

THE UNIVERSITY OF CALGARY

Models of Plant Communities for Image Synthesis

by

Brendan Lane

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

JUNE, 2002

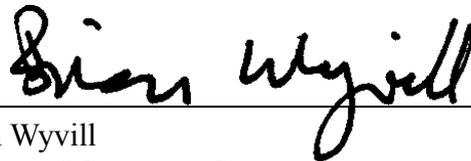
© Brendan Lane 2002

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 "Models of Plant Communities for Image Synthesis" submitted by Brendan Lane in partial fulfillment of the requirements for the degree of Master of Science.



Chairman, Dr. Przemyslaw Prusinkiewicz
Department of Computer Science



Dr. Brian Wyvill
Department of Computer Science



Dr. Richard M. Levy
Faculty of Environmental Design

2002 - 06 - 24

Date

ABSTRACT

In this thesis I examine the problem of modeling and visualizing large groups of plants. The extreme visual complexity of such scenes can be captured using multilevel models, which combine a hierarchy of models at different levels of abstraction. The modeling of plant communities is done by two-level models, where the higher-level model describes the spatial distribution of plants, and the lower-level model describes individual plants' shapes.

Models of plant communities can be divided based on the direction of information flow: local-to-global models are rooted in individual-based ecosystem simulations, while the properties of individuals in global-to-local models are inferred from a given distribution of plant densities. Multiset L-systems are introduced as a formalism for local-to-global models of the spatial distribution of plants. A new global-to-local model is developed which uses the idea of iterative deformation of a probability density.

Examples of both local-to-global and global-to-local models of the spatial distribution of plants are demonstrated which exhibit various ecological properties, including succession, clustering, and self-thinning. These examples are coupled to a low-level model of individual plants to create realistic visualizations of the plant community.

PREFACE

It almost goes without saying that this research did not occur in a vacuum. Much of this work (and of Chapters 4 and 5 in particular) was published in the paper “Generating spatial distributions for multilevel models of plant communities”, by myself and Dr. Przemek Prusinkiewicz, in the proceedings of Graphics Interface 2002. Chapter 6 is largely based on the work I produced for Section 8 of “The use of positional information in the modeling of plants” by Dr. Prusinkiewicz, Lars Mündermann, Radek Karwowski, and myself, in the proceedings of SIGGRAPH 2001.

Most of the individual plants were modeled using the methods described in Chapter 6. Some of the models, however, were manually created. The grass model used in all of the rendered images was developed by Lars Mündermann. Lars also produced the daisy models used in Figures 5.5 and 5.6. The model of fireweed used in Figure 4.5 was initially developed by Martin Fuhrer. The terrain models were created with Peter MacMurchy’s SaLaD modeling system.

WORD OF THANKS

Throughout my time as a graduate student, many people have supported me both personally and academically. I cannot possibly name them all here, but I would like to single out special thanks to a few.

First, thanks to the members of the Graphics Jungle research group, in particular Pavol Federl, Mark Fox, Callum Galbraith, Pauline Jepp, Radek Karwowski, Peter MacMurchy, Mark Matthews, Lars Mündermann, and Ryan Schmidt, for letting me blather on about probability density functions, spherical calculus, and God knows what else whenever they were trying to work. Thanks also to the members of the CSUS, without whom this thesis would have been finished a year ago....

A great thanks to my professors, especially Drs. Ted Bisztriczky, Richard Cleve, and Jim Parker; and, of course, to my supervisor, Dr. Przemek Prusinkiewicz. Dr. P's knowledge is tremendous, and his enthusiasm infectious. Several times I explained to him my own muddled synthesis of a concept, and with a few words he made me look at the whole field in a new light. Much of the compactness of this thesis is due to our clarifying conversations. I also thank Drs. Richard Levy and Brian Wyvill for making the examination process surprisingly pleasant.

Finally, much of the credit has to go to my family, and most especially my parents, who put up with their "full-time student" for far too long, and still supported me right to the end. To them I owe my eternal thanks.

TABLE OF CONTENTS

Approval Page	ii
Abstract	iii
Preface	iv
Word of Thanks	v
Table of Contents	vi
List of Algorithms	viii
List of Tables	ix
List of Figures	x
1 Introduction	1
1.1 Statement of problem	1
1.2 Contributions of this research	2
1.3 Organization of the thesis	4
2 Ecological phenomena of plant communities	5
2.1 Succession	6
2.2 Thinning and self-thinning	7
2.3 Size hierarchies	9
2.4 Overdispersion and clustering	9
3 Individual-based models of plant communities: A review	13
3.1 Local-to-global models	15
3.2 Global-to-local models	21
3.3 Multilevel modeling	23
4 Local-to-global models of plant communities	24
4.1 L-systems	24
4.2 Multiset L-systems	27
4.3 The self-thinning model	29
4.4 Plant succession	30
4.5 Clustering through plant propagation	33

4.6	Conclusion	36
5	Global-to-local models of plant communities	38
5.1	The deformation kernel method	38
5.2	Implementation of the deformation kernel method	42
5.3	Results	46
5.4	Extensions	49
6	Multilevel modeling of plant communities	53
7	Conclusions	58
	Bibliography	61

LIST OF ALGORITHMS

4.1	L-system illustrating axiom and productions	25
4.2	L-system illustrating parameters	26
4.3	Example of a multiset L-system	28
4.4	The self-thinning model described in [DHL ⁺ 98]	29
4.5	A model of interspecies competition	32
4.6	A model of propagation	35
5.1	Naïve implementation of the deformation kernel method	44
5.2	More efficient implementation of the deformation kernel method	45
6.1	Generation of a tree from its silhouette curve	53

LIST OF TABLES

4.1	Operation of a sample multiset L-system	28
7.1	Statistics pertinent to scenes in this thesis	59

LIST OF FIGURES

2.1	The self-thinning curve	8
2.2	Size distribution in a size hierarchy	9
2.3	How the Hopkins index is calculated	11
2.4	The effect of clustering on plant distribution	12
3.1	Diagrammatic representation of the self-thinning model from [DHL ⁺ 98] .	21
4.1	The execution of L-system 4.1	25
4.2	The execution of L-system 4.2	26
4.3	Three stages of simulation of the self-thinning process	30
4.4	Self-thinning curve generated by Algorithm 4.4	31
4.5	Tree succession model simulated using L-system 4.5	34
4.6	Three stages of the evolution of the clustering model	37
5.1	The effect a plant has on the probability of finding neighbouring plants. .	39
5.2	Examples of deformation kernels	39
5.3	An example of the deformation kernel algorithm	43
5.4	Some kernels and the point patterns they generate	46
5.5	Point patterns rendered as fields of daisies	47
5.6	Plant distributions created from a user-defined density map	48
5.7	The concept of the kernel matrix for two species	49
5.8	An example of the deformation kernel algorithm with two species	51
5.9	Example of the multispecies kernel method	52
6.1	Some trees and their defining silhouette curves	54
6.2	Comparison between silhouette shapes and realistic tree models	57

CHAPTER 1

INTRODUCTION

1.1 Statement of problem

One of the dominant threads in computer graphics is the recreation of natural phenomena. Some early work [Gar85, MWC80, Max81] recreated the *appearance* of nature by artificial mechanisms. Other authors [FR86, Rey87] attempted to recreate the actual *processes* behind the phenomena. Both approaches are still used, modeling many types of natural phenomena, from social [PM01] to meteorological [DKY⁺00, Fea00].

These efforts, especially the ones based on reproducing the processes behind the phenomena, are made difficult by the extreme complexity of nature. Not only do natural systems have many components, but these components interact in a complex manner, sometimes with substantial effect. For instance, the quality of light received by a plant greatly affects its amount, direction, and type of growth [MP96, GMPVG00]. Yet each of the thousands of leaves on a tree reflects light on each of the other leaves. If the modeller is trying to faithfully recreate the processes present in nature, the interactions between all of these leaves must be taken into account. In a forest of a thousand trees, there are millions of leaves, and a million times as many possible interactions to consider.

This thesis is concerned with the particular problem of modeling and visualizing scenes of large numbers of plants. There are many reasons we might want to create a realistic visualization of a plant community. For the purposes of education, a visualization of important phenomena in ecology or of an extinct ecosystem may be created. In the realm of forestry, the models used to predict the results of a decision to cut or replant a patch of

forest can be visualized to provide more comprehensible data for policy makers. In urban planning or architectural design, the interaction of a surrounding ecosystem with human habitation can be predicted and previewed. Finally, for the purposes of art or entertainment, the visualization of an ecosystem based on an underlying ecological model will look realistic, and can be produced with less effort on the artist's part.

Leaves, flowers, and other plant organs are important in the visualization of plants. However, a direct simulation of an entire plant community with these organs as basic modules is, as noted above, extremely complex and requires a great deal of computation. The model cannot be naïvely simplified by taking collections of organs as basic modules, as the organs themselves are important, at least for visualization. Multilevel modeling deals with this problem by replacing a single complex model with a hierarchy of models at a sequence of scales [GGCC97].

A paper from SIGGRAPH 1998 by Deussen et al. [DHL⁺98] describes an application of the multilevel modeling idea to a two-level model of plant ecosystems. A high-level model determines the spatial distribution of the plants, and lower-level models determine the plants' shapes and the positions of the plant organs. The models are coupled so that information created at a higher level can affect the outcome of the model at the lower level. The SIGGRAPH 1998 paper represents the starting point of the work described in this thesis.

1.2 Contributions of this research

The contributions of this research to computer graphics and the ecological modeling of plant communities are as follows. First, it extends to the modeling of plant communi-

ties the systemization of individual-based models described in [PMKL01], dividing such models into two categories based on the direction of information flow: *local-to-global* or *global-to-local*. In local-to-global models, global features emerge from the local interactions of individuals. In global-to-local models, on the other hand, global features are decomposed to instantiate local properties. These two types of models also differ in the type of control the modeler has over the model. In local-to-global models, the parameters are local and thus fitting the model to a global boundary condition is difficult. In global-to-local models, the parameters are global, and fitting to global boundary conditions is considerably simpler.

New methodologies for modeling the spatial distribution of plants are described. The formalism of *multiset L-systems* is introduced as a framework for local-to-global individual-based models of the spatial distributions of plant communities. This extension allows L-systems to model populations of individual organisms which reproduce, interact, and die. The idea of the *deformation kernel* is introduced, and used in a global-to-local method which decomposes an initial density distribution into individual plant positions.

Improved methods for joining higher-level spatial distribution models and lower-level individual plant models are described. Using a global-to-local model of individual plants described by Prusinkiewicz et al. [PMKL01], the plants created by the lower-level model can be of a species, shape, and size defined by the higher-level model.

These modeling methodologies are illustrated using ecologically relevant examples. Local-to-global models are described which display the phenomena of *self-thinning*, *succession*, and *clustering*. The deformation kernel method is shown to allow control of clustering as well as the interaction of multiple species. Finally, the methodologies are applied to image synthesis, creating realistic visualization of these models.

1.3 Organization of the thesis

In Chapter 2 I give general background in the ecological phenomena which are relevant to this thesis. In Chapter 3 I discuss previous work in the area of individual-based modeling of plant distributions.

In Chapter 4 I introduce the formalism of L-systems (Section 4.1) and multiset L-systems (Section 4.2), then use this formalism to present three local-to-global models of plant spatial distributions. These models capture the essence of the ecological phenomena of self-thinning (Section 4.3), plant succession (Section 4.4), and clustering through local propagation (Section 4.5). In Chapter 5 I describe the global-to-local deformation kernel method for modeling the spatial distribution of plants, which allows control over both local density and clustering. In Chapter 6 I show how the high-level models of the spatial distribution of plant communities can be coupled to low-level models of individual plants to produce realistic renderings of the communities.

Finally, in Chapter 7 I analyze the running time and space requirements of the algorithms discussed in the text, present further research directions, and conclude the thesis.

CHAPTER 2

ECOLOGICAL PHENOMENA OF PLANT COMMUNITIES

In this chapter, I discuss some ecological phenomena which are particularly pertinent to the spatial distribution of plant communities. *Succession* (Section 2.1) is a process in which the species makeup of plant communities changes. Competition between individual plants lead to the process of *thinning* (Section 2.2), during which the plants form a marked *hierarchy of sizes* (Section 2.3). Thinning leads to a spatial *overdispersion* of plants; other processes, such as propagation, lead to *clustering* (Section 2.4).

Plants and their environment

The structure and composition of plant communities is a function of their environment. What kinds of plants are present, where they are located, and how they grow is largely dependent on what environment they find themselves in. This environment, in turn, can be decomposed into two parts: the *physical* environment, principally meteorological and geological; and the *biological* environment, predominantly the plants themselves [Per94].

The physical environment includes the amount and quality of sunlight; the speed and prevailing direction of winds; the rainfall and moisture content of the air; the type of soil and its quality; and various landforms such as hills or rivers. All of these qualities affect what plants will grow in the environment: some plant species require direct sunlight, while others need only indirect light; some species thrive only in high-moisture areas, while others will not survive if there is too much moisture. Heterogeneities of this type in the physical environment will affect the structure and composition of the community.

The biological environment also has direct effects on the structure and composition of

the plant community. A tree dropping seeds which grow into new trees clearly changes the community's structure. However, many effects of the biological environment are indirect, and occur because the biological environment changes the physical environment. For instance, plants shade the sun and block the wind, draw moisture and nutrients from the soil, and release oxygen and water vapour into the atmosphere [Har77].

2.1 Succession

The physical and biological environments at any location affect the sorts of plants which will grow there. It is possible for a community with one species composition to create an environment which is more favourable to a different composition of species. This is at the heart of the phenomenon of succession.

Suppose two species of trees are growing in a field. The first grows quickly, but requires direct sunlight; the second tolerates shade, but grows relatively slowly. If there are initially only saplings of both species present, then the first species, because of its faster growth, will come to dominate the field, effectively turning it into a forest largely consisting of the trees of the first species. However, turning the field into a forest creates a shady environment, which is not favourable to the first species. The second species, however, can survive in a shady environment, and it performs better in this environment than the first species. Eventually, the forest consists primarily of trees of the second species.

This is a classic illustration of the phenomenon of succession [Per94]. It can be observed at many scales, with many different sets of species, across all environments capable of supporting life [Per94]. The final stage of succession, which is self-supporting, is called the *climax*. In the two-trees example given above, the forest primarily consisting of trees

of the second species is the climax stage. The properties of the climax stage of succession are largely determined by the physical environment and by what species are available in the area. For instance, the climax stage of a tropical ecosystem is quite different from that of a temperate ecosystem, due largely to differences such as rainfall and temperature.

