

Simulation of insect movement with respect to plant architecture and morphogenesis

Jim Hanan¹, Przemyslaw Prusinkiewicz², Myron Zalucki³, David Skirvin⁴

¹ Centre for Plant Architecture Informatics, The University of Queensland, St. Lucia, Qld 4072, Australia

² Department of Computer Science, University of Calgary, Calgary, Canada T2N 1N4

³ Department of Zoology and Entomology, The University of Queensland, St. Lucia, Qld 4072, Australia

⁴ Horticulture Research International, Wellesbourne, CV 35 9EF, UK

Abstract

Developments in computer and three dimensional (3D) digitiser technologies have made it possible to keep track of the broad range of data required to simulate an insect moving around or over the highly heterogeneous habitat of a plant's surface. Properties of plant parts vary within a complex canopy architecture, and insect damage can induce further changes that affect an animal's movements, development and likelihood of survival. Models of plant architectural development based on Lindenmayer systems (L-systems) serve as dynamic platforms for simulation of insect movement, providing an explicit model of the developing 3D structure of a plant as well as allowing physiological processes associated with plant growth and responses to damage to be described and simulated. Simple examples of the use of the L-system formalism to model insect movement, operating at different spatial scales—from insects foraging on an individual plant to insects flying around plants in a field—are presented. Such models can be used to explore questions about the consequences of changes in environmental architecture and configuration on host finding, exploitation and its population consequences. In effect this model is a 'virtual ecosystem' laboratory to address local as well as landscape-level questions pertinent to plant-insect interactions, taking plant architecture into account.

Reference

J. Hanan, P. Prusinkiewicz, M. Zalucki, D. Skirvin: Simulation of insect movement with respect to plant architecture and morphogenesis. *Computers and Electronics in Agriculture* 35, pp. 256–269.

Simulation of insect movement with respect to plant architecture and morphogenesis

Jim Hanan ^{a,*}, Przemyslaw Prusinkiewicz ^b, Myron Zalucki ^c,
David Skirvin ^d

^a *Centre for Plant Architecture Informatics, The University of Queensland, St. Lucia, Qld 4072, Australia*

^b *Department of Computer Science, University of Calgary, Calgary, Canada T2N 1N4*

^c *Department of Zoology and Entomology, The University of Queensland, St. Lucia, Qld 4072, Australia*

^d *Horticulture Research International, Wellesbourne, CV35 9EF, UK*

Abstract

Developments in computer and three dimensional (3D) digitiser technologies have made it possible to keep track of the broad range of data required to simulate an insect moving around or over the highly heterogeneous habitat of a plant's surface. Properties of plant parts vary within a complex canopy architecture, and insect damage can induce further changes that affect an animal's movements, development and likelihood of survival. Models of plant architectural development based on Lindenmayer systems (L-systems) serve as dynamic platforms for simulation of insect movement, providing an explicit model of the developing 3D structure of a plant as well as allowing physiological processes associated with plant growth and responses to damage to be described and simulated. Simple examples of the use of the L-system formalism to model insect movement, operating at different spatial scales—from insects foraging on an individual plant to insects flying around plants in a field—are presented. Such models can be used to explore questions about the consequences of changes in environmental architecture and configuration on host finding, exploitation and its population consequences. In effect this model is a 'virtual ecosystem' laboratory to address local as well as landscape-level questions pertinent to plant–insect interactions, taking plant architecture into account. © 2002 Elsevier Science B.V. All rights reserved.

Keywords: Individual-based modelling; Insect movement; L-systems; Plant–insect interactions

* Corresponding author

E-mail address: jjim@cpai.uq.edu.au (J. Hanan).

1. Introduction

Ecologists, evolutionary biologists and entomologists interested in plant–insect interactions have by and large ignored the three dimensional (3D) structure or architecture of plants, and the explicit movements of the insects over the plant surface, even though this is of express (e.g. Edwards and Wratten, 1983) or at least implicit interest (e.g. Wilson and Waite, 1982). The same is true for movements, usually of the adult insect, among plants (e.g. Zalucki, 1983). The environment in which an insect finds itself seems extremely complicated when one considers the details of the branching architecture of some plants, and even more so if one considers the added diversity of ages and conditions of the individual components. Analysis of patterns of damage within such structures is even more complex (Alonso and Herrera, 1996; Neuvonen, 1999). In addition, habitats in the plant canopy are very diverse, affected by such factors as variable amounts of shade and protection from wind, rain, and the watchful eyes of predators. However, advances in computer and 3D digitising technologies make it possible to keep track of a broad range of this type of information (Hanan and Room, 1997). Mathematical plant models have been successfully used to simulate the developing 3D structure of a plant, as well as physiological processes driving plant growth (Prusinkiewicz, 1998). These models explicitly describe the relative 3D location of changing plant parts and capture the spatial heterogeneity of component properties. Information flow through the plant structure can be used to model physiological effects such as hormone flow and induced defences. The resulting structural models can be used to infer information that is not easy to measure, such as light intensity in various parts of a plant canopy (Chelle and Andrieu, 1998).

