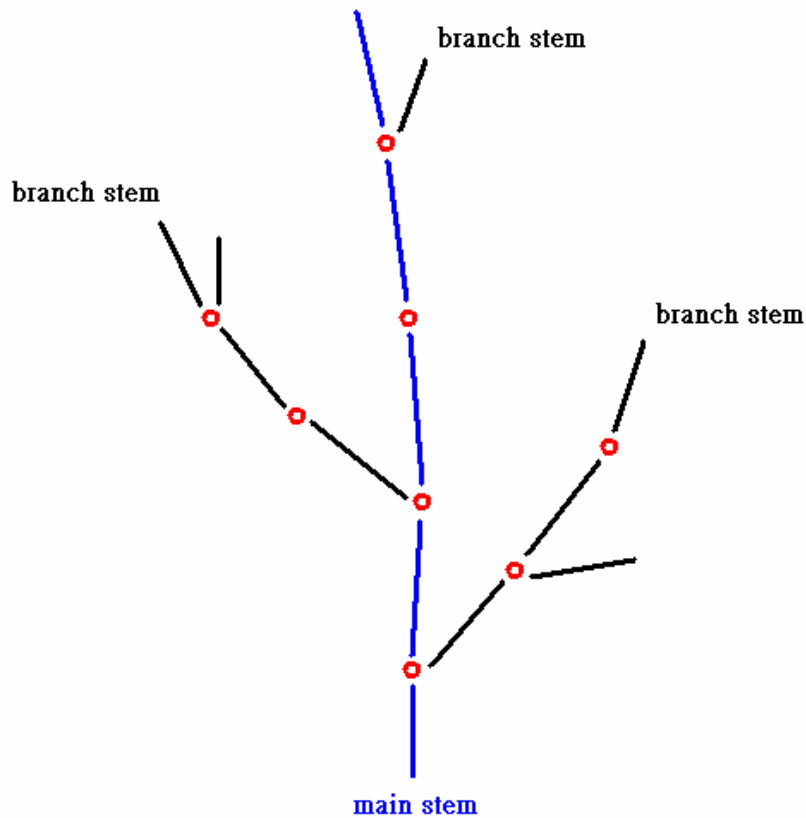


圖 5.7 main stem and branch stem

若是外力所造成的偏向角度大於該threshold，我們就必須處理該skeleton會對其parent所造成的影響。我們在這個地方使用雙單擺(double pendulum)這個物理模型，去模擬兩段相連的質點系統對彼此所造成的影響。在這由terminal skeleton跟其parent skeleton所構成的雙單擺系統中，擺長分別為兩根skeleton的長度，單擺質量則為兩根skeleton的質量。不過由於就實際情況來說，莖的質量並不是集中在其末端，其質量中心應該是在代表莖的圓柱體的正中央，所以我們將擺長設為該skeleton長度的一半來計算雙單擺系統。接下來要解釋如果計算雙單擺系統，首先將最末端的單擺的質點稱為 $M_2$ ，擺長為 $L_2$ ，接著此一單擺的質點為 $M_1$ ，擺長為 $L_1$ 。如下圖 5.8。

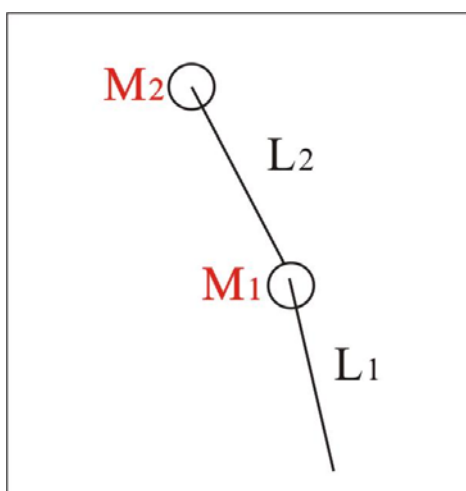


圖 5.8 倒過來的雙單擺

當 $M_2$ 移動後， $M_1$ 會受到一個由 $M_1$ 指向 $M_2$ 方向的力 $F_1$ (其實也就是連接 $M_1$ 及 $M_2$ 擺繩上的張力)， $M_1$ 也就是根據此力去決定它該偏轉的方向與角度。之後就重覆地根據 $M_2$ 的位移來計算 $M_1$ 的位移，這樣就完成這一組雙單擺系統的計算。我們可以重覆雙單擺的計算，往主幹的根部去做模擬。其概念如圖 5.9。

