

國立政治大學資訊科學研究所

Department of Computer Science

National Chengchi University

碩士論文

Master's Thesis

基於 L-System 之動態模擬東方風格立體雲紋生成技術
L-system base Dynamic Simulation for 3D Oriental Cloud Pattern

研究生：郭明諺

指導教授：紀明德

中華民國一百零一年八月

August 2012

基於 L-system 之動態模擬東方風格立體雲紋生成技術

L-system base Dynamic Simulation for 3D Oriental
Cloud Pattern

研究生：郭明諺

Student：Ming-Yen Kuo

指導教授：紀明德

Advisor：Ming-Te Chi



中華民國一百零一年八月

August 2012

致謝

進入政大將近兩年的時間，日子轉眼即逝，在這期間我非常由衷的感謝我的指導老師紀明德老師，從碩一剛入學時對於許多課業與研究方面皆遇到許多困難，而老師一直很耐心地指引我方向，隨著時間慢慢地過去，也逐漸進入狀況。這期間我非常謝謝我的父母、弟弟等全家人，每每週末回家都放鬆的休息，沒有太多時間陪伴你們，謝謝你們的關心與體諒。研究生的生活平常雖然乏味，日子卻相當充實，有實驗室的學長、學弟妹以及夥伴們的陪伴之下，生活當中也有許多不同的一面，謝謝阿三啟發了我另一個運動人生、謝謝小炫炫對我的關心與照顧、謝謝章魚哥幫我 debug，有問必答、謝謝壯壯阿轟管理整個實驗室、謝謝小正給我的新年快樂電話、謝謝嗣心、以及 bb 與夫夫兩位和藹可親的學弟，真的為整間實驗室帶來許多歡樂、謝謝細心的欣瑤、還有承峰及漢光，謝謝你們總是為實驗室製造許多話題與歡樂氣氛、還有從大學時就陪伴著我的雅玲，總是給予我許多幫助、還有岱珊學姊，一起運動健身很充實、還有我身邊的好朋友們，思瑩、香香、小柯、阿寧、黑妹，我愛妳們，友誼長存；感謝凱彬，謝謝你的貼心並且給予我信心與鼓勵，有你的陪伴真好。在此將我所完成的這篇論文這份喜悅分享給大家，也同時獻給在天上的奶奶，謝謝。

基於 L-System 之動態模擬東方風格立體雲紋生成技術

摘要

本篇論文對於傳統的東方風格雲紋進行分析，歸納其中的生成規則與排列特性，並將雲紋圖樣作簡單的分類，發現雲紋圖樣具有自我相似與遞迴組成的特性，此點特性非常符合原本用於模擬植物生長的 L-system，所以我們就針對所蒐集的眾多雲紋圖樣參考資料所分析出的資訊定義了一個適用於立體東方風格雲紋的生成生長規則，以 L-system 之技術為基礎進行撰寫規則語法，利用所定義結構化之規則循序地在三維空間中動態模擬雲紋的生長與變化。更進一步地，加入具時間序列的參數調整，可以使得雲紋隨著時間軸的變化產生東方風格立體動態雲紋。

L-system base Dynamic Simulation for 3D Oriental Cloud Pattern

Abstract

Cloud Pattern is an important and common element in oriental decoration art, it is a challenge to extend the 2D abstract line into 3D objects. In this thesis, we try to induce the pattern generation rule and the feature of oriental cloud by analyzing the ancient oriental cloud pattern. We make classification from the oriental 2D cloud pattern, and discover the pattern is combined with symmetric, self-similar and repetitive features. These characteristic are the core of L-System, which is a suitable framework for plant growth and simulation, so we generate a new rule with specific parameter from the feature of 2D cloud pattern to generate oriental cloud pattern in 3-dimension space, simulating the growth of cloud and variety in structure. Further, we add the time temporal parameter to control the 3D oriental cloud pattern generation process dynamically.

目錄

摘要.....	i
Abstract.....	ii
目錄.....	iii
圖目錄.....	v
表目錄.....	viii
第一章 緒論.....	1
1.1 研究動機與目的.....	1
1.2 問題描述.....	2
1.3 論文貢獻.....	2
1.4 論文章節架構.....	3
第二章 相關研究.....	4
2.1 東方風格雲紋.....	4
2.2 L-System 相關研究.....	5
2.3 NPR.....	8
第三章 東方風格雲紋分析.....	9
3.1 雲的分析.....	9
3.2 東方風格雲紋分析.....	12
3.2.1 雲紋圖樣結構分類.....	14
3.2.2 雲尾結構.....	16
3.3 立體雲紋結構分析.....	18
第四章 研究方法與步驟.....	21
4.1 研究步驟.....	21
4.2 Parametric L-System.....	24

4.2.1 雲頭 L-system 語法.....	27
4.2.2 雲尾 L-system 語法.....	40
4.2.3 雲紋結構.....	44
4.2.4 立體雲紋 L-system 語法架構.....	47
4.3 Dynamic System.....	54
4.4 風格化.....	55
第五章 實驗結果與討論.....	57
5.1 實作與實驗環境.....	57
5.2 東方風格雲紋結果與分析.....	57
5.3 動態雲紋.....	66
5.4 效能評估.....	69
5.5 系統限制.....	70
第六章 結論與未來工作.....	71
6.1 結論.....	71
6.2 未來工作.....	72
參考文獻.....	73
附錄 A：基於 L-system 之東方風格立體雲紋語法規則.....	75

圖目錄

圖 2.1 東方風格雲紋的代表種類。[1].....	4
圖 2.2 東方風格雲紋-朵雲紋相關研究。[2].....	5
圖 2.3 東方風格雲紋-如意雲紋相關研究。[3].....	5
圖 2.4 L-system 依語法定義生長示意圖.....	6
圖 2.5 藉由筆觸為輸入資訊搭配 L-system 語法模擬植物.....	7
圖 2.6 植物生長在時間序列下之動態模擬結果呈現 [16].....	7
圖 2.7 以不同基本幾何元素模擬之立體雲朵合體。[9].....	8
圖 2.8 以不同平面圖樣所繪製模擬出之各式煙霧結果。[10].....	8
圖 3.1 真實雲紋與東方風格雲紋圖樣比較	9
圖 3.2 各樣式的東方風格雲紋.....	10
圖 3.3 夢工廠電腦動畫「功夫熊貓 2」開場片段截圖.....	11
圖 3.4 傳統平面雲紋圖樣與三維空間中的立體雲紋及風格化後之結果.....	11
圖 3.5 雲紋結構分析圖.....	13
圖 3.6 以平面雲紋圖樣為基礎模擬其立體雲紋之示意圖.....	13
圖 3.7 雲紋圖樣分類.....	14
圖 3.8 雲尾末端。[2].....	16
圖 3.9 雲尾與雲朵之結構分類。.....	17
圖 3.10 立體雲紋於雕刻品上的應用.....	18
圖 3.11 模擬平面雲紋與立體雲紋的符號對應.....	19
圖 4.1 系統流程圖.....	22
圖 4.2 L-system 立體雲紋生成規則流程圖.....	22
圖 4.3 Turtle interpretation of L-system 之方向軸	24
圖 4.4 朵雲紋 C1-A 生長規則流程圖.....	28

圖 4.5 朵雲紋 C1-A 幾何結構生成模擬步驟.....	29
圖 4.6 朵雲紋 C1-A 系統生成之逐步結果.....	29
圖 4.7 朵雲紋 C1-B 生長規則流程圖.....	31
圖 4.8 朵雲紋 C1-B 幾何結構生成模擬步驟.....	31
圖 4.9 朵雲紋 C1-B 系統生成之逐步結果.....	31
圖 4.10 朵雲紋 C2 生長規則流程圖.....	33
圖 4.11 朵雲紋 C2 幾何結構生成模擬步驟.....	33
圖 4.12 朵雲紋 C2 系統生成之逐步結果.....	33
圖 4.13 朵雲紋 C3 生長規則流程圖.....	35
圖 4.14 朵雲紋 C3 幾何結構生成模擬步驟.....	35
圖 4.15 朵雲紋 C3 系統生成之逐步結果.....	36
圖 4.16 朵雲紋 C4 生長規則流程圖.....	37
圖 4.17 朵雲紋 C4 幾何結構生成模擬步驟.....	38
圖 4.18 朵雲紋 C4 系統生成之逐步結果.....	38
圖 4.19 尾部-分離雲尾生成流程圖.....	40
圖 4.20 尾部-分離雲尾生成流程圖.....	41
圖 4.21 尾部-密合雲尾生成流程圖.....	42
圖 4.22 尾部-密合雲尾生成流程圖.....	42
圖 4.23 將傳統雲紋圖樣之分類套用至 L-system 語法結構示意圖.....	47
圖 4.24 結構生成雲紋圖樣 Example 1 之結果圖.....	50
圖 4.25 結構生成雲紋圖樣 Example 2 之結果圖.....	51
圖 4.26 結構生成雲紋圖樣 Example 3 之結果圖.....	51
圖 4.27 結構生成雲紋圖樣 Example 4 之結果圖.....	52
圖 4.28 結構生成雲紋圖樣 Example 5 之結果圖.....	53
圖 4.29 Prusinkiewicz[7]之 L-system 5 規則.....	54

圖 4.30 以不同色調 shading 之立體雲紋結果	55
圖 4.31 經過影像處理過後之結果比較.....	56
圖 5.1 朵雲紋 C1-A 搭配不同雲尾結果呈現.....	61
圖 5.2 朵雲紋 C4 搭配尾部-分離之雲尾與風格化之結果.....	61
圖 5.3 朵雲紋 C1-A 與 C3 利用尾部-密合雲尾組合與風格化之結果.....	62
圖 5.4 以疊代式的 L-system 語法組合出不同類型的立體雲紋結果.....	64
圖 5.5 以平面結構 SYM2 逐步生成的動態截圖結果.....	66
圖 5.6 SYM1-C4 與 SYM4-C1-A 平面結構之組合式雲紋系統動態截圖結果..	67
圖 5.7 SYM8-C1-A 與 SYM4-C4 立體結構之組合式雲紋動態截圖結果	68



表目錄

表 1 雲紋圖樣整理表。[2].....	12
表 2 朵雲紋、雲尾的平面轉換立體之模擬.....	20
表 3 L-system 符號與生長動作對應.....	25
表 4 朵雲紋平面雲紋圖樣編號.....	27
表 5 立體雲紋結構表.....	45
表 6 朵雲紋結果分析與比較.....	58
表 7 以不同遞迴次數與旋轉角度之組合生成之雲紋末端捲曲組合變化.....	59
表 8 以不同風格的 Cylinder 樣貌搭配不同旋轉角度模擬之立體雲紋結果.....	60
表 9 以多次遞迴與角度的變化模擬出各式不同的捲曲立體雲紋.....	60
表 10 多種不同結構之立體雲紋結果.....	62
表 11 平面雲紋圖樣與系統生成之立體雲紋整理表.....	65
表 12 實際測試系統做動態呈現時之效能比較分析.....	69

第一章

緒論

1.1 研究動機與目的

近日來，動畫的繪圖技巧越來越趨近成熟，動畫主題的取材也越來越多元化，以動畫的故事內容背景大致可分為歐美等西方的主題或是帶有東方色彩的改編故事，無論是何種故事背景，動畫之中除了主角之外，其中有許多的背景裝飾更是襯托著以利場面更為豐富與多元，以東方文化為故事主軸背景為例，裝飾紋路在整體動畫的使用也極為普遍，最常以陪襯角度出現的「雲紋」就是一個最顯著的實例，不管是白天還是黑夜，都可加上雲紋的搭配，增加畫面的流動感，所以如果可以把傳統雲紋動態化地變化與生長或許是另一種特殊的表現方式。

在傳統中國東方風格之古蹟文物、器皿、水墨畫等許多作品都可以發現具有相當大量的雲紋裝飾藝術，從各個歷史朝代的更迭演變，也可以發現雲紋也有著其相對應的演變過程，在這些傳統的平面雲紋為基礎，想像一下如果可以將這些平面紋路轉換為立體的呈現，或許會有更不一樣的效果，本篇論文將針對各樣式的東方風格雲紋，找出其規則性，將之轉換為以 L-system 為基礎的語法作結構化地生成，並搭配動態地變化模擬，則可將原先只是在背景單純飄動的雲紋圖樣轉換為更具真實感團聚與消散的立體雲紋。

1.2 問題描述

本篇論文參考了許多的平面雲紋圖樣，發現看似沒有規則的雲紋也具有重複性高且自我相似的特性，此點特性恰好與現今用於模擬根莖花葉植物的技術 L-system 相似，以此為我們的相關研究背景，要將雲紋的結構轉換為許多的規則參數表達與生成，利用 L-system 高擴充性的語法將平面雲紋轉換為三維空間當中立體化動態雲紋的生成，在原先 L-system 之應用往往皆應用在植物的生長，所以該如何設置參數使得 L-system 在此做為雲紋的生長規則，以及在系統當中如何將雲紋自然地隨時間生長與變化並搭配時間軸 step 的參數進入系統控制其動態的節奏性，使得立體雲紋會逐步慢慢地生長變化或是經由外力，例如風的吹動，而使雲紋產生相對應動態的結果。

1.3 論文貢獻

在本研究論文當中，我們蒐集了許多具有東方風格的 2D 雲紋圖樣，且在二維平面的圖樣上進行分析，將其所對應的 3D 立體雲紋藉由先前 L-system 的技術為基礎，我們提出一個的新的演算法及應用，本研究的方向主要可以分為以下兩大部分，各部分的內容簡單敘述如下：

- 將大量東方風格雲紋作分析並分類，找出其排列與組合之方法。
- 將雲紋以階層及結構化的方式轉換成為 L-system 疊代的架構。
- 實作出 L-system 之語法規則將雲紋之生長規則系統化，使用者只要選擇不同參數則可以順利產生立體的東方風格動態雲紋。

1.4 論文章節架構

在第二章中，我們將會介紹與本研究相關的研究背景知識。在第三章為東方風格雲紋的分析與歸類。在第四章當中則是主要的研究方法與步驟。在第五章則是實驗結果的呈現與分析比較討論。第六章是結論與未來研究。



第二章

相關研究

2.1 東方風格雲紋

徐麗慧與郑军等人[1]整理出許多在中國各個朝代當中雲紋藝術的發展與歷史，觀察這些紋飾的變化也可漸漸地了解在當時代背景的社會風氣習俗，在這些資料當中所觀察到的皆為 2D 平面之裝飾紋路，而現今是否有方法可以產生具有東方風格雲紋之「立體」、「動態」裝飾雲紋？如圖 2.1。另一方面，詹毓君[2]提出了使用 sketched base 的方法讓使用者可以針對雲頭及雲尾的簡單手寫描繪當作輸入，再利用這 stroke 作生成，產生相對應的東方風格雲紋圖樣，如圖 2.2。陳奕愷[3]則是提出以 L-system 為其雲紋生成技術之基礎，利用不同參數的組合生成許多如意雲紋的圖樣，如圖 2.3。

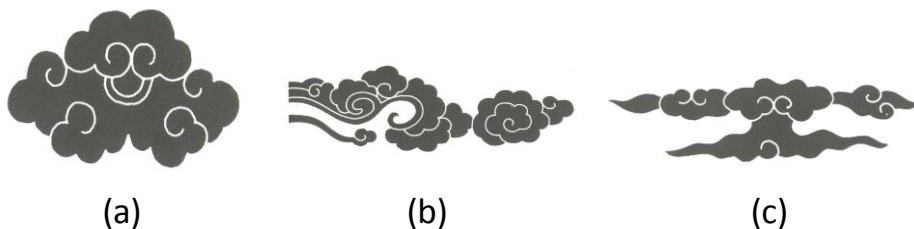


圖 2.1 東方風格雲紋的代表種類(a)如意雲紋 (b)風雲 (c)工字雲。[1]

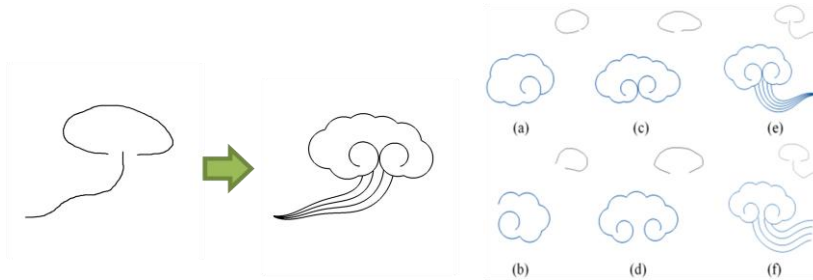


圖 2.2 東方風格雲紋-朵雲紋相關研究。[2]



圖 2.3 東方風格雲紋-如意雲紋相關研究。[3]

2.2 L-System 相關研究

L-system 為一個具有自我遞迴的生長規則，在一開始會給定其一初始值稱之 Axiom，而再根據所定義之生長規則去作相對應的生長。L-system 在最早時期被提出主要是用於模擬簡單的單細胞生物，隨著時間的發展，在約略 90 年代時由 Prusinkiewicz 提出將 L-system 應用於植物生長及模擬[5][6]，隨後 L-system 就常常與植物生長畫上等號。另外 Stelios [16]有詳細地介紹到對於 L-system 之分類，總共可以做以下幾種分類：Context-free(OL) or Context-sensitive(IL)、Deterministic(DL) or Non-deterministic(stochastic)、Bracketed L-systems、Propagative L-systems(PL) or Non-Propagative L-systems、Table L-systems(TL)、Parametric L-systems、L-systems with extension (EL)。在此簡單的以最基本型之 L-system，稱作 context-free L-system 介紹一下其規則與定義，如圖 2.4 所示，其生長規則為左邊從 **bud** → **flower**、**flower** → **young fruit**、**young fruit** → **old fruit** 等四種，而右邊則為其植物的生長變化，類似像這樣的迭代規則如果

把圖 2.4 利用字母取代符號來表示，則會產生出顯得更具系統化的規則方法，而對於我們所要使用於生長東方風格雲紋之規則在這裡所要採用的類型為 Parametric L-system，其原因在於要生成出立體的東方風格雲紋往往在規則的迭代過程當中其使用的參數數值必須亦隨之改變，也必須根據規則當中條件的判定以決定下一步的流程，前面的幾類 L-system 規則往往無法做到這點，所以在 L-system 方法的類型上我們會採取 Parametric L-system。

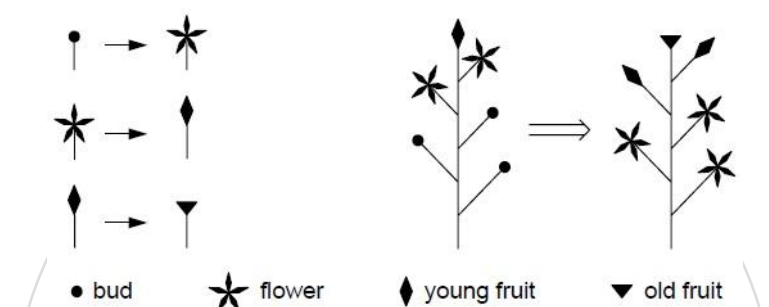


圖 2.4 L-system 依語法定義生長示意圖

Prusinkiewicz[7]也將 Parametric L-system 視為一種 programming language，其格式如下：

Predecessor : *condition* → *successor*

由初始的 predecessor 開始，根據參數與 condition 的定義，產生相對應 successor 之生成結果，就是利用這參數化的生成原理，將之應用於我們的雲紋生長規則。

另一方面，在[14]中也提出了另一種基於使用者可直接將手畫的 strokes 當作輸入，系統則會針對所輸入的資訊做分析，將參數傳遞至系統作為 L-system 規則的生長語法之中，從圖 2.5 可以看得出來，使用者只要輸入 boundary, stem and layer 等概略的筆觸資訊，系統即可將這些筆觸的資訊分析出來作為 L-system 之生長

語法中的參數，並搭配植物相對應的模型檔，最後將其資訊合併後，結果如圖 2.5 最右方的模擬植物圖，從整體的系統流程圖來看，可以想像成是 L-system 與 sketch based 的結合，我們必須將平面雲紋中的參數取出，應用至本篇論文當中的雲紋 L-system 生長規則之中。

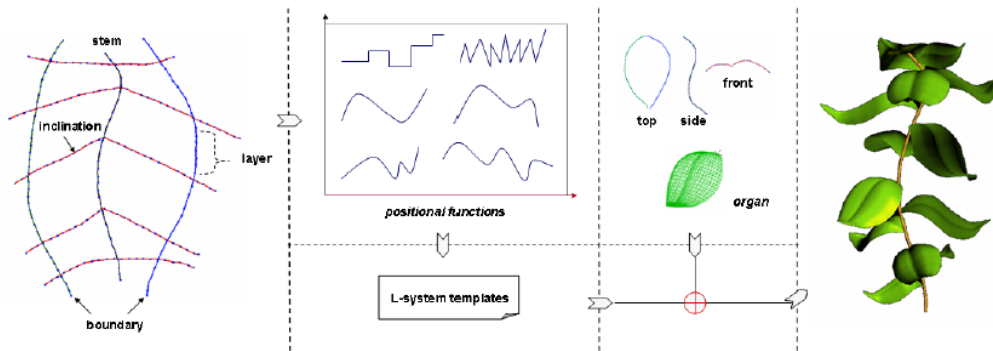


圖 2.5 藉由筆觸為輸入資訊搭配 L-system 語法模擬植物

另一方面，植物與樹木的模擬其實是一個具有緩慢變化的動態式生長，以及會因外力的介入而間接地影響到植物的動態擺動，東方風格雲紋的模擬也可以以此為參考，在[16]中除了利用 Parametric L-system 的語法模擬植物生長，也針對植物動態的生長做了很清楚的呈現，根據系統模擬的時間序列將結果逐步地將動態的感覺呈現出來，如圖 2.6，在實作雲紋動態模擬的部分也會參考到此篇論文。

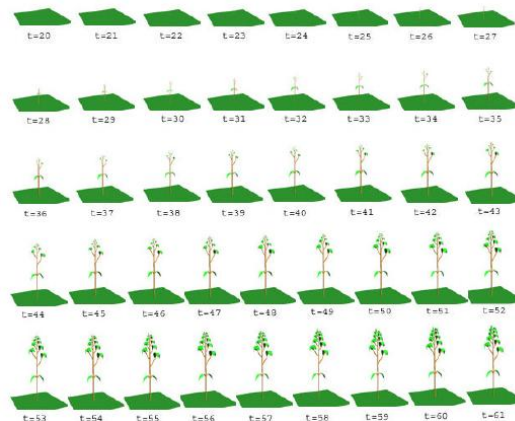


圖 2.6 植物生長在時間序列下之動態模擬結果呈現 [16]