In this paper, we show how mathematical models of plant architecture can serve as a dynamic platform for simulation of insect movement and associated behaviours, such as feeding, on and around plants. The approach uses the plant modelling system's ability to transfer information through the simulated structure to represent insect movement on the plant's surface (Prusinkiewicz et al., 1997). Coordination of the insect model with the plant model makes it easy to simulate plant responses to damage, such as release of apical dominance when a meristem is removed, or production of induced defences. The integrated models can be extended to tri-trophic interactions by modelling predators searching for prey on arrays of virtual plants, as well as oviposition behaviour by adult butterflies moving between host plants growing in a virtual field. The latter step closes the loop. Virtual eggs that have been laid on virtual plants hatch to become virtual caterpillars damaging the plants and being subject to predation by virtual predators. Damaged plants age and change as a result of induced changes (among other things), and new plants germinate and become available for oviposition by the next generation of butterflies. These are the first steps in creating a 'virtual ecosystem' laboratory to address local landscape-level questions in ecology in a realistic way. The techniques described here give a focus for researchers to make the painstaking effort required to collect the data necessary to examine questions at this little-studied scale of insect-plant interactions.

This paper will first give a background on how plant models can be derived for these purposes, then look at how insect movement can be simulated, both on and around the plants. An example incorporating reaction of a plant to insect damage will be presented, and avenues for further research discussed.

2. Methods

2.1. Modelling plant architecture

The modular nature of plants makes it possible to easily represent plant structure and development (Room et al., 1994). Plants are built by repeated addition of units of construction, each typically consisting of a segment of stem, or internode, supporting one or more leaves, with an axillary bud or buds at the base of each leaf. Static models of a plant can be produced by measurement of these components and their relative positions, followed by reconstruction of the plant's architecture by representation of these measurements in a computer. In many situations, it is preferable to use a more dynamic platform for our insect models, since plant responses can affect future activity of the insects. A variety of techniques exist for creating developmental models of plants (Hanan and Room, 1997). Usually, a series of measurements is made to determine the timing of the production of plant components by the plant's apical meristems, which depends on the prevalent conditions and resources available to the plant. Additionally, component growth is characterised by analysis of repeated measurements of component size, which yields growth functions to be applied in the model. Alternatively, estimates can be made based on the history captured in the final structure of a plant. The pattern of growth and development is then expressed using one of a number of modelling approaches (for a review, see Prusinkiewicz, 1998). For the purposes of this discussion we will focus on the Lindenmayer system (L-system) formalism (Lindenmayer, 1968; Prusinkiewicz and Lindenmayer, 1990), since it provides a generic tool for modelling any plant.

In the L-system formalism, a module, a symbol with an optional list of numerical parameters, represents each type of plant component. For instance, a leaf might be represented by the module **L(2,3.5)**, the symbol **L** with associated parameters for age and length, while an internode could be represented by the module **I(2.5)**, the symbol **I** with a parameter for length. The plant structure is then represented as a string of these modules, with their relative position determining their neighbourhood in the modelled plant structure (Fig. 1).

The lateral position of components such as leaves and branches is represented by enclosing the symbols making up the component in square brackets and placing them between the symbols neighbouring the point of attachment of the lateral structure. For instance, the string **I[L]I** would represent two internodes **I**, with a leaf **L** attached at the node between them. Geometric information is incorporated by including auxiliary symbols that capture the relative positioning of a component compared to its predecessor. For instance, in the string **I[(30)L]I** the symbol and

parameter $\alpha(30)$ cause the leaf **L** to be pitched down by an angle of 30° from the parent internode. Details of geometric aspects of plant modelling can be found in Prusinkiewicz and Lindenmayer (1990), and Prusinkiewicz et al. (1997).