Once a climax state has become established, it is self-supporting; in this sense, it is the 'final' stage of the ecosystem. In any real ecosystem, however, catastrophic events will eventually change the composition even of climax states. These *disturbances* can be either internal to the ecosystem or come from outside it. External disturbances include human clearcuts or forest fires; internal disturbances include pest infestations or even the death of one particularly dominant tree [Per94]. After a disturbance, the climax state has been disrupted and a new succession process begins.

2.2 Thinning and self-thinning

As a plant grows, so do the effects it has on the environment. It consumes more resources, so that less light and fewer nutrients are available for neighbouring plants. Eventually, as all plants in a community grow and consume more resources, some plants will not have enough resources to survive, and will die. This mortality is defined as *density-dependent*, as it increases as the density of the community increases. Such density-dependent mortality is called *thinning*, and occurs in most plant communities. A plant thinned by plants of a different species is *alien-thinned*; a plant thinned by plants of its own species is *self-thinned* [Har77].

An interesting practical example of a plant community is the even-aged monoculture, a community of a single species of plants all seeded at the same time. Such communities

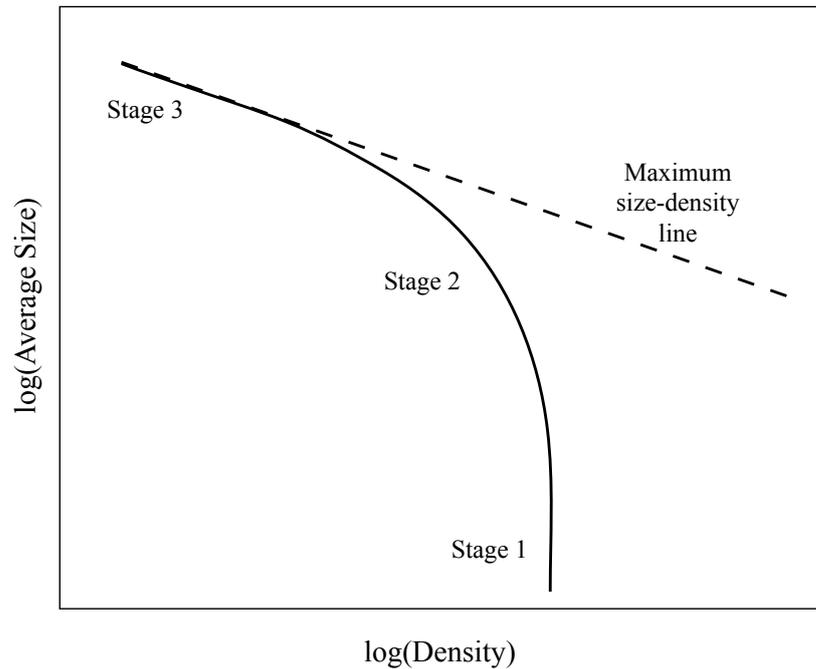


Figure 2.1: The self-thinning curve. The solid line represents the trajectory of the community in state space; the dotted line has a slope of $-3/2$. After [Per94], figure 12.1.

are common in forests raised for cultivation. As these monocultures undergo the process of self-thinning, an interesting phenomenon can be observed [YKOH63]. If the average mass of a plant is plotted against the number of plants per unit of area, the *self-thinning curve* is obtained (Figure 2.1).

In stage 1, the plants grow without interacting; the number of plants, and thus the number of plants per unit of area, does not change. In stage 2, the plants begin to interact; many of the plants die off in competition for resources. Finally, in stage 3, the community reaches the *self-thinning line*, which has a slope of $-3/2$. The state of the community moves along this line until all of the mature plants die. This self-thinning line is found in monocultures of many plant species, both woody and herbaceous [Per94, Har77].

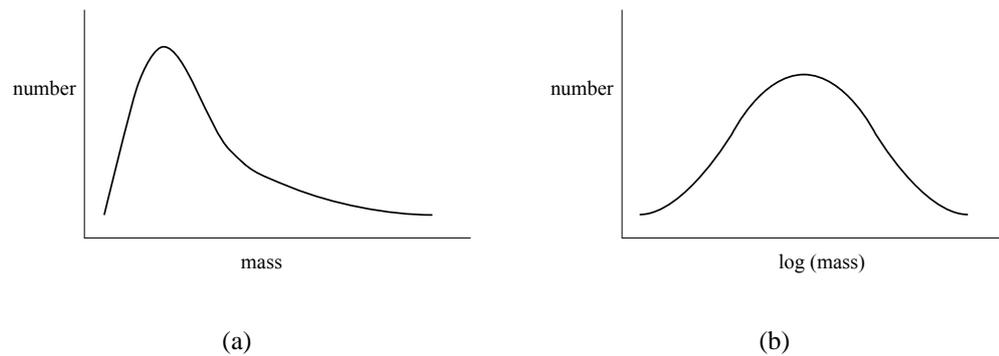


Figure 2.2: Size distribution in a size hierarchy.

2.3 Size hierarchies

Thinning also results in another interesting pattern in the community distribution. If the size distribution of natural plant communities is recorded, a *size hierarchy* [Har77] is observed (Figure 2.2(a)). This size hierarchy shows vastly more small, dominated plants than large, dominating plants. In fact, the hierarchy shows a *log-normal* distribution of plants; the logarithm of the mass of the plants is distributed normally (Figure 2.2(b)). Log-normal size hierarchies are observed in a number of cultivated and natural populations, including even-aged monocultures [Har77].

2.4 Overdispersion and clustering

A plant's spatial location is not independent of the spatial locations of other plants. Under a thinning process, plants which are too close to other plants die off, causing the average distance between plants to increase. This causes *overdispersion* of plants within the environment. *Clustered* distributions can also occur; in these distributions, the presence of one plant increases the probability of finding another plant in its vicinity. This clustering can

have several causes, such as environmental heterogeneity (plants of the same type clustering in an environment favourable to them) or propagation (seeds falling close to their parent plants).

As plants are largely discrete entities, they can be analyzed by abstracting them to points. Many of the properties of the spatial distribution of a plant community can then be drawn from the study of spatial point patterns [Dig83].

A spatial point pattern is a set of *points* in a two dimensional space [GBS02]. The simplest statistical model for point patterns is the *completely random* pattern, also called the *Poisson* pattern. In this pattern, the number of points in an area A has expected value $\lambda|A|$. If $N(A)$ denotes the number of points in the set A , then

$$\langle N(A) \rangle_A = \lambda|A|$$

This equation also serves as a definition for the *density* λ .

The density only describes an aggregate property of a set of points, not the relationships between them. When the points represent plants, these relationships are quite important. Indeed, the most common deviation from complete randomness is clustering [GBS02], in which points have a relatively greater tendency to be found near each other.

The degree of clustering can be quantified using several statistical measures [CE54, Hop54, Sin85]. For example, the *Hopkins index* [Hop54] is defined as the average distance from a randomly chosen point in the plane (which is not necessarily a member of the point pattern) to its nearest point in the pattern, divided by the average distance from a randomly chosen point in the pattern to its nearest other point in the pattern:

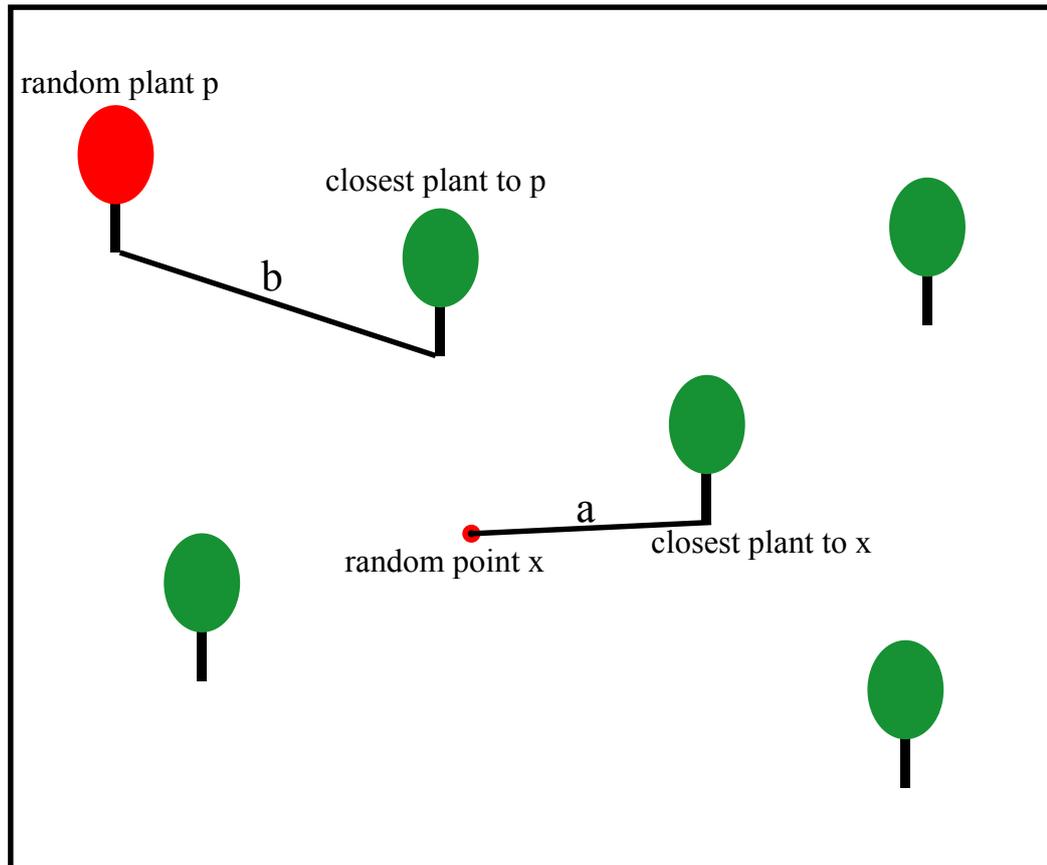


Figure 2.3: How the Hopkins index is calculated. a is the distance from a point in the plane to its nearest point in the pattern and b is the distance from a point in the pattern to the nearest other point in the pattern; the Hopkins index is then $H = \frac{\langle a \rangle}{\langle b \rangle}$

$$H = \frac{\langle a \rangle}{\langle b \rangle} = \frac{\langle \min_i (\|x - p_i\|) \rangle_x}{\langle \min_i (\|p_j - p_i\|) \rangle_j}$$

(See also Figure 2.3).

Distributions which are completely random have an H value of 1. Distributions that are more dispersed than random ('regular') have an H value less than 1, and distributions that are clustered have an H value greater than 1. For example, Figure 2.4(a) shows an

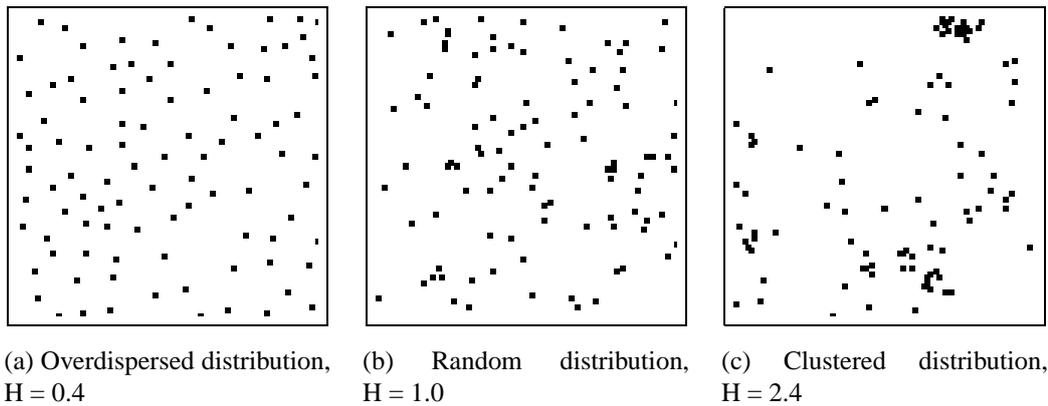


Figure 2.4: The effect of clustering on plant distribution.

overdispersed distribution of points with a Hopkins index of 0.4, Figure 2.4(b) shows a random distribution of points with a Hopkins index of 1.0, and Figure 2.4(c) shows a clustered distribution with a Hopkins index of 2.4.

CHAPTER 3

INDIVIDUAL-BASED MODELS OF PLANT COMMUNITIES:

A REVIEW

Two modeling paradigms

In this thesis I make use of a classification of individual-based models based on the direction of information flow. The first class is of *local-to-global* models, in which global structure emerges from local interactions among individuals. The second class is of *global-to-local* models, in which global structure is decomposed to instantiate individuals.

The global-to-local paradigm arises because of a limitation of simulation-based, local-to-global models. The amount of control a modeler has over the final outcome — the global structure — is limited. The modeler can control the model by setting boundary conditions — specifying the average protein concentration, or the crown shape of a tree. In very simple models with very few parameters, setting the parameters to achieve the specified boundary conditions may be as simple as solving a linear equation. As the model becomes more complex, however, finding the correct parameter settings becomes more difficult. The relation between parameters and boundary conditions becomes less tractable, and direct solution becomes infeasible. It is even possible that there are no parameter settings which produce the required boundary conditions.

One way to solve this problem is to *invert* the model, turning boundary conditions into parameters. A simple way to do this is to marginally alter the original model to fit the boundary conditions. For example, if the density of trees in one area of a forest simulation must be increased, a single extra tree can be added at each simulation step, in

addition to the trees added by the simulation. A more complete way to invert a model is to reformulate it in order to completely reverse the direction of information flow, creating a global-to-local model.

The division between local-to-global and global-to-local models has arisen in areas other than the modeling of plant communities. In [PMKL01], the distinction is discussed in the modeling of individual plants. In that paper, individual plant organs are instantiated to fit user-specified global organ distributions.

Multilevel modeling

A second drawback to the simulation-based approach is that as the number of individuals and their complexity grows, the number of possible interactions, and hence the amount of computation required to reproduce a scene, increases. One way to handle this drawback is to use *multilevel modeling* [GGCC97]. Multilevel modeling involves coupling models of a system at successively lower levels of abstraction. The higher-level, more abstract models are created first; then, the information from them is used to parameterize the lower-level models. The models of plant communities described in [DHL⁺98] are two-level models: the top level describes the community's spatial structure, while the lower level is of the geometry of individual plants.

Multilevel models are a useful way to handle the complexity of a simulation because of the multiscale nature of natural phenomena. A forest as a whole has a far greater influence on an individual tree than that tree has on the forest; a tree as a whole has far more influence on an individual leaf than that leaf has on the tree. Thus, while describing higher-level phenomena with a model appropriate to that scale loses some information which would be taken into account in a single-level model at a finer scale, it gains tremendously because

the individual interactions between low-level components is not considered; for instance, two leaves on different plants do not have to interact at all. This savings in the number of interactions allows more complex scenes to be modeled than would be computationally feasible in the single-scale approach.