2.3 NPR

在電腦圖學當中，對於非相片寫實類物體之繪圖技術稱為 Non-Photorealistic Rendering(NPR) ，其中對於 3D 物體 Rendering 的研究更是不勝枚舉，例如 Hertzmann[7]提出在空間中求出物體的 Depth Map 與 Normal Map，利用這些資訊則可簡單繪製出 NPR 效果之 Silhouettes 和 Outlines。另外，McGuire 將 billboards 之概念用於 Particle 的 Rendering，成功的將具有卡通風格的雲繪製出來[8]，本篇也將會參考其研究技術，找出更具有風格化的東方風格雲紋繪製。除了上述的研究之外，Selle 則是加上不同雲紋圖樣模板樣式的繪製[9]，使得在卡通風格之雲或煙霧模擬變得更為多元化，利用此一概念，在繪製立體雲紋時可以利用不同的 shading 風格，增加更為多變的風格。

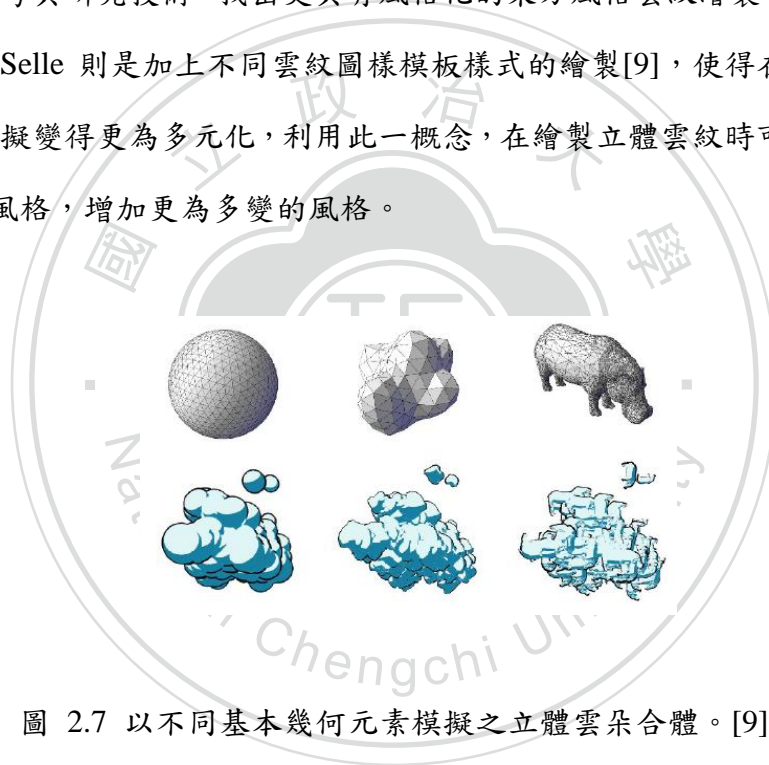


圖 2.7 以不同基本幾何元素模擬之立體雲朵合體。[9]

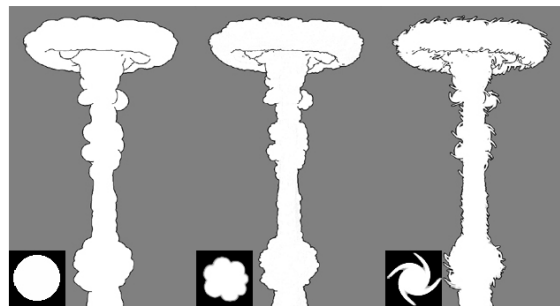


圖 2.8 以不同平面圖樣所繪製模擬出之各式煙霧結果。[10]

第三章

東方風格雲紋分析

3.1 雲的分析

在真實世界中的雲紋與傳統的東方風格雲紋，兩者實際上所呈現出的感覺截然不同，如圖 3.1(a)所示，我們日常生活當中所看到的自然界雲紋，古代的人們就是觀察這些自然界的紋路將之加入當時的文化背景與藝術風格勾勒出這些紋路，常常在許多古文物、書畫、古蹟文物等等上面出現，而圖 3.1(b)是傳統的東方風格雲紋圖樣，較傾向於將雲的形體利用線條勾勒出雲的流動感，在《中國歷代雲紋紋飾藝術》中提及到雲紋在中國歷史的演變上可以充分的表現出當時代的風俗民情與審美觀的演進，許多的雲紋裝飾藝術也可以從這些紋路當中得到啟發。

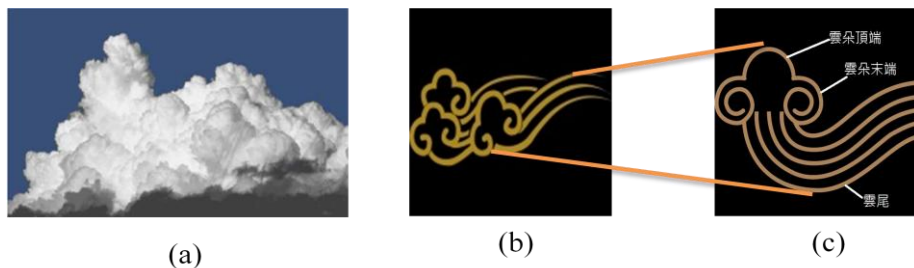


圖 3.1 真實雲紋與東方風格雲紋圖樣比較
(a)自然界雲朵 (b)傳統東方風格雲紋 (c)以(b)之局部放大

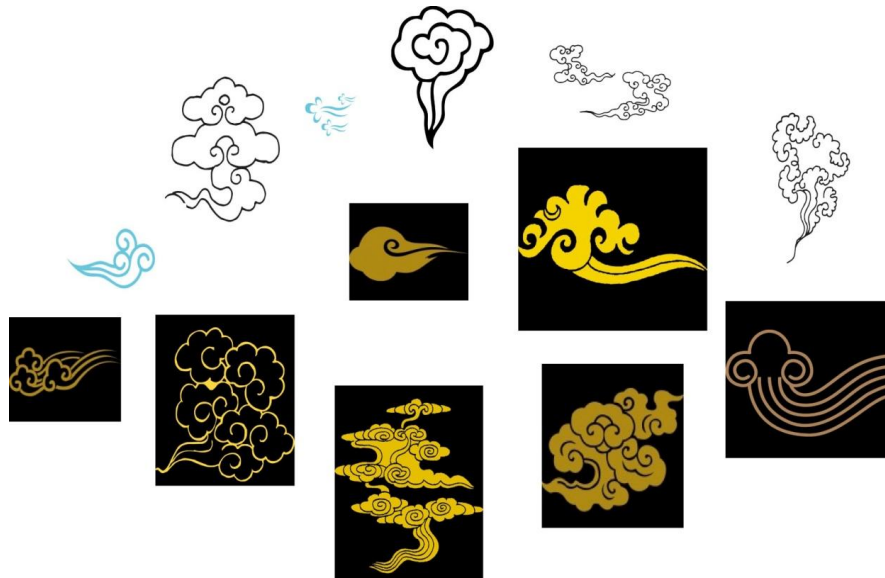


圖 3.2 各樣式的東方風格雲紋

觀察如圖 3.2 這些傳統雲紋圖樣，如果仔細觀察可以發現在這些不同圖樣當中也存在著一些相同結構所組成，如圖 3.1(b)所示，從其結構面來看，雲紋圖樣皆可分為「雲朵頂端」、「雲朵末端」、「雲尾」這些部分的結構組合[2][3]，藉由雲頭的團聚及雲尾的走向而感覺出雲的型體以及雲的飄逸方向，在這樣的出發點下，從許多電影動畫中可以到許多具有雲紋背景裝飾的片段，例如在迪士尼電腦動畫「花木蘭」片頭的山水畫風動態雲紋以及夢工廠電腦動畫「功夫熊貓 2」開場片頭的簡介片段也帶有相當多的剪紙風動態雲紋作為裝飾，如圖 3.3，這些動態的裝飾雲紋在影片當中雖然未帶有劇情的成分，但往往在鋪陳許多的關鍵劇情帶有若隱若現的驚喜感，想像一下，如果這些動態雲紋可以在三維空間以立體方式隨著時間變化作生長與消散，所呈現出來的結果想必更為豐富與多元。簡單的從原本的平面雲紋與立體雲紋的對應下，如圖 3.4，我們可以簡單做個比較，平面雲紋的呈現較為單純，利用線條的方式表現出雲頭及雲尾，立體雲紋的表現方式可藉由深度值的變化呈現出立體感，以及經過風格化後之處理其結果顯得更為多元。另一方面，人的注意力往往對於會動的東西較為關注，立體雲紋如果可以搭配系統「動態」且經由時間的變化而改變立體雲紋的呈現，會動態成長的東方風格雲紋圖樣又是另一種特別的裝飾藝術。



圖 3.3 夢工廠電腦動畫「功夫熊貓 2」開場片段截圖

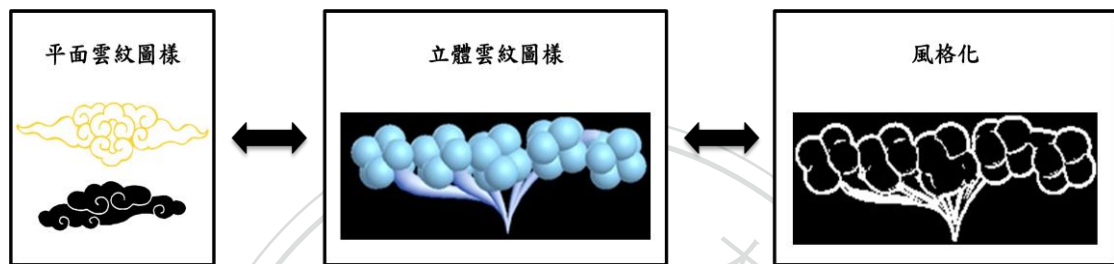





圖 3.4 傳統平面雲紋圖樣與三維空間中的立體雲紋及風格化後之結果

思考到東方風格雲紋圖樣的設計方法，許多使用者或是藝術家們往往只能依照所搜尋到的參考資料，以純手工的方式進行描繪或是設計其結構再使用繪圖軟體繪製出二維平面的裝飾雲紋結果，如今我們想要創造出一套具有規則性的生長結構，讓使用者在想要使用東方風格雲紋的圖樣時，可以藉由本研究所提出的方法很簡單地生成出想要的雲紋圖樣，而在介紹本研究方法之前，我們必須針對所謂的「東方風格雲紋」做個初步的分析。

3.2 東方風格雲紋分析

詹毓君[2]將雲紋依照中國古歷史朝代的演變根據時間的變化上，很清楚地呈現出雲紋在歷史上的演變，而其所研究的重點在對於「朵雲紋」的結構分析下，可以將其想像成數個圓形所堆疊而成，另外陳奕愷[3]清楚地描述到針對如意雲紋的 L-system 生長規則，如果我們仔細觀察這些參考雲紋圖樣，常常具有外部結構不盡相同但內部結構卻具有自我相似的特性，而此特性就相當適合搭配 L-system 的語法進行生長。

表 1 雲紋圖樣整理表(擷取自[2])

分類時期	圖例	雲紋總類
隋唐時期		朵雲紋
宋元時期		團雲紋
明清時期		如意雲紋 朵雲紋

我們可以根據上一小節圖 3.1(b)，可以發現東方風格雲紋在每張不相同的雲紋圖樣中都可簡單分成「雲朵」與「雲尾」這樣的結構，而在較為複雜一點的雲紋圖樣當中，如圖 3.5 又可以進一步分為「如意雲紋」、「朵雲紋」、「團雲紋」、「雲尾」四大部分，為各個不同雲紋圖樣所組合而成之多重雲紋圖樣，而其中較為特別的則是「團雲紋」的結構較類似於如意雲紋與朵雲紋的綜合體，簡單來說，雲紋本身的重複性其實很高，所以我們可以利用此特性，在之後的主要生長方法中

將一層一層的結構定義好，一開始像是朵雲紋雲頭的部分即藉由生成基本單元球體利用重複遞迴疊代的方法推疊而成，可以參考圖 3.6 (a)、(b)，接下來則可以藉由將原本基本的朵雲紋加以擴充，生長出較大朵的朵雲紋，之後多重雲紋圖樣之結構則可以藉由這些不同的朵雲紋結果加入結構當中組合建構出來，結構的疊代可以生長出較為複雜的雲紋圖樣，而結構本身也可以再加以組合疊代，藉由這樣一層一層的建構方式可以逐步地建構出多變化的東方風格雲紋，實作方法就是利用 L-system 語法串連起來生成出同時具有朵雲紋、如意雲紋、團雲紋及雲尾的綜合體複雜東方風格雲紋，本論文的研究重點就在於將這些平面裝飾紋路擴展成為在三維立體空間當中可以生長的動態雲紋。

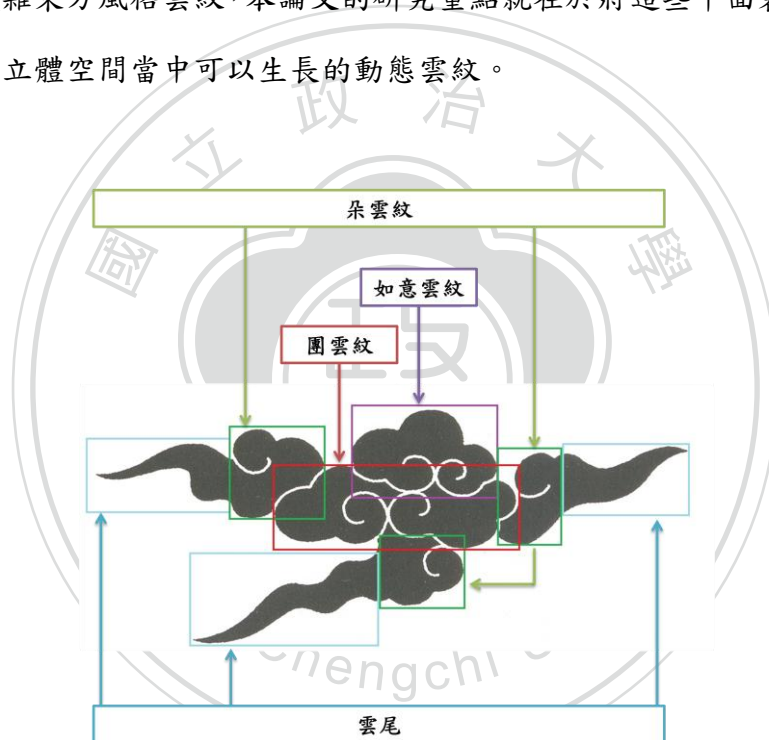


圖 3.5 雲紋結構分析圖

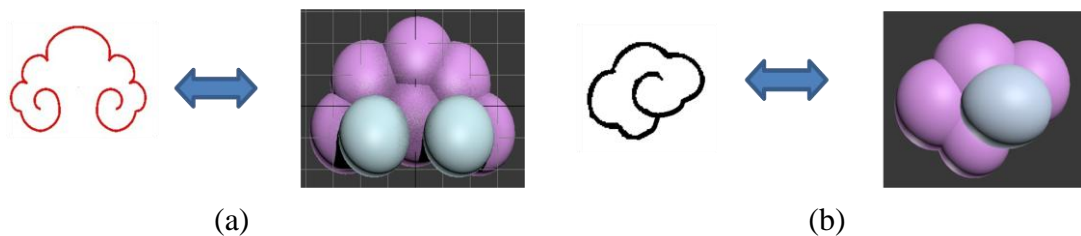


圖 3.6 以平面雲紋圖樣為基礎模擬其立體雲紋之示意圖 (a)如意雲紋(b)朵雲紋

以上簡介了本研究針對 L-system 語法之遞迴生長規則，而從二維到三維空間的呈現，將原本平面的雲紋想像成使用圓型所排列組合而成，要將之呈現為立體的結果，可以試著想像一下將原本的圓將之利用球體取代，立體之東方風格雲紋則可以很直覺地表現出來。原本之二維平面雲紋圖樣裝飾的應用較為普遍，而如今藉由我們的方法則可將原本較為死板的裝飾紋路轉換為立體且藉由 L-system 的生長方式產生出多種變化的動態立體東方風格雲紋圖樣。

3.2.1 雲紋圖樣結構分類

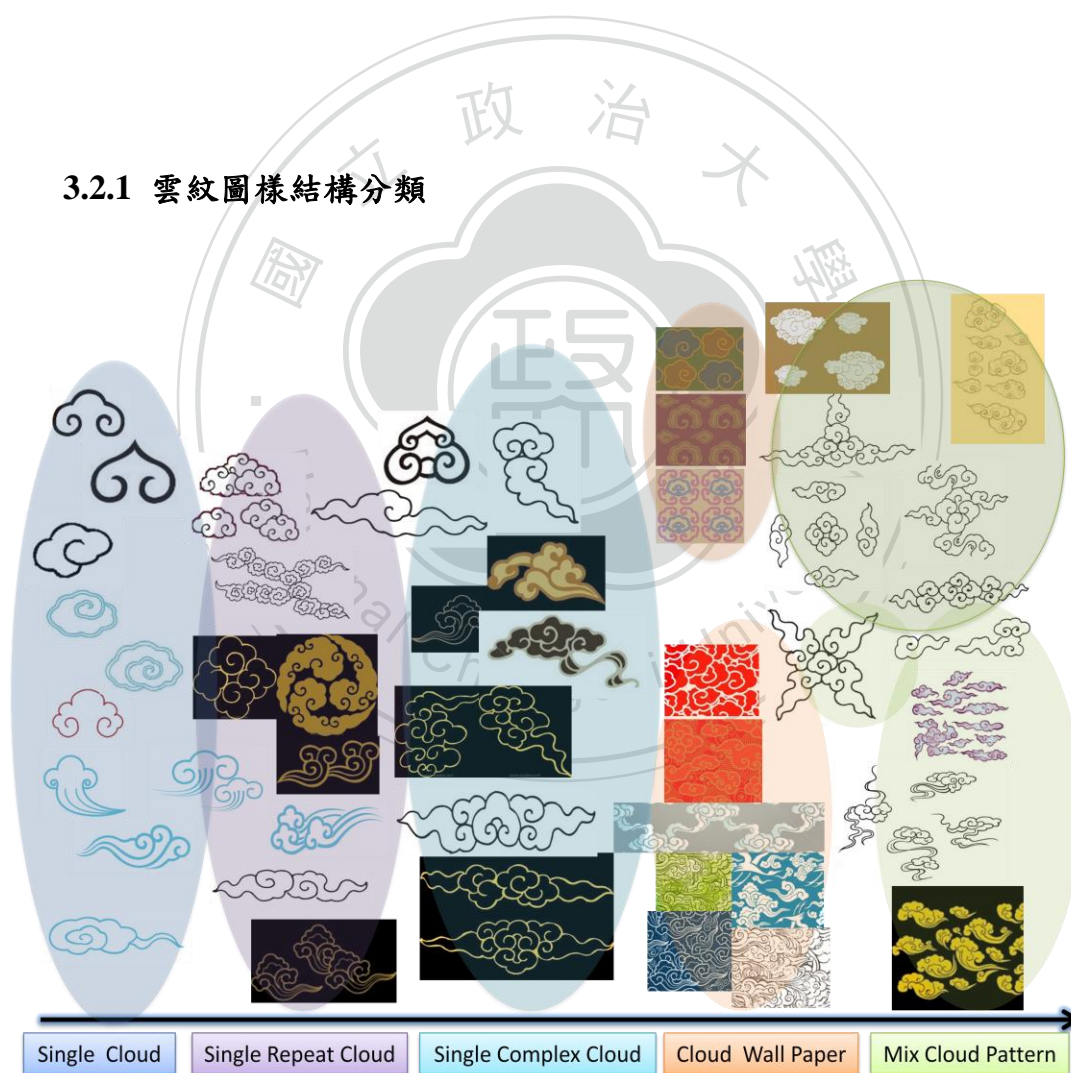


圖 3.7 雲紋圖樣分類

在本研究當中，我們蒐集了大量的雲紋圖樣為參考：平面雲紋 50 張，從這些圖樣當中仔細分析出其種類結構，依照結構先做分類，平面雲紋可以分為五種：單朵雲紋(Single cloud pattern)、單朵複雜雲紋(Single complex cloud pattern)、單一雲朵重複雲紋(Single repeat cloud pattern)、雲紋桌布圖樣(Cloud wall paper)、組合式雲紋(Mix Cloud pattern)，每一種類的圖樣皆有十張，所整理之結果如圖 3.7 所示，從不同類別的圖樣當中，就單朵雲紋來說，就可以很清楚的看出朵雲紋與如意雲紋的單一結構，並可以選擇性地搭配雲尾；單朵複雜雲紋則較偏向將數個朵雲紋與如意雲紋作組合；單一雲朵重複雲紋為以一朵雲紋為基礎，以不同角度或方向排列而成；雲紋桌布圖樣則是另一種華麗的雲紋圖樣樣式；組合式雲紋圖樣是將單朵與單朵複雜或是單朵重複再進一步地加以組合而成，在下一章研究方法當中我們會針對單朵雲紋、單一雲朵重複雲紋、單朵複雜雲紋以及組合式雲紋搭配 L-system 語法做更進一步的分析說明。雖然東方風格之雲紋看似很難找出其規則與結構，但是從本研究中所分析的結構來看的話，可以發現雲紋也是從最基本的小單位慢慢的疊加上去，如果我們可以找到其搭配組合之角度與位移量，則可以有效地將雲紋組合起來。

3.2.2 雲尾結構

傳統的東方風格雲紋，其中「雲尾」的結構組合常常是使得雲紋可以更為飄逸與產生流動性的關鍵點，所以在許多的雲紋圖樣當中，我們可以發現雲尾的組合也可以很多元且具變化性，從雲尾的末端也可以看出其感覺大不相同，如圖 3.8 所示[2]，(a)分離的雲尾末端與(b)密合的雲尾末端，由於平面的雲紋圖樣對於雲尾的表現往往都是利用簡單的波浪線條來呈現，波浪可以表現出雲的飄逸感，而線條的走向則是表現出飄逸的方向。