Development of the plant and growth of components is modelled using rules called ‘productions’. In each time step of the simulation, the rule applicable to each component is applied. These rules take the form:

predecessor: condition → *successor*

where the *predecessor* is a ‘formal’ module (i.e. a module with variable names standing for the associated parameters), *condition* is a logical statement including constants and variable names that must evaluate as true for the production to be applied, and *successor* is a list of modules to replace the given predecessor in the string representing the plant. For instance, a leaf’s growth may be captured by a rule such as:

L(age,length): (age < 5) → L(age + 1,length*1.10)

which means that a leaf would grow exponentially, increasing in length by 10% in each time step, until it reaches the age of 5. Thus, given a string

I[L(5,1.54)][B]I[L(4,1.4)][B]I

with no other rules for internode **I** and bud **B**, the string in the following step will be

I[L(5,1.54)][B]I[L(5,1.54)][B]I

In a plant, development is typically controlled by apical meristems (or apices) that produce metamers, units of construction composed of an internode, leaf and axillary bud.

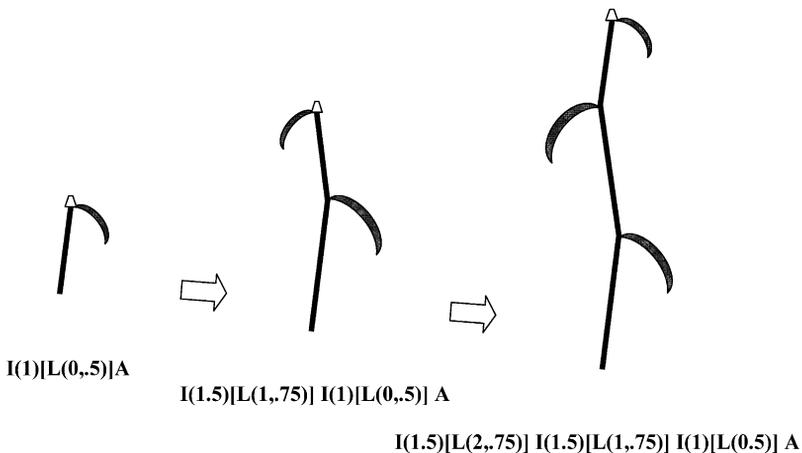


Fig. 1. Graphic visualisation and corresponding L-system string for three successive stages in a plant’s development.



Fig. 2. Three stages in a plant's development.

This can be captured by a rule such as $A \rightarrow I(1)[L(0,1)][B]A$ where the apex **A** produces an internode **I**, leaf **L**, axillary bud **B** and continuing apex **A** in each time step.

As an example of how the L-system approach makes it possible to simulate dynamic changes in an insect's environment, let us suppose that the feeding quality of leaf tissue declines with leaf age, leaves of age 4 or more being considered inedible. The following L-system keeps track of the age, and therefore quality of leaves in the plant:

p1: $S \rightarrow I[C][C]A$

p2: $A \rightarrow I[L(0)][B]A$

p3: $B \rightarrow A$

p4: $L(\text{age}) \rightarrow L(\text{age} + 1)$

Here, **S** represents the seed, **A** an apical meristem, **B** an immature bud, **L** a leaf with parameter **age**, **I** an internode, and **C** a cotyledon. Production **p1** captures the transformation of the seed to a stem segment **I** (the hypocotyl), two cotyledons **C** and the apical meristem **A**. Production **p2** controls the production of a new metamer comprised of internode **I**, leaf **L** and axillary bud **B** by the apex. Production **p3** describes the development of an immature axillary bud into a mature apex, and production **p4** captures the aging process of the leaves. Three stages from the sequence of plant development beginning from seed **S** can be seen in Fig. 2 with varying quality of leaves highlighted.

Whereas the formalism outlined above makes it possible to model plant development controlled by lineage, endogenous control mechanisms such as photosynthate or hormone flow in the plant structure require the use of context-sensitive L-systems (Lindenmayer, 1982). In these systems, application of a production can depend on the neighbouring symbols (context) in the string.

Since information flow can be from the bottom to top of a plant, or from the top to the bottom, productions have the following general format:

left context < *predecessor* > *right context*: *condition* → *successor*

where the symbols < and > separate the predecessor from the strings representing the left and right context. Such a production will be applied to a symbol matching the predecessor only if the symbols on its left and right sides in the string match the left and right context, respectively. If either the left or right context is not required, they may be left out of the production specification. For instance, to use a signal to represent hormone flow up a stem, the signal will be moved into an internode once the signal has reached the internode to the left of the current internode by employing a production rule like:

I(sigleft) < I(sig) → I(sigleft)

where the parameter of the internode **I** has a value of 0 if the signal has not reached it yet, and a value of 1 if it has. Given the starting string

I(1)I(0)I(0)I(0)

representing a stem of four internodes **I**, the next three strings resulting from the application of the production would be

I(1)I(1)I(0)I(0)

I(1)I(1)I(1)I(0)

I(1)I(1)I(1)I(1)

the flow of 1s moving from left to right in the strings over time representing flow of the hormone.