In this chapter, I discuss previous work in the individual-based modeling of plant communities: local-to-global (Section 3.1) and global-to-local (Section 3.2) models of spatial distributions, and multilevel models (Section 3.3) of entire plant communities.

3.1 Local-to-global models

Local-to-global individual-based models of plant communities have been extensively studied in the field of ecology. Aikman and Watkinson [AW80] and, later, Firbank and Watkinson [FW85] and Lepš and Kindlmann [LK87], created models to explain the emergence of the $3/2$ power law in self-thinning monocultures (Section 2.2). Aikman and Watkinson derive a differential equation for the mass of a plant from “quasi-logical postulates”; the equation they use to calculate the mass m_i of an isolated plant i is

$$\frac{dm_i}{dt} = a_i q_i - b_i m_i^2.$$

Here q_i is the area of the plant’s *ecological neighbourhood*, the area from which it can extract resources, and is related to the area of the plant. The parameters a_i and b_i are growth constants which vary from plant to plant. The term $a_i q_i$ can be interpreted as the total resources extracted from the plant’s environment, while the term $b_i m_i^2$ is an amount which must be spent to maintain the plant’s structures. a_i and b_i are set so that the plant grows logistically to some maximum size M .

The plants interact through competition. The equation for the growth of a competing plant is

$$\frac{dm_i}{dt} = a_i q_i f(q_i) - b_i m_i^2,$$

where f is a function which relates to the amount of competition experienced by the plant. If dm_i/dt becomes negative, the plant does not have enough material to maintain itself, and it dies. This happens when it experiences too much competition. The competition function f is not based on the size of the plant's neighbours, however, but only on the average size of all plants in comparison to the area of the plot. Computer simulations do, however, show a fit to the self-thinning curve. As well, the size hierarchy formed fits well to a log-normal distribution.

In Firbank and Watkinson's model, the size of a plant *is* dependent on the sizes of neighbouring plants. The mass of a plant m_t is given by

$$m_t = m_{It} \left(\frac{z_t}{q_t} \right)^r,$$

where m_{It} is the mass of an isolated plant at time t , q is the size of the plant's ecological neighbourhood, r is the plant's efficiency of resource use, and z_t is the *zone of influence* of the plant. When the zone of influence is the same as the ecological neighbourhood, the plant is the same size as it would be if it were not competing for resources. In this case, m increases logistically to some maximum M .

If the plant is competing for resources, then the zone of influence will not be the entire ecological neighbourhood. The ecological neighbourhood is a circle of area q_t , and intersections of two or more circles imply competition between the plants so represented.

A plant's zone of influence includes the portions of its ecological neighbourhood to which it has exclusive access, as well as a fraction of the shared area. The overlapping area is divided between the plants sharing it, with the tallest plant getting a fraction d of the area, the next tallest getting a fraction d of the remainder, and so on. A value of $d = 1$ represents completely one-sided competition, in which the tallest plant gets all of the resources of the shared areas.

The height h , which determines the resource allocation in overlapping areas, is related to m through a power function:

$$h_t = um_t^v.$$

If a plant's mass decreases, it is counted as dead and removed from the simulation. Firbank and Watkinson examine the effect of changing the parameters r (the efficiency of resource utilization), d (the competition factor), and u and v (the allometric parameters relating height to mass) on the model. They found good fit to the self thinning curve and a log-normal distribution, with the allometric parameter v having the greatest effect: values of v around $1/3$ (that is, plants whose height is proportional to the cube root of the mass) create self-thinning lines with slopes of around $-3/2$, while higher and lower values steepen and flatten the curve, respectively.

Lepš and Kindlmann [LK87] use a similar model to examine the results of different initial configurations on the development of a stand of trees. Their model computes the height h at time $t + 1$ as

$$h_i(t + 1) = h_i(t) \left[1 + a \left(1 - \frac{h_i(t)}{H} \right) f_i(t) \right]$$

where a is the growth rate, H is the maximum possible height of the tree, and f_i is a term expressing the competitive influence of neighbouring trees. In the absence of competition ($f_i = 1$), a tree will grow logistically to the maximum height H .

The plant's ecological neighbourhood is a circle with radius $r_i = kh_i$ and area $q_i = \pi r_i^2$. The competition factor f_i gives the proportion of the ecological neighbourhood which is available for use. If ℓ_{ij} is the area of overlap between individuals i and j , then

$$f_i = 1 - \frac{\sum_{j \neq i} \ell_{ij} \frac{h_j}{h_i}}{Qq_i}.$$

Here Q is a parameter which describes the intensity of neighbours' influence — for large values of Q , the neighbours have very little competitive influence on the plant. The area of overlap ℓ_{ij} is weighted by the ratio of the heights of influencing and influenced plants. An individual is more influenced by taller neighbours than shorter neighbours. For very tall competitors or large areas of overlap, f_i may drop below 0; it is thus explicitly bounded to be nonnegative. If f_i is zero, the plant will experience no growth.

The competition factor also affects the survival probability of the plant. If very low, the plant has a high probability of dying. Plants are randomly chosen to die each simulation step.

Lepš and Kindlmann used their model to examine how plant distributions with various initial configurations evolved differently under a self-thinning process. They found that, no matter what the initial configuration — regular, clustered, or random — the stand evolved toward an overdispersed distribution.

Pacala and Silander [PS85] described an individual-based model for the analysis of plant monocultures. The model represents plants as circles in a two-dimensional space —

the plants interact when their respective circles intersect. There are three submodels — one for propagation (which includes fecundity and seed dispersal), one for survival (which depends on ground conditions, neighbouring plants, and the amount of sunlight the plant can absorb), and one for growth (which involves different parameters for germination, growth as a seedling, and mature growth).

Pacala and Silander’s model is more complex than the models described above which only seek to explain self-thinning. For one thing, it includes the propagation model, which must take into account all of the neighbours of the plant. All of the models described in this paper include only the *number* of neighbours, and not their spatial relation or relative size. However, in a further series of papers by Pacala and his collaborators [PCS93, PD95, DLDB97], the model and its submodels are developed into the extremely successful SORTIE model of the forests of eastern North America, which explicitly takes into account the position, size, and species of neighbours, the underlying terrain and soil content, and many other factors.

More recently, Enquist and Niklas [EN01] have examined plant community invariants with an individual-based simulation model. Plants in this model have three “compartments” for energy acquired from sunlight: some goes into leaves, which increases the crown radius and allows the capture of more light; some goes into the stem, which allows the plant’s height to increase; and some is kept in reserve for reproduction:

$$aM_{TOT}^{\alpha} = bM_{REPR}^{\beta} + cM_{LEAF}^{\gamma} + dM_{STEM}^{\delta}$$

where species differ only in the parameters a , b , c , and d ; α , β , γ , and δ are constant across species. The plants are modeled as circles with area q a function of M_{LEAF} , at height h a

function of M_{STEM} . If a plant's area is unobstructed (if it has no competitors, or it is taller than its competitors) then it receives the full benefit from photosynthesis; plants which are located below the canopy receive a fraction of this amount, determined by a predetermined light attenuation factor, which also varies across species.

Running simulations of this model, Enquist and Niklas found allometric scaling relationships which confirm empirical observations made across ecosystem types. For instance, if N is the total number of plants with some range of masses $[m_{low}, m_{high}]$ and M is the total mass of those plants, then, once the distribution reaches equilibrium, $N \propto M^{-3/4}$.

Progress on local-to-global individual-based models of plant communities in ecology led to similar models within the field of computer graphics, along with realistic visualizations of the communities. The SIGGRAPH 1998 paper on ecosystem simulation [DHL⁺98] includes two individual-based local-to-global models of plant distributions. The first, a simplified version of the model of Firbank and Watkinson [FW85], was designed to exhibit the self-thinning property. Plants are represented by circles with radius r . The operation of the model can be reduced to three rules (Figure 3.1). The first rule states that whenever the circles representing two plants intersect, the smaller is considered *dominated* and is removed from the population. The second rule states that once a plant reaches its maximum size, it is considered *mature* and ceases to grow. The third rule states that a plant which is neither dominated nor mature will grow. The growth rule is

$$r_{i+1} = r_i(1 + \text{random}(\Delta r)).$$

Under this growth rule, plants in the absence of competitors will grow exponentially up to the size of maturity. Simulation with this model produced a good fit to the self-thinning curve.

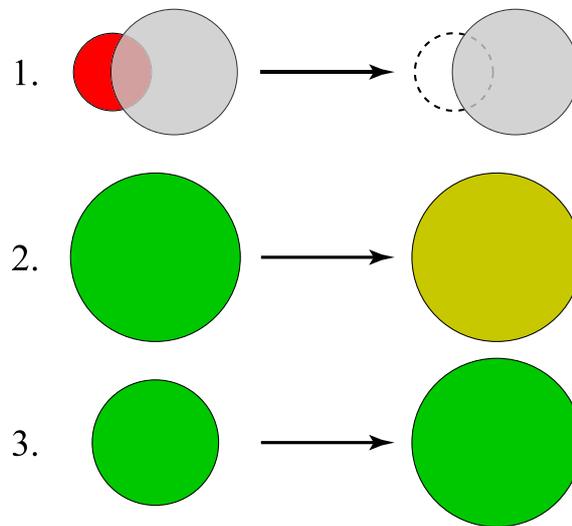


Figure 3.1: Diagrammatic representation of the model of self-thinning described in [DHL⁺98]. Green circles represent growing plants, the red circle represents a dominated plant, and the yellow circle represents a mature plant that no longer grows.

The second model was slightly more complicated and involved several species of plants with different preferences for moisture level. Competing in a heterogeneous environment, a segregation of plants emerged between wet and dry areas. A log-normal size hierarchy was observed in the mature populations.

3.2 Global-to-local models

Within ecology, the explanation and exploration of observed phenomena is of greater importance than what particular microscopic state is obtained. Thus, the statistics of plant distributions is more important than where, specifically, the plants are. Global-to-local models are, then, of limited use within ecology.

In the field of computer graphics, however, the particular microscopic state is impor-

tant, as that is what is rendered. Several methods for generating forest scenes have been proposed. All of these use a global-to-local method for generating the spatial distribution of trees.

Chiba et al. [CMDH97] place the trees randomly, using Poisson disc sampling [WW92]. This technique places points one at a time and at random, throwing out all points which are closer than a given distance to a previously generated point. Neyret [Ney96] places plants on every vertex of a rectangular grid, then jitters the positions. In both cases, the desired global state is a constant tree density in the forest, with regular spacing between trees.

Reeves and Blau [RB85] place the trees on the vertices on a rectangular grid, with the added condition of a minimum distance between trees. The tree's species can be chosen interactively or procedurally. In one illustrated method, the species is determined by an underlying heightfield, which otherwise has no effect. Higher trees are more likely to be evergreens, lower trees more likely to be deciduous. Again, the desired global state is a constant tree density with regular spacing, along with a greater concentration of conifers in the heights and deciduous trees in the valleys.

Deussen [DHL⁺98] describes a global-to-local model which takes in a user-specified density map. This density map is a greyscale image specifying how densely packed the plants are in each area. The density map is run through a Floyd-Steinberg error diffusion process [FS75] which creates individual points situated on grid vertices. After jittering these points, plants are placed at each one.

All of these methods create plant distributions that are overdispersed; indeed, in most they are overdispersed by design, explicitly enforcing minimum distances between the plants.

3.3 Multilevel modeling

The closest parallel to multilevel models in ecology is the technique of *aggregation models* [ACVP00], which allow the modeling of higher ecological levels by grouping individuals or species together into aggregates that are treated as units.

In computer graphics, however, all of the models described above are multilevel models, in the sense that they determine a plant distribution, then place individual plant models at those positions. The models of Neyret [Ney96] and Chiba et al. [CMDH97], as well as the global-to-local model described in Deussen et al. [DHL⁺98] have the barest possible connection between the ecological levels; the actual plant models which are placed at each position are randomly determined, independent of the spatial distribution.

The spatial distribution model of Reeves and Blau [RB85] can also calculate the species of each tree procedurally; for instance, dependent on altitude. The tree models placed at each position are of the computed species.

The local-to-global spatial distribution models of the SIGGRAPH 1998 paper [DHL⁺98] describe both the species and the size of each plant. The plant models are procedurally determined; a pre-generated model of the plant's species is executed a number of development steps proportional to the size determined in the spatial distribution model. Smaller plants are thus instantiated with models of younger plants; larger plants with models of older plants.

CHAPTER 4

LOCAL-TO-GLOBAL MODELS OF PLANT COMMUNITIES

Local-to-global models derive global patterns — the spatial distribution of plants — from local interactions between individuals. All of the local-to-global models of plant communities described in this chapter are implemented in L-systems [PL90], which were originally designed to represent individual branching plant structures. Section 4.1 describes L-systems, and Section 4.2 describes multilevel L-systems [LP02], an extension which allows their use in the modeling of populations of individuals. Three L-system based models of plant communities follow, capturing the phenomena of self-thinning (Section 4.3), succession (Section 4.4), and clustering (Section 4.5).

4.1 L-systems

L-systems were introduced in [Lin68], and were designed to model individual branching plant structures, which they represent by strings of symbols. Formally, an L-system consists of three components $\langle V, \Omega, P \rangle$: V is the *alphabet*, which is a set of all of the symbols which the system will use; $\Omega \subset V^*$ is the *axiom*, which represents the initial state of the system; and $P \subset V \times V^*$ is a set of *productions*, which define how the system develops over steps of time [PL90]. In each iteration, the productions are applied exactly once to each symbol in the string, in parallel, yielding a new string. By convention, if more than one production applies, the production which appears first in the production list is used. Both the topology and the geometry of the plant can be included in the model. To model topology, *bracketed string notation* [Lin68] is used; with this notation, anything inside a

pair of brackets $[]$ is a branch. To model geometry, we can use *turtle notation* [PL90], in which symbols are interpreted as commands to a ‘turtle’ that moves in a two- or three-dimensional space.

The following L-system illustrates the essential features of the language:

L-system 4.1:

Alphabet: $\{ I, B, [,] \}$

Axiom: IB

Production: $B \rightarrow [IB]IB$

The alphabet consists of the symbols I, and B. In a botanical context, these can be interpreted as *Internodes* and *Buds*. The axiom is IB, an internode topped with a bud. The production grows the plant from the bud — a bud yields a branch, another internode, and another bud. The string [IB] represents a branch in the *bracketed string notation* [Lin68]; using this notation, everything inside a pair of brackets $[]$ is a branch.

Figure 4.1 illustrates the execution of this L-system, visualizing internodes as lines and buds as circles.