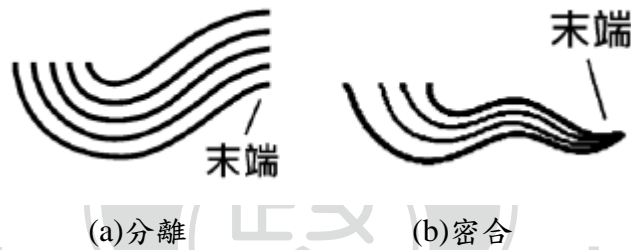


圖 3.8 雲尾末端(a)分離 (b)密合。[2]

以上簡單的介紹了雲尾末端的分類，而雲朵本身與雲尾的組合也可以很多元，如圖 3.9，上方的彩色圖為雲紋的原始圖檔，下方為利用紅色方框標示出雲朵；藍色方框標示為雲尾的模擬示意圖，由左至右可以看出圖 3.9(a)其雲朵與雲尾的走向，雲朵與雲尾呈現出彼此垂直 90 度 T 字型的結構，這樣的結構可以使雲紋表現出緩慢上升飄逸的感覺，而圖 3.9 (b)的雲朵與雲紋則是呈現相差少許的位移量呈現出彼此平行的構圖，這樣子的雲紋結構比較相似於雲朵經過外力的影響，例如風的吹動，而產生出緩慢的飄移感，最右邊圖 3.9 (c)，則是雲朵與雲朵之間利用雲尾串接起來，可以表現出雲朵在飄移之時，雲朵與雲朵之間因為互相產生外力影響而呈現出的互動變化。

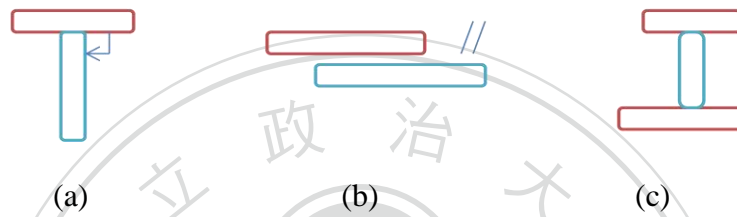
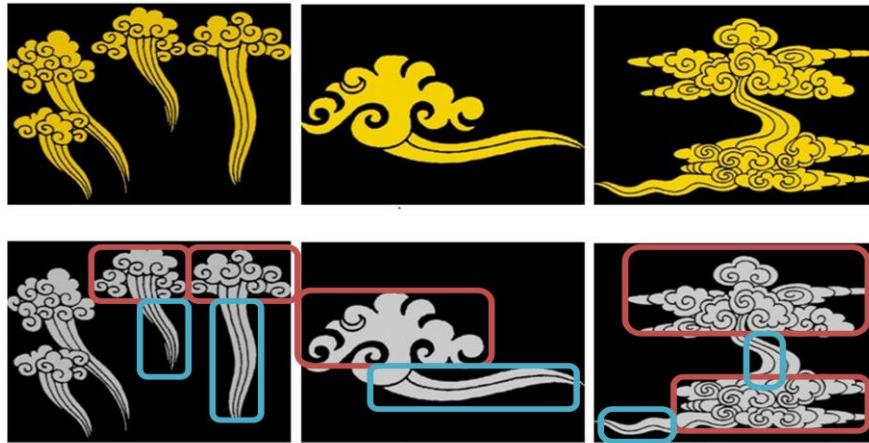


圖 3.9 雲尾與雲朵之結構分類。

(a)彼此垂直 (b)彼此平行 (c)雲朵與雲朵之間利用雲尾串聯

根據我們對於雲紋與雲尾之組合的分類來看，雲尾與雲朵的搭配也可以產生許多不同風格的東方風格雲紋，並搭配以上雲尾分離或是密合的結構，雲尾可以呈現出更為豐富的組合結果，必須加入在 L-system 雲尾的語法生成系統當中，透過參數的設定及各種角度與位移量的組合，產生出多樣的雲紋組合。

3.3 立體雲紋結構分析

立體雲紋圖樣的部分大多都出現在古文物等雕刻物上，雕刻品上的雲紋要呈現出立體感，最重要的就是在雲紋的捲曲部分加上深度的變化，利用高低的落差將雲紋的紋路立體感呈現出來，也因為雕刻的技巧比較精細與費工，在現實生活當中要設計出這樣的裝飾紋路往往也較費時也不易取得，另一方面 L-system 的特點就是其自我相似及重複性，而如果我們可以利用將這些重複性質很高的紋路轉換為 L-system 的語法當中，或許生成立體的雲紋會簡單許多。



圖 3.10 立體雲紋於雕刻品上的應用

根據上一小節的雲紋分析，本研究的重點將著重於在「朵雲紋」與「如意雲紋」及「雲尾」的部分進行分析，並可隨使用者的喜好可選擇性地搭配雲尾，而團雲紋的部分則會利用 L-system 的語法來控制，將所生成的朵雲紋與如意雲紋加以組合，則可以更加多元化地生長出不同風格的立體雲紋，我們先定義如圖的模擬符號，在平面雲紋下，可以將其雲朵頂端與雲朵尾端想像為一個一個的“圓”所組合而成，而其所相對應於立體雲紋的配對就定義為“球體”，直覺地，目的就是將原本利用圓所組合而成的平面雲紋，將其利用球體推疊組合起來，而至於球體與球體之間的推疊規則，我們將留到 4.2 小節研究方法中做更詳細的介紹，雲頭的部分是使用圓來模擬，至於雲尾的部分，由於雲尾本身就帶有使雲看起來具有飄逸的感覺，所以常常會有自然彎曲的平滑曲線，考量到平滑曲線必須要對應為立體的彎曲弧度，我們將其平面曲線從雲朵與雲尾的交界處開始只要有遇到

轉折性彎曲的部分就利用一個“四邊形”所表示，直到雲尾的末端處，最後再使用一個三角形做結尾，而從交界處開始計算會有 n 個四邊形與一個三角形，而與之所搭配的立體結構則使用“圓柱體”與“圓錐體”作組合，定義好這些平面與立體之模擬對應後，我們要開始針對所蒐集到的平面雲紋圖樣作進一步的模擬。

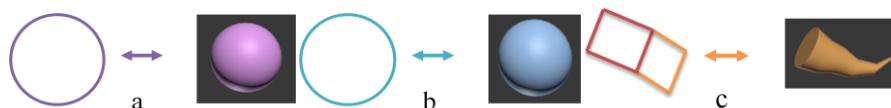



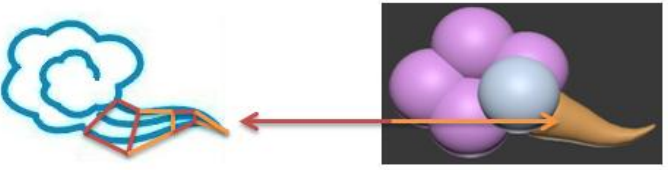


圖 3.11 模擬平面雲紋與立體雲紋的符號對應
(a)淡紫色-雲朵頂端 (b)淺藍色-雲朵尾端 (c)雲尾

有了這個平面轉立體的初步想法之後，就以如意雲紋、朵雲紋及雲尾三個簡單的參考圖樣如圖 3.11 所示作模擬，分別使用(a)淡紫色與(b)淺藍色所示，雲朵頂端的部分也是主要雲朵的整體架構的分佈，其分佈的規則趨近於植物莖部的分支，只是改從由上往下作生長，而如意雲紋雲尾的部分則是承接自最末端下方的球體，接續作生長，而雲朵尾端的部分最可以表現出立體雲紋的一大特點，我們必須將球體在座標系上位移一個深度值，使得其雲朵尾端的部分特別立體以呈現出立體雲紋的效果。而朵雲紋的部分概念也類似於如意雲紋，其不同之處在於朵雲紋的雲朵身體部分是依照一個螺旋狀式的結構作生長，此螺旋的位移量也必需加入深度值的位移量，會使得朵雲紋看起來更為飽滿，最後再接續生長雲朵尾端使得朵雲紋看起來更具立體感。最後則是雲尾的模擬，這部分則採取彈性的生長方式，基本上簡單的雲尾大約在 3~4 個四邊形加上一個三角形可以表現出來，相對應的就為四個連續的圓柱體與一個圓錐體接連起來較為自然，而較為複雜的雲尾，可能就必須用到 5 個以上的四邊形與一個三角形作模擬，當然雲紋的呈現可以依使用者的需求選擇是否要搭配雲尾或是單純的雲紋，通常一般而言，無雲尾的雲紋所表現出來的感覺較為靜態，搭配雲尾的雲紋則較為動態，不過在最後系統所生長出來的雲紋都可以呈現出動態生長的過程。

表 2 朵雲紋、雲尾的平面轉換立體之模擬

平面雲紋圖樣	平面雲紋與立體雲紋的結構分析
 <p data-bbox="375 504 470 548">朵雲紋</p>	
 <p data-bbox="391 739 454 784">雲尾</p>	



第四章

研究方法與步驟

4.1 研究步驟

在實作立體雲紋 L-system 語法前，先將系統大略的步驟以流程圖來做說明，從平面雲紋所觀察到的資訊，例如雲紋中圓與圓之間的比例、位移及角度這些資訊我們必須先將之定義清楚，以便於之後語法規則上對於參數的使用，並且再根據初始值 Axiom 所定義的規則語法及括弧內的參數作判斷，系統會依據參數的判斷跑相對應的生長規則，使用者只須在 Axiom 的地方下指定的參數即可。根據圖 4.1 系統流程圖，系統最為核心的部分即為 L-system 語法根據不同條件的判斷遞迴呼叫的過程，這個步驟包含了雲朵的生長、雲尾的延伸、以及不同雲朵之間的結構呈現，而由 Axiom 中之參數統一指定立體雲紋的生成規則總類，系統在逐步生成雲紋之時，整體結果之呈現即為一個動態的立體雲紋生長過程，細節部分的語法流程說明則如圖 4.2。

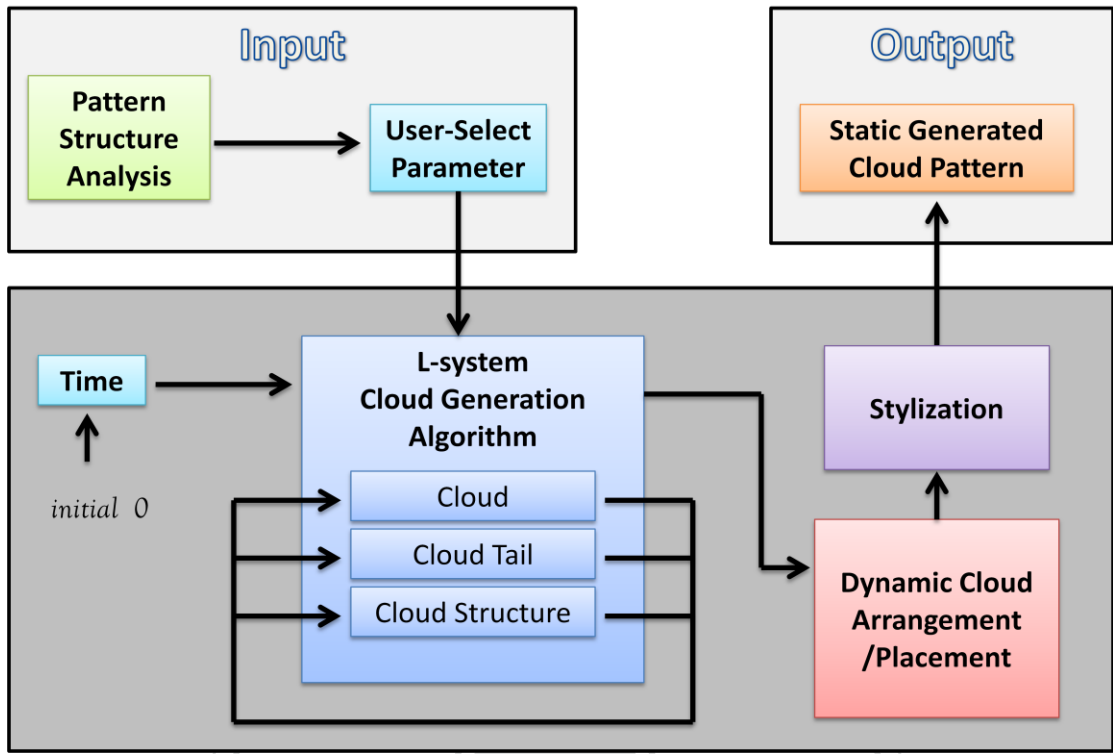


圖 4.1 系統流程圖

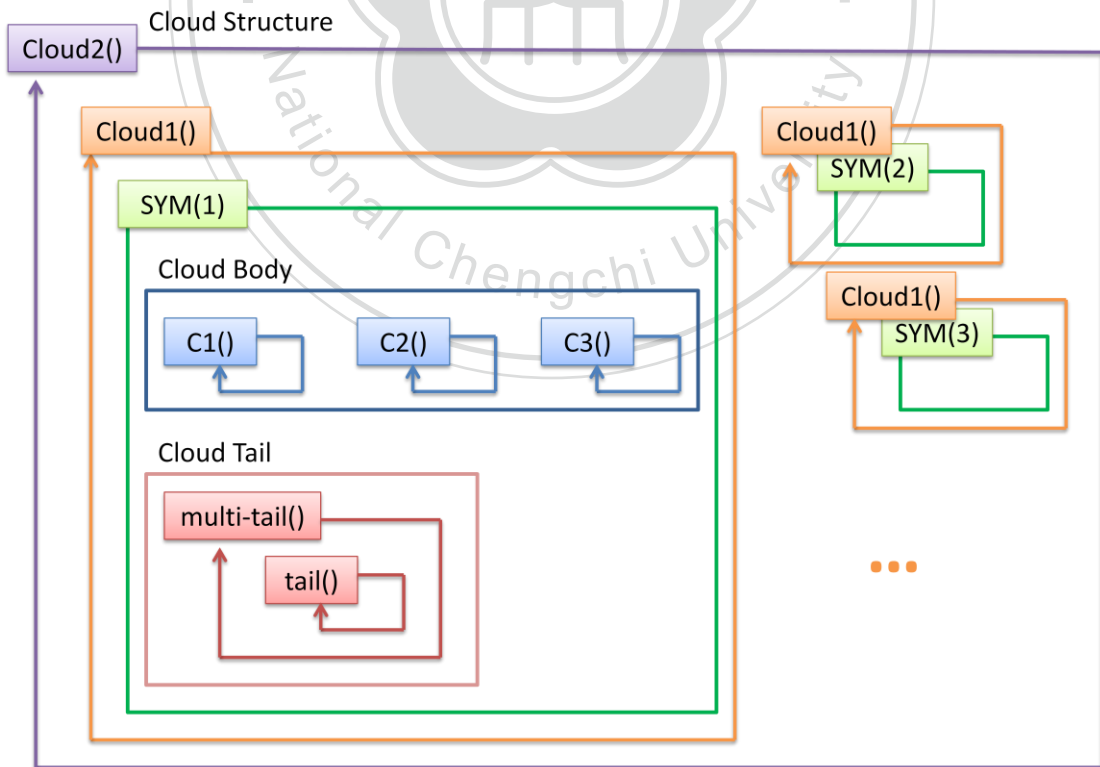


圖 4.2 L-system 立體雲紋生成規則流程圖

請參考圖 4.2，圖中有箭頭再次地指向規則本身則代表此規則會進行自我呼叫的迭代，對於 L-system 語法的流程我們可以大略的將規則以巢狀的圖示表達出層層遞迴的概念，其中雲朵的生長即可分為五種主要規則：C1-A()、C1-B()、C2()、C3()、C4()，皆會自我呼叫遞迴生成不同的雲朵結果，雲尾的規則分為 tail()、multi-tail()，分別負責產生雲尾一段一段的彎曲變化以及多條雲尾的生成，雲朵及雲尾之組成則藉由 SYM() 這項規則來定義，根據結構的不同產生相對應的雲朵及雲尾組合，至於 Cloud1() 則是負責在指定相同結構下進行結構式雲紋的重複生成，Cloud2() 則是在不同結構與結構之間的重複生成，如此一來，東方風格雲紋可以利用此 L-system 規則轉化成為較有系統的生成方式，產生出許多不同樣式的立體雲紋結果。

在此章接下來的小節當中會依序介紹針對雲朵的生成方法、根據不同動態模擬所生成的雲尾結果、最後則是將雲朵與雲尾組合起來以及搭配結構化的語法將雲朵與雲朵之間產生更多不同的組合方式。

4.2 Parametric L-System

從相關研究 Parametric L-system 來看，Prusinkiewicz 不僅僅只是提出一個具有語言式可結構化地描述出植物逐步生長的過程，另外在不久後又提出一個概念稱為” turtle interpretation of L-system” [5][6]，想像為一個 turtle 在三維空間當中的行徑過程，可以將原先文字串列並搭配指定的參數轉換為相對應的幾何模型並呈現在三維空間當中，如圖 4.3。

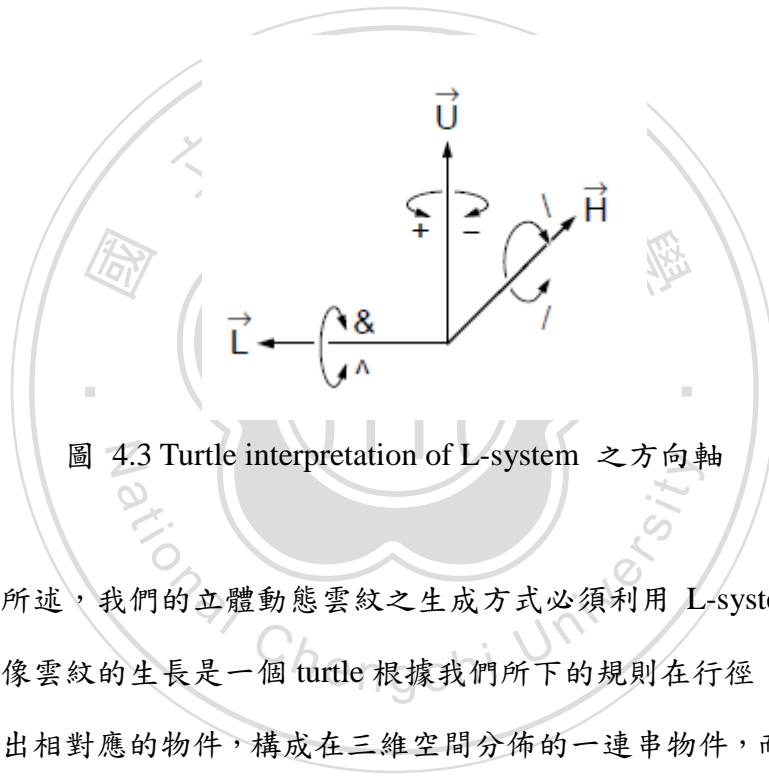


圖 4.3 Turtle interpretation of L-system 之方向軸

從上一小節所述，我們的立體動態雲紋之生成方式必須利用 L-system 之生長規則成長，想像雲紋的生長是一個 turtle 根據我們所下的規則在行徑，而在行徑的過程逐步畫出相對應的物件，構成在三維空間分佈的一連串物件，而所呈現出來的結果就是立體的雲紋圖樣，而對於立體雲紋的許多規則，如表 3 所列，各種的符號有其相對應的動作，且座標環境是三維的空間，在接下來要介紹的規則語法皆是由此表所對應的動作組合而成。

表 3 L-system 符號與生長動作對應

符號	相對應之動作
#define	Define parameter
Axiom	Axiom 初始值
&	Pitch down
^	Pitch up
+	Turn left
-	Turn right
/	Roll left
\	Roll right
]	Pop matrix
[Push matrix
· 	Rotate 180°
g	translate
O	Draw sphere
ran(x)	Generate 0~x random parameter
@Gs	Start draw cylinder
@Gt	Tangent
@Gc	End draw cylinder

在建置與定義立體雲紋圖樣時目前的方法策略是將雲頭、雲尾及雲紋結構三大類分開處理，有負責生成雲頭的函式，例如 C1-A、C1-B、C2 等；也有負責生成多重雲尾的函式，tail 與 milt-tail；以及將雲紋組合起來的結構語法規則，SYM、Cloud1、Cloud2，在接下的小節當中會依序針對雲頭、雲尾、以及立體雲紋結構的語法做更詳細的方法說明。

首先以下則是系統的基本參數設定，目前 STEP 設定為 25，這個參數會搭配 Animation 參數之設定以調整出動態的雲紋生長過程，動態參數的設定上 Step = 2；Swap interval = 5，即表示每兩個 Steps 用一個 frame 呈現出來，更可以設定 frame 交換的速度，目前設定為每 5*0.01 秒做置換，這些參數皆可以使用者的需求做進一步設定以符合裡的動態結果。

Rule parameter define：



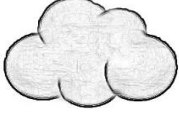

```
/* ----- System define ----- */
#define STEPS 25
Lsystem: 1
derivation length: STEPS

/* ----- Animation define ----- */
Step: 2
Swap interval: 5 *0.01 s
```

4.2.1 雲頭 L-system 語法

在這一小節當中，會分別介紹不同大小朵雲紋的生成方式，搭配平面雲紋定義其編號分別為 C1、C2、C3、C4，如表 4 所示，在生長規則命名上我們也根據其編號為參考，分別為 C1()、C2()、C3()、C4()，其中 C1 為朵雲紋之最基本構成元素，在此又分為 C1-A()及 C1-B()，兩種不同的立體模擬方法，將會依序在下面做說明：

表 4 朵雲紋平面雲紋圖樣編號

編號	C1	C2	C3	C4
平面雲紋圖樣				
生長規則	C1-A() C1-B()	C2()	C3()	C4()

雲紋的圖樣不管多複雜，在實作方法上會根據其外觀找出適當的模擬方法，例如角度與位移量等等資訊皆會以平面圖樣作為基礎來建構我們的 L-system 東方風格雲紋語法，從單一雲朵的最小朵雲紋，編號“C1”，在實作 C1 立體雲紋方法上，又可再細分為 C1-A、C1-B 兩個不同結構的結果呈現，觀察其平面雲紋的圖樣來看，其雲紋的外觀具有波浪狀的圓弧形輪廓，類似於將數個圓擺放成螺旋狀，C1-A 的方法即利用球體排列將立體感呈現出來，故在規則上的撰寫將會依照其螺旋狀的位移並加上具深度的旋轉角度，將原先的平面圖樣轉換為立體的雲紋結果；而 C1-B 為利用立體的管狀物件模擬出雲紋外輪廓的波狀感，有別於球體的填充飽滿感，此方法較強調於外輪廓的美感。