The open L-system formalism (Mech and Prusinkiewicz, 1996) makes it possible to simulate interactions with the environment. An environment communication module, **?E**, introduced before a symbol representing a component causes information such as its position, orientation and size to be passed from the plant model to an environment program. These properties may be used in the simulation of environmental factors, such as the amount of light energy intercepted by individual leaves, taking surrounding vegetation into account (Gautier et al., 2000). Information is returned to the plant model via the parameters of the environment communication module, and their values can then affect subsequent production-rule application. The simulation of plant interactions with the environment is important in the context of plant-insect interactions, as it makes it possible to model microclimates such as local temperature and humidity in a plant canopy, which in turn may affect insect behaviour (e.g. Willmer et al., 1996).

2.2. Modelling insect motion

We have taken an individual-based modelling approach (DeAngelis and Gross, 1992) to the simulation of insect motion, tracking each insect's position and internal state. The general problem to be addressed is the transfer of information between

the plant and the insect. The insect model needs to know where it is on the structure, and needs to be able to detect the condition of the plant components (e.g. is this leaf edible?). The plant model needs to know what effect the insect has on it, for instance, whether, where and how much damage occurs when an insect feeds. Movements of an insect following the branching structure of a plant can be conveniently captured using the context-sensitive mechanisms of L-systems to closely couple the insect model with the plant model. An insect is represented as a signal being passed through the plant structure. This allows the information transfer issue to be handled in a straightforward manner, with the context in a production capturing insect movement holding the parameters representing the plant information that the insect can perceive, and with the context of a production pertinent to plant development holding the hungry insect that causes modification of the plant component's state by that production. A simple model of insect movement on a stem made up of a series of internodes can be stated as follows:

```
# define UP 1
# define DOWN -1
Axiom: S(UP) I I I A
p1: S(direction) < I: direction == UP → IS(direction)
p2: I > S(direction): direction == DOWN → S(direction)I
p3: S(direction) > A: direction == UP → S(DOWN)
p4: S(direction) < A → X
p5: S(direction) → *
```

In this L-system the symbol **I** represents an internode, **A** represents an apex, **X** a damaged apex and **S** represents the insect. Properties of the individual insect are represented by parameters of the **S**, in this case the single parameter specifying the insect's direction of movement on the plant. The **# define** statements specify potential values of this parameter, the constants **UP** and **DOWN** (values 1 and -1 respectively), representing movement up in the plant (to the right in the string) and down in the plant (to the left in the string). Production **p1** captures movement up the stem, while production **p2** captures movement down. Productions **p3** and **p4** capture the insect's perception and damage of the apex, and the change of its direction state initiating downward motion. **P5** is a housekeeping production that removes the insect from its previous position once it has moved. The sequence of strings produced by the L-system and the resulting graphical representation of an insect moving up and down a stem are shown in Fig. 3.

Modelling of movement of insects between plants or between branches on the same plant is more involved as the insect must perceive and then move to a plant component that is not represented by an immediately neighbouring module in the string representing the plant. This requires use of the open L-system formalism, with all components that an insect may move on being exported to the environment. As an example, consider the case where the insect can move between components that touch, but are not directly connected. When the insect model determines that it should move, a call to the environment program is made specifying the current component and the insect's position on it. If the environment program determines that no other component touches the current component a flag is returned indicating this and the

insect model must determine another action. Otherwise the touching component is notified by the environment that an insect is coming across, the insect is removed from its current position and introduced to the string at the new location. Assuming that each plant component has an associated environment communication module ?E, the following L-system fragment handles these operations.

```

# define UP 1
# define DOWN -1
# define CROSS 2
# define NOACTION 0
# define CHECK 1
# define GONE 1
# define ARRIVING 2
# define REMAINING 3
p1:S(state) < ?E(request,response):state == CROSS&&request == NoAction
    → ?E(Check,NoAction)
p2:S(state) > ?E(request,response):state == CROSS&&response == GONE → *
p3:?E(request,response):response == ARRIVING
    → S(DOWN)?E(NoAction,NoAction)
p4:S(state) > ?E(request,response):state == CROSS&&response == REMAINING
    → S(DOWN)
p5:?E(request,response) → ?E(NoAction,NoAction)
    
```