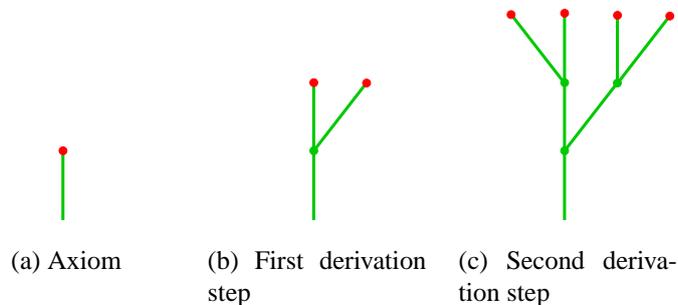


Figure 4.1: The execution of L-system 4.1.

This thesis uses several extensions to the L-system formalism described above. First, symbols such as I and B can be extended by *parameters* [PL90, PH90, Han92], which can be acted upon as variables. This is illustrated in L-system 4.2:

L-system 4.2:Axiom: $IB(1)$

1. $B(t) : t \geq 2 \rightarrow L$
2. $B(t) \rightarrow IB(t+1)$

This L-system is similar to the one above; however, in this case, the symbol B takes a single parameter, which may be interpreted as 'age'. The first production turns a bud with parameter 2 or greater into a *Leaf*. The second production is only used if the first fails; that is, if B 's parameter is less than 2. In that case, it produces an internode topped by a bud which is one unit older. This L-system is illustrated in Figure 4.2.

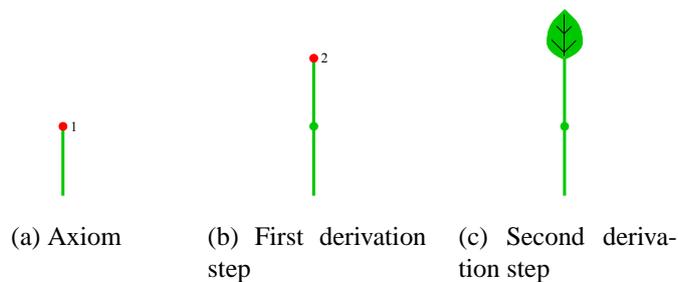


Figure 4.2: The execution of L-system 4.2.

Another extension of L-systems used in this thesis is *pseudo-L-systems* [Pru86], which makes it possible to rewrite two or more symbols using a single production. Finally, the extension of *open L-systems* [MP96] makes it possible to capture the interactions between the modeled plants and their environment.

In open L-systems, the environment is conceived as an external system which is linked to the L-system through the *environmental communication symbol* $?E$. The communication symbol is treated like any other within the L-system during the derivation step. After derivation, the environmental communication symbol, the symbol immediately preceding

it, and whatever parameters they hold are sent to the environmental program. The environment performs whatever processing is required, then changes the parameters of the communication symbol and sends it back to the L-system. The altered parameters can then be tested by the L-system and different actions taken depending on their value.

4.2 Multiset L-systems

L-systems are largely used to model the development of a single organism. This thesis uses L-systems to model interaction between many plants at the ecosystem level as well. To this end, it uses the formalism of *multiset L-systems*, which were introduced in [LP02]. That paper described them as follows.

Multiset L-systems unify and extend to branching structures two previously defined notions of the L-system theory: developmental systems with finite axiom sets [RL74] and L-systems with fragmentation [RRS76]. In multiset L-systems, the set of productions operates on a multiset of strings that represent many plants, rather than a single string that represents an individual plant. New strings can be dynamically added to or removed from this multiset, representing organisms that are added to or removed from the population. All interaction between strings is handled through the environment.

Formally, a context-free non-parametric multiset L-system is a four-tuple $G = \langle V, \%, \Omega, P \rangle$ where V is the *alphabet* (a finite set of symbols), $\% \notin V$ is a reserved *fragmentation symbol*, $\Omega \subset V^*$ is a finite set of words over V called the *axiom*, and $P \subset V \times (V \cup \{\%\})^*$ is a finite set of *productions*. The alphabet V may contain, in particular, a pair of brackets, [and], which are used to delimit branches as described above and in [PL90].

A derivation step in a multiset L-system consists of two sub-steps. First, all words

x_i in the predecessor multiset are replaced by the intermediate successor words y_i using productions in P . The individual derivations $x_i \rightarrow y_i$ are performed as in an ordinary L-system. Second, the words y_i that contain one or more fragmentation symbols % are subdivided. In this process, the symbol % acts as the marker of positions at which branches y_{ik} are cut off the tree y_i . The remaining part of the tree y_i and the cut off branches y_{ik} become the members of the successor multiset.

For example, let us consider the multiset L-system 4.3:

L-system 4.3:

Alphabet: {A, B, I, [,]}
 Axiom: { A, B }
 Productions: 1. $A \rightarrow I[B]A$
 2. $B \rightarrow B\%A$

Starting with the axiom, the first two derivation steps yield the multisets listed in Table 4.1.

step	intermediate multiset	final multiset
0	{A, B}	{ A, B }
1	{ I[B]A, B% A }	{ I[B]A, B, A }
2	{ I[B% A]I[B]A, B% A, I[B]A }	{ I[B]I[B]A, A, B, A, I[B]A }

Table 4.1: Operation of a sample multiset L-system

Extensions of L-systems, such as pseudo-L-systems and open L-systems, also apply to multiset L-systems. In particular, in the simulation of ecosystems we rely extensively on the environmental query symbol ?E to simulate interaction of plants with their environment. The models of the plants themselves are extremely simplified, in order to accommodate a large number of plants. The L-system-based plant modeling software

L-studio/cpfg [PKMH00] was used to both generate the plant distributions and to model the individual plants.

4.3 The self-thinning model

As an illustration of the use of multiset L-systems, consider the implementation of the individual-based model of self-thinning outlined in [DHL⁺98]. We construct an L-system from the three rules of plant development (Figure 3.1). In the L-system, each plant is parameterized only by position and size; in module form, $T(\vec{x}, r)$ ¹.

L-system 4.4:

Axiom: $\{ T(\vec{x}_1, r_1)?E(1),$
 $T(\vec{x}_2, r_2)?E(1),$
 $\dots,$
 $T(\vec{x}_n, r_n)?E(1) \}$

1. $T(\vec{x}, r)?E(c) : c == 0 \rightarrow \epsilon$
2. $T(\vec{x}, r) : r \geq R \rightarrow T(\vec{x}, R)$
3. $T(\vec{x}, r) \rightarrow T(\vec{x}, r + \text{grow}(r, \Delta t))$

Each plant is represented by the module $T(\vec{x}, r)$ followed by the communication symbol $?E(c)$. The parameter c is used for communication with the environmental process, which sets it to 0 if the plant is dominated and 1 if the plant is not dominated. The environmental program treats all of the trees as circles of radius r and determines which circles are intersecting; the smaller of any pair of intersecting circles is considered dominated.

The axiom introduces n plants with random positions and sizes; the initial distribution of plants could also be generated algorithmically. Under production 1 any dominated plant (that is, a plant with $c == 0$) will be removed from the population in the next iteration,

¹In L-studio/cpfg, vectors cannot be parameters of modules. However, for the sake of simplicity, the vector \vec{x} will stand in here for the double (x, y) .

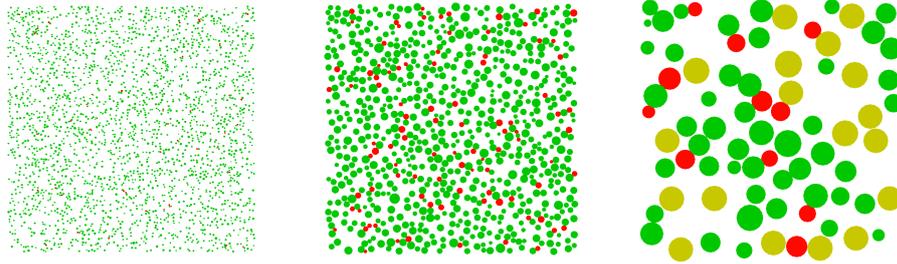


Figure 4.3: Three stages of simulation of the self-thinning process. Green circles are growing plants, red circles are dominated plants, and yellow circles are mature plants, as in Figure 3.1.

along with its associated communication module. Production 2 stops the growth of a plant which has reached the maximum size R . Finally, production 3 increases the size of a plant which is neither dominated nor mature. The user-defined function $\text{grow}(r, \Delta t)$ captures the growth of a plant of radius r over time interval Δt .

Figure 4.3 shows three stages of a self-thinning process simulated using this L-system. As the plant community develops over time, dominated plants gradually disappear and thin out the distribution. Figure 4.4 shows a graph obtained by sampling the state of the system at each time step of a single simulation — compare it with the theoretical self-thinning graph in Figure 2.1.

4.4 Plant succession

An extension to L-system 4.4 transforms it into a model of interaction between two plant species.

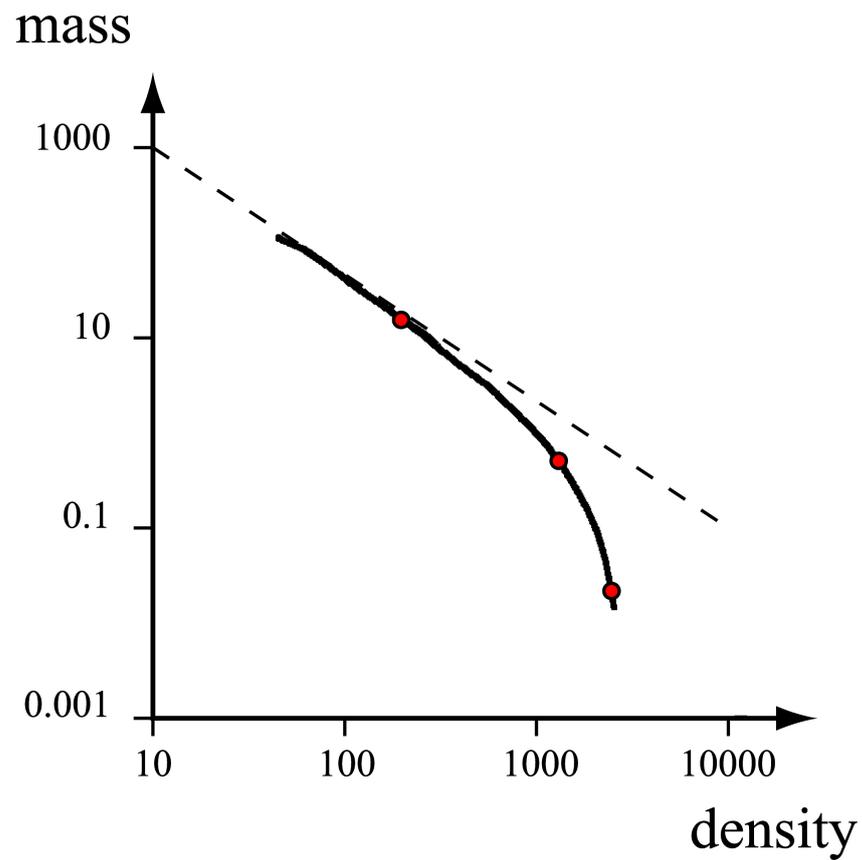


Figure 4.4: A self-thinning curve which tracks the progress of the simulation shown in Figure 4.3. The dotted line has a slope of $-3/2$ and the thick line is made up of samples taken from the simulation at each time step. The circles represent the three steps shown in Figure 4.3.

L-system 4.5:

Axiom: { X }

1. $X \rightarrow T(\vec{x}_1, r_1, 1)?E(1) \%$
 \dots
 $T(\vec{x}_n, r_n, 1)?E(1) \%$
 $T(\vec{x}_{n+1}, r_{n+1}, 2)?E(1) \%$
 \dots
 $T(\vec{x}_{n+m}, r_{n+m}, 2)?E(1) \% X$
2. $T(\vec{x}, r, sp) ?E(c) : c == 0 \ \&\& \ \text{random}(1) < \text{shaded}[sp] \rightarrow T(\vec{x}, r, sp) ?E(1)$
3. $T(\vec{x}, r, sp) ?E(c) : c == 0 \rightarrow \epsilon$
4. $T(\vec{x}, r, sp) : r \geq R \ \&\& \ \text{random}(1) < \text{oldage}[sp] \rightarrow T(\vec{x}, R, sp)$
5. $T(\vec{x}, r, sp) : r \geq R \rightarrow \epsilon$
6. $T(\vec{x}, r, sp) \rightarrow T(\vec{x}, r + \text{grow}(r, sp, \Delta t), sp)$

In this model a plant is represented by the module $T(\vec{x}, r, sp)$. Parameters \vec{x} and r represent the plant's position and size, as in the previous model; the parameter sp is the plant's species identifier, either 1 or 2. Production 1 adds n new plants of species 1 and m new plants of species 2 to the population. The production predecessor X reappears in the successor multiset; thus, new plants are added in every simulation step. Productions 2 and 3 removes a dominated plant with probability $1 - \text{shaded}[sp]$. The value $\text{shaded}[sp]$, called the *shade tolerance* of the plant, is a measure of how well it can handle being in shadow. Regardless of shade tolerance, a plant that is dominated will not grow.

Productions 4 and 5 model the senescence of plants. Once a plant has reached the radius R , it survives with the probability $\text{oldage}[sp]$; a plant that does not survive dies and is removed from the community. Production 6 uses the growth function $\text{grow}(r, sp, \Delta t)$ to simulate the growth of plants that are neither dominated nor old, according to their size and species.

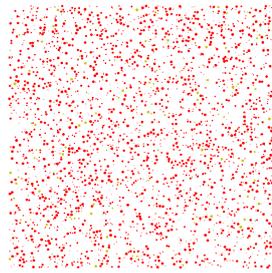
With the right parameterization, this model captures the phenomenon of succession (Section 2.1). If species 1 has a higher growth rate but lower shade tolerance and old-age survivorship than species 2 ($\text{grow}(r,1,\Delta t) > \text{grow}(r,2,\Delta t)$, $\text{shaded}[1] < \text{shaded}[2]$, $\text{oldage}[1] < \text{oldage}[2]$), then an initially empty field will be populated in stages. First, the field will be dominated by species 1. Then, as the largest members of species 1 die, smaller members of species 2, which have survived due to their greater shade tolerance and now have a size advantage over young seedlings of species 1, fill in the gaps. Eventually, the field will be dominated by members of species 2. A straightforward extension of this model to three plant species is illustrated in Figure 4.5, with realistic plant models created by the methods discussed in Chapter 6.

Models of plant succession are important in practice, specifically in the simulation of tree regrowth after cutting. Our simple model is just the first step toward addressing the problem of visualizing plant succession with enough accuracy for predictive purposes.

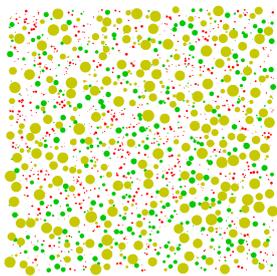
4.5 Clustering through plant propagation

The evaluation of the Hopkins index for the distributions shown in Figure 4.3 yield values of H equal to 0.8, 0.4, and 0.4, respectively. Similarly, calculating the Hopkins index for the distributions in Figure 4.5 results in H values of 0.6, 0.7, 0.7, and 0.6, respectively. This shows that the competition for space in a thinning process leads to overdispersed plant distributions.