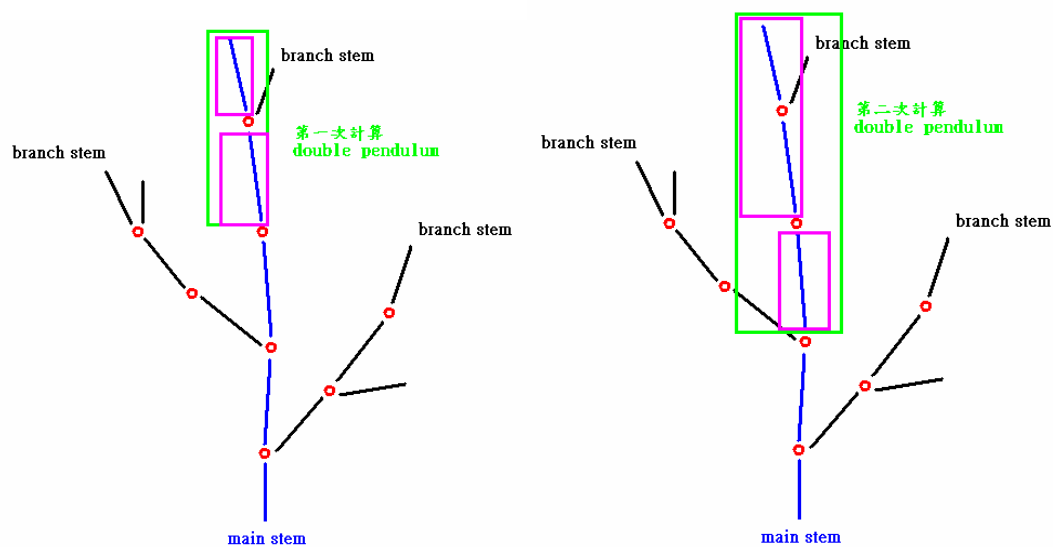


圖 5.9 多次計算雙單擺過程

由圖 5.8 可知，當做完第一次的雙單擺計算後，如果系統判斷會影響到它們的 parent skeleton，則會將第一次單擺運動的兩根 skeleton 當做一個質點考慮，繼續與其 parent

skeleton 做第二次單擺運動的計算，其下依此類推。不過由於能量會因為阻尼的存在而逐漸減小，所以越接近 root 的 skeleton 其擺動的幅度也會越小，或者甚至根本不會因外力影響而擺動。最後要考慮的是 branch stem 的受力情形，為了降低系統的計算量，我們對於 branch stem 的部分並不去仔細的考慮其真實情況，而用比較直觀而且簡單的方式來產生擺動的效果。我們將由多根 skeleton 所組成的 branch stem 當成一根 skeleton 來考慮，其質量也就是該 branch stem 所有 skeleton 之總和。所以當 branch stem 擺動時，只有跟 main stem 相接的 skeleton 的角度會改變，其餘 skeleton 之間的夾角則保持不變。概念示意圖如圖 5.10。