The insect S has a parameter indicating its current state, with values of UP indicating movement towards the plant's extremities, DOWN indicating the opposite direction, and CROSS indicating movement to a touching component if possible. It is assumed that sometime after arrival at the leaf with state UP the insect state will be set to CROSS by some other production. Given that no previous

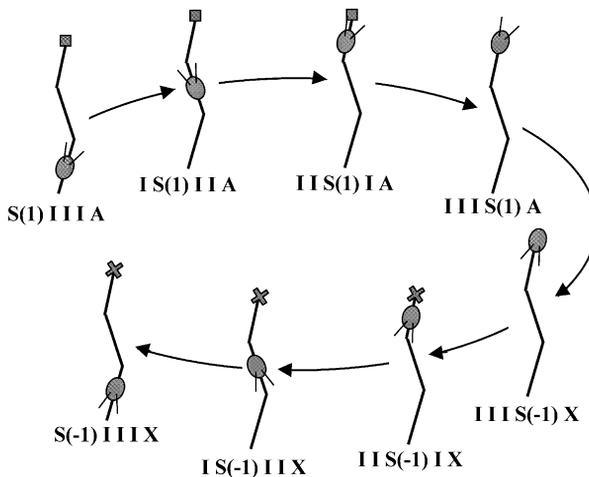


Fig. 3. A simple example of insect movement, showing consecutive strings and graphical representations.

request is being processed (i.e. request has value **NoAction**) production **p1** then sets the environment query operator's first parameter to **Check** indicating that the associated insect wishes to cross to another component if possible. The second parameter is initialised to **NoAction**. The environment program will assign a value to the query operator's **response** parameter indicating that the insect is not being moved (**REMAINING**) or has been moved (**GONE**). If the insect is moving, the environment program will set the parameter of the query operator at the new location to indicate its arrival (**ARRIVING**). Following the environment processing step, production **p2** removes the insect if it is leaving, while production **p3** introduces the insect at the new location, setting the environment operator's parameters to **NoAction** indicating that the insect doesn't want to cross. If the insect is not leaving, production **p4** chooses a new action. Production **p5** is a housekeeping production to make sure other queries are reset to **NoAction**. The following sequence of strings illustrates the process assuming that the first and second leaves **L** are touching.

...S(2)?E(0,0)L....?E(0,0)L...

...S(2)?E(1,0)L....?E(0,0)L... < after application of **p1**

...S(2)?E(1,3)L....?E(0,2)L... < after the environment step

...?E(0,0)L....S(-1)?E(0,0)L... < after application of productions **p2**, **p5**, and **p3**

Movement of insects around a field of plants by jumping or flying can also be modelled using open L-systems. In this case the insect is entirely reliant on perceptions received via the environment communication module, as the movement occurs between the plant components. The insect makes an information request equivalent to a scan of the local environment. In these cases the information returned would be direction and distance to a type of module, such as a flower. Details of such a program are too involved to be described here, but the following high-level description gives a flavour for the approach.

Find closest flower position and distance

If (insect state = = HUNGRY and distance = = 0) then

 feed

 state = SATIATED

else if (insect state = = HUNGRY and distance > 0) then

 move toward position

else if (insect state = = SATIATED) then

 move in random direction

end

Production rules would then be provided to move the insect in the desired direction, independent of the structure of the plant.

2.3. Plant reactions

Plant reactions to insect damage are many and varied, such as:

- compensatory growth (Rosenthal and Kotenan, 1994),

- induced defences resulting in an increase in distasteful or harmful chemicals in plant components (e.g. Agrawal, 1998), and
- release of volatiles that attract predators (e.g. Thaler, 1999).

The plant modelling system based on L-systems is flexible enough to handle all these responses to damage.