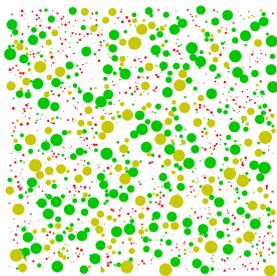
We can see why this is the case. If any two plants so much as touch each other, one of them will become dominated. In the self-thinning model, the dominated plant immediately dies; in the succession model, it dies with some probability per derivation step. In either



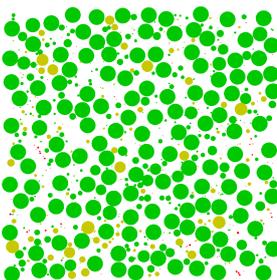
(a) After 20 iterations



(b) After 150 iterations



(c) After 300 iterations



(d) After 1000 iterations

Figure 4.5: Four stages of ecosystem simulation using the plant succession model of L-system 4.5. Left: results of coarse-level simulation, using pink circles to indicate position of herbaceous plants (fireweed), orange circles to indicate positions of the early-succession deciduous trees, and green circles to indicate positions of the late-succession coniferous trees. Right: synthetic images obtained by placing tree models at the locations generated by the coarse-level simulation.

case, the competition for space drives the plants apart, and there is no opposite mechanism encouraging plants to cluster.

One clustering mechanism observed in nature is local propagation. We can capture it, for instance, by ‘sowing’ new plants near parent plants of the same species, instead of making them appear at random throughout the field. The resulting alteration of the succession model is given in L-system 4.6.

L-system 4.6:

Axiom: $\{ T(\vec{x}_1, r_1, 1) ? E(1) ,$

$\dots ,$
 $T(\vec{x}_n, r_n, 1) ? E(1) ,$
 $T(\vec{x}_{n+1}, r_{n+1}, 2) ? E(1) ,$
 $\dots ,$
 $T(\vec{x}_{n+m}, r_{n+m}, 2) ? E(1) \}$

1. $T(\vec{x}, r, sp) ? E(c) : c == 0 \ \&\& \ \text{random}(1) < \text{shaded}[sp] \rightarrow T(\vec{x}, r, sp)$
2. $T(\vec{x}, r, sp) ? E(c) : c == 0 \rightarrow \epsilon$
3. $T(\vec{x}, r, sp) : r \geq R \ \&\& \ \text{random}(1) > \text{oldage}[sp] \rightarrow T(\vec{x}, R, sp)$
4. $T(\vec{x}, r, sp) : r \geq R \rightarrow \epsilon$
5. $T(\vec{x}, r, sp) ? E(c) \rightarrow T(\vec{x}, r + \text{grow}(sp, r, \Delta t), sp) ? E(1) \% T(\vec{x} + \Delta \vec{x}, r_0, sp) ? E(1)$

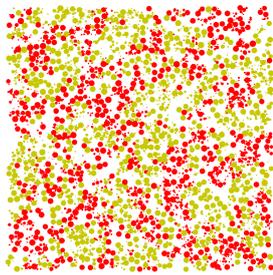
The axiom defines the initial state of the model by placing n plants of species 1 and m plants of species 2 at random in the field. The subsequent productions are the same as in the succession model, except for production 5. According to it, a plant that is not dominated creates a new plant at position $\vec{x} + \Delta \vec{x}$, where $\Delta \vec{x}$ is a small random vector. Since the new plant is in close proximity to its parent, this propagation mechanism encourages clustering in the distribution.

Figure 4.6 illustrates the operation of this model. At the beginning, plants are randomly distributed. As the ecosystem develops, the two species become spatially segregated, cre-

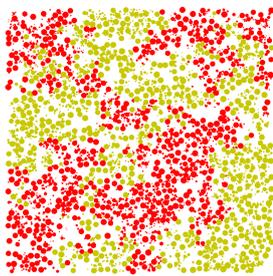
ating large clusters of plants of each species. For example, the Hopkins indices of species 1 at the three stages shown are equal to 1.1, 4.2, and 11, respectively.

4.6 Conclusion

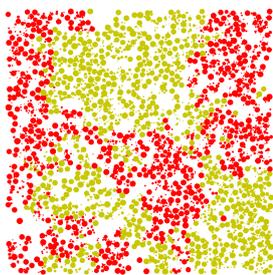
The three models described in this chapter capture many of the ecological phenomena discussed in Chapter 2. The thinning model of Section 4.3, originally described in [DHL⁺98], exhibits a very good fit to the theoretical self-thinning curve. The succession model (Section 4.4) produces the expected waves of dominance by different species, from fast-growing to shade-tolerant. Finally, a high Hopkins index indicates that the model of clustering by propagation (Section 4.5) does indeed create spatial distributions that exhibit clustering.



(a) After 20 iterations



(b) After 500 iterations



(c) After 1000 iterations

Figure 4.6: Three stages of ecosystem development simulated using the plant propagation model of L-system 4.6. Left: results of coarse-level simulation, using orange circles to indicate positions of poplar trees and green circles to indicate positions of spruce trees. Right: synthetic images obtained by placing tree models at the locations generated by the coarse-level simulation.

CHAPTER 5

GLOBAL-TO-LOCAL MODELS OF PLANT COMMUNITIES

Global-to-local models derive local instantiations — the locations and other properties of individual plants — from specified global properties. The global-to-local methods described in this chapter use a *deformation kernel* to decompose a global probability distribution into individual plants. The deformation kernel describes how the presence of a reference plant alters the probability that a second plant will be found at a given distance from it. The global-to-local model using deformation kernels is described in Section 5.1. Implementation is discussed in Section 5.2, and results are shown in Sections 5.3 and 5.4.

5.1 The deformation kernel method

The effect one plant in the self-thinning model of L-system 4.4 has on the probability of finding another plant nearby is shown diagrammatically in Figure 5.1. Within the reference plant radius r_t , the probability of finding another plant is very small; outside that radius, the probability of finding another plant is not affected by the reference plant.

The function K shown in Figure 5.1 is an example of a *deformation kernel*. If we suppose there is a probability field P which gives the probability of finding a plant at each point, the deformation kernel captures the impact of an existing plant on this field. Various interactions between plants can be described using deformation kernels of different shapes, as suggested in Figure 5.2.

A simple global-to-local placement algorithm can now be developed using this deformation kernel idea. We maintain a *joint probability density function* [Ros97] $f(x, y)$,

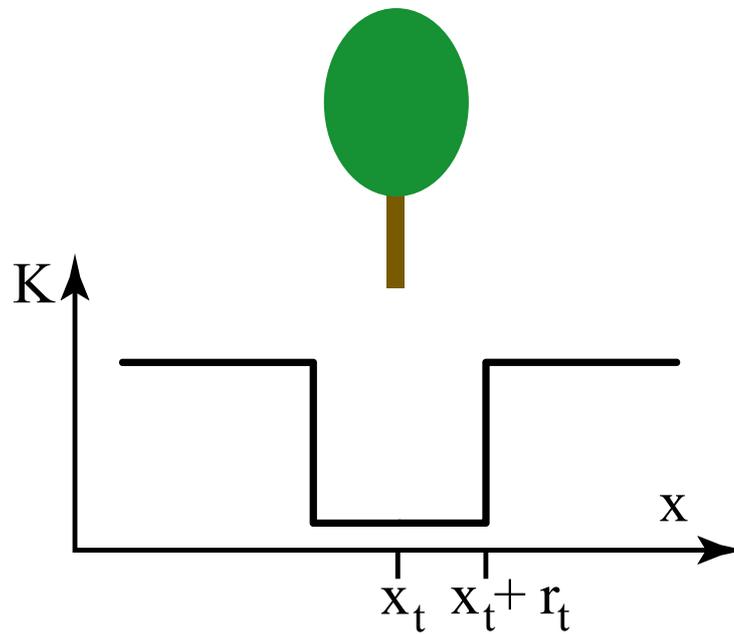


Figure 5.1: The effect a plant has on the probability of finding neighbouring plants.

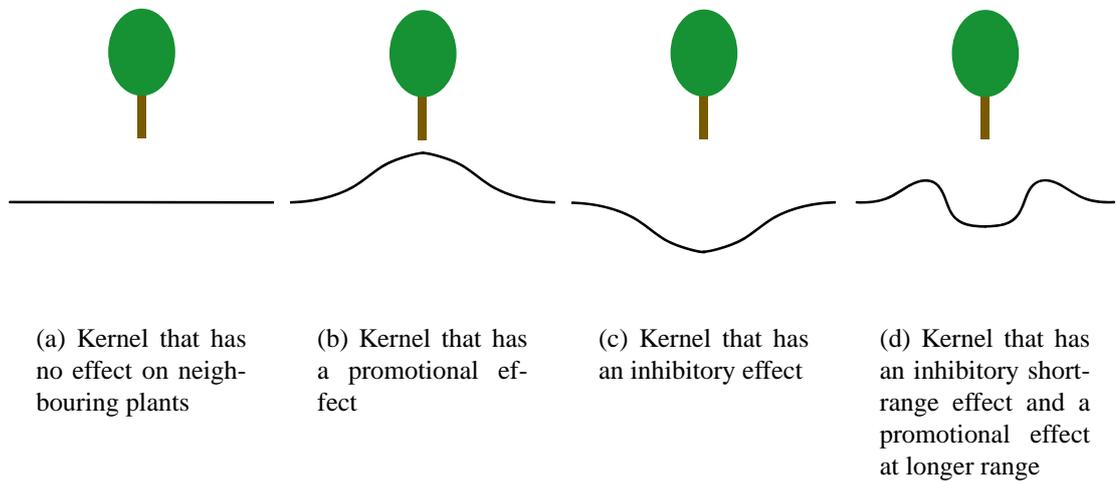


Figure 5.2: Examples of deformation kernels.

which characterizes the probability $f(x, y)dxdy$ of placing a new plant in the area $dxdy$ centered at $f(x, y)$. The plants are placed one at a time; as each is placed, its deformation kernel is applied to the probability function f that will be used to determine the position of the next plant. In this way, a distribution of plants will eventually be formed by decomposing f .

Formally, the joint density function f defines a probability field, where the probability of a new plant growing in the rectangle $[0, x_s] \times [0, y_s]$, with $0 \leq x_s \leq x_{max}$ and $0 \leq y_s \leq y_{max}$, is given by the cumulative probability distribution function

$$\begin{aligned} F(x_s, y_s) &= P\{x_t \leq x_s, y_t \leq y_s\} \\ &= \int_0^{x_s} \int_0^{y_s} f(x, y) dx dy. \end{aligned}$$

The plant must be somewhere in the field, so the probability that the plant will be found in the rectangle $[0, x_{max}] \times [0, y_{max}]$ is one, and the density function must satisfy the normalizing equation

$$\int_0^{x_{max}} \int_0^{y_{max}} f(x, y) dx dy = 1. \quad (5.1)$$

We find the position (x_t, y_t) of the plant to be added by calculating first its y , then its x coordinate. To this end, given the two-dimensional density function $f(x, y)$, we create the *marginal distribution function* $F_Y(y_s)$. That distribution function describes the probability

that $y_t \leq y_s$ independently of the choice of x_t ¹ :

$$F_Y(y_s) = P\{x_t \leq x_{max}, y_t \leq y_s\} = F(x_{max}, y_s).$$

We choose the y coordinate for the plant using the *inverse transformation method* [Ros97]. To this end, we generate a random number u from the uniform distribution on $[0, 1]$. We then search $F_Y(y)$ to find the value y_t such that $F_Y(y_t) = u$. As F_Y is monotone and continuous, y_t exists and is unique. This is our plant's y coordinate.

Once we have chosen y_t , we use the *conditional distribution* $F_{X|Y}(x_s | y_t)$, which describes the probability that $x_t \leq x_s$, given a y value y_t :

$$F_{X|Y}(x_s | y_t) = P\{x_t \leq x_s | y_t\} = \frac{\int_0^{x_s} f(x, y_t) dx}{\int_0^{x_{max}} f(x, y_t) dx}.$$

We then apply the inverse transformation method to find the coordinate x_t , given $F_{X|Y}(x_s | y_t)$.

Having placed a plant of size r_t at position (x_t, y_t) , we now deform the probability density function $f(x, y)$ in order to simulate the effects of that plant on the placement of nearby plants. To this end, we first multiply the probability density function $f(x, y)$ by the plant's deformation kernel $K(x, y)$,

$$f_{temp}(x, y) = f(x, y)K(x, y),$$

then renormalize the function $f_{temp}(x, y)$ to satisfy Equation 5.1. The deformation kernel typically is a function of the form

¹There is a corresponding marginal distribution function $F_X(x_s)$, which describes the probability that $x_t \leq x_s$, independent of what is chosen for y_t .

$$K(x, y) = \kappa \left(\frac{\sqrt{(x - x_t)^2 + (y - y_t)^2}}{r_t} \right),$$

where the function $\kappa(r)$ measures the effect a unit-sized plant has on the formation of plants at a distance r from it.

The operation of the kernel method is illustrated in Figure 5.3. The deformation kernel is Figure 5.2(d). The initial distribution is uniform; $f(x, y) = c$. In the middle, a single plant has been added to the field; the density function has been altered in the plant's neighbourhood. On the bottom, four more plants have been added, and the density function has been modified near each of them.

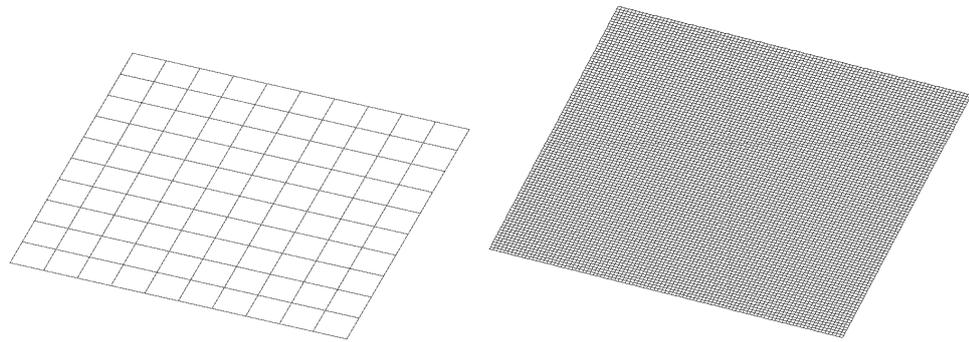
5.2 Implementation of the deformation kernel method

An obvious way to implement the above concepts is to represent values of the probability density function $f(x, y)$ using an n by n array of samples f_{ij} . To calculate position of a new plant, we create a vector \mathbf{R} of partial sums of the rows, where

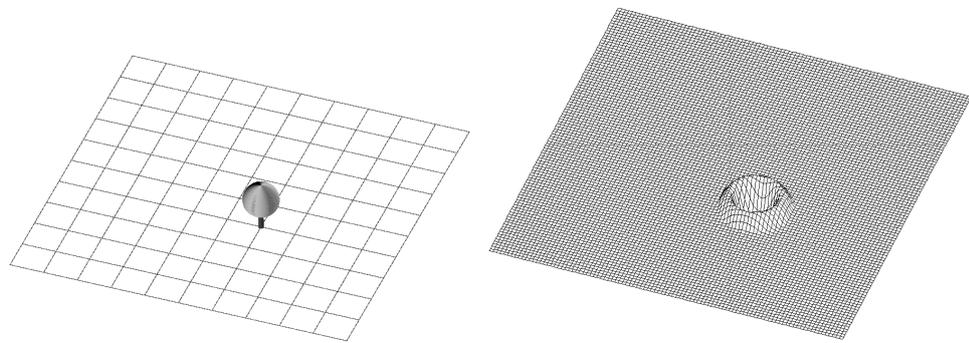
$$\mathbf{R}_k = \sum_{i=0}^k \sum_{j=0}^{n-1} f_{ij}, \quad k = 0, 1, \dots, n-1.$$

We determine the coordinate y_t of the newly placed plant using the inverse transformation method. To this end, we pick a random number u from the uniform distribution on the interval $[0, \mathbf{R}_{n-1}]$, then perform a binary search to locate \mathbf{R}_i such that $\mathbf{R}_i \leq u < \mathbf{R}_{i+1}$. We then linearly interpolate between (i, \mathbf{R}_i) and $(i+1, \mathbf{R}_{i+1})$ to find (y_t, u) .

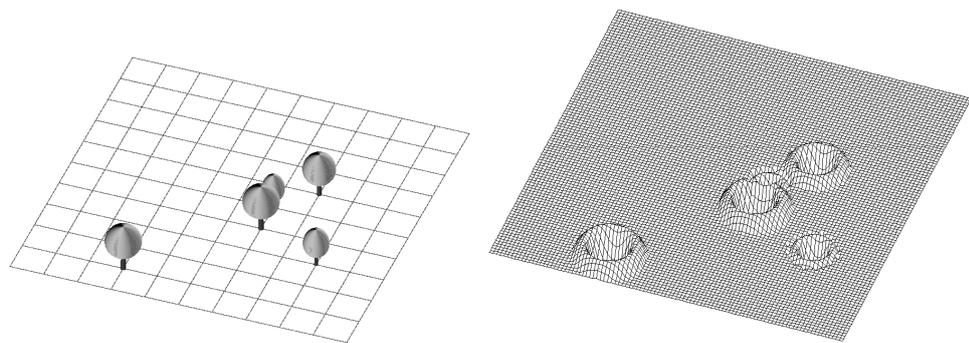
Now a vector \mathbf{C} of values representing the conditional distribution $F_{X|Y}(x_s | y_t)$ is computed by interpolating rows i and $i+1$ of the array f_{ij} .



(a) A uniform initial probability density



(b) After placing a single plant and deforming



(c) After placing several plants

Figure 5.3: An example of the deformation kernel algorithm, using the kernel of Figure 5.2(d). Left: the plant distribution. Right: the function f .

$$\mathbf{C}_k = (y_t - i) \sum_{j=0}^k f_{(i+1)j} + ((i + 1) - y_t) \sum_{j=0}^k f_{ij},$$

Given the values \mathbf{C}_k , $k = 0, 1, \dots, n - 1$, we choose a value from the uniform random distribution on the interval $[0, \mathbf{C}_{n-1}]$, and use the inverse transformation method to find x_t .

The kernel is applied by simply calculating the distance d of every sampling point (i, j) from (x_t, y_t) , then multiplying the value f_{ij} of the distribution function f at that point by $\kappa(\frac{d}{r_t})$. The entire process is captured by Algorithm 5.1.

Algorithm 5.1:

for each plant t do

$\mathbf{R}_0 \leftarrow 0$

 for i from 0 to $n - 1$ do

 for j from 0 to $n - 1$ do

$\mathbf{R}_i \leftarrow \mathbf{R}_i + f_{ij}$

$\mathbf{R}_{i+1} \leftarrow \mathbf{R}_i$

$u \leftarrow \text{rand}(\mathbf{R}_{n-1})$

 find k such that $\mathbf{R}_k \leq u < \mathbf{R}_{k+1}$

$y_t \leftarrow k + (u - \mathbf{R}_k) / (\mathbf{R}_{k+1} - \mathbf{R}_k)$

$\mathbf{C}_0 \leftarrow 0$

 for j from 0 to $n - 1$ do

$\mathbf{C}_j \leftarrow \mathbf{C}_j + ((y_t - k) \times f_{(k+1)j} + ((k + 1) - y_t) \times f_{kj})$

$\mathbf{C}_{j+1} \leftarrow \mathbf{C}_j$

$v \leftarrow \text{rand}(\mathbf{C}_{n-1})$

 find ℓ such that $\mathbf{C}_\ell \leq v < \mathbf{C}_{\ell+1}$

$x_t \leftarrow \ell + (v - \mathbf{C}_\ell) / (\mathbf{C}_{\ell+1} - \mathbf{C}_\ell)$

 for i from $(\lfloor y_t - r_t \rfloor)$ to $(\lceil y_t + r_t \rceil)$ do

 for j from $(\lfloor x_t - r_t \rfloor)$ to $(\lceil x_t + r_t \rceil)$ do

$d \leftarrow \sqrt{(x_t - j)^2 + (y_t - i)^2}$

$f_{ij} \leftarrow f_{ij} \times \kappa(d/r_t)$

If there are m plants to be placed and the function f is represented using n^2 values f_{ij} , the above algorithm will take $O(mn^2)$ time to run, since the sums R_k must be recalculated each time a new plant is placed.

We improve on this result by updating the array \mathbf{R} incrementally. When a kernel is applied to the distribution, the differences between f_{ij} and $\kappa(\frac{d}{r_t})f_{ij}$ are summed for each (i, j) within the range of the plant, and the differences are applied to the array \mathbf{R} . Assuming that the kernel is only applied to a small fraction of the cells in the grid, this operation can be performed in $O(n)$ time per plant. The modified algorithm is shown as Algorithm 5.2.

Algorithm 5.2:

```

 $\mathbf{R}_0 \leftarrow 0$ 
for  $i$  from 0 to  $n - 1$  do
  for  $j$  from 0 to  $n - 1$  do
     $\mathbf{R}_i \leftarrow \mathbf{R}_i + f_{ij}$ 
   $\mathbf{R}_{i+1} \leftarrow \mathbf{R}_i$ 

for each plant  $t$  do
   $u \leftarrow \text{rand}(\mathbf{R}_{n-1})$ 
  find  $k$  such that  $\mathbf{R}_k \leq u < \mathbf{R}_{k+1}$ 
   $y_t \leftarrow k + (u - \mathbf{R}_k) / (\mathbf{R}_{k+1} - \mathbf{R}_k)$ 

   $\mathbf{C}_0 \leftarrow 0$ 
  for  $j$  from 0 to  $n - 1$  do
     $\mathbf{C}_j \leftarrow \mathbf{C}_j + ((y_t - k) \times f_{(k+1)j} + ((k + 1) - y_t) \times f_{kj})$ 
   $\mathbf{C}_{j+1} \leftarrow \mathbf{C}_j$ 
   $v \leftarrow \text{rand}(\mathbf{C}_{n-1})$ 
  find  $\ell$  such that  $\mathbf{C}_\ell \leq v < \mathbf{C}_{\ell+1}$ 
   $x_t \leftarrow \ell + (v - \mathbf{C}_\ell) / (\mathbf{C}_{\ell+1} - \mathbf{C}_\ell)$ 

   $\Delta R \leftarrow 0$ 
  for  $i$  from  $(\lfloor y_t - r_t \rfloor)$  to  $(\lceil y_t + r_t \rceil)$  do
    for  $j$  from  $(\lfloor x_t - r_t \rfloor)$  to  $(\lceil x_t + r_t \rceil)$  do
       $d \leftarrow \sqrt{(x_t - j)^2 + (y_t - i)^2}$ 
       $\Delta R \leftarrow \Delta R + (f_{ij} - f_{ij} \times \kappa(d/r_t))$ 
       $f_{ij} \leftarrow f_{ij} \times \kappa(d/r_t)$ 
     $\mathbf{R}_i \leftarrow \mathbf{R}_i + \Delta R$ 

```

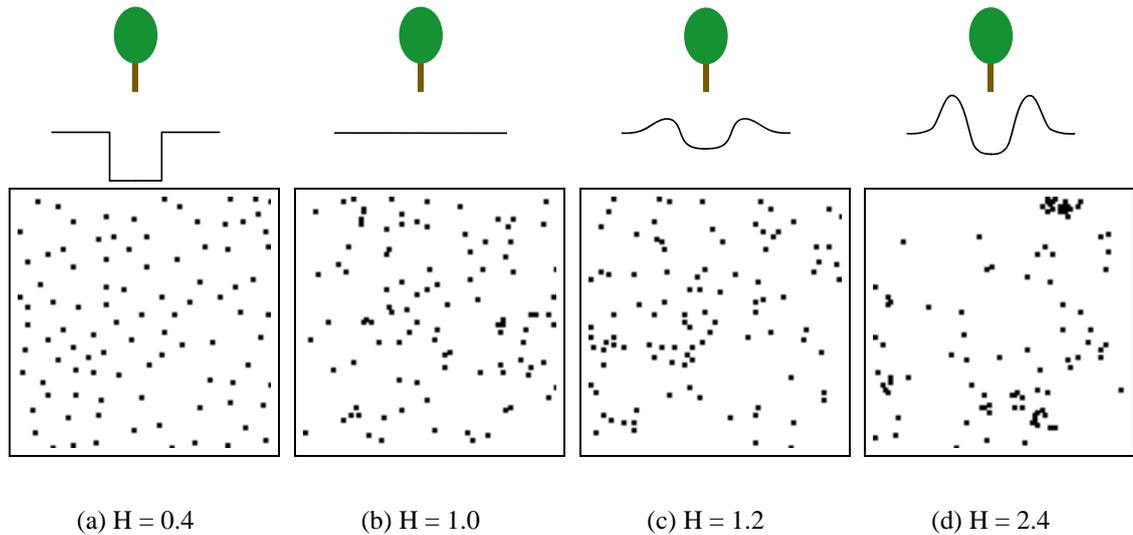


Figure 5.4: Some kernels and the point patterns they generate. The kernels are drawn at a larger scale than the point patterns.

```

for  $i$  from  $(\lceil y_t + r_t \rceil + 1)$  to  $n$  do
   $\mathbf{R}_i \leftarrow \mathbf{R}_i + \Delta R$ 

```

5.3 Results

Figure 5.4 shows point patterns generated using the above algorithm with different deformation kernels. We can check that they vary between uniformly distributed and clustered by calculating their Hopkins indices. The Hopkins indices of the figures shown are 0.4, 1.0, 1.2, and 2.4, confirming the visual observation that the kernel method is capable of creating a range of distributions, from regular to random and clustered.

In Figure 5.5 points have been replaced by simple models of daisies, created using an L-system. The very different visual impact of the three images demonstrates the importance of clustering to synthetic images of plant communities.

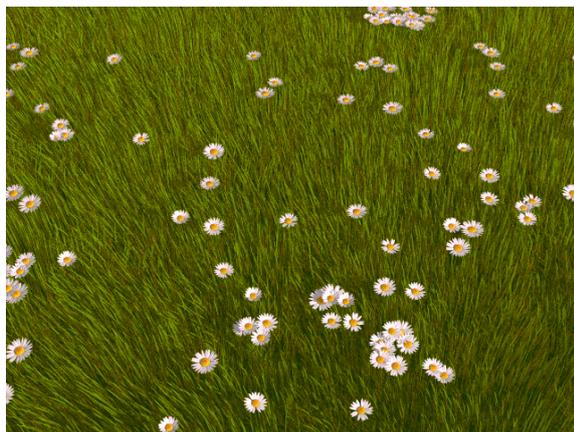
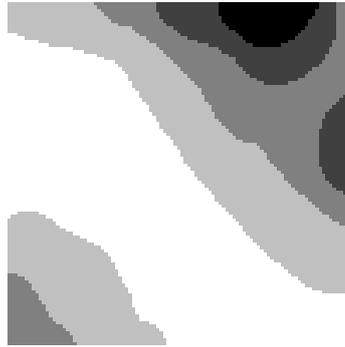
(a) $H = 0.4$ (b) $H = 1.2$ (c) $H = 2.4$

Figure 5.5: Point patterns corresponding to Figures 5.4(a), 5.4(c), and 5.4(d), rendered as fields of daisies.



(a) The density map

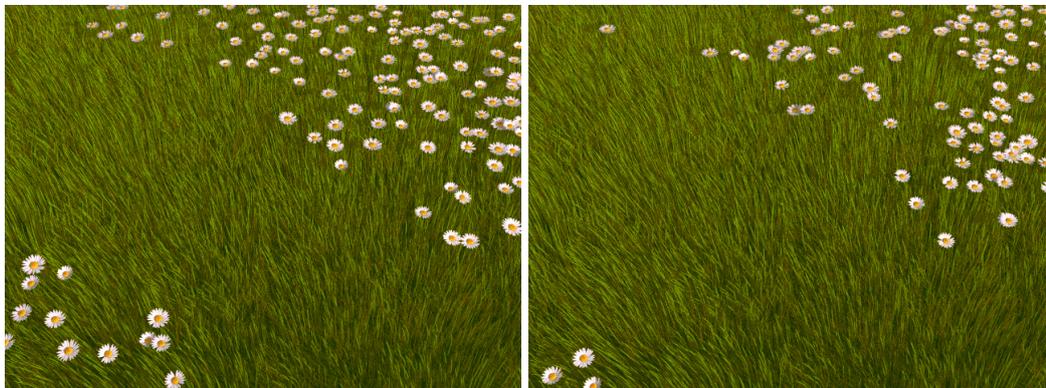
(b) Low clustering ($H \approx 1.2$).(c) Higher clustering ($H \approx 1.9$).

Figure 5.6: Plant distributions created from a user-defined density map.

The distributions shown in Figure 5.5 are created by initializing the probability density function f to a constant value. If, instead, we initialize f to a user-defined field (with a paint program, for example), we can generate spatial distributions of plants that conform to this field, as shown in Figure 5.6. This result improves on the methodology created in [DHL⁺98], which allowed for the application of a user-specified density map, but did not make it possible to control the degree of plant clustering.

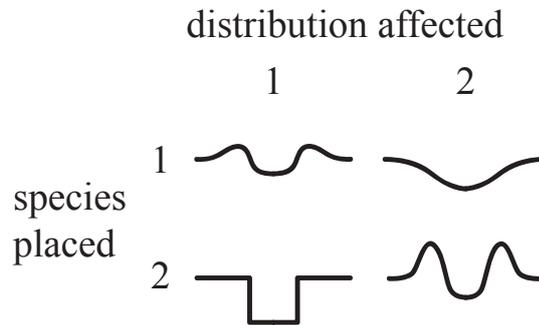


Figure 5.7: The concept of the kernel matrix M for two species.

5.4 Extensions

The deformation kernel method can be extended to include information about the plants' sizes, as well as to model the interaction of several species.

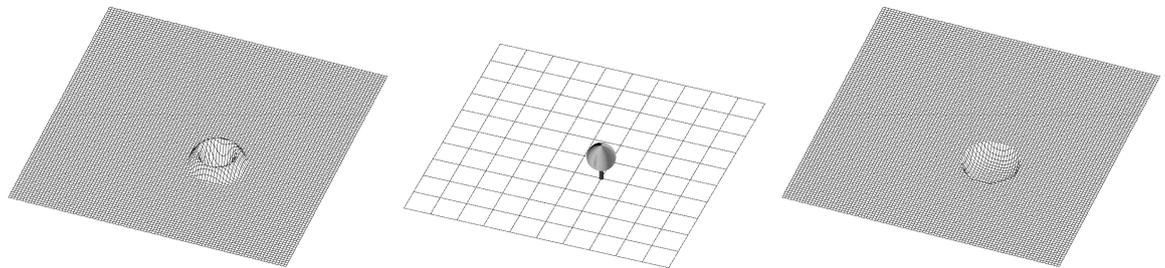
The sizes of the plants can be drawn from a distribution which fits the size hierarchy observed in nature (Section 2.3). This information is taken into account while generating the distribution by placing plants in order of size, largest first. Larger, older plants then affect the positioning of smaller, younger plants, as is to be expected.

To incorporate multispecies information, it is noted that plants of species 1 may have a different effect on other plants of species 1 than they do on plants of species 2. Different kernels are thus required to capture these different effects. In fact, for two species, a total of four kernels are required: one for the effects species 1 has on itself, one for the effects species 1 has on species 2, one for the effects species 2 has on species 1, and one for the effects species 2 has on itself. This leads, in the general case, to a *kernel matrix* M (Figure 5.7), which defines the effects that each species has on itself and on each other species.

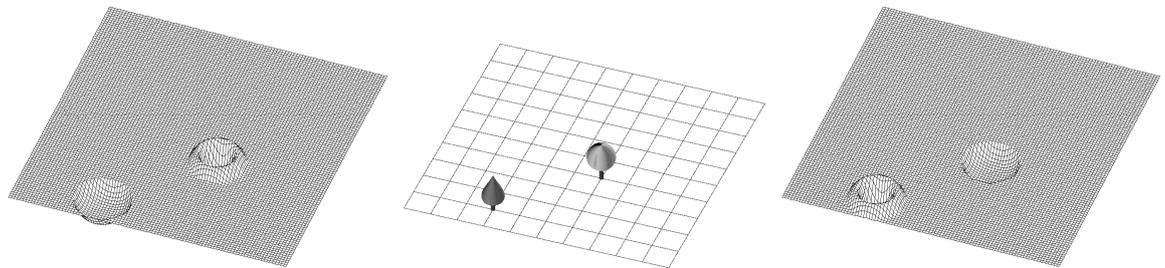
In an extension of the deformation kernel method to n species, we keep track of n

different probability density functions f_i . As plants are placed, each probability density function is deformed by the relevant kernel; thus, if a plant of species j is placed, each function f_i is deformed by kernel $M_{i,j}$, where $i = 1, 2, \dots, n$. This process is illustrated in Figure 5.8.

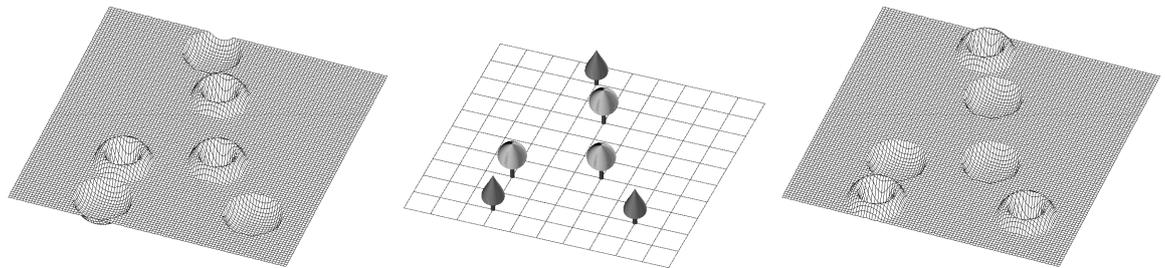
A full-scale instance of these techniques is the forest community model shown in Figure 5.9, with tree models generated using the method of Chapter 6.



(a) After placing the first plant of species 1

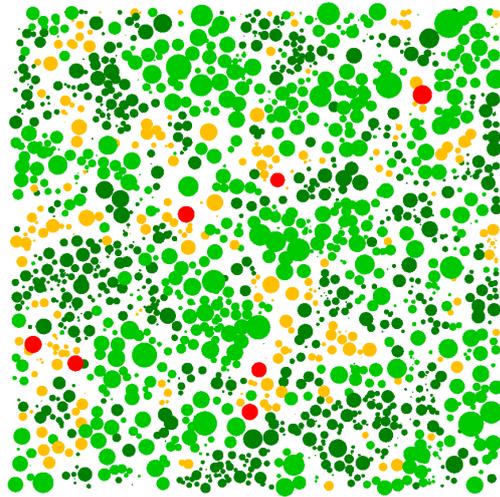


(b) After placing the first plant of species 2



(c) After placing several plants of each species

Figure 5.8: An example of the deformation kernel algorithm with two species. On the left, the probability density function of species 1; on the right, that of species 2.



(a)



(b)

Figure 5.9: A forest community distribution created by the multispecies kernel method. Top: the distribution created, using dots to represent trees. Bottom: synthetic images obtained by placing tree models at the locations generated by the coarse-level simulation.

CHAPTER 6

MULTILEVEL MODELING OF PLANT COMMUNITIES

The methods described in Chapters 4 and 5 create spatial distributions of plants within plant communities. These spatial distributions are ideally used in conjunction with some individual plant models to create a multilevel model of plant communities [DHL⁺98]. This multilevel model allows us to depict plant scenes in greater detail than the higher-level spatial distribution models alone, and to efficiently model scenes of much greater complexity than lower-level individual models alone.

Like the community models, individual plant models can be categorized into local-to-global and global-to-local models. Local-to-global approaches, in which a plant develops according to a series of rules and an initial state, were used in the multilevel models of [DHL⁺98]. Unfortunately, local-to-global plant models suffer from the same problem of *lack of control* as their counterpart community models. This is problematic in the case of multilevel models, because it is the plant's global characteristics that are specified by the elements of the higher-level model. Thus, it seems that global-to-local plant models are far more useful in the business of multilevel modeling.

Global-to-local plant models are discussed in depth in [PMKL01]. That paper includes a model of a plant based on its *silhouette curve* (Figure 6.1).

L-system 6.1:

Axiom: A(0,0)

1. $A(o,s) : o < \text{MAX_ORDER} \ \&\& \ s < \text{max_len}[o]$
 $\{ \text{rel} = s/\text{max_len}[o]; \} \rightsquigarrow$
 $\#(\text{int_width}(o,\text{rel})) \text{F}(\text{int_len}(o,\text{rel}))$
 $[+(\text{branch_ang}(o,\text{rel})) \text{A}(o+1,\text{max_len}[o+1] - \text{branch_len}(o,\text{rel}))]$
 $/(\text{phyllo_ang}[o]) \text{A}(o,s+\text{int_len}(o,\text{rel}))$
2. $A(o,s) : s \geq \text{max_len}[o] \rightsquigarrow \sim \mathbf{K}$

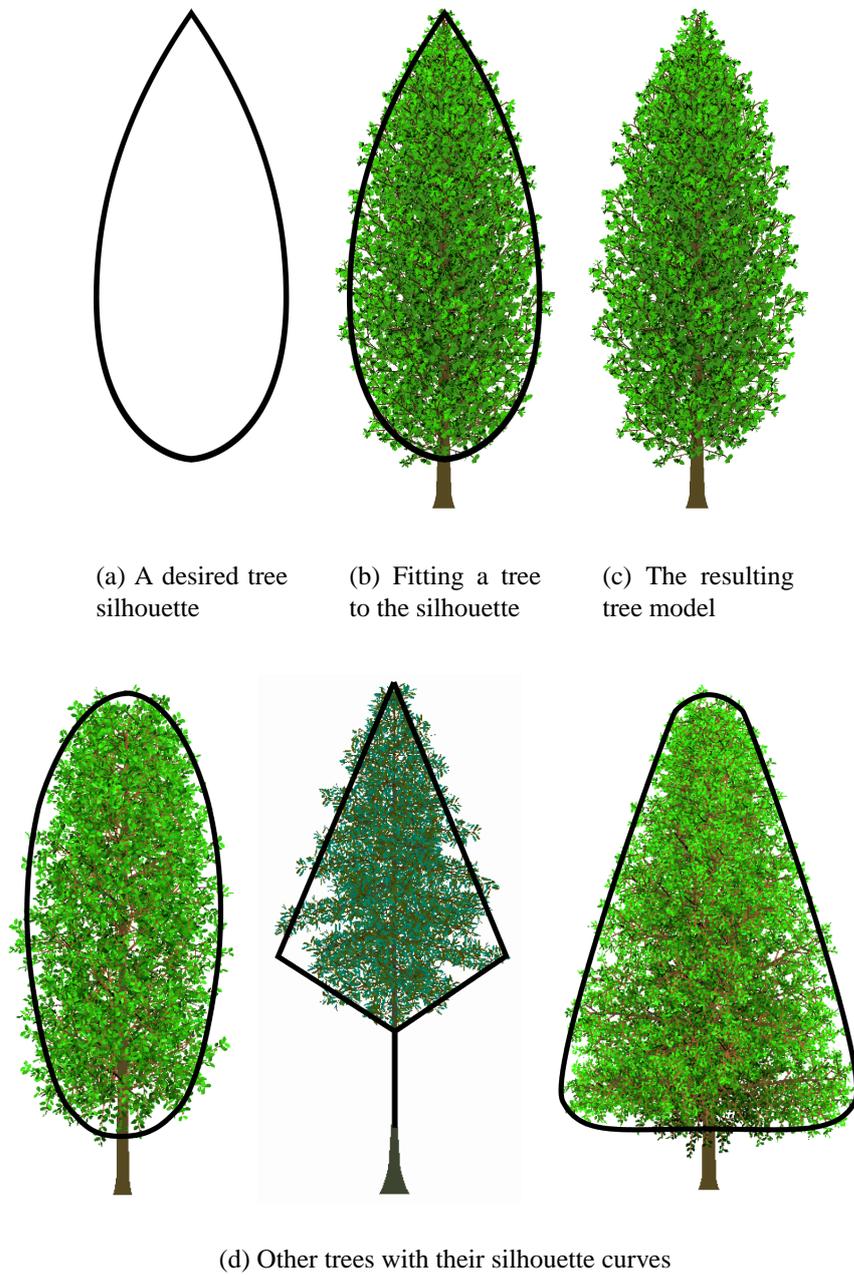


Figure 6.1: Trees generated by L-system 6.1 and their defining silhouette curves.

L-system 6.1 is based on the concept of *branch mapping*. Under this assumption, short branches are identical to the top parts of longer branches of the same order. A series of functions define the shape of the tree: $\text{max_len}[o]$ is the length of the longest branch of order o ; $\text{int_len}(o, \text{rel})$ and $\text{int_width}(o, \text{rel})$ define the length and width of internodes of order o at relative position rel along the parent branch; $\text{branch_ang}(o, \text{rel})$ defines the branching angle of a new branch; and $\text{branch_len}(o, \text{rel})$ defines its length. The constant $\text{phyllo_ang}[o]$ is the *phyllotactic angle*, which defines the rotation about the branch between successive internodes of order o .

L-system 6.1 can then be understood as follows: The module $A(o, s)$ represents a branch of order o and length $\text{max_len}[o] - s$. Thus, the axiom represents a ‘branch’ of order 0 (that is, a trunk) of length $\text{max_len}[0]$, which is the height of the tree. Production 1 is activated only if the branch is of order less than the maximum and has not yet reached its maximum length. The local variable rel is set to the relative position of s from 0 to $\text{max_len}[o]$. The production then creates an internode of width $\text{int_width}(o, \text{rel})$ and length $\text{int_len}(o, \text{rel})$; a branch of order $o + 1$ and length $\text{branch_len}(o, \text{rel})$; and, finally, a rotation by the phyllotactic angle $\text{phyllo_ang}[o]$ and a continuation of itself, of the same order but $\text{int_len}(o, \text{rel})$ shorter. The second production turns the tip of every branch into a leaf K.

If the branches are relatively straight, the silhouette of the tree is provided by the function $\text{branch_len}(0, \text{rel})$, and the height and width of the tree can be said to be $\text{max_len}[0]$ and $\text{max_len}[1]$, respectively. This provides a useful set of global parameters to match to the parameters provided by the higher-level model.

In the case of the models described in Chapters 4 and 5, the only parameters provided to a plant are its position, size, and species. The actual height and width of the tree object which is generated are related to the size r through a power function [Nik94]:

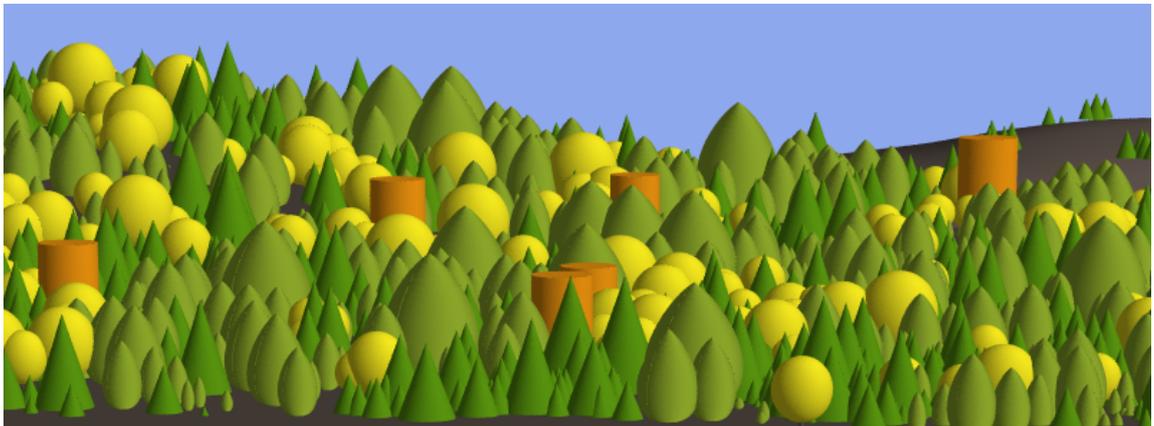
$$H = A_H r^{c_H},$$
$$W = A_W r^{c_W}$$

where A_H , A_W , c_H , and c_W are dependent on the species.

The plant models created in Chapters 4 and 5 are quite simply created using L-system 6.1. A “species” is created manually as a given silhouette specification and set of parameters. For instance, the “poplar” model of Figure 4.6 has an ellipsoid silhouette; the “conifer” model of the same image has a conical silhouette, and, in addition, has the phyllotactic angle of first-order branches set to 180° , creating the horizontal branch tiers characteristic of such trees.

In theory, every plant could be created individually; however, given the size of the geometric description of even a single tree model (up to sixty megabytes), some form of *approximate instancing* [DHL⁺98] is required. At most, a few dozen plant objects are used in the images shown in this thesis (See Table 7.1). A k-means clustering algorithm [LL83] is used to find a small number of exemplar sizes which are representative of each species.

As an illustration of the entire process, we consider the steps taken to produce Figure 5.9(b) from Figure 5.9(a). A silhouette and set of parameters are modeled by hand for each species. Ten exemplar sizes are chosen for each species and each plant in the scene is replaced by one of the exemplar sized plants. Figure 6.2(a) shows the scene from Figure 5.9 rendered using the silhouette shapes instead of realistic tree models. Finally, ten plant objects of each species are created, filling the silhouettes, and placed at the required positions (Figure 6.2(b)).



(a) With silhouette shapes



(b) With realistic tree models

Figure 6.2: A comparison between silhouette shapes and realistic tree models in the rendering of the distribution of Figure 5.9.

CHAPTER 7

CONCLUSIONS

Statistics pertinent to the scenes shown in this thesis are given in Table 7.1. The times taken to generate the spatial distribution vary greatly, from under a second (for the deformation kernel models) to nearly an hour (for one of the local-to-global models). It must be remembered, however, that the local-to-global models create an entire history of the spatial distribution, while the global-to-local models create the spatial distribution of one moment in time. Each timestep of the local-to-global models takes, on average, under three seconds.

None of the images took, from the start of the spatial distribution generation to the end of rendering, more than an hour to create. Once the form of the individual plant models for each species are created, the entire task, except for lighting and other aesthetic considerations in rendering, is automated. This makes the process suitable for the practical application to realistic image synthesis.

Possible further directions for this research include various practical applications, such as scene dressing for computer animation purposes and visual impact analysis of tree cutting and regrowth. The realism of the resulting scenes can be further improved using more accurate models of the underlying biological processes and more sophisticated rendering methods. Finally, the idea of the deformation kernel is not inherently a global-to-local one; it should be possible to create a local-to-global model of the spatial distribution of plants using deformation kernels.

Fig.	Number of			Time taken ^a		
	plants	different plants	primitives (millions)	distribution generation ^b	plant generation	rendering ^c
4.5a	2688	20	34	9 sec	30 sec	8 min
b	1593	20	18	72 sec	27 sec	8 min
c	1409	20	30	2 min	30 sec	9 min
d	834	20	78	3 min	40 sec	11 min
4.6a	5584	24	141	2 min	30 sec	15 min
b	5526	24	147	22 min	30 sec	14 min
c	3427	24	100	44 min	30 sec	12 min
5.5	100	8	1.5	0.05 sec	* ^d	2 min
6.2	1599	36	147	0.5 sec	65 sec	11 min

^aTimes recorded on a 733 MHz Pentium III processor.

^bFor Figures 4.5 and 4.6, the times given show how long the simulation took to reach the given frame.

^cRaytraced with `rayshade`, 9 samples per pixel. For Figures 4.5, 4.6, and 6.2, the image was 1024×512 pixels; for Figure 5.5, the image was 640×480 pixels.

^dThe plant models for Figure 5.5 were manually generated and randomly assigned.

Table 7.1: Statistics pertinent to Figures 4.3, 4.5, 4.6, 5.5, and 6.2.

Final words

This thesis addresses several open problems in the modeling of plant communities for image synthesis. First, the distinction between local-to-global and global-to-local models originally made in [PMKL01] is extended to models of plant communities, and new modeling methodologies are applied to each. The formalism of multiset L-systems provides a framework for individual-based local-to-global models of plant communities; the global-to-local deformation kernel method allows greater control over the decomposition of an initial density function.

Second, the global-to-local plant modeling techniques described in [PMKL01] are combined with the spatial distribution models to create multilevel models of plant communities. The global-to-local nature of the individual plant model allows tighter coupling between the two levels of the model. The size, shape, and species of each plant in the

higher-level model can be reflected in the individual plant model.

Third, these methods have been illustrated using examples with ecological relevance. Local-to-global models expressed in multiset L-systems captured the phenomena of self-thinning, succession, and clustering. The deformation kernel method is used to create models with variable clustering and multiple interacting species. Finally, the methodologies have been applied to image synthesis with the creation of realistic visualizations of these models.

BIBLIOGRAPHY

- [ACVP00] Pierre Auger, Sandrine Charles, Muriel Viala, and Jean-Christophe Pogiale. Aggregation and emergence in ecological modeling: integration of ecological levels. *Ecological modeling*, 127:11–20, 2000.
- [AW80] D. P. Aikman and A. B. Watkinson. A model for growth and self-thinning in even-aged monocultures of plants. *Annals of Botany*, 45:419–427, 1980.
- [CE54] Philip J. Clark and Francis C. Evans. Distance to nearest neighbor as a measure of spatial relationships in populations. *Ecology*, 35:445–453, 1954.
- [CMDH97] Norishige Chiba, Kazunobu Muraoka, Akio Doi, and Junya Hosokawa. Rendering of forest scenery using 3D textures. *Journal of Visualization and Computer Animation*, 8(4):191–199, 1997.
- [DHL⁺98] Oliver Deussen, Pat Hanrahan, Bernd Lintermann, Radomír Měch, Matt Pharr, and Przemyslaw Prusinkiewicz. Realistic modeling and rendering of plant ecosystems. *Proceedings of the 25th annual conference on computer graphics and interactive techniques*, pages 275–286, 1998.
- [Dig83] Peter J. Diggle. *Statistical Analysis of Spatial Point Patterns*. Mathematics in Biology. Academic Press, Toronto, 1983.
- [DKY⁺00] Yoshinori Dobashi, Kazufumi Kaneda, Hideo Yamashita, Tsuyoshi Okita, and Tomoyuki Nishita. A simple, efficient method for realistic animation of clouds. In *Proceedings of the 27th annual conference on Computer graph-*

- ics and interactive techniques*, pages 19–28. ACM Press/Addison-Wesley Publishing Co., 2000.
- [DLDB97] D. H. Deutschman, S. A. Levin, C. Devine, and L. A. Buttel. Scaling from trees to forests: Analysis of a complex simulation model. *Science*, 277, 1997.
- [EN01] Brian J. Enquist and Karl J. Niklas. Invariant scaling relations across tree-dominated communities. *Nature*, 410:655–660, 2001.
- [Fea00] Paul Fearing. Computer modelling of fallen snow. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 37–46. ACM Press/Addison-Wesley Publishing Co., 2000.
- [FR86] Alain Fournier and William T. Reeves. A simple model of ocean waves. In *Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, pages 75–84. ACM Press, 1986.
- [FS75] R. W. Floyd and L. Steinberg. An adaptive algorithm for spatial greyscale. *Proceedings of the Society for Information Display*, 17:75–77, 1975.
- [FW85] F. G. Firbank and A. R. Watkinson. A model of interference within plant monocultures. *Journal of Theoretical Biology*, 116:291–311, 1985.
- [Gar85] Geoffrey Y. Gardner. Visual simulation of clouds. In *Proceedings of the 12th annual conference on Computer graphics and interactive techniques*, pages 297–304. ACM Press, 1985.

- [GBS02] Peter Guttorp, David R. Brillinger, and Frederic Paik Schoenberg. Point processes, spatial. In Abdel H. El-Shaarawi and Walter W. Piegorsch, editors, *Encyclopedia of environmetrics*, pages 1571–1573. Wiley, 2002.
- [GGCC97] Christophe Godin, Yann Guédon, Evelyne Costes, and Yves Caraglio. Measuring and analysing plants with AMAPmod software. In Marek T. Michalewicz, editor, *Plants to ecosystems: Advances in computational life sciences*, pages 53–84. CSIRO Publishing, 1997.
- [GMPVG00] H el ene Gautier, Radom ır M ech, Przemyslaw Prusinkiewicz, and Claude Varlet-Grancher. 3D architectural modelling of aerial photomorphogenesis in white clover (*Trifolium repens* L.) using L-systems. *Annals of Botany*, 85:359 – 370, 2000.
- [Han92] J. S. Hanan. *Parametric L-systems and their application to the modelling and visualization of plants*. PhD thesis, University of Regina, June 1992.
- [Har77] John L. Harper. *Population Biology of Plants*. Academic Press, 1977.
- [Hop54] Brian Hopkins. A new method for determining the type of distribution of plant individuals. *Annals of Botany*, XVIII:213–226, 1954.
- [Lin68] Aristid Lindenmayer. Mathematical models for cellular interaction in development, Parts I and II. *Journal of Theoretical Biology*, 18:280–315, 1968.
- [LK87] Jan Lep s and Pavel Kindlmann. Models of the development of spatial pattern of an even-aged plant population over time. *Ecological Modelling*,

- 39:45–57, 1987.
- [LL83] Louis Legendre and Pierre Legendre. *Numerical Ecology*. Elsevier, Amsterdam, 1983.
- [LP02] Brendan Lane and Przemyslaw Prusinkiewicz. Generating Spatial Distributions for Multilevel Models of Plant Communities. In *Proc. Graphics Interface*, pages 69–80, May 2002.
- [Max81] Nelson L. Max. Vectorized procedural models for natural terrain: Waves and islands in the sunset. In *Proceedings of the 8th annual conference on Computer graphics and interactive techniques*, pages 317–324, 1981.
- [MP96] Radomír Měch and Przemyslaw Prusinkiewicz. Visual models of plants interacting with their environment. *Proceedings of the 23th annual conference on computer graphics and interactive techniques*, pages 397–410, August 1996.
- [MWC80] Robert Marshall, Rodger Wilson, and Wayne Carlson. Procedure models for generating three-dimensional terrain. In *Proceedings of the 7th annual conference on Computer graphics and interactive techniques*, pages 154–162, 1980.
- [Ney96] Fabrice Neyret. Synthesizing verdant landscapes using volumetric textures. In Xavier Pueyo and Peter Schröder, editors, *Eurographics Rendering Workshop 1996*, pages 215–224, New York City, NY, June 1996. Eurographics, Springer Wein.

- [Nik94] Karl J. Niklas. *Plant allometry: the scaling of form and process*. University of Chicago Press, Chicago, 1994.
- [PCS93] S. W. Pacala, C. D. Canham, and J. A. Silander. Forest models defined by field measurements: I The design of a northeastern forest simulator. *Canadian Journal of Forest Research*, 23:1980–1988, 1993.
- [PD95] S. Pacala and D. H. Deutschman. Details that matter: The spatial distribution of individual trees maintains forest ecosystem function. *Oikos*, 74:357–365, 1995.
- [Per94] David A. Perry. *Forest Ecosystems*. The Johns Hopkins University Press, 1994.
- [PH90] P. Prusinkiewicz and J. Hanan. Visualization of botanical structures and processes using parametric L-systems. In D. Thalmann, editor, *Scientific visualization and graphics simulation*, pages 183–201. J. Wiley & Sons, Chichester, 1990.
- [PKMH00] Przemyslaw Prusinkiewicz, Radoslaw Karwowski, Radomír Měch, and Jim Hanan. L-studio/cpfg: A software system for modeling plants. In M. Nagl, A. Schürr, and M. Münch, editors, *Applications of graph transformation with industrial relevance*, Lecture Notes in Computer Science 1779, pages 457–464. Springer-Verlag, Berlin, 2000.
- [PL90] Przemyslaw Prusinkiewicz and Aristid Lindenmayer. *The Algorithmic Beauty of Plants*. Springer-Verlag, New York, 1990. With J. S. Hanan, F. D. Fracchia, D. R. Fowler, M. J. M. de Boer, and L. Mercer.

- [PM01] Yoav I. H. Parish and Pascal Müller. Procedural modeling of cities. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 301–308. ACM Press, 2001.
- [PMKL01] Przemyslaw Prusinkiewicz, Lars Mündermann, Radoslaw Karwowski, and Brendan Lane. The use of positional information in the modeling of plants. *Proceedings of the 28th annual conference on computer graphics and interactive techniques*, pages 289–300, 2001.
- [Pru86] Przemyslaw Prusinkiewicz. Graphical applications of L-systems. In *Proceedings of Graphics Interface '86 — Vision Interface '86*, pages 247–253, 1986.
- [PS85] Stephen W. Pacala and John A. Silander. Neighborhood models of plant population dynamics: I. Single-species models of annuals. *The American Naturalist*, 125:385–411, 1985.
- [RB85] William T. Reeves and Ricki Blau. Approximate and probabilistic algorithms for shading and rendering structured particle systems. *Proceedings of the 12th annual conference on computer graphics and interactive techniques*, pages 313–322, July 1985.
- [Rey87] Craig W. Reynolds. Flocks, herds and schools: A distributed behavioral model. In *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pages 25–34. ACM Press, 1987.
- [RL74] G. Rozenberg and K. P. Lee. Developmental systems with finite axiom sets.

- Part I. Systems without interactions. *International Journal of Computer Mathematics*, 4:43–68, 1974.
- [Ros97] S. M. Ross. *Introduction to Probability Models*. Academic Press, 1997.
- [RRS76] G. Rozenberg, K. Ruohonen, and A. Salomaa. Developmental systems with fragmentation. *International Journal of Computer Mathematics*, 5:177–191, 1976.
- [Sin85] Dennis F. Sinclair. On tests of spatial randomness using mean nearest neighbor distance. *Ecology*, 66:1084–1085, 1985.
- [WW92] Alan Watt and Mark Watt. *Advanced animation and rendering techniques : theory and practice*. ACM Press, 1992.
- [YKOH63] K. Yoda, T. Kira, H. Ogawa, and K. Hozumi. Self-thinning in overcrowded pure stands under cultivated and natural conditions. *Journal of Biology Osaka City University*, 14:107–129, 1963.