C1-A :

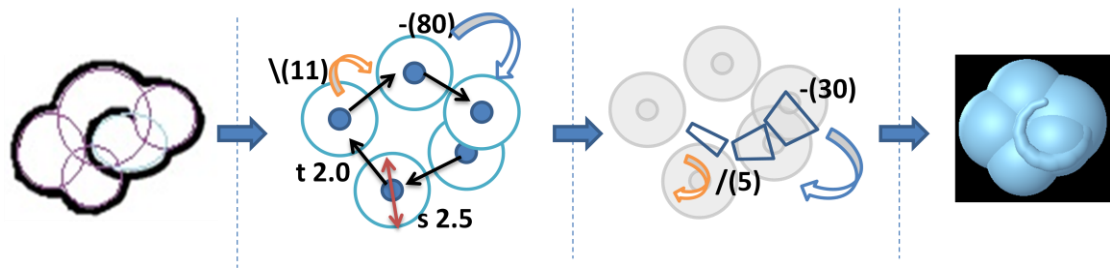


圖 4.4 朵雲紋 C1-A 生長規則流程圖

如圖 4.4 所示，為 C1-A 方法之流程圖，依序為傳統平面雲紋圖樣之參考、依照其圓與圓之間位移量與旋轉角度模擬示意圖、模擬雲朵末端之結合與相對應的立體雲紋生成結果。下一頁為生成朵雲紋 C1-A 的 L-system 語法規則，C1-A 這個函式會根據 Axiom 所給定的參數進行遞迴呼叫在相對位置上生成球體模擬出立體感的朵雲紋，而在 C1-A 規則當中有四個參數， d_0 用於控制球體之間的位移量、 db 用於控制球體的大小、 n 用於控制 C1-A 朵雲紋之球體生成數量、 T 則是將雲尾的參數傳遞下去用於產生相對應的雲尾結果，至於要生成雲朵末端勾捲的部分我們在這使用寬度由寬至細的平滑圓柱體搭配角度與旋轉變化以模擬出平面雲紋當中勾捲的特色，搭配的規則語法為 C1-Aend，其中的參數 c 為控制漸變的圓柱體其疊代呼叫次數，而在疊代過程當中，參數 w 用於調控圓柱體的寬度調整。

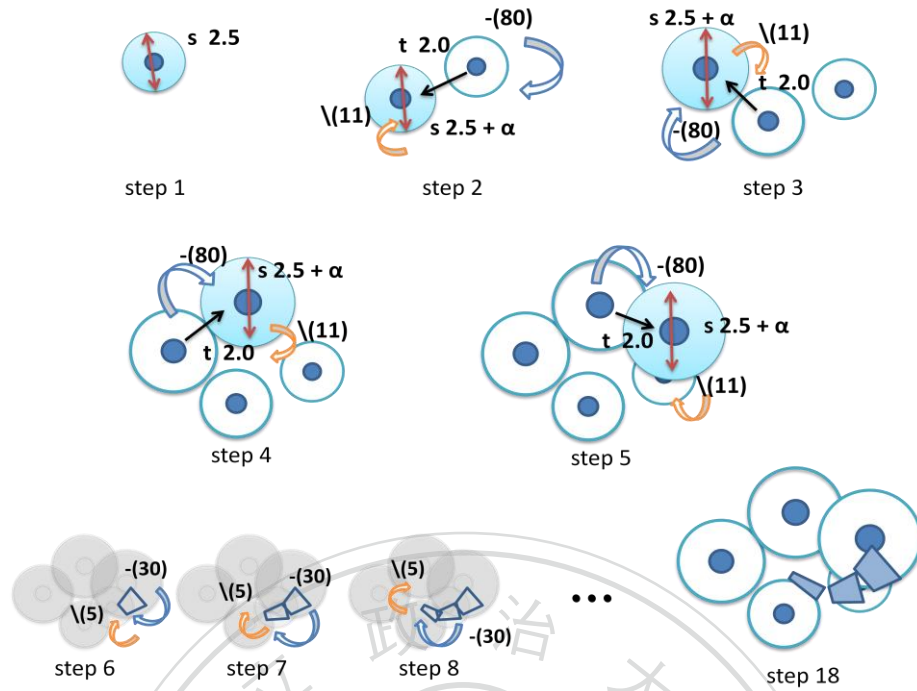


圖 4.5 朵雲紋 C1-A 幾何結構生成模擬步驟

圖 4.5 為系統隨著遞迴的過程配合指定旋轉角度與位移量逐步地將球體畫出，且球體亦會隨時間而成長、圖 4.6 則是系統逐步所生成出來的朵雲紋 C1-A 結果。

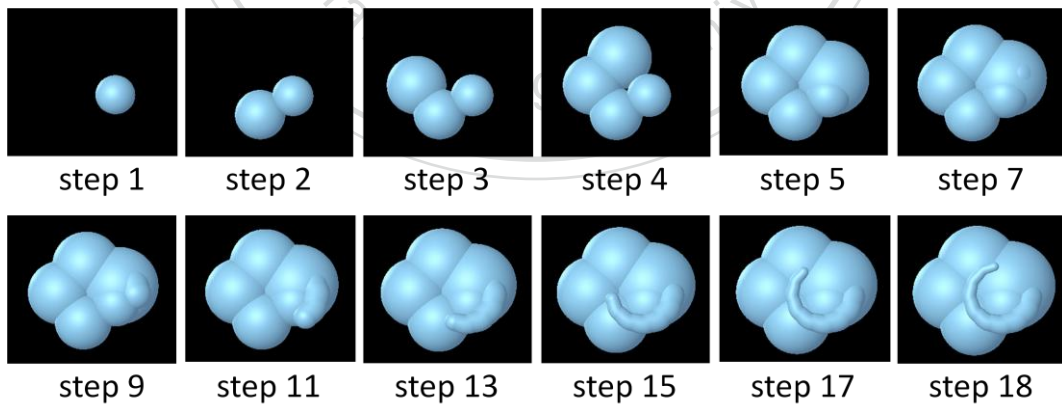


圖 4.6 朵雲紋 C1-A 系統生成之逐步結果

L-system 語法規則(C1-A) :

```

/* ----- C1A define ----- */
#define C1Aang1 80
#define C1Aang2 11
#define C1Aang3 30
#define C1Aang4 5
#define CR 0.87

Axiom : -(80)[ C1A( 2 , 2.5 , 5 , T ) ]

/* Parameter : */
/* d0 : translate amount */
/* db : sphere size */
/* n : iteration count */
/* T : tail parameter */

/* Generate Cloud */

C1A(d0,db,n,T): n == 0 →
  ^((65)g(1.1)+(90)\(3)g(0.6)[-(65)&(55)C1Aend(10)][milt-tail(T,T)]
C1A(d0,db,n,T) : n > 0 →
  -(C1Aang1)\(C1Aang2)g(d0)[O(db)]C1A(d0,ran(0.5)+db,n-1,T)

/* Parameter : */
/* c : iteration count */
/* w : cylinder width control */

/* Generate Cloud Curve */
C1Aend(c) : c > 0 → C1Aend_s(c)C1Aend_e(c,1.6)
C1Aend_s(c) → #(0.2)@Gt(1)@Gs
C1Aend_e(c,w) : c > 0 →
  -( C1Aang3)/( C1Aang4)f(0.9)#(w*CR)@Gc(10)C1Aend_e(c-1,w*CR)

/* End Cloud Tail */
C1Aend_e(c,w) --> -(20)/(15)@Ge(80)@O(w)

```

C1-B :

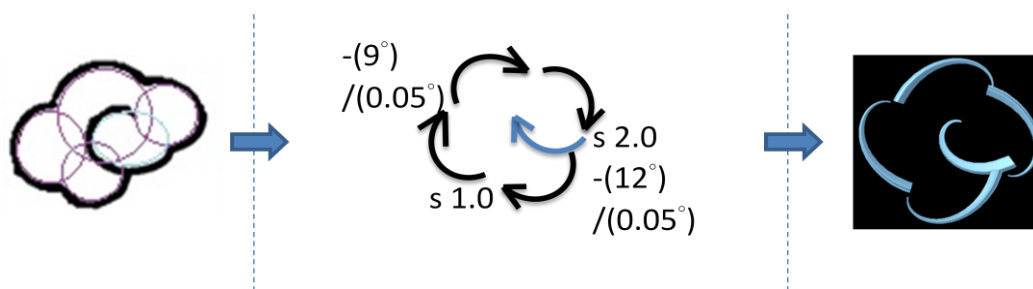


圖 4.7 朵雲紋 C1-B 生長規則流程圖

基於同樣的雲紋圖樣，C1-B 採用另一種管狀立體形式來模擬生成，如圖 4.7 之流程圖所示，而中間詳細的逐步生成過程可以再參考圖 4.8，從初始值設定以及隨著疊代的次數增加、位移量與角度的累加，管狀物件則可慢慢地勾勒出平面雲紋圖樣的外圍輪廓，模擬出具流線美感的立體雲紋圖樣，圖 4.9 則是系統所逐步生成之動態結果。

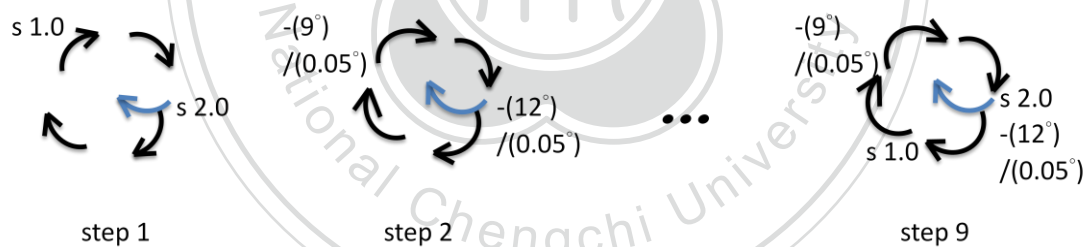


圖 4.8 朵雲紋 C1-B 幾何結構生成模擬步驟

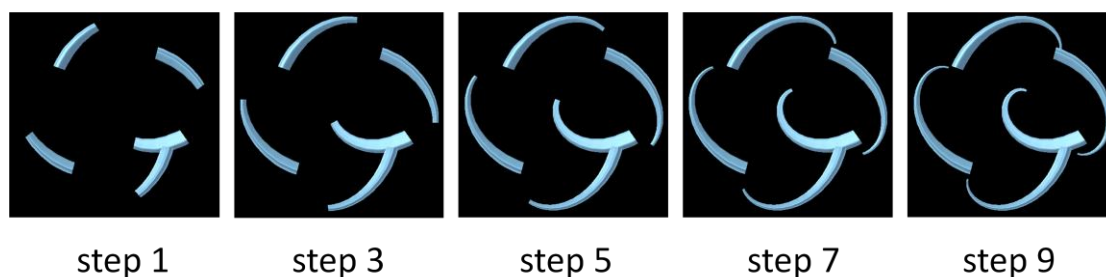


圖 4.9 朵雲紋 C1-B 系統生成之逐步結果

L-system 語法規則(C1-B) :

```
/* ----- C1B define ----- */
#define C1BR1 0.9
#define C1BR2 0.9
#define C1BWid1 1
#define C1BWid2 2
#define C1Bang1 9
#define C1Bang2 0.05
#define C1Bang3 12

Axiom : C1B(0)

/* Parameter : */
/* T : tail parameter */

C1B(T) → [+ (55)g(4)[- (75)C1Bcurve1(1)]
          [- (40)g(4)[- (65)C1Bcurve1(1)]
          [+ (145)g(4.5)[- (70)C1Bcurve1(1)]
          [- (130)g(4)[- (60)C1Bcurve1(1)]
          [- (115)g(5)[- (120)C1Bcurve3(1)]]

C1Bcurve1(s) → @#(4)#(C1BWid1)@Gr(1)@Gt(1.0)@GsC1Bcurve2(s)
C1Bcurve2(s) → -(C1Bang1)/(C1Bang2)C1BLine1(s)C1Bcurve2(s*C1BR1)
C1Bcurve3(s) → @#(3)#(C1BWid2)@Gr(1)@Gt(1.0)@GsC1Bcurve4(s)
C1Bcurve4(s) → -(C1Bang3)\(C1Bang2)C1BLine2(s)C1Bcurve4(s*C1BR2)

/* Draw Cylinder */
C1BLine1(s) → g(s)#(s*C1BWid1)@Gc(5)
C1BLine2(s) → g(s)#(s*C1BWid2)@Gc(5)
```

C2 :

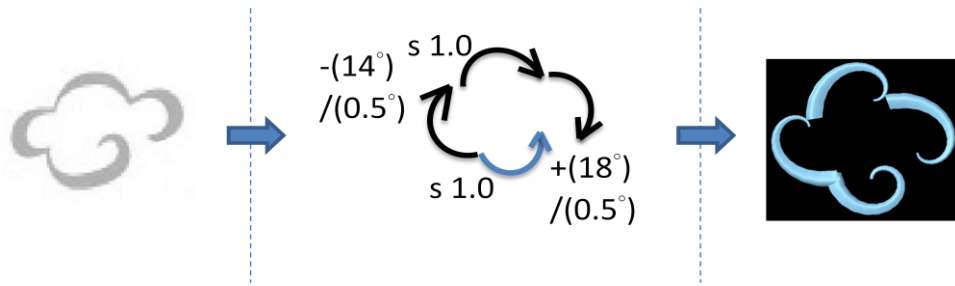


圖 4.10 朵雲紋 C2 生長規則流程圖

圖 4.10 所示為朵雲紋 C2 之生成流程圖，中間則是系統在生成 C2 之許多參數的設定，詳細的逐步模擬過程如圖 4.11 所示，而最後的逐步生成結果可以參考圖 4.12，利用倍數的設定可以將管狀物件隨著疊代的增加，使結果由寬變細，模擬出平面雲紋圖樣的圓弧勾勒狀觀感。

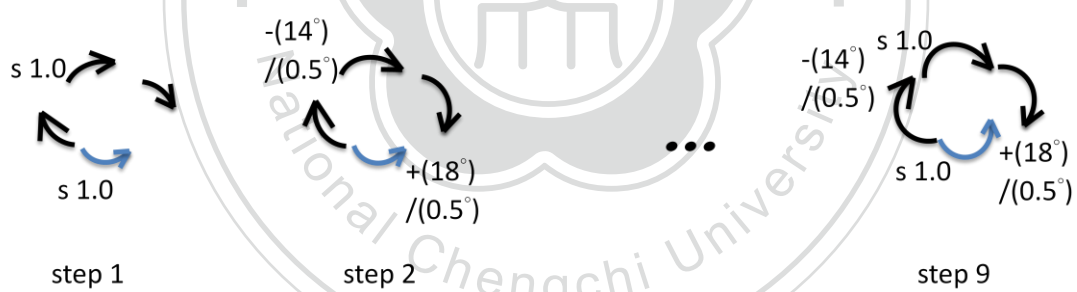


圖 4.11 朵雲紋 C2 幾何結構生成模擬步驟

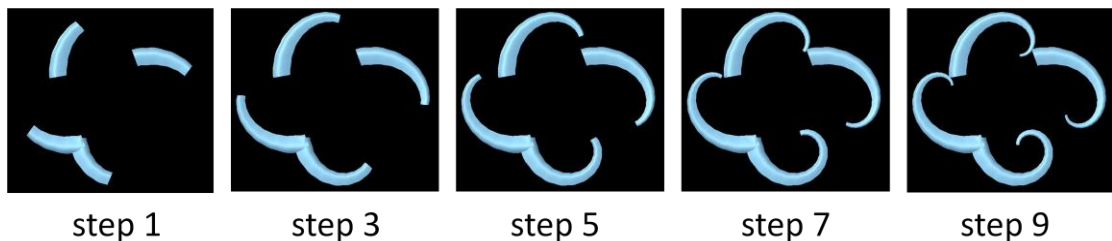


圖 4.12 朵雲紋 C2 系統生成之逐步結果

L-system 語法規則(C2) :

```
/* ----- C2 define ----- */
#define C2R1 0.9
#define C2R2 0.92
#define C2Wid1 1
#define C2Wid2 1
#define C2ang1 14
#define C2ang2 0.5
#define C2ang3 18
#define C2ang4 0.5

Axiom : +(10)[C2(0)]

/* Parameter : */
/* T : tail parameter */

C2(T) → [+ (55)g(3)[- (55)C2curve1(1)]]
        [+ (145)g(3)[- (45)C2curve1(1)]]
        [- (50)g(3)[- (30)C2curve1(1)]]
        [- (225)g(3)[+ (30)C2curve3(1)]]
        [+ (135)g(1.5)milt-tail(T,T)]

C2curve1(s) --> @#(1)#(C2Wid1)@GsC2curve2(s)
C2curve2(s) --> -(C2ang1)/(C2ang2)C2Line1(s)C2curve2(s*C2R1)
C2curve3(s) --> @#(1)#(C2Wid2)@GsC2curve4(s)
C2curve4(s) --> +( C2ang3)/(C2ang4)C2Line2(s)C2curve4(s*C2R2)
```

C3:

朵雲紋編號“C3”方法說明，根據所參考到的平面雲紋圖樣來看是稍微較大朵的朵雲紋，可參考如圖 4.13 之流程圖最左方一張，我們將其生長結構定義為從側面依螺旋狀路線來生長，與 C1-A 的主要差別之處在於其生長的方向及旋轉角度不同，利用側面生長的方式模擬出較大朵的朵雲紋。

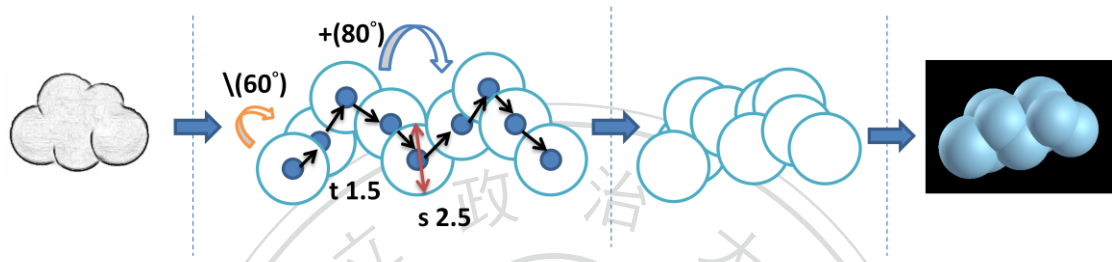


圖 4.13 朵雲紋 C3 生長規則流程圖

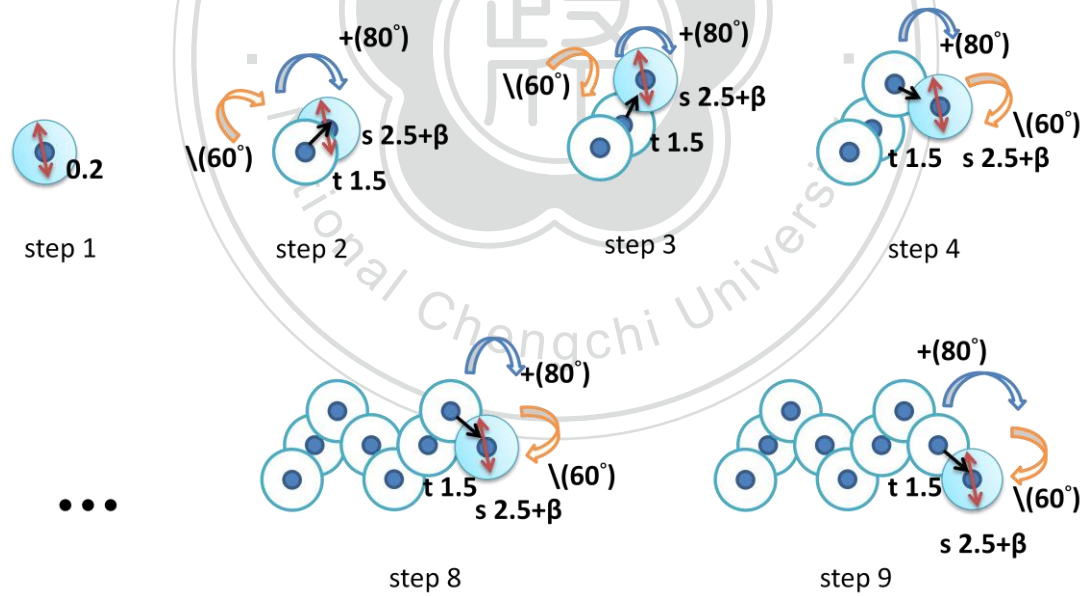


圖 4.14 朵雲紋 C3 幾何結構生成模擬步驟

在 C3 之 L-system 語法規則上與 C1-A 相似，比較特別一提的是由於 C3 其生長的方向是從側面螺旋狀旋轉，C3 函式當中之參數 d_0 、 d_b 之用途與 C1-A 相同，

主要第三個參數 n 的初始值設定在於其差異之處，使之進行迭代的 C3 的規則較多次，生成球體物件較多，從 C3 之 Axiom 當中 n 的值可以看出，而 T 一樣為用於傳遞雲尾生成所要的參數設定。圖 4.14 為朵雲紋 C3 之逐步生成示意圖，在每一次的迭代都會根據規則當中所給的位移量與角度作變化，且球體的大小也會隨時間成長，圖 4.15 則是系統逐步生成的朵雲紋 C3 的結果圖。

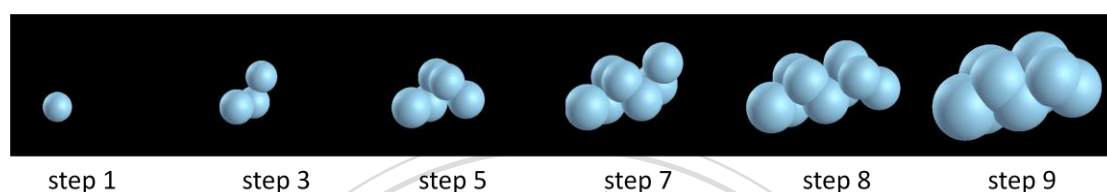


圖 4.15 朵雲紋 C3 系統生成之逐步結果

L-system 語法(C3)：

```

/* ----- C3 define ----- */
#define C3ang1 60
#define C3ang2 80

Axiom: [ ^ (70) [ \ (60) C3 ( 1.5 , 2.5 , 9 , T ) ] ]

/*Parameter : */
/*d0 : translate amount */
/*db : sphere size */
/*n : iteration count */
/*T : tail parameter */

C3(d0,db,n,T) : n == 0 && T != 0 →
[-(C3ang1)\(C3ang2)g(d0)][milt-tail(T,T)]
C3(d0,db,n,T) : n > 0 →
\[C3ang1)+(C3ang2)g(d0)[O(db)]C3(d0,ran(0.2)+db,n-1,T)