As a simple example we will look at a system where the plant responds by inducing defences in its leaves. Insect damage is assumed to convert the leaf module **L(age)** to a damaged leaf module **X(age,signal)** where **signal** is set to one to indicate the damage has just occurred. A constant **STRENGTH** determines how far the signal may travel. The following L-system fragment captures the damaged structure's response.

```
# define SIGNAL 1
# define NOACTION 0
# define STRENGTH 3
p1: I(sig) > X(age,signal): sig == 0 && signal == 1 → I(STRENGTH)
p2: X(age,signal) → X(age + 1,0)
p3: I(sigL) < I(sig) > I(sigR): sig == 0 && sigR + sigL > 0
    {if (sigR > sigL) {newsig = sigR;} else {newsig = sigL;}}
    → I(newsig - 1)
p4: I(sig) < L(age): sig >= 0 → L(age + 1,sig)
```

Production **p1** captures the transmission of the signal from the damaged leaf to its supporting internode, while production **p2** turns off the signal in the damaged leaf once it has been passed. In Production **p3** an internode collects any signals from neighbouring internodes, with the resulting signal strength determined by the stronger if two are present. Production **p4** transmits the signal from an internode to its subtending leaf, transforming the leaf module to have an extra parameter of the signal strength, indicating the level of defence induced. Fig. 4 gives a visualisation of the signal's transmission from two different damage sites in the same plant architecture, showing how damage in different locations could result in very different patterns of defence.

The release of apical dominance caused by damage to an apex can be modelled in a very similar way. The signal would originate in a damaged apex, and would result in outgrowth of axillary buds when it reached them, rather than in an induced defence in a leaf.

3. Discussion

The central result presented in this paper is the demonstration of a capability for creating models of insect–plant interactions, incorporating insect movement on and around dynamic plant architecture, using L-systems. Development of useful models will require collection of data on insect movement and on plant responses to damage, analysis of the data to determine hypothesised models of movement, and testing of the models to determine their validity. As more actual systems are modelled, the techniques for modelling will need to be expanded. Some areas that will require further thought and development are discussed in this section.

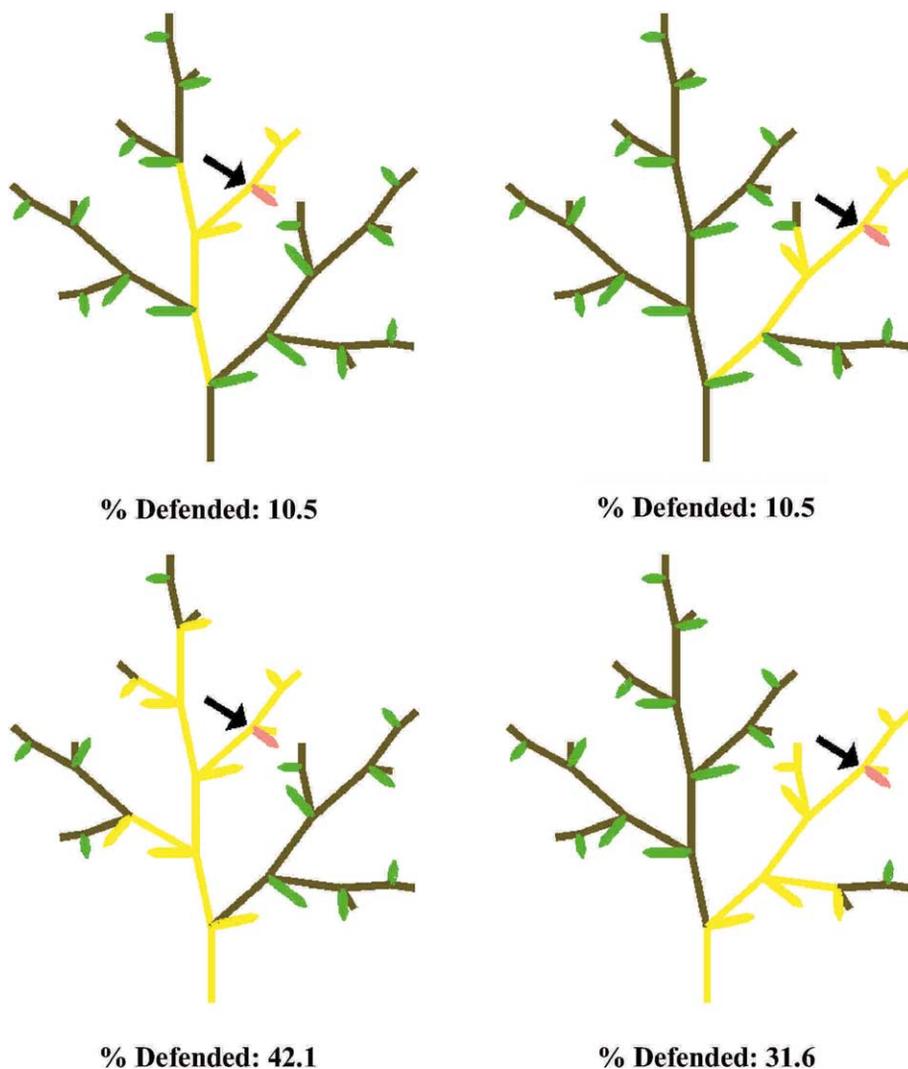


Fig. 4. Induced defences in a branching structure.