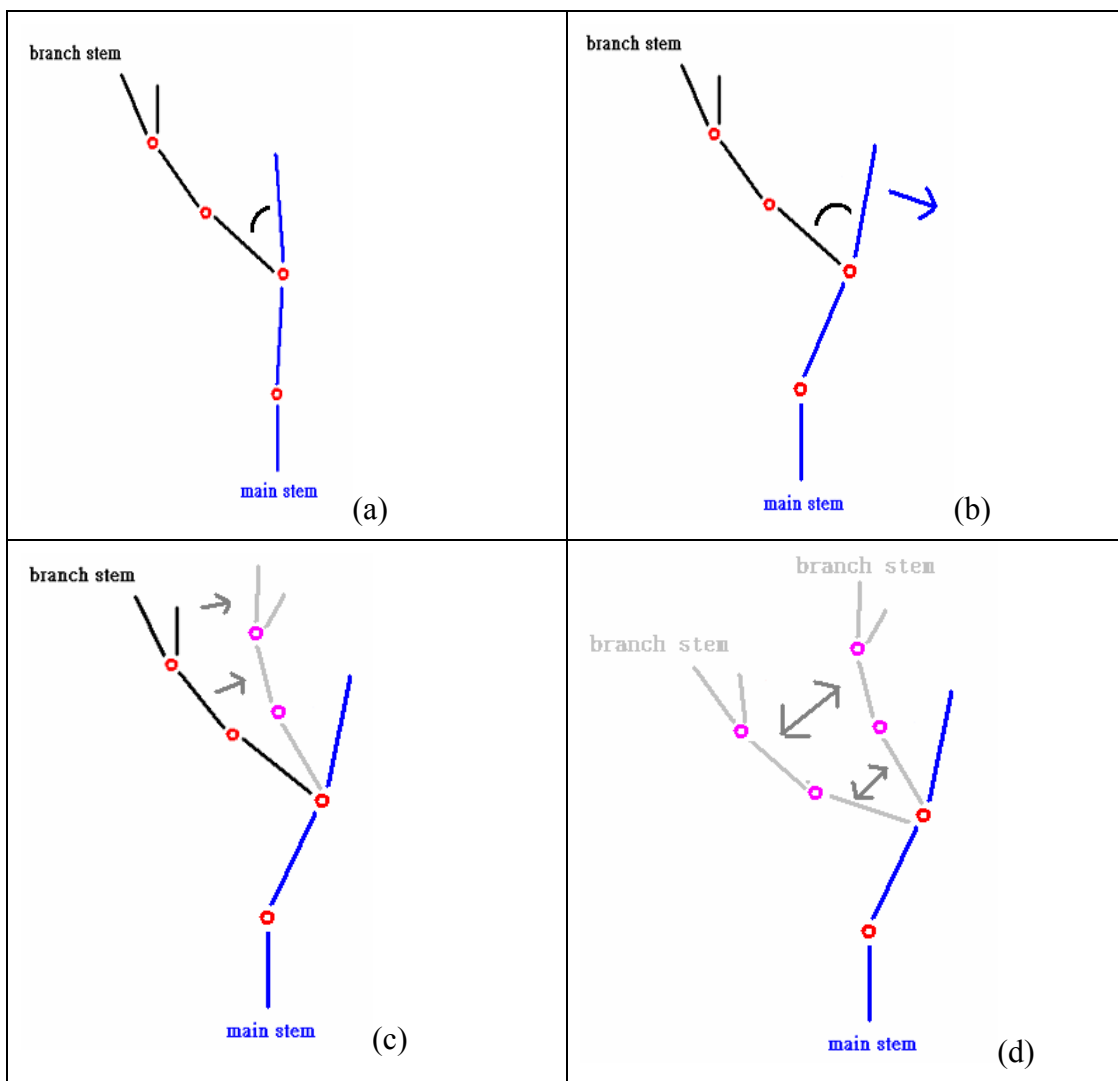


圖 5.10 branch stem 的模擬

在圖 5.9 中，(a)代表靜止的狀況，(b)表示當 main stem 受到外力影響後，會使 main stem 跟 branch stem 間的角度改變，(c)因為角度的改變，branch stem 產生一恢復力，(d)使用彈簧/阻尼單元，計算出 branch stem 的擺蕩效果，直到 branch stem 跟 main stem 之間的夾角回覆到靜止時的角度為止。

## 5.4 研究成果

研究成果分為兩個部分，一個是L-grammar Generator所產生的生長文法，以及根據這個生長文法所模擬出的生長過程動畫。另一部分為根據物理模型所計算出植物受外力影響的動畫，所有的影像都是由SimEco根據生長文法所rendering出來的。由於L-grammar Generator所產生出的玫瑰生長文法太過於冗長，因此我們僅將生長規則(production rules)的部分列於附錄當中。至於動畫部分的檔案請至我們實驗室的網站(<http://163.22.21.164/web>)下載，論文上僅列出部分成果影像。下圖 5.11 為植物照片跟SimEco產生的植物影像的對照圖：





照片右視圖



影像右視圖



照片頂視圖



影像頂視圖

圖 5.11 結果對照圖







## 第六章 結論

在目前的各種 3D 遊戲及各種虛擬場景的建構中，植物的存在已經是無法避免且不可缺少的，因此增加植物的真實度將可以大大的提升場景的真實感，且對於研究植物生長和環境互動的學者來說，如果能發展一套所知即所見的植物模擬系統將可以減少植物研究的時程，因此發展出相對應的系統可以確定是一件極為重要的課題，然而要開發如此的系統並不是簡單的事情，不但要了解植物的分佈狀態，如：水源的影響、光源的影響、重力的影響…等，單單上述所提的幾項就是一門專業的課題，更不要說要產生擬真的影像還需要有更先進以及快速的植物生長文法的產生器。

雖然現在的 L-grammar Generator 已經可以根據文字及照片資料產生出可表示生長過程的 L-grammar，但是還是有幾樣需要解決的問題。關於植物的生長過程其實我們是用幾點從植物學書籍中歸納出的要點，不過不同的植物應該會有各自獨特的生長特性，另外在不同的外環境因子影響下，也會對植物的生長行為造成影響。如果要忠實的將這些特性表現出來，就需要長時間去記錄植物的生長過程，再從中去分析其特性。

至於我們所提出的物理模擬部分，其實也有很多問題並沒有真的去處理。我們是將主幹與分枝分開討論，並且簡化了分枝部分的計算，其實這都無法真正表現出植物受力後的運動行為。實際上應該使用類似有限元素法(finite element)的方式來處理植物的每一個質點互相拉動的關係，不過在硬體能力的限制之下，這樣的做法會造成計算上的瓶頸。所以之後我們希望在可接受的計算時間內，盡量使用更真實且精確的物理模型來模擬植物的運動。

另外對於環境因子對植物生長影響的程度，也是個研究的題目，像是植物向光性的行為，我們也只是大約模擬出有此影響，但是我們模擬出的結果是否正確則不

得而知，至少以我們過去對植物的了解來說，此一結果並不會太背離我們過去的經驗，所以此一部分我們需要更多植物學方面的專業知識來輔助我們的系統，以產生出更準確的植物影像。

上述幾個議題是目前在我們的系統中比較不足的部分，另外系統對植物生長預測的問題，並還沒有一個很好的解決辦法，因此我們先著重在從種子到植物照片中的那個時間點這段生長過程的模擬，接下來植物會如何繼續生長下去，則留到未來再做處理。

## 参考文献

- [1] G. V. G. Baranoski and J. G. Rokne. *An algorithmic reflectance and transmittance model for plant tissue*. Computer Graphics Forum, Volume 16, Issue 3, pages 141 – 150, 1997.
- [2] G. V. G. Baranoski and J. G. Rokne. *Efficiently simulating scattering of light by leaves*. The Visual Computer 17(8), pages 491 – 505, 2001.
- [3] O. Deussen, P. Hanrahan, B. Lintermann, R. Měch, M. Pharr, and P. Prusinkiewicz. *Realistic modeling and rendering of plant ecosystems*. Proceedings of the 25th annual conference on Computer graphics and interactive techniques, pages 275 – 286, July 1998.
- [4] O. Deussen, C. Colditz, M. Stamminger, and George Drettakis. *Interactive Visualization of Complex Plant Ecosystems*. Proceedings of the conference on Visualization '02, pages 219 – 226, October 2002.
- [5] T. Ijiri, S. Owada, M. Okabe, and T. Igarashi. *Floral diagrams and inflorescences: interactive flower modeling using botanical structural constraints*. ACM Transactions on Graphics (TOG), Volume 24, Issue 3, pages 720 – 726, July 2005.
- [6] G. Kokai, Z. Toth, and R. Vanyi. *Evolving artificial trees described by parametric L-systems*. 1999 IEEE Canadian Conference on Electrical and Computer Engineering, Volume 3, pages 1722 – 1727, May 1999.
- [7] A. Lindenmayer and G. Rozenberg. *Developmental systems and languages*. Proceedings of the fourth annual ACM symposium on Theory of computing, pages 214 – 221, May 1972.
- [8] B. Lintermann and O. Deussen. *Interactive Modeling of Plants*. IEEE Computer Graphics and Applications, Volume 19, pages 56 – 65, January 1999.
- [9] J. L. Power, A.J. B. Brush, P. Prusinkiewicz, and D. H. Salesin. *Interactive arrangement of botanical L-system models*. Proceedings of the 1999 symposium on Interactive 3D graphics, pages 175 – 182, April 1999.
- [10] P. Prusinkiewicz, A. Lindenmayer, and J. 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'88, Volume 22, Issue 4, pages 141 – 150, June 1988.
- [11] P. Prusinkiewicz, M. S. Hammel, and E. Mjolsness. *Animation of plant development*. Proceedings of the 20th annual conference on Computer graphics and interactive techniques, pages 351 – 360, September 1993.

- [12] P. Prusinkiewicz, L. M<sup>u</sup>ndermann, R. Karwowski, and B. Lane. *The use of positional information in the modeling of plants*. Proceedings of the 28th annual conference on Computer graphics and interactive techniques, pages 289 – 300, August 2001.
- [13] W. T. Reeves and R. 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, Volume 19, Issue 3, pages 313 – 322, July 1985.
- [14] M. Renton, J. Hanan, and P. Kaitaniemi. *The inside story: including physiology in structural plant models*. Proceedings of the 1st international conference on Computer graphics and interactive techniques in Australasia and South East Asia, pages 95 – ff, February 2003.
- [15] A. R. Smith. *Plants, fractals, and formal languages*. ACM SIGGRAPH Computer Graphics. Proceedings of the 11th annual conference on Computer graphics and interactive techniques, Volume 18, Issue 3, pages 1 – 10, January 1984.
- [16] L. Wang, W. Wang, J. Dorsey, X. Yang, B. Guo, and H. Y. Shum. *Real-time rendering of plant leaves*. ACM Transactions on Graphics (TOG), Volume 24 Issue 3, pages 712 – 719, July 2005.
- [17] A. Watt. *3D Computer Graphics*. Third Edition, Addison Wesley, U.S.A., 2000.
- [18] Granieri, John, et al, *Production and Playback of Human Figure Motion for Visual Simulation*. Modeling and Computer Simulation, 5(3):222-241, 1995.
- [19] Hutchinson, Dave, et al, *Adaptative Refinement for Mass-Spring Simulations*. Eurographics Workshop on Animation and Simulation, 1997.

## 附錄 玫瑰的生長規則

Grow {  
w = A(0);  
A(0) -> S(2) ;  
S(2) -> S(100) [rotr(1.078734)rotu(-2.678500)S(15)]rotr(0.168208)rotu(0.000000)S(3) ;  
S(3) -> S(100) [rotr(0.968645)rotu(0.413011)S(16)]rotr(0.254051)rotu(2.740428)S(4) ;  
S(4) -> S(100) [rotr(0.80000)rotu(1.00000)S(18)] rotr(0.199278)rotu(-3.027257)S(5) ;  
  
S(5) -> S(100) [rotr(1.328597)rotu(0.596998)S(20)] rotr(0.095898)rotu(-2.481976)S(6) ;  
S(6) -> S(100) [rotr(0.851592)rotu(1.077676)S(23)] rotr(0.360359)rotu(-1.494222)S(7) ;  
S(7) -> S(100) [rotr(0.875823)rotu(0.895065)S(26)] rotr(0.374151)rotu(-2.917036)S(8) ;  
S(8) -> S(100) [rotr(1.163584)rotu(0.569395)S(29)] rotr(0.517579)rotu(-2.164740)S(9) ;  
S(9) -> S(100) A(10) ;  
A(10) -> rotr(0.131326)rotu(2.957337)S(10) ;  
S(10) -> S(100) [rotr(1.286899)rotu(-2.640274)S(32)] [rotr(0.781501)rotu(-2.525453)S(33)]  
rotr(0.185149)rotu(-0.248169)S(11) ;  
S(11) -> S(100) [rotr(0.575201)rotu(1.566550)S(38)] [rotr(1.121019)rotu(1.406090)S(40)]  
rotr(0.176646)rotu(-1.533860)S(12) ;  
S(12) -> S(100) [rotr(0.521141)rotu(0.433825)S(41)] rotr(0.336766)rotu(-3.016415)S(13) ;  
S(13) -> S(100) A(11) ;  
A(11) -> S(100) F(0) ;  
  
S(15) -> S(200) [rotr(1.417282)rotu(1.4)S(1)L(0)] [rotr(1.464461)rotu(-1.4)S(1)L(0)]  
rotr(0.000000)rotu(-2.484141)S(1)L(0) ;  
  
S(16) -> S(200) [rotr(1.417282)rotu(1.4)S(1)L(0)] [rotr(1.464461)rotu(-1.4)S(1)L(0)]  
rotr(0.488999)rotu(-1.667325)S(17) ;  
S(17) -> S(200) [rotr(1.417282)rotu(0.0)S(1)L(0)] [rotr(1.464461)rotu(3.14)S(1)L(0)]  
rotr(0.000000)rotu(-2.539291)S(1)L(0) ;  
  
S(18) -> S(200) [rotr(1.417282)rotu(1.4)S(1)L(0)] [rotr(1.464461)rotu(-1.4)S(1)L(0)]  
rotr(0.519472)rotu(-1.551973)S(19) ;  
S(19) -> S(200) [rotr(1.417282)rotu(0.0)S(1)L(0)] [rotr(1.464461)rotu(3.14)S(1)L(0)]  
rotr(0.000000)rotu(-1.978830)S(1)L(0) ;

S(20) -> S(200) [rotr(1.417282)rotu(1.0)S(1)L(0)] [rotr(1.464461)rotu(4.14)S(1)L(0)]  
 rotr(0.632549)rotu(-2.387246)S(21) ;

S(21) -> S(200) [rotr(1.417282)rotu(0.0)S(1)L(0)] [rotr(1.464461)rotu(3.14)S(1)L(0)]  
 rotr(0.263607)rotu(1.978250)S(22) ;

S(22) -> S(200) [rotr(1.417282)rotu(1.4)S(1)L(0)] [rotr(1.464461)rotu(-1.4)S(1)L(0)]  
 rotr(0.000000)rotu(1.882501)S(1)L(0) ;

S(23) -> S(200) [rotr(1.417282)rotu(1.4)S(1)L(0)] [rotr(1.464461)rotu(-1.4)S(1)L(0)]  
 rotr(0.340223)rotu(-1.997677)S(24) ;

S(24) -> S(200) [rotr(1.417282)rotu(-0.3)S(1)L(0)] [rotr(1.464461)rotu(2.8)S(1)L(0)]  
 rotr(0.725477)rotu(0.063823)S(25) ;

S(25) -> S(200) [rotr(1.417282)rotu(0.0)S(1)L(0)] [rotr(1.464461)rotu(3.14)S(1)L(0)]  
 rotr(0.000000)rotu(0.914290)S(1)L(0) ;

S(26) -> S(200) [rotr(1.417282)rotu(1.4)S(1)L(0)] [rotr(1.464461)rotu(-1.4)S(1)L(0)]  
 rotr(0.163021)rotu(-1.223614)S(27) ;

S(27) -> S(200) [rotr(1.417282)rotu(0.0)S(1)L(0)] [rotr(1.464461)rotu(3.14)S(1)L(0)]  
 rotr(0.258636)rotu(-0.178403)S(28) ;

S(28) -> S(200) [rotr(1.417282)rotu(0.0)S(1)L(0)] [rotr(1.464461)rotu(3.14)S(1)L(0)]  
 rotr(0.000000)rotu(2.628339)S(1)L(0) ;

S(29) -> S(200) [rotr(1.417282)rotu(1.4)S(1)L(0)] [rotr(1.464461)rotu(-1.4)S(1)L(0)]  
 rotr(0.170239)rotu(-1.269727)S(30) ;

S(30) -> S(200) [rotr(1.417282)rotu(0.0)S(1)L(0)] [rotr(1.464461)rotu(3.14)S(1)L(0)]  
 rotr(0.240432)rotu(1.351168)S(31) ;

S(31) -> S(200) [rotr(1.417282)rotu(1.4)S(1)L(0)] [rotr(1.464461)rotu(-1.4)S(1)L(0)]  
 rotr(0.000000)rotu(-2.331613)S(1)L(0) ;

S(32) -> S(200) [rotr(1.417282)rotu(1.4)S(1)L(0)] [rotr(1.464461)rotu(-1.4)S(1)L(0)]  
 rotr(0.000000)rotu(-2.011859)S(1)L(0) ;

S(33) -> S(200) [rotr(1.816122)rotu(2.0)S(47)] rotr(0.211526)rotu(-1.330530)S(37) ;

S(37) -> S(200) [rotr(1.047198)rotu(0.0)S(1)L(0)] rotr(0.074787)rotu(0.155047)S(34) ;

S(34) -> S(200) [rotr(0.671015)rotu(1.895298)S(48)] rotr(0.060269)rotu(-1.932377)S(35) ;

S(35) -> S(500) A(12) ;

A(12) -> rotr(0.116573)rotu(-2.984062)F(0) ;

S(38) -> S(200) [rotr(0.762545)rotu(2.268994)S(45)] rotr(0.225475)rotu(2.559772)S(39) ;  
 S(39) -> S(200) A(13) ;  
 A(13) -> rotr(0.0)rotu(0.0)S(200)[rotr(1.396263)rotu(2.386454)S(1)L(5)]rotr(0.0)rotu(0.0)S(36);

S(40) -> S(200) [rotr(1.417282)rotu(1.4)S(1)L(5)] [rotr(1.464461)rotu(-1.4)S(1)L(5)]  
 rotr(0.000000)rotu(-1.031213)S(1)L(4) ;

S(41) -> S(500) A(14) ;  
 A(14) -> rotr(0.239182)rotu(-1.590040)S(43) ;  
 S(43) -> S(500)[rotr(1.064597)rotu(-0.2)S(1)L(5)] rotr(0.02)rotu(-1.412020)S(44) ;  
 S(44) -> S(500)A(20) ;  
 A(20) -> S(500) [rotr(1.063686)rotu(3.4)S(1)L(5)] rotr(0.0)rotu(0.0)F(0);

S(45) -> S(300) A(15) ;  
 A(15) -> rotr(0.072193)rotu(-2.643216)S(46) ;  
 S(46) -> S(300) [rotr(1.417282)rotu(0.0)S(1)L(5)]  
 [rotr(1.464461)rotu(3.14)S(1)L(5)]rotr(0.500000)rotu(1.629560)S(50);  
 S(50) -> S(1)L(4);

S(47) -> S(300) [rotr(1.417282)rotu(0.5)S(1)L(6)] [rotr(1.464461)rotu(3.54)S(1)L(6)]  
 rotr(0.000000)rotu(2.982925)S(1)L(6) ;

S(48) -> S(300) A(16) ;  
 A(16) -> S(300)[rotr(0.698132)rotu(-1.018114)S(1)L(3)]rotr(0.0)rotu(0.0)S(300) ;

S(100) -> S(101) ;  
 S(101) -> S(102) ;  
 S(102) -> S(103) ;  
 S(103) -> S(104) ;  
 S(104) -> S(105) ;  
 S(105) -> S(106) ;

S(200) -> S(201) ;  
 S(201) -> S(202) ;

S(300) -> S(301) ;



L(0) -> L(1) ;  
L(1) -> L(2) ;  
L(2) -> L(3) ;  
L(3) -> L(4) ;  
L(4) -> L(5) ;  
L(5) -> L(6) ;  
L(6) -> L(7) ;

F(0) -> F(1);  
F(1) -> F(2);  
F(2) -> F(3);

}