```

C4:

編號“C4”則是我們系統所要生成最大朵的朵雲紋，在建構大朵的立體雲紋時必須考慮整朵雲紋的大輪廓必須接近一個橢圓的外觀，而要將這個大範圍利用的遞迴的方式生長出來不容易，參數也必須隨之改變，故在此利用既有的朵雲紋 C1-A 生成規則加以組合，利用左右各一朵 C1-A 與正中間數個球體作為填充以構成飽滿雲朵，而正中間則為主要 C4 遞迴規則模擬平面雲紋之螺旋捲曲樣貌，詳細之流程圖如圖 4.16 所示。

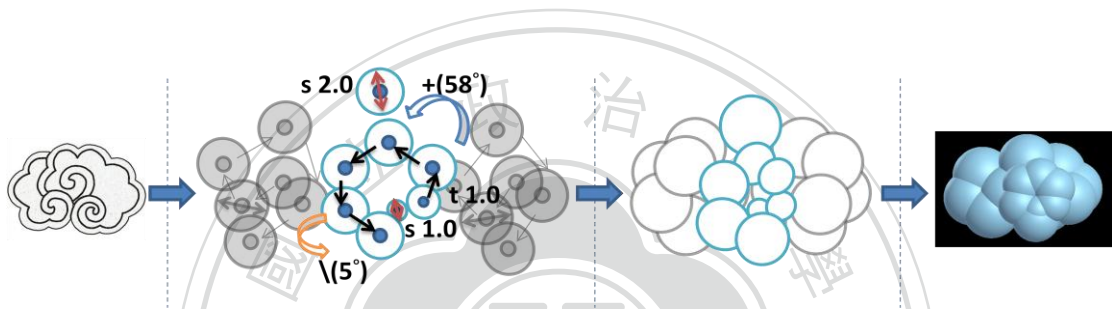


圖 4.16 朵雲紋 C4 生長規則流程圖

朵雲紋 C4 之生成方式相較於先前的生長結構顯得複雜許多，也是利用前述的生長概念，但在 C4 之 Axiom 之設定就較為複雜，首先將 Axiom 分成四個部分來看，首行與末行為針對 C1-A 的生成，但是必須組合出符合橢圓形飽滿的雲朵外觀故必須稍微作角度的調整；接下來第三行則是針對 C4 雲紋圖樣正中間之內捲曲螺旋紋路，要生成更為捲曲的排列方式其球體大小初始值就必須大幅縮小，旋轉之角度也必須隨之變動，最後一步為第二行直接畫球體以將整體朵雲紋 C4 之立體感更為均勻且飽滿，另外一點對於 C4 雲尾的生成我們也一樣是設置在 C4() 規則當中的 T 來控制，而有呼叫到 C1-A() 函式則必須將 C1-A 的參數 T 設為 0，以調整出單朵雲尾的生成，而使用者如果想要多條雲尾的生成，在此也可將 T 之參數皆選擇在 C1-A() 與 C4() 皆作設定，這邊的結果留至下一章結果討論再進一步分析。

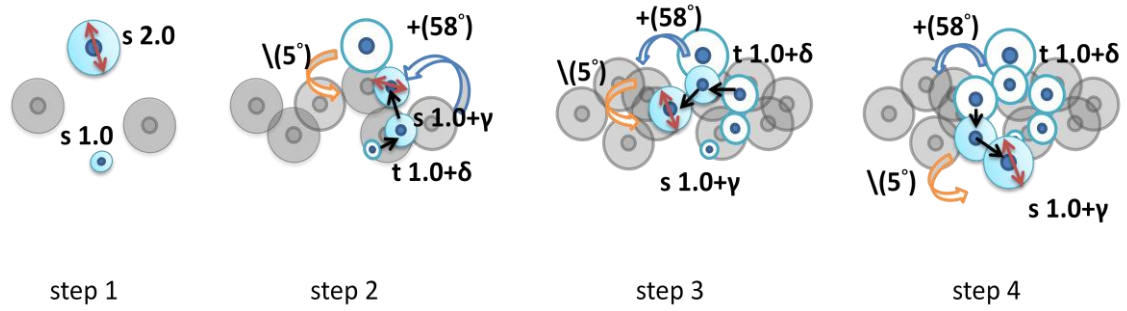


圖 4.17 朵雲紋 C4 幾何結構生成模擬步驟

圖 4.17 為朵雲紋 C4 之逐步模擬生長圖，圖中灰階的為 Axiom 中生成 C1-A 的模擬圖，彩色的為模擬 C4 雲朵捲曲的部分，可以看出球體大小明顯較小，螺旋狀的位移也較明顯，圖 4.18 為朵雲紋 C4 之逐步生成結果圖，可根據語法規則及步驟圖比較參考。

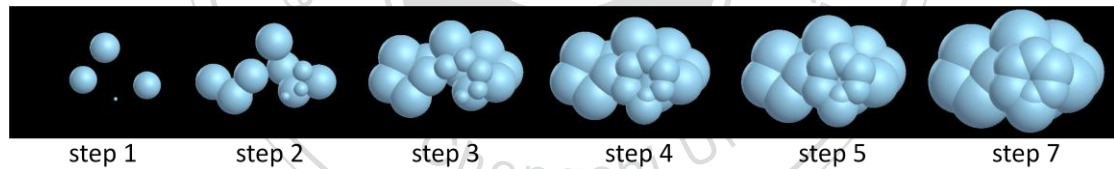


圖 4.18 朵雲紋 C4 系統生成之逐步結果

L-system 語法(C4) :

```
/* ----- C4 define ----- */
#define C4ang1 58
#define C4ang2 5
#define C4ang3 80
#define C4ang4 11
#define ford3 3

Axiom: [(+90)g(ford3)[-(150)C41(2,2,5,0)]]

      [gO(2)]

      [^(110)g(ford3)[&(110)g(-ford3/2)C4(1,1,6,T)]]

      [-(90)g(ford3)[C42(2,2,5,0)]]

/* Parameter : */
/* d0 : translate amount */
/* db : sphere size */
/* n : iteration count */
/* T : tail parameter */

C41(d0,db,n,T) : n > 0 →
  -(C4ang3)\(C4ang4)g(d0)[O(db)]C41(d0,ran(0.7)+db,n-1,T)
C4(d0,db,n,T) : n == 0 → [-(20)/(30)g(d0)][milt-tail(T,T)]
C4(d0,db,n,T) : n > 0 →
  +(C4ang1)\(C4ang2)g(-d0)[dO(db)]C4(ran(0.3)+d0,ran(0.3)+db,n-1,T)
C42(d0,db,n,T) : n > 0 →
  -(C4ang3)\(C4ang4)g(d0)[O(db)]C42(d0,ran(0.7)+db,n-1,T)
```

4.2.2 雲尾 L-system 語法

東方風格雲紋圖樣當中雲尾的走向與結構組合常常影響著雲紋的整體飄逸流動感，所以我們的系統在生成雲尾時可以想像成 L-system 在模擬植物的分支，一段一段的隨著時間變化拉長與角度變化的影響以生成出流暢的飄逸雲尾，在這裡系統實作出的雲尾模擬可以分成兩大類，A.尾部分離；B.尾部密合，以上兩種皆為套用 L-system 模擬植物枝幹莖的概念，根據語法規則的定義生長出平滑的雲尾。

「尾部-分離」：

由圖 4.19 可以看的出來雲尾分離的生成方式從最左邊開始，平面雲紋尾部分離圖樣的參考資料、利用矩形來模擬立體圓柱體之立體雲尾、最後系統所生成的尾部分離雲尾結果，從中間的模擬過程圖來看，要產生連續平滑圓柱體的銜接，我們必須將遞迴的次數拉高，才不至於產生圓柱體之間有銜接斷層的結果，以 L-system 的語法來說，產生單一條的連續平滑雲尾就定義為一個遞迴函式來處理，搭配語法來看的話 $tail(t)$ 這個函式就是針對平滑雲尾的遞迴呼叫生成連續圓柱體模擬雲尾，而另外一點則是分離雲尾的控制，勢必要有另一項參數與函式來重複的呼叫產生出多條的雲尾結果，搭配的函式則是 $milt-tail(t)$ 用於遞迴呼叫 $tail(t)$ 以產生多條的分離雲尾結果。

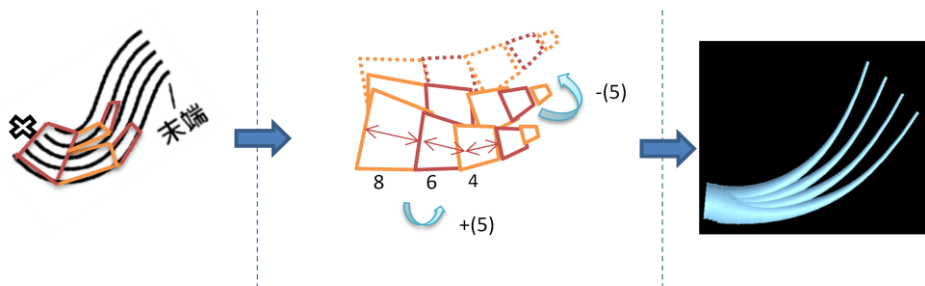


圖 4.19 尾部-分離雲尾生成流程圖

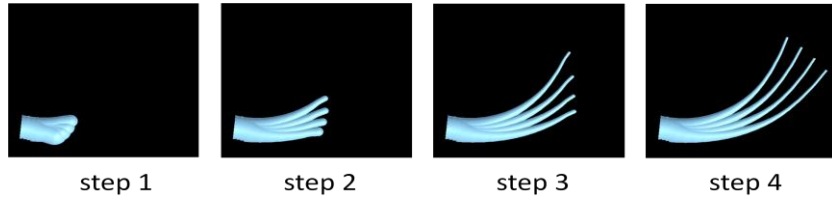


圖 4.20 尾部-分離雲尾生成流程圖

L-system 語法規則(tail - diverge) :

```

/* ----- TAIL ----- */
#define WIDS 1.5      /* cylinder start width */
#define WIDE 0.01    /* cylinder end width */
#define WR 0.85      /* cylinder width scale factor*/
#define MAX0 5
Define { array len[MAX0]= { 8, 6, 4, 3, 2 } ; }
/* define translate amount */

Axiom: multi -tail(4)

/* Parameter : */
/* t : iteration count */
/* n : translate amount index */
/* w : cylinder width control */

multi-tail(t) : t > 0 → -(5)g(0.1)[tail(t+3)] multi -tail(t-1)
tail(t) : t > 1 → tail_s(t) tail_e(15,WIDS)
tail_s(t) → #(WIDS)@Gt(1)@Gs#(WIDE)
tail_e(n,w): n > 0 →
+(5)f(len[n])#(w*WR)@Gc(60)tail_e(n-1,w*WR)

```

語法當中的許多參數都可以在更進一步的做調整，雲尾在與雲朵做搭配時，也會有其相對應的參數關係，所以在接下來下一小節立體雲紋結構的許多複雜雲紋圖樣結果當中其雲尾的參數也不盡相同，所生成出的雲尾更為多元化。

「尾部-密合」：

尾部密合的生成規則也類似於分離的語法，只是生長的順序從細變寬，從圖 4.21 之流程圖可以看出由 root 點開始向上，每一段圓柱體搭配相對應的位移量與轉角度，語法來看 $tail(t)$ 函式一樣是針對於圓柱體的遞迴呼叫並生成，並且在進行疊代同時會將圓柱體之寬度隨著呼叫次數而變寬，最後成為單一條由細變寬的雲尾，而 $milt-tail(t)$ 則是對於多重密合雲尾的呼叫控制，語法的概念與尾部-分離相近，主要差別在於其圓柱體之初始寬度與結束寬度設定不同。

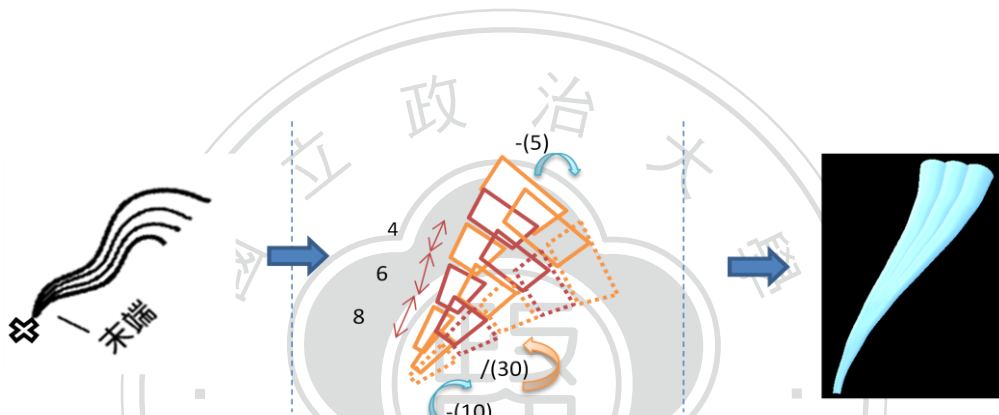


圖 4.21 尾部-密合雲尾生成流程圖

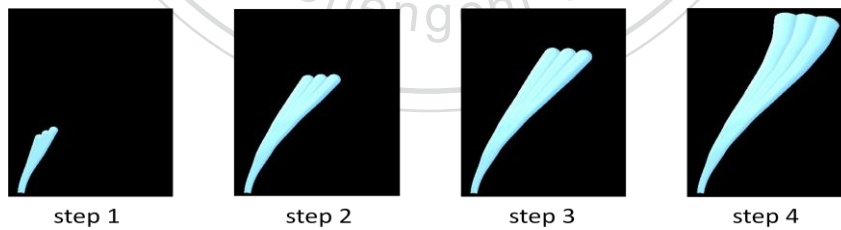


圖 4.22 尾部-密合雲尾生成流程圖

L-system 語法規則(tail – converge) :

```
/* ----- TAIL ----- */
#define WIDS 0.05 /* cylinder start width */
#define WIDE 0.3 /* cylinder end width */
#define WR 1.45 /* cylinder width scale factor*/

Axiom: multi -tail(4)

/* Parameter : */
/* T : tail cylinder count */
/* t : iteration count */
/* n : translate amount index */
/* w : cylinder width control */

milt-tail(T):T==1→#(0.3)@Gt(1)@Gs#(0.01)+(30)f(2)@Gc(60)
milt-tail(T):T==2→g(0.05)[tail(6)]+(3)g(0.05)[tail(6)]
milt-tail(T):T==3→-(5)g(0.05)[tail(6)]-(5)g(0.05)[tail(6)]
-(5)g(0.05)[tail(6)]
milt-tail(T):T==4→ ...

tail(t) → tail_s(t)tail_e(t,WIDE)
tail_s(t) → #(WIDE)@Gt(1)@Gs#(WIDS)
tail_e(n,w):n>0→ -(10)\(30)g(0.3)#(w*WR)@Gc(60)tail_e(n-1,w*WR)
```

4.2.3 雲紋結構

以上介紹過東方風格雲紋對於 L-system 之雲頭與雲尾的語法生長規則，接下來這一小節要介紹的是如何藉由結構化的方式將各式的立體雲紋組合起來，如下一頁表 5 可以很清楚的看出，在我們的研究當中所定義的結構總類共有十種，可以分為六個平面結構以及四個立體結構，藉由 L-system 最重要的遞迴概念，我們也可以將結構與結構之間的遞迴成長利用 L-system 的語法所生成出來。







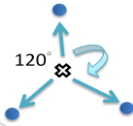
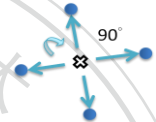

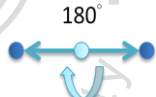

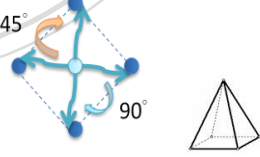
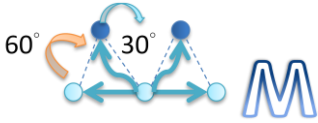
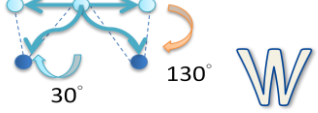


請參考表 5，符號表示 ：root 點；：畫雲紋/結構疊代點；：畫雲紋；：平面位移，首先針對 I. 平面結構依序地做介紹，編號 No.1 為生成最為基本的單一雲朵結構；編號 No.2 為從 root 點分別向左向右旋轉 90 度，結果則呈現出對稱 180 度的結構，考慮到結構與結構也可以重複疊代的特性，會以左右各 90 度以組合出對稱 180 度的結果；No.3 為從 root 點分別向上一段位移量，並從左右各旋轉 120 度再畫出雲朵，呈現出對稱 120 度的結果；No.4 為從 root 點向上一段位移量並且分別向左右各旋轉 90 度及下方旋轉 180 度再畫出雲朵，整體的組合則呈現出彼此對稱 90 度的結果；No.5 則是類似於山型結構，左右兩側各旋轉 130 度至下方，所夾之角度為 100 度對稱，並再往下重複一次以上動作，完成之結構就類似於三角狀的山形；No.6 則是較為簡單的直線型，初始位置上直接可以畫上雲紋並在左右兩側各旋轉 90 度畫上雲紋，而同時也是結構繼續疊代的位置，以上是平面雲紋結構的簡單介紹。

表 5 立體雲紋結構表

分類	結構編號	結構說明	圖示
一 二維平面結構	No.1	單一雲朵 SYM 1	
	No.2	對稱 180 度 SYM 2	
	No.3	對稱 120 度 SYM 3	
	No.4	對稱 90 度 SYM 4	
	No.5	夾角 100 度 SYM 5	
	No.6	直線型 SYM 6	
二 三維立體結構	No.7	正四面體 SYM 7	
	No.8	四角錐 SYM 8	
	No.9	M 型 SYM 9	
	No.10	W 型 SYM 10	

再來 II.立體結構的部分，由於立體結構包含許多的具有深度的旋轉量，所以在此再額外定義立體結構的符號表示，：具深度值的旋轉；：具深度值的位移；No.7 為正四面體，從正四面體的頂端出發，分別向上並往後方旋轉 45 度，再依序左右旋轉 120 度並再向後方旋轉 45 度，呈現出四面體頂端為第一階層，四面體底部的三個座標為第二階層的立體結構；No.8 四角錐的結構原理也類似於正四面體，從四角錐頂部出發，分別向上並往後方旋轉 45 度，再依序左右旋轉 90 度及向下 180 度並再向後方旋轉 45 度，呈現出四角錐頂部為第一階層，四角錐底部四個座標為第二階層的立體結構；No.9 與 No.10 為針對傳統東方風格雲紋所設計的立體結構，No.9 為 M 型，從起始位置畫一朵雲紋再依序從四個方向出發，分別向左右旋轉 90 度，與起始位置的雲朵為 M 型結構的第一階層，再往斜後方旋轉 60 度位移量接著分別旋轉左右各 30 度，這部分則是 M 型結構中的第二階層，呈現出來的感覺就是三朵的平面雲紋與後方的兩朵雲紋所組合而成的 M 型結構，No.10 為 W 型結構，結構方式的生成也類似於 M 型，差別在於 W 型為三朵平面雲紋與前方的兩朵雲紋組合而成。

4.2.4 立體雲紋 L-system 語法架構

接下來就必須將表 5 中之結構在空間中的幾何關係轉換到 L-system 的語法當中，也就是將前兩小節對於雲朵及雲尾的組合，以結構化的方式組合起來，呼應到第三章對於雲紋圖樣的分類來看，對於使用者來說只需要指定參數的動作即可產生出其相對應的雲紋結構、雲朵範圍、雲尾的總類，利用結構的方式我們可以透過不同的參數畫出單一雲朵、單一雲朵重複雲紋、單一雲朵複雜雲紋以及組合式雲紋，如下一頁圖 4.23 所示，左半邊為平面傳統雲紋圖樣，右邊則是相對應於要產生立體雲紋的 L-system 規則，從語法可以得知，透過一層一層往下迭代與組合，可將原本複雜沒有結構的雲紋圖樣慢慢組織建構起來。

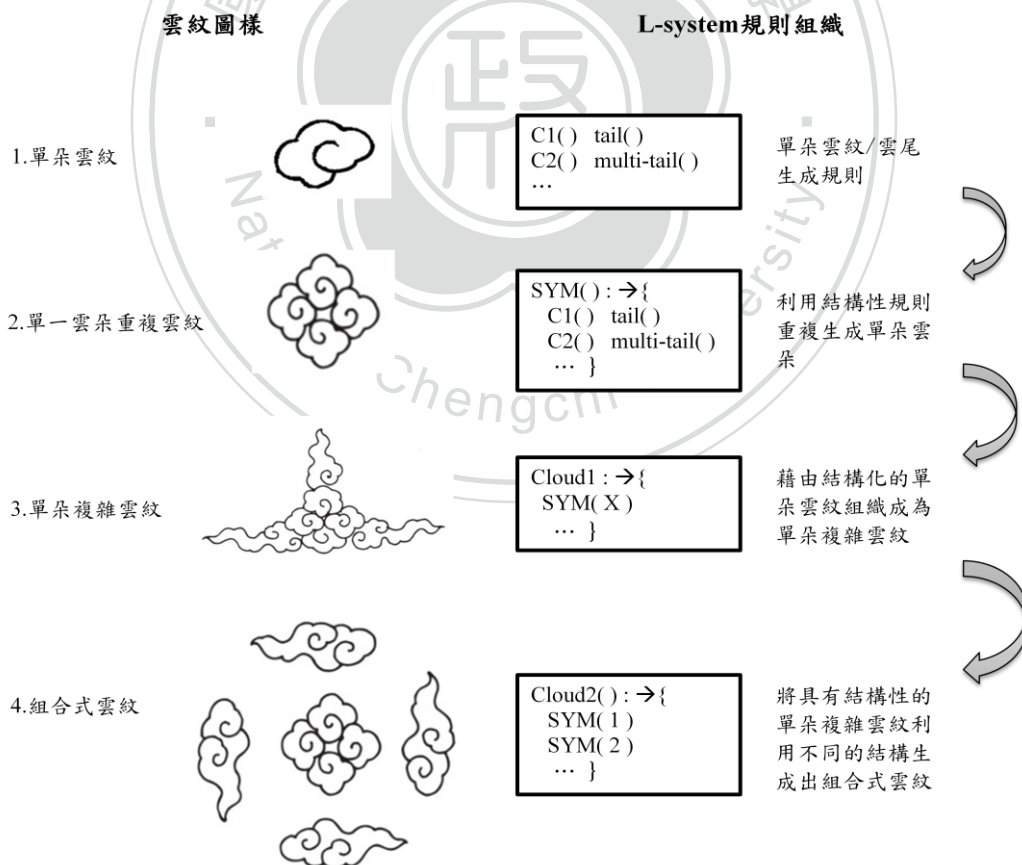


圖 4.23 將傳統雲紋圖樣之分類套用至 L-system 語法結構示意圖

以使用者來說只需要在 Axiom 指定使用 Cloud1()與 Cloud2()規則與參數的初始設定，接著系統即會根據規則的流程做生成，如下方的部分規則所示，Cloud1()用於重複生成相同結構的立體雲紋，括弧內的第三個參數即為指定在結構式 str1 之下做 recr 次的生成；Cloud2()用於將不同結構之雲紋做組合，括弧內的第五個參數即為指定要疊代下一個結構式指定，使用者可以考量自己的需求選擇參數做立體雲紋生成。

L-system 語法(System)：

```

/* ----- AXIOM ----- */
Axiom: Cloud1( x , x , x , x )

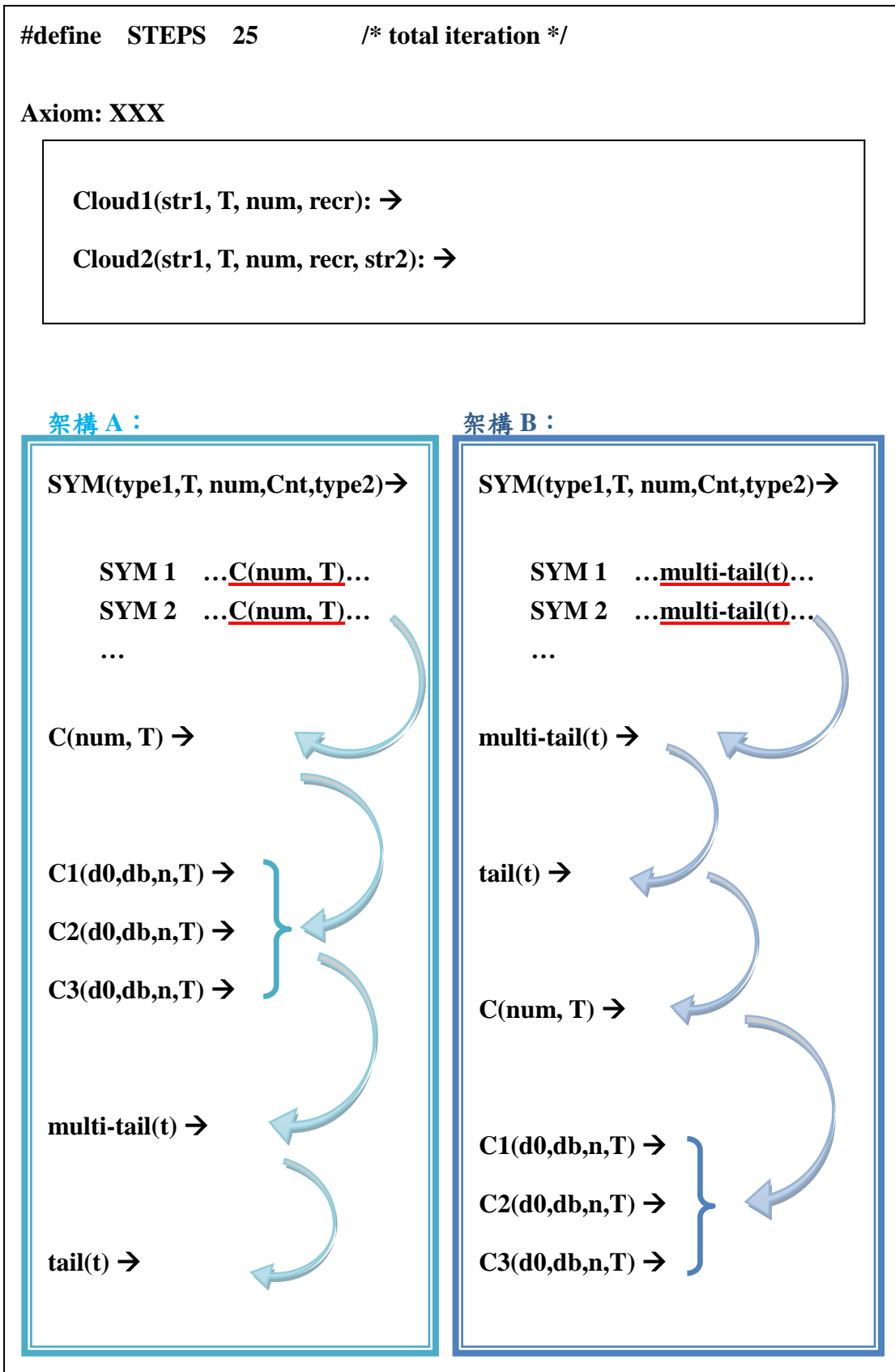
/* ----- CLOUD1 & CLOUD2 ----- */

Cloud1(str1,T,num,recr) --> SYM( str1, T, num, recr, str1)
Cloud2(str1,T,num,recr,str2): recr>0 --> SYM( str1, T, num, recr, str2)

```

我們的規則撰寫採取 parameter L-system 之語法結構，規則會根據其 condition 所給的限制，依序往下跑相對應的語法規則，如下一頁語法結構圖所示，我們系統會依照動態生長的順序分類，語法架構 A：雲朵生長→雲尾生長；語法架構 B：雲尾生長→雲朵生長，主要差別在於結構函式 SYM() 當中決定先生順序，舉架構 A 為例，根據結構的參數跑相對應的規則，語法執行順序 SYM() → C() → CI() → multi-tail() → tail()，這樣的動態生長會先生成出雲朵並且在雲朵生長完全後緊接著雲尾的生成；而架構 B 則其生成順序則是 SYM() → multi-tail() → tail() → C() → CI()，生長順序就會先順著雲尾拉長並在雲尾生成完成時同一時間團聚的雲朵也逐漸慢慢生長出來，兩者的語法類似但是動態的感覺與流暢度不盡相同，使用者也可以根據需求選擇何種動態的呈現。

東方風格立體動態雲紋之 L-system 語法架構：



以上是本研究的雲紋生長結構在 L-system 語法上的實作架構，接下來會針對實際的例子解說，從 Axiom 開始會根據指定的規則名稱與參數跑規則並呈現出動態的雲紋逐步生長結果：

Example 1

架構 A	Axiom : Cloud1(<u>1</u> , <u>0</u> , <u>1</u> , <u>1</u>)
參數 1 -	structure type 1 : str1 = 1 → SYM 1
參數 2 -	cloud tail : T = 0 → no cloud tail
參數 3 -	cloud type : num = 1 → C1-A cloud
參數 4 -	iteration : recr = 1 → recursion once

Front
Left
Right
Back

圖 4.24 結構生成雲紋圖樣 Example 1 之結果圖

Example 1 說明：

Cloud1 此函式為針對相同結構做遞迴生成，參數 1- str1 = 1 即為指定結構 SYM(1) 為單朵雲紋；參數 2 - T 即為指定雲尾之圓柱體個數，T = 0 表示無雲尾；參數 3- num=1 表示所生成雲朵編號 C1-A 的雲紋；參數 4 - recr = 1 則是表示執行規則一次。

Example 2

架構 A	Axiom : Cloud1(<u>2</u> , <u>1</u> , <u>1</u> , <u>1</u>)
參數 1 -	structure type 1 : str1 = 2 → SYM 2
參數 2 -	cloud tail : T = 1 → 1 level (1 diverge cylinder) cloud tail
參數 3 -	cloud type : num = 1 → C1-A cloud
參數 4 -	iteration : recr = 1 → recursion once

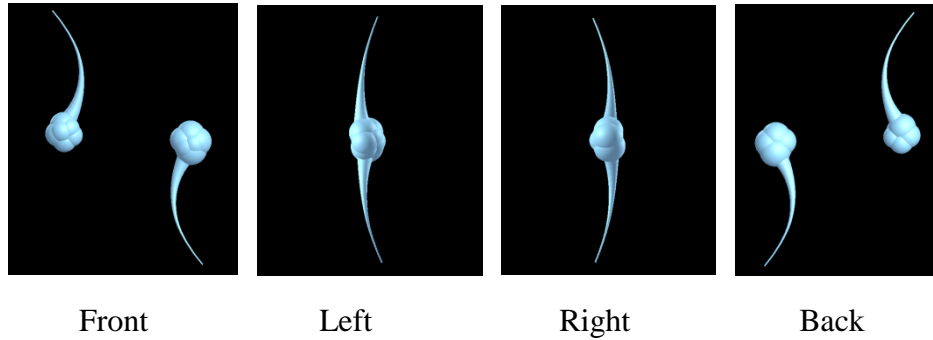


圖 4.25 結構生成雲紋圖樣 Example 2 之結果圖

Example 2 說明：

一樣為 Cloud1，參數 1- str1 = 2 即為指定結構 SYM(2)為對稱 180 度結構；參數 2 - T 雲尾之圓柱體個數，採用架構 A 尾部分離，T = 1 生成單一圓柱體模擬雲尾；參數 3- num=1 生成雲朵編號 C1-A 的雲紋；參數 4 - recr = 1 執行規則一次。

Example 3

架構 A	Axiom : Cloud1(<u>9</u> , <u>0</u> , <u>1</u> , <u>2</u>)
參數 1 -	structure type 1 : str1 = 9 → SYM 9
參數 2 -	cloud tail : T = 0 → no cloud tail
參數 3 -	cloud type : num = 1 → C1-A cloud
參數 4 -	iteration : recr = 2 → recursion twice

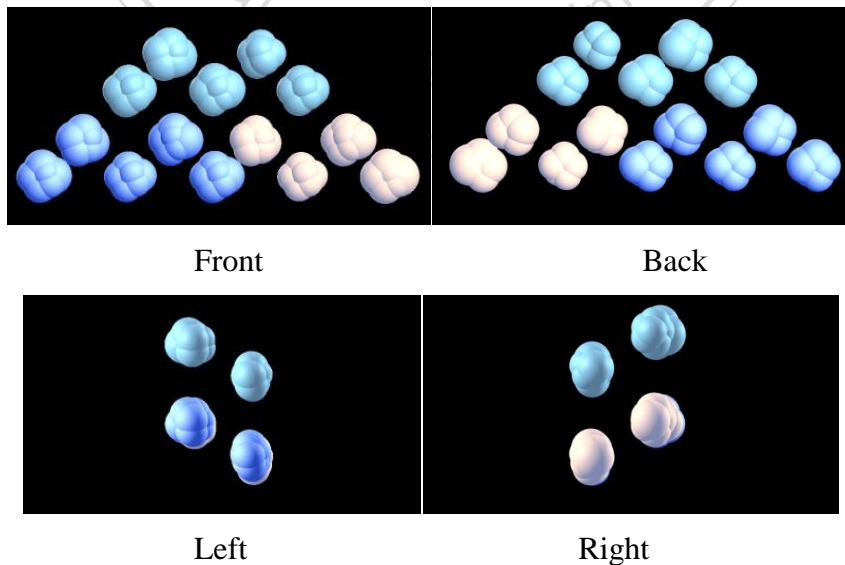


圖 4.26 結構生成雲紋圖樣 Example 3 之結果圖

Example 3 說明：

規則 Cloud1，參數 1- str1 = 7 即為指定結構 SYM(7)為 M 型立體結構；參數 2 - T = 0 無雲尾；參數 3- num=1 生成雲朵編號 C1-A 的雲紋；參數 4 - recr = 2 遞迴規則二次。

Example 4

架構 A	Axiom : Cloud2(<u>1</u> , <u>0</u> , <u>1</u> , <u>2</u> , <u>3</u>)
參數 1 -	structure type 1 : str1 = 1 → SYM 1
參數 2 -	cloud tail : T = 0 → no cloud tail
參數 3 -	cloud type : num = 1 → C1-A cloud
參數 4 -	iteration : recr = 2 → recursion twice
參數 5 -	structure type 2 : str2 = 3 → SYM 3

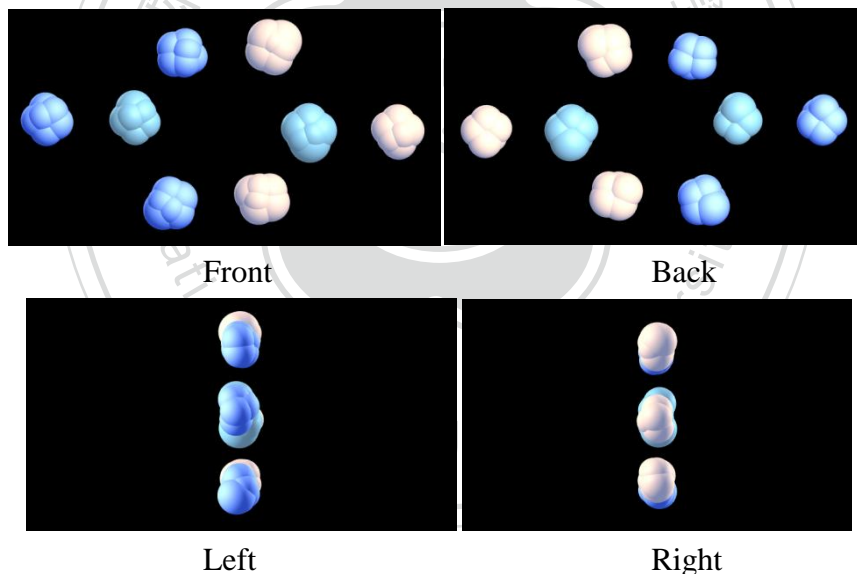


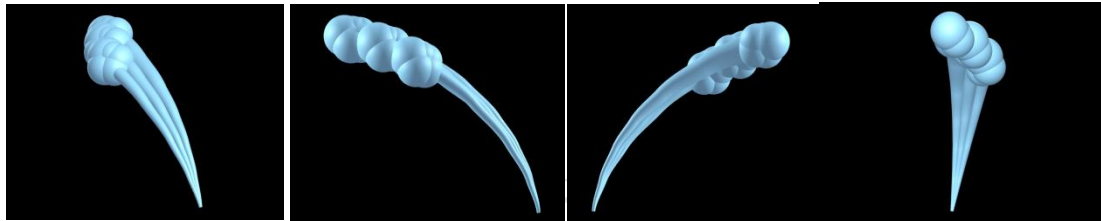
圖 4.27 結構生成雲紋圖樣 Example 4 之結果圖

Example 4 說明：

規則 Cloud2 用於生成雲紋結構與結構之間之組合，參數 1- str1 = 2 即為指定結構 SYM(2) 對稱 180 度結構；參數 2 - T = 0 無雲尾；參數 3- num=1 生成雲朵編號 C1-A 的雲紋；參數 4 - recr = 2 執行規則二次；參數 5 - str2 執行規則第二次所要迭代的結構參數，str2 = 3 表示第二次生成 SYM(3)對稱 120 度結構。

Example 5

架構 B	Axiom : Cloud1(<u>1</u> , <u>3</u> , <u>1</u> , <u>1</u>)
參數 1 -	structure type 1 : str1 = 1 → SYM 2
參數 2 -	cloud tail : T = 3 → 3- level (3 converge cylinder) cloud tail
參數 3 -	cloud type : num = 1 → C1-A cloud
參數 4 -	iteration : recr = 1 → recursion once



Front

Left 政治 Right

Back

圖 4.28 結構生成雲紋圖樣 Example 5 之結果圖

Example 5 說明：

規則 Cloud1，參數 1- str1 = 1 即為指定結構 SYM(1)為單朵雲紋；參數 2 - T 採用架構 B 尾部密合，T = 3 生成三條圓柱體模擬雲尾結果；參數 3- num=1 生成雲朵編號 C1-A 的雲紋；參數 4 - recr = 1 執行規則一次。

4.3 Dynamic System

Prusinkiewicz 在撰寫隨著時間生長的 L-system 5 規則[7]，其中所用到之參數 t 即為用於控制其時間軸之變化量，如圖 4.29 其生長規則所示，另外在[13]也提及要成功的模擬結構化的動態系統，有一個觀念稱為"step"，在 state 與 state 間做轉換並搭配時間軸在跑，則可以產生連續的動態雲紋變化。

```
/* L-system 5 */
#define LEFT -1
#define RIGHT 1
#define t_div 1.0 /* cell age at division */
#define t_s 0.0 /* initial age - short cell */
#define t_l 0.2 /* initial age - long cell */
#define dt 0.7 /* time increment per step */
Axiom: M(0,RIGHT)
M(t,p) : t+dt<t_div --> M(t+dt,p)
M(t,p) : t+dt>=t_div && p == LEFT -->
    M(t+dt-t_div+t_s,LEFT) M(t+dt-t_div+t_l,RIGHT)
M(t,p) : t+dt>=t_div && p == RIGHT -->
    M(t+dt-t_div+t_l,LEFT) M(t-t_div+t_s+dt,RIGHT)
```

圖 4.29 Prusinkiewicz[7]之 L-system 5 規則

所以將上述兩種觀念做結合，在雲紋 L-system 生長規則下將時間 t 與 step 的參數加入，幫助我們對於雲紋生長的方式更加有掌控度，另外由於在生長過程當中必須利用球體來模擬雲紋外觀，而球體本身可以利用其半徑長度做為控制大小的依據，所以隨著時間軸慢慢往後移動，可以在每增加一個 step 就增加或是減少球體的半徑長度，所以長時間下則可以觀察到立體雲紋的動態模擬，至於雲尾的變化上也可經由圓柱體寬度的變化，控制其圓柱體之比例以模擬出平滑飄逸的立體雲尾。

4.4 風格化

當立體雲紋有基礎模型後，為了使雲紋有更為多樣化的圖樣風貌，我們會進一步的對其進行風格化動作，使外觀上更貼近於東方風格雲紋，在此我們會以 shading 的方式及應用一些既有的影像處理方法來裝飾立體雲紋。在 NPR 這個領域當中，往往將利用光線與物體之間之角度之變化，對其作分區的 shading，共分為 Ambient、Diffuse、Specular，目前所採用的 shading 方式為 “Toon shading”，其顏色上的設置，會先利用深度值來判斷我們所要進行 shading 的層度，以建立出雲紋的立體深度，我們會將所生成之立體雲紋，利用其深度值的差異，會在不同深度的位置根據所設定顏色之 index 依序地將所訂的顏色順序呈現出來，可以很清楚地看出立體雲紋所呈現出的波浪紋路，如圖 4.30，在立體雲紋動態的結果呈現上 shading 效果更為顯著。

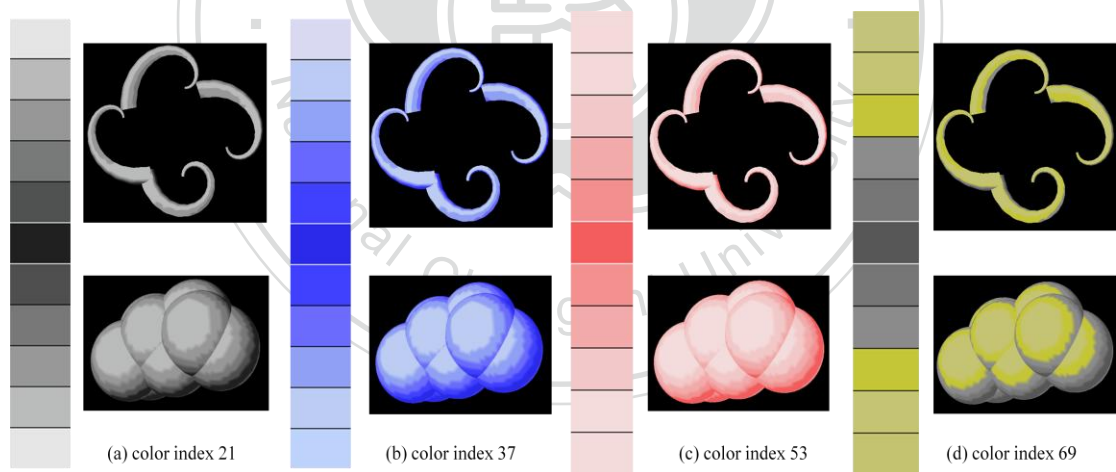


圖 4.30 以不同色調 shading 之立體雲紋結果

針對立體雲紋所產生出的最後的靜態結果我們也可以對其進行加上影像處理等不同的風格化處理，例如將所截取之圖檔加上 threshold，以及加上邊緣偵測等處理，則可以將原先的立體雲紋輪廓有轉化成為另外一種風格的平面式東方風格雲紋圖樣，如圖 4.31 所示。

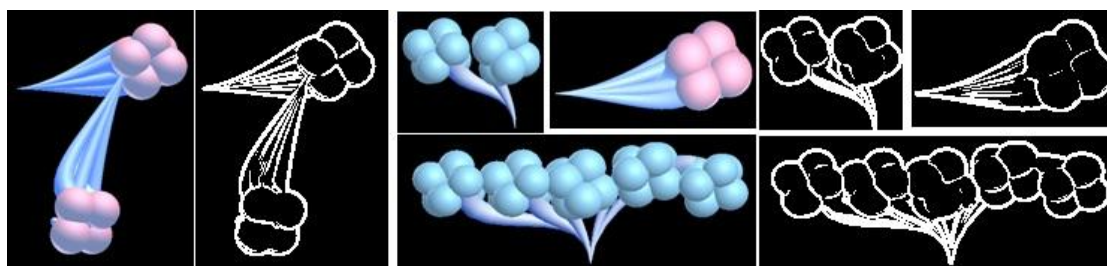


圖 4.31 經過影像處理過後之結果比較



第五章

實驗結果與討論

5.1 實作與實驗環境

我們目前的開發環境是加拿大卡爾加里大學(University of Calgary)中資訊工程學系 Virtual Lab 所開發的 L-studio 為平台[17]，使用其開發環境撰寫 L-system 之東方風格雲紋的語法規則。實驗環境上，使用的機器之 CPU 為 Intel(R) Xeon(R) CPU E5620 @ 2.4GHz，記憶體大小為 4.00GB，作業系統為 Window 7 64 位元作業系統，顯示卡為 NVIDIA Quadro 2000。


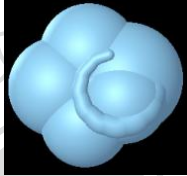







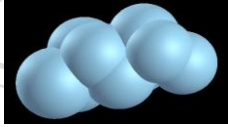


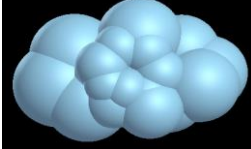

5.2 東方風格雲紋結果與分析

在這一小節當中，會從分類上較為簡單的雲紋與雲尾結果作一些比較與分析，且根據各式不同的平面傳統雲紋與系統實作出的立體雲紋作比較，最後則是經過風格化後所呈現出另一種不同的結果。

雲朵結果：

參考表 6 之結果比較，以平面雲紋圖樣為基礎，系統生成出朵雲紋 C1-A、C1-B、C2、C3、C4 立體雲紋之比較，以及經過 shading 後所產生的風格化結果，從表格可以很清楚的比較出相似度，在此章節之後的許多複雜型雲紋圖樣皆會以此為基礎，透過結構組合的方式產生更為多元化的立體雲紋結果。

表 6 朵雲紋結果分析與比較

編號	平面雲紋圖樣	系統生成立體雲紋	系統 shading 結果
C1-A			
C1-B			
C2			
C3			
C4			









另外從朵雲紋 C1-A 再作進一步的分析，藉由 L-system 語法參數的調控可以很清楚的由表 7 看出從不同的旋轉角度以及指定進行疊代的參數之不同組合，我們可以變化出許多不同的樣貌的立體雲紋結果，使用者也可以根據自行的喜好做選擇。

表 7 以不同遞迴次數與旋轉角度之組合生成之雲紋末端捲曲組合變化

		遞迴次數(Clend(x))		
		x = 8	x = 10	x = 12
角 度 變 化	$-(25)/(3)$			
	$-(30)/(5)$			
	$-(35)/(9)$			





C1-B 的變化如表 8 所示，在此利用 L-studio 之 Contours 勾勒出欲模擬之 Cylinder 曲線，系統所生成之立體雲紋輪廓則為其相對應的立體造型，再搭配朵雲紋雲尾溝捲的程度變化以組合出各式不同結果。

表 8 以不同風格的 Cylinder 樣貌搭配不同旋轉角度模擬之立體雲紋結果

Cylinder Style	ratate(10°)	ratate(12°)	ratate(14°)
			
			

另外如表 9 則是 C2 的變化擴充，大幅增加 STEPS 生成疊代次數的數值，使之模擬出螺旋感較強烈的捲曲立體雲紋，再利用 Scale 與 Rotate 之不同變數組合，創造出更多元化的結果，從這些結果可以看出，任何數值的變化組合，使立體雲紋結果更為多元化與風格化了。

表 9 以多次遞迴與角度的變化模擬出各式不同的捲曲立體雲紋

Cylinder Style	STEPS=60 Scale factor = 0.95 Rotate angle = -(20)	STEPS=55 Scale factor = 0.95 Rotate angle = -(15)	STEPS=45 Scale factor = 0.9 Rotate angle = -(18.6)
			

雲尾結果：

如圖 5.1 所示，皆為朵雲紋 C1-A 搭配不同雲尾參數所產生之結果，(a)為 multi-tail(1)搭配 tail(1)所生成之結果，為最基本型之雲尾；(b)其差別在於雲尾飄逸之長度的設定較長；(c)則是 multi-tail(5)尾部分離的雲尾結果；(d)為 multi-tail(6)尾部密合的雲尾結果，從這些結果比較即可看出雲尾參數的取決也影響了立體雲紋所呈現出的不同風貌。

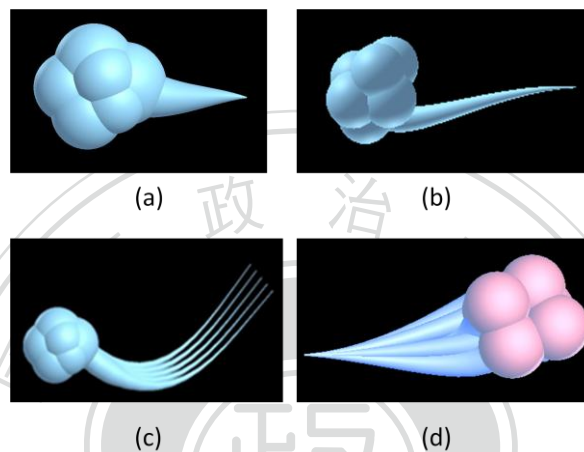


圖 5.1 朵雲紋 C1-A 搭配不同雲尾結果呈現
(a)單一雲尾- tail(1) (b)單一雲尾- tail(2)
(c)多條「尾部分離」雲尾 (d)多條「尾部密合」雲尾

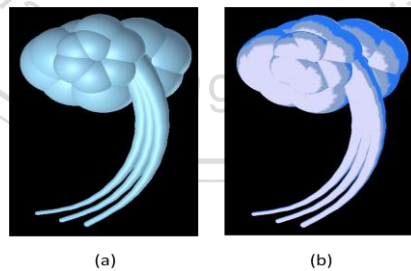


圖 5.2 (a)朵雲紋 C4 搭配尾部-分離之雲尾結果
(b) 經過風格化 shading 之結果

「尾部分離」的另一個例子，朵雲紋 C4 搭配 multi-tail(3)之分離雲尾組合如圖 5.2(a)，而圖 5.2(b)則是經過風格化 shading 之結果呈現，利用較大朵的雲紋，以及圖 5.3(a)則是朵雲紋 C1-A 搭配 multi-tail(2)之密合雲尾與 C3 之組合，(b)為利

用灰階色調 shading 之風格化結果，呼應到第三章當中對於雲尾的分析，圖 5.3 的雲尾串連起了兩朵分開的朵雲紋 C1-A 與 C3，以這樣的組合方式可以更加表現出複雜型的雲紋圖樣。東方風格雲紋圖樣當中對於雲尾的表現皆會以雲朵作為搭配，往往不會單獨的出現，所以在接下來更多的結果比較上也會以更多種不同參數所生成之雲尾與雲朵作結果呈現。

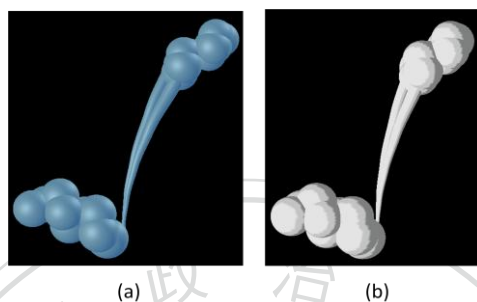
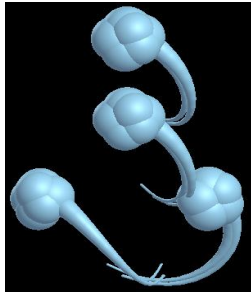
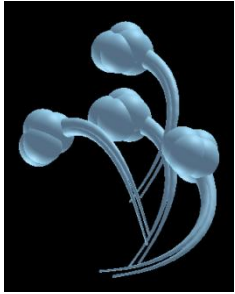
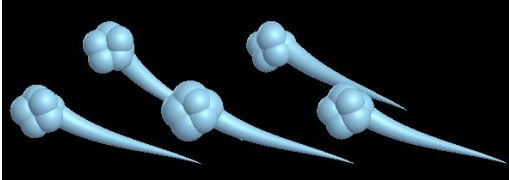
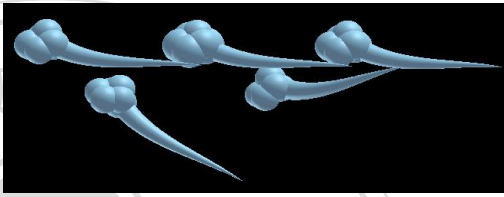


圖 5.3 (a)朵雲紋 C1-A 與 C3 利用尾部-密合雲尾組合之結果
(b) 經過風格化 shading 之結果

結構式立體雲紋結果：

表 10 多種不同結構之立體雲紋結果

二 維 平 面 結 構	對稱 180 度 SYM 2		
	對稱 120 度 SYM 3		
	對稱 90 度 SYM 4		

三 維 立 體 結 構	正四面體 SYM 7	 Top  Side
	M 型 SYM 9	 Front  Bottom

根據表 5 所列出的許多結構組合，可以將雲朵之排列以不同結構的組合呈現出來，如表 10 所示，即為根據所定義之結構語法在相對位置上作雲朵及雲尾的生成，表中可以分為兩大部分，平面結構在此以 SYM2、SYM3、SYM4 作不同參數產生出之立體雲紋比較，以 SYM4 對稱 90 度為例，兩張截然不同的立體雲紋圖樣皆是以 SYM4 結構作生成，但是雲朵與雲尾的生長我們則可以使用不同的參數作生成，由此可見，利用不同結構的排列以及雲朵及雲尾參數上的選擇則可以產生出多變化的立體雲紋結果；立體結構方面則是以 SYM7 及 SYM9 為例，立體結構的呈現必須以多視角之圖像看出其立體感，SYM7 之 Top 面可以清楚看出正四面體的四個頂點，由 Side 面可明顯看出正四面體的底面三朵雲紋位於深度較深的位置。

組合式雲紋結果：

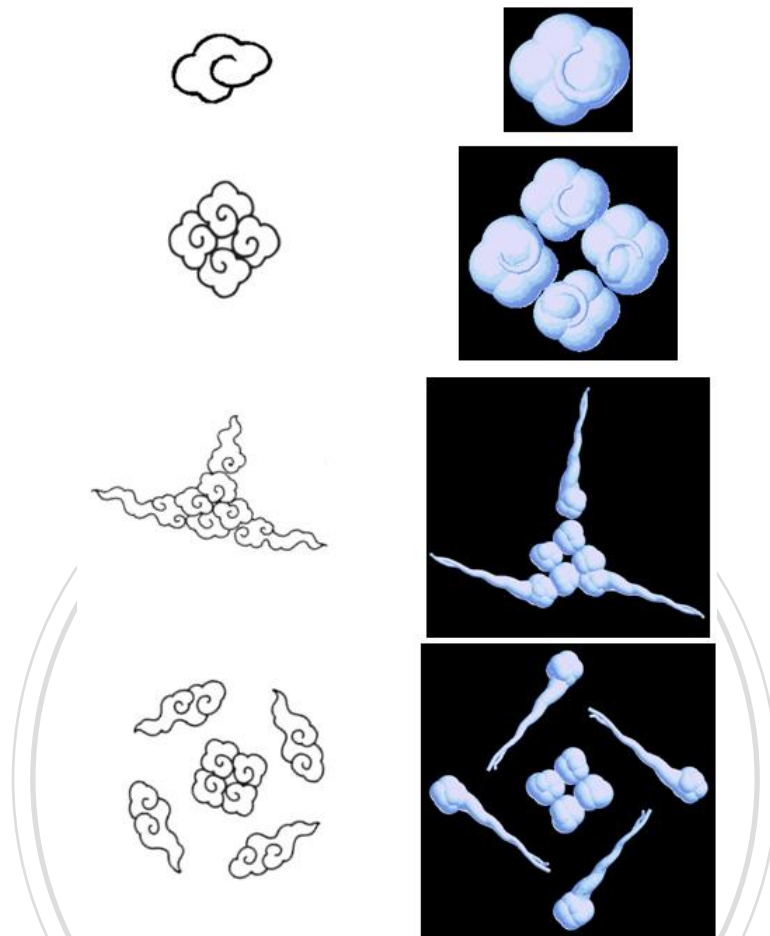



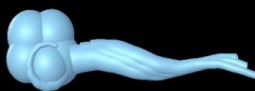


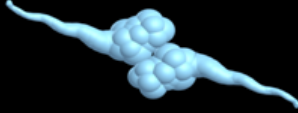


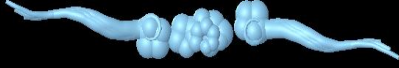


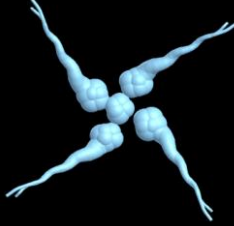
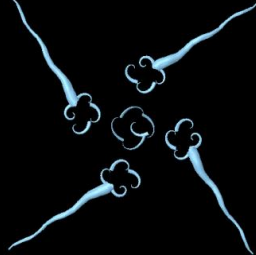
圖 5.4 以疊代式的 L-system 語法組合出不同類型的立體雲紋結果

參考圖 5.4 為根據上一章節當中圖 4.23 平面雲紋的分類搭配疊代 L-system 之語法規則的方法所生成之結果，從圖中由上往下來看：單一朵的 C1-A 朵雲紋、以 SYM4 所產生的對稱式團聚雲紋、以 SYM3 及 SYM4 搭配雲尾所共同組合而成的單朵複雜型雲紋、至利用雙層 SYM4 結構不同位移量所生成之組合式雲紋，利用此方法則可以將原本簡單至複雜之雲紋圖樣，透過 L-system 規則疊代生成的方式簡易的將立體雲紋建構出來。

各式平面雲紋圖樣與系統生成結果之比較：

以平面雲紋圖樣為參考，利用系統所定義之語法，所產生各樣式之結果整理表如表 11 所示，這些平面雲紋圖樣皆可將之做拆解，再利用我們所定義的結構組合而成，組合語法規則之參數取決於所對應之平面雲紋圖樣，以及系統所生成之結果比較。

表 11 平面雲紋圖樣與系統生成之立體雲紋整理表

平面雲紋圖樣	系統生成之立體雲紋與相對應之生成規則	
	 C1-A multi-tail(3) SYM1	 C1-B multi-tail(3) SYM1
	 C4 multi-tail(2) SYM2	 C2 multi-tail(2) SYM2
	 C4 multi-tail(0) SYM1 C1-A multi-tail(3) SYM2	 C2 multi-tail(0) SYM3 C1-B multi-tail(3) SYM2
	 C1-A multi-tail(0) SYM1 C1-A multi-tail(2) SYM4 Cloud1(SYM1 , SYM4)	 C1-B multi-tail(0) SYM1 C1-B multi-tail(2) SYM4 Cloud1(SYM1 , SYM4)

5.3 動態雲紋

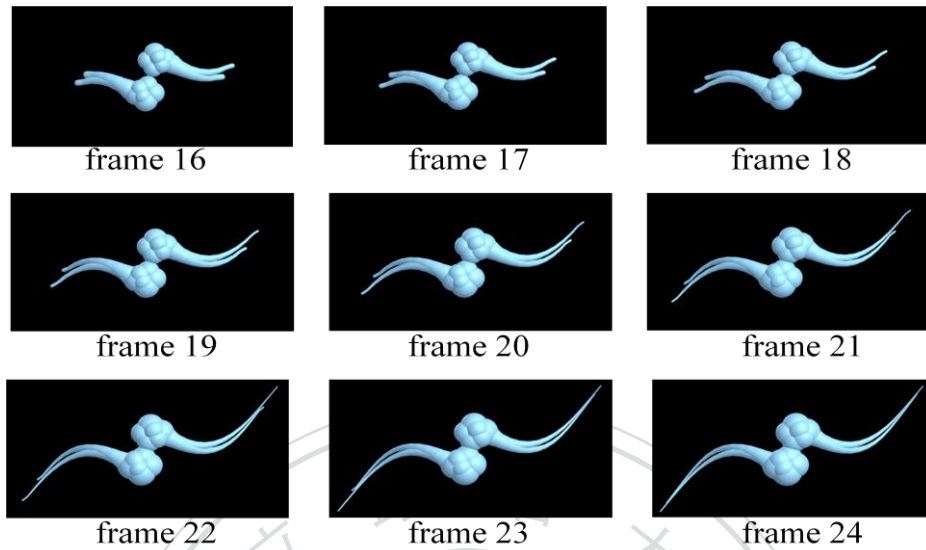


圖 5.5 以平面結構 SYM2 逐步生成的動態截圖結果

在這一小節當中主要將逐步生長的動態雲紋之變化在不同 frame 下所擷取之結果圖整理出來，以更清楚的方式呈現出動態的立體雲紋生長過程，從圖 5.5 的例子來看，以簡單的 SYM2 對稱 90 度結構搭配 C1-A 與飄逸雲尾的組合繪製而成，從其 frame 的逐步變化可以看出雲尾的生長可以使得立體雲紋更有飄逸感。圖 5.6 的例子以平面結構為基礎 SYM1 與 SYM4 之組合，並且加上風格化後所產生另一種不同風格的動態效果。圖 5.7 的例子則是以立體結構 SYM1 與 SYM1 之結構的組合，也因為其立體結構的變化較無法在論文當中清楚呈現，故提供了在不同視角所擷取的圖檔以供參考，可將系統逐步結果與風格化後之變化做比較，在較複雜的組合式雲紋中利用 shading 的色彩調整可以變化出許多不同風貌的結果呈現。

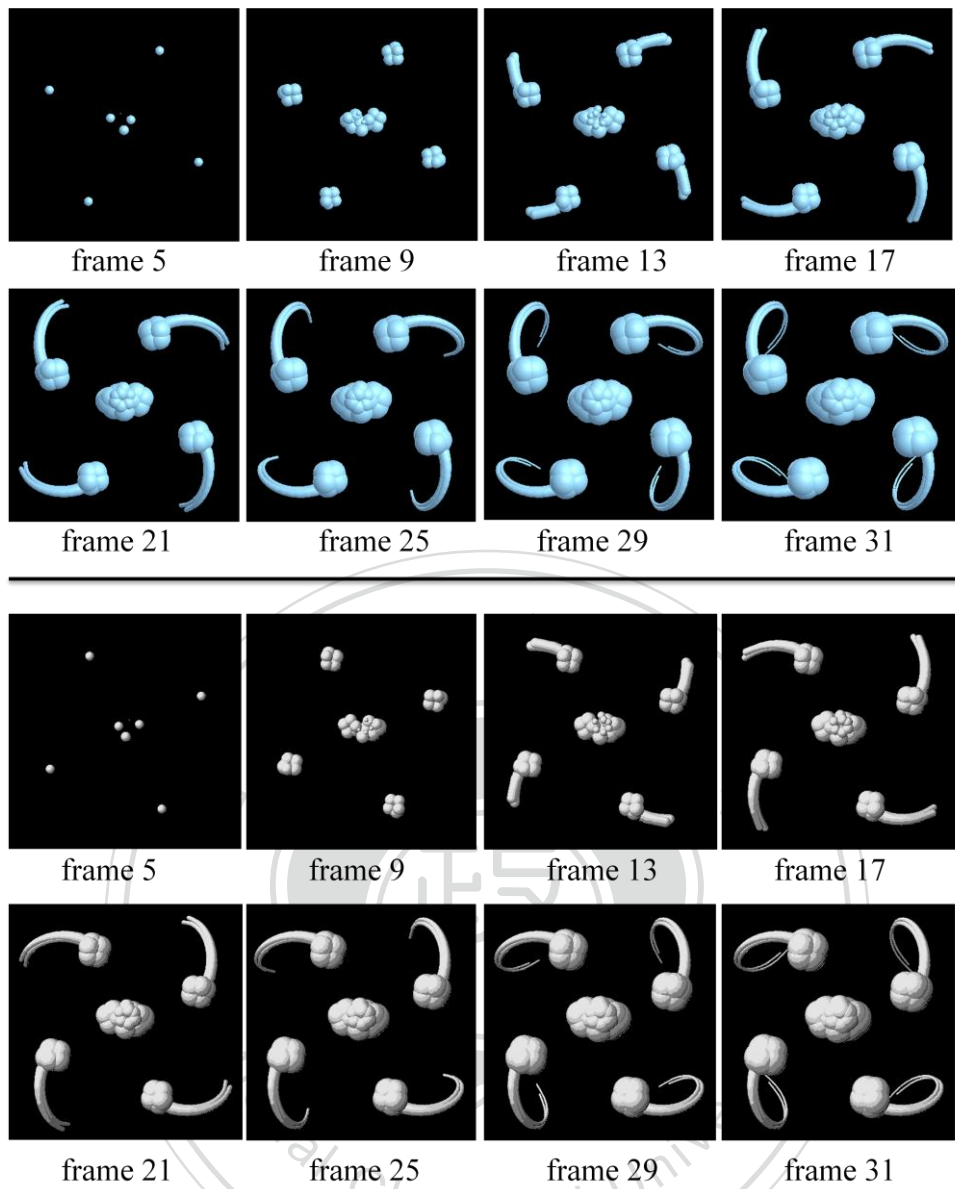


圖 5.6 SYM1-C4 與 SYM4-C1-A 平面結構之組合式雲紋

系統逐步生成與風格化 shading 動態截圖結果

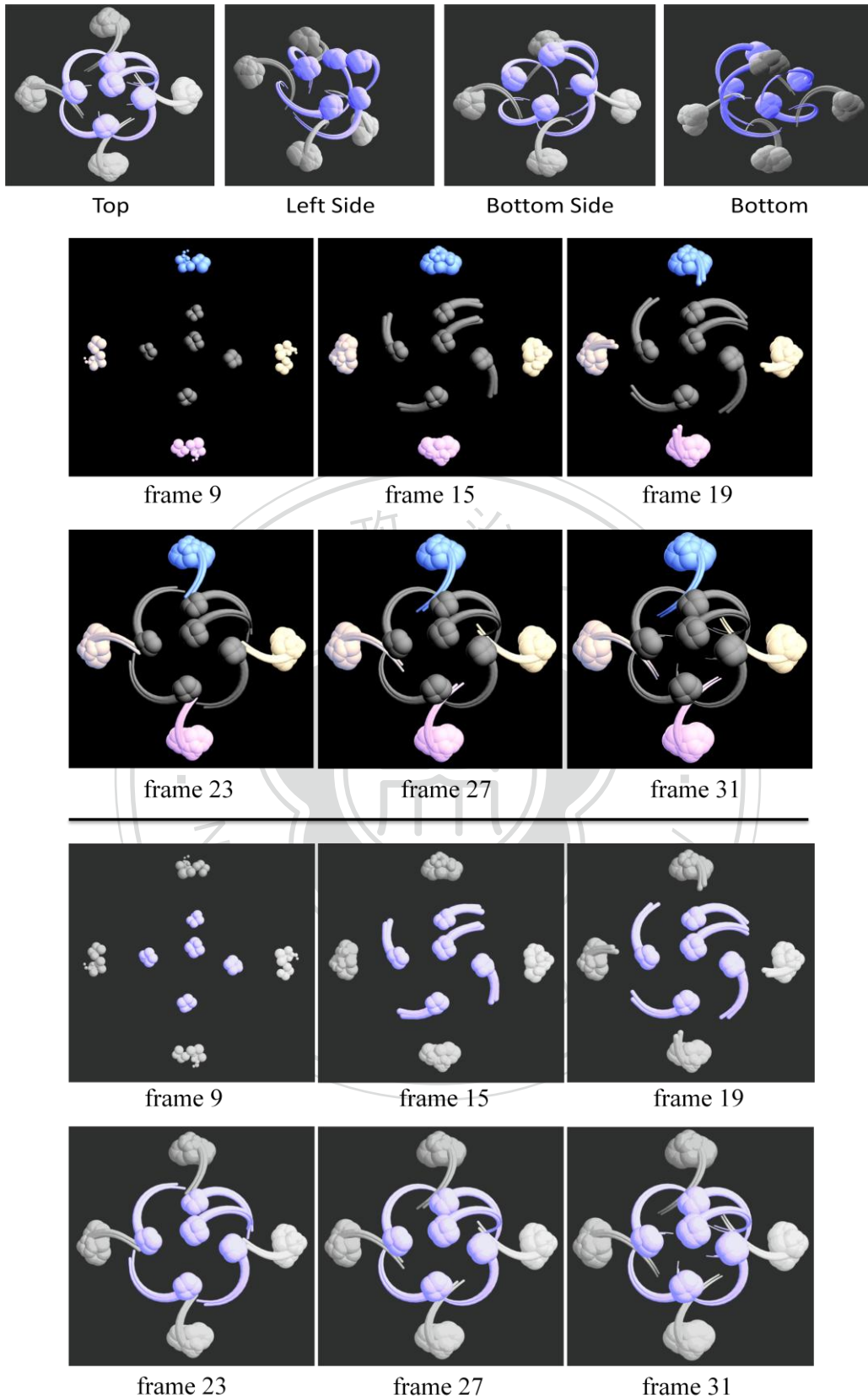


圖 5.7 SYM8-C1-A 與 SYM4-C4 立體結構之組合式雲紋動態截圖結果

5.4 效能評估

從前述的方法介紹我們可以藉由巢狀式的語法規則生成許多想要的結果，而在系統進行動態生成時如果隨著疊代的次數越高，由表 12 可以看出在不同的語法與疊代的次數不同對與動態結果其呈現的效能比較，系統動態結果的效能評估以 FPS(Average FPS)及動態生成總時間(Total time)做比較。

根據表 12 可得知立體雲紋在做動態生成時，其中若單純生成朵雲紋(Cloud)、朵雲紋搭配雲尾(C1-A with tail)，整體的生成具連續且平順的動態呈現，平均 FPS 均在 12 以上，總生成時間表現也尚可。

表 12 實際測試系統做動態呈現時之效能比較分析

評估類別	效能[Avg FPS(fps) Total time(s)]		
朵雲紋 Cloud	C1	C2	C3
	19.67 1.50	23.25 1.27	18.00 14.71
朵雲紋搭配雲尾 Cloud (C1-A with tail)	Level -1	Level -2	Level -3
	21.00 1.55	15.33 2.20	14.00 2.53
結構雲紋 SYM (C1-A no tail)	SYM 2	SYM 4	SYM 8
	14.85 2.04	9.67 3.09	9.75 3.61
疊代式結構雲紋 Cloud1 (SYM3-C1-A no tail)	once	twice	3 times
	13.33 2.55	4.29 6.76	2.69 14.67
組合式結構雲紋 Cloud2 (SYM3 – SYM2 C1-A no tail)	twice	4 times	6 times
	6.50 5.57	1.45 21.91	0.40 72.54

而結構雲紋(SYM)之效能也因不同結構本身之複雜程度而不同，在此舉的例子為SYM2、SYM4、SYM8，相對簡單的SYM2動態連續程度較佳；疊代式結構雲紋(Cloud1)則是明顯隨著疊代次數越高而影響了動態呈現品質，疊代三次以上就會產生不連續感；組合式結構雲紋(Cloud2)的動態結果則是明顯的隨著疊代次數越高，其不連續感越為明顯，建議使用者結構組合各一次為本系統可呈現出較佳的組合式結構式雲紋連續動態結果。

5.5 系統限制

由於開發環境目前為使用L-studio為平台，故在規則的撰寫上將研究的重心放在L-system規則的定義與撰寫，而我們用於模擬立體雲紋的方式也是利用L-studio平台所提供之既有的語法定義規則去撰寫雲紋生成規則，例如在模擬立體雲紋的幾何物體在這裡我們所採用的為“球體”，而是否有其他更多元的幾何物體更適合於模擬立體的東方風格雲紋圖樣，這個問題也是一個有趣且值得嘗試的比較與討論。另外一點則在於雲紋風格化的處理，在一開始生成立體的雲紋結果時所採用的著色方式是使用materials以觀察出立體感，而若要生成出具有shading效果的風格化結果，其上色方式則就要改為color map定義shading的色彩集，從前面小節的結果可以看出其成效，shading之方式仍然有改進的空間，在未來可以將之與OpenGL Shader作結合，似乎可以產生出更符合於東方風格立體雲紋的風格化結果。

第六章

結論與未來工作

6.1 結論

在本篇論文當中，針對所參考到的平面雲紋圖樣做整理與分析，觀察到雲紋其中也按照其某種規則與結構性的排列與組合，於是藉由所參考之平面雲紋圖樣定義出許多東方風格雲紋圖樣之分類，接著再將複雜的雲紋圖樣做切割，以 L-system 語法為基礎，將構成雲紋圖樣之基本單位，如朵雲紋與雲尾建構出來，接著再以重複疊代的方式將不同類型的雲朵做結合，使得複雜多變的雲紋可以藉著本系統提出的東方風格立體雲紋之 L-system 的語法規則串聯起來，而本論文提出了許多將東方風格立體雲紋做生成的許多參數與規則，整合出一套屬於立體雲紋生長的語法規則，藉由這套規則，可將原先平面的雲紋轉換為三維空間當中逐步成長的結構化東方風格立體雲紋，從原先看似複雜的雲紋圖樣當中，藉由我們所分析出的資訊，加入時間軸的概念，並且調整其動態生長的參數設定，將原本靜態的雲紋轉換成為動態的模擬，再搭配雲紋風格化後的處理，呈現出更多元的東方風格立體雲紋結果。

6.2 未來工作

由於東方風格的雲紋圖樣資料非常眾多，目前所提出之語法在實作上可能無法讓使用者完全地滿意所相對應符合的立體雲紋圖樣，在往後還會再嘗試更多不同的參數以接近符合於與傳統東方風格雲紋的結構比例，並再整合進 L-system 的語法當中可以更加擴充延伸至各式不同的組合式雲紋生成，對於雲紋結構的組合搭配，也可以再進一步地分析出哪些結構的搭配組合較符合於傳統東方風格雲紋。

由於目前的實作環境是以 L-studio 為基礎，對於動態的呈現以及風格化的處理實作上無法加以擴充，可以此語法結構為基礎，在 OpenGL 的環境下作進一步的研究與開發，使得針對風格化的處理更為彈性，以模擬建構出最貼近於東方風格的立體雲紋圖樣。

另外一點，對於許多藝術家而言，設計新圖騰往往仍然需要以他們本身的手繪方式最為直接，本系統目前在做初始值設定與參數調整仍然以輸入參數的方式做系統生成，在未來工作當中，必須更進一步地將 UI 介面建構出來並且以 Sketch-base 的方式做初始的雲紋輪廓外觀等指定，系統再根據這些輸入資訊作分析，將數據結合至 L-system 生長規則語法當中，則可幫助使用者更便利的設計出許多東方風格立體雲紋圖樣，而無須理會參數指定的麻煩。

參考文獻

- [1] 徐丽慧 郑军 《中国历代云纹纹饰艺》 人民美术出版社 2009 年 12 月
- [2] 詹毓君 東方風格雲紋圖樣生成技術 政治大學資訊科學系 100 碩士論文
- [3] 陳奕愷 基於 L-system 之動態裝飾雲紋生成技術 政治大學資訊科學系 100 碩士論文
- [4] Prusinkiewicz and Aristid Lindenmayer “The Algorithmic Beauty of Plants.” Springer Verlag New York, Inc., New York, NY, USA 1990
- [5] Przemyslaw Prusinkiewicz, Mark Hammel, Jim Hanan, and Radomir Mech. L-systems: from the theory to visual models of plants. In M. T. Michalewicz, editor, Proceedings of the 2nd CSIRO Symposium on Computational Challenges in Life Sciences. CSIRO Publishing, 1996
- [6] Przemyslaw Prusinkiewicz, Introduction to Modeling with L-systems, University of Calgary
- [7] HERTZMANN, A. 1999. Introduction to 3D non-photorealistic rendering: Silhouettes and outlines. In Non-Photorealistic Rendering. SIGGRAPH 99 Course Notes
- [8] McGuire, M., & Fein, A.. Real-time rendering of cartoon smoke and clouds. In NPAR '06: Proceedings of the 4th international symposium on Non-photorealistic animation and rendering (pp. 21 - 26). New York, NY, USA: ACM2006
- [9] A. Selle, A. Mohr, and S. Chenney. Cartoon rendering of smoke animations. In NPAR '04: Proceedings of the 3rd international symposium on Non-photorealistic animation and rendering, pages 57 - 60, 2004
- [10] Hampshire, N.. Dynamic animation and re-modelling of l-systems, Master's thesis, Bournemouth University, Bournemouth, UK 2009
- [11] Y.J. Yu, H.Y. Jung, H.G. Cho, A new water droplet model using metaball in the gravitational field, Comput. Graph. 23(1999) 213-222

- [12] Przemyslaw Prusinkiewicz , Structured Dynamical Systems, University of Calgary
- [13] Antoine Bouthors and Fabrice Neyret. Modeling clouds shape. In E. Galin M. Alexa, editor, Eurographics (short papers), august 2004
- [14] Fabricio Anastacio, Przemyslaw Prusinkiewicz, Mario Costa Sousa: Sketch-based parameterization of L-systems using illustration-inspired construction lines and depth modulation. Computers & Graphics 33(4): 440-451 2009
- [15] Yodthong Rodkaew, Somporn Chuai-Aree, Suchada Siripant, Chidchanok Lursinsap, Prabhas Chongstitvatana: Animating plant growth in L-system by parametric functional symbols. Int. J. Intell. Syst. 19(1-2): 9-23 2004
- [16] S. Manousakis. Musical l-systems. Master's thesis, Conservatoire Royal de La Haye, 2006.
- [17] L-studio <http://algorithmicbotany.org/lstudio/index.html>



附錄 A : 基於 L-system 之東方風格立體雲紋語法規則

```
/* ----- C1A define ----- */
#define C1Aang1 80
#define C1Aang2 11
#define C1Aang3 30
#define C1Aang4 5
#define CR 0.87
/* ----- C1B define ----- */
#define C1BR1 0.9
#define C1BR2 0.9
#define C1Bwid1 1
#define C1Bwid2 2
#define C1Bang1 9.0
#define C1Bang2 0.05
#define C1Bang3 12
/* ----- C2 define ----- */
#define C2R1 0.9
#define C2R2 0.92
#define C2wid1 1
#define C2wid2 1
#define C2ang1 14
#define C2ang2 0.5
#define C2ang3 6
#define C2ang4 0.5
/* ----- C3 define ----- */
#define C3ang1 60
#define C3ang2 80
/* ----- C4 define ----- */
#define C4ang1 58
#define C4ang2 5
#define C4ang3 80
#define C4ang4 11
```

```

/* ----- TAIL ----- */
#define WIDS 1.5
#define WIDE 0.01
#define WR 0.85
#define MAXNO 5
/* ----- SYM ----- */
#define ford 15
#define ford2 0.3
#define ori 0.07
#define ball_s 0.35

/* ----- System define ----- */
#define STEPS 25
Lsystem: 1
derivation length: STEPS
Define: {array len[MAXNO] = {8,6,4,3,2};}

/* ----- AXIOM ----- */
Axiom: Cloud1(x,x,x,x)
/* ----- CLOUD1 & CLOUD2 ----- */
Cloud1(str1,T,num,recr) → SYM(str1,T,num,recr,str1)
Cloud2(str1,T,num,recr,str2): recr>0 → SYM(str1,T,num,recr,str2)

/* ----- SYM ----- */
SYM(type1,T,num,Cnt,type2):Cnt>0&&type1==1 →
  [C(num,T)gSYM(type2,T,num,Cnt-1,type1)] /* 1 single cloud */
SYM(type1,T,num,Cnt,type2):Cnt>0&&type1==2 →
  [(+90)g(ford)[C(num,T)]SYM(type2,T,num,Cnt-1,type1)]
  [-(90)g(ford)[C(num,T)]SYM(type2,T,num,Cnt-1,type1)] /* 2 180 degree */
SYM(type1,T,num,Cnt,type2):Cnt>0&&type1==3 →
  [g(ford)[C(num,T)]gSYM(type2,T,num,Cnt-1,t)]
  [(+120)g(ford)[C(num,T)]gSYM(type2,T,num,Cnt-1,type1)]
  [-(120)g(ford)[C(num,T)]gSYM(type2,T,num,Cnt-1,type1)] /* 3 120 degree */
SYM(type1,T,num,Cnt,type2):Cnt>0&&type1==4 →
  [-(5)g(ford)[C(num,T)]gSYM(type2,T,num,Cnt-1,type1)]
  [-(95)g(ford)[C(num,T)]gSYM(type2,T,num,Cnt-1,type1)]
  [(+85)g(ford)[C(num,T)];(5)gSYM(type2,T,num,Cnt-1,type1)]
  [(+175)g(ford)[C(num,T)]gSYM(type2,T,num,Cnt-1,type1)] /* 4 90 degree */

```

```

SYM(type1,T,num,Cnt,type2) : Cnt>0 && type1==5 →
  [C(num,T)]
  [(130)g(ford)[-(130)C(num,T)]g(ford)[-(130)C(num,T)]
  ;(4)-(130)g(-ford)SYM(type2,T,num,Cnt-1,type1)]
  [-(130)g(ford)[+(130)C(num,T)]g(ford)[+(130)C(num,T)]
  ;(5)+(130)g(-ford)SYM(type2,T,num,Cnt-1,type1)]      /* 5 mountain cloud */
SYM(type1,T,num,Cnt,type2) : Cnt>0 && type1==6 →
  [C(num,T)]
  [(90)g(ford)[-(90)C(num,T)];(4)-(90)g(1.5)SYM(type2,T,num,Cnt-1,type1)]
  [-(90)g(ford)[+(90)C(num,T)];(5)+(90)g(3.2)SYM(type2,T,num,Cnt-1,type1)]
                                                                /* 6 line cloud */
SYM(type1,T,num,Cnt,type2) : Cnt>0 && type1==7 →
  [g(0.2)[C(num,T)]]
  [&(45)g(ford)[^(45)C(num,T)];(4)g(ford)SYM(type2,T,num,Cnt-1,type1)]
  [/(45)[+(120)g(ford)[-(120)C(num,T)];(5)g(ford)SYM(type2,T,num,Cnt-1,type1)]]
  [^(45)[-(120)g(ford)[+(120)C(num,T)];(6)g(ford)SYM(type2,T,num,Cnt-1,type1)]]
                                                                /* 7 Tetrahedron */
SYM(type1,T,num,Cnt,type2) : Cnt>0 && type1==8 →
  [g(0.5)C(num,T)]
  [&(45)[g(ford)[C(num,T)];(4)g(ford)SYM(type2,T,num,Cnt-1,type1)]]
  [/(45)[+(90)g(ford)[C(num,T)];(5)g(ford)SYM(type2,T,num,Cnt-1,type1)]]
  [^(45)[g(ford)[C(num,T)];(6)g(ford)SYM(type2,T,num,Cnt-1,type1)]]
  [^(45)[-(90)g(ford)[C(num,T)];(7)g(ford)SYM(type2,T,num,Cnt-1,type1)]]
                                                                /* 8 Square pyramid */
SYM(type1,T,num,Cnt,type2) : Cnt>0 && type1==9 →
  [C(num,T)]
  [(90)g(ford)[-(90)C(num,T)];(4)-(90)g(-ford)SYM(type2,T,num,Cnt-1,type1)]
  [-(90)g(ford)[+(90)C(num,T)];(5)+(90)g(-ford)SYM(type2,T,num,Cnt-1,type1)]
  [&(60)+(30)g(ford)[^(60)-(30)C(num,T)]]
  [&(60)-(30)g(ford)[^(60)+(10)C(num,T)]]      /* 9 M-type I */
SYM(type1,T,num,Cnt,type2) : Cnt>0 && type1==10 →
  [C(num,T)]
  [(90)g(ford)[-(90)C(num,T)];(4)-(90)g(ford)SYM(type2,T,num,Cnt-1,type1)]
  [-(90)g(ford)[+(90)C(num,T)];(5)+(90)g(ford)SYM(type2,T,num,Cnt-1,type1)]
  [^(130)+(30)g(ford)[&(130)+(30)C(num,T)]]
  [^(130)-(30)g(ford)[&(130)-(30)C(num,T)]]      /* 10 W-cloud */

```

```

/* ----- CLOUD ----- */
C(num,T) : num==1 → -(80)[C1A(2,2.5,5,T)]
C(num,T) : num==2 → C1B(T)
C(num,T) : num==3 → +(10)[C2(T)]
C(num,T) : num==4 → [^(70)[\ (60)C3(1.5,2.5,9,T)]
C(num,T):num==5 → [+ (90)g(3)[- (150)C41(2,2,5,0)]] [g(1)O(2)]
                [^(110)g(3)[&(110)g(-1.5)C4(1,1,6,T)]] [- (90)g(3)[C42(2,2,5,0)]]

/* cloud 1 A */
C1A(d0,db,n,T):n==0 →
    ^ (65)g(1.1)+ (90)\ (3)g(0.6)[- (65)&(55)C1Aend(10)] [milt-tail(T,T)]
C1A(d0,db,n,T):n>0 →
    - (C1Aang1)\ (C1Aang2)g(d0)[O(db)]C1A(d0,ran(0.5)+db,n-1,T)
C1Aend(c) : c>0 → C1Aend_s(c)C1Aend_e(c,1.6)
C1Aend_s(c) → # (0.2)@Gt(1)@Gs
C1Aend_e(c,w):c>0 →
    - (C1Aang3)\ (C1Aang4)f(0.9)#(w*CR)@Gc(10)C1Aend_e(c-1,w*CR)

/* cloud 1 B */
C1B(T) → [+ (55)g(4)[- (75)C1Bcurve1(1)]
          [- (40)g(4)[- (65)C1Bcurve1(1)]
          [+ (145)g(4.5)[- (70)C1Bcurve1(1)]
          [- (130)g(4)[- (60)C1Bcurve1(1)]
          [- (115)g(5)[- (120)C1Bcurve3(1)]
          [+ (155)g(5)milt-tail(T,T)]
C1Bcurve1(s) --> @#(4)#(C1BWid1)@Gr(1)@Gt(1.0)@GsC1Bcurve2(s)
C1Bcurve2(s) --> - (C1Bang1)\ (C1Bang2)C1BLine1(s)C1Bcurve2(s*C1BR1)
C1Bcurve3(s) --> @#(3)#(C1BWid2)@Gr(1)@Gt(1.0)@GsC1Bcurve4(s)
C1Bcurve4(s) --> - (C1Bang3)\ (C1Bang2)C1BLine2(s)C1Bcurve4(s*C1BR2)

/* cloud 2 */
C2(T) → [+ (55)g(3)[- (55)C2curve1(1)]
         [+ (145)g(3)[- (45)C2curve1(1)]
         [- (50)g(3)[- (30)C2curve1(1)]
         [- (225)g(3)[+ (30)C2curve3(1)]
         [+ (135)g(4)milt-tail(T,T)]
C2curve1(s) --> @#(1)#(C2Wid1)@GsC2curve2(s)
C2curve2(s) --> - (C2ang1)\ (C2ang2)C2Line1(s)C2curve2(s*C2R1)

```

```

C2curve3(s) --> @#(1)#(C2Wid2)@GsC2curve4(s)
C2curve4(s) --> +(18)/(C2ang4)C2Line2(s)C2curve4(s*C2R2)

/*      cloud 3      */
C3(d0,db,n,T): n==0 && T!=0 → [-(C3ang1)\(C3ang2)g(d0)][milt-tail(T,T)]
C3(d0,db,n,T):n>0 → \((C3ang1)+(C3ang2)g(d0)[O(db)]C3(d0,ran(0.2)+db,n-1,T)

/*      cloud 4      */
C41(d0,db,n,T):n>0→
-(C4ang3)\(C4ang4)g(d0)[O(db)]C41(d0,ran(0.7)+db,n-1,T)
C4(d0,db,n,T): n==0 → [-(20)/(30)][milt-tail(T,T)]
C4(d0,db,n,T):n>0→
+(C4ang1)\(C4ang2)g(-d0)[dO(db)]C4(ran(0.3)+d0,ran(0.3)+db,n-1,T)
C42(d0,db,n,T):n>0→
-(C4ang3)\(C4ang4)g(d0)[O(db)]C42(d0,ran(0.7)+db,n-1,T)

/*      -----      TAIL      -----      */
milt-tail(T,t):T==1&&t>-2→
[#(1.5)@Gt(1)@Gs#(0.1)-(50)\(10)f(8)@Gc(60)]milt-tail(T,t-1)      /* single tail */
milt-tail(T,t):T>1&&t>0→g[tail(t+6)]milt-tail(T,t-1)      /* multi tail */
tail(t): t>0 → tail_s(t)tail_e(2*t,WIDS)
tail_s(t) → #(WIDS)@Gt(1)@Gs#(WIDE)
tail_e(n,w) : n>0 → -(6)\(5)f(len[n])#(w*WR)@Gc(80)tail_e(n-1,w*WR)

/*      -----      SPHERE      -----      */
O(o) : o>0 → @O(o+ori)

homomorphism
tail_e(n,w) : n>0 → -(5)f(0.3)@Ge(60)@O(w)
C1Aend_e(c,w) → -(20)/(15)@Ge(80)@O(w)
C1BLine1(s) → g(s)#(s*C1BWid1)@Gc(5)
C1BLine2(s) → g(s)#(s*C1BWid2)@Gc(5)
C2Line1(s) → g(s)#(s*C2Wid1)@Gc(5)
C2Line2(s) → g(s)#(s*C2Wid2)@Gc(5)

endsystem

```