3.1. Movement issues

The models described above have been constructed under the assumption that the insect's size and movement rate are commensurate with the plant architecture. This may not always be the case:

- Insects larger than plant components. This will occur for large insects such as a stick insect, but it may also occur for smaller insects moving near the apex of a plant where the internodes are very small. For the situations where the insect would normally traverse a number of components in a single time-step, reduc-

tion of the time-step may suffice to handle this problem. For cases of large differences in size, the situation may become equivalent to movement of insects between non-neighbouring components and modelling techniques such as those described above may need to be applied.

- Insects much smaller than plant components (e.g. mites). If an insect is small relative to stem diameter, movement routines should account for positioning around the stem as well as along it. This is particularly important at branching points, as the insect may not have immediate access to or even be aware of a branch on the opposite side of the stem, despite being at its node of origin.
- Expression of movement rates. The models described above express movement rate relative to internode length. Rates may be modified by delaying the movement step, by changing the time-step or by representing components in greater detail (i.e. breaking internodes into a number of shorter segments). A more flexible approach would be to use a parameter to keep track of insect position relative to a plant component on which it is located.
- Multiple insects. The parallel nature of L-systems allows multiple insects to be handled easily, as long as they are not at the same location. However, rules must be provided to allow insects to pass each other, either crossing in opposite directions or overtaking.
- Collisions. In the models presented here, the general case of insect collisions with plant structures is not handled. It is assumed that there is always enough space available for the insect to move where it wishes, for example that nothing intervenes when it flies towards the closest flower. Collision detection is a computationally expensive proposition, but may be tractable for some situations. For instance, development of multiscale plant models would allow the plant to be treated as a volume until a flying insect approaches, at which time a more detailed representation of the structure could be used, for instance to give locations of individual flowers, leaves, buds, etc.

3.2. Perception issues

Insects may employ a variety of sensory modalities including tactile, visual, chemical or auditory (Bell, 1991). Short-range senses using tactile and chemical apparatus requiring contact with the plant are easily expressed using the simple approach to perception inherent in the models already described, with the insect being able to perceive those components within a rules context. While there is no limit on the size of context that could be checked, increasing context size causes an increase in the number of rules that must be provided, unless the plant has a very regular structure.

Longer-range senses such as vision and chemical apparatus that sample the air aren't restricted to the branching pattern of the plant, so are most naturally expressed using the open L-system approach. Other considerations may include field of view and obscuration of objects by intervening plant components. Having the 3D structural model of the plant environment available yields some interesting possibilities for modelling in this type of situation. For instance, for vision the

actual light signals reaching an insect's eyes could be simulated (c.f. Renault et al., 1990).

Complications arise when an insect's perception of other insects is considered, since both are moving. Problems derive from the discrete nature of the simulation, as movement may cause insects to pass by each other but not to occupy the same position in the string (e.g. if they cross over an internode). Usually these situations can be avoided by decreasing the time step. On the other hand, insects may be at the same position in the string, but not able to perceive each other, e.g. because they are not in each others field of view, or are on opposite sides of the stem. Rules must also be provided to handle these situations.

Consideration also has to be given to what properties can be perceived. We may model plant components such as internodes and petioles as different components, while to the insect they may be indistinguishable. On the other hand, leaves, although modelled as instances of the same template, may in reality be perceived as different by an insect, with a variety of properties contributing to their edibility. Both these situations can be handled by judicious use of parameters.

3.3. Plant models

The L-system formalism is well suited to the modelling of signal flows capturing effects such as induction of defences and release of apical dominance. However, a wider range of mechanistic models of plant physiology are required to capture responses to damage that affect internal resource allocation patterns. For example, leaf damage or removal causes reduced photosynthesis which in turn diminishes resources necessary for the maintenance and development of architecture, while galls and sap-suckers may consume these resources directly. Such models can be expressed using L-systems, but require a large amount of data collection and parameter estimation. We are pursuing two approaches in this area:

- Integrating crop-level physiological models with structural models (see also Jallas et al., 1999).
- Using canonical power-law models of allocation patterns (Kaitaniemi, 2000), coupled with L-system expression of associated plant structure.

3.4. Behavioural Modelling

We have taken a state-based approach to behaviour, with perceived external and internal states considered. For example, if the internal state is 'Hungry' and the location is a young leaf, the insect will eat, whereas if 'Hungry' and the location was stem, the response might be to move upwards. The movement models described here can be extended to include other aspects of insect behaviour, in particular aspects of insect–insect interactions, such as mating and predator–prey situations. The state-based approach may need to be extended to a hierarchy of responses: for instance, in either of the above cases, if the insect also perceives an enemy a fight or flight response might take priority. Alternative approaches from robotics and artificial-intelligence research (e.g. Maes, 1991; Webb, 2000) could also be consid-

ered to see if they are more appropriate. While these types of models can be expressed using L-systems, it may be useful to explore the option of designing an insect behaviour modelling language, just as we have a plant-development modelling language, to allow a more natural expression of the models.

4. Conclusions

L-systems provide a convenient foundation for expressing individual-based models of insect movement operating in the 3D environment of a developing plant that can react to insect damage. The use of the same system for modelling both plants and insects makes it easy to specify interactions between them. The modeller has a great degree of flexibility in expressing insect-behaviour choices, due to the appropriate structural form of the programming language underlying the modelling system. Combining dynamic models of plants, herbivores and predators will allow consideration of questions at a scale that has been difficult to tackle before now.

References

- Agrawal, A.A., 1998. Induced responses to herbivory and increased plant performance. *Science* 279, 1201–1202.
- Alonso, C., Herrera, C.M., 1996. Variation in herbivory within and among plants of *Daphne laureola* (Thymelaeaceae): correlation with plant size and architecture. *Journal of Ecology* 84, 495–502.
- Bell, W.J., 1991. *Searching Behaviour: The Behavioural Ecology of Finding Resources*. Chapman and Hall, London.
- DeAngelis, D.L., Gross, L.J. (Eds.), 1992. *Individual Based Models and Approaches in Ecology: Populations, Communities and Ecosystems*, Chapman and Hall, London.
- Edwards, P.J., Wratten, S.D., 1983. Wound induced defences in plants and their consequences for insect foraging. *Oecologia* 57, 88–93.
- Chelle, M., Andrieu, B., 1998. The nested radiosity model for the distribution of light within plant canopies. *Ecological Modelling* 111, 75–91.
- Gautier, H., Mech, R., Prusinkiewicz, P., Varlet-Grancher, C., 2000. 3D architectural modelling of aerial photomorphogenesis in white clover (*Trifolium repens* L.) using L-systems. *Annals of Botany* 85, 359–370.
- Hanan, J.S., Room, P.M., 1997. Practical aspects of virtual plant research. In: Michalewicz, M.T. (Ed.), *Plants to Ecosystems. Advances in Computational Life Sciences 1*. CSIRO, Melbourne, pp. 28–44.
- Jallas, E., Cretenet, M., Martin, P., Turner, S., Sequeira, R., 1999. Cotons, a new approach in crop simulation model. In: Donatelli, M., Stockle, C., Villalobos, F., Mir, J.M.V. (Eds.), *Proceedings of the International Symposium on Modelling Cropping Systems*. European Society for Agronomy, Catalonia, Spain, pp. 85–86.
- Kaitaniemi, P., 2000. A canonical model of tree resource allocation after defoliation and bud consumption. *Ecological Modelling* 129, 259–272.
- Lindenmayer, A., 1968. Mathematical models for cellular interaction in development, Parts I and II. *Journal of Theoretical Biology* 18, 218–315.
- Lindenmayer, A., 1982. Developmental algorithms: lineage versus interactive control mechanisms. In: Subtelny, S., Green, P.B. (Eds.), *Developmental Order: Its Origin and Regulation*. Alan R. Liss, New York, pp. 219–245.
- Maes, P. (Ed.), 1991. *Designing Autonomous Agents*. The MIT Press, Cambridge, MA.

- Mech, R., Prusinkiewicz, P.W., 1996. Visual models of plants interacting with their environment. Proceedings of SIGGRAPH'96 (New Orleans, Louisiana, 4–9 August 1996), in Computer Graphics Proceedings, Annual Conference Series, 1996, ACM SIGGRAPH, pp. 397–410.
- Neuvonen, S., 1999. Random foraging by herbivores: complex patterns may be due to plant architecture. *Journal of Ecology* 87, 526–528.
- Prusinkiewicz, P.W., 1998. Modelling of spatial structure and development of plants: a review. *Scientia Horticulturae* 74, 113–149.
- Prusinkiewicz, