THE UNIVERSITY OF CALGARY

Modeling Biological Patterns using the Space Colonization Algorithm

by

Adam Runions

A THESIS
SUBMITTED TO THE FACULTY OF GRADUATE STUDIES
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE
DEGREE OF MASTER OF SCIENCE

DEPARTMENT OF COMPUTER SCIENCE

CALGARY, ALBERTA

January, 2008

THE UNIVERSITY OF CALGARY

FACULTY OF GRADUATE STUDIES

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies for acceptance, a thesis entitled “Modeling Biological Patterns using the Space Colonization Algorithm” submitted by Adam Runions in partial fulfillment of the requirements for the degree of Master of Science.

Supervisor, Dr. Przemyslaw Prusinkiewicz
Department of Computer Science

Dr. Faramarz Samavati
Department of Computer Science

Dr. Lawrence Harder
Department of Biological Sciences

Date

Abstract

This thesis introduces a class of algorithms for modeling biological patterns with branching (tree-like) and network (with loops) topologies. The key idea behind these algorithms is the marking and subsequent colonization of empty space. Models are formulated in terms of iterative geometric operations on sets of points representing the elements of the pattern and markers of free space. This concept is formalized as the space colonization algorithm.

The practical value of this approach is demonstrated by modeling the architecture of trees and vasculature in plants. Trees are modeled using markers of empty space to mediate competition between branches. When vascular patterns are modeled, the markers of empty space represent sources of a vein inducing signal (auxin). Several algorithms are introduced to simulate vein development in a growing leaf blade. Additionally, a model simulating vasculature patterning in the stem is proposed and used to examine the relation between phylotaxis and stem vasculature

The applications explored in this thesis demonstrate that a common mechanism, competition for space, is sufficient to recreate both the development of vascular patterns and the architecture of trees.

Acknowledgements

First and foremost I would like thank my supervisor Dr. Przemyslaw Prusinkiewicz. His guidance and support during this process have been invaluable. By sharing his expertise and enthusiasm he has had a profound impact, not just on the ideas described in this thesis, but on my development as a researcher.

Thanks to Faramarz Samavati and Lawerence Harder for serving on my examination committee and providing many insightful comments on the draft of this thesis. I did my best to incorporate the multitude of editorial comments from Dr. Harder to improve the final presentation of my thesis.

I would also like to thank my co-collaborators who made this work possible. My appreciation goes to Brendan Lane for his collaboration on the papers on which Chapter 3 and 4 of this thesis are based. The high-quality ray-traced renderings in Chapter 3 were created by Martin Fuhrer, who can not be thanked enough. The programing work done by Dr. Pavol Federl was of great importance to the work on veins, for which he deserves many thanks. For contributing her efforts and her vast biological knowledge I would like to thank Dr. Anne-Gaëlle Rolland-Lagan. Additionally, I would like to thank Anne-Gaëlle for providing the experiment performed using Mitchison's model in Chapter 4.

I would also like to acknowledge the generous financial support I received from the Natural Sciences and Engineering Research Council of Canada (NSERC) and the Alberta Informatics Circle of Research Excellence (iCore).

During my Masters I have been lucky enough to receive supported from numerous individuals. I cannot possibly name them all here, but there are a few who deserve special acknowledgements.

For many reasons I would like to thank Dr. Faramarz Samavati. It was his introductory course in computer graphics that sparked my interest in the subject. Our many discussions and subsequent work together has been a great pleasure, and enriched my studies immensely.

I'd also like to thank Dr. Enrico Scarpella. The time I spent in his lab helped to clarify many details of the biology of vascular patterning. For the time and effort Dr. Scarpella took to share his deep understanding of this topic I am truly grateful.

Thanks to all the members of the Graphics Jungle research group (and a few honorary Junglers) for making this such a pleasant experience, and just being awesome in general. In particular (but in no particular order) I would like to thank Mikolaj Cieslak, Fabricio Anastatacio, Richard Pusch, Joseph Cherlin, Pauline Jepp, Lakin Wecker, Helen He, Jason Chen, Richard Smith, Luke Olsen, Mahsa Eshraghi, Erwin de Groot, Carla Davidson, John Brosz and Colin Smith.

Finally, I would like to thank my family and friends for their constant encouragement during this process. I would like to acknowledge the incredible support both my parents have given me in all my academic endeavours, and my brother Cale for helping to keep me sane.

Table of Contents

Approval Page ii

Abstract iii

Acknowledgements iv

Table of Contents vi

1 Introduction 1
- 1.1 Organization of Thesis 7

2 Background 9
- 2.1 Models of biological pattern formation 11
- 2.2 Branching and network patterns 13
  - 2.2.1 Diffusion limited aggregation 13
  - 2.2.2 Laplacian Growth 15
  - 2.2.3 Biological models with growing tips 18

3 Modeling trees with the Space Colonization Algorithm[1] 24
- 3.1 The method 28
  - 3.1.1 Procedure overview 28
  - 3.1.2 The space colonization algorithm. 30
  - 3.1.3 Example 34
- 3.2 Design considerations 37
  - 3.2.1 Crown Specification 38
  - 3.2.2 Attraction point generation 42
  - 3.2.3 Modeling the branching structure of the tree 43
  - 3.2.4 Algorithm run-time 45
- 3.3 Exploration of the parameter space[1] 50
  - 3.3.1 Kill Distance and Attraction Points 50
  - 3.3.2 Radius of Influence 52
  - 3.3.3 Crown Shape 52
  - 3.3.4 Addition of attraction points during simulation 55
  - 3.3.5 Tropisms 56
  - 3.3.6 Competition 59
  - 3.3.7 Post-Processing 60
- 3.4 Discussion 61

4 Modeling leaf venation patterns I: Basic models with marginal and uniform growth[1] 69
- 4.1 Background and related work 71
  - 4.1.1 Leaf shape 71
  - 4.1.2 Taxonomy of leaf growth. 71
  - 4.1.3 Venation patterns 72
  - 4.1.4 Mechanism of vein pattern development 73
  - 4.1.5 Models of vein pattern development 76
- 4.2 Generation of open venation patterns 79
  - 4.2.1 Preliminaries 80
  - 4.2.2 The initial state of the model 82
  - 4.2.3 The simulation loop 82
  - 4.2.4 Example 84
  - 4.2.5 Calculation of vein width 85
- 4.3 Generation of closed venation patterns 87
- 4.4 Implementation of the algorithms 90
  - 4.4.1 Time complexity of the algorithm for open patterns 90
  - 4.4.2 Time complexity of the algorithm for closed patterns 91
- 4.5 Results 93
- 4.6 Conclusions 101

5 Modeling leaf venation patterns II: Accounting for areoles and non-uniform growth 108
- 5.1 Leaf growth and vascular development 109
- 5.2 Generation of the developmental sequence of venation patterns 112
  - 5.2.1 Modeling growth 113
  - 5.2.2 Source placement 118
  - 5.2.3 Generation of the structure of areoles 119
  - 5.2.4 Hypothetical progression of differentiation 120
  - 5.2.5 Example 122
  - 5.2.6 Refining vein nodes 124
- 5.3 Case study: The first leaf of *Arabidopsis thaliana* 126
  - 5.3.1 Development of patterns using the local-to-global growth model 126
  - 5.3.2 Development of patterns using key-frame surface interpolation 130
- 5.4 Conclusions 136

6 Modeling vascular patterns in stems 138
- 6.1 Background 139
  - 6.1.1 Vascular development in the stem 140
  - 6.1.2 Topology of veins in the stem 141

6.2 Generation of vascular patterns in the stem 142
    6.2.1 Placement of new vein nodes 142
    6.2.2 Visualization of vascular patterns in the stem 144
6.3 Results and discussion 148
6.4 Conclusion 151

7 Summary and future work 156
7.1 Summary 156
7.2 Future work 159
    7.2.1 Tree architecture 160
    7.2.2 Vascular patterns and leaf modeling 161
    7.2.3 Further extensions 162

Bibliography 165

A Proof of Theorem 1 188

List of Tables

3.1 Values of parameters used to generate selected models 68

4.1 Values of parameters used to generate venation patterns presented in this chapter 107

List of Figures

1.1 Examples of branching and networked patterns in nature 2
1.2 Examples of vascular patterns in leaves. 5

2.1 DLA and variants 15
2.2 Models of Laplacian growth. 19
2.3 Biological models with growing tips 23

3.1 Key steps of the proposed method for generating trees. 29
3.2 The relation between E and $\hat{n}$ visualized for two attraction points. . . 33
3.3 The relation between E and $\hat{n}$ visualized for three attraction points. . 33
3.4 Sub-figures illustrate the impact of varying $\vec{g}$ 35
3.5 The space colonization algorithm 36
3.6 A generalized surface of revolution and associated generating curves. . 41
3.7 A slice of the generalized surface of revolution shown in the previous figure 41
3.8 Decimation of a single branch. 43
3.9 Process run time plotted as a function of the number of attraction points (N). 46
3.10 Process run time plotted as a function of the number of nodes per unit ($1/D$). 48
3.11 Process run time plotted as a function of the kill distance (d_k). 49
3.12 Impacts of the number of attraction points, N, and the kill distance, d_k, on the tree form. 51
3.13 Impact of the radius of influence d_I on the form of shrubs. 53
3.14 Impact of the radius of influence d_I on the form of trees. 54
3.15 Impact of the envelope on the crown shape 55
3.16 A shrub generated with attraction points placed exclusively near the envelope. 56
3.17 A tree generated using continuously added attraction points. 57
3.18 The impact of strong tropisms on the resulting form. 58
3.19 Two trees competing for space 59
3.20 A hedge of shrubs competing for space. 60
3.21 The effect of post-processing a tree skeleton. 62
3.22 a) A birch tree. b) A tree with autumn foliage. 63
3.23 A tree with dense foliage. 63
3.24 A tree with compound leaves. 64
3.25 Four trees representing pine trees of different ages. 64
3.26 An Oak tree. 65

3.27 A tree with leaves. 66

4.1 Terms pertinent to the description of leaf shapes. 70
4.2 Varieties of leaf growth. 70
4.3 Some terms pertinent to the description of venation patterns. 72
4.4 Examples of venation patterns obtained from dicotyledons. 73
4.5 Examples of venation pattern in monocotyledons 74
4.6 Example of a venation pattern with actinodromous venation 74
4.7 Examples of tertiary and higher-order veins. 75
4.8 Hypothetical causal relations underlying vein pattern development. . 79
4.9 An experiment performed using Mitchison's model with two sinks and one source (courtesy of Anne-Gaëlle Rolland-Lagan). 81
4.10 Illustration of the algorithm for generating open venation patterns. . 86
4.11 Illustration of the notion of relative neighbourhood. 87
4.12 Comparison of the relative neighbourhood to the structure of highest order venation. 88
4.13 The Urquhart neighbourhood is only an approximation. 93
4.14 Parameter space exploration of venation patterns generated by the algorithms. 94
4.15 A photograph (left) and a rendered model of venation of a *Ginkgo biloba* leaf (right). 95
4.16 A photograph (left) and a rendered model of the venation of a lady's mantle (*Alchemilla mollis*) (right). 96
4.17 A model of a sweetgum (*Liguidambar styraciflua*) leaf. 96
4.18 A Nankin cherry (*Prunus tomentosa*) bough. 98
4.19 Patterns illustrating several emergent properties of the algorithms. . . 99
4.20 An orchid (*Bauhinia sp.*) tree leaf and a segment of a grass leaf. . . . 100
4.21 A reticulate venation pattern emerging in an isogonically growing leaf. 101
4.22 Grass leaves with venation patterns from Figure 4.20. 102
4.23 Two models of reticulate patterns of poplar (*Populus sp.*) leaves, compared to a photograph of a real leaf. 103
4.24 A simulation model of a poplar branch 104
4.25 A model of trillium flower. 105
4.26 Letters with procedurally generated vein patterns. 106

5.1 The sequence of developmental events observed in *Arabidopsis thaliana* based on [150]. 111
5.2 Modeling a growing leaf using local-to-global growth. 114
5.3 Modeling a growing leaf using B-spline surfaces. 117
5.4 Hypothetical sequence of vein development in a growing areole. . . . 120

5.5 Images from the modified model of vascular patterning in the leaf. . . 123
5.6 Adaptive vein refinement. 125
5.7 Developmental pattern simulated using the local-to-global growth model. 128
5.8 Final patterns produced by the local-to-global growth model. 129
5.9 Key frames representing the leaf at various developmental stages. . . 130
5.10 Developmental pattern simulated using key-frame interpolation of leaf blade development. 132
5.11 Final patterns produced by key-frame interpolation. 133
5.12 The final pattern produced when mid-point subdivision is applied . . 133
5.13 Mature pattern produced using key-frame interpolation shown without blindly terminating veins. 134
5.14 Mature pattern produced using key-frame interpolation shown with blindly terminating veins. 135

6.1 Diagram illustrating the anatomy of the shoot 140
6.2 A simple example using the model of vascular patterning in the stem. 144
6.3 Idealized diagrams of stem vasculature. 145
6.4 Visualizing patterns as in idealized diagrams. 146
6.5 Growth mapping created from data for the rosette of *Arabidopsis* [26]. 146
6.6 Application of mappings simulating growth and unrolling the stem to a sample pattern. 147
6.7 Two patterns of stem vasculature produced with different plastochrons but otherwise identical parameters. 149
6.8 Parameter space exploration where only the step between primordia (H) is varied, with $\theta = 137.5$ and $R = 115D$. 152
6.9 Parameter space exploration where only the radius of the stem (R) is varied, with $\theta = 137.5$ and $H = 10D$. 153
6.10 Parameter space exploration where only the radius of the stem (R) is varied, with $\theta = 99.5$ and $H = 10D$. 154

7.1 An ornamental kale leaf (Courtesy of P. Prusinkiewicz). 162

Chapter 1

Introduction

Nature abounds with patterns. Ubiquitous and inescapable, they provide texture to our surroundings, from the grand majesty of river networks to the elegant branching structure of trees and to the elaborate crack patterns observed in drying mud. Each class of patterns may exhibit remarkable variation while maintaining consistent global characteristics. Considering that these structures arise primarily through local interactions, they represent curious visual enigmas. Unravelling the apparent complexity of these patterns and their development into a small set of rules is a joy and a challenge. On one hand, such enquiries provide insights into the processes governing a pattern and help to connect disparate phenomena. On the other hand, from a computer graphics perspective, these investigations provide methods suitable for image synthesis in computer games and animation.

Two intriguing classes of natural patterns are those composed of linear elements arranged into *branching* or *network* patterns. Branching patterns have a tree-like structure and can be seen in the architecture of trees and the river drainage patterns (fig.1.1 top row). In contrast, the elements of network patterns reconnect to form cycles and loops, as is seen in leaf vasculature and crack-patterns (fig.1.1 bottom row). In many cases the arrangement of elements in these patterns arises from a number of spatial or functional constraints (e.g. each branch on a tree is constrained in its growth by the spatial arrangement of other branches [147, 109]).

For branched and networked patterns observed in biological systems an interest-

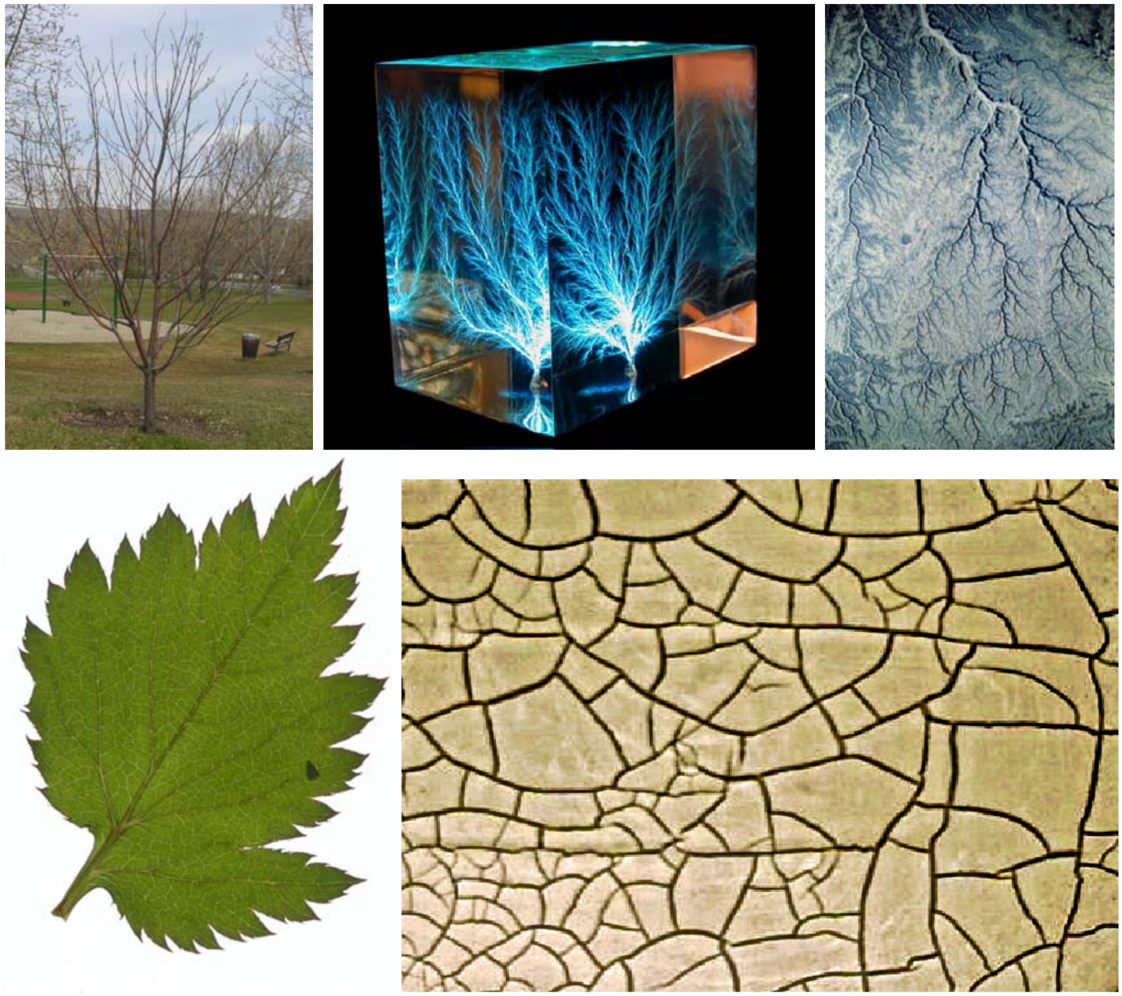


Figure 1.1: Examples of branching and networked patterns in nature. Top left to bottom right: Leaf less tree, Lichtenberg figure (a branching electrical discharge within an insulating material, here a block of clear acrylic) [59], river net in southern Yemen(Courtesy of NASA), leaf vein pattern, cracks in mud [35]

ing question is whether developmental and genetic factors can be represented abstractly as competition for space during growth. Historically, this question has been explored, for example, in the scope of studies of phyllotaxis, for which competition for space determines the position of new organs at the growing apex of a plant [158]. In ecology this question has been addressed for plant populations using spatial competition models [78, 83]. In the context of branched and networked patterns the current body of work is less extensive, and many questions remain unanswered.

Choosing the correct level of abstraction to answer these questions is difficult. The visual complexity and variability of the patterns addressed in this thesis, such as vasculature in leaves and the branching structure of trees, makes concise descriptions difficult. This limits the power of a quantitative or descriptive approach (such as that employed by Mündermann et al. to model *Arabidopsis*[106]). Conversely, simulating patterns at the cellular level based on molecular data suffices for recreating some features of observed patterns, but does not directly yield or relate to macroscopic observations [123].

In this thesis I investigate a new model for generating branching and networked patterns, using competition for space as the driving force of pattern formation. Interaction between elements is formulated geometrically, allowing developmental and genetic factors to be represented abstractly. Modeling phenomena at the geometric level provides a compromise between the extremes of descriptive and molecular-level models, which allows spatial relations between pattern elements to be expressed concisely and examined directly.

Working above the level of individual cells allows the algorithmic essence of patterns to be discussed without simulating cellular and molecular processes, the details of which can detract from higher-level observations. Additionally, processes that are different at the genetic level can be related at the geometric level, providing insight into factors common to distinct patterning processes. On a more practical note, geometric models tend to permit efficient implementations, allowing for easier experimentation, and are more intuitive than cellular and chemical models (for macroscopic observations).

The key-idea behind the approach presented in this thesis is to mark unoccupied space, which is subsequently colonized by extension of the existing structure into empty regions. This approach is formalized as the *Space Colonization Algorithm* (SCA). Unoccupied regions are represented by a set of points, which interact with the pattern. The pattern is extended iteratively towards the markers of free space, which are removed as the pattern advances, until the final pattern is produced. The algorithm is efficiently implemented using a Voronoi diagram to perform space subdivision [121]. The basic algorithm is extended to recreate several categories of biological patterns plausibly, in two or three dimensions.

Historically, the SCA was first applied to model leaf vein patterns [142]. These patterns are a functionally important and visually complex part of the plant. In this context, SCA simulates the interplay of three processes, which govern the development of leaf venation patterns: development of veins towards hormone (auxin) sources embedded in the leaf blade; modification of the hormone source distribution by the proximity of veins; and modification of both the vein pattern and source distribution by leaf growth. In this sense, the algorithm is biologically motivated.

To recreate leaf venation patterns, three variants of the basic algorithm are used: *the open algorithm* for recreating patterns when veins in a leaf blade do not recon-

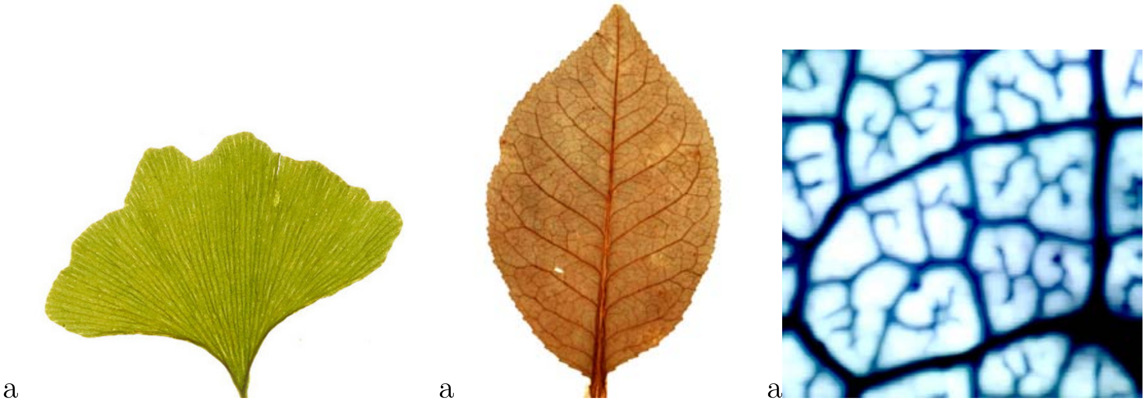


Figure 1.2: Examples of vascular patterns in leaves. Images demonstrating (left to right) an open pattern, a closed pattern, and the blindly terminating veins seen in otherwise closed patterns (final image courtesy of C.C. Chinappa).

nect (open patterns, fig.1.2a), *the closed algorithm* for recreating reconnecting veins that form a networked pattern (closed patterns, fig.1.2b), and *the areole algorithm*, which can generate blindly terminating veins observed in otherwise closed patterns (fig.1.2c).

Leaf growth plays an important role in vascular development [75, 149]. To take this into account, several techniques were used to simulate leaf-blade growth. Initial work focused on reproducing the final pattern of leaves, and the leaf was grown using accretive, or uniform growth. To recreate the temporal development of patterns more precisely required more sophisticated techniques, and two approaches were considered. The first, uses a growth field specified over the leaf-blade. By growing the leaf according to this field the form of the leaf emerges. Although technically possible, the observed forms of developing leaves were difficult to reproduce using this technique, so I developed a second, more controllable approach. The latter approach, takes as input the form of the leaf at various stages and blends the surfaces over time, to grow the leaf. This proved easier to control and allowed for the recreation of the developmental progression observed by Scarpella et al. [150] for *Arabidopsis thaliana*.

In addition to the visually conspicuous venation of the leaf, plants contain vasculature within the tissues of the stem. This vasculature is comprised of a network of veins originating at the base of the stem [34, 76]. Running the length of the stem, they connect all organs to the roots. I recreated these patterns by extending the closed algorithm for veins in the leaf blade to operate in the stem. Though the geometry of stem vasculature has not been well described in biological literature, the topology has been documented for several species [76, 84, 46, 165, 68] and is recreated by the proposed model.

Finally, I extended the algorithm for open patterns in the leaf to three dimensional branching structures, especially the branching structure of trees and shrubs [143]. Visual modeling of tree architecture is usually addressed by exploiting the recursive structure of tree development, often using a formalism such as L-systems [109, 128]. Additional factors, such as stochastic elements [25], shedding [109], tropisms, and mechanical bending are incorporated as modifiers [70] on a fundamentally recursive model. In contrast to these approaches, the SCA models trees using competition for space between branches as the dominant factor determining the form of trees and shrubs. This algorithm produces surprisingly realistic trees, with parameters corresponding to visually relevant tree characteristics identified in landscaping. Thus, offering convenient control of tree shape and structure. Although the method for generating trees was developed after the techniques for modeling vasculature, it is presented first in this thesis, as this application is conceptually simpler.

The correspondence of generated patterns to those observed in nature was primary validated using visual inspection. Objective validation of these patterns is a difficult and important problem and has only been partially addressed in the literature. For example, in the scope of leaves, the work of Bohn et al. provides relations between the branching angle and width of veins [13], and for branching structures an objective comparison is possible at the topological level using the edit-distance measures developed by Ferraro et al. [36]. Unfortunately, using these measures in a meaningful way requires real-world data. Acquiring sufficient data is an open problem in itself. Thus this topic was deemed to be beyond the scope of this thesis, but it is an important direction of future-work.

1.1 Organization of Thesis

This thesis comprises seven chapters. In Chapter 2, I provide a general background in models of pattern formation, emphasizing models that produce branching or network patterns (Section 2.2). In Chapter 3, I introduce a method for modeling trees. The space colonization algorithm is presented, and used to generate the branching structure of trees. An efficient implementation of the SCA using Voronoi diagrams is described, and generalized surfaces of revolution are proposed to model tree crowns (Section 3.2). In Chapter 4, the space colonization algorithm is adapted to model leaf venation patterns. In this chapter, I propose algorithms for generating open (branching, Section 4.2) and closed (networked, Section 4.3) venation patterns in growing leaf blades. In Chapter 5, I focus on precisely recreating the developmental progression of actual leaves, using *Arabidopsis thaliana* as a case study. The algorithm for generating closed patterns is reformulated to reflect the observed progression of differentiation (Section5.2.3). Then I introduce two techniques for simulating non uniform leaf growth (Section5.2.1) and discuss the resulting developmental progressions (Section5.3). In Chapter 6, I extend the closed model for leaves to the volume of the stem. Using this model the observed topology of several patterns is recreated (Section 6.3). Finally, Chapter 7 summarizes my contributions and discusses possible directions for further research.

Chapter 2

Background

Simulation models of pattern formation have been the subject of substantial research. This large body of research, and the variety of methodologies used to model the formation of natural patterns makes it difficult to place models of morphogenesis in a consistent frame work. Nevertheless, a characterization of this work is useful in discussing the relations between models. A classification of the features of models of pattern formation was proposed by Prusinkiewicz [122]. A subset of these features are described here, and aid in the discussions of previous work presented in the following sections.

The first category discussed by Prusinkiewicz differentiates models based on the topology of the elements comprising a pattern. Elements can have a linear topology, creating a non-branching *filament*, as in the developmental models of *Anabaena catenula* proposed by Hammel and Prusinkiewicz [53], and Cieslak [18]. Alternatively, elements may be connected into a *branching structure* (as in architectural plant models [128]), a *network* (graph with cycles) (as in models of vascular development [49] and fracture formation [35]), a *2D surface* (as in models of the pigmentation of animal coats [167, 170]), or a *3D solid object* (as the voxel automata used by Greene to model the details of tree architecture [50]).

Prusinkiewicz also distinguished between modes of *communication* between pattern elements. Expanding on a categorization proposed by Bell [8], he identified three key categories. The first mode operates through *lineage* with information transfered from an element to the elements that replace it. The remaining categories involve interactions, both between the elements of a pattern (*endogenous*) and between elements and their environment (*exogenous*). Endogenous communication occurs commonly in biological patterns when signals travel along specialized tissues; such as vasculature, and in plants enable spatio-temporal coordination of flowering [128, 66] and the hormonal control of bud fate [127]. Many natural pattens also represent responses to exogenous factors. For example, lichens have little internal coordination, but are affected by lighting and moisture [28]. The many tropisms in plants fall in this category, as well as the tendency of corals and sponges to grow towards dissolved nutrients [74]. Physical fields also provide environmental cues, such as the electrostatic field that guides lightning and determines Litchenburg figures [111]. Local properties of the medium surrounding a pattern, such as the stresses that govern crack formation, are also considered to be environmental cues. An analogous biological situation occurs when hormonal cues in surrounding tissues guide further development, as in axons [45] and vascular patterns in plants [144].

Finally, models may occupy constant space or expand (grow). Growth decreases the locality of information by creating distance between elements that were once close to each other. Leaf venation displays a global organization that implies a hierarchy of veins, but this global organization likely results from growth [75]. Similarly, phyllotaxis positions organs on the scale of a few cells, but through growth this arrangement governs the placement of leaves and petals on a much larger scale. Growth plays a significant role in the development of numerous natural patterns including the shells of sea urchins [95], the petals of *Anirrhinum majus* (snapdragon) [95] (see also [20]), and the wavy edge of many leaves [22].

In practise, determining the impact of factors in different categories can be difficult due to equivalences between categories at a theoretical level [122, 124, 57]. For example, determining the impact of exogenous vs endogenous factors in trees is difficult due to theoretical equivalences [61] (e.g. different mechanisms can produce the same patterns).

Prusinkiewicz examined these relations formally from the perspective of information flow during morphogenesis [124]. He showed that giving a pattern-forming agent access to environmental cues, internal cues, or incorporating growth all increased the range of patterns that the agent could produce. Additionally, he demonstrated that identical patterns could be produced using only one of these factors. Equivalences among categories can be exploited when modeling pattern formation, and understanding these equivalences is helpful when establishing relations between models.

2.1 Models of biological pattern formation

Choosing the appropriate abstraction when modeling a biological pattern is difficult, due to the diversity of phenomena and various scales at which they can be modelled. As a result, many formalisms have been developed to model the formation of biological patterns. A few of the most prominent are discussed below.

Lindenmayer considered the growth and development of organisms using a rewriting system called *L-systems* [86]. L-systems represent an object as a string of symbols, which is rewritten in parallel using a set of rules. These rules can specify the development of an organism over time. A wide range of biological structures, in particular plants [128], have been examined and recreated using L-systems. Many extensions of the formalism proposed by Lindenmayer have been developed, allowing for inclusion of parameters [126], stochastic phenomena, and environmental cues [127], to highlight a few. However, the descriptive power of L-systems is limited to linear and branching topologies. Thus, to simulate a wider range of topologies a number of generalizations have been developed. Lindenmayer proposed *Map-L-systems*, which extend L-systems to work on graphs with cycles [87], and Smith et al. [154] proposed another formalism in the form of a *vertex-vertex system* (*vv*) operating on 2D surfaces. Both extensions allow for the structure of cells to be represented, and in particular *vv* was used as the basis for a molecular model of phyllotaxis [157].

Another paradigm, emphasizing chemical interactions as the cause of morphogenesis, is *reaction-diffusion*, proposed by Turing [166]. In reaction-diffusion, patterns are generated by simulating the diffusion and interaction of chemicals, described by partial differential equations. This formalism can account for some natural patterns, such as the pigmentation patterns of animal coats [101, 167] and seashells [102, 39].

In contrast to the basic assumption of reaction-diffusion (that the chemicals responsible for establishing patterns propagate by diffusion alone), Sachs proposed that the hormone controlling differentiation of vascular tissues in plants may instead be actively transported out of cells [144]. Sachs formalized this observation as the *canalization hypothesis*, stating that veins form in a feedback process whereby auxin causes a cell to begin to differentiate, but differentiation increases the ability of a cell to transport auxin.

Based on Sachs' experimental results, Mitchison proposed two mathematical models [105]. The first models auxin transport as facilitated diffusion, whereby flux through a cell wall increases the wall's permeability. The second models polar auxin transport, whereby flux of a hormone through a cell wall further encouraged the flux of hormone in the same direction (allowing auxin to be transported against the gradient of diffusion). Subsequently, Mitchison's models have been the subject of much study [140, 38, 37] establishing correspondences with molecular data. Mitchison's model and its variants are of great relevance to modeling vascular development, and are discussed in greater detail in Chapter 4.

A complementary formalism to reaction-diffusion and polar-transport is that of *positional information*, proposed by Wolpert [178]. According to this model, cells have access to positional information, in the form of a coordinate system, based on their genetic and developmental history. This approach was used to explain the development of limbs in vertebrates [180], and most notably the segmentation of *Drosophila* embryos [179].

Branching and network patterns, in both physics and biology, have garnered significant attention, leading to the development of a more specialized set of techniques and formalisms. These are of particular relevance to the models presented in this thesis and are discussed in more detail in the following section.

2.2 Branching and network patterns

2.2.1 Diffusion limited aggregation

Many structures, such as bacterial colonies [94], lichens [28], and metal-particle aggregates [177], can be modeled as a random aggregation of elements. This process was formalized mathematically by Witten and Sander [177] as *diffusion limited aggregation* (DLA). DLA simulates the formation of an aggregate from a single immobile particle. The aggregation grows by adding mobile particles iteratively. Each mobile particle performs a random walk until it is adjacent to an immobile particle, when it sticks to the aggregate, becoming immobile. As the structure grows, complex branches form, blocking interior regions from wandering mobile points. Consequently, the likelihood that exposed branches will be extended increases, creating a feedback of the growth of a branch on itself (i.e. as the branch grows it is encouraged to grow further). This process creates an irregularly branching structures (fig.2.1a). The emergent properties and simple formulation of DLA make it possible to achieve different patterns by slightly modifying the basic model and have lead to the creation of many variants based on Witten and Sander's original model. The following discussion is limited to the variants of particular relevance to the work outlined in this thesis.

Desbenoit et al. [28] modeled the formation of lichens using Open-DLA (similar connotation to Open L-systems [109]) using environmental cues, such as the local density of pattern elements and access to sunlight and moisture, to allow elements to interact over a greater distance. These cues are used to modify the probability of aggregation for wandering particles in diffusion limited aggregation (fig.2.1d).

A variant of DLA proposed by Meakin [99, 98] uses internal cues to increase the distance over which interactions occur. The aggregation initially forms as in DLA, but branches of the aggregation that grow slower than a given threshold are removed. The growth at each point is determined by summing the number of particles added to the sub-tree of the element during a given number of iterations. The removal of slow-growing branches creates a more hierarchical structure in the final pattern (fig.2.1 b). Meakin's variant allows communication across arbitrary distances in the

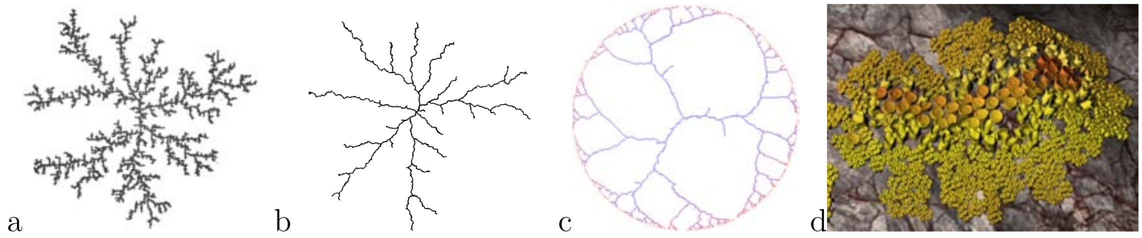


Figure 2.1: DLA and variants. a)DLA [177] b) Meakin's variant [99] c) Roberts' variant [135] d)Open-DLA (lichen) [28]

aggregate.

The distance over which interactions occur can be extended similarly using environmental cues; as in Roberts' model [135] (fig.2.1c). This model incorporates the assumption that a new particle immediately affixes to the closest side of the nearest particle in the aggregation. This process is equivalent to assuming that the particle will join the aggregation at the expected, or most probable, destination of its random walk. Thus, each particle in Robert's model can interact with another particle in the aggregate, regardless of the distance separating the two. Like the patterns produced by Meakin's model, a more pronounced hierarchy emerges, where the feedback of a branch's growth on itself, as seen in DLA, is amplified. Roberts' work served as an initial inspiration for the SCA, particularly the image shown in figure 10 of [135], which resembles leaf vein patterns near the margin.

2.2.2 Laplacian Growth

Diffusion-limited aggregation is one example of a broader set of patterns generated by *Laplacian growth.* Laplacian growth is related to the Laplace equation

$$\nabla^2\phi = 0,$$

where ϕ is a scalar field defined over the space of the simulation (making $\nabla^2\phi$ the sum of all the unmixed partial derivatives of ϕ). Here, the elements of the pattern serve as one of potentially many boundary conditions on the Laplace equation. The pattern represents an evolving interface, which responds to the Laplacian field ϕ. Laplacian growth can be defined algorithmically, as is specified in Alg.1 and discussed below.

The simulation takes an initial pattern, often referred to as the seed, as input. Boundary conditions are initialized using the elements of the seed (in addition to a number of boundary conditions associated with the phenomena being modeled). The Laplace equation is then solved with respect to these boundary conditions over the space of the simulation, yielding the function ϕ (line 2). The pattern is extended using ϕ (line 3). The new element(s) of the pattern are now considered to be part of the boundary conditions and change the solution to the Laplace equation (line 4). The simulation of growth continues by solving the Laplace equation with respect to the new boundary conditions and extending the pattern as described in lines 2-5. This process is repeated until the final pattern is produced.

```
1  Initialize boundary conditions including the initial pattern
2    Solve the Laplace equation with respect to boundary conditions to obtain $\phi$
3    Extend the pattern based on $\phi$
4    Add the new element(s) to the boundary conditions
5    Goto step 2
```

Algorithm 1: Algorithm for Laplacian growth

Meakin proposed a model of Laplacian growth simulating the differentiation of a tissue towards a point source of hormone [97] on a square lattice. The source produces hormone at a constant rate, and differentiated tissue removes all the hormone locally, establishing a diffusion gradient between source and differentiated tissue. The hormone-concentration field must be recalculated every time the pattern is extended. During each iteration the pattern is extended by one cell. First a cell adjacent to a differentiated cell is chosen. The cell then differentiates with some probability (dependent on the concentration of the hormone in the cell). Varying the impact of concentration on this probability produces a range of patterns from a circular Eden cluster [33] to a single branch connected to the source.

Several physical phenomena are examples of Laplacian growth. A particularly well studied form of Laplacian growth is dielectric breakdown, of which lightning and Lichtenberg figures are noteworthy examples. To reproduce these patterns Niemeyer et al. [111] proposed the *dielectric breakdown model* (DBM). The pattern in DBM is assumed to consist of a perfect conductor, initially a single point, surrounded by a perfect insulator. A spherical shell with an electric charge opposite to that of the conductor is assumed to surround the initial seed at some radius; this provides the initial boundary conditions (although other boundary conditions are also possible [79]). The pattern is expanded by converting part of the perfect insulator into a perfect conductor, thus modifying the previous boundary conditions. During each iteration the electric field at each point in space is determined (line 2). A cell adjacent to the existing pattern is chosen randomly, and then added to the pattern. The probability of choosing cell i, denoted p_i, is calculated by raising its field value v_i to the n th power, and normalizing this value,

$$p_i = \frac{v_i^n}{\sum_{j=1}^{m} v_j^n}.$$

Iterative addition of elements recreates patterns similar to that of lightning (fig.2.2a), or even DLA. Different patterns can be obtained by varying n, ranging from circular Eden clusters for low powers through to hierarchical branching patterns or even a single straight line for higher powers.

Interestingly, the converse process (the breakdown of a perfect conductor into an insulator) produces similar visual results, but the resulting boundary conditions can be solved much more efficiently yielding a faster simulation [80].

Fracture patterns are not generally considered to be examples of Laplacian growth, but Federl et al.'s [35] model of this phenomenon is analogous to the Laplacian algorithm presented here. Federl et al. simulated crack formation using the finite element method, with cracks providing the boundary conditions. First the strain is calculated throughout the material (line 2). If the strain exceeds the material strength at any point a crack is introduced (line 3), and strains recalculated (line 4). Cracks propagate as indicated by the strain in the material, with new cracks introduced as specified by this calculation, producing elaborate crack patterns (lines 2-5 and fig.2.2b). The quality of results and efficiency of calculation are increased significantly through an adaptive re-meshing strategy.

2.2.3 Biological models with growing tips

The growth of many branching structures, such as corals, trees, and axons, is restricted to branch tips. In spite of this limitation, a rich variety of forms arise. Models of these structures can often be differentiated by how growth is directed at branch tips, and where branching occurs.

An example of a model with growing tips was proposed by Ulam [168]. He con-

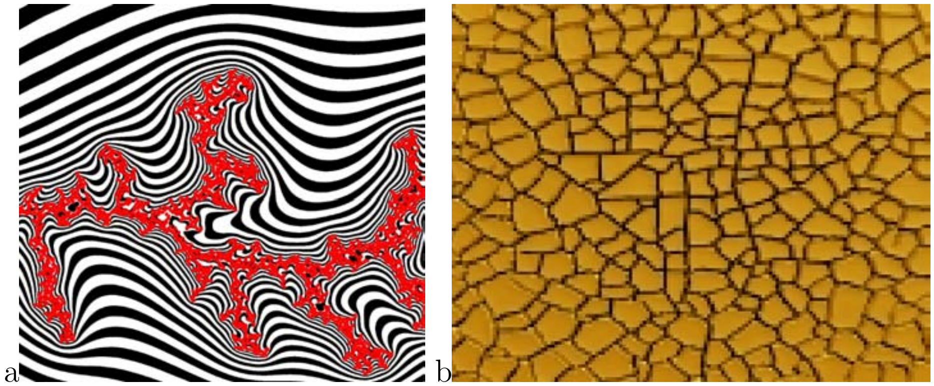


Figure 2.2: Models of Laplacian growth. a) Dielectric Breakdown Model [111] b) Model of crack formation [35]

sidered tree patterns emerging from extremely local competition for space. These patterns were constrained to a grid and initially consisted of a single grid-tile. Adjacent tiles are added synchronously to the pattern during each iteration, provided the resulting branches would not collide. This simple rule emulates competition for space using environmental cues, and yields fractal patterns with a highly structured spatial organization (fig.2.3 e). An equivalent interpretation of Ulam's model is that tips grow in the same direction while unblocked, and branching occurs at predefined angles when there is enough space to support a branch.

The models of coral formation developed by Abraham [1], and Kaandorp [74, 73] simulate the growth of the coral's structure at the developing tips. The model proposed by Kaandorp, uses *radial accretive growth* [74, 73]. In Kaandorp's model the surface of the coral is represented as a connected set of edges in 2D, or faces in 3D. An axis of growth is associated with each growing tip, with growth occurring along the normal of each edge (2D) or face (3D). Growth is varied so that elements facing away from the axis grow less. The presence of light and nutrients affect growth and modify the axis of growth over time. A tip bifurcates when two distinct growth maxima emerge. This creates bifurcating structures reminiscent of many corals, and sponges (fig.2.3a).

In contrast to Kaandorp's model, which represents the surface of the coral, Abraham [1] models the skeleton of the coral structure as a tree graph. Branch tips follow a gradient of nutrients in their environment. Nutrients are assumed to diffuse through the environment and are consumed by branches. As branches grow, the pattern and distribution of nutrients are updated, directing the path of tips away from each other over time. Tips branch when there is enough nutrients to the left and right of the tip to support additional branches. The resulting patterns resemble some corals, and the pattern of vasculature observed in *Ginkgo biloba.* This model is closely related to Laplacian growth, and follows the algorithmic interpretation of Laplacian growth provided in the previous section (Alg 1). In contrast to models of Laplacian growth, the pattern is explicitly represented as a tree graph and is extended only at branch tips.

Radial accretive growth tends to be the exception, instead of the rule, in the class of biological models with growing tips, in that the surface geometry and their interactions with the environment are modeled explicitly. The approach taken by Abraham of modeling the structure of a branching pattern as linear segments (representing the skeleton of the pattern) is much more common. Branching and growth is then handled via some decision process at each element, providing a higher level of abstraction, whereby endogenous and exogenous factors affect development and growth directly, rather than as the outcome of a secondary process.

Cohen proposed a related simulation of biological pattern generation [21]. The model operates on tree graphs and distinguishes between apices and internal nodes. Additionally, the impact of internal and environmental cues are incorporated as a small set of growth and branching rules. Each tip grows away from nearby pattern elements guided by a continuous density field. Additionally, tips grow with some inertia, causing the direction of the tip to change slowly over time. Branching occurs probabilistically and can occur at any internal node during any timestep. The probability that a branch will be produced is increased in less dense regions, and at elements distant from branch tips and branching points. Cohen's patterns suggest a hierarchical structure and spatial organization similar to that seen in trees and the vein patterns of leaves (fig.2.3 b).

The growth of tips in Cohen's model is guided by the density field of pattern elements. The model of axon formation proposed by Gierer [45] also uses a field to direct growth, but does not consider the feedback of the pattern on the field. His model was based on several observations of axon growth. Specifically, he used the observations that axons grow towards a target, meandering and bifurcating along this path, with many branches near the target.

In Gierer's model the axon follows a global gradient with a minimum at the target. Meandering results from random local fluctuations in the global gradient. The axon branches when moving forward no longer minimizes the gradient. This leads to emergent branching at the target, as any forward movement will be up the gradient. Gierer's model convincingly recreates the pattern of a single axon based on a small number of geometric observations (fig.2.3c).

The models of Abraham, Cohen and Gierer all rely on a field to guide development. As this field is defined over the space containing the pattern, it is difficult to grow the interior of this space. In contrast, Gottlieb formulated a model of vascular development in which the outcome of chemical interactions was interpreted geometrically [49], which eliminates the reliance on a field to guide development and allows growth to be considered. This model starts with a set of lines representing vasculature embedded in a coarse grid. Growth is simulated by refining the grid during each iteration of the model, and extending veins to the centres of new grid cells within a predefined radius, which decreases over time. Cell centres in this context represent new sources of a vein-inducing hormone. Refining the grid for a few iterations produces complex patterns reminiscent of vascular patterns in plants and animals (fig.2.3d).

Gottlieb's model demonstrates the interplay between growth and the range of exogenous communication. Over time, the distance over which elements can communicate decreases. This process is conceptually equivalent to growing the pattern uniformly while keeping this distance constant. The geometric reasoning applied in Gottlieb's model is very similar to that employed in SCA, making this work of particular relevance.

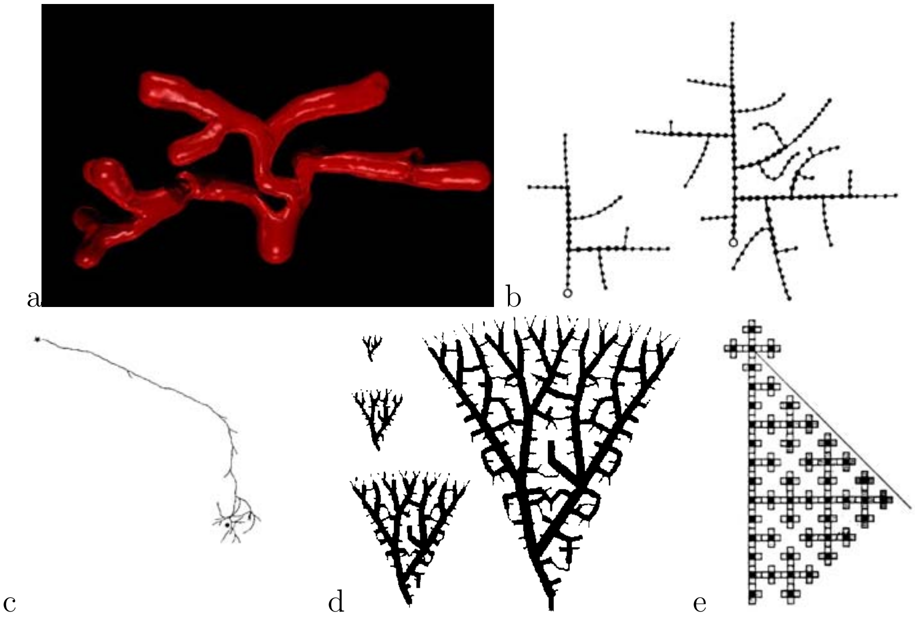


Figure 2.3: Biological models with growing tips a) Radial accretive growth [73] b) Cohen's simulation of biological pattern generation. The structure on the left is shown at a later stage of development on the right [21]. c) Grierer's model of axon formation [45] d) Gottlieb's model of vascular patterning [48]. Four iterations of the model are shown. e) An example of Ulam's growing figure [168].

Chapter 3

Modeling trees with the Space Colonization Algorithm[1]

Visual modeling of tree architecture began with the work of Honda [60]. He proposed a model of trees as recursive branching structures characterized by a small number of geometric attributes: branching angles and length ratios of consecutive branch segments (internodes). The basic tenet of Honda's approach — treating a tree as a recursive branching structure — underlies most generative tree models proposed to date. Early examples include a direct adaptation of Honda's model to computer graphics [4] and tree models proposed by Bloomenthal [11] and Oppenheimer [116]. Reeves and Blau [131], Weber and Penn [175], Lintermann and Deussen [88], and Prusinkiewicz et al. [129] improved the visual realism of recursive tree models by introducing random and organized variation of parameter values as a function of position of the affected branches within the tree structure.

Biologically, the view of a tree as a recursive structure is justified by the process of tree development [176, 25]. For temperate-climate trees, development typically begins with a single stem that carries leaves and lateral buds, arranged in a regular phyllotactic pattern. These buds may in turn give rise to new branches.

Without doubt, the regular pattern of bud distribution and the repetitive character of their potential development are important determinants of the architecture

---

[1] This chapter is based on [143]

of young trees. These factors, however, play a reduced role when considering more mature trees and shrubs [147]. The first reason is the diversity of the fates of buds: only some buds produce branches, whereas others produce flowers, remain dormant or abort. These different fates break the regularity implied by the initial bud arrangement. Second, branches differ significantly in their growth and development: some twigs become major limbs, whereas others remain small or are shed. Third, the initial directions of branch growth, determined by bud arrangement and branching angles, are significantly modified by branch reorientation, tropisms, and mechanical bending.

These phenomena have been considered in plant models constructed for biological and computer graphics purposes alike. A statistical description of the fates of buds is the cornerstone of the models of de Reffye et al. [25]. A combination of a statistical and hormone-driven control of the fate of buds was incorporated into topiary tree models by Prusinkiewicz et al. [127]. Competition for light was used to control bud fate and branch shedding by Teak [162] and reproduced by Měch and Prusinkiewicz [109]; a related approach was proposed by Chiba et al. [17]. Further work considered the impact of light quality [44] and the effects of gravity and tropisms [70].

Although these models incorporated a variety of processes, they preserved the fundamental role of recursive branching; the other factors were just modifiers. In the structure of a mature tree, however, the regularity of the recursive branching is largely lost, overridden by subsequent development. This phenomenon was highlighted by Sachs and Novoplansky [147, page 206]:

> Only a small proportion of available apices actually develops. The excess developmental capacity is evident in the presence of many undeveloped buds and small branches, many of which could develop to entire trees if other apices were removed. There is often no strict determination of the precise location and number of shoot apices that develop into large branches.

As indicated above, the fates of buds do not seem to follow a strictly predetermined genetic program. To account for this, Sachs and Novoplansky hypothesized that the environment strongly influences the fates of buds, reflecting the developmental plasticity of the tree. They suggested that characterizing the mature form of a tree as a recursive structure provides an incomplete picture, arguing that environmental factors play a major role in determining the detailed structure of tree canopies. They proposed that the developmental plasticity of a tree could be explained by competition between branches to grow. To provide a plausible mechanistic basis for this argument, they proposed that this competition is regulated by apical dominance and enhanced by the response of branches to local shading. The spatial arrangement of branches thus significantly influences their subsequent development and, consequently, the architecture of a tree.

Using the point of view of Sachs and Novoplansky as the point of departure, we explore below the Space Colonization Algorithm as an alternative to the recursive branching process as the basis for modeling trees. Our model considers the competition of branches for space, mediated by environmental cues, as the dominant factor determining the form of trees and shrubs.

Competition for space has been used previously to generate bifurcating structures, including generic branching patterns [21], vascular patterns [48], and corals [74, 1]. Even the concept of explaining tree architecture in terms of competition for space is not new; in fact, it precedes the recursive model of Honda [60]. As early as 1962, Ulam [168] exploited competition for space to formulate a 2D cellular-automaton model of abstract tree-like branching structures. Stevens [159] augmented the topologies created by Ulam's model with biologically-motivated geometric attributes (branch widths and branching angles), obtaining visually realistic models of young trees. More recently, Rodkaew et al. [136] proposed an algorithm that distributes particles within the shape of a tree crown, then traces their motion down to the root. The converging paths of the particles, which are attracted both to their neighbours and toward the tree base, form a tree. Rodkaew's algorithm and its recent extension [110] produce surprisingly realistic-looking tree models in spite of the disregard for the processes of tree development (the algorithm generates branches from the tips inward).

These results provide the background for the procedural tree generation method presented here. Our key idea is an iterative addition of new elements (nodes) to the tree structure formed in previous steps. This process is guided by the proximity of points marking the availability of free space. Historically, the proposed method was developed as a 3D extension of the algorithm for generating leaf veins described in Chapter 4. (Interestingly, Rodkaew et al. [136] also derived their tree-generation algorithm from a vein-generation method.) Although developed later, in this thesis the algorithm for trees is presented first, as this application is conceptually simpler (unlike leaf blades, where vein patterns are formed, the space in which trees grow does not expand).

3.1 The method

3.1.1 Procedure overview

The steps of the procedure are shown in Figure 3.1. A three-dimensional envelope of the tree crown is provided as an input (a). It can be specified using any method that makes it easy to test whether a point lies inside or outside the enclosed volume. In this thesis a generalized surface of revolution is used, which is obtained by rotating a planar generating curve (possibly with a changing shape) around a vertical tree axis (Section 3.2.1). At the beginning of tree generation, the space within the envelope is seeded with a set of *attraction points* (b), which signal the availability of empty space for growth and are removed when reached by a branch. The distribution of the attraction points is a user-controlled attribute of the method; some possibilities are outlined in Section 3.2.2. Given the attraction points, the tree skeleton is formed iteratively, beginning with a single node at the base of the tree (b). During each iteration, new nodes, delimiting short branch segments, extend the skeleton toward nearby attraction points (c, d). This process terminates when all attraction points have been removed, when no nodes are within the radius of influence of the remaining attraction points, or after a user-specified number of iterations.

The resulting tree skeleton may now be manipulated further. First, the skeleton nodes may be decimated to reduce the amount of data representing the tree geometry (e). Moving each remaining node in parallel half way toward its more basal neighbour reduces the branching angles (compare the insets in Figures e and f) and can

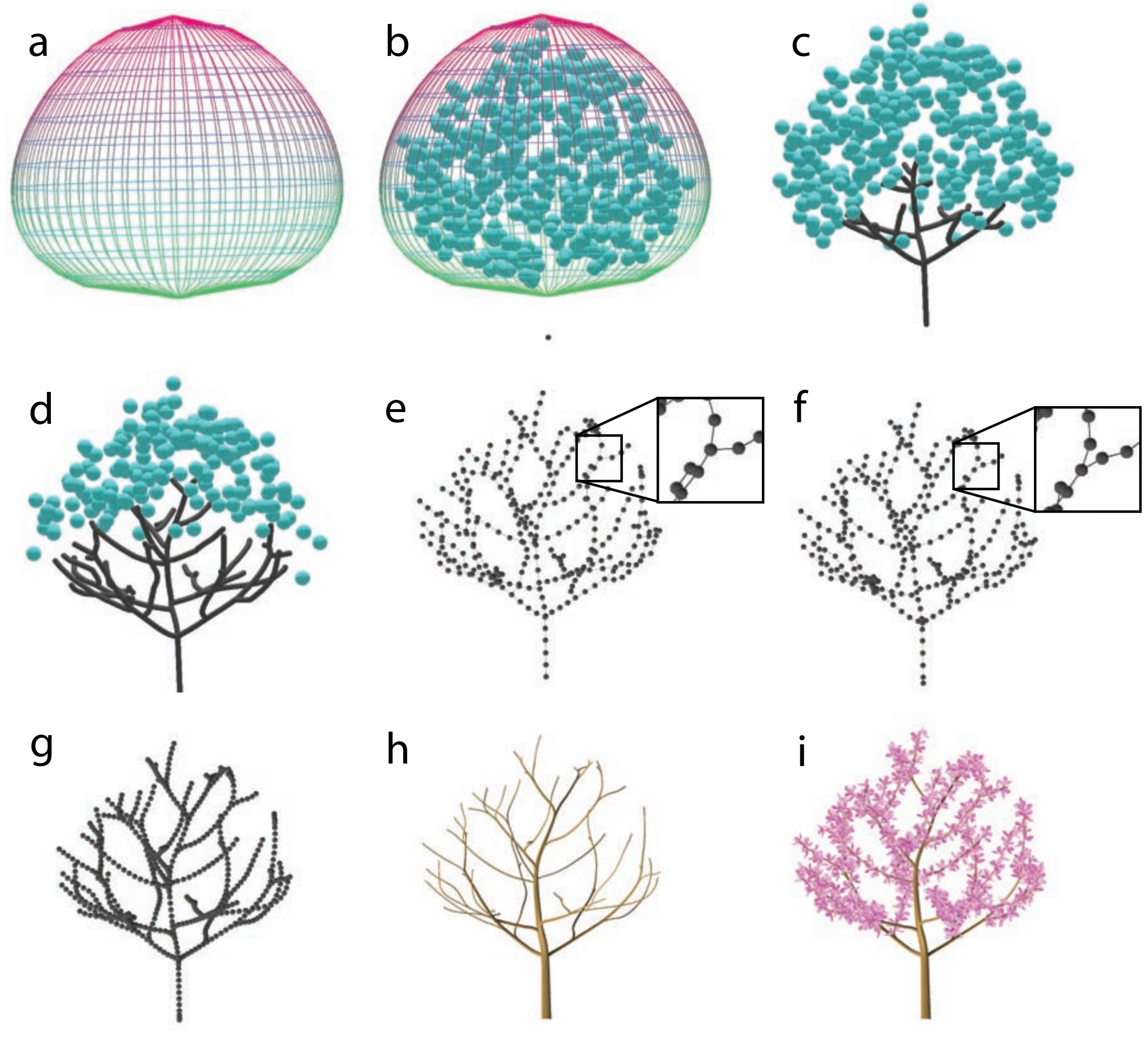


Figure 3.1: Key steps of the proposed method for generating trees. a) Specification of the input envelope; b) Placement of the initial tree node and attraction points in the envelope; c, d) Generation of the tree skeleton; e) Node decimation; f) Node relocation; insets show the modified branching angle; g) Subdivision; h) Construction of generalized cylinders; i) Addition of organs.

significantly impact the overall appearance of the tree. Curve subdivision, extended to branching structures [130], may be applied to the original or decimated skeleton to create more smoothly curved limbs (g). Once these steps are completed, the tree geometry is modeled using generalized cylinders [11] centred on the axes of the skeleton (h), with the diameter of each limb calculated using the pipe model [152](Section 3.2.3). If needed, organs, such as leaves, flowers and small branches, are added to the tree (i).

3.1.2 The space colonization algorithm.

The cornerstone of the proposed method is the space colonization algorithm (Figure 3.1b and c), which treats competition for space as the key factor determining the branching structure of trees. The structure of the tree is represented as a tree graph $G = \langle V, E \rangle$. The nodes $v \in V$ of this graph represent small internode segments, which we refer to as *tree nodes*. Adjacent nodes are connected by edges $e \in E \subset V \times V$. The edges are oriented from the base of the tree to its extremities. Connections between tree nodes play a key role when determining branch width using the pipe model (Section 3.2.3).

The operation of the algorithm begins with an initial configuration of N attraction points (usually hundreds or thousands) and one or several tree nodes. The tree is generated iteratively. During each iteration, an attraction point may influence the nearest tree node. This influence occurs if the distance between the point and the closest node is less than the *radius of influence* d_I. There may be several attraction points that influence a single tree node v: this set of points is denoted by $S(v)$. If $S(v)$ is not empty, a new tree node v' will be created and attached to v by the segment (v, v'). The node v' is positioned at distance D from v, in the direction defined as the average of the normalized vectors toward all the sources $s \in S(v)$. Thus, $v' = v + D\hat{n}$, where

$$\hat{n} = \frac{\vec{n}}{\|\vec{n}\|} \text{ and } \vec{n} = \sum_{s \in S(v)} \frac{s - v}{\|s - v\|}. \tag{3.1}$$

The distance D serves as the basic unit of model and provides control over the resolution of the resulting structure.

Once the new nodes have been added, a check is performed to test which, if any, of the attraction points should be removed due to the proximity of tree branches that have grown toward these points. Specifically, attraction point s is removed when at least one tree node v is closer to s than a threshold *kill distance* d_k.

Discussion of $\vec{n}$

The definition of $\vec{n}$ may seem somewhat arbitrary, but it has some biological justification. Trees transport water and nutrients between the roots and leaves along their branches. Reducing the total length of branches decreases the cost of building and maintaining this transport system. As such, it is plausible that the architecture of trees attempts to minimize the total length of branches. This has been postulated previously by Leopold, who found that the most probable arrangement of branches appears to minimize this value, while respecting other constraints (such as light capture) [85].

When tree node v is selected by $S(v)$, the branch segment stemming from it extends towards the empty space indicated by the points in $S(v)$. As the branch is extended it should do so in as economical a fashion as possible (i.e. it should colonize the space using the smallest number of tree nodes). An infinitesimal extension of v in direction $\hat{n}$ minimizes the sum of the Euclidean distances between v and the points in $S(v)$ (This is stated formally in the following theorem, and the accompanying proof is provided in Appendix A). This property is well approximated by a finite segment provided that it is short in relation to the distance between attraction points. Given this result, extending v in the direction of $\hat{n}$ thus helps minimize the lengths of current and subsequent branches, in accordance with Leopold's postulate.

Theorem 1. *Let* $\mathbf{v} \in \Re^n$ *and* $S(\mathbf{v})$ *be as defined above, let* $E : \Re^n \to \Re$ *be the scalar field defined as the sum of the Euclidean distances between* $\mathbf{v}$ *and each point in* $S(\mathbf{v})$*:*

$$E = \sum_{s \in S(\mathbf{v})} \|\mathbf{s} - \mathbf{v}\| \tag{3.2}$$

Then

$$\vec{n} = \sum_{\mathbf{s} \in S(v)} \frac{\mathbf{s} - \mathbf{v}}{\|\mathbf{s} - \mathbf{v}\|}$$

is equal to $-\nabla E$*, the direction in which* E *decreases fastest.*

Although the algorithm operates in 3d when trees are modeled, this property holds in any finite-dimensional Euclidean space.

The relation between E and $\hat{n}$ is illustrated in Figures 3.2 and 3.3 using two and three attraction points, respectively. The scalar field E, and the direction vector $\hat{n}$ are visualized in (a) and (b). In (c) of both figures the initial tree node extends along the direction field shown in (b) until a bifurcation occurs (bifurcation is an emergent property of the algorithm and is demonstrated in the example provided in Section 3.1.3). In Figure 3.2(c) the resulting branches are selected by a single attraction point, and subsequently extend directly towards the two attraction points. In Figure 3.3(c), following the first bifurcation the left branch is only selected by the

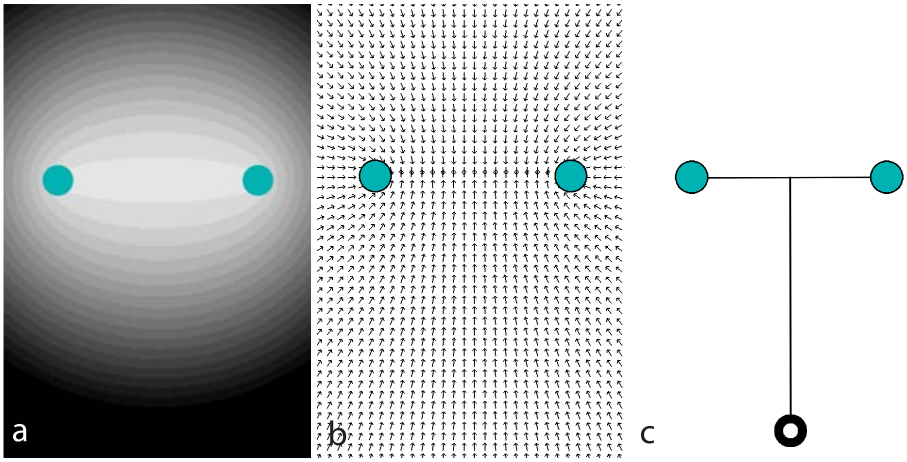


Figure 3.2: The relation between E and $\hat{n}$ visualized for two attraction points. a) The scalar field E is visualized for two attraction points (blue disks). b) The direction of $\hat{n}$ is visualized. c) A tree node (black disk with white centre) is added, and extended as described in the the preceding section.

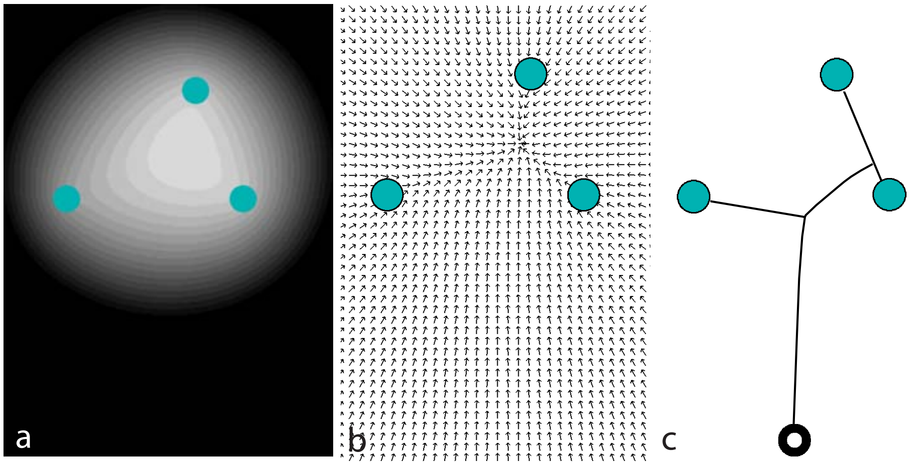


Figure 3.3: The relation between E and $\hat{n}$ visualized for three attraction points. a) The scalar field E is visualized for three attraction points (blue disks) b) The direction of $\hat{n}$ is visualized. c) A tree node (black disk with white centre) is added. The lines show the branches produced when the tree node is extended as described in the preceding section.

left-most attraction point, causing the branch to extend directly towards the selecting point. The right branch is selected by two attraction points, and the scenario described above for two attraction points applies.

Application of a directional bias to growth

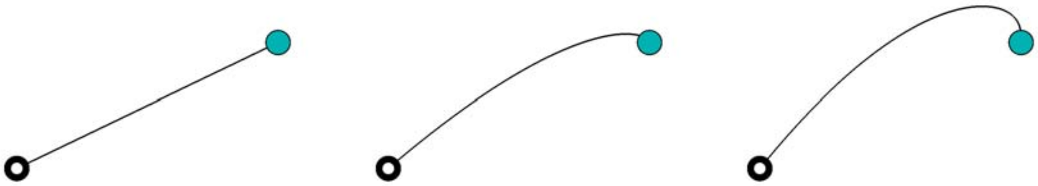
Optionally, the direction of growth can be biased by a vector $\vec{g}$ representing the combined effect of branch weight and tropisms using the equation

$$\tilde{n} = \frac{\hat{n} + \vec{g}}{\|\hat{n} + \vec{g}\|}. \tag{3.3}$$

The impact of varying $\vec{g}$ is shown for a single branch in Figure 3.4. Introducing a small bias causes the branch to curve slightly as it approaches the attraction point ((b) and (d)). Larger biases cause the branch to curve away from the attraction point before following an almost vertical path to the attraction point ((c) and (e)).

3.1.3 Example

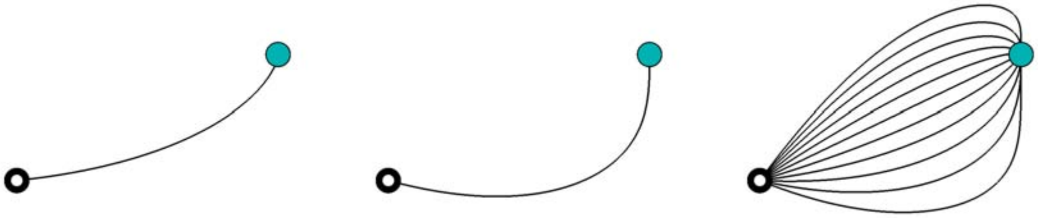
The SCA is illustrated in Figure 3.5. We begin following its operation at the stage when the tree structure consists of six nodes (black disks with white centres) and there are four attraction points (blue disks) (a). First, each attraction point is associated with the closest tree node, provided that it is within the radius of influence (b, blue lines); this establishes the set of attraction points that influence each node. The normalized vectors from each tree node to each source that influences the node are then found (c, black arrows). These vectors are added and their sum is normalized again according to Eq. 3.1 (d, red arrows), providing the basis for locating new tree nodes (d, red circles). No bias is used in this simple example. The new nodes are incorporated into the tree structure, in this case extending the main axis and initiating



(a) $\vec{g} = (0, 0, 0)$, no bias is applied.

(b) $\vec{g} = (0, 0.35, 0)$, slight upward bias

(c) $\vec{g} = (0, 0.7, 0)$, strong upward bias



(d) $\vec{g} = (0, -0.35, 0)$, slight downward bias

(e) $\vec{g} = (0, -0.7, 0)$, strong downward bias

(f) Effect of varying $\vec{g}$ from $(0, 0.8, 0)$ (top curve) to $(0, -0.8, 0)$ (bottom curve).

Figure 3.4: Sub-figures illustrate the impact of varying $\vec{g}$. In each image the simulation is initialized with a single tree node (the black disk with a white centre) and attraction point (blue disk). Adding additional tree nodes using Eq.3.3 generates the curves connecting the initial tree node to the attraction point.

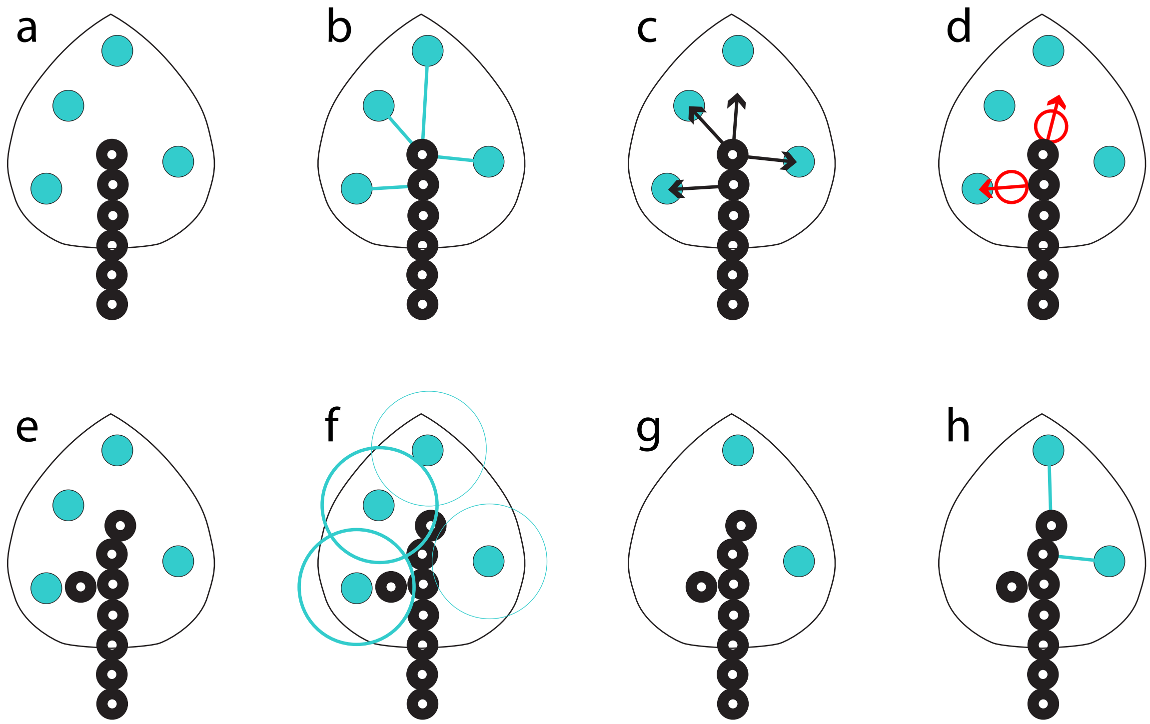


Figure 3.5: The space colonization algorithm. A detailed description is provided in the text.

a lateral branch (e). The neighbourhoods of the attraction points (blue circles) are now tested for the inclusion of (the centres of) tree nodes (f). The neighbourhoods of the two leftmost sources have been penetrated by the new branches, as indicated by the bolder representation of the corresponding circles. The affected attraction points are thus removed (g). The tree nodes closest to these points are now identified (h), beginning the next iteration of the algorithm.

Note that the top and the right attraction points jointly influence the top tree node in Figure 3.5b, but the same two attraction points influence different tree nodes in Figure 3.5h. Such splits in the set of attraction points, which at some stage influence the same tree node, but later affect different points, are an essential feature of the algorithm, as they lead to the emergence of branches.

3.2 Design considerations

The algorithm involves repetitively testing the set of attraction points for proximity to the tree nodes. Specifically, the closest tree node to each attraction point must be determined. The straightforward algorithm for determining the closest node to an attraction point checks the distance from each tree node to the attraction point. If k attraction points and n tree nodes exist during an iteration, then performing this calculation for all attraction points requires $O(kn)$ calculations.

We perform these calculations by constructing a three-dimensional Voronoi diagram of the set of nodes and testing the resulting domains for the inclusion of attraction points, which improves run times considerably. To this end, we employed the 3D Delaunay triangulation routines included in the Computational Geometry Algorithms Library [118]. The complexity and run-time of the algorithm are discussed in detail in Section 3.2.4.

The space colonization algorithm (steps a–c in Figure 3.1) has been implemented using the `vv` relational modeling system [155], which provides convenient operations for manipulating structures with dynamically changing topology. The generated skeletons are further processed (steps d–h in Figure 3.1) using L-systems [77]. In fact, the entire procedure for generating trees can be implemented using L-systems. The primary reason the SCA was implemented in `vv` is historical, as the implementation for trees was developed as an extension of the model used to generate veins in leaves, which was implemented in `vv`. Additional details important to implementation of the proposed method are provided below.

3.2.1 Crown Specification

The crown of a modeled tree, where attraction points are placed, is specified by enclosing a volume within a surface. The surface used to specify this volume must meet several requirements:

1. have an efficient, and easy to implement, inside/outside test;
2. be easy to specify; and
3. allow most tree crowns to be modeled (using one or several surfaces).

Optionally, the surface should provide a measure of distance of a point to the boundary. In this work *generalized surfaces of revolution* were used, although in principal many surfaces, such as implicit surfaces, are suitable for this purpose.

A generalized surface of revolution is obtained by rotating a planar *generating curve* around a vertical axis. The generating curve may change size and shape during rotation, as for envelopes that are not axially symmetric. This can be contrasted with surfaces of revolution where the generating curve remains constant.

In our implementation, this variation was captured by interpolating between a set of curves placed at specific rotation angles around the z-axis (fig.3.6). The resulting surface, parameterized by two variables, $Q(\theta, z) : [0, 2\pi] \times [a, b] \rightarrow \Re^3$ is defined by a set of 1D curves with associated angles:

$$C = \{(C_1, \theta_1), (C_2, \theta_2), \cdots, (C_n, \theta_n)\},$$

where $C_i : [a, b] \rightarrow \Re$, and $\theta_i < \theta_{i+1}$. C_i defines the profile of the surface at angle θ_i. We compute the function $Q(\theta, z)$ from C as

$$Q(\theta, z) = \mathit{CatmullRom}(z, \theta)(\cos(\theta), \sin(\theta), z),$$

where *CatmullRom* interpolates between the z values of the curves in C as θ is varied using Catmull-Rom splines [15]. Catmull-Rom splines are piecewise cubic polynomials that interpolate a set of control points, here the z value of generating curves, while maintaining first-order continuity. Tangents at each control point are calculated using neighbouring control points, which makes them less flexible than Hermite splines, but allows for easier user specification.

To define the function $\mathit{CatmullRom}(z, \theta) : [a, b] \times [0, 2\pi] \rightarrow \Re$ let us consider a given θ, and let $i \in \{1, 2, \cdots, n\}$ such that $\theta_i \leq \theta < \theta_{i+1}$. If $i + 1$ is greater than n, then $\theta_1 + 2\pi$ is used in place of θ_{i+1}. Now let:

$$t = \frac{\theta - \theta_i}{\theta_{i+1} - \theta_i}$$

This makes $t = 0$ when $\theta = \theta_i$, and $t = 1$ when $\theta = \theta_{i+1}$. Using t the function

$CatmullRom$ is:

$$\begin{array}{c} CatmulRom(z,\theta) = C_i(z)h_1(t) + C_{i+1}(z)h_2(t) \\ +\frac{C_{i+1}(z)-C_{i-1}(z)}{2}h_3(t) + \frac{C_{i+2}(z)-C_i(z)}{2}h_4(t) \end{array}, \tag{3.4}$$

where h_1, h_2, h_3 and h_4 are the cubic polynomials

$$h_1(t) = 2t^3 - 3t^2 + 1,$$

$$h_2(t) = -2t^3 + 3t^2,$$

$$h_3(t) = t^3 - 2t^2 + t,$$

$$h_4(t) = t^3 - t^2.$$

According to equation 3.4, polynomial coefficients h_1 and h_2 interpolate between the values at $C_i(z)$ and $C_{i+1}(z)$ as t increases from 0 to 1, whereas the remaining coefficients, h_3 and h_4, are applied to the tangents calculated from neighbouring points, allowing first-order continuity to be maintained as θ is varied. Figure 3.7 shows a slice through the crown in Figure 3.6 and illustrates the interpolation performed using Catmull-Rom splines.

The surface Q, as defined above, produces points between $z = a$ and $z = b$. The axis of Q can then be modified by rotating, and translating the output of Q.

More complex crowns can be created by generating several generalized surfaces of revolution and combining the surfaces using simple constructive solid geometry operations (such as the union, intersection, and difference operators). These operations may be applied repeatedly to obtain increasingly complex crowns based on a small number of simpler crowns.

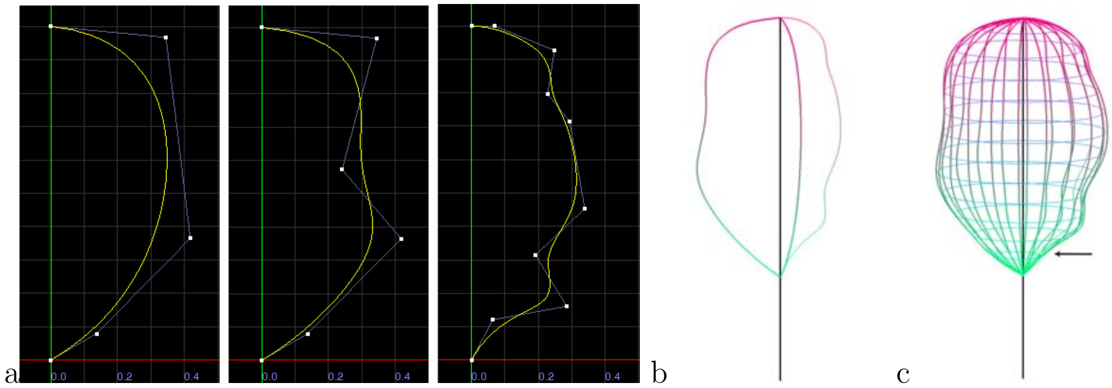


Figure 3.6: A generalized surface of revolution and associated generating curves. a) The three curves defining the generalized surface of revolution. b) The curves placed at regular angles around the z-axis. c) The surface generated by sweeping the curves around the axis as described in the text. The arrow indicates the location of the slice illustrated in the following figure.

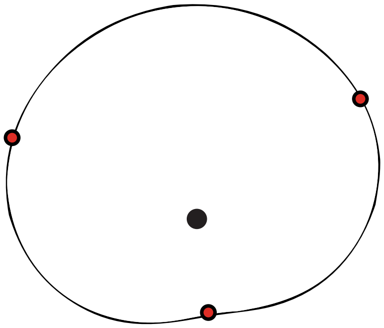


Figure 3.7: A slice of the generalized surface of revolution shown in the previous figure. This slice was created by holding z constant and rotating through all θ. Red disks mark points on the three generating curves and the black disk marks the axis of rotation.

3.2.2 Attraction point generation

To place attraction points in Q a function indicating whether a point lies inside or outside the volume enclosed by the surface is needed. This inside/ouside function can be phrased concisely using the $CatmullRom$ function discussed above.

To this end we define the characteristic function for Q, denoted by $\chi_Q(P) : \Re^3 \rightarrow \{1, 0\}$. This function takes a value of 1 at any point in the region bounded by Q and 0 at all other points. Specifically, given point $P \in \Re^3$ with associated cylindrical coordinates (r, θ, z),

$$\chi_Q(P) = \begin{cases} 1 & CatmullRom(z, \theta) \geq r \\ 0 & otherwise. \end{cases}$$

Attraction points are generated iteratively by generating point P randomly with a 3D uniform distribution[1], and verifying that it lies in the crown of the tree using $\chi_Q(P)$.

Optionally, a minimum distance between attraction points, denoted b_d, can be enforced. In this case, when point P is generated it is placed in the tree crown only if it is further than b_d from the attraction points that have already been placed, producing an *overdispersed point distribution* [23, 104].

The specifics of attraction-point generation are a parameter of the simulation, and are discussed further in Section 3.3.

---

[1] The algorithm uses a random distribution of points as opposed to a regular grid as a way of introducing variation into the model. The advantage of this approach is its simplicity compared to biologically more sound methods [161]

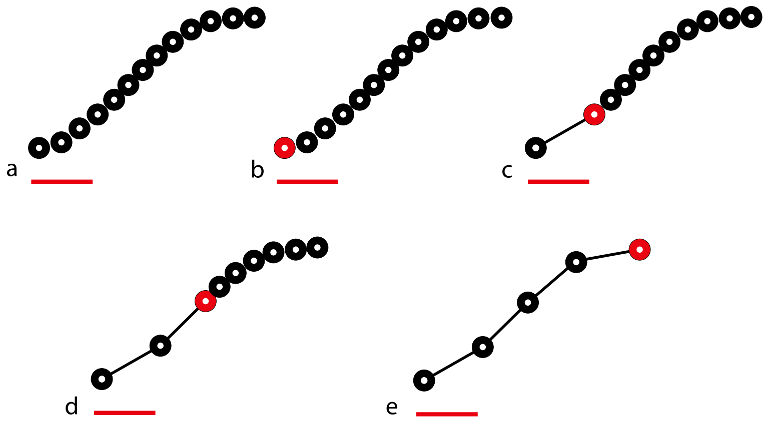


Figure 3.8: Decimation of a single branch, τ_d is indicated by the red line in each sub-figure. The tree-node being considered for decimation is shown as a red disk.

3.2.3 Modeling the branching structure of the tree

Post processing

The tree-graph generated by the SCA provides the skeleton of the tree. This skeleton can be further manipulated using the post-processing steps outlined in (e-g) of Figure 3.1. Stages (f), branch point modification, and (g), subdivision, have been described already, but the details of stage (e), decimation, require further explanation.

Decimation is performed on each branch independently, as illustrated in Figure 3.8. A branch is decimated from the base (b) to the tip (e). The decimation procedure uses a threshold, τ_d, denoting the desired minimum spacing between adjacent tree-nodes in a branch. Nodes are processed sequentially, and the distance between nodes is accumulated in d_A. At each node the accumulated distance is compared to τ_d. If $d_A < \tau_d$ then the node is removed from the branch, otherwise d_A is reset to 0 and the node is kept (b-e). Decimation decreases the number of points in a branch while maintaining a uniform density of points along the branch (compare (a) and (e)).

Determination of the diameter of branches

Once the final skeleton of a tree has been generated, the diameters of the limbs are determined according to the pipe model [152], which relates the cross-section of a limb below a branching point to the combined cross-sections of the limbs above. The pipe model assumes that each leaf is connected by a pipe to the base of the tree, supplying the leaf with water, and that the cross-sectional area of this pipe is constant. At any point in the tree there must be enough pipes to supply the leaves above it; so the cross-section at this point is determined by the number of leaves it must support.

In accordance with the preceding discussion: calculation begins with the assumption that all branch tips have the same initial radius r_0, and proceeds basipetally, from the branch tips toward the tree base. If branches of radii r_1 and r_2 meet at a branching point, the radius r of the supporting branch is found using the formula

$$r^n = r_1^n + r_2^n, \tag{3.5}$$

where n is a parameter of the method (usually between 2 and 3 [92, pages 131–135]). The relation given by Eq. 3.5 was proposed by Murray [108] based on measurements obtained from several tree species.

The geometry of branches

Generalized cylinders [11] are used to model the branching structure of the tree. The geometry at each branch is created by sweeping a 2d generating curve along the skeleton of the tree, while varying the scaling factor applied based on the diameter calculated by the pipe model. During this process a parallel transport frame [55, 129] is calculated to orient the generating curves (cross-sections of the generalized cylinder) in a manner that minimizes the twist between consecutive cross-sections. If needed, organs, such as leaves, flowers and small branches, are added to the tree; their spatial distribution is determined with respect to the parallel transport frame and/or absolute directions in the world space in which the tree has been placed.

3.2.4 Algorithm run-time

Typically, generation of an entire tree using the SCA takes a few seconds or minutes. A significant portion of the algorithm's execution time is spent testing the proximity of attraction points to tree nodes and updating the Voronoi diagram containing the set of tree nodes. Thus, the runtime of each iteration of the algorithm depends primarily on the number of tree nodes and attraction points considered. Varying the parameters controlling these quantities changes the resulting pattern as well, and causes a trade-off between achieving particular visual effects and optimizing the runtime of the algorithm. The visual impact of parameters is discussed in greater detail in the results section (Section 3.3).

The run-time of our implementation was tested experimentally. In each experiment a unit sphere was filled with attracting points and a tree node was placed at the origin. The simulation ran until no attraction points remained. Each experiment was run 10 times, with default parameters of $N = 3000$ (number of attraction points), $d_k = 2$ (kill distance), and $D = 0.01$ (the basic unit of the model). These experiments serve as a basis for discussing the impact of pertinent parameters on the run-time of our implementation.

Several parameters affect, directly or indirectly, the number of attraction points and tree nodes tested during a simulation. As the number of attraction points is a

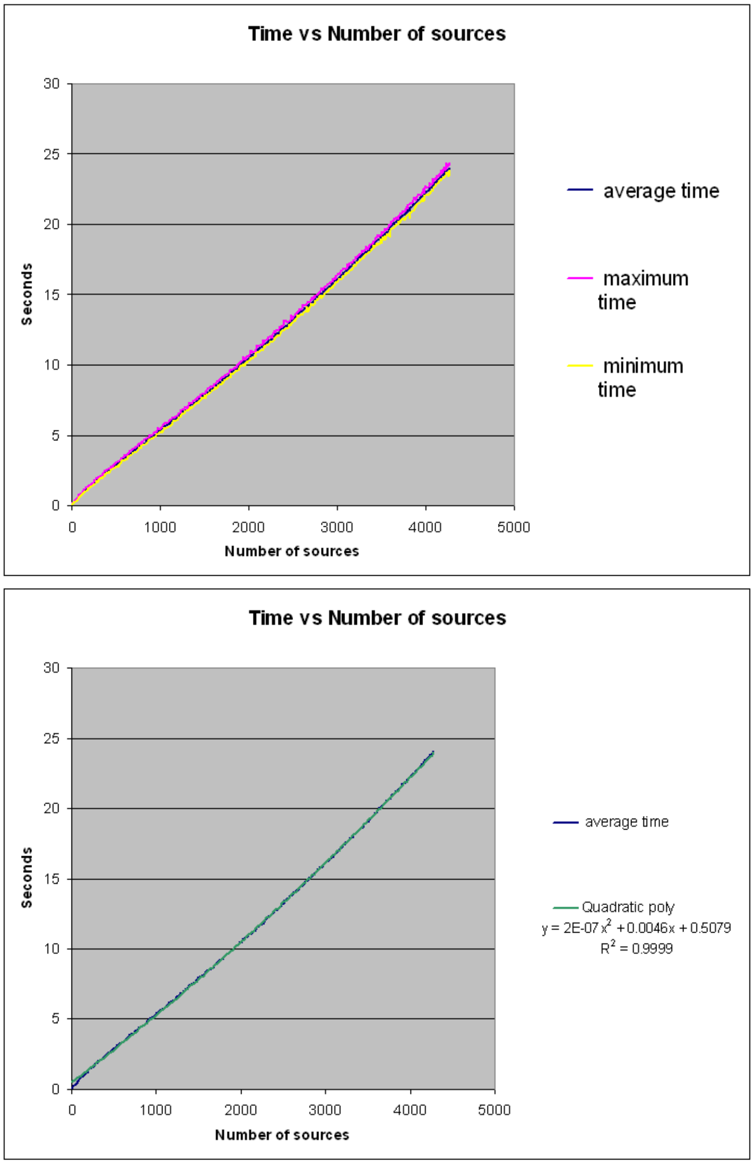


Figure 3.9: Process run time plotted as a function of the number of attraction points (N). (Top) The average, maximum, and minimum run-times are plotted as N is varied. (Bottom) The average run-time is plotted in the same chart as the best fit quadratic polynomial described below the legend.

parameter itself it can be varied directly. Reducing the number of attraction points decreases the runtime, because fewer tree nodes are produced and fewer attraction points must be tested during each iteration (Figure 3.9). The second graph in Figure 3.9 shows the best fit quadratic curve. The coefficient of the quadratic term is several orders of magnitude smaller then the coefficient on the linear term. This tends to indicate that the relationship between N and run-time is approximately linear when $N \in \{1, \cdots, 4000\}$.

Unlike the number of attraction points, the number of tree-nodes is not specified directly. Instead the number of tree nodes present during an iteration is determined indirectly by several parameters. First, the basic unit D used in the model is of particular interest. As D increases, fewer tree nodes are needed to reach the same set of sources (as this decreases the resolution of the model), thereby reducing the number of iterations needed to reach each attraction point (as well as the number of tree-nodes that exist during each iteration).

Instead of using D directly, the relation of runtime of to $1/D$ is examined (the number of nodes needed to span one unit, Figure 3.10), because as D goes to zero the runtime of the algorithm tends towards infinity, making analysis difficult. As with N, the coefficient on the quadratic term of the polynomial is very small compared to the linear term, indicating that the runtime may be a sub-quadratic function of $1/D$ when $0 < 1/D \leq 150$ (the values of D used in this chapter all fall in this range). D significantly impacts the algorithm's run-time. Presently, D is a user-specified constant and it is an interesting question whether an adaptive step could be used instead.

The other parameter to note is d_k, the kill distance used to remove sources.

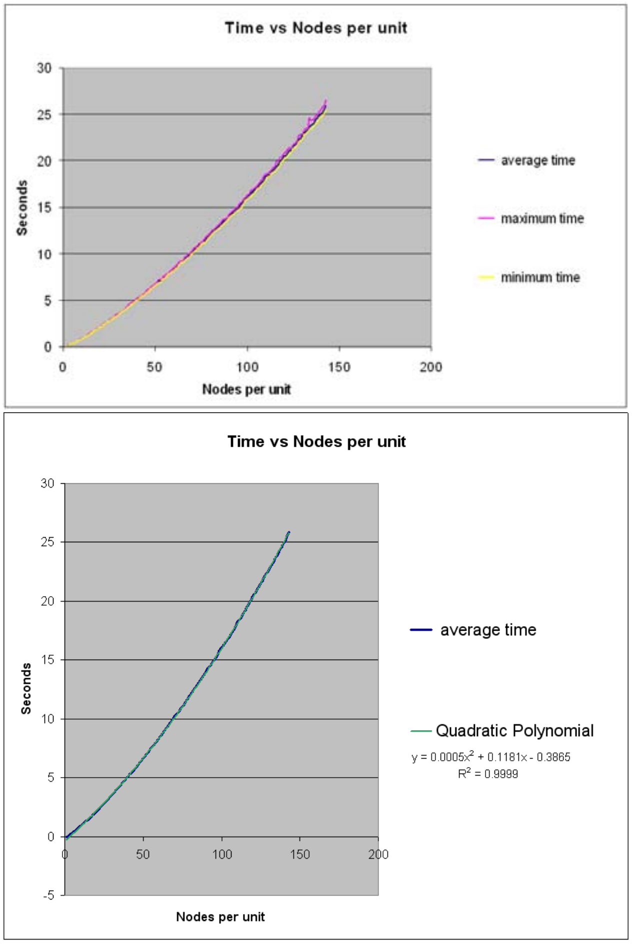


Figure 3.10: Process run time plotted as a function of the number of nodes per unit ($1/D$). (Top) The average, maximum, and minimum run-times are plotted as $1/D$ is varied. (Bottom) The average run-time is plotted in the same chart as the best fit quadratic polynomial described below the legend.

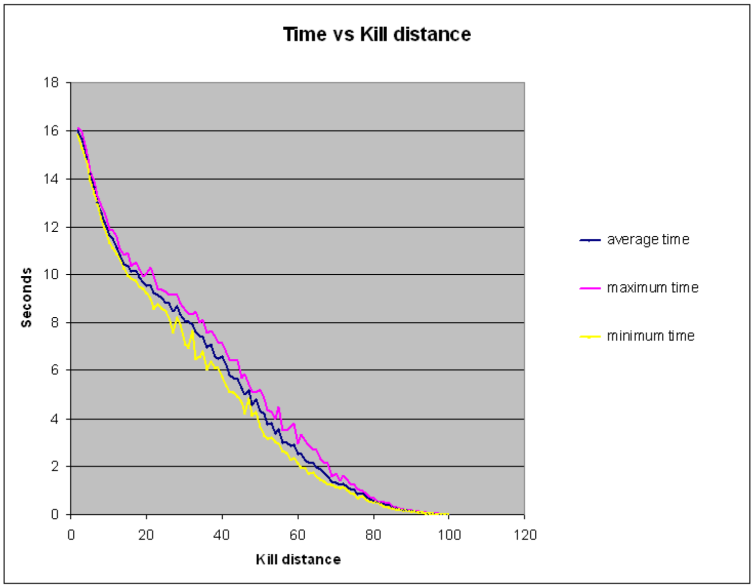


Figure 3.11: Process run time plotted as a function of the kill distance (d_k). The average, maximum, and minimum run-times for each value of d_k are plotted.

Increasing d_k causes attraction points to be removed earlier, thus decreasing the computational cost of subsequent iterations. Additionally, as branches need not penetrate the space surrounding attraction points as closely, fewer tree nodes are needed to reach the same number of attraction points. Figure 3.11 reveals that in this experiment increasing d_k decreases the run-time of our implementation, but in a diminishing manner.

The run-time of a single iteration can be discussed more rigorously. Let k be the current number of attraction points, n the number of tree nodes present at the beginning of the iteration, and r the number of tree nodes created during the iteration. Asymptotically, runtime depends on the calculation of the nearest tree node to each attraction point and the insertion of new tree nodes in the Voronoi diagram containing all tree nodes. This diagram can be constructed “from scratch” during each simulation step in $O(n \log n)$ time [114]; however, the run-time is improved by updating the diagram incrementally from one simulation step to the next [114]. The calculation of the nearest neighbour in the Voronoi diagram takes logarithmic time [121], thus the search for the tree nodes that are nearest to k attraction points require $O(k \log n)$ time, and the insertion of r new tree nodes then takes $O(r \log n)$ time, resulting in a combined run-time of $O((k + r) \log n)$. As each new tree node must be selected by an attraction point and each attraction point can select only one tree node, it follows that k must be an upper bound for r. Therefore, a single iteration of the SCA takes $O(k \log n)$ time.

3.3 Exploration of the parameter space[1]

3.3.1 Kill Distance and Attraction Points

The proposed method generates a wide variety of trees and shrubs, controlled by a small number of parameters and algorithm variations. Figure 3.12 illustrates the impacts of the number of attraction points N and kill distance d_k (parameters of the pipe model and decimation were also adjusted). Decreasing N and increasing d_k yields crowns that are increasingly sparse. In addition fewer attraction points lead to irregular branches. The reason is that, in this case, the addition or removal of a single attraction point to or from the set affecting a branch tip can significantly change the direction of branch growth. In contrast, with larger kill distances the set of attraction points affecting individual branch tips increases. The individual points matter less, and smoothly curved branches result.

---

[1] Parameters used to generate selected models are collected in Table 3.1.

(a) $N = 12000, d_k = 2D$ (b) $N = 12000, d_k = 20D$ (c) $N = 12000, d_k = 40D$

(d) $N = 1500, d_k = 2D$ (e) $N = 375, d_k = 20D$ (f) $N = 375, d_k = 40D$

Figure 3.12: Impacts of the number of attraction points, N, and the kill distance, d_k, on the tree form. The kill distance is expressed as a multiple of D, the distance between adjacent nodes of the tree skeletons.

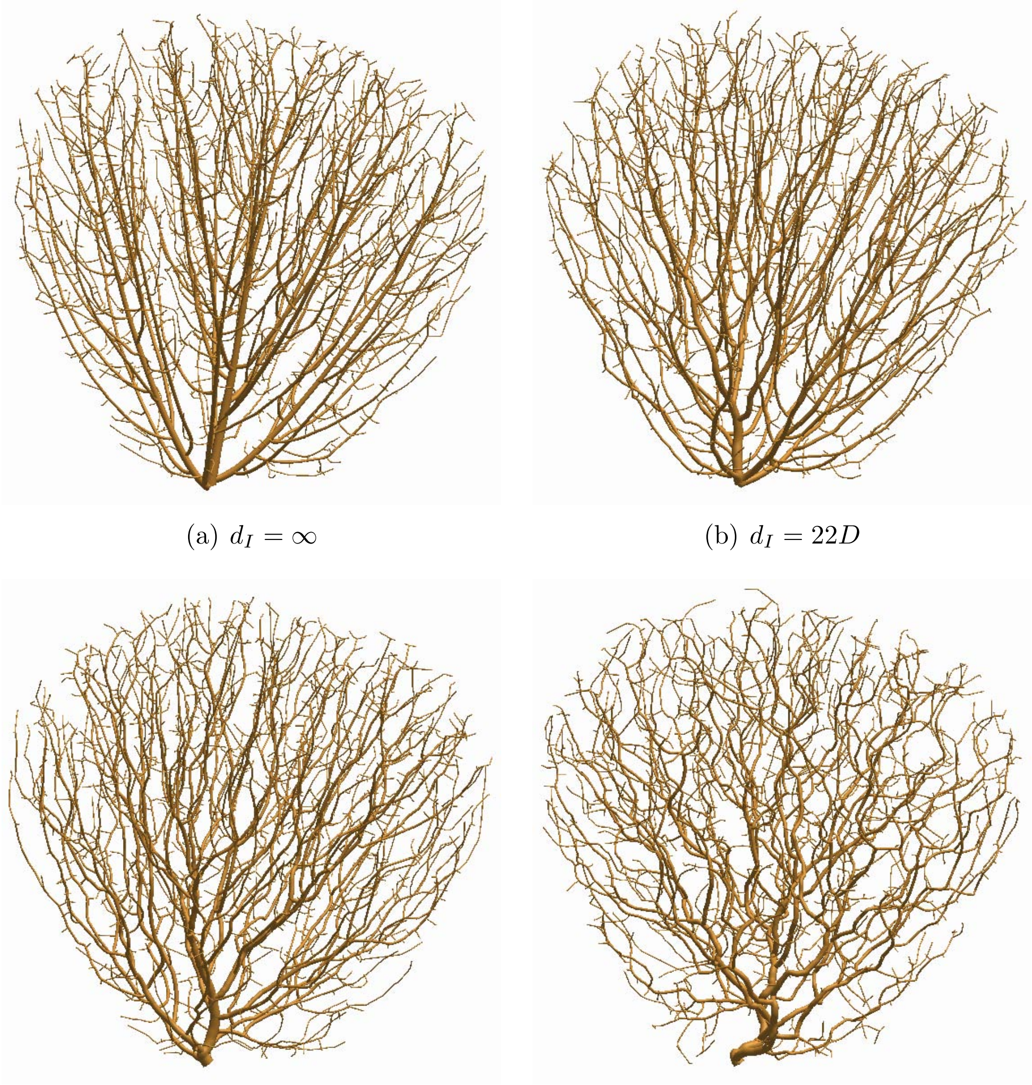
3.3.2 Radius of Influence

Figures 3.13 and 3.14 illustrate the role of the third numerical parameter of the algorithm, the radius of influence d_I on trees and shrubs. As d_I decreases, branch tips tend to meander between attraction points, coming into, then leaving their zones of influence, resulting in a wiggly or gnarly appearance.

3.3.3 Crown Shape

Figures 3.13 and 3.14 also illustrate the impact of the envelopes on crown shape: the shrubs in Figure 3.13 were generated using fan-shaped envelopes, whereas the trees in Figure 3.14 were generated using conceptually infinite envelopes (the simulations were stopped after a prescribed number of steps before any branch tips were closer than the radius of influence to the top of the tree crown). Further examples of the impact of the envelopes are given in Figure 3.15, which shows two trees generated using highly elongated cylindrical and conical envelopes. Comparison of Figures 3.14 and 3.15 also shows that narrower trees have a clearly delineated trunk, whereas in widely spread trees even the main limbs are highly ramified. This correlation between the overall form of the trees and their branching habits is an interesting emergent property of the algorithm, and captures the defining properties of excurrent (with the main stem) and decurrent (without a distinct main stem) tree forms [133].

In all examples considered so far, attraction points had a uniform distribution within the tree crowns, resulting in approximately uniform branch densities. However, in many trees and shrubs the branch density increases near the crown surface due to better access to light. We generate the resulting forms by only placing attraction points near the envelope. For example, Figure 3.16 shows a shrub generated



(a) $d_I = \infty$

(b) $d_I = 22D$

(c) $d_I = 17D$

(d) $d_I = 13D$

Figure 3.13: Impact of the radius of influence d_I on the form of shrubs.

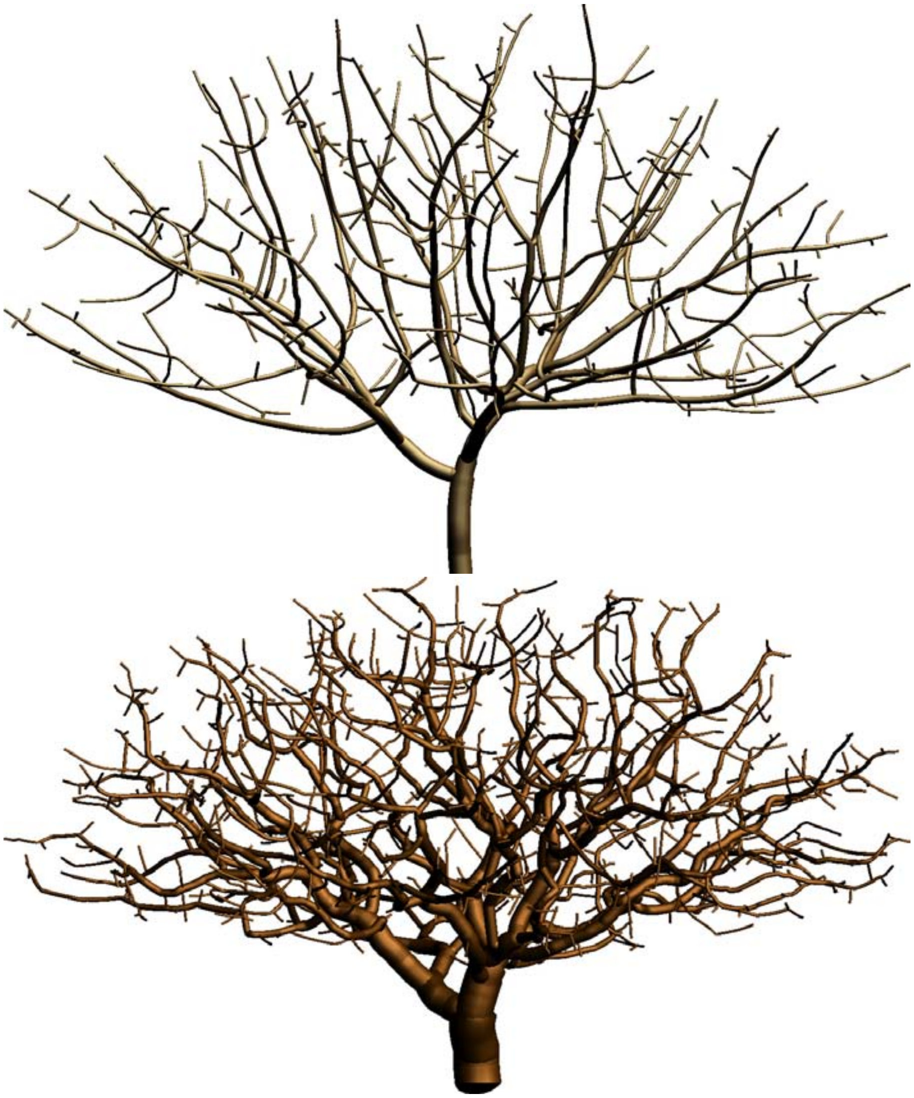


Figure 3.14: Impact of the radius of influence d_I on the form of trees. (Top) $d_I = \infty$; (Bottom) $d_I = 8D$.

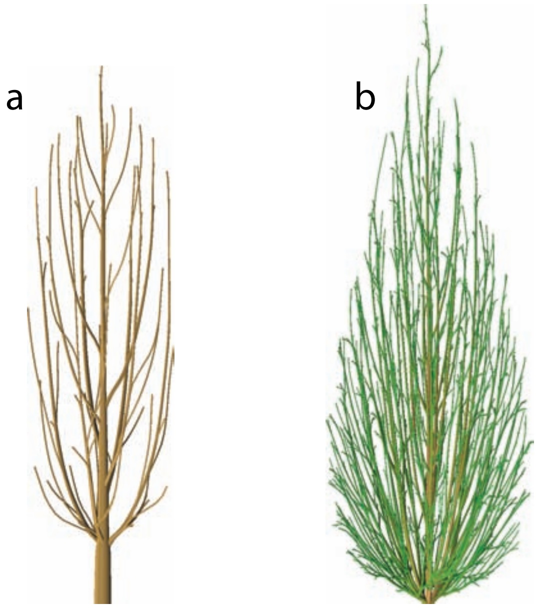


Figure 3.15: Impact of the envelope on the crown shape. a) columnar crown; b) conical crown.

with attraction points located exclusively near the envelope. The structure has an open, sparse branch system, with small twigs limited to the crown surface.

3.3.4 Addition of attraction points during simulation

The tree shown in Figure 3.17 was synthesized using an overdispersed point distribution generated using the method described in Section 3.2.2. Furthermore, new points were added while the tree structure was forming, with a gradually decreasing distance between the points (b_d is decreased with time). This led to the emergence of small twigs that filled the space between large branches. The resulting hierarchy of branch sizes gives the resulting structure the appealing appearance of a large, mature tree.

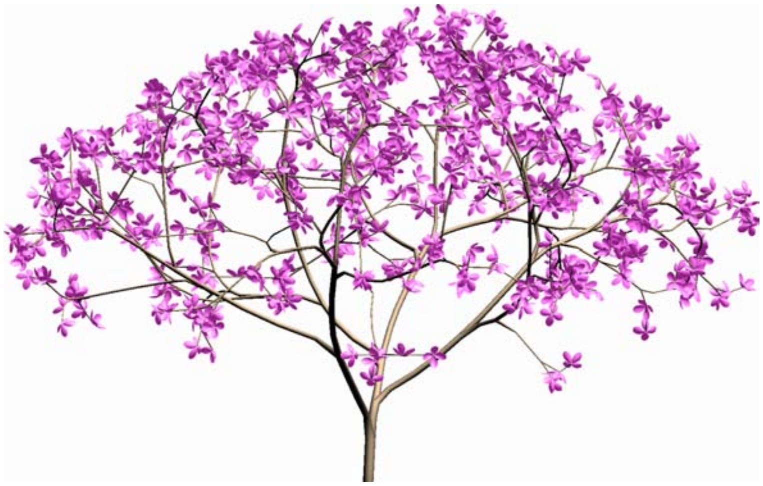


Figure 3.16: A shrub generated with attraction points placed exclusively near the envelope.

3.3.5 Tropisms

Tropism — the tendency of branches to turn in a particular direction — is a distinctive feature of tree architecture identified by Hallé, Oldeman and Tomlinson [52]. Tropisms and bending of branches due to their weight are incorporated into the models by vertically biasing the direction of branch growth (Equation 3.3). A slight upward or downward bias adds subtle curving to the branches of the tree, and is used in most examples.

In contrast, strong biases overcome the impact of attraction points and create particularly interesting effects. An upward bias may cause the branches to grow above their attraction points, before eventually turning down, resulting in a pendulous form (Figure 3.18a). This simple approximation of tropisms can create slightly pendulous forms, but is incapable of recreating the strongly pendulous form of some trees (such as weeping willows). When a strong downward bias is applied branches turn downwards away from the attraction points selecting them, leading to the emer-

Figure 3.17: A tree generated using continuously added attraction points.

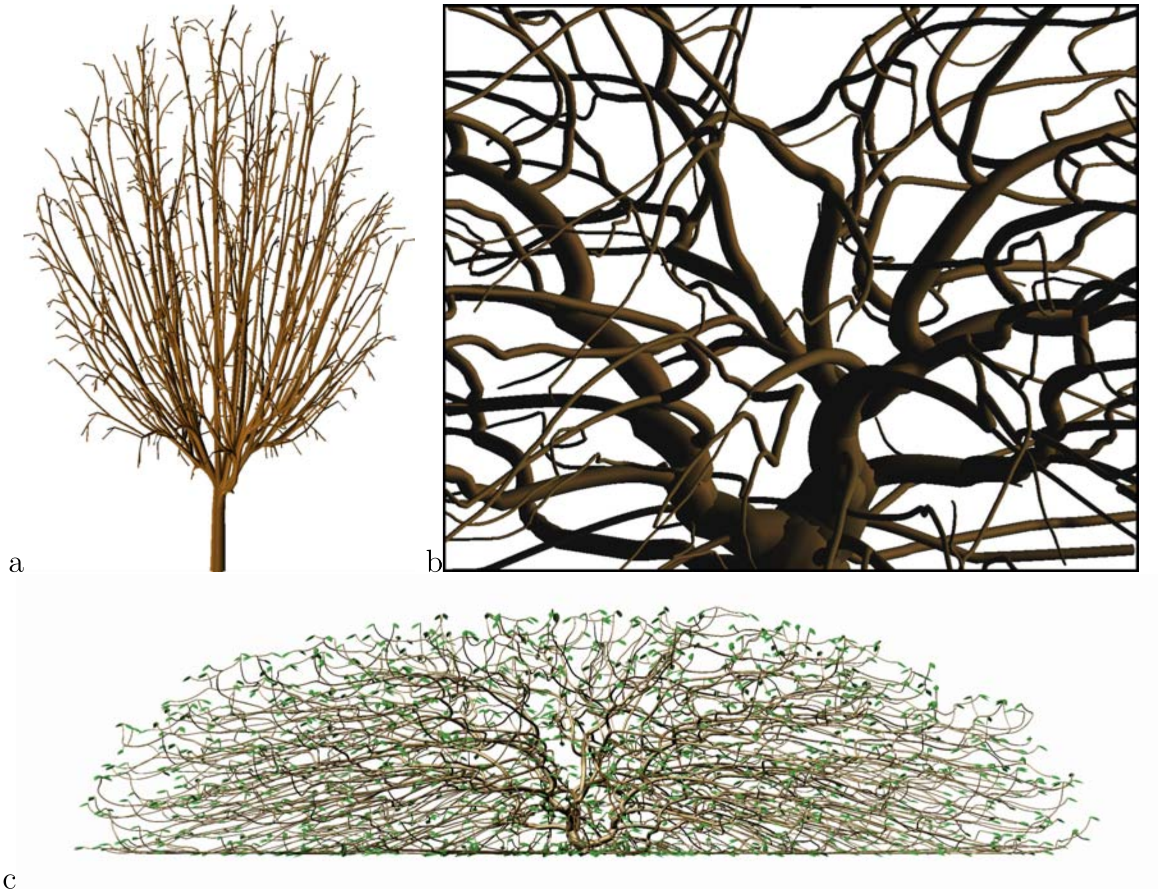


Figure 3.18: The impact of strong tropisms on the resulting form: a) a tree with pendulous branches resulting from a strong upward tropism; b) a model of a "crooked" tree (strong downward tropism); c) a low-lying bush or creeping plant (strong downward tropism)

gence of relay shoots on top of existing branches. This behaviour is observed in some shrubs (such as the *Sambucus nigra*), and *crooked* or *twisted* trees (such as *Fagus sylvatica var. tortuosa*, and the "crooked" mutant of *Populus tremuloides*) [134, 43] which is recreated in Figure 3.18b-c.

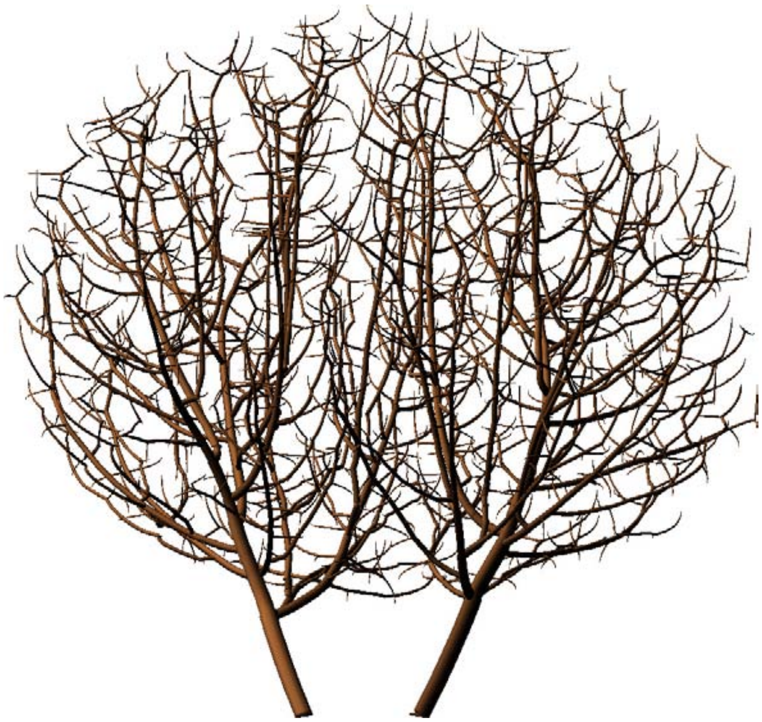


Figure 3.19: Two trees competing for space

3.3.6 Competition

In addition to the arrangement of branches within a single tree or shrub, the presented method can automatically capture the adaptation of the shape of the plants to the presence of their neighbours. This is illustrated in Figure 3.20, in which shrubs in a hedge (each initialized with a separate tree node) compete for space with their neighbours, and in Figure 3.19, where the same phenomena is recreated with two trees. The method can also account for the presence of obstacles to growth, by eliminating the attraction points beyond the surfaces of collision.

Figure 3.20: A hedge of shrubs competing for space.

3.3.7 Post-Processing

The described algorithm produces realistic-looking tree structures without any post-processing of skeletons. However the algorithm does not provide direct control over branching angles, which tend to be close to 90 degrees (especially in minor branches). To provides more control over the final form of trees and shrubs post-processing was used. The impact of moving the branching points and subdividing the skeleton is illustrated in Figure 3.21. In order to have a visual impact, these operations were applied to a decimated skeleton, with perceivably spaced nodes. The branches of the post-processed structure are smoother, and the branching angles are smaller, than in the original tree.

During the last stage of model construction the branching structure can be complemented with the addition of organs. In the case of the shrub shown in Figure 3.16, flowers were positioned and oriented on the branches using the parallel transport frame as a reference. This is sufficient to achieve a wide range of visual effects as demonstrated in Figures 3.22-3.24. In 3.22 and 3.23 leaves are positioned using the parallel transport frame, and in 3.24 compound leaves are placed at the tip of each branch. In Figure 3.25 the impression of pine trees of different age is created by placing a collar of pine needles at each internode of the thinnest branches. The size of these organs provides a sense of scale, and in Figure 3.26 many small leaves are used to give the impression of a haze of foliage in a large oak tree. Finally, in the tree shown in Figure 3.27, leaves were arranged around their supporting branches using the parallel transport frame, then brought to an approximately horizontal orientation in the world coordinate system.

3.4 Discussion

We have introduced an algorithm that generates trees and shrubs by simulating competition for space between growing branches. The initially empty space is represented as a set of attraction points, which are gradually removed as they are approached by the branches. Attributes and parameters of the model specify the shape and granularity of the empty space, the distance from which the branches can sense it, and the degree to which it can be penetrated by the branches. Additional parameters control the spatial and temporal distribution of the attraction points, allowing for increased branch density near the boundary of the tree crown and the formation of a hierarchy of branches with different sizes. A directional growth bias makes it possible to approximate the effects of tropisms and branch bending due to their weight. Model parameters correlate well with the notions used to characterize the appearance of trees and shrubs in landscaping (e.g., spreading or columnar crown shape, excurrent or decurrent branching habit, open or dense branch system, ascending or pendulous

Figure 3.21: The effect of post-processing a tree skeleton. (Top) The original structure. (Bottom) The structure obtained obtained by decimating the skeleton, moving the branch points, and subdividing the skeleton.

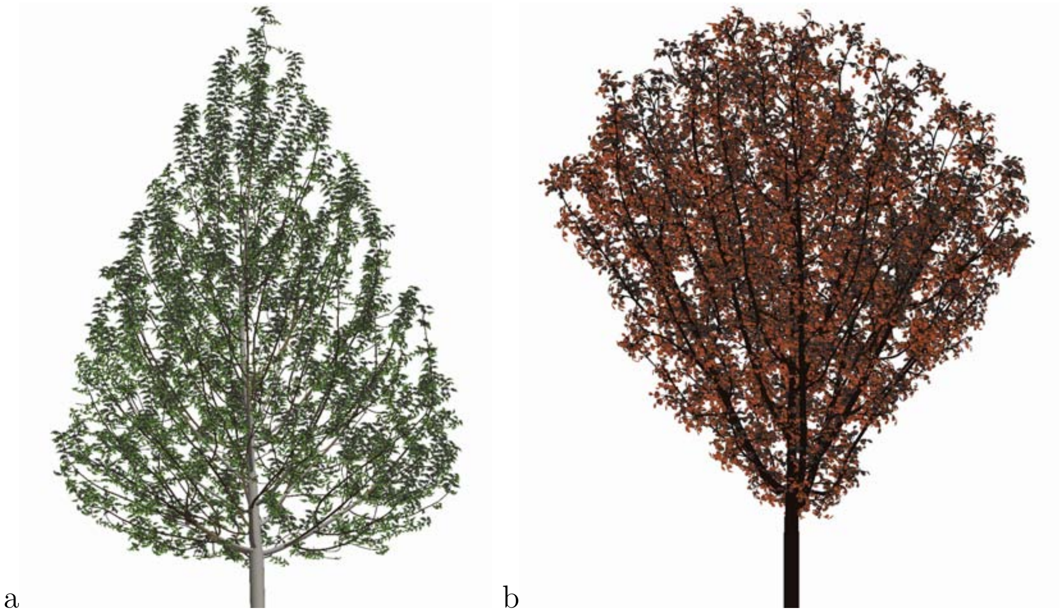


Figure 3.22: a) A birch tree. b) A tree with autumn foliage.

Figure 3.23: A tree with dense foliage.

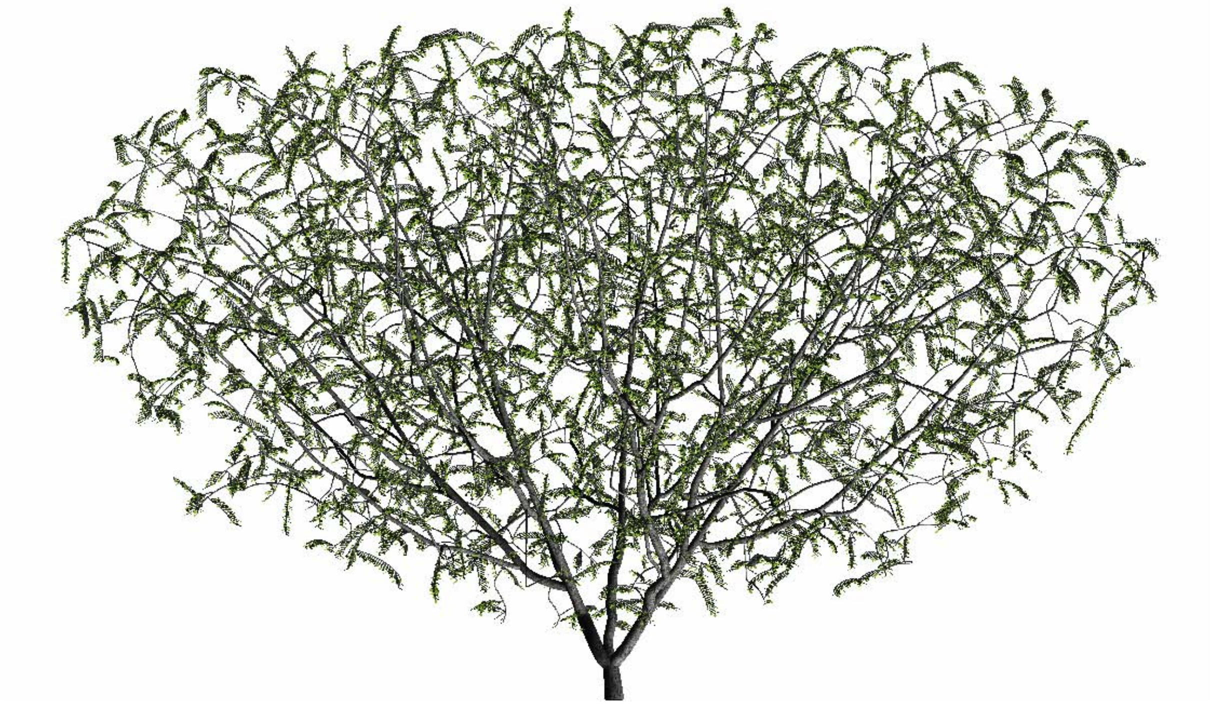


Figure 3.24: A tree with compound leaves.

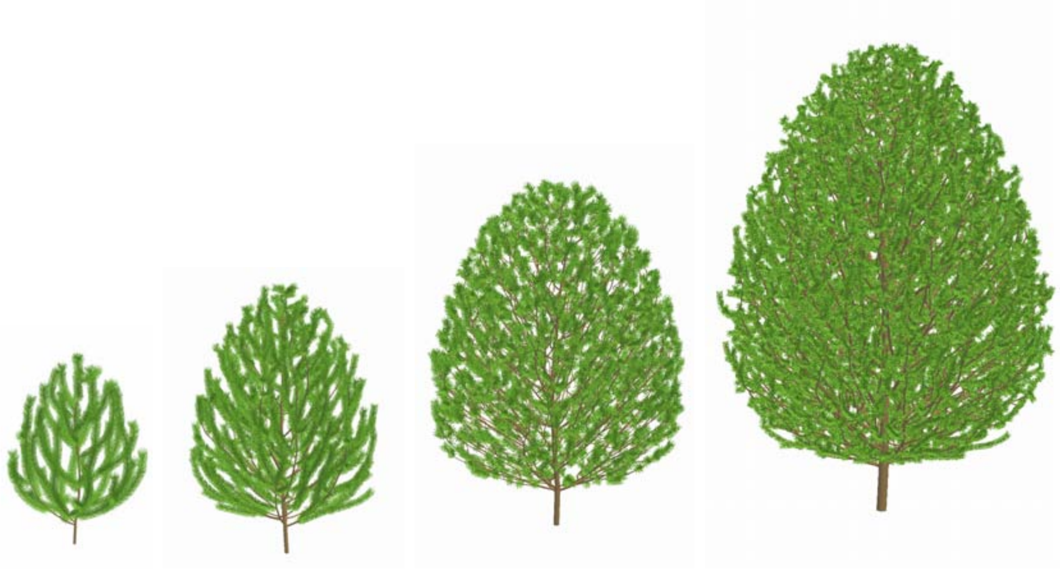


Figure 3.25: Four trees representing pine trees of different ages.

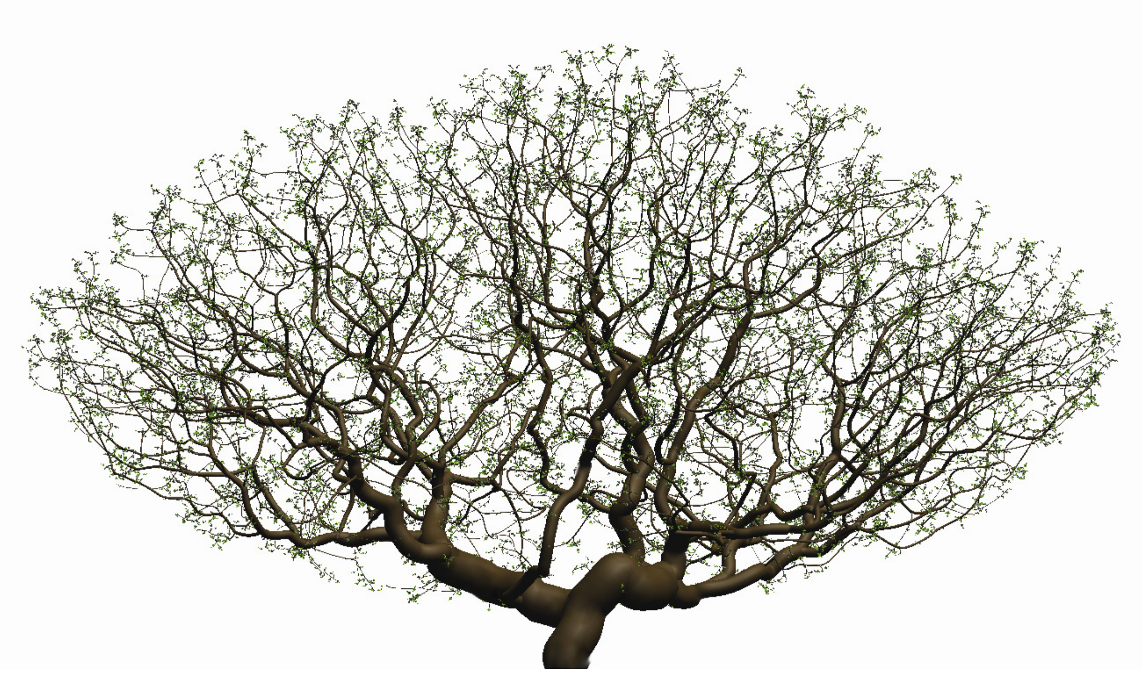


Figure 3.26: An Oak tree.

Figure 3.27: A tree with leaves.

branch orientation [133]), and can be tuned to generate diverse branching structures.

The proposed method is particularly useful in simulating irregular forms of temperate-climate deciduous trees. These forms are difficult to capture with older modeling methods, which emphasize recursive aspects of tree structure [109, 128, 25, 70]. The models generated with the space colonization algorithm are visually plausible even as bare trees and shrubs, without leaves that could potentially mask shortcomings of the branching structures. In particular, branch intersections are prevented by the nature of the algorithm. When needed, the generated branching structures can be complemented with leaves, flowers, buds, and fruits.

Although the space colonization algorithm has been formulated in abstract geometric terms, it is biologically justifiable. In nature, competition for space is likely mediated by quantity and quality of light [113, 5]. It has been previously postulated [147, 162, 109, 44] that this competition significantly affect plant form, and therefore should be incorporated into plant models. Our results amplify the observation of Novoplansky and Sachs that competition for space plays an important role in determining the overall branching structure of temperate-climate trees and shrubs and in fact suggest that it is the dominant mechanism.

Name	Figure	d_I	d_k	b_d	N	$\hat{g}$	n
Param. search N and d_k	3.12	∞	—	0	—	-0.1	2.5
Param. search d_I	3.13	—	$5D$	0	4000	0.2	2.5
Tree with a large d_I	3.14(Top)	∞	$10D$	0	2500	0.0	2.5
Tree with a small d_I	3.14(Bottom)	$8D$	$5D$	0	2500	0.0	2.5
Columnar crown	3.15a	∞	$20D$	0	6000	0.4	2.3
Conical crown	3.15b	∞	$5D$	0	6500	0.995	2.0
Mature tree	3.17	∞	$5D$	$20D - 5D$	5 per step	0.35	2.2
Strong upward tropism	3.18a	∞	$3D$	0	5000	0.9	2.0
Strong downward tropism	3.18b	$30D$	$10D$	0	650	-0.7	2.5
Two trees	3.19	∞	$2D$	0	4000	0.0	2.5
Hedge	3.20	$60D$	$5D$	0	12000	0.2	2.2
Original skeleton	3.21(Top)	$30D$	$15D$	0	4500	0.0	2.5
Birch tree	3.22a	∞	$15D$	0	5000	-0.2	2.1
Tree in autumn	3.22b	∞	$2D$	0	2200	0.1	2.5
Oak tree	3.26	$13D$	$8D$	0	8000	-0.1	1.7

Table 3.1: Values of parameters used to generate selected models. d_I: radius of influence; d_k: kill distance; b_d: minimum distance between samples; N: number of attraction points; $\hat{g}$: bias vector, only vertical component is shown (other components are zero); n: parameter used to calculate vein width (Equation 3.5).

Chapter 4

Modeling leaf venation patterns I: Basic models with marginal and uniform growth[1]

Venation is a functionally important and visually conspicuous component of plant leaves. Together with spiral phyllotaxis and the branching structures of tree architecture, venation patterns are among the most admirable aspects of the natural beauty of plants. Yet, in comparison, venation patterns and their development are poorly understood [27], which makes the visual modeling of venation patterns a particularly challenging and timely problem. As a step toward its solution we propose a model inspired by the current theories of hormonal control of vein morphogenesis. To this end, the geometric assumptions introduced to model trees are re-interpreted to make the model consistent with current understanding of vascular patterning.

The model presented in this chapter generates visually realistic venation patterns, reproduces in part their natural diversity, and captures the close relation between venation and leaf shape. For image synthesis applications, this model offers a useful alternative to scanned textures when leaf specimens are not readily available, leaves are not flat (and therefore are difficult to scan), a large number of leaf models with different yet related venations is needed, leaf development is animated, or when the topology of the leaf venation is needed. The model can also be used as a stepping stone to study and visualize leaf venation patterns for biological purposes. As when

---

[1] This chapter is based on [142]

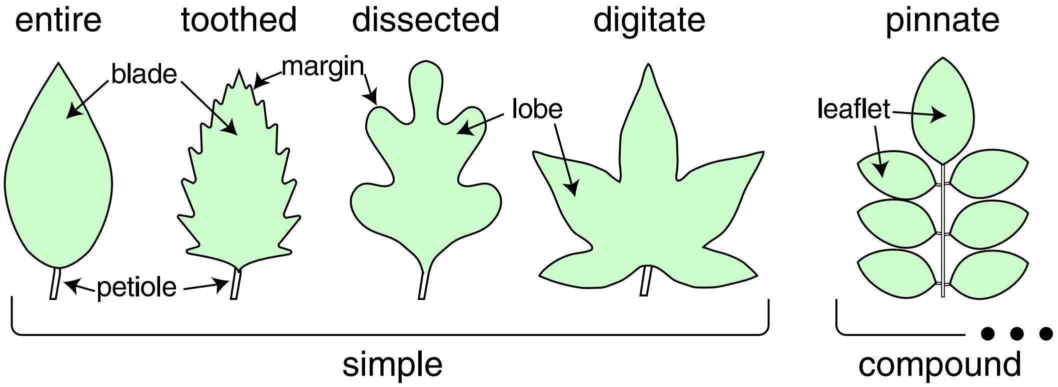


Figure 4.1: Terms pertinent to the description of leaf shapes.

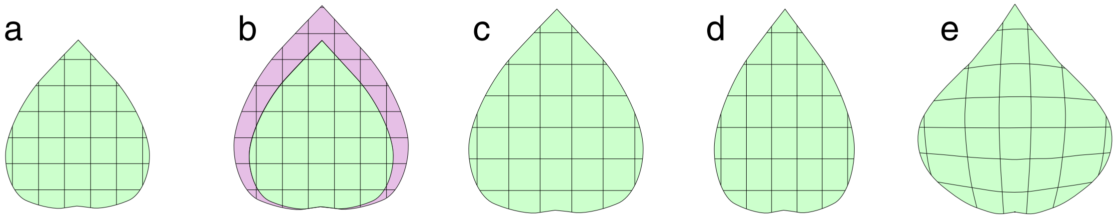


Figure 4.2: Varieties of leaf growth. A sample leaf (a) and the results of its: (b) marginal growth, (c) uniform isotropic (isogonic) growth, (d) uniform anisotropic growth, and (e) non-uniform anisotropic growth.

trees were modeled, realistic visualization plays a critical role as an element of model evaluation and validation [123], since current objective measures for comparing complex venation patterns with reality capture only a limited set of features [13]. This chapter outlines the basic algorithm using simple growth, and the essence of developing patterns is not addressed. Modeling the development of vascular patterns benefits from more sophisticated techniques and is discussed in the following chapter.

4.1 Background and related work

4.1.1 Leaf shape

Venation patterns are strongly correlated with leaf shapes [27] and thus must be considered in that context. A useful summary of the terminology for describing leaf shape is given by Judd *et al.* [72]. A typical leaf consists of a *blade* (*lamina*), attached by a *petiole* (*stalk*) to the stem (Figure 4.1). *Simple* leaves have a single, connected blade. A simple leaf is *entire* if its *margin* (edge) forms a smooth arc, *toothed* if the margin has small protrusions, and *lobed* if the margin is significantly indented, dividing the blade into distinguishable *lobes*. Lobed leaves are further categorized as *dissected*, with the indentation approximately perpendicular to the leaf axis, and *digitate*, with the lobes organized radially (like fingers on a hand). In contrast to simple leaves, *compound* leaves have blades partitioned into separate subunits called *leaflets*. Compound leaves are not considered directly, as it is assumed that their venation can be modeled at the level of individual leaflets.

4.1.2 Taxonomy of leaf growth.

The development of venation patterns is correlated with the growth of leaf blades. In this chapter a number of simple growth types are considered. Growth is *marginal* if it is concentrated on the border and *diffuse* if it is spread throughout the surface [141]. Diffuse growth is *isotropic* if expansion is equal in all directions at each point in the leaf blade, and *anisotropic* otherwise. Furthermore, growth is *uniform* if the growth is the same at all points of the surface, and *non-uniform* if it is not. A uniform isotropic growth is called *isogonic* [20]. These variations are illustrated in Figure 4.2.

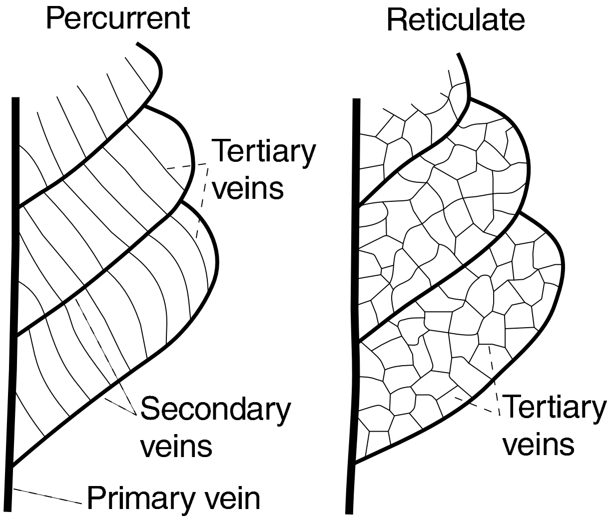


Figure 4.3: Some terms pertinent to the description of venation patterns.

4.1.3 Venation patterns

Leaf venation patterns are described in this work using the terminology of Hickey [58] and its simplification by Judd *et al.* [72]. A fundamental notion is that of vein *order*. Generally, first-order veins are the widest veins originating at the leaf *base* (the point of attachment to the petiole), and finer veins and veinlets have progressively higher orders (Fig. 4.3). Venation patterns are correlated with the taxonomic groups of plants and with the shapes of leaves. Leaves of monocotyledons (Fig. 4.5) usually have approximately parallel *primary* (first-order) veins, which is consistent with the highly elongated leaf shape and wide leaf base. Dicotyledons with simple entire leaves often have *pinnate* venation, characterized by a single primary vein (the *midvein*) that originates at the base and extends towards the leaf tip (Fig. 4.4). Dicotyledons with digitate leaves typically have *actinodromous* venation, in which three or more primary veins diverge radially from a single point (Fig. 4.6). Primary veins support

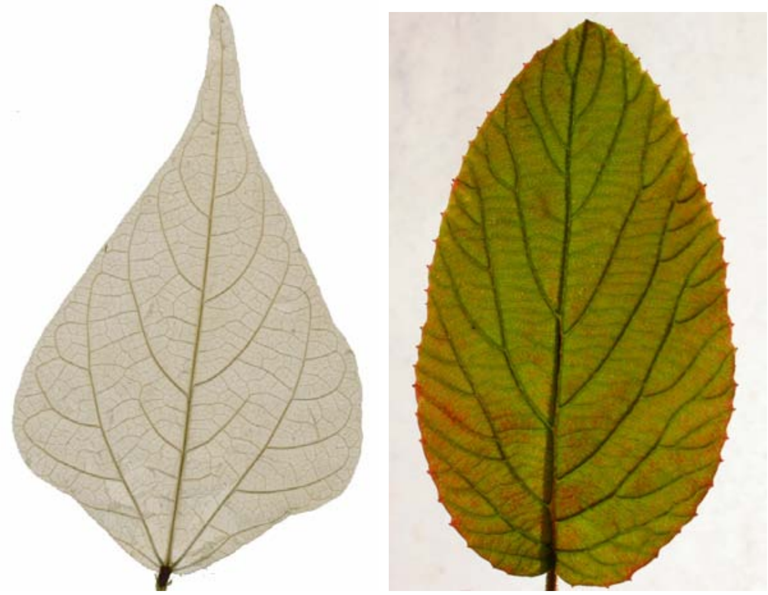


Figure 4.4: Examples of venation patterns obtained from dicotyledons. On the left is a closed pattern, and an open pattern is shown on the right

sequences of *secondary* (*lateral*) veins, which may branch further into higher-order veins. The secondary veins and their descendants may be *free-ending*, which produces an *open*, tree-like venation pattern, or they may connect (*anastomose*), forming loops characteristic of a *closed* pattern. *Tertiary* and higher-order veins usually link the secondaries, forming a ladder-like (*percurrent*) or netlike (*reticulate*) pattern (Fig. 4.3, 4.7).

4.1.4 Mechanism of vein pattern development

The most widely accepted theory of vein pattern formation is the *canalization hypothesis*, proposed by Sachs [145]. According to this hypothesis, vein patterning is initiated by a signal that propagates in the leaf blade, and converges on future veins.

Figure 4.5: Examples of venation pattern in monocotyledons

Figure 4.6: Example of a venation pattern with actinodromous venation

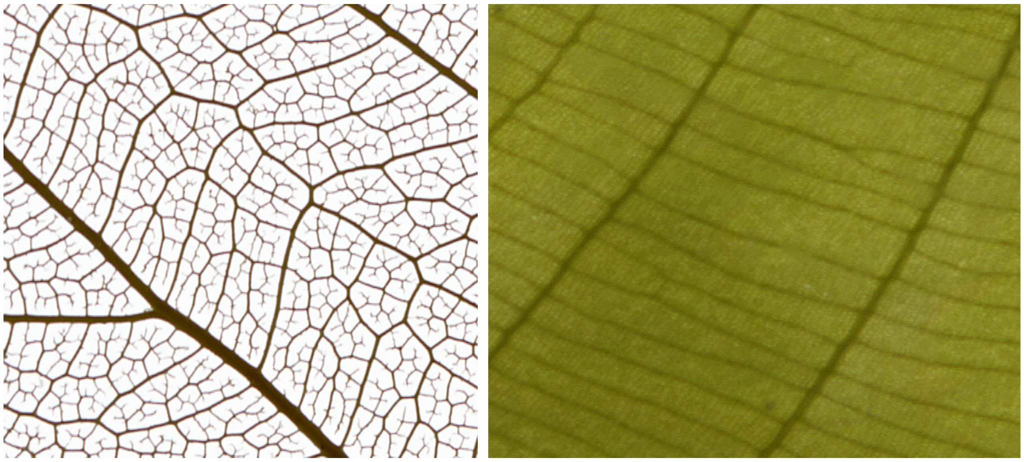


Figure 4.7: Examples of tertiary and higher-order veins. (Left) High-order reticulate network with free-ending veins. (Right) Ladder-like percurrent tertiary veins.

At least part of this signal is the plant hormone *auxin* [153, 146]. Auxin is thought to originate in the leaf blade and flow toward existing veins, which transport it to the leaf base. During this flow, auxin is *canalized* into narrow paths, in a manner analogous to water carving riverbeds in soft terrain [146]. These paths will differentiate into new vein segments. Experimental evidence suggests that auxin sources may be discrete [2].

Sachs' analogy between vascular pattering and canalization implies that the tissue of the leaf can respond to and enhance auxin flux anisotropically. This hypothesis is supported by the expression of the protein *PIN1* observed in developing leaves, which is expressed in larger regions that are eventually refined to vascular precursors [150]. Additionally, PIN1 is localized non-uniformly on cell walls and is thought to export auxin from the cell, thus indicating the direction of auxin transport in the leaf [150]. The canalization hypothesis is supported by increasing amounts of molecular data, but work is hindered by the difficultly of measuring auxin concentration directly [10]. Additionally, the details of vascular patterning are still not well understood from either a biological [27] or modeling [140] standpoint, and are further obfuscated by the complex and apparently contradictory roles that auxin and PIN1 play in other patterning processes in plants [10].

4.1.5 Models of vein pattern development

The first computational model of venation patterns was a four-substance reaction-diffusion model proposed by Meinhardt [100]. This model postulates that auxin is produced everywhere in the leaf blade and diffuses towards veins, which remove it from the system. The resulting gradient of auxin concentration directs differentiation of new veins towards regions where vein density is low. This model generates branching networks with occasional anastomoses, but these networks are not visually similar to real venation patterns.

The canalization hypothesis was the basis of a computational model developed by Mitchison [105]. Mitchison assumed that the transport depend on the flux itself. The resulting feedback loop between transport characteristics and flux leads to the formation of high-flux canals between sources of auxin and sinks (the existing veins) distributed throughout the leaf blade; these high-flux canals become new veins. This model is particularly attractive due to its consistency with the putative molecular mechanisms of active auxin transport [139]. As a result, there has been significant work extending Mitchison's model to examine the range of phenomena it can reproduce [140], the relation between PIN1 and auxin [38], the establishment of loops [37], and the impact of marginal growth and differentiation on pattern formation [41]. A related model, which modeled the formation of higher order veins, was proposed by Dimitrov et al. [29]. This model forms veins by following the auxin gradient towards a maximum.

Somewhat troublesome is the apparent contradiction between the dynamics of auxin transport as proposed by Mitchison and those observed by Reinhardt et al [132] in phyllotactic patterning, where PIN1 is thought to respond to auxin concentration and transport it up the gradient (as opposed to with the flux). Simulating the dynamics proposed by Reinhardt in a growing cellular space is sufficient to recreate the positioning of primordia as observed in the shoot apex of a plant [158, 71]. Merks et al. attempted to reconcile this inconsistency by modifying the dynamics of auxin transport proposed by Jonsson et al. [71] and Smith et al. [158] to operate in the leaf blade [103]. Their model recreates the formation of a midvein by creating an auxin maximum that travels as a wave from the tip to the base of the leaf leaving a polarized file of cells in its wake.

In contrast to Mitchison's and Merks et al.'s models, which simulate the transport processes leading to vein formation, the model of angiogenesis (vasculature formation) proposed by Gottlieb [49] directly simulates the outcome of these processes: the insertion of new veins. In Gottlieb's model the venation pattern is embedded in a growing medium. Growth increases the distance between the existing veins, allowing new sources of an auxin-like signal to be inserted in the resulting spaces. New veins form by connecting these sources to the closest older veins, or — in the case of anastomoses — to all veins within some distance range. The expression of the algorithm in geometric terms allowed Gottlieb to generate complex patterns, approximating diverse vascular systems in animals and plants. Nevertheless, the realism of these patterns is limited by several simplifying assumptions. Positions of sources are constrained to the grid, which results in a visible regularity in the layout of vein endings. Veins grow only to a single source and reach them in a single step making all vein segments straight. Growth is simulated by doubling the (linear) grid size in every simulation step, which precludes continuous simulation of growth. As vein segments are straight, and segments double in length in each growth step, artificial-looking long straight lines are produced running through the pattern. Unnatural sharp angles may form between anastomosing veins.

Leaf growth also plays an essential role in the biomechanical model of vein pattern formation proposed by Couder *et al.* [24]. This model exploited a hypothetical analogy between vein pattern formation and fracture propagation in a stretched material. Although the authors used physical experiments to test their model, the same analogy could underlie computer simulations, for example based on the fracture-simulation software described in [35].

Rodkaew et al.[137] proposed an algorithm specifically for the purpose of generating vein patterns for the synthesis of realistic leaf images. Their algorithm begins with a set of particles distributed over the leaf blade. These particles move towards a sink placed at the base of the leaf; in their motion they are attracted towards each other and merge if a threshold distance between particles has been reached. The venation pattern is formed by the particle trajectories. Some of the generated patterns suggest the appearance of primary and secondary veins in leaves with open venation. However, the conceptual framework and the resulting generating procedure have not been related to the current biological understanding of vein morphogenesis. Consequently, it is difficult to improve the results by incorporating biological

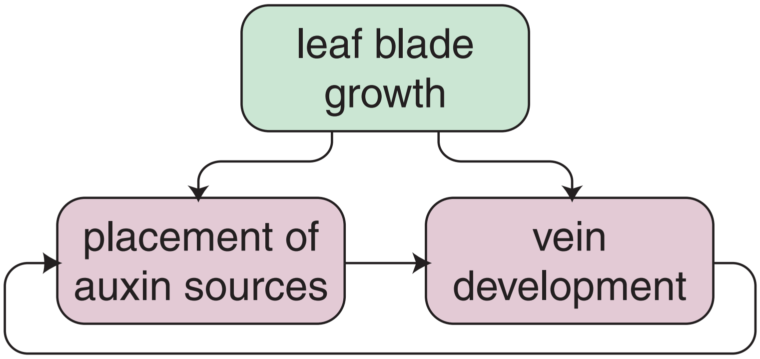


Figure 4.8: Hypothetical causal relations underlying vein pattern development.

knowledge.

4.2 Generation of open venation patterns

We assume that leaf venation patterns develop in a feedback process, coupled with leaf growth, in which discrete auxin sources direct the development of veins, and the veins reciprocally affect the placement of sources (Figure 4.8). Specifically, auxin sources are assumed only to appear far from the existing veins and other sources. Although different parts of this process have been described in the biological literature and included in previous models, the only explicit reference to the feedback loop between source placement and vein development was made by Gottlieb [49]. Like Gottlieb's model, the model presented here is expressed in geometric terms and uses proximity criteria to determine new vein locations. However, the algorithm operates in continuous space, and does not rely on the simplifying assumptions introduced by Gottlieb. As a result, the venation patterns generated and presented here are very different from those created using previous methods.

Another key inspiration for this work is the model of vascular patterning proposed by Mitchison [105]. Although Mitchison's model operates at the scale of individual veins, it offers valuable insights for constructing models of whole patterns. This model was reimplemented and the experiment shown in Figure 4.9 demonstrates that it tends to create a canal between a source and the vein node that is closest to it. These observation relate directly to Gottlieb's model, which in turn is the cornerstone of our model.

Below, the version of the algorithm that generates open venation patterns is described. The modifications that extend it to closed patterns are presented in Section 4.3.

4.2.1 Preliminaries

Input to the algorithm consists of: (1) the initial state (the initial shape of the leaf and the placement of the "seed" vein node or nodes), (2) functions and/or parameters characterizing leaf growth, and (3) parameters characterizing the interplay between the auxin sources and vein development.

The algorithm and data structures used to model open patterns are the same as those described previously for trees (Section 3.1.2), but additional parameters incorporating the impact of growth and affecting the placement of sources are used as well. In the context of vein patterning a slightly different terminology is used. Auxin sources are represented by a set S of attraction points s embedded in the leaf blade. As such, attraction points are referred to as *sources* and tree nodes become *vein nodes*. Additionally, the graph G containing the structure of the vein pattern is now called the *venation graph*.

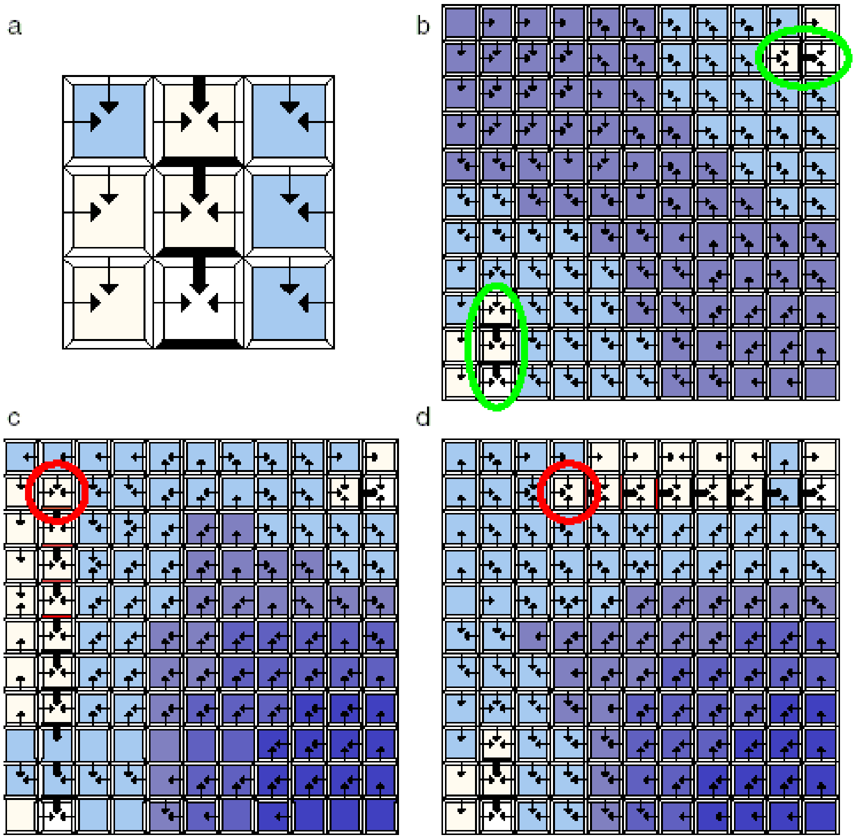


Figure 4.9: An experiment performed using Mithcison's model with two sinks and one source (courtesy of Anne-Gaëlle Rolland-Lagan). a) Explanation of the symbolic representation: The shade of blue in a cell indicates the concentration of auxin, with dark blue indicating high concentrations. The width of arrows denotes the flux between two cells, and the black lines on the edge of the cell indicates the polarity of the cell. b) The initial state of the model with two sinks, circled in green. c) When a source, circled in red, is added closer to the left sink than the right sink, a strand forms to the left sink. d) The process shown in (c) is repeated with the source placed closer to the sink at the right hand side. This time the strand connects to the sink on the right. Taken together these simulations show that when a source is added it tends to form a strand connecting it to the closest sink.

4.2.2 The initial state of the model

The initial leaf shape is specified interactively by the user, as a parametric curve that defines the leaf contour. In the case of toothed leaves, protrusions are introduced algorithmically, by summing triangular waveforms of different amplitudes and frequencies. The initial venation graph usually has a single vein node, which coincides with the attachment point of the blade to the petiole. In the case of leaves with parallel venation, the initial graph includes several isolated nodes, positioned along the leaf base. In all cases, positions of the initial points are specified by the user.

4.2.3 The simulation loop

Our algorithm consists of simulating within an iterative loop the three processes shown in Figure 4.8: leaf blade growth, the placement of auxin sources, and the addition of new vein nodes. We assume that the pattern of veins and auxin sources in the leaf blade does not affect leaf growth. This is a simplifying assumption that allows growth to be modeled using a purely descriptive approach.

Leaf-blade growth. Given the initial leaf shape at time t_0 and the growth description, the leaf-growth model must be able to determine leaf shape at any time $t_1 > t_0$ and, for any material point p embedded in the leaf blade at time $t_1 \geq t_0$, to find the position of that point at any time $t_2 > t_1$. Two methods to model leaf growth are outlined here, which allow the development of patterns to be simulated. However, this chapter focuses on producing realistic final patterns. The problem of precisely matching the sequence in which veins and sources emerge is addressed in Chapter 5, where non-uniform anisotropic growth is considered.

Marginal growth is modeled by scaling the leaf edge with respect to the attachment point of the leaf to the petiole, without modifying the position of sources or vein nodes. We assume that the scaling factor σ is a linear function of time, $\sigma_{t+\Delta t} = \sigma_t + \Delta\sigma$, where $\Delta\sigma \geq 0$ is a constant describing the increase of margin size per simulation step. This model is not applicable to dissected leaves, since in this case the scaling of the edge may gradually erode parts of the leaf blade.

Uniform growth (isogonic) can be regarded as a scaling of the entire leaf, including veins and auxin sources that exist at time t, using the same formula as above, $\sigma_{t+\Delta t} = \sigma_t + \Delta\sigma$. For computational efficiency, instead of scaling up the leaf, we scale down the unit distance λ used while inserting and removing auxin sources: $\lambda_t = \lambda_0 \cdot \sigma_0/\sigma_t$.

Auxin source placement. Auxin sources are assumed to emerge at locations farther than a threshold *birth distance* b_s from the set of all other sources, and farther than a threshold birth distance b_v from the set of all vein nodes. These points are computed using the *dart-throwing* algorithm [23, 104]. This algorithm consists of repeatedly generating points distributed at random (with a uniform distribution) within a given domain, then testing each new point s' against all points already in the set. A point is accepted as a new member of the set if it is sufficiently far from other points. In this application of the SCA, it is necessary to "throw darts" at every iteration of the algorithm. The number of darts per step per unit area of the leaf, denoted ρ, controls the regularity of the venation pattern (Section 4.5).

In practise, computation of the set of auxin sources depends on the assumed leaf-growth kinetics. For uniform growth, the initial set is usually empty, and new sources are added using dart throwing after each growth step, whereas for marginal growth, new sources appear only on the margin. In this case, we precompute the set of sources in a square that embeds the maximum leaf size, and include new sources as they appear within the expanding leaf contour.

Sources continue to exist until they are removed due to the proximity of veins that grow towards them. In the case of open venation patterns, a source s is removed when at least one vein node v is closer to s than the *kill distance* d_k.

Other than the exceptions noted above, the description of the SCA algorithm given in the previous chapter applies. Differences between the operation of the algorithm in the context of tree and leaf modeling are highlighted in the following example.

4.2.4 Example

The execution of the algorithm for generating open venation patterns is illustrated in Figure 4.10. We begin following it at the stage when the vein system consists of three nodes (black disks with white centres) and there are four auxin sources (red disks) (a). Initially, the open algorithm proceeds as was described for trees in §3.1.3, with steps (a-g) corresponding to (a-g) in Figure.3.5. The only exception involves the bias vector used for trees, which is not considered in the context of leaf venation. For completeness we repeat the description of steps (a-g) here. First, each source is associated with the vein node that is closest to it (b, red lines); this establishes the set of sources that influences each node. The normalized vectors from each vein node to each source that influences it are then found (c, black arrows). These vectors are added and their sum normalized again (d, violet arrows), providing the basis for locating new vein nodes (d, violet circles). The new nodes are incorporated into the venation, in this case extending the midvein and initiating a lateral secondary vein (e). The neighbourhoods of sources (red circles) are now tested for the inclusion of (the centers of) vein nodes (f). The neighborhoods of the two leftmost sources have been penetrated by the veins, as indicated by the bolder representation of the corresponding circles. The affected sources are removed from the set of sources (g).

Unlike when trees are modeled, growth and the addition of subsequent sources play important roles. This causes the examples to diverge when the leaf grows (h). In this example we have assumed marginal growth, so the existing sources and vein nodes are not moved. The candidate new sources are now placed randomly within the expanded blade (i). Their neighbourhoods, indicated by dashed circles, are checked for the inclusion of (the centres of) previously placed vein nodes and sources. The only candidate source with an empty neighbourhood is incorporated into the set of sources (j) and the vein nodes closest to these sources are identified (k). This is the beginning of the next iteration of the algorithm execution, with stages (j) and (k) corresponding to the stages (a) and (b) from the previous iteration.

4.2.5 Calculation of vein width

The final component of the algorithm is the calculation of vein width. Width is calculated using Murray's law [107] and employs Equation 3.5 as described in Section 3.2.3. Although the general applicability of Murray's law to leaf venation is a matter of discussion, it has been supported by tests on sunflower leaves [141]. Calculation of vein width begins with the veinlets, assumed to have the minimum width, and proceeds towards the base of the leaf. In models that do not include a detailed pattern of tertiary veins, we additionally increase the width of primary and secondary veins at each node, to approximate the influence of veinlets that are not modeled explicitly.

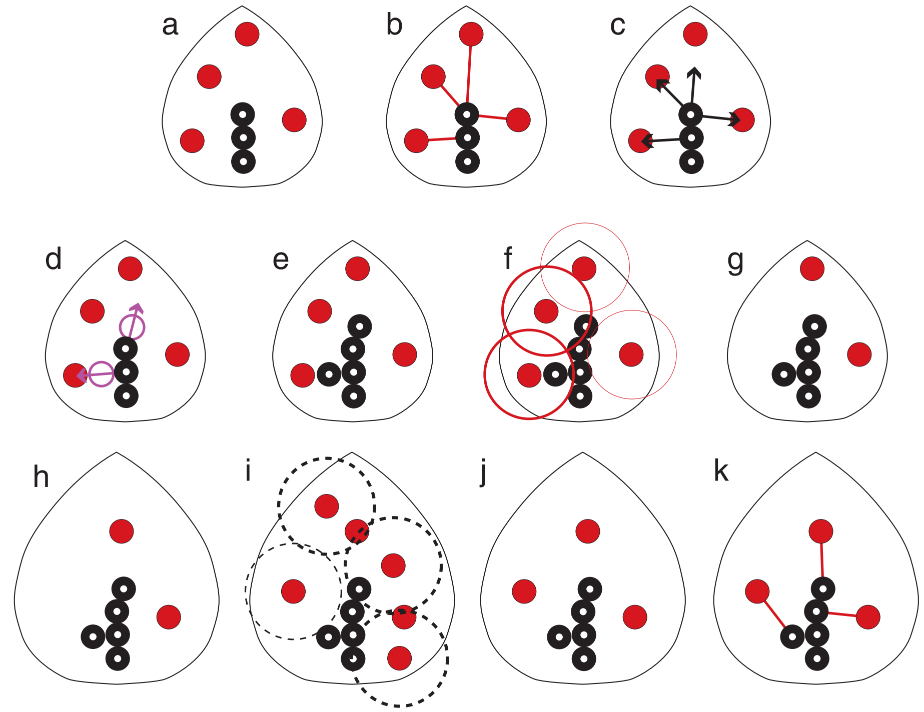


Figure 4.10: Illustration of the algorithm for generating open venation patterns. A detailed description is provided in the text

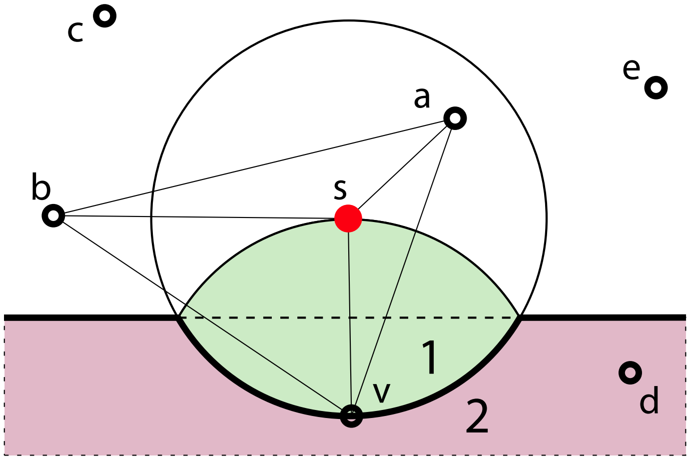


Figure 4.11: Illustration of the notion of relative neighbourhood.

4.3 Generation of closed venation patterns

In order to generate closed venation patterns, the previous algorithm is modified so that more than one vein may grow toward the same source. Hypothetically, such situations occur in nature when several veins are close to the same source, yet are relatively far from each other. We formalize this concept using the notion of *relative neighbourhood* [163, 67, 114]. Given a point set A, point $v \in A$ is a *relative neighbour* of a point $s \in A$ if and only if for any point $u \in A - \{v\}$ that is closer to s than is v, v is closer to s than to u. Mathematically, point v belongs to the relative neighbourhood of s, denoted $V(s)$, if and only if

$$(\forall u \in A - \{v\}) \|v - s\| < \max\{\|u - s\|, \|v - u\|\}. \tag{4.1}$$

This definition is illustrated in Figure 4.11. Points v, a and b are relative neighbours of s, with the relevant distances shown by lines; points c, d, and e are not. Shaded areas illustrate the influence of point v on the relative neighbourhood of s. Area 1 (green) is the locus of points that are closer to s than is v; this area must be empty

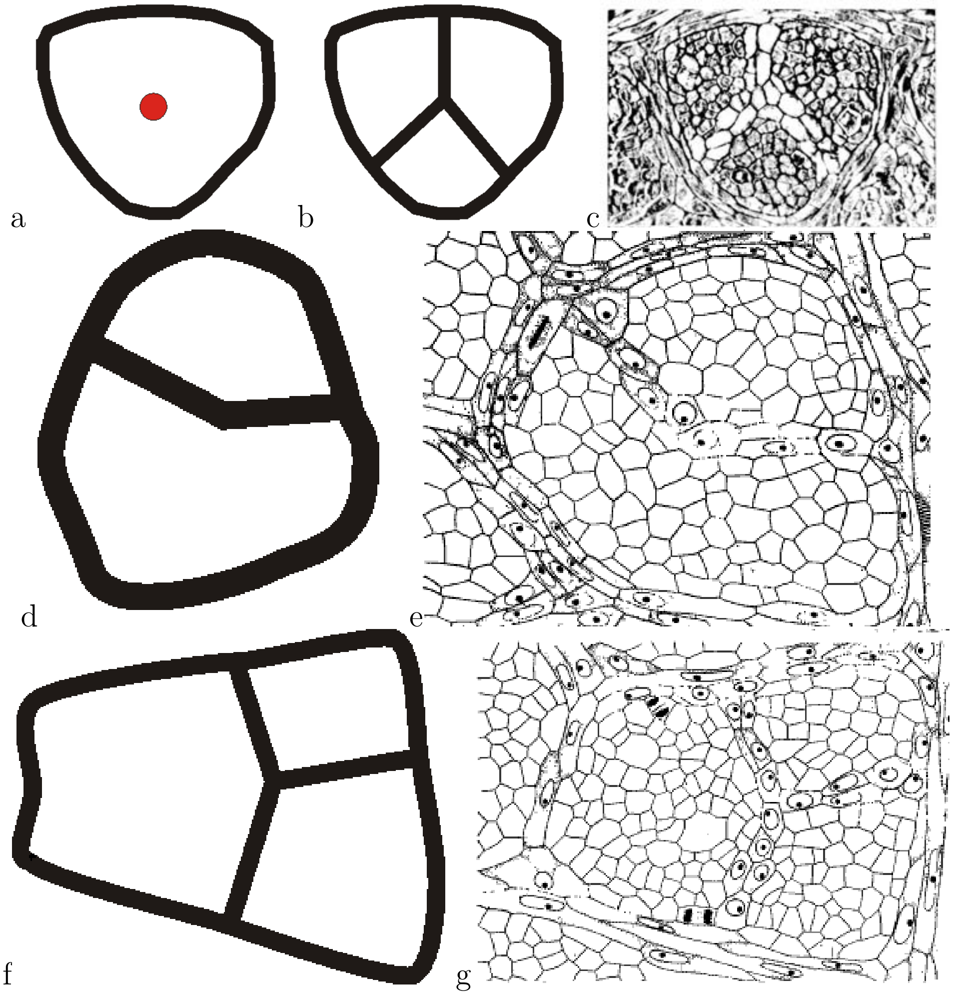


Figure 4.12: Comparison of the relative neighbourhood to the structure of highest order venation (sub-figure c,e and g), taken from Pray [120]. a) Existing veins bounding region seen in black, and the proposed auxin source represented as a red circle. b) Interior structure suggested by the relative neighbourhood. The same experiment is repeated with different boundary curves in sub-figure d and f.

in order for v to be a relative neighbour of s. Area 2 (pink) is the locus of points that are closer to v than to s, and therefore are excluded from $V(s)$ by v. The following illustration, Figure 4.12, compares structures created with the notion of the relative neighbourhood to those of actual vein patterns as observed by Pray [120], assuming a single source.

In general, the set $S(v)$ of sources that influence vein node v consists of all sources s such that v is a relative neighbour of s in the set $A_s = V \cup \{s\}$:

$$S(v) = \{s \in S | v \in V(s)\}. \tag{4.2}$$

Directions of vein development are calculated from $S(v)$ as in the case of open venation patterns and trees (Equation 3.1).

Veins growing towards the same source are unlikely to reach it at the same time. Thus, to create anastomoses, the source must remain active until it is reached by all veins that are growing towards it. We achieve this by modifying the criterion for source removal. When a vein node v is placed within the kill distance d_k of the source s, this source is not immediately removed; instead, the set $V(s)$ of all nodes influenced by s is tagged. In subsequent simulation steps, these tags are passed by vein nodes to their descendants. The source s is removed when all tagged veins have either reached it or left its zone of influence.

The final modification concerns the calculation of vein width. In our implementation, we assume that all child veins entering a vein node contribute equally to the width of the parent nodes.

4.4 Implementation of the algorithms

The proposed algorithms involve repeated tests of proximity amongst potentially large sets of points: hundreds of auxin sources against thousands of vein nodes. As before, execution times can be improved with space subdivision techniques, such as grids [9] or Voronoi diagrams [121, 114]. The implementation employed uses Voronoi diagrams, which are particularly useful in the creation of closed patterns. The time complexity of a single iteration of the open and closed algorithms, with and without Voronoi diagrams, is discussed below.

The open and closed algorithms were implemented using the `vv` programming environment [155]. Computations related to Voronoi diagrams and Delaunay triangulations (the dual of the Voronoi diagram) were preformed using the GNU Triangulated Surface library [119].

4.4.1 Time complexity of the algorithm for open patterns

Let c be the number of candidate auxin sources to be inserted, k be the number of existing sources, and n be the number of vein nodes. The straightforward algorithm for inserting new auxin sources tests whether a candidate source is farther than the birth distances b_s and b_v from any other source or vein node, respectively. Assuming that the number of new sources is small compared to $k+n$, this requires $O(c(k+n))$ distance calculations per iteration step. The algorithm for vein development finds, for every source s, the closest vein node v and thus requires $O(kn)$ calculations. The algorithm for source removal does not require additional distance calculations, since a source can be removed as soon as it is found to be closer to the nearest vein than the kill distance, k_d.

To improve the running time, separate Voronoi diagrams are used containing the set S of auxin sources and the set V of vein nodes, respectively. These diagrams can be constructed "from scratch" each simulation step in $O(n \log n + k \log k)$ time in two dimensions [114]. Updating the diagrams incrementally from one iteration step to the next improves on this time [114]. Given the Voronoi diagrams, a candidate auxin source is accepted if it is farther than b_s from the nearest vertex of the set S, and farther than b_v from the nearest vertex of the set V. The calculation of the nearest neighbour in the Voronoi diagram takes logarithmic time [121], thus, for c candidate auxin sources, the insertion tests take $O(c(\log n + \log k))$ time. The search for the vein nodes that are nearest to k sources require $O(k \log n)$ time. As in the straightforward approach, removal of sources does not involve additional calculations.

4.4.2 Time complexity of the algorithm for closed patterns

The critical component of the algorithm for closed patterns is the computation of relative neighbourhoods of the sources. A straightforward algorithm for computing the relative neighbourhood $V(s)$ of a source s is based directly on Equation 4.1. We suppose that $V(s) = V$, then consider every node $u \in V$ and eliminate from $V(s)$ every node v that is excluded from the neighbourhood of s by u. The remaining set $V(s)$ is the output of the algorithm. Unfortunately, the time needed to find the relative neighbourhood of a single source using this algorithm is $O(n^2)$; for k sources it is $O(kn^2)$.

A significant speedup is possible due to two computational geometry theorems: The relative neighbourhood graph is a subset of the Delaunay triangulation (the dual of the Voronoi diagram) [163], and the average degree of a vertex in a Delaunay triangulation is 6 [114]. From the first theorem, it follows that the search for the relative neighbourhood $V(s)$ of s can be limited to testing every node v adjacent to s in the Delaunay graph of $A_s = V \cup \{s\}$ (*i.e.*, the *Delaunay neighbourhood* of s) against all points u in the set A_s. Points v that pass the criterion of Equation 4.1 form the relative neighbourhood. Since the average number of Delaunay neighbours is constant, this approach reduces the average time complexity of finding the relative neighbourhood of s to $O(n)$; thus, finding the relative neighbourhood of every source can be performed in average time $O(kn)$.

The above analysis is true assuming that the Voronoi diagram, and thus the Delaunay graph, is given for each set $A_s = V \cup \{s\}$. In practise, we maintain the Voronoi diagram of the set of vein nodes V as in the algorithm for open venation patterns, and compute the Voronoi diagram for set A_s by inserting node s [14, 173]. The worst-case time for this operation is $O(n)$, which amounts to $O(kn)$ for k sources.

A significant further acceleration is possible if the computation of relative neighbourhoods is approximated using Urquhart's [169] algorithm. Instead of testing points in the Delaunay neighbourhood of s against all points in A_s, Urquhart's algorithm tests only these neighbourhood points against each other. Given the limited average degree of a vertex in a Delaunay graph, this test can be performed in constant average time. Despite being an approximation (Figure 4.13), the Urquhart neighbourhood serves as well as the relative neighbourhood in selected computational morphology applications [3]. In our model, using the Urquhart approximation also gives qualitatively similar results to those obtained with the actual relative neighbourhoods, while drastically reducing the simulation time (Section 4.5).

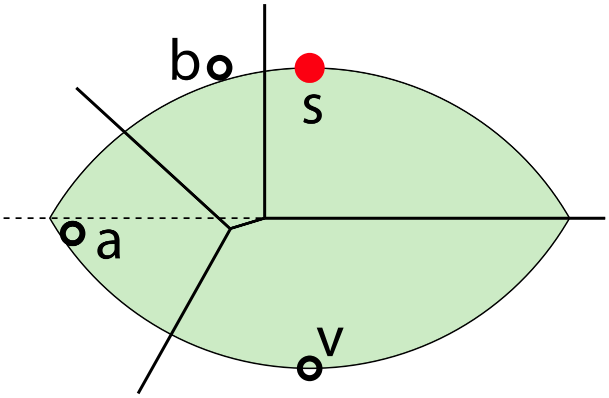


Figure 4.13: The Urquhart neighbourhood is only an approximation: a excludes v from the relative neighbourhood of s, although a is not in the Delaunay neighbourhood of s.

4.5 Results

In order to acquire an intuitive appreciation for the generated patterns, the parameter space of the algorithms was explored by varying one parameter while keeping other parameters constant. Two examples of this exploration are shown in Figure 4.14. Figures 4.14a–e show the impact of the kill distance d_k. As its value decreases, the density of the venation pattern increases, higher-order veins emerge, and eventually anastomose. Figures 4.14f–h illustrate the role of the number of auxin sources that the dart-throwing algorithm attempts to insert per iteration step. As one would expect, higher values of this parameter result in a denser and more regular distribution of the sources; the venation systems generated have smoother primary and secondary veins, and more uniform tertiary veins.

Although more difficult to quantify, the shape and growth of a leaf also have an essential impact on the resulting patterns. For example, Figure 4.14i was generated using similar parameters to Figure 4.14b, except that the leaf grew slowly. As a

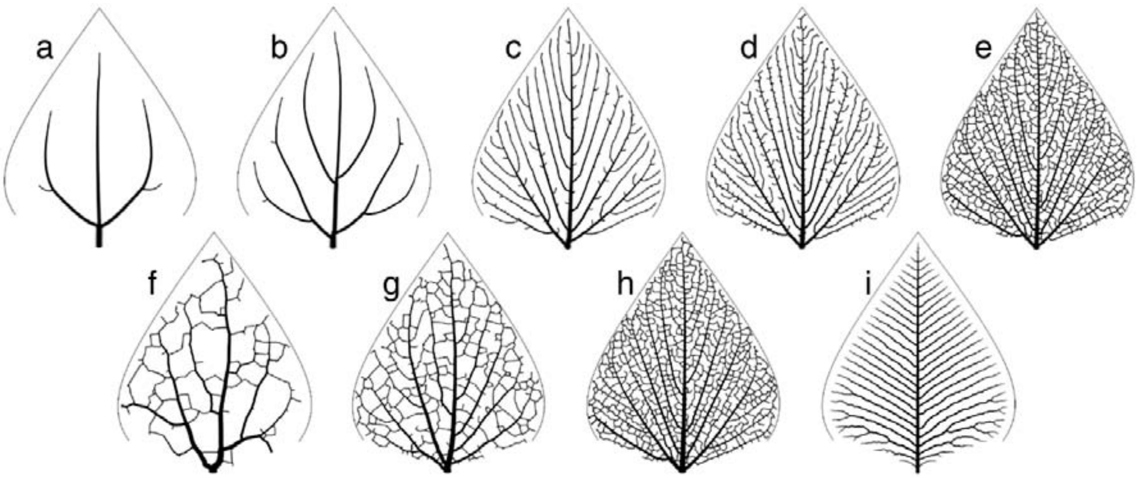


Figure 4.14: Parameter space exploration of venation patterns generated by the algorithms. (a)–(e) The impact of the kill distance on venation patterns. From left to right, the kill distance is 40, 20, 10, 5, and 1. (f)–(h) The impact of the number of sources inserted per step (parameter ρ from Section 4.2.3). From left to right: 0.00006, 0.0003 and 0.006 insertions per unit leaf area per step. (i) A venation pattern generated in a leaf with slow marginal growth.

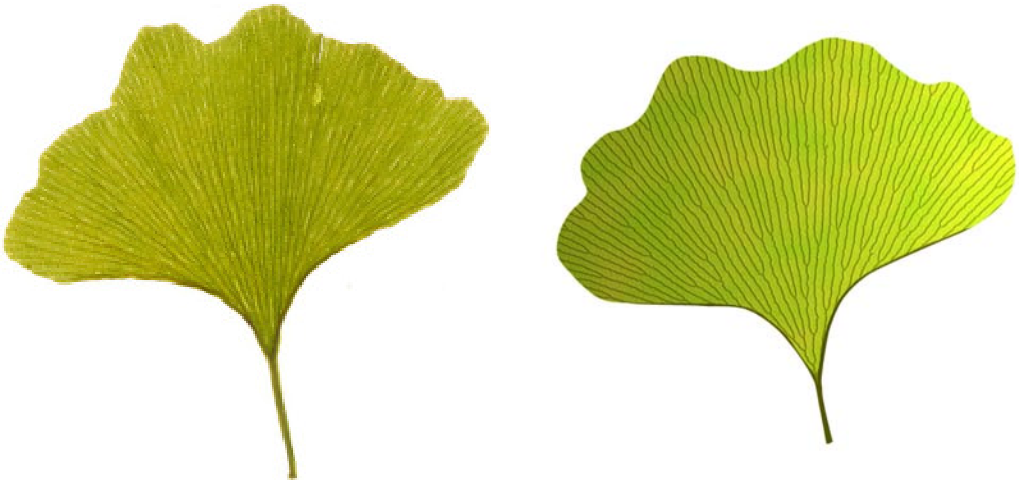


Figure 4.15: A photograph (left) and a rendered model of venation of a *Ginkgo biloba* leaf (right).

result, all vein tips closely followed the leaf margin, with the sources distributed along different sections of the margin affecting individual veins. Similar parameters underlie the model of a *Ginkgo biloba* leaf shown in Figure 4.15. In this case the rounded, tangentially expanding leaf margin prompted bifurcation of veins when the distance between their tips became too large.

The leaves of lady's mantle (*Alchemilla mollis*, Figure 4.16) and sweetgum (*Liquidambar styraciflua*, Figure 4.17) were modeled using relatively faster growth rates, lower values of kill distance (corresponding to Figure 4.14c), and a large number of inserted sources. These are examples of actinodromous patterns, with veins adapting to the rounded or digitate leaf blades. A comparison of generated images to photographs suggests that while the overall patterns of veins are reproduced faithfully there is a discrepancy between the width of veins simulated using the pipe model and those observed in reality.

Ginkgo, lady's mantle, sweetgum, and Nankin cherry (*Prunus tomentosa*) (Fig-

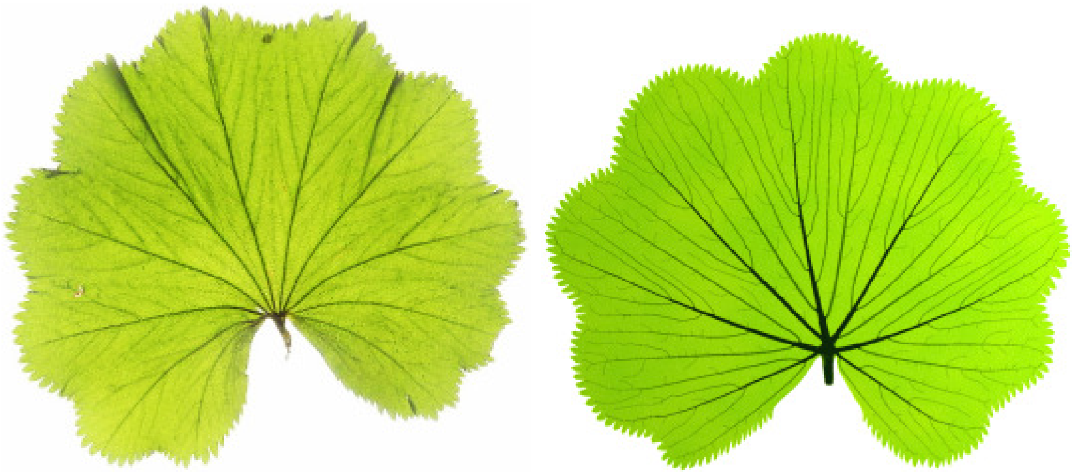


Figure 4.16: A photograph (left) and a rendered model of the venation of a lady's mantle (*Alchemilla mollis*) (right).

Figure 4.17: A model of a sweetgum (*Liguidambar styraciflua*) leaf.

ure 4.18) leaves were modeled using open venation patterns. In contrast, the venation of the toothed leaf shown in Figure 4.19a was generated with anastomoses. The kill distance was low, consistent with Figure 4.14e. An intermediate number of sources was inserted each step, as in Figure 4.14g, in order to produce relatively smooth primary and secondary venation, and irregular reticulate venation. Additionally, veins extend into the lobe at the base of the leaf. A similar set of parameters, but for a different leaf shape, was used to generate the venation pattern in the entire leaf shown in Figure 4.19b. This pattern includes sequences of tertiary veins growing towards the margin in the basal part of the leaf. Such sequences are found in many dicotyledonous leaves.

Figure 4.14h indicated that high density of inserted sources combined with marginal leaf growth produces smooth primary venation and ladder-like tertiary venation. Patterns sharing these features may significantly depend on the leaf shape and the initial conditions of the simulation. This is illustrated in Figure 4.20, in which an orchid tree (*Bauhinia sp.*) leaf with a hierarchically organized system of primary and secondary veins, and a grass leaf with parallel veins, were generated using similar parameter values. In the grass leaf model, a row of initial vein nodes was located at the leaf base, instead of the usual single node. In addition, the birth distance for placing new sources was tuned to match the distance between veins. An application of this pattern as a texture is illustrated in Figure 4.22.

All the venation models presented so far were generated assuming marginal growth of the leaf blade. In contrast, the pattern shown in Figure 4.21 was generated using isogonic growth. This produces a fine reticulate pattern in the tertiary venation, in contrast to the percurrent patterns in Figure 4.20. An even finer reticulation

Figure 4.18: A Nankin cherry (*Prunus tomentosa*) bough.

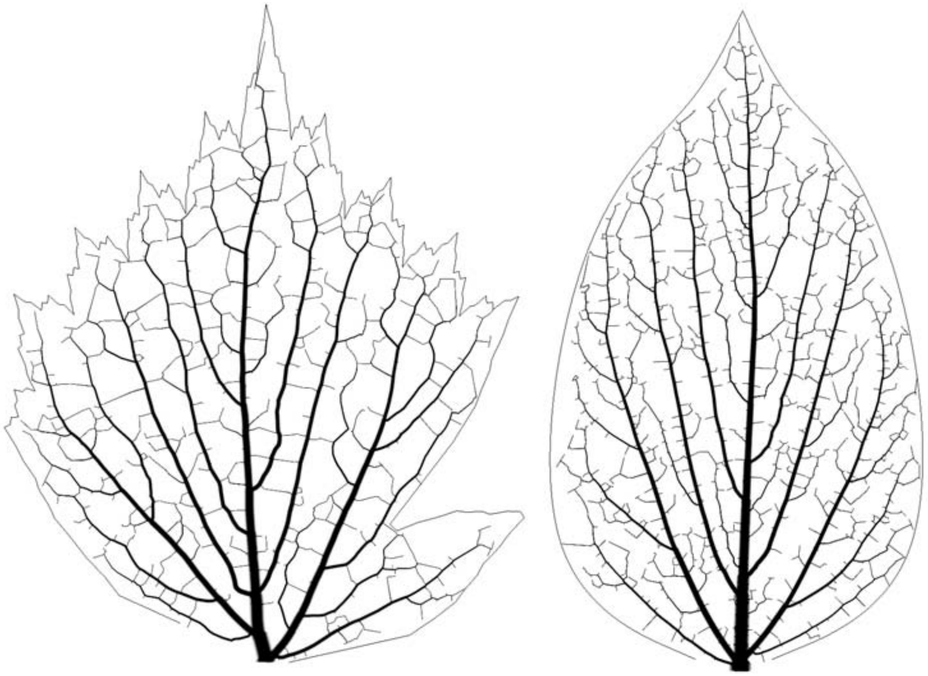


Figure 4.19: Patterns illustrating several emergent properties of the algorithms. Left: A toothed leaf with a lobe. Right: Emergence of sequences of tertiary veins off the bottom secondary veins.

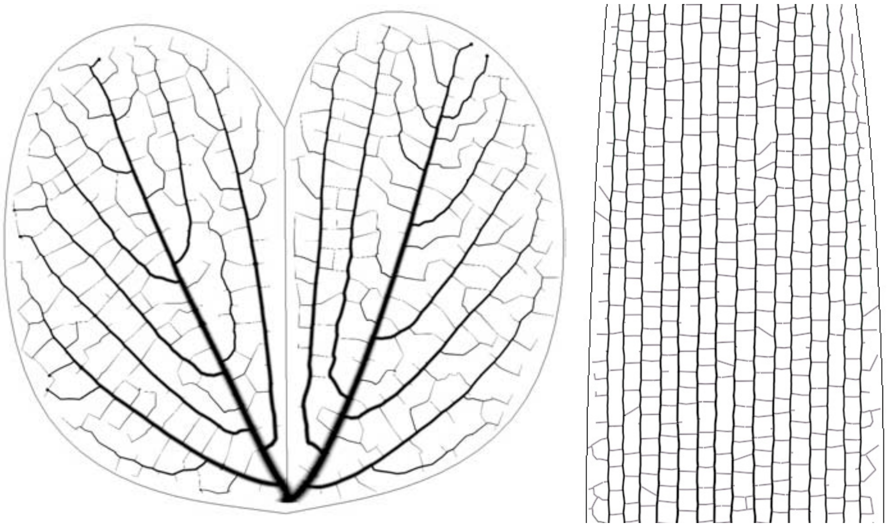


Figure 4.20: An orchid tree (*Bauhinia sp.*) leaf and a segment of a grass leaf. The difference in venation patterns is due primarily to different leaf shapes.

is present in the poplar (*Populus sp.*) leaf models shown in Figure 4.23. In addition to providing another example of patterns generated with isogonic leaf growth, these models illustrate the difference between patterns generated using correctly calculated relative neighbourhoods and their Urquhart approximation. Although the layout of secondary veins differs, both patterns are similar in their general character.

In order to test the potential of the generated patterns in realistic image synthesis, we have included some of them as textures in plant models. The results, obtained using the renderer Dali [69], are shown in Figures 4.18, 4.22, 4.24, and 4.25. Additional detail on the process of importing and rendering the generated textures into a rendering system such as Dali are presented in the thesis of Martin Fuhrer ([40], Chapters 6-7, Appendix A), who was instrumental in creating these images. Finally, the proposed method is quite controllable, as can be seen from the vascular patterns produced for the unrealistic leaf blades in Figure 4.26.

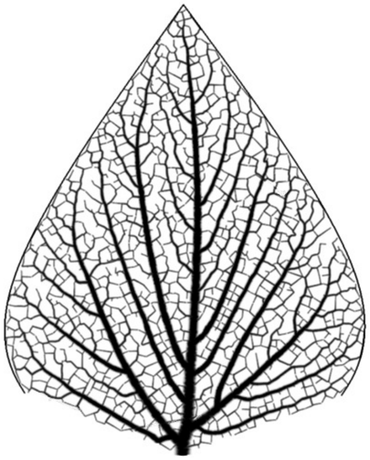


Figure 4.21: A reticulate venation pattern emerging in an isogonically growing leaf.

Parameters used to generate the presented leaf patterns are collected in Table 4.1.

4.6 Conclusions

An algorithm for synthesizing leaf venation patterns has been proposed using the SCA. The algorithm is based on the biologically plausible hypothesis that venation results from an interplay between leaf growth, placement of auxin sources, and the development of veins (which has been subsequently explored and corroborated by Fujita and Mochizuki [41]). Our results suggest that the apparent complexity of leaf venation may emerge from the iteration of a simple elemental mechanism. An effective implementation of this algorithm represents an unexpected application of computational geometry to a biological problem.

By representing the process at a geometric level vascular patterns can be reproduced without relying on molecular data that are still incomplete. Furthermore, the proposed model is robust under a number of interpretations. For example, although

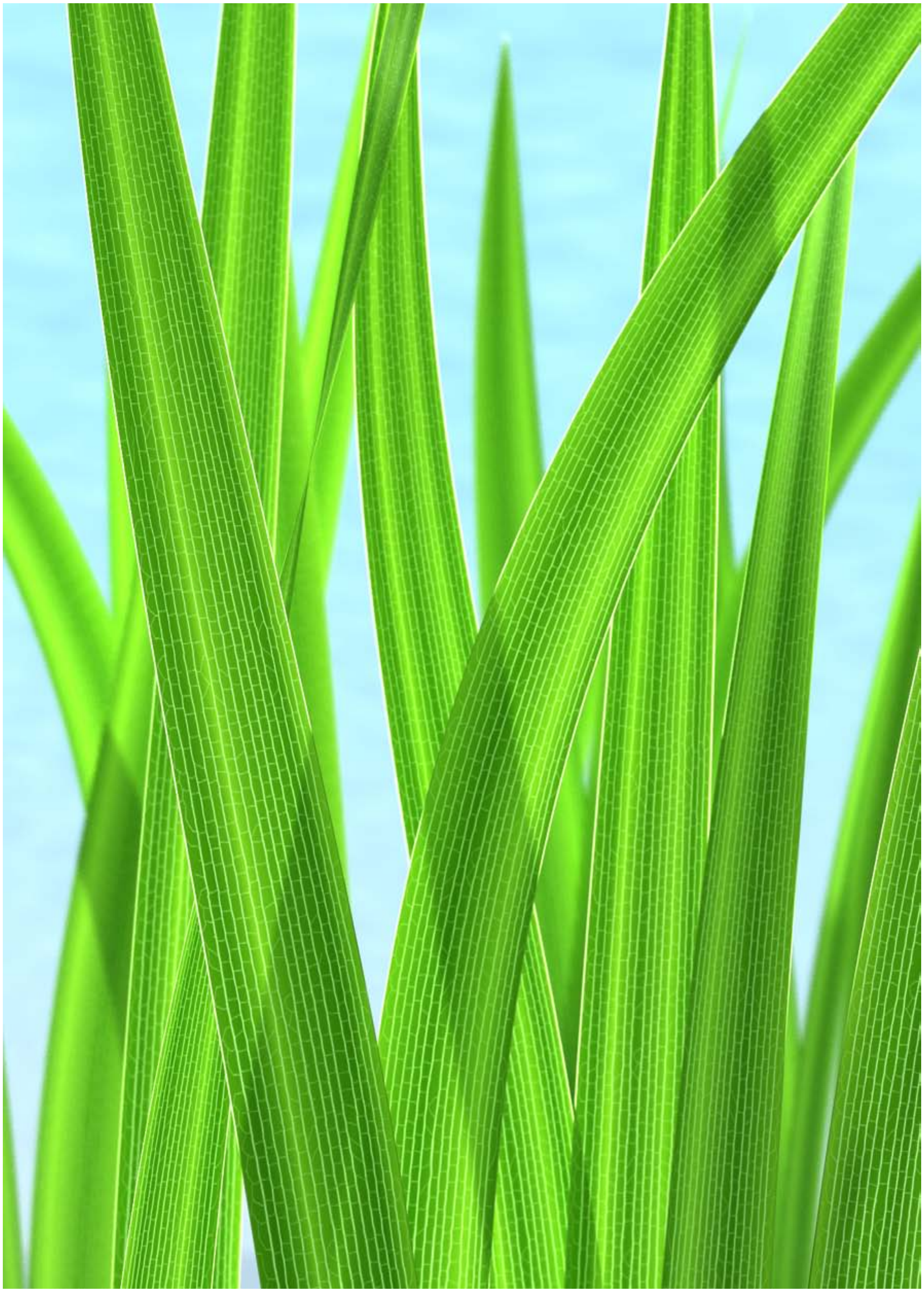


Figure 4.22: Grass leaves with venation patterns from Figure 4.20.

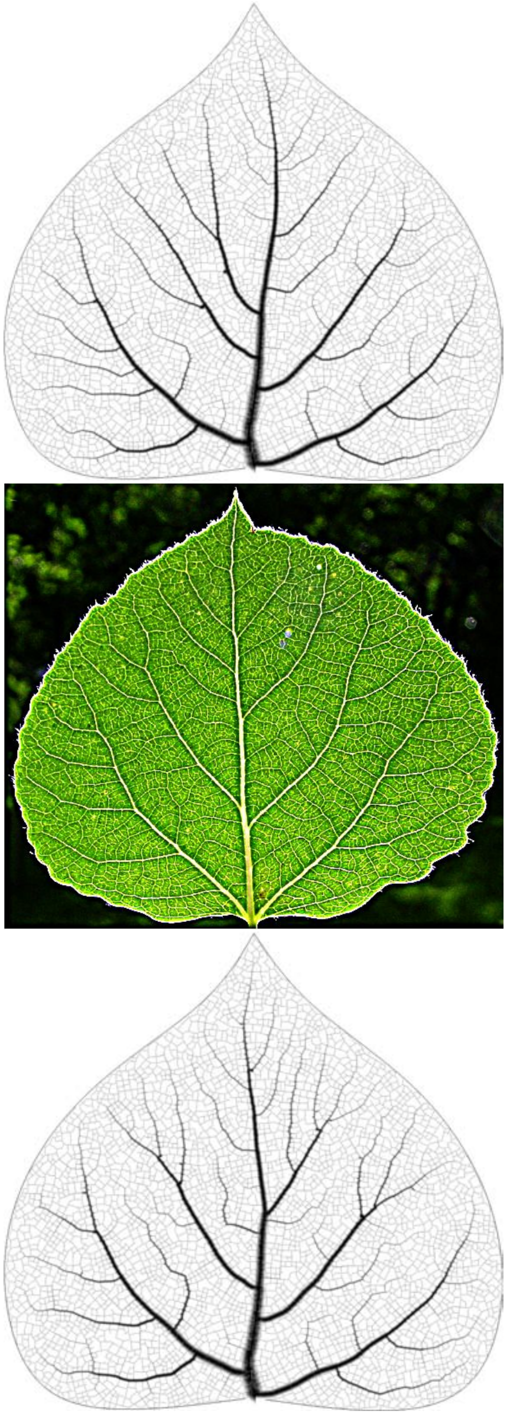


Figure 4.23: Two models of reticulate patterns of poplar leaves, compared to a photograph of a real leaf. The upper pattern was generated using relative neighbourhoods in two hours; the lower pattern was generated using the Urquhart approximation in two minutes.

Figure 4.24: A simulation model of a poplar branch (branching architecture has been modeled using l-systems). The leaves have textures generated using the same parameters as those in Figure 4.23. The outlined area is shown magnified in the inset.

Figure 4.25: A model of *Trillium sp.* flower. Venation patterns are not limited to leaves, but also include flower petals.

Figure 4.26: Letters with procedurally generated vein patterns.

there is some evidence that auxin sources may be discrete [2], this is not known with certainty. Yet, whether the auxin sources in the model represent point sources, auxin maximas, or even the discretization of a continuous concentration field, the geometric model remains valid.

In addition to offering biological insights, the resulting patterns are applicable to texture synthesis. In the examples, we employed classical methods of texture and bump mapping, but the availability of a leaf venation model opens the door for more advanced techniques, such as the incorporation of veins into a detailed geometric model of the leaf [12], detailed simulation of light propagation in a leaf [54, 6, 31], and the use of fast-rendering methods specialized for leaves [172].

Name	Figure	Algorithm	d_k	b_s	L_i	Growth type	L_f	ΔL	$\rho \times 10^6$
Param. search d_k	4.14a–e	closed	—	10	270	marginal	1600	8	600
Param. search ρ	4.14f–h	closed	1	10	270	marginal	1600	8	—
Slow marginal growth	4.14i	open	40	1	4500	marginal	112,000	34	300
Ginkgo	4.15	open	100	1	10	marginal	50,000	15	300
Alchemilla	4.16	open	10	1	3200	marginal	4500	9	200
Sweetgum	4.17	open	1	1	1200	marginal	6700	7	200
Toothed margin	4.19 left	closed	1	10	6600	marginal	26,000	40	10
Emergent tertiary	4.19 right	closed	1	1	3500	marginal	24,000	60	20
Orchid	4.20	closed	1	10	6500	no growth			600
Reticulate pattern	4.21	closed	1	15	225	isogonic	4500	40.5	200
Nankin cherry	4.18	open	30	1	1100	marginal	5500	11	40
Grass	4.22	closed	1	30	13,000	no growth			100
Poplar	4.23, 4.24	closed	1	2	115	isogonic	3500	9	200
Trillium	4.25	closed	1	30	4500	no growth			100

Table 4.1: Values of parameters used to generate venation patterns presented in this chapter. In all cases, the length of a vein segment is equal to 1. d_k: kill distance; $b_s = b_v$: birth distance; L_i: initial leaf length; L_f: final leaf length; ΔL: length increase per step; ρ, number of darts per unit area per step. Parameter n used to calculate vein width, (Section 4.2.5) is 2 for all models.

Chapter 5

Modeling leaf venation patterns II: Accounting for areoles and non-uniform growth

Venation patterns have a hierarchical appearance, manifested in the different orders of veins that can be identified visually. This hierarchy is reflected in vein development, as major loops and higher order veins form by a reiterative process driven by leaf growth [150, 149]. Many developmental events appear quite predictably, and some data concerning the growth of the leaf during this process are available [150, 30].

These observations coincide with the hypothetical causal relations underlying the model outlined in the previous chapter (Figure 4.8). Unfortunately, although growth and development were considered, it was difficult to match precisely the sequence in which veins and sources emerge using these techniques. Additionally, the blindly terminating veins seen in otherwise closed patterns (areoles) were not reproduced (Figure 4.7).

In this chapter we improve on the results from Chapter 4 by simulating leaf development and embedding vein nodes and sources in a growing leaf blade with a flexibly defined growth pattern. Two techniques for specifying anisotropic non-uniform growth are introduced, allowing for a wider range of growth patterns to be described, compared to those described in the previous chapter. New veins and auxin sources are introduced as space becomes available using an adaptation of the algorithm for closed vein pattern formation (introduced in Section 4.3). The shapes of the veins adapt dynamically to the growing leaf blade, creating space for additional veins to form. These modifications simulate the reiterative process of vein pattern formation, as proposed by Scarpella et al. [150]. Blindly terminating veins are introduced by simulating an event which interrupts the process of forming closed loops, which may correspond to the differentiation of *mesophyll* (non-vascular tissues) in actual leaves [149].

As a case study for the model proposed in this chapter, the sequence of developmental events observed in *Arabidopsis thaliana* leaves is recreated. Specifically, the placement of auxin sources on the leaf margin and the emergence of loops and higher-order veins are reproduced.

5.1 Leaf growth and vascular development

The protein PIN1 is currently thought to provide the earliest indication of vascular patterning in the leaf [150]. This protein is localized non-uniformly on cell membranes, thus providing a polarity for each cell. During vascular patterning the files of cells that will give rise to veins take on a consistent polarity. These polarized files of cells are thought to correspond to the *canals* of Sachs' canalization hypothesis [148].

The localization of PIN1 proteins in developing *Arabidopsis* leaves was observed in great detail by Scarpella et al. [150]. These observations provide a sequence of developmental events that we used as a reference for simulations. The main results of Scarpella et al. are described below, focusing on the first leaf in the rosette of *Arabidopsis thaliana* (diagrams and details supporting the text are given in Figure 5.1).

Each leaf is initiated at a point of high auxin concentration on the apex of the plant (a). The PIN1 proteins in surrounding tissues polarize towards this point. Given that PIN1 proteins act as efflux carriers, the auxin peak may be created by collecting auxin from surrounding tissues, and is thus called a *convergence point*. As the leaf grows, the tissue surrounding the convergence point bulges outward, growing away from the meristem. Auxin from the convergence point at the tip of the leaf flows into subepidermal layers of the meristem, leading to the formation of the midvein (b).

As growth continues, a second set of convergence points gradually emerge on the margin of the leaf, preceding the formation of the lower half of the first loops in the leaf (c). The upper half of the loop is patterned soon afterwards (d). Additional convergence points appear sequentially near the base of the leaf as it grows, preceding the formation of additional loops (e,f). Each of these additional loops connects to the midvein as well as to the older, more distal loop. The first leaf usually has three loops, but later leaves may have several additional loops.

After a loop has formed, higher-order veins develop inside the region enclosed by that loop. These veins appear and are elaborated upon as the loop expands, dividing the area bounded by the loop into smaller domains. In some leaves, veins of increasing orders may emerge, resulting in further subdivisions of the domains. Following some event, which appears to be correlated with the cessation of growth and mesophyll differentiation, developing veins stop subdividing the regions containing them, and instead create open structures with the highest-order veins terminating blindly [149].

Using the vein loops and convergence points as landmarks, growth during development can be inferred. The developmental progression shown in Figure 5.1 suggests

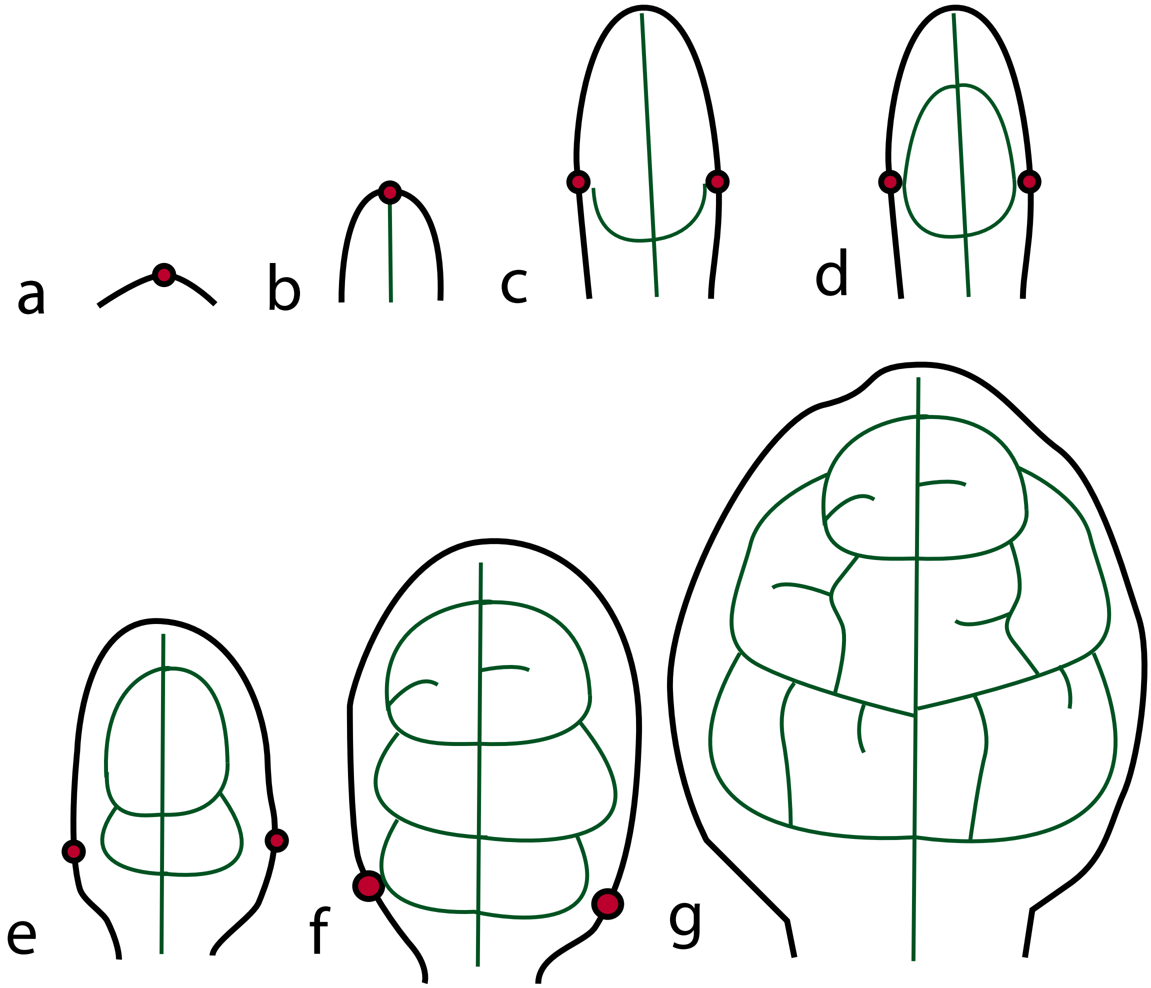


Figure 5.1: The sequence of developmental events observed in *Arabidopsis thaliana* based on [150]. a)Convergence point at tip of incipient leaf appears (shown in red). b)The midvein forms (shown in green). c)New convergence points appear on the leaf margin approximately half-way between the tip and base of the leaf. Veins form connected to the mid-vein and terminating at the margin (lower loop domains). d)A second set of veins connected to the midvein appear (upper loop domains). e)As the leaf grows, new convergence points appear on the margin, shortly afterwards new loops form connected to the first loops and the midvein. f) A third set of convergence points appear on the margin, preceding the formation of the third set of loops. The first loops have expanded and tertiary veins have formed. g) The second and third loops have expanded and tertiary veins have formed (placement of tertiaries is based in part on the model interpretation proposed by Scarpella et al. (Figure 7 [150]))

that initially growth is fairly uniform, but anisotropic with much greater longitudinal expansion (a-d). Subsequently, growth appears to focus at the base of the leaf, where the small loops formed in (f) and (g) later become larger than the older loops above them. These observations are supported by the pattern of cell divisions observed in *Arabidopsis* by Donnelly et al. [30]. Cell divisions are correlated with growth, thus indicating where growth occurs in a tissue. Donnelly et al. observed that a gene associated with cell divisions was initially expressed throughout the leaf. As the leaf grew, a longitudinal gradient emerged with greatest expression at the base of the leaf and in the petiole.

5.2 Generation of the developmental sequence of venation patterns

The hypothetical causal relations between growth, vein development, and source placement assumed in the previous chapter also underlie the extended algorithm described in this chapter. Although similar in spirit, several details of the closed model have been modified to approximate better the growth and development of patterns over time.

At the beginning of a simulation, no sources are present and new sources are added deterministically in the largest available space in the leaf blade (Section 5.2.2). Veins extend toward sources based on the relative neighbourhood, but this process is modified to reflect the hypothetical differentiation of veins (Section 5.2.3).

Leaf blades may exhibit non-uniform anisotropic growth, but the recreation of such growth patterns was not addressed in the previous chapter and is considered in

Section 5.2.1. Vein nodes and sources are now embedded in a growing leaf blade, and their positions are modified by growth. Consequently, growth plays a more significant role in pattern generation. In this context, maintaining a plausible geometry for vein patterns as they deform becomes a problem and is addressed in Section 5.2.6. Additionally, width is no longer calculated using the pipe model. Instead, the width of each vein node v increases in proportion to the elongation of the vein segments containing v.

5.2.1 Modeling growth

The dynamics of early leaf growth, when the first veins are determined, is not yet well known and is an object of active studies (see [20, 138] for related work on petals, and [82] for related work on the shoot apical meristem). Qualitatively leaf-growth is non-uniform and anisotropic, and changes as the leaf develops [149, 150, 30]. Two general approaches have been used previously to model leaf growth, and specific implementations of both techniques are considered in this thesis.

The first approach is to specify a growth field, either explicitly or as a result of a physically-based expansion model [138, 171]. These techniques specify expansion locally, at each point in the leaf blade. The global form of the leaf emerges by integrating the local expansion over time. The first implementation described in this section focuses on a simple special case of this technique called the *local-to-global growth model*.

The second approach is to specify the progression of leaf shape over time. Marginal growth was first simulated in this way by Scholten and Lindenmayer [151]; and diffuse growth has been described in terms of warping and morphing of graphical ob-

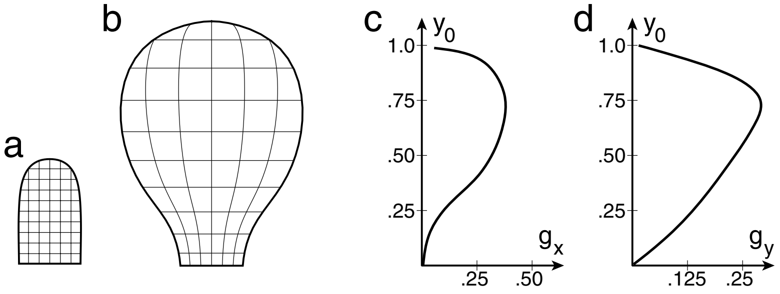


Figure 5.2: Modeling a growing leaf using local-to-global growth. Leaf growth from shape (a) to shape (b) is specified by two functions that control growth rates in the horizontal (c) and vertical (d) direction according to the formulas $RERG_x = 1 + g_x$ and $RERG_y = 1 + g_y$, respectively. The arguments to both functions are the initial vertical positions of the points within the growing leaf blade. The functions have been specified using a graphical editor.

jects [125, 47]. The second implementation described in this section simulates diffuse growth using key-frame interpolation between parameterized surfaces [156, 7].

Local-to-global growth model

Growth can be characterized by the *growth tensor* field [56], which specifies the magnitude of the expansion of infinitesimal surface regions in various directions, and may include a possible rotation of each region. The growth tensor is a generalization of the *relative elementary rate of growth* (RERG), which is defined as the rate at which an infinitesimal distance Δs, measured in the direction of a line l at a point p of a growing object, increases over time. This rate is normalized with respect to the distance Δs, yielding $\text{RERG}_l(p) = (1/\Delta s)(d\Delta s/dt)$ [56].

A limited version of this model was implemented, in which relative elementary rates of growth in the horizontal and vertical direction are specified as user-defined functions of the initial position of a point along the y axis (Figure 5.2).

As described in Section 4.2.3, given the position of a material leaf point p at time $t_1 \geq t_0$, we need to find the position of p at time $t_2 > t_1$. We solve this problem under the simplifying assumption that the relative elementary rates of growth in the x and y directions depend only on y_0, the y coordinate of p at the initial time t_0. By solving the equation $dx/dt = x \ \mathrm{RERG}_x(y_0)$, we find that a point with horizontal coordinate x_1 at time t_1 will be displaced to $x_2 = x_1[\mathrm{RERG}_x(y_0)]^{(t_2-t_1)}$ at time t_2.

In the vertical direction the RERG value is no longer constant; thus, we calculate the vertical coordinate y_1 at t_1, given the coordinate y_0 at t_0, as the integral:

$$y_1 = \int_0^{y_0} [RERG_y(y)]^{t_1-t_0} dy. \tag{5.1}$$

In order to answer the general question of calculating the y displacement between times t_1 and t_2, we first solve Equation 5.1 for y_0, given y_1. For a material point with the known position y_1 at time t_1 this solution describes the position y_0 that this point would have if it already existed at time t_0. Knowing this, we use Equation 5.1 again to find y_2, the y coordinate of the same material point at time t_2. These calculations are performed efficiently using a lookup table with precomputed values of integral (5.1).

Key-frame interpolation of B-spline surfaces

To allow for additional flexibility in the specification of growth, leaf development was also addressed using a technique that describes leaf development using a series of surfaces, or key-frames, corresponding to different stages of growth. The key-frames are interpolated over time to produce a growing surface. Each vein node and source is embedded in the growing surface when it is introduced, and deformed with the surface over time.

The *growth progression* of a developing leaf is defined by a set of n parameterized surfaces with associated time values:

$$S = \{(Q_1, t_1), (Q_2, t_2), \cdots, (Q_n, t_n)\},$$

where $Q_i(u, v) : [0, 1] \times [0, 1] \to \Re^2$ takes the parameters (u, v) to a position in the plane, and $t_i < t_{i+1}$. At a given time t the corresponding leaf surface is Q_t where:

$$Q_t(u, v) = \frac{t_{i+1} - t}{t_{i+1} - t_i} Q_i(u, v) + \left(1 - \frac{t_{i+1} - t}{t_{i+1} - t_i}\right) Q_{i+1}(u, v)$$

with i such that $t_i \leq t \leq t_{i+1}$. Thus, Q_t linearly interpolates between the key-frames placed at t_i and t_{i+1}. B-spline surfaces created using an interactive editor were used for this purpose.

Each vein node and attraction point embedded in the growing surface has an associated (u, v) value determined when it is generated. The 2D position of each embedded node is determined at time t by evaluating $Q_t(u, v)$. This moves the point with the leaf as is shown in Figure 5.3. All calculations, such as the placement of additional vein nodes and auxin sources, use the 2D position of a point. This poses a problem as, once created, a new point must be embedded in the growing leaf. Thus, using the 2D position of the point the (u, v) parameter of its embedding in Q_t must be computed. Given a parametrized surface $Q(u, v)$ and a point P, the problem of finding the pre-image of P is called the *point inversion problem.* We solve this problem by numerically minimizing the value of

$$E_P(u, v) = \|Q(u, v) - P\|. \tag{5.2}$$

When $E_P(u, v) = 0$ then (u, v) is the pre-image of P. In our implementation, this equation is minimized with respect to (u, v) using a steepest descent search; which

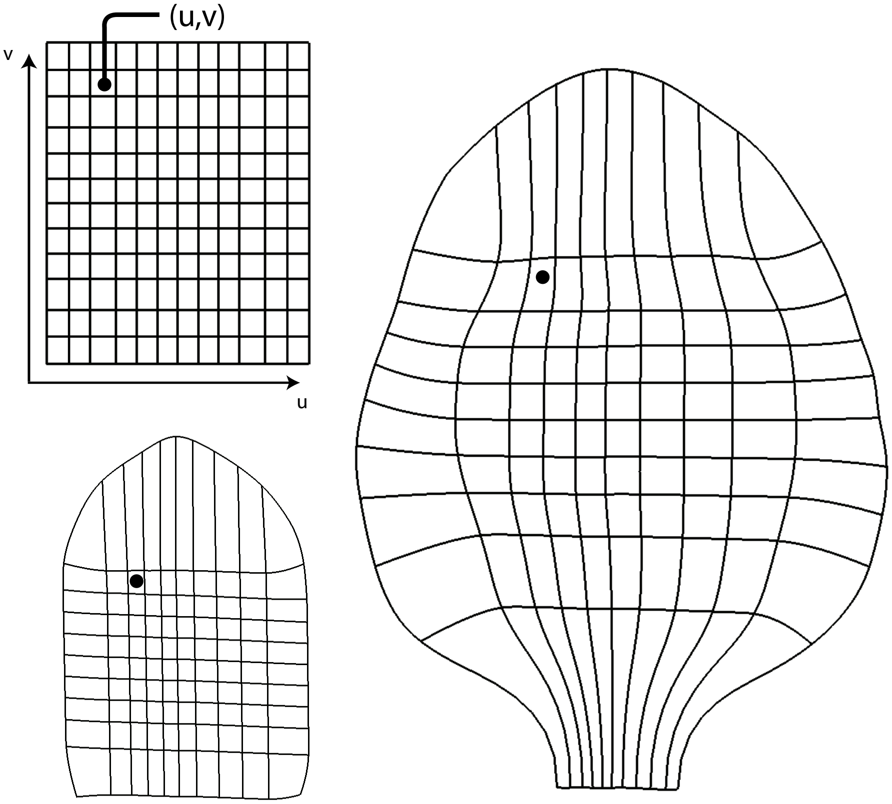


Figure 5.3: Modeling a growing leaf using B-spline surfaces.*(Upper-Left)* The u,v parameter space with a single (u,v) point marked *(Bottom-Left)* The initial B-spline surface with the complementary (u,v) point marked. *(Right)* The surface after a period of growth.

requires a set of parameter values, an initial guess, from which to begin optimizing Eq.5.2. The parameter values of the closest point already embedded in the surface provide this estimate, which can be obtained using the Delaunay triangulation of vein nodes and sources. Although more sophisticated methods exist [91, 64], this approach suffices due to the availability of a good initial value.

5.2.2 Source placement

To match the emergence of convergence points on the margin, sources are placed deterministically. As before, it is assumed that a source will emerge in any space large enough to support it, as determined by the birth distance b_s. Sources are placed at the beginning of each iteration by repeatedly placing a source at the centre of the largest empty circle in the leaf blade (while the circle is large enough to support a new source).

This operation is supported by a Delaunay triangulation containing sources and vein nodes, and relies on two properties of the triangulation. Specifically, the circumscribed circle of any triangle in a Delaunay triangulation of a point set will not contain any other point in this point set [114]. Moreover, the largest empty circle in a bounded region of the plane is either a circumscribed circle or a point on the boundary [164]. Consequently, the set of circumscribed circles of the Delaunay triangulation, along with a set of sample points on the margin of the leaf, are used to determine the largest empty space in the leaf blade.

Although the centre of the largest empty circle on the leaf margin could be calculated exactly, it is simpler to approximate the position of the centre by considering a set of samples on the leaf margin as candidate sites for the circle's centre. This set is created when the simulation is initialized and updated when the distance between samples grows too large. A circle is generated, centred on each sample point, with radius equal to the distance to the closest point in the Delaunay triangulation. The source is then placed at the centre of the largest circle, corresponding to either a point on the margin or a circumscribed circle. After the source is placed, the Delaunay triangulation is updated.

5.2.3 Generation of the structure of areoles

Areoles are the smallest closed regions in the vascular patterns of leaves. In some species of dicotyledons the veins within the areoles terminate blindly (fig.4.7). This feature was not accounted for by the models proposed in the preceding chapter. Using the *areole algorithm*, open structures in the areoles are introduced into otherwise closed patterns by modifying the closed algorithm presented in Chapter 4.

A hypothetical progression of vascular differentiation underlies the areole algorithm. This hypothesis is outlined here and elaborated upon in the following section. When a source is placed, the location of polarized files of cells, pre-patterning veins, is determined using the relative neighbourhood. The differentiation of the pre-pattern into vasculature is inferred from the following observations. Second and higher order veins are initiated at existing veins and extend progressively (a vein extends at the tip) [75]. Additionally, veins differentiate continuously, that is to say that during development the vascular network in the leaf connects all veins [75, 149]. As open structures in the areoles arise progressively and must be connected to pre-existing vasculature, a single vein must give rise to the tree contained in the areole. On this basis we infer that files of cells polarized away from a source of auxin do not produce veins simultaneously; instead, only one vein is formed and bifurcates along the remaining files of cells upon reaching the source. In contrast to the closed algorithm, the model of areoles is consistent with this assumption. The details of this process are demonstrated in the example provided in Section 5.2.5.

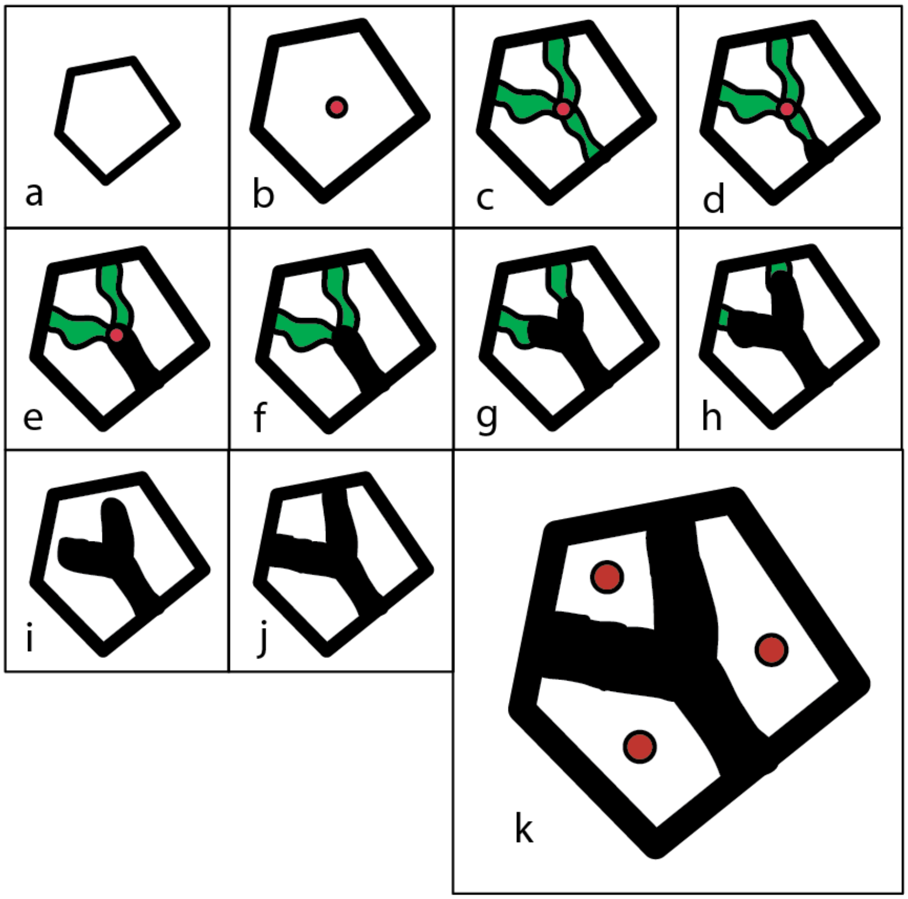


Figure 5.4: Hypothetical sequence of vein development in a growing areole. Differentiated vasculature is shown in black, pre-patterned vasculature is shown in green, and auxin sources are shown in red. See the text for a detailed explanation

5.2.4 Hypothetical progression of differentiation

To explain the complexity of mature patterns we propose a hypothetical progression of differentiation for the structures observed in the areoles of leaves. We assume that mature patterns arise from the repeated patterning and differentiation of veins according to this hypothesis, where the pattern of veins is elaborated upon as space becomes available due to growth.

The hypothetical sequence of events giving rise to the structure in areoles is illustrated in Figure 5.4 and described as follows, beginning with an empty areole (a). As the leaf grows, the area enclosed by the areole likewise increases, eventually providing enough space for an auxin source to form (b). Canals connecting the new source to existing vasculature then emerge (c). In reality, these canals consist of polarized files of cells, with high PIN1 expression. In accordance with the canalization hypothesis, these canals transport auxin and are assumed to become further specialized in auxin transport, providing the pre-patterning for vascular differentiation. It is assumed that only one strand begins to differentiate at this time, as other scenarios tend to produce unrealistic open structures. The selection of a single strand could be the result of competition between the pre-patterned strands for auxin, based on the dynamics proposed by Sachs [144] and modeled by Mitchison [105]. Differentiation progresses from the existing vasculature (d) into the areole. This is consistent with the progressive differentiation of higher order veins observed in *Arabidopsis* leaves [149, 75]. Here we assume that the vein extends towards the source (e). As the vein approaches the source, it is assumed to drain the source of auxin. The source then differentiates into vasculature and ceases to produce auxin (f). The polarized canals induced by the now absent source are already committed to a vascular fate and persist in the absence of auxin. Now, differentiation continues away from the source along the remaining polarized canals (g). This process continues until a single cell separates the developing veins from existing vasculature (h).

We suppose that the event leading to open structures in the areoles interrupts the differentiation of the final cell in the developing vein, thus precluding the closure of loops. The development of additional veins, however, is still possible, allowing higher-order branching structures to form. This interpretation is based on our visual examination of the structures in the areoles of dicotyledon leaves, which may contain open structures of several orders, with veins terminating very close to existing vasculature.

In accordance with the preceding discussion, one of two scenarios plays out. If the event leading to blindly terminating veins has occurred then the cell separating the developing vein from existing vasculature does not differentiate, producing an open structure (i). Otherwise, this cell differentiates (j), dividing the areole into several smaller areoles. Additional growth leads to the emergence of sources in these smaller areoles, initiating the development of additional vasculature (k).

5.2.5 Example

The execution of the areole algorithm is illustrated in Figure 5.5, which begins with the formation of a closed areole (shown in green), which does not contain any vein nodes or auxin sources (a). An auxin source (red disk) is placed at the centre of the largest empty circle, corresponding to the centre of the areole (b). The vein nodes in its relative neighbourhood are computed (c) and stored. As observed in the previous chapter, the relative neighbourhood provides a geometric characterization of the structure of highest-order venation (Fig.4.12). Here it is used to determine the location of the polarized files of cells that pre-pattern vasculature (red lines). Geometrically, the closest relative neighbour is assumed to win the competition to differentiate. Consequently, the closest relative neighbour is selected by the source and a vein node is placed in the direction of the source (d), initiating a new vein. This ends the current iteration of the algorithm.

During subsequent iterations, the new vein extends towards the source until a

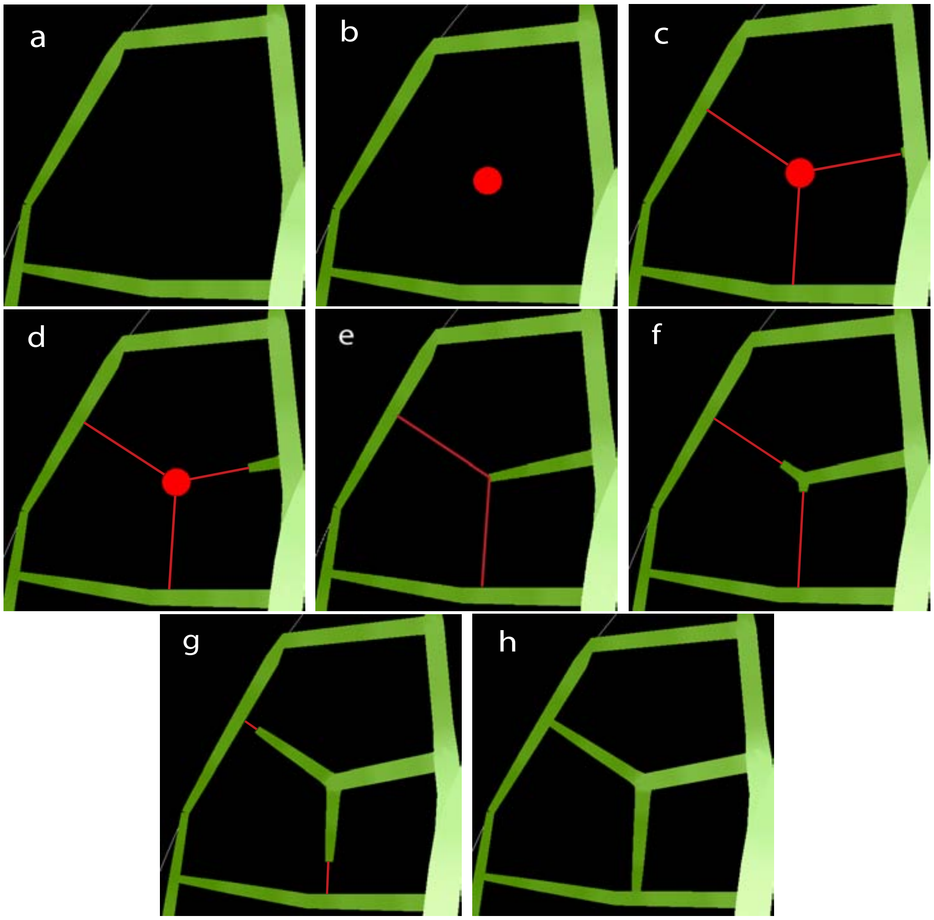


Figure 5.5: Images from the modified model of vascular patterning in the leaf.

vein node is placed within the kill distance of the source. The source is then replaced by a vein node (e). Following the removal of the source the vein node at the source bifurcates and extends towards the remaining relative neighbours of the source, stored when the source was added to the simulation (f). The resulting veins continue to extend towards the relative neighbours until the distance separating an extending vein from the relative neighbour is less than D, the basic unit of the model (g).

One of two scenarios now plays out. If the event leading to blindly terminating veins has occurred, then the extending vein is not connected to the relative neighbour. Otherwise, the extending vein is connected to the relative neighbour by an edge in the vein graph. The second scenario causes the areole to divide into several smaller areoles (h). The time when this event occurs is a parameter of the model, and effects all subsequent vein formation.

When the veins forming in the areoles are always allowed to reconnect, the resulting final patterns are identical to those produced by the closed model of Chapter 4. The key difference between the closed and areole algorithm is the order in which vein nodes are added to the pattern. With the areole algorithm, addition of a source results in a single vein extending towards the source, which then bifurcates at the source and reconnects.

5.2.6 Refining vein nodes

When patterns are grown without vein subdivision unnaturally angular loops are produced. Compared to this, the veins in real leaves appear to form smooth curves. This smooth appearance most likely results from elongation and division of vascular

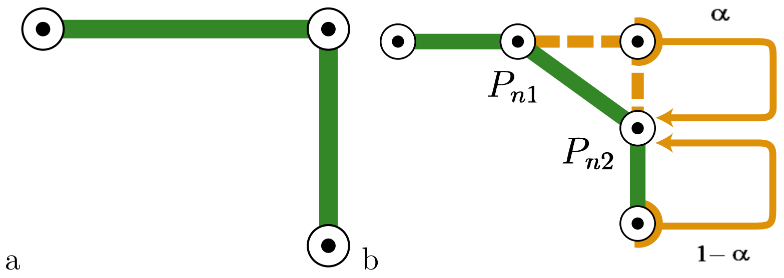


Figure 5.6: Adaptive vein refinement. a)Three consecutive vein nodes. b)Two new nodes are introduced, smoothing the curve and replacing the corner point. The labelled yellow arrows indicate the contribution of the original points to the new point P_{n2}.

cells, which tend to elongate in the direction of the vein and divide perpendicular to this direction [34]. If each cell wall along the vein represents a joint where curvature is concentrated, then each division may serve to diffuse the curvature locally, smoothing the path of the vein.

We emulate this process geometrically by subdividing the segments between vein nodes when their length exceeds a threshold. A generalized version of Chaikin's corner cutting method is employed to this end [16]. Classically, Chaikin's method is applied to all points in parallel, effectively doubling the number of points. As growth can be quite non-uniform, this would quickly increase the number of vein-nodes to an unmanageable number. Hence, here this method is applied adaptively at a single point.

To describe this more formally, let P_1 and P_3 be the end points of an open polygon and P_2 be the interior vertex (Figure 5.6a). The line segments connecting the three points are subdivided when the sum of $\|P_1 - P_2\|$ and $\|P_2 - P_3\|$ exceeds a user-defined threshold, with P_2 being replaced with two new points, P_{n1} and P_{n2} (Fig. 5.6b). The positions of these points are determined by cutting the corner created by the three original points according to

$$P_{n1} = (1-\alpha)P_1 + \alpha P_2 \tag{5.3}$$

and

$$P_{n2} = (1-\alpha)P_3 + \alpha P_2. \tag{5.4}$$

The constant α is a user-defined parameter chosen heuristically to produce better visual results. In the experiments reported here we use values between 0.5 and 0.75. The method proposed by Chaikin used $\alpha = 0.75$ to obtain a smooth curve, but in the context of leaf vein modeling where subdivision is applied adaptively, other values can yield more realistic vein patterns.

The visual plausibility of patterns can also be improved by jittering positions of P_{n1} and P_{n2} to introduce high frequency noise (which in reality may be due to the cellular nature of the leaf).

5.3 Case study: The first leaf of *Arabidopsis thaliana*

The sequence of developmental events, and the final vein pattern of the first leaf of *Arabidopsis* were simulated using the two growth techniques described in Section 5.2.1. It is important to note that, except for a single initial vein node, all vein nodes and sources are placed algorithmically.

5.3.1 Development of patterns using the local-to-global growth model

When vasculature is patterned, leaf growth is non-uniform and anisotropic, with leaf shape changing from narrow to almost circular. A corresponding model of leaf blade growth using local-to-global growth is shown in Figure 5.2, and the resulting simulation of vein pattern development is presented in Figure 5.7. This progression is described in the following text.

Initially, the leaf expands vertically from a small protrusion (a), with a single vein node placed at the base of the leaf. As the leaf tip moves away from the initial vein node a source emerges at the tip (b), initiating midvein formation (c). The leaf continues to elongate, but now the distal half of the leaf also expands laterally, leading to the placement of two sources on the margin (d), which generate the lower half of the first loops (e). Additional growth leads to the simultaneous formation of the upper half of the first loops, and a second pair of loops (f). The final image in the progression shows the subsequent formation of a third loop near the base of the leaf, and tertiary veins in the enlarged first and second loops (g). As growth continues, the pattern is elaborated upon, creating veins of increasing order and eventually blindly terminating veins in the areoles (Fig. 5.8).

The developmental progression generated using local-to-global growth recreates most of the events observed by Scarpella et al. [150]. Unfortunately, there are several problems with this progression which could not be resolved using this simple growth model. Firstly, the upper half of the first pair of loops and the second pair of loops form simultaneously. Additionally, when the second pair of loops forms they enclose a larger area than the first loops, which should happen much later. Finally, the third pair of loops are much smaller than the second loops when formed, as is observed in *Arabidopsis*, but should enclose an area approximately the same size in the final pattern (as in the wild-type patterns in Figure 2(A) of [160] and Figure 1(A) of [19]).

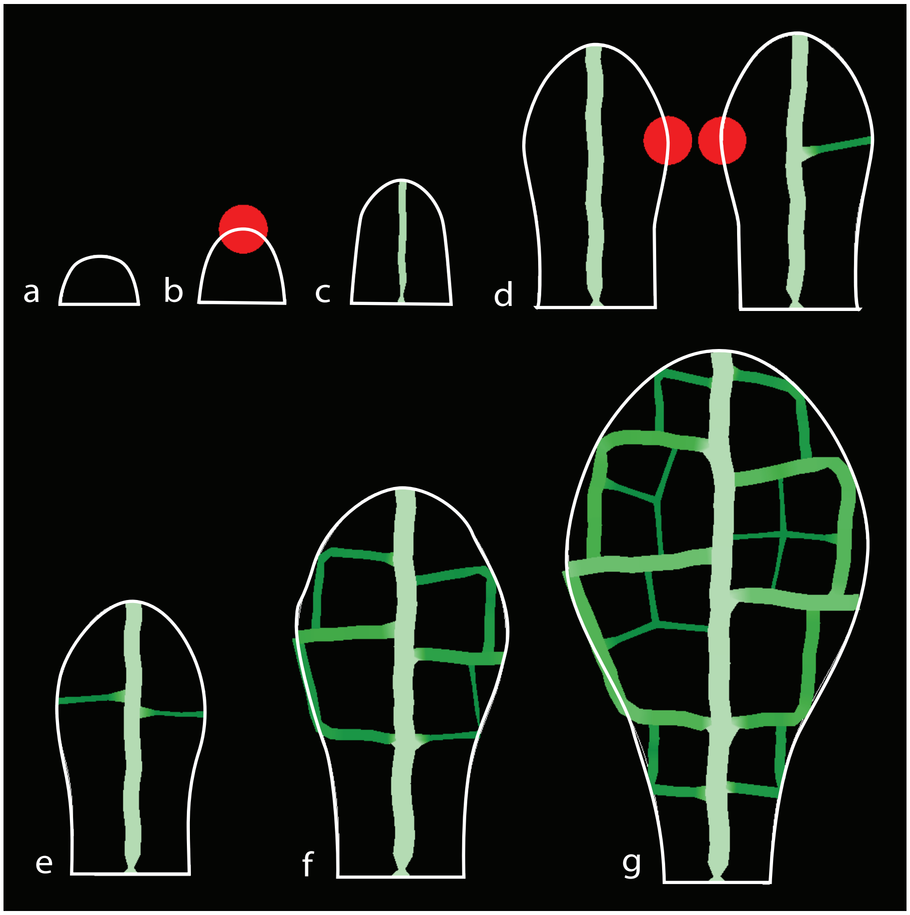


Figure 5.7: Developmental pattern simulated using the local-to-global growth model. The sequence progresses from (a-g).

Using the local-to-global technique, the global form of the leaf emerges from the local specification of expansion. As such, given the initial shape of the leaf the form of the leaf at later time steps is difficult to predict. Additionally, in the simple model used here, growth in the x direction is a function of y, which limits the range of growth patterns the local-to-global growth model can produce.

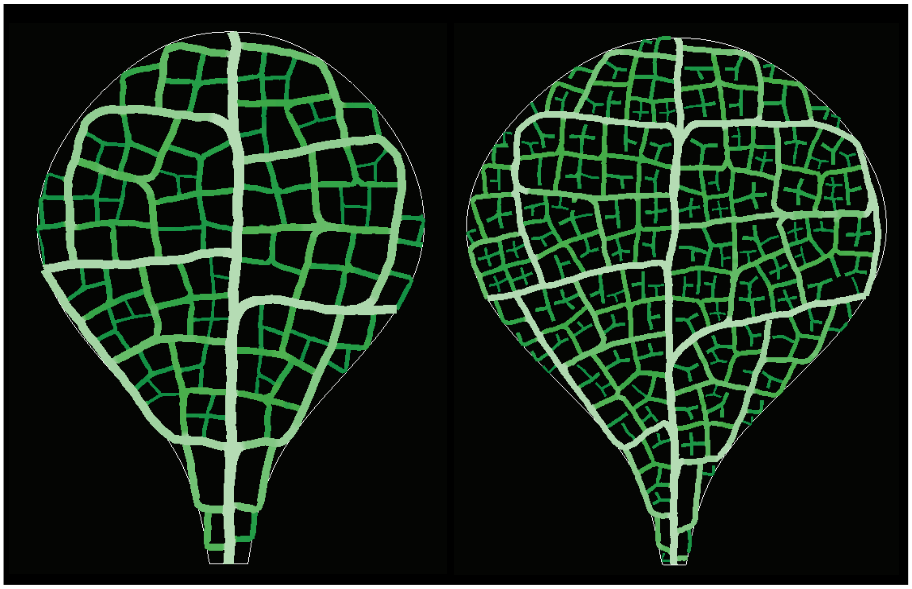


Figure 5.8: Final patterns produced by the local-to-global growth model. On the left, a pattern produced using the local-to-global growth model. The event leading to blindly terminating veins has not yet occurred. The image on the right shows a later stage from the same simulation, following the event leading to blindly terminating veins .

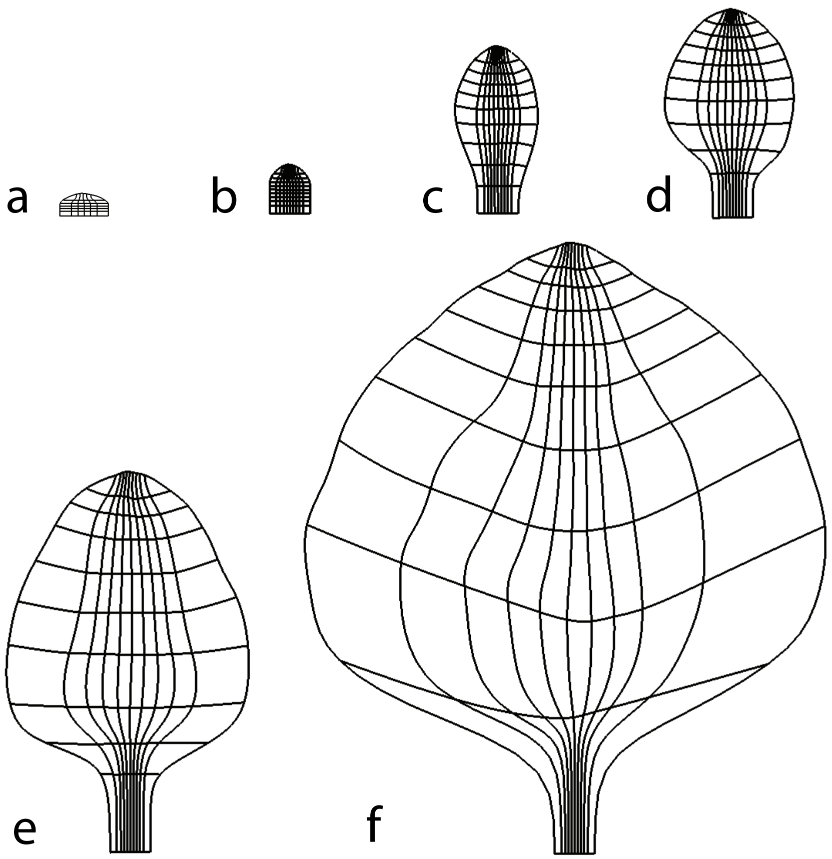


Figure 5.9: Key frames representing the leaf at various developmental stages, progressing from youngest to oldest (a-f).

5.3.2 Development of patterns using key-frame surface interpolation

The progression of forms during development can be specified more easily by interpolating between different "key-frame" shapes, since these shapes, and not growth rates, are directly observable in nature. We can think of this as global-to-local specification of growth, as in this case local growth rates are not given directly but can be deduced from the progression of overall shapes. A model of leaf-blade growth created using a sequence of B-spline surfaces is shown in Figure 5.9.

Surfaces were constructed using the leaf shapes and descriptions reported in the biological literature [149, 150, 30] as a reference and are shown approximately to scale. Internal growth was interpreted by manually tracking landmarks, such as veins and convergence points, throughout development.

The resulting simulation of vein pattern formation (Figures 5.10 and 5.11) correctly recreates the developmental events observed by Scarpella et al. [150], and addresses the issues highlighted for the previous progression. The final pattern corresponds well with observed patterns, plausibly recreating the form of the three loops and higher order veins (compared to the wild-type patterns in Figure 2(A) of [160] and Figure 1(A) of [19]). Addition of sources as the leaf expands creates patterns reminiscent of many mature dicotyledons, as can be seen in Figure 5.13 and with blindly terminating veins in Figure 5.14.

The growth patterns created using the global-to-local growth model generate the midvein, second order veins, and an emerging hierarchy of veins in the correct progression. Subdividing vein nodes as described in Section 5.2.6 eliminates angular features in lower order veins (compared to Figure 5.12). On the other hand, the branching angles between different orders of veins are inconsistent and contrary to that of natural leaves. This is due in part to the simplicity of the technique used to subdivide vein nodes.

Some of the discrepancies between real and simulated patterns may be due to the approximate character of the growth models, for which quantitative experimental data are yet unavailable.

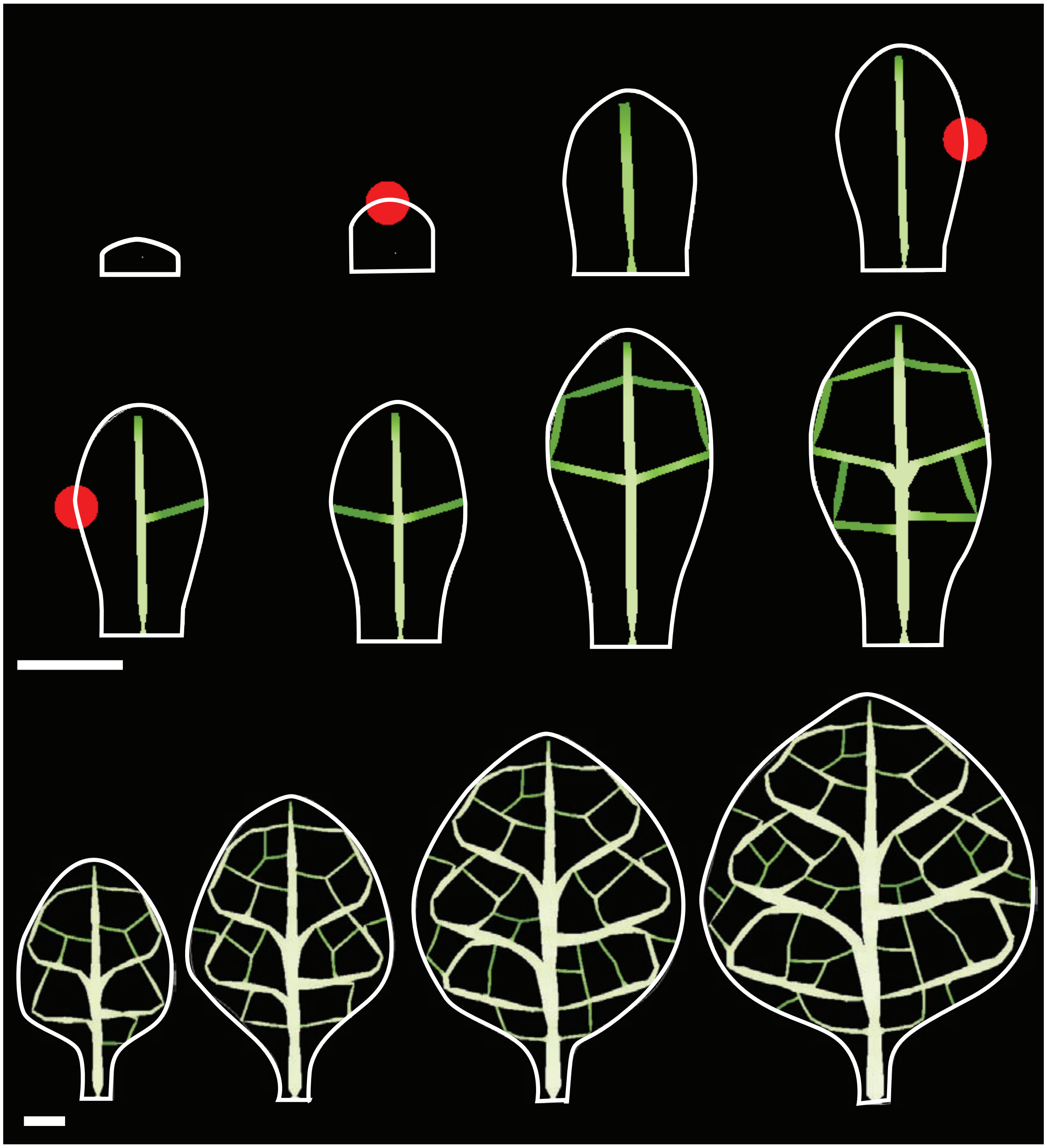


Figure 5.10: Developmental pattern simulated using key-frame interpolation of leaf blade development. The sequence progresses from left to right and top to bottom. The bottom row has been scaled by 0.4 compared to the top two rows (scale bars are provided for comparison).

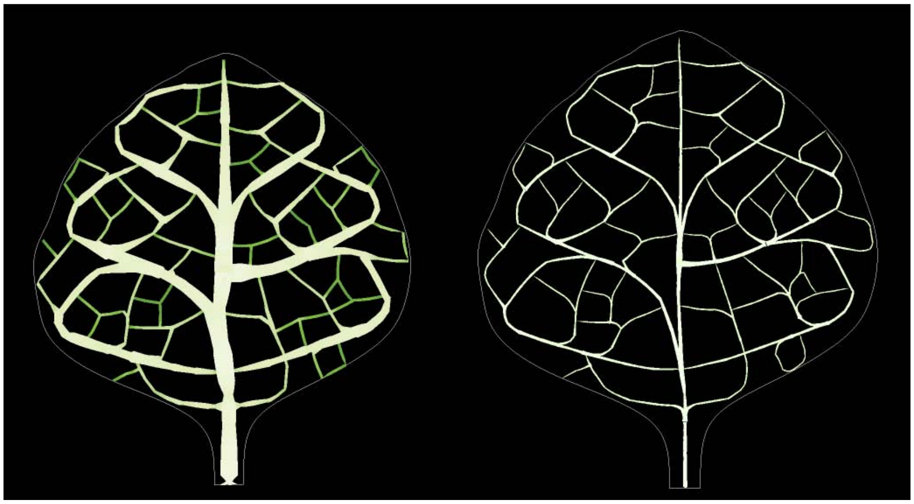


Figure 5.11: Final patterns produced by key-frame interpolation. The final pattern in the developmental progression is shown on the left, and on the right the final pattern produced when the threshold for subdividing vein nodes is lower

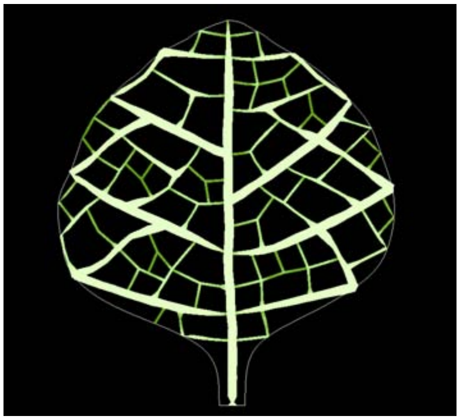


Figure 5.12: The final pattern produced when mid-point subdivision is applied

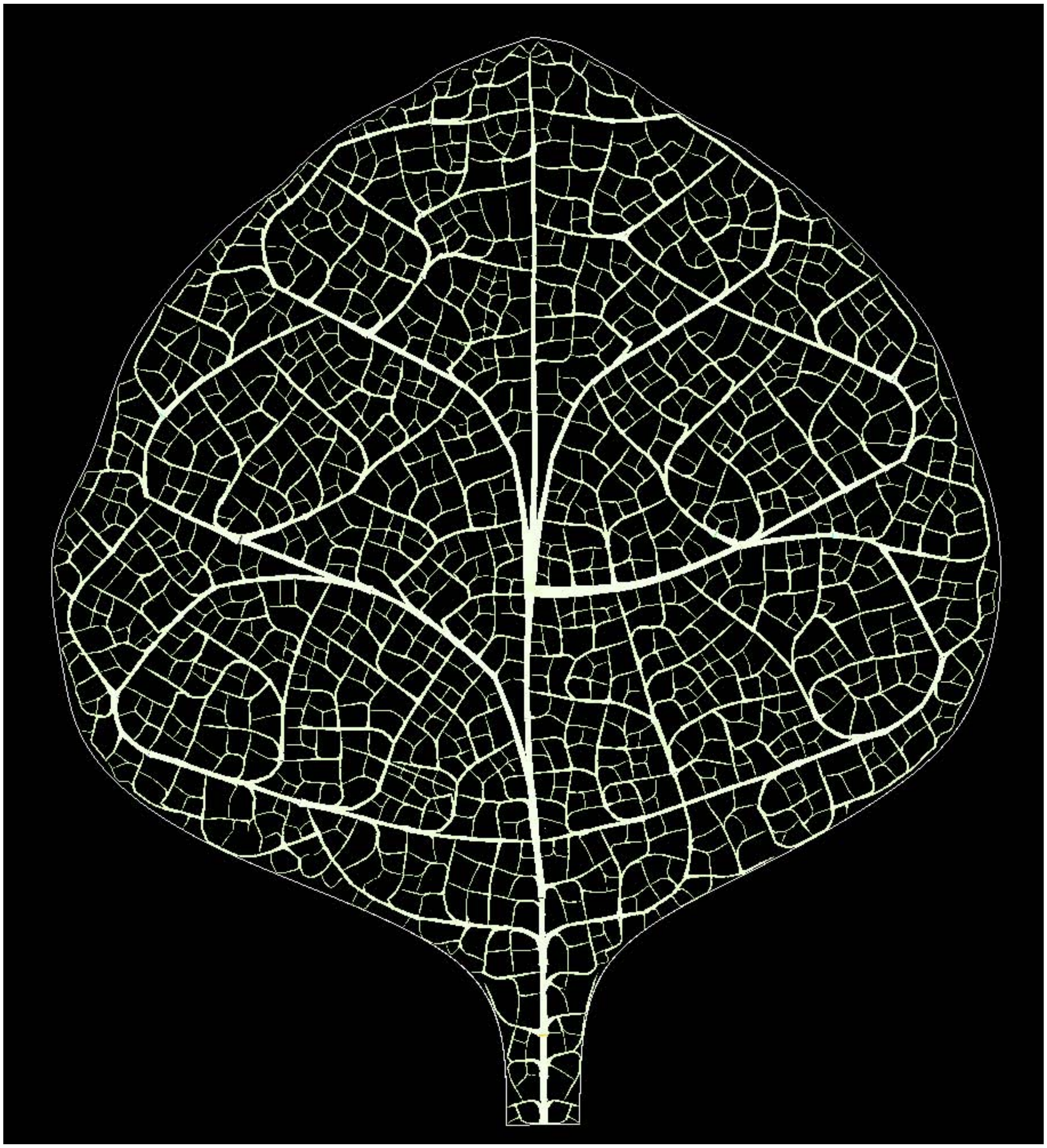


Figure 5.13: Mature pattern produced using key-frame interpolation shown without blindly terminating veins.

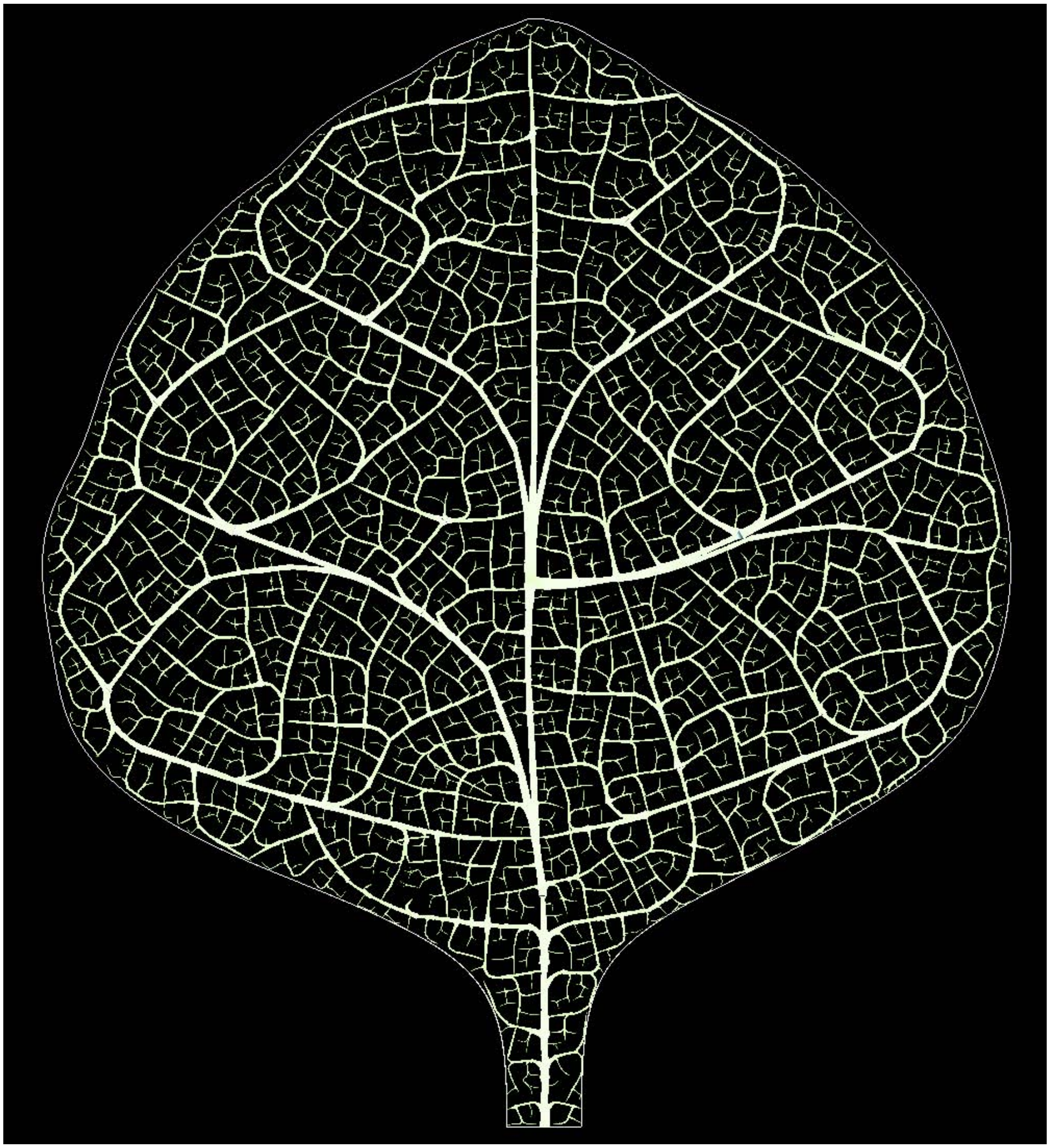


Figure 5.14: Mature pattern produced using key-frame interpolation shown with blindly terminating veins.

5.4 Conclusions

In this chapter the closed algorithm introduced in the previous chapter has been adapted to simulate the development of patterns *de novo* in a growing leaf blade. This was demonstrated by recreating the development of the first leaf of *Arabidopsis*. The key modifications to the model include an increased reliance on growth and are based on two key phenomena. First, highest order veinlets may form open branching structures in the areoles. Second, main-veins are relatively smooth. The model operates as a reiterative process, creating successively higher orders of venation in a manner which is consistent with current biological hypotheses.

In the algorithm for creating areoles, the order in which vein nodes are placed when selected by an auxin source was modified, allowing for the creation of open structures in otherwise closed patterns. Vein nodes are placed according to a hypothetical progression of differentiation in a pre-patterned tissue.

In contrast, the closed model always extends veins towards a source, causing reconnections to occur at the source. Experiments performed using Mitchison's model have shown that it is possible to form polarized files of cells starting at a vein and extending towards a source of auxin [140]. Thus, the formation of veins in the original model appears consistent with the pre-patterning of strands from sink to source, as can be created with Mitchison's model.

With this in mind, the closed and areole algorithms seem to simulate different facets of the same phenomenon. The closed model simulates the formation of a pre-pattern, which eventually differentiates into vasculature, and the areole model simulates the differentiation of vasculature following this pre-patterning process.

As a side note, the structure of areoles was reproduced by assuming that the event leading to blindly terminating veins stops veins from reconnecting, but does not prevent further patterning and differentiation. Scarpella et al [149] proposed that this event precludes the formation of any additional vascular tissue. This assumption seems to be incongruent with the open structures of two or even three orders seen in the areoles of some species. Interestingly, the FORKED genes discovered by Steynen et al. [160] may play a role in this process. In plants mutant in the *forked* genes, vascular patterning occurs, but veins do not reconnect, thus providing a plausible mechanism for this assumption.

The technique used to subdivide the structure of veins allows the curving of secondary veins to be addressed, but also introduces artifacts. As the technique described in Section 5.2.6 does not take the direction, or order of veins into account, the angle between branches can be inconsistent with what is observed in nature. A subdivision scheme that accounted for these characteristics would improve the visual quality of results. Subdivision for branching topologies was addressed by Prusinkiewicz et al. [130], and it is an interesting question whether this work could be extended to the network topology of vascular patterns.

Finally, using the growth models proposed in this chapter, it should be possible to elucidate through simulation the manifold relationships between various growth patterns and the resulting vasculature.

Chapter 6

Modeling vascular patterns in stems

In addition to the visually conspicuous venation of leaves, vasculature is also found in the stem. The vasculature in the shoot connects all organs to the roots, and extends dynamically to accommodate the functional needs of new organs as they are produced. Auxin originating in developing leaves is pumped into the stem, driving the formation of new veins [132].

Modeling vasculature in the stem allows for several interesting biological questions to be addressed. The relation between vascular patterning in the stem and the placement of primordia at the apex of a plant has been a subject of debate within biology [76], leading to the emergence of two opposing viewpoints. The first, is based on the hypothesis that primordia form at the apex of the plant, and are unaffected by the vasculature present in the stem, so the development of primordia drives the vein formation in the stem [132]. The opposing viewpoint is consistent with the hypothesis that vascular patterning in the stem determines the position of new primordia at the apex [84]. Several models of phyllotaxis operating on the surface of the shoot apical meristem have recently been proposed [157, 71], creating impetus to model the formation of stem vasculature as a logical consequence of phyllotaxis, according to the first hypothesis.

The model presented in this chapter incorporates the assumption that the patterning of veins in the stem is an extension of the process in the leaf. Although plausible, this assumption needs to be tested.

These questions are examined below by extending the closed model of leaf venation from Chapter 4 to the cylindrical geometry of the stem. In contrast to previous chapters, which describe models in which veins grew towards sources isolated from existing vasculature, veins now extend toward each other, reflecting the observed development of veins in the stem. This modified model recreates the observed patterns of stem vasculature, with the repeated addition of organs at the apex as a driving mechanism.

6.1 Background

The primary vasculature in a plant shoot supplies nutrients to other organs. Consisting of vascular bundles which extend longitudinally through the stem, veins branch regularly to connect to vascular strands, called *leaf traces*, originating at the *primordia* (undeveloped organs) produced at the apex [117], as illustrated in Figure 6.1.

As a new primordium develops, a leaf trace emerges, connecting the primordium to the vasculature in the stem. At the base of the stem veins connect to the *hypocotyl region*, a solid ring of vasculature joining the shoot to the root. In dicotyledons two seed leaves, or cotyledons, are present while the plant is still an embryo with vasculature that connects directly to the hypocotyl.

As with leaves, vascular patterns in the stem can be classified as *open* or *closed* [32, 117, 76] where open patterns do not reconnect and form tree-like structures, and closed patterns contain cycles.

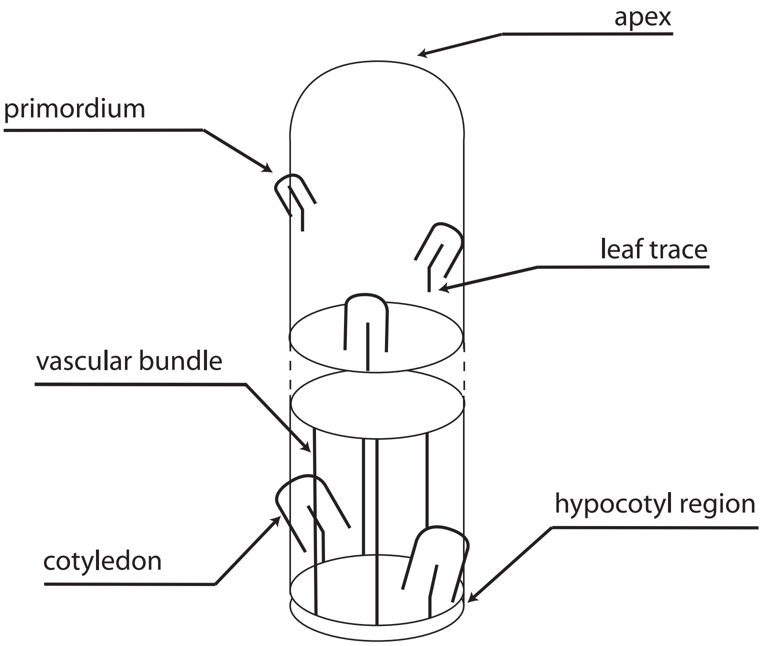


Figure 6.1: Diagram illustrating the anatomy of the shoot

6.1.1 Vascular development in the stem

Reinhardt et al. [132] observed the initiation of vascular strands during primordium development at the apex of the *Arabidopsis thaliana*. When a primordium forms, the auxin that it produces is transported into the stem, leading to the expression of PIN1 in subepidermal tissues. The auxin originating in the primordium is transported basipetally, pre-patterning the vascular tissues giving rise to the leaf trace. In contrast, differentiation seems to proceed acropetally, from existing vasculature toward a developing primordium [96, 34]. Consequently, the leaf trace becomes connected to older veins below [34].

6.1.2 Topology of veins in the stem

Although auxin and PIN1 are known to play a role in vascular patterning of the stem, little information concerning their roles in this process is available. Compared to the detailed descriptions of vascular patterning in the leaf, much less is known about vascular patterning in the stem.

Fortunately, the topology of veins connecting primorida in the stem has been documented for several species. In the following discussion primordia are numbered in the order they were produced. Thus, primordia that develop later have higher numbers and are found above (distally to) primordia with lower numbers on the stem.

The phyllotactic arrangement of primordia is correlated with the architecture of primary vasculature inside the stem [76, 165, 46, 174]. When spiral phyllotaxis is considered this can be further quantified. Spiral phyllotaxis is characterized by the divergence angle θ, between successive primordia.

In plants exhibiting spiral phyllotaxis, characterized by the golden angle ($\theta = 137.5$), and open primary vasculature it has been observed that the primordium numbered n tends to connect to the primordium numbered $n + F_i$ above it [68], where F_i is an element of the Fibonacci sequence $< 1, 2, 3, 5, 8, 13, 21, \cdots >$. For patterns with closed primary vasculature, primordium n connects to the primordium numbered $n+F_i$ and $n+F_{i+1}$ above it, where F_i and F_{i+1} are consecutive elements of the Fibonacci sequence [76, 46, 174]. The same relations have been observed in plants whose phyllotactic angle is the Lucas angle ($\theta = 99.5$), where primordia connect along consecutive elements of the Lucas sequence $< 1, 3, 4, 7, 11, \cdots >$ [46, 68], instead of

Fibonacci numbers .

6.2 Generation of vascular patterns in the stem

To generate vascular patterns in the stem the closed model described in Section 4.3 is modified to operate in the volume of the stem. Additionally, to allow veins to grow towards each other all vein nodes are considered to be sources. The latter modification can be supported to some degree by the fact that veins originating at a developing primordium transport auxin into the stem, thus serving as sources to the vasculature below them [132]. The specifics of this process are given in Section 6.2.1. The model was implemented in *vv*, using the 3D Delaunay triangulation routines included in CGAL [118] to accelerate relative neighbourhood calculations.

Patterning is simulated in a non-elongating cylinder of radius R, representing the stem, with new primordia being added to the distal end with a constant *plastochron* (interval between the placement of two primordia). A vein node is placed at each primordium, representing the tip of a leaf trace entering the stem. The first primordium is placed H units above the base of the cylinder. Each successive primordium is placed relative to the previous primordium, H units above and at a rotation of θ, which produces a spiral phyllotactic pattern. The model is initialized with a ring of vein nodes at the base of the stem, representing the hypocotyl region.

6.2.1 Placement of new vein nodes

The structure of vein nodes is represented by the venation graph $G = \langle V, E \rangle$ as defined in Section 3.1.2. The set $S(v)$ of sources that influence a vein node v is again calculated using the relative neighbourhood, but is modified to reflect the assumption that all vein nodes function as sources. Additionally, v may not be selected by any vein node to which it is connected by an edge in E, this set is denoted $N(v)$ where:

$$N(v) = \{p \in V | (p, v) \in E \vee (v, p) \in E\}. \tag{6.1}$$

Taking these modifications into account, the definition of $S(v)$ given in equation 4.2 becomes:

$$S(v) = \{s \in V - N(v) | v \in V(s)\}, \tag{6.2}$$

where $V(s)$ is the relative neighbourhood of s. Directions of vein development are calculated from $S(v)$ as in the case of closed patterns. A new vein node v' is connected by a segment to any vein node in the set $S(v')$, closer to v' than D, the basic unit of the model. This allows veins originating at an primordium to connect to the veins growing upwards towards them, and vice-versa. A simple example demonstrating the modified model is provided in Figure 6.2.

The implications of these changes may not be immediately clear. First, calculating the set $S(v)$ requires searching much larger sets than in the algorithm presented in Section 4.3 (all nodes formed so far as opposed to sources). Second, the Urquhart approximation of the relative neighbourhood is not used. A comparison of patterns produced with and without the Urquhart approximation exhibit significant differences in the connections between primordia and as such the relative neighbourhood is used instead.

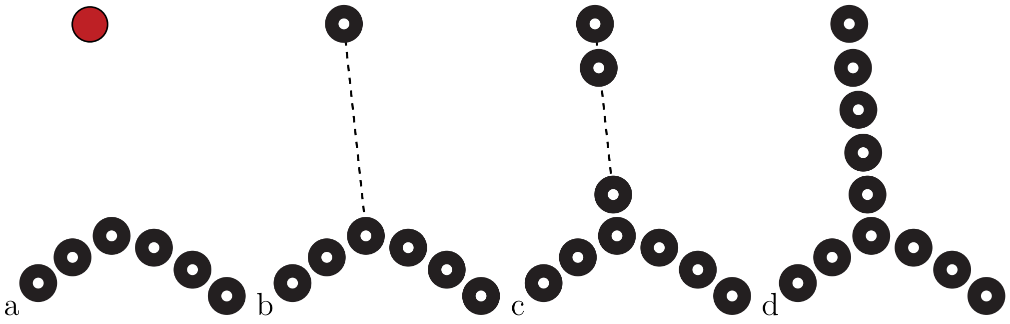


Figure 6.2: A simple example using the model of vascular patterning in the stem. Vein nodes are visualized as black circles with white centres. a) The vein nodes at the base of the image represent the existing vasculature in the stem, the red circle above represents the leaf trace of a new primordium. b) The dashed line connects vein nodes which select each other. c) Nodes are placed as described in the text, this process is repeated in (d) connecting the new primordium to the existing vasculature below it.

6.2.2 Visualization of vascular patterns in the stem

Veins are rendered using generalized cylinders as described in Section 3.2.3 for trees. The diameter of the veins is specified as a linearly decreasing function of height, making veins near the base of the stem thicker than those at the tip. Primordia are represented by spheres offset from the cylinder's surface.

Biologists often visualize the topology of stem vasculature with idealized diagrams, where vasculature is presented in two dimensions as if the stem has been split and unrolled (e.g. Figure 6.3). Such diagrams allow the topology of patterns to be examined visually, without being obscured by the 3d structure of the vasculature. Synthesized patterns can be visualized similarly, allowing the topology of patterns to be verified easily. This is performed by mapping the surface of the cylinder to a rectangular region of the x,y plane (Figure 6.4). Each vein node $p = (x, y, z)$

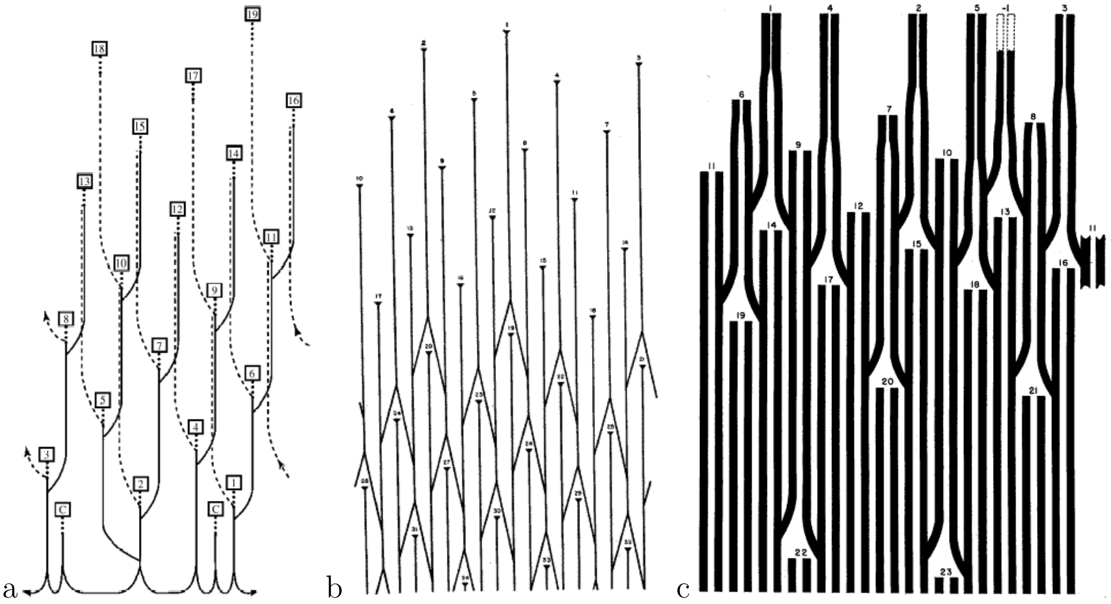


Figure 6.3: Idealized diagrams of stem vasculature. a) An idealized diagram of stem vasculature of the rosette in *Arabidopsis thaliana* [76]. In (b) and (c) similar diagrams are shown, created for the stem of *Linum usitatissimum* (the flax plant) [46] and *Ginkgo biloba* [51], respectively.

is mapped as follows:

$$(x, y, z) \longrightarrow (\arctan(x/y), z, 0) = (\theta, z, 0) \tag{6.3}$$

Patterns are still produced in 3d, but are mapped to the x,y plane prior to the creation of generalized cylinders.

Internodal growth is not simulated as part of the patterning process. For visualization purposes however, internodal growth is simulated as a post-processing step by mapping vein nodes based on their normalized height in the stem to a new height using the function shown in Figure 6.5.

Examples illustrating the visual impact of unrolling the stem and applying the growth map are shown in Figure 6.6.

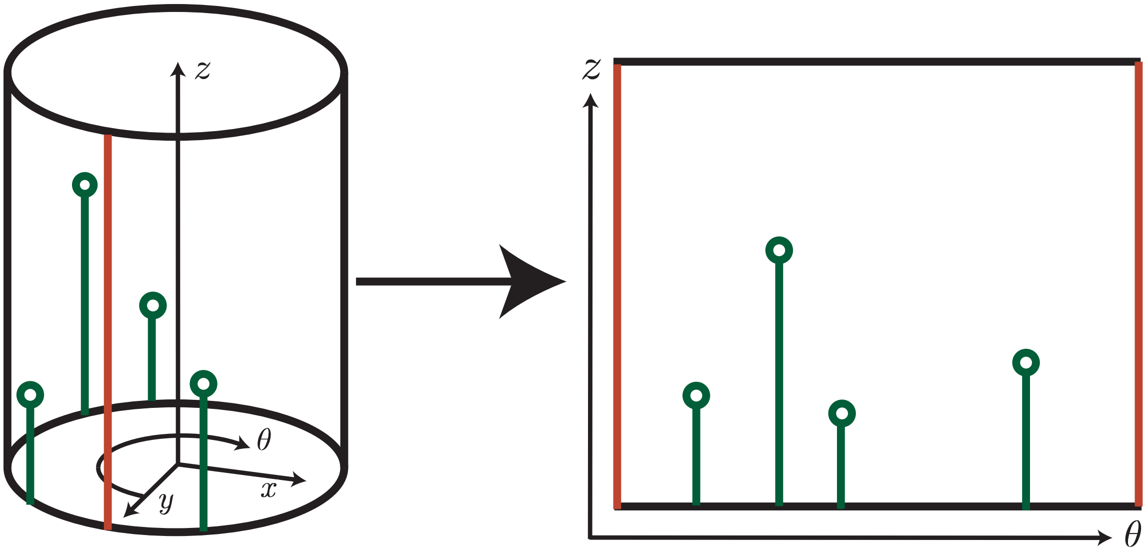


Figure 6.4: Visualizing patterns as in idealized diagrams. The stem is unrolled by splitting it along the red line (at $\theta = 0$) and mapping it to a rectangle in the plane.

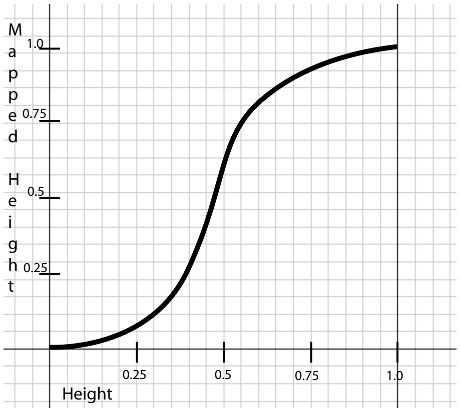


Figure 6.5: Growth mapping created from data for the rosette of *Arabidopsis* [26]. Given a normalized height (x-axis) the function returns the height of the point after growth has been simulated (y-axis).

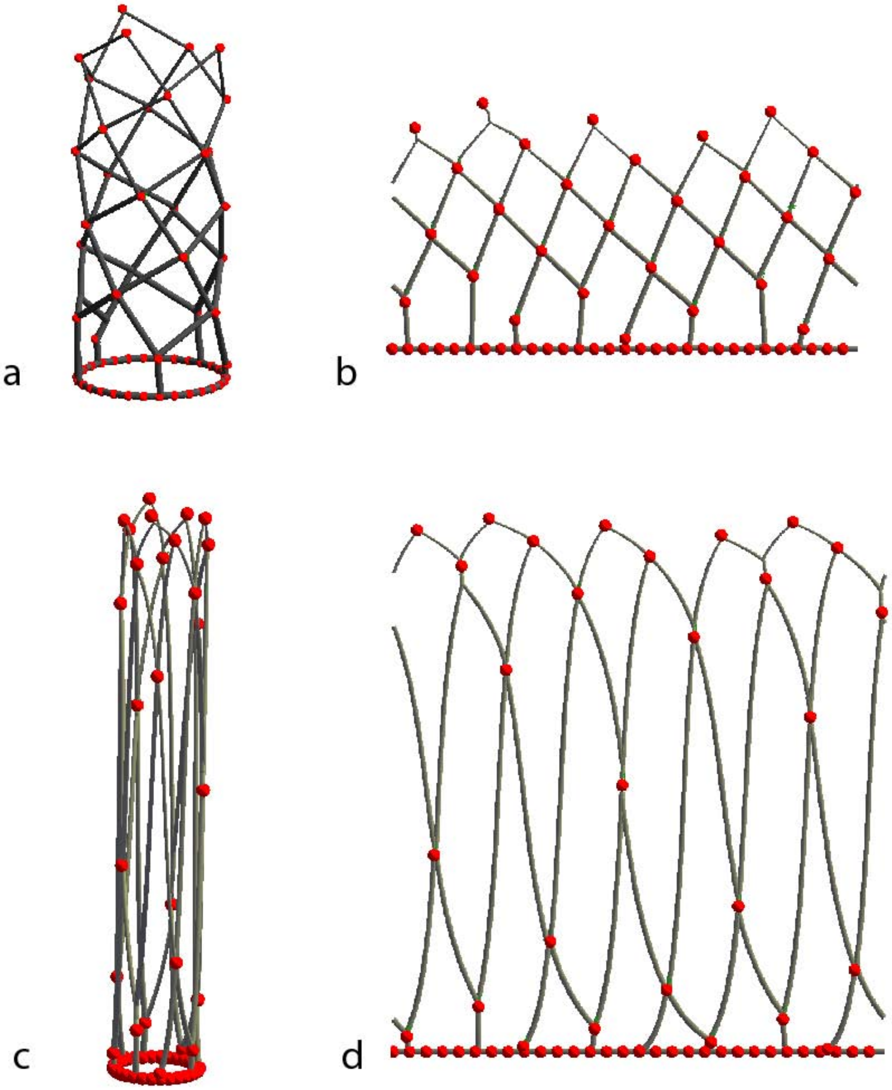


Figure 6.6: Application of mappings simulating growth and unrolling the stem to a sample pattern. A pattern produced using the model is shown in (a), the same pattern is shown: (b) unrolled, (c) after the application of the growth map (c), and (d) unrolled and scaled using the growth map.

6.3 Results and discussion

The model postulates a causal relation between the development of the phyllotactic pattern, stem growth, and the resulting vasculature in the stem. To verify whether this model produces patterns as observed in biological literature [76, 84, 46, 165] patterns with a spiral arrangement of primordia, placed using the Lucas or golden angle, were created and examined, and in this sense validate the model. Patterns generated using these two angles while varying R and H were considered and the resulting topologies documented.

The simulation is initialized with a set of vein nodes at the base of the stem. As the simulation begins, the points quickly connect into a solid ring (corresponding to the solid vascular cylinder of the hypocotyl). Following the initialization of the model, primordia are added with a constant plastochron. The duration of the plastochron compared to the rate of vein propagation affects the geometry of patterns, but in our experiments the topology of the pattern was unaffected by these parameters (Figure 6.7). In contrast, the parameters controlling the radius of the stem, R, and vertical displacement between primordia, H, change the topology of patterns considerably.

When the golden and Lucas angles are used, patterns exhibit extremely regular connections between primordia, which can be used to categorize the patterns produced by the model. The *connectivity* of a vascular pattern in the stem is denoted (a, b), where a, b are integers such that primordium n connects to primordia numbered $n + a$ and $n + b$ above (distal to) it.

When the golden angle is used, after some potential irregularities near the base, primordium n connects to the primordia numbered $n + F_i$ and $n + F_{i+1}$ above it, for

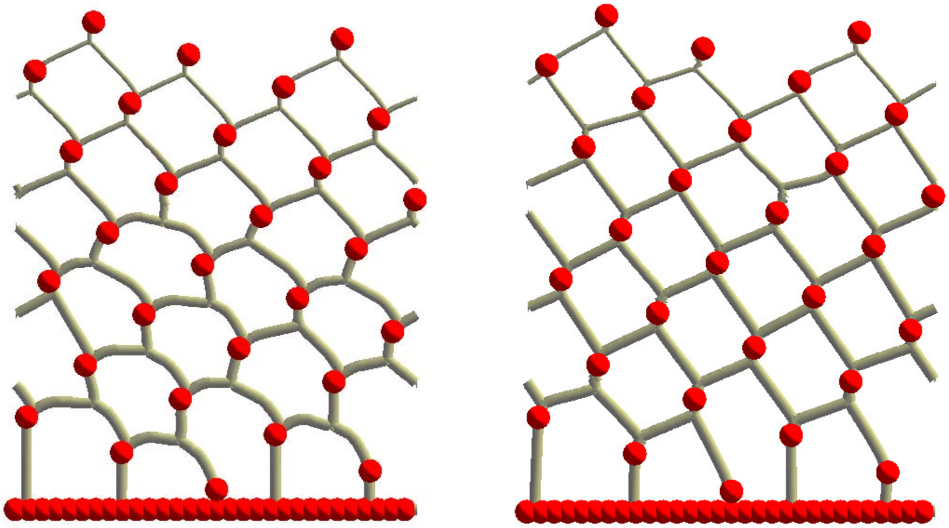


Figure 6.7: Two patterns of stem vasculature produced with different plastochrons but otherwise identical parameters (the patterns are visualized as unrolled stems). In both simulations each primordium was placed relative to the previous primordia, $10D$ above, and at a rotation of 137.5 degrees; the radius of the stem was $50D$. The pattern on the left was generated with a much shorter plastochron then that the image seen on the right.

some $i \geq 0$. Over a range of parameters only a few primordia near the base exhibited a connectivity other than (F_i, F_{i+1}), regardless of the number of primordia produced. Patterns of type (1,2) provide the single observed exception to this characterization, as primordium n would occasionally fail to connect to primordium $n+1$.

The ratio of R/H affects the connectivity of the pattern and seems to determine which i, or successive Fibonacci numbers, are chosen. As R/H increases i also increases, causing primordia to connect to younger and younger primordia above. Decreasing R/H has the opposite effect, resulting in primordia connecting to older and older primordia, until each primordium connects to the primordia immediately preceding and following it, producing a (1,1) pattern. Maintaining the same ratio between R/H while varying the absolute values of R and H does not change the connectivity of patterns.

The relation between R and H when $\theta = 137.5$ is illustrated in Figures 6.8 and 6.9. Of particular interest is the pattern exhibiting (5,8) connectivity in Figure 6.9 which is consistent with patterns observed in the rosette of *Arabidopsis thaliana* and the stem of *Ginkgo biloba* [51]. The instability observed in (1,2) patterns can be seen in the penultimate image in Figure 6.9, where primordia do not always connect to the following primordium (specifically, the primordia numbered 2, 3, 7, 9, 10, 12, 14, 15, 17, 18, 20, 21, and 23).

Spiral phyllotaxis with the Lucas angle produced similar results, with connections occurring along successive Lucas numbers, as is observed in nature. Varying the ratio R/H affected connectivity exactly as when the golden angle was considered, but with Lucas numbers playing the role of Fibonacci. Several example patterns, produced by varying R while holding H constant are shown in Figure 6.10. The (7,11) pattern in particular has been observed in flax by Girolami [46].

6.4 Conclusion

Through simulation the regular connectivity of vascular patterns in the stem was recreated for plants exhibiting spiral phyllotaxis. The generated patterns exhibit a tendency for connections to form according to successive Fibonacci and Lucas numbers, as is observed in a variety of plants [76, 84, 46, 165, 68].

Based on this model the questions posed at the beginning of this chapter can be addressed. The geometric assumptions underlying the algorithm are derived from the canalization hypothesis, indicating that at a geometric level canalization is sufficient to explain vascular development in the stem, as well as in the leaf.

The model also shows that it is possible to obtain the vascular patterning seen in the stem when phyllotaxis is assumed to occur independent of vascular patterning, and even drive the process of vascular patterning in the stem. This does not preclude the possibility that feedback occurs between phyllotaxis and vascular development in the stem. Most likely these processes stabilize each other to some degree [76], and it is possible that phyllotactic patterns could be completely accounted for by both processes.

In our simulations the parameters R and H control the topology of patterns in the stem. This tends to indicate that differences in the topology of vascular tissues in the stem among species could be explained by the rate at which primordia are produced or differences in apex size. Interestingly, changes in apex size during development correlate with changes in phyllotaxis [81, 165], and presumably the topology of veins

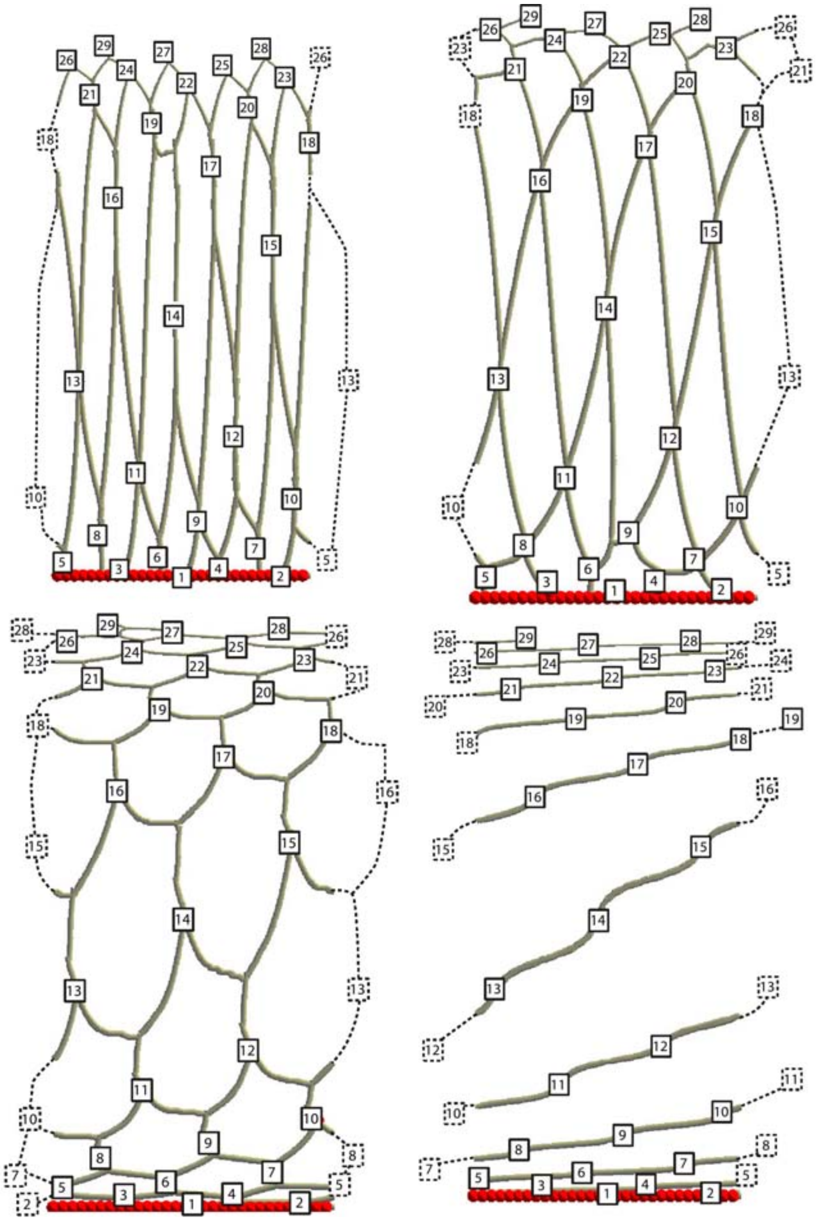


Figure 6.8: Parameter space exploration where only the step between primordia (H) is varied, with $\theta = 137.5$ and $R = 115D$. The step between primordia is smallest for the top-left image and increases in subsequent images. Primordia are numbered in the order which they were produced. Patterns are visualized as an unrolled stem, and the growth map shown in Figure 6.6 has been applied. Boxes with solid outlines indicate the location of primordia. Dashed boxes and lines indicate connection that were severed when the stem was unrolled, and extend the pattern at the boundary. The connectivity and H value of the images from top-left to bottom-right is (5,8) with $H = 10D$, (3,5) with $H = 23D$, (2,3) with $H = 33D$ and (1,1) with $H = 96D$.

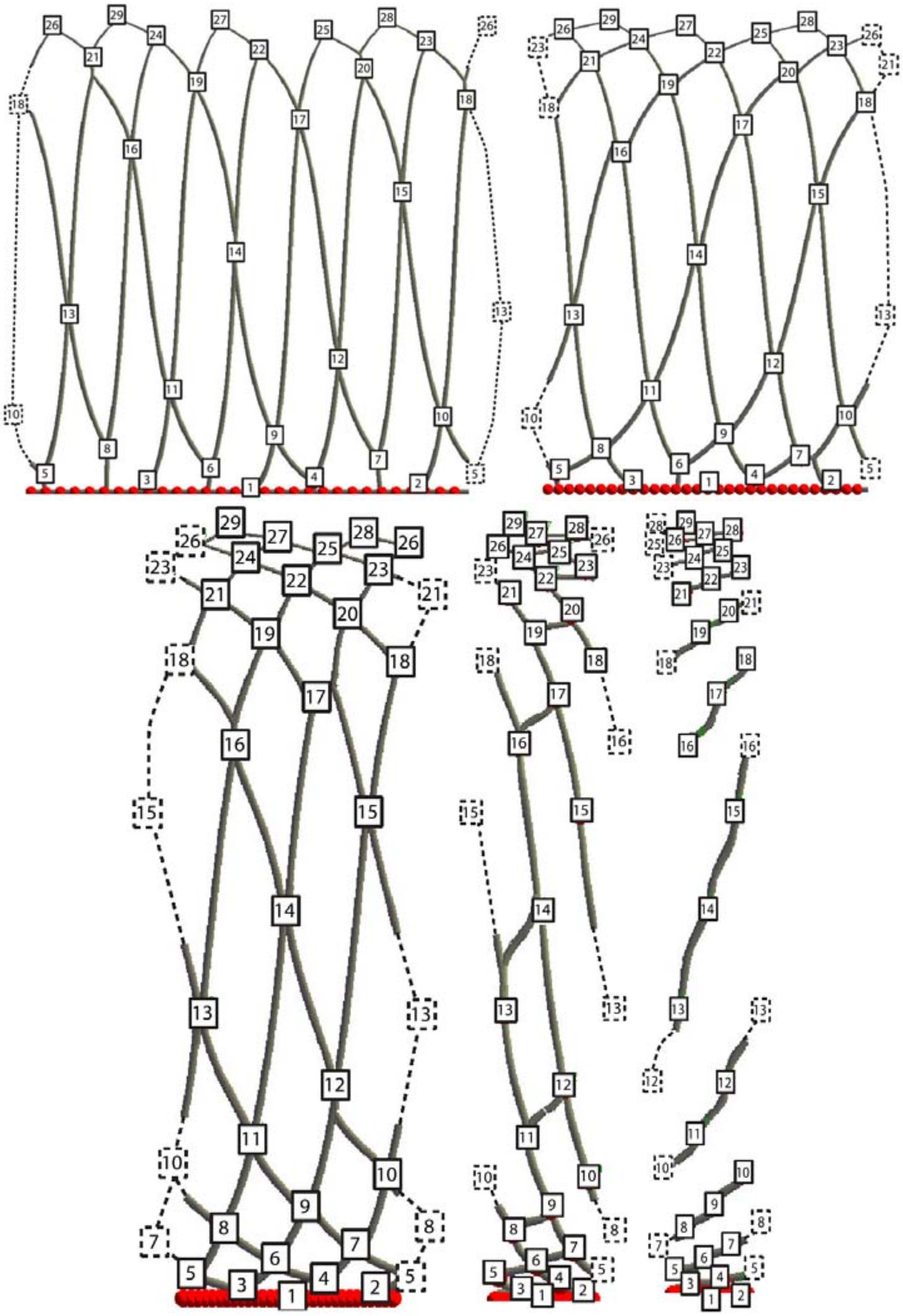


Figure 6.9: Parameter space exploration where only the radius of the stem (R) is varied, with $\theta = 137.5$ and $H = 10D$. The radius is largest for the top-left image and decreases in subsequent images. The connectivity and R value of the images from top-left to bottom-right is (5,8) with $R = 115D$, (3,5) with $R = 50D$, (2,3) with $R = 35D$, predominantly (1,2) with $R = 16D$ and (1,1) with $R = 12D$.

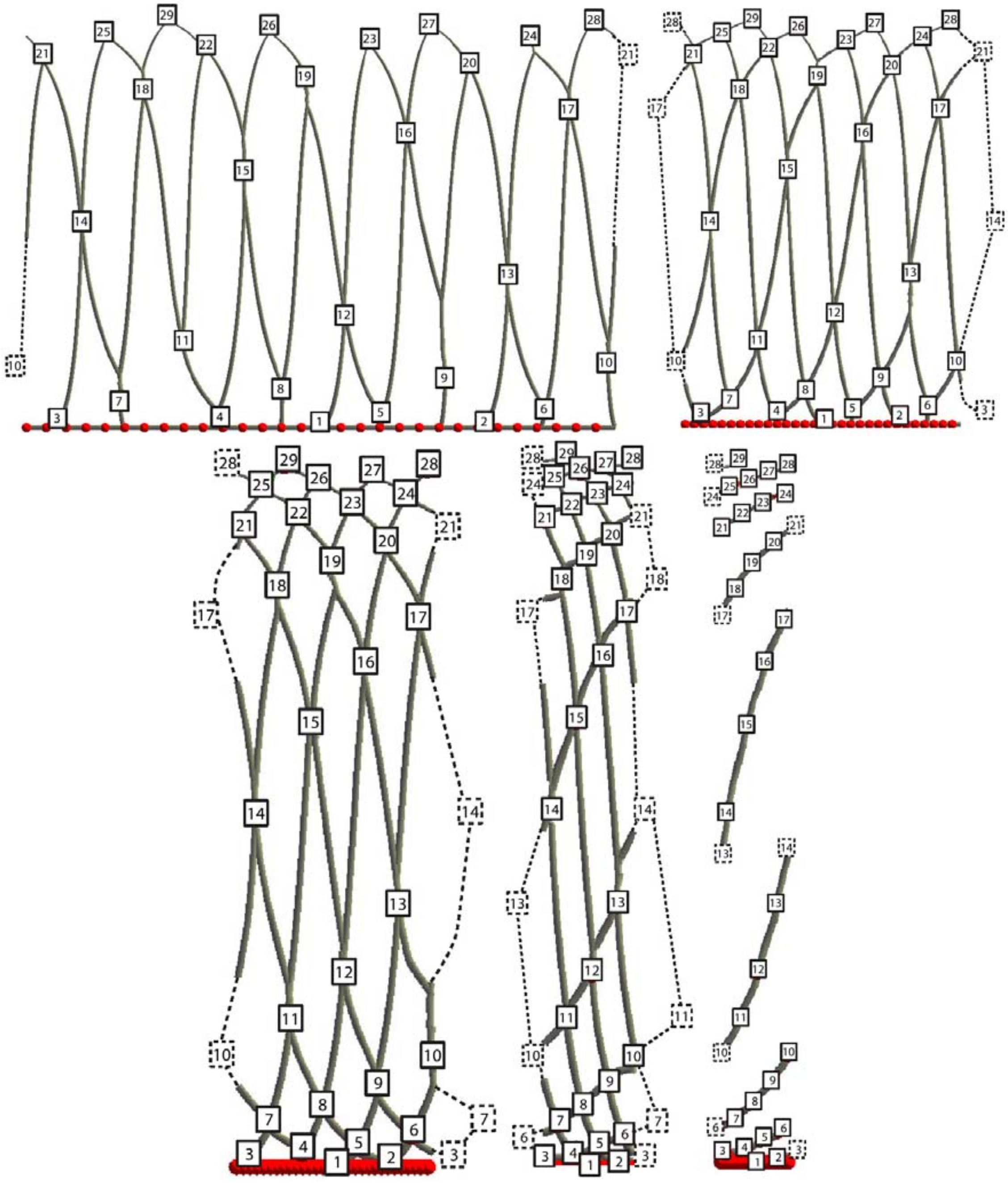


Figure 6.10: Parameter space exploration where only the radius of the stem (R) is varied, with $\theta = 99.5$ and $H = 10D$. The radius is largest for the top-left image and decreases in subsequent images. The connectivity and R value of the images from left to right and top to bottom is (7,11) with $R = 180D$, (4,7) with $R = 115D$, (3,4) with $R = 50D$, (1,3) with $R = 25D$ and (1,1) with $R = 12D$.

in the stem as well. Additionally, in the model of phyllotaxis proposed Smith et al. in [158], varying the apex size or the rate at which primordia are produced changes the pattern of phyllotaxis in the model.

Finally, the pattern of vasculature plays a key role in carbon distribution in plants [174]. Hence, it would be interesting to examine the model from the viewpoint of carbon allocation, and the subsequent effect on plant development.

Chapter 7

Summary and future work

7.1 Summary

The objective of my thesis was to examine the role of competition for space in the generation of branching and networked biological patterns. Spatial relations between pattern elements have been expressed and examined directly in geometric terms. This allowed for focusing on the emergence of global patterns from local interactions, introducing an intermediate level of abstraction between molecular-level phenomena and final forms. The outcome of this work was a class of space-colonization algorithms, which operate by simulating the competition between pattern elements for space. This competition is formulated in terms of iterative geometric operations on sets of points that represent pattern elements and/or markers of free space. The proposed algorithms, and the explicit reliance on competition for space, provide a novel approach for generating branching and networked patterns. I demonstrated the utility of this approach by modeling the branching structure of trees and both open (branching) and closed (networked) pattern of vasculature in plants. Although formulated in abstract geometric terms, the algorithms are justified biologically in each instance.

I first presented a model for generating the branching structure of trees and used it to illustrate the essence of the space-colonization algorithm. In contrast to previous algorithms for generating trees, which emphasized their recursive structure, the proposed algorithm treats competition for space as the dominant factor determining the forms of trees and shrubs. In nature, this competition may be mediated by the quantity and quality of light [113, 5]. Branches are inherently self-avoiding, a property that can be difficult to guarantee with recursive approaches. The trees and shrubs generated be the SCA assume forms that are automatically adapted to the presence of neighbouring plants. I illustrated this algorithm using realistic models of diverse trees and shrubs. The parameters of these models correspond to visually relevant tree characteristics identified in landscaping, and thus offer convenient control of tree shape and structure.

I then used the basic tenets of the algorithm for trees, the marking and subsequent colonization of empty space, to generate leaf venation patterns. Here, the points denoting empty space conceptually represent sources of the hormone auxin, and thus can be interpreted at the level of molecular biology. To take into account that vascular patterning occurs in a growing leaf blade I proposed methods for simulating marginal and uniform leaf growth. By directly adapting the model used for trees to a growing leaf blade I produced open patterns. Closed patterns were generated by allowing multiple veins to extend toward the same auxin source. These two variants of vein-generating algorithms recreated the complex patterns of a variety of mature leaves: a result that had not been achieved previously.

A limitation of the closed and open models is that they cannot reproduce the blindly terminating veins seen in the areoles of many dicotyledons. Consequently, I proposed the areole algorithm, a variant of the closed model that accounted for this feature. Instead of simulating the convergence of growing veinlets on a common auxin source, in the areole algorithm only one veinlet “wins the competition” and grows towards the source. The vein then bifurcates and extends by continuing to grow beyond the source. To account for the non-uniform anisotropic growth observed in some leaves, I proposed two additional methods for simulating the growth of the leaf blade. The first technique makes use of a growth tensor field, specified over the leaf. The second technique simulates the development of leaf shape over time using key-frame interpolation of parameterized surfaces. As a case study, I recreated the development of the first leaf in the rosette of *Arabidopsis*. This case study demonstrated how the leaf vein patterns observed in nature could arise in a self-regulated reiterative process, as proposed by Scarpella et al [149].

Finally, I modified the closed model to operate in the volume of the stem. Vascular bundles of plants may develop acropetally from the hypocotyl region of the stem, and basipetally from the leaf traces that emanate from primordia. To reflect these developmental patterns I allowed veins to grow towards each other, with each vein node acting as a potential source for other nodes. I used this last model to investigate the relation between phyllotaxis and stem vasculature. Current experimental data are inconclusive: some suggest that the formation of stem vasculature is driven by phyllotaxis, whereas others suggest that stem vasculature plays an active role in phyllotactic patterning. I examined the first hypothesis by placing primordia independently of stem vasculature. The vascular pattern emerging in this process exhibited connections with consecutive Fibonacci or Lucas numbers. This result is consistent with vascular patterns observed in several plant species [76, 84, 46, 165, 68]. Furthermore, changes in phyllotaxis lead to changes in the vasculature, which matches the pattern of parastichies (the visually conspicuous lines formed by primordia).

The use of Delaunay triangulations allowed complex vascular patterns to be generated efficiently. Furthermore, each model can be controlled by a small number of intuitive parameters. Thus the algorithms are relevant to image synthesis in computer graphics. Additionally, the algorithms rely on, and are intricately related to, the biological processes they emulate, making this work relevant to the modeling of biological systems.

The wide range of venation patterns generated by the proposed models, and the simulation of the development of venation over time, support the claim that the Space Colonization Algorithms may capture the essence of vein pattern formation in nature. Reaching even further, the method for generating trees is closely related to the methods for generating leaf venation patterns, which in turn are related to phyllotactic patterns [10]. This leads to a unifying view of competition for space as the key factor that determines different aspects of plant patterns and form across many scales of their organization. A further exploration of this view is necessary and should focus on establishing a better understanding of the developmental and molecular mechanisms mediating this competition. Some of these ideas are explored in the following section.

7.2 Future work

The results presented in this thesis demonstrate the fertility of the concept of space colonization, illustrated using applications to the modeling of trees, leaf veins, and stem vasculature. It seems, however, that these results are just the first step in exploring space colonization from the biological, mathematical, and image synthesis perspectives. Some direction for future work along these lines are outlined below.

7.2.1 Tree architecture

In this thesis I considered only competition between branch segments in a given tree or shrub or among plants of the same species. When the later case is considered the assumption that plants belong to the same species is simulated by using the same attraction points for several trees or shrubs. As the number of attraction points, and the distance at which they are removed are species-dependent parameters it is not immediately clear how the current algorithm can allow for competition between plants of different species. It nevertheless should be possible to extend the algorithm to account for inter-species interactions. This would provide a formalism for simulating plant populations at the level of branch segments, allowing for the re-creation of complex interactions that could not be addressed using individual-based techniques [78, 83].

A second major extension required for this task would aim at simulating and animating the development of trees over their life span. This requires incorporating a number of developmental factors into the model, such as the placement of buds, distinction between long and short shoots and dropping of branches. In addition, envelopes should not be explicitly specified, but should be an emergent property of the model.

From the visual perspective, the use of generalized cylinders to model branches does not allow for precise shaping of branching points; my models could benefit from alternative methods addressing this issue [42, 89, 93]. To validate the space colonization algorithm quantitatively, it would be useful to compare the generated structures with measured plants; such comparisons may also help to set model parameters according to the measurements of chosen tree species. Moreover, the use of envelopes makes our method potentially suitable to model trees from photographs; an open question is the possibility of incorporating acquired data that specifies the courses of main branches (c.f. [110]). A further question is the possibility of using the space colonization algorithm in the context of interactive procedural modeling, where key aspects of plant form are specified by the user and elaborated upon by an algorithm [115, 65, 182].

7.2.2 Vascular patterns and leaf modeling

The models of leaf venation generate a wide variety of venation patterns and allow for the incorporation of realistic patterns into synthetic images of plants. Further applications include phenomena that are greatly impacted by the pattern of veins, such as the colouring of leaves and petals and the curling and ageing of leaves [90, 62]. Also, the open and closed algorithms produce interesting space filling patterns that could used as the basis for a system generating complex ornamental patterns (c.f. Wong et al. [181]).

Currently, the growth progressions used in developing patterns are specified manually. This process is both time consuming and quite subjective. Automating the specification of growth is an appealing alternative. It may be possible to achieve this by using image processing techniques to segment images of developing leaves, and register landmarks such as major veins and trichomes. By tracking landmarks over time it should be possible to infer the growth of the leaf algorithmically, as opposed to manually. This is not without precedent, as similar techniques have already been developed to analyze the geometry and expansion of the shoot apical meristem[82].

Figure 7.1: An ornamental kale leaf (Courtesy of P. Prusinkiewicz). This leaf illustrates a strong correlation between the venation pattern and the 3D shape of the leaf.

Finally, the assumption that vein development in the leaf has no feedback on the growth or form of the leaf needs to be revisited. In some leaves this assumption seems overly simplistic (see fig.7.1). Compared to other tissues in the leaf, veins have different mechanical properties [112] and growth patterns [34]. As such, this assumption requires revising. Simulating the development of patterns using the techniques described in Chapter 5 should allow this relation to be examined, and provide insights as to how this may be achieved.

7.2.3 Further extensions

Many problems remain open for future research. Expanding on the final paragraph of Section 7.1, competition for space seems an apt geometrical abstraction for modeling the development of biological patterns. It is an interesting question how broad is the idea of modeling morphogenesis at a geometric level [63], i.e., what is the range of phenomena to which this idea can be applied.

The basic version of SCA outlined for trees has a simple formulation, leading to interesting emergent behaviour. Additionally, considering variants of the SCA algorithm has proved fruitful in this thesis. This seems to indicate that systematic analysis and modification of the basic algorithm should produce a wide range of interesting patterns. In this work the impact of changing the definition of $S(v)$, the sources selecting a vein node, was considered and additional variants in this theme could be imagined.

Given the range of vascular patterns generated in this thesis it would be interesting to investigate the applicability of the SCA to vascularization processes in animals. Additionally, there are striking visual similarities between the patterns synthesized in this thesis and other natural patterns, with river and road networks, fracture patterns, and the structure of some corals providing other potential applications of the SCA model.

A more practical problem that should be addressed by future work is the acceleration of computation. Generation of a tree or venation pattern takes between a few seconds and a few minutes on the current generation of desktop computers with a 3GHz processor. This time strongly depends on model parameters, especially the segment size D used as the unit of length in the models. At present, we do not have an algorithmic criterion for choosing the optimal value of D. It may be possible to calculate D adaptively, by reformulating algorithms as discrete-time simulations. Accelerating the computation of models would allow applications requiring interactive rates to be considered, providing easy specification of trees, or the automatic generation of venation patterns for leaves while the form of the leaf is modified.

Objective evaluation of methods

In this thesis, the plausibility of patterns was assessed primarily by visual inspection. As such, performing some manner of objective validation should be an important aspect of future work. Some measures have already been developed [13, 36], and provide a starting point for further investigation. Nevertheless, some caution should be taken when employing objective measures, as discussed in the following quote from Prusinkiewicz [123, page 1]:

> ...an emphasis on objective, measurable characteristics, as promoted by current scientific practises, may lead in the wrong direction. An easily measurable characteristic may turn out to be irrelevant; on the other hand, a feature that eludes precise definition or measurement may be of central importance.

In fact, modeling is an important tool in understanding and determining which features of a pattern are of central importance. Although important, relying too heavily on objective measures may obscure important details or overly constrain initial attempts to model natural phenomena.

Bibliography

[1] E. Abraham. *The algorithmic beauty of seaweeds, sponges and corals*, chapter A Laplacian model of branching growth, pages 109–113. Springer, 2001.

[2] R. Aloni, K. Schwalm, M. Langhans, and C. Ullrich. Gradual shifts in sites of free auxin-production during leaf-primordium development and their role in vascular differentiation and leaf morphogenesis in *Arabidopsis*. *Planta*, 216:841–853, 2003.

[3] D. Andrade and L. de Figueiredo. Good approximations for the relative neighbourhood graph. In *Proceedings of the 13th Canadian Conference on Computational Geometry (CCCG'01)*, pages 25–28, 2001.

[4] M. Aono and T. L. Kunii. Botanical tree image generation. *IEEE Computer Graphics and Applications*, 4(5):10–34, 1984.

[5] P.J. Aphalo and C.L. Ballare. On the importance of information-acquiring systems in plant-plant interactions. *Functional Ecology*, 9(1):5–14, feb 1995.

[6] G. Baranoski and J. Rokne. Light interaction with plants, 2002. SIGGRAPH 2002 Course Notes 26.

[7] P. Barbier de Reuille, A. Runions, R. Smith, E. Coen, and P. Prusinkiewicz. Tricome patterning on growing tissue. In *Proceedings of the workshop on Functional Structural Plant Models, to appear*, 2007.

[8] A. D. Bell. The Simulation of Branching Patterns in Modular Organisms.

Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences, 313(1159):143–159, August 1986.

[9] J. L. Bentley, B. W. Weide, and A. C. Yao. Optimal expected-time algorithms for closest point problems. *ACM Transactions on Mathematical Software*, 5(4):563–580, 1980.

[10] T. Berleth, E. Scarpella, and P. Prusinkiewicz. Towards the systems biology of auxin-transported-mediated patterning. *Trends in Plant Science*, 12(4):151–159, 2007.

[11] J. Bloomenthal. Modeling the Mighty Maple. *Computer Graphics (SIGGRAPH '85 Conference Proceedings)*, 19(3):305–311, July 1985.

[12] J. Bloomenthal. *Skeletal design of natural forms.* PhD thesis, University of Calgary, Calgary, Alberta, January 1995.

[13] S. Bohn, B. Andreotti, S. Douady, J. Munzinger, and Y. Couder. Constitutive property of the local organization of leaf venation networks. *Phys. Rev. Lett.*, 65(6):061914, June 2002.

[14] A. Bowyer. Computing Dirichlet tessellations. *The Computer Journal*, 24(2):162–166, 1981.

[15] E. Catmull and R. Rom. *Computer Aided Geometric Design*, chapter A Class of Local Interpolating Splines. Academic Press, 1974.

[16] G. Chaikin. An algorithm for high speed curve generation. *CGIP*, 3:346–349, 1974.

[17] N. Chiba, S. Ohkawa, K. Muraoka, and M. Miura. Visual simulation of botanical trees based on virtual heliotropism and dormancy break. *The Journal of Visualization and Computer Animation*, 5:3–15, 1994.

[18] M. Cieslak. Stochastic simulation of pattern formation: An application of l-systems, July 2006.

[19] N. K. Clay and T. Nelson. Arabidopsis thickvein Mutation Affects Vein Thickness and Organ Vascularization, and Resides in a Provascular Cell-Specific Spermine Synthase Involved in Vein Definition and in Polar Auxin Transport. *Plant Physiol.*, 138(2):767–777, 2005.

[20] E. Coen, A. Rolland-Lagan, M. Matthews, J. A. Bangham, and P. Prusinkiewicz. Inaugural Article: The genetics of geometry. *PNAS*, 101(14):4728–4735, 2004.

[21] D. Cohen. Computer simulation of biological pattern generation processes. *Nature*, 216:246–248, Oct 1967.

[22] J. Combaz and F. Neyret. Semi-interactive morphogenesis. In *Shape Modeling International*, page 35, 2006.

[23] R. L. Cook. Stochastic sampling in computer graphics. *ACM Transaction on Graphics*, 5(1):225–240, 1986.

[24] Y. Couder, L. Pauchard, C. Allain, M. Adda-Bedia, and S. Douady. The leaf venation as formed in a tensorial field. *European Physical Journal B*, 28:135–138, 2002.

[25] P. de Reffye, C. Edelin, J. Françon, M. Jaeger, and C. Puech. Plant models faithful to botanical structure and development. In *SIGGRAPH '88: Proceedings of the 15th annual conference on Computer graphics and interactive techniques*, pages 151–158, 1988.

[26] N. Dengler. Personal communication, 2005.

[27] N. Dengler and J. Kang. Vascular patterning and leaf shape. *Current Opinion in Plant Biology*, 4(1):50–56, 2001.

[28] B. Desbenoit, E. Galin, and S. Akkouche. Simulating and modeling lichen growth. *Computer Graphics Forum (Proceedings of Eurographics 2004)*, 23:341–351, September 2004.

[29] P. Dimitrov and S. W. Zucker. A constant production hypothesis guides leaf venation patterning. *Proceedings of the National Academy of Sciences*, 103(24):9363–9368, 2006.

[30] P.M. Donnelly, D. Bonetta, H. Tsukaya, R.E. Dengler, and N.G. Dengler. Cell cycling and cell enlargement in developing leaves of arabidopsis. *Developmental Biology*, 215:407–419(13), 1999.

[31] C. Donner and H. W. Jensen. Light diffusion in multi-layered translucent materials. In *SIGGRAPH '05: ACM SIGGRAPH 2005 Papers*, pages 1032–1039, 2005.

[32] K. Dormer. An investigation of the taxonomic value of shoot structure in angiosperms with special reference to leguminosae. *Annals of Botany*, pages

141–153, 1945.

[33] M Eden. A two-dimensional growth process. In *Fourth Berkeley Symposium on Mathematical Statistics and Probability.*, volume 4, pages 223–239, 1961.

[34] K. Esau. *Vascular Differentiation in Plants.* Holt, Rinehart and Winston, Inc., 1965.

[35] P. Federl and P. Prusinkiewicz. Finite element model of fracture formation on growing surfaces. In *Proceeding of Computational Science. ICCS 2004*, pages 138–145. Springer-Verlag, 2004.

[36] P. Ferraro and C. Godin. A distance measure between plant architectures. *Annals of Forest Science*, 57(5/6):445–461, 2000.

[37] F. Feugier and Y. Iwasa. How canalization can make loops: A new model of reticulated leaf vascular pattern formation. *Journal of Theoretical Biology*, 243:235–244, 2006.

[38] F. Feugier, A. Mochizuki, and Y. Iwasa. Self-organization of the vascular system in plant leaves: Inter-dependent dynamics of auxin flux and carrier proteins. *Journal of Theoretical Biology*, 236:366–375, 2005.

[39] D. R. Fowler, H. Meinhardt, and P. Prusinkiewicz. Modeling seashells. In *SIGGRAPH '92: Proceedings of the 19th annual conference on Computer graphics and interactive techniques*, pages 379–387, 1992.

[40] M. Fuhrer. Hairs, textures, and shades: Improving the realism of plant models generated with l-systems, August 2005.

[41] H. Fujita and A. Mochizuki. The origin of the diversity of leaf venation pattern. *Developmental Dynamics*, 235(10):2710–2721, 2006.

[42] C. Galbraith, P. MacMurchy, and B. Wyvill. BlobTree trees. In *Proceedings of Computer Graphics International 2004*, pages 78–85, 2004.

[43] A Gallois, J.C. Audran, and M. Burrus. Assessment of genetic relationships and popularion discrimination among fagus sylvatica l. by rapd. *Theoretical and Applied Genetics*, 97(1-2):211–219, 1998.

[44] H. Gautier, R. Měch, P. Prusinkiewicz, and C. Varlet-Grancher. 3D architectural modeling of aerial photomorphogenesis in white clover (*Trifolium repens* L.) using L-systems. *Annals of Botany*, (85):359–370, 2000.

[45] A. Gierer. Directional cues for growing axons forming the retinotectal projection. *Development*, 101:479–489, 1987.

[46] G. Girolami. Relation between phyllotaxis and primary vascular organization in linum. *The American Journal of Botany*, 40(8):618–625, 1953.

[47] J. Gomes, L. Darsa, B. Costa, and L. Velho. *Warping and morphing of graphical objects.* Morgan Kaufmann, San Francisco, 1999.

[48] M. E. Gottlieb. The VT model: A deterministic model of angiogenesis. In *IEEE Transactions on Biomedical Engineering*, pages 38–39, 1991.

[49] M. E. Gottlieb. Angiogenesis and vascular networks: complex anatomies from deterministic non-linear physiologies. In J. M. Garcia-Ruiz, E. Louis,

P. Meakin, and L. M. Sander, editors, *Growth patterns in physical sciences and biology*, pages 267–276. Plenum Press, New York, 1993.

[50] N. Greene. Detailing tree skeletons with voxel automata. In *SIGGRAPH '91 Course Notes on Photorealistic Volume Modeling and Rendering Techniques*, 1991.

[51] J. E. Gunckel and R. H. Wetmore. Studies of development in long shoots and short shoots of ginkgo biloba l. ii. phyllotaxis and the organization of the primary vascular system; primary phloem and primary xylem. *American Journal of Botany*, 33(6):532–543, jun 1946.

[52] F. Hallé, R. A. A. Oldeman, and P. B. Tomlinson. *Tropical trees and forests: An architectural analysis.* Springer-Verlag, Berlin, 1978.

[53] M. Hammel and P. Prusinkiewicz. Visualization of developmental processes by extrusion in space-time. In *GI '96: Proceedings of the conference on Graphics interface '96*, pages 246–258, 1996.

[54] P. Hanrahan and W. Krueger. Reflection from layered surfaces due to subsurface scattering. In *Proceedings of SIGGRAPH 1993*, pages 165–174, 1993.

[55] A. J. Hanson. Quaternion Gauss maps and optimal framings of curves and surfaces. Technical Report 518, Computer Science Department, Indiana University, Bloomington, IN, 1998. Available at: http://www.cs.indiana.edu/ftp/techreports/TR518.html.

[56] Z. Hejnowicz and J. Romberger. Growth tensor of plant organs. *Journal of Theoretical Biology*, 110:93–114, 1984.

[57] L. I. Held. *Models for embryonic periodicity*, volume 24 of *Monographs in Developmental Biology*. Karger, Basel, 1994.

[58] L. Hickey. A revised classification of the architecture of dicotyledonous leaves. In C. R. Metcalfe and L. Chalk, editors, *Anatomy of the dicotyledons. Second Edition, Vol. 1*, pages 25–39. Clarendon Press, Oxford, 1979.

[59] Bert Hickman, 2005. `http://www.teslamania.com`.

[60] H. Honda. Description of the form of trees by the parameters of the tree-like body: Effects of the branching angle and the branch length on the shape of the tree-like body. *Journal of Theoretical Biology*, 31:331–338, 1971.

[61] Honda, H., Tomlinson, P. B., and Fisher, J. B. Computer simulation of branch interaction and regulation by unequal flow rates in botanical trees. *American Journal of Botany*, 68(4):569–585, apr 1981.

[62] S. M. Hong, B. Simpson, and G. V.G. Baranoski. Interactive venation-based leaf shape modeling. *Computer Animation and Virtual Worlds*, 16(3-4):415–427, 2005.

[63] Scott Hotton, Valerie Johnson, Jessica Wilbarger, Kajetan Zwieniecki, Pau Atela, Christophe Gole, and Jacques Dumais. The possible and the actual in phyllotaxis: Bridging the gap between empirical observations and iterative models. *Journal of Plant Growth Regulation*, 25:313–323(11), December 2006.

[64] S.-M. Hu and J. Wallner. A second order algorithm for orthogonal projection onto curves and surfaces. *Comput. Aided Geom. Des.*, 22(3):251–260, 2005.

[65] T. Ijiri, S. Owada, and T. Igarashi. The sketch L-system: Global control of tree modeling using free-form strokes. In *Smart Graphics*, pages 138–146, 2006.

[66] Janssen, J. M. and Lindenmayer, A. Models for the control of branch positions and flowering sequences of capitula in mycelis muralis (l.) dumont (compositae). *New Phytologist*, 105(2):191–220, feb 1987.

[67] J. W. Jaromczyk and G. T. Toussaint. Relative neighborhood graphs and their relatives. *Proceedings of the IEEE*, 80(9):1502–1517, 1992.

[68] R. V. Jean. The hierarchical control of phyllotaxis. *Annals of Botany*, pages 747–760, 1982.

[69] H. W. Jensen. Dali rendering system, 2004. Software.

[70] C. Jirasek, P. Prusinkiewicz, and B. Moulia. Integrating biomechanics into developmental plant models expressed using L-systems. In H.-Ch. Spatz and T. Speck, editors, *Plant biomechanics 2000*, pages 615–624. Georg Thieme Verlag, Stuttgart, 2000.

[71] H. Jonsson, M. G. Heisler, B. E. Shapiro, E. M. Meyerowitz, and E. Mjolsness. An auxin-driven polarized transport model for phyllotaxis. *Proceedings of the National Academy of Sciences*, 103(5):1633–1638, 2006.

[72] W. W. Judd, C. S. Campbell, E. A. Kellogg, and P. F. Stevens. *Plant systematics: A phylogenetic approach.* Sinauer Associates, Sunderland, MA, 1999.

[73] J. A. Kaandorp. *Fractal Modeling: Growth and Form in Biology.* Springer-Verlag New York, Inc., New York, NY, USA, 1994.

[74] J. A. Kaandorp and J. Kübler. *The algorithmic beauty of seaweeds, sponges, and corals.* Springer-Verlag New York, Inc., New York, NY, USA, 2001.

[75] J Kang and N Dengler. Vein pattern development in adult leaves of arabidopsis thaliana. *International Journal of Plant Sciences*, (2):231–242, 2004.

[76] J Kang, J Tang, P Donnely, and N Dengler. Primary vascular pattern and expression of athb-8 in shoots of arabidopsis. *New Phytol*, pages 443–454, 2003.

[77] R. Karwowski and P. Prusinkiewicz. Design and implementation of the L+C modeling language. *Electronic Notes in Theoretical Compute Science*, 86(2):134–152, 2003.

[78] N.C. Kenkel. Spatial competition models for plant populations. *Coenoses*, 5:149–158, 1990.

[79] T. Kim and M. C. Lin. Physically based animation and rendering of lightning. In *PG '04: Proceedings of the Computer Graphics and Applications, 12th Pacific Conference on (PG'04)*, pages 267–275. IEEE Computer Society, 2004.

[80] T. Kim, J. Sewall, A. Sud, and M. C. Lin. Fast simulation of laplacian growth. *IEEE Computer Graphics and Applications*, 27(2):68–76, 2007.

[81] D. Kwiatkowska and J. Florek-Marwitz. Ontogenetic variation of phyllotaxis and apex geometry in vegetative shoots of sedum maximum. *Acta societatis botanicorum Poloniae*, 68(2):85–95, 1999.

[82] Dorota Kwiatkowska and Jacques Dumais. Growth and morphogenesis at the vegetative shoot apex of Anagallis arvensis L. *J. Exp. Bot.*, 54(387):1585–1595, 2003.

[83] B. Lane and P. Prusinkiewicz. Generating spatial distributions for multilevel models of plant communities. In *Proceedings of Graphics Interface*, pages 69–80, 2002.

[84] P. R. Larson. Development and organization of the primary vascular system in populus deltoides according to phyllotaxy. *American Journal of Botany*, 62(10):1084–1099, 1975.

[85] L. Leopold. Trees and streams: The efficientcy of branching patterns. *Journal of theoretical Biology*, 31(2):339–354, 1970.

[86] A. Lindenmayer. Mathematical models for cellular interaction in development. *Journal of Theoretical Biology*, 18:280–315, 1968.

[87] A. Lindenmayer. An introduction to parallel map generating systems. In *Proceedings of the 3rd International Workshop on Graph-Grammars and Their Application to Computer Science*, pages 27–40, London, UK, 1987. Springer-Verlag.

[88] B. Lintermann and O. Deussen. Interactive modeling of plants. *IEEE Computer Graphics and Applications*, 19(1):56–65, 1999.

[89] J. Lluch, R. Vivo, and C. Monserrat. Modeling tree structures using a single polygonal mesh. *Graphical Models*, 66(2):89–101, March 2004.

[90] S. Lu, C. Zhao, X. Guo, and C. Li. Modeling curled leaves. In *Fourth International Conference on Image and Graphics*, pages 909–914, 2007.

[91] Y. Liang Ma and W. T. Hewitt. Point inversion and projection for nurbs curve and surface: control polygon approach. *Comput. Aided Geom. Des.*, 20(2):79–99, 2003.

[92] N. MacDonald. *Trees and networks in biological models.* J. Wiley & Sons, New York, 1983.

[93] Peter MacMurchy. The use of subdivision surfaces in the modeling of plants, August 2004.

[94] M. Matsushita and H. Fujikawa. Diffusion-limited growth in bacterial colony formation. *Physica A Statistical Mechanics and its Applications*, 168:498–506, September 1990.

[95] M. Matthews. *Physically Based Simulation of Growing Surfaces.* PhD thesis, University of Calgary, Canada, December 2002.

[96] M. W. McGahan. Vascular differentiation in the vegetative shoot of xanthium chinense. *American Journal of Botany*, 42(2):132–140, feb 1955.

[97] P. Meakin. A new model for biological pattern formation. *Journal of Theoretical Biology*, 118:101–113, 1986.

[98] P. Meakin. *Growth Patterns in Physical Sciences and Biology*, chapter Adaptive cluster growth models, pages 233–244. Plenum press, 1991.

[99] P. Meakin, J. Feder, and T. Jøssang. Growth of adaptive networks in a modified diffusion-limited-aggregation model. *Phys. Rev. A*, 44(8):5104–5110, Oct 1991.

[100] H. Meinhardt. Morphogenesis of lines and nets. *Differentiation*, 6:117–123, 1976.

[101] H. Meinhardt. *Models of Biological Pattern Formation*, pages 172–185. Academic Press Inc, 1982.

[102] H. Meinhardt. *Algorithmic Beauty of Sea Shells*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2003. Illustrator-D. R. Fowler and Illustrator-P. Prusinkiewicz.

[103] R. M.H. Merks, Y. Van de peer, D. Inze, and G. T.S. Beemster. Canalization without flux sensors: a traveling-wave hypothesis. *Trends in Plant Science*, 12(9):384–390, Sep.

[104] D. Mitchell. Generating antialiased images at low sampling densities. *Computer Graphics (SIGGRAPH '87 Conference Proceedings)*, 21(4):65–78, 1987.

[105] G. J. Mitchison. A model for vein formation in higher plants. *Proc. R. Soc. London B*, 207:79–109, 1980.

[106] L. Mündermann, Y. Erasmus, B. Lane, E. Coen, and P. Prusinkiewicz. Quantitative Modeling of Arabidopsis Development. *Plant Physiol.*, 139(2):960–968, 2005.

[107] C. D. Murray. The physiological principle of minimum work. *Proceedings of the National Academy of Sciences*, 12:207–214, 1926.

[108] C. D. Murray. A relationship between circumference and weight in trees and its bearing on branching angles. *J. Gen. Physiol.*, 10(5):725–729, 1927.

[109] R. Měch and P. Prusinkiewicz. Visual models of plants interacting with their environment. *Computer Graphics (SIGGRAPH '96 Conference Proceedings)*, 30:397–410, 1996.

[110] B. Neubert, T. Franken, and O. Deussen. Approximate image-based tree-modeling using particle flows. *ACM Transactions on Graphics*, 26(3):88, 2007.

[111] L. Niemeyer, L. Pietronero, and H. J. Wiesmann. Fractal dimension of dielectric breakdown. *Phys. Rev. Lett.*, 52(12):1033–1036, Mar 1984.

[112] K. J. Niklas. A mechanical perspective on foliage leaf form and function. *New Phytologist*, 143(1):19–31, 1999.

[113] A. Novoplansky, D. Cohen, and T. Sachs. How porulaca seedlins avoid their neighbours. *Oecologia*, 82(4):490–493, 1990.

[114] A. Okabe, B. Boots, and K. Sugihara. *Spatial tesselations: Concepts and applications of Voronoi diagrams.* J. Wiley and Sons, Chichester, 1992.

[115] M. Okabe, S. Owada, and T. Igarashi. Interactive design of botanical trees using freehand sketches and example-based editing. *Computer Graphics Forum*, 24(3):487–496, 2005. Proceedings of Eurographics '05.

[116] P. Oppenheimer. Real time design and animation of fractal plants and trees. *Computer Graphics (SIGGRAPH '86 Conference Proceedings)*, 20(4):55–64, 1986.

[117] W. R. Philipson and E. E. Balfour. Vascular patterns in dicotyledons. *Botanical Review*, pages 382–404, 1963.

[118] S. Pion. Computer Geometry Algorithms Library, 2007. `http://www.cgal.org`.

[119] S. Popinet. The GNU triangulated surface library, 2004. `http://gts.sourceforge.net`.

[120] T. Pray. Foliar venation of angiosperms, ii, histogenesis of the venation of liriodendron. *American Journay of Botany*, 42(1):18–27, 1955.

[121] F. P. Preparata and M. I. Shamos. *Computational geometry: an introduction.* Springer-Verlag New York, Inc., New York, NY, USA, 1985.

[122] P. Prusinkiewicz. Visual models of morphogenesis. *Artificial Life*, 1(1/2):61–74, 1994.

[123] P. Prusinkiewicz. In search of the right abstraction: The synergy between art, science, and information technology in the modeling of natural phenomena. *Art @ Science*, pages 60–68, 1998.

[124] P. Prusinkiewicz. *Pattern Formation in Biology, Vision and Dynamics*, chapter Paradigms of pattern formation: towards a computational theory of morphogenesis, pages 3–23. World Scientific Publishing Company, 1999.

[125] P. Prusinkiewicz, M. Hammel, and E. Mjolsness. Animation of plant development. In *Proceedings of SIGGRAPH 1993*, pages 351–360, 1993.

[126] P. Prusinkiewicz and J. Hanan. Visualization of botanical structures and processes using parametric l-systems. pages 183–201, 1990.

[127] P. Prusinkiewicz, M. James, and R. Měch. Synthetic topiary. In *SIGGRAPH '94: Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 351–358, New York, NY, USA, 1994. ACM Press.

[128] P. Prusinkiewicz and A. Lindenmayer. *The algorithmic beauty of plants*. Springer-Verlag New York, Inc., New York, NY, USA, 1990.

[129] P. Prusinkiewicz, L. Mündermann, R. Karwowski, and B. Lane. The use of positional information in the modeling of plants, 2001. Proceedings of SIGGRAPH 2001, ACM SIGGRAPH, 2001, pp. 289–300.

[130] P. Prusinkiewicz, F. Samavati, C. Smith, and R. Karwowski. L-system description of subdivision curves. *International Journal of Shape Modeling*, 9(1):41–59, 2003.

[131] W. T. Reeves and R. Blau. Approximate and probabilistic algorithms for shading and rendering structured particle systems. *Computer Graphics (SIGGRAPH '85 Conference Proceedings)*, 19(3):313–322, July 1985.

[132] D. Reinhardt, E.R. Pesce, P. Stieger, T. Mandel, K. Baltensperger, M. Bennett, J. Traas, J. Friml, and C. Kuhlemeier. Regulation of phyllotaxis by polar auxin transport. *Nature*, 426:255–260, 2003.

[133] W. Remphrey. Woody plants in the praire landscape, 1999. CD-ROM, Remphrey Botanical Publications, St. Norbert.

[134] W Remphrey and L. Pearn. Crown development of a clone of populus tremuloides exhibiting "crooked" architecture and a comparison with wild-type trees. *Canadian Journal of Botany*, 81(4):345–359, 2003.

[135] J. Roberts. Sticky Pixels: Evolutionary Growth by Random Drop Ballistic Aggregation. In *Eurographics UK 2001 Conference Proceedings*, pages 149–155, April 2001.

[136] Y. Rodkaew, P. Chongstitvatana, S. Siripant, and C. Lursinsap. Particle systems for plant modeling. In B.-G. Hu and M. Jaeger, editors, *Plant growth modeling and applications. Proceedings of PMA03*, pages 210–217. Tsinghua University Press and Springer, Beijing, 2003.

[137] Y. Rodkaew, S. Siripant, C. Lursinsap, and P. Chongstitvatana. An algorithm for generating vein images for realistic modeling of a leaf. In *Prodeedings of the International Conference on Computational Mathematics and Modeling*, page 9 pp., Bangkok, Thailand, May 22–24 2002.

[138] A.-G. Rolland, J. A. Bangham, and E. Coen. Growth dynamics underlying petal shape and asymmetry. *Nature*, 422:161–163, 2003.

[139] A.-G. Rolland-Lagan, P. Federl, and P. Prusinkiewicz. Reviewing models of auxin canalisation in the context of vein pattern formation in *Arabidopsis* leaves. In *Proceedings of the 4th International Workshop on Functional-Structural Plant Models*, pages 376–381, 2004.

[140] A.-G. Rolland-Lagan and P. Prusinkiewicz. Reviewing models of auxin canalization in the context of leaf vein pattern formation in arabidopsis. *The Plant Journal*, 44(5):854–865, 2005.

[141] A. Roth-Nebelsick, D. Uhl, V. Mosbugger, and H. Kerp. Evolution and function of leaf venation architecture: a review. *Annals of Botany*, 87:553–566, 2001.

[142] A. Runions, M. Fuhrer, B. Lane, P. Federl, A. Rolland-Lagan, and P. Prusinkiewicz. Modeling and visualization of leaf venation patterns. *ACM Trans. Graph.*, 24(3):702–711, 2005.

[143] A. Runions, B. Lane, and P. Prusinkiewicz. Modeling trees with a space colonization algorithm. In *Proceedings of the Eurographics Workshop on Natural Phenomena*, pages 63–70, 2007.

[144] T Sachs. Patterned differentiation in plants. *Differentiation*, 11(65), 1978.

[145] T. Sachs. The control of patterned differentation of vascular tissues. In H. W. Woolhouse, editor, *Advances in botanical research*, volume 6, pages 152–262. Academic Press, London, 1981.

[146] T. Sachs. Collective specification of cellular development. *BioEssays*, 25.9:897–903, 2003.

[147] T. Sachs and A. Novoplansky. Tree form: Architectural models do not suffice. *Israel Journal of Plant Sciences*, 43:203–212, 1995.

[148] M. Sauer, J. Balla, C. Luschnig, J. Wisniewska, V. Reinohl, J. Friml, and E. Benkova. Canalization of auxin flow by Aux/IAA-ARF-dependent feedback regulation of PIN polarity. *Genes Dev.*, 20(20):2902–2911, 2006.

[149] E. Scarpella, P. Francis, and T. Berleth. Stage-specific markers define early steps of procambium development in *Arabidopsis* leaves and correlate termination of vein formation with mesophyll differentiation. *Development*, 131:3445–3455, 2004.

[150] E. Scarpella, D. Marcos, J. Friml, and T. Berleth. Control of leaf vascular patterning by polar auxin transport. *Genes and Development*, 20:1015–1027, 2006.

[151] H. Scholten and A. Lindenmayer. A mathematical model for the laminar development of simple leaves. In W. van Cotthem, editor, *Morphologie-Anatomie und Systematic der Pflanzen 5*, pages 29–37. Waegeman, Ninove, Belgium, 1981.

[152] K. Shinozaki, K. Yoda, K. Hozumi, and T. Kira. A quantitative analysis of plant form, the pipe model theory i. basic anyalysis. *Japanese Journal of Ecology*, 14(3):97–105, 1964.

[153] L. E. Sieburth. Auxin is required for leaf vein pattern in *Arabidopsis*. *Plant Physiology*, 121:1179–1190, 1999.

[154] C. Smith. *On Vertex-Vertex Systems and Their Use in Geometric and Biological Modelling.* PhD thesis, University of Calgary, Canada, January 2006.

[155] C. Smith, P. Prusinkiewicz, and F. Samavati. Local specification of surface subdivision algorithms. In *Applications of Graph Transformations with Industrial Relevance*, pages 313–327, 2004. Lecture Notes in Computer Science 3062.

[156] R. Smith. *Simulation Models of Phyllotaxis and Morphogenesis in Plants.* PhD thesis, University of Calgary, Calgary, Alberta, September 2007.

[157] R. S. Smith, S. Guyomarc'h, T. Mandel, D. Reinhardt, C. Kuhlemeier, and P. Prusinkiewicz. A plausible model of phyllotaxis. *Proceedings of the National Academy of Sciences*, 103(5):1301–1306, 2006.

[158] R. S. Smith, C. Kuhlemeier, and P. Prusinkiewicz. Inhibition fields for phyllotactic pattern formation: a simulation study. 84(11):1635–1649, 2006.

[159] P. S. Stevens. *Patterns in nature.* Little, Brown and Co., Boston, 1974.

[160] Q. J. Steynen and E. A. Schultz. The FORKED genes are essential for distal vein meeting in Arabidopsis. *Development*, 130(19):4695–4708, 2003.

[161] L. Streit, P. Federl, and M.C. Sousa. Modelling plant variation through growth. *Comput. Graph. Forum*, 24(3):497–506, 2005.

[162] A. Takenaka. A simulation model of tree architecture development based on growth response to local light environment. *Journal of Plant Research*, 107:321–330, 1994.

[163] G. T. Toussaint. The relative neighborhood graph of a finite planar set. *Pattern Recognition*, 12:261–268, 1980.

[164] G.T. Toussaint. Computing largest empty circles with location constraints. *International Journal of Parallel Programming*, 12(5):347, 1983.

[165] S. C. Tucker. Phyllotaxis and vascular organization of the carpels in michelia fuscata. *American Journal of Botany*, pages 60–71, 1960.

[166] A.M. Turing. The Chemical Basis of Morhpogenesis. *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences*, 237(641):37–72, August 1952.

[167] G. Turk. Generating textures on arbitrary surfaces using reaction-diffusion. In *SIGGRAPH '91: Proceedings of the 18th annual conference on Computer graphics and interactive techniques*, pages 289–298, New York, NY, USA, 1991. ACM Press.

[168] S. Ulam. On some mathematical properties connected with patterns of growth of figures. *Proceedings of Symposia on Applied Mathematics*, 14:215–224, 1962.

[169] R. B. Urquhart. Algorithms for computation of relative neighbourhood graph. *Electronics Letters*, 16(14):556–557, 1980.

[170] M. Walter, A. Fournier, and M. Reimers. Clonal mosaic model for the synthesis of mammalian coat patterns. In *Proceedings of Graphics Interface '98*, pages 82–91, 1998.

[171] I. Wang, J. Wan, and G. Baranoski. Physically-based simulation of plant leaf growth. *Computer Animation and Virtual Worlds*, 15(3–4):237–244, 2004.

[172] L. Wang, W. Wang, J. Dorsey, X. Yang, B. Guo, and H. Shum. Real-time rendering of plant leaves. *ACM Trans. Graph.*, 24(3):712–719, 2005.

[173] D. F. Watson. Computing the n-dimensional Delaunay tessellation with application to Voronoi polytopes. *The Computer Journal*, 24(2):167–172, 1981.

[174] M.A. Watson and B.B. Casper. Morphogenetic constraints on patterns of carbon distribution in plants. *Annual Review of Ecology and Systematics*, 15:233–258, 1984.

[175] J. Weber and J. Penn. Creation and rendering of realistic trees. *Computer Graphics (SIGGRAPH '95 Conference Proceedings)*, 29:119–128, 1996.

[176] B. F. Wilson. *The growing tree.* The University of Massachusetts Press, Amherst, 1984.

[177] T. A. Witten and L. M. Sander. Diffusion-limited aggregation, a kinetic critical phenomenon. *Phys. Rev. Lett.*, 47(19):1400–1403, Nov 1981.

[178] L. Wolpert. Positional information and the spatial pattern of cellular differentiation. *Journal of Theoretical Biology*, 25:1–47, 1969.

[179] L. Wolpert. Positional information and pattern formation in development. *Developmental Genetics*, 15(6):485–490, 1994.

[180] Wolpert, L. and Dover, G. Positional information and pattern formation [and discussion]. *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences*, 295(1078):441–450, oct 1981.

[181] M. T. Wong, D. E. Zongker, and D. H. Salesin. Computer-generated floral ornament. In *SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 423–434, 1998.

[182] M. N. Zakaria, S. Rokhmah, and M. Shukri. A sketch-and-spray interface for modeling trees. In *Smart Graphics 2007*, pages 23–35, 2007.

Appendix A

Proof of Theorem 1

Theorem 1. *Let* $\mathbf{v} \in \Re^n$ *and* $S(\mathbf{v})$ *be a finite set of points in* $\Re^n$*, let* $E : \Re^n \to \Re$ *be the scalar field defined as the sum of the Euclidean distances between* $\mathbf{v}$ *and each point in* $S(\mathbf{v})$*:*

$$E = \sum_{\mathbf{s} \in S(\mathbf{v})} \|\mathbf{s} - \mathbf{v}\|$$

Then

$$\vec{n} = \sum_{\mathbf{s} \in S(\mathbf{v})} \frac{\mathbf{s} - \mathbf{v}}{\|\mathbf{s} - \mathbf{v}\|}$$

is equal to $-\nabla E$*, the direction in which* E *decreases fastest.*

This property holds in any finite-dimensional Euclidean space. As such the proof below shows the theorem to be true in $\Re^n$, of which $\Re^3$ and $\Re^2$ are special cases.

Proof. Let $E : \Re^n \to \Re$ be as defined above, $\mathbf{v} \in \Re^n$, and $S(\mathbf{v})$ is a finite set of points in $\Re^n$:

$$\mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}, \mathbf{s} \in S(\mathbf{v}) \text{ is } \mathbf{s} = \begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_n \end{bmatrix}, \text{ and } \nabla E(\mathbf{v}) = \begin{bmatrix} \frac{\partial E}{\partial v_1} \\ \frac{\partial E}{\partial v_2} \\ \vdots \\ \frac{\partial E}{\partial v_n} \end{bmatrix}$$

We want to show that $\vec{n} = -\nabla E(\mathbf{v})$, so let us examine $E(\mathbf{v})$:

$$E(\mathbf{v}) = \sum_{\mathbf{s} \in S(\mathbf{v})} \|\mathbf{s} - \mathbf{v}\| = \sum_{\mathbf{s} \in S(\mathbf{v})} \left(\sum_{j=1}^{n} (s_j - v_j)^2 \right)^{\frac{1}{2}}$$

Thus for each $i \in \{1, 2, \cdots, n\}$:

$$\begin{aligned}
\frac{\partial E}{\partial v_i} &= \sum_{s \in S(\mathbf{v})} (\frac{1}{2})(2)(-1)(s_i - v_i) \left(\sum_{j=1}^{n} (s_j - v_j)^2 \right)^{-\frac{1}{2}} \\
&= \sum_{\mathbf{s} \in S(\mathbf{v})} - \frac{s_i - v_i}{\left(\sum_{j=1}^{n} (s_j - v_j)^2 \right)^{\frac{1}{2}}} \\
&= - \sum_{\mathbf{s} \in S(\mathbf{v})} \frac{s_i - v_i}{\|\mathbf{s} - \mathbf{v}\|}
\end{aligned}$$

And as this holds for $\forall i \in \{1, \cdots, n\}$ we can write $\nabla E(\mathbf{v})$ as:

$$\begin{aligned}
\nabla E(\mathbf{v}) &= - \sum_{\mathbf{s} \in S(\mathbf{v})} \begin{bmatrix} s_1 - v_1 \\ s_2 - v_2 \\ \vdots \\ s_n - v_n \end{bmatrix} \frac{1}{\|\mathbf{s} - \mathbf{v}\|} \\
&= - \sum_{\mathbf{s} \in S(\mathbf{v})} \frac{\mathbf{s} - \mathbf{v}}{\|\mathbf{s} - \mathbf{v}\|} = -\vec{n}
\end{aligned}$$

Hence, it follows that $\vec{n} = -\nabla E(\mathbf{v})$.

$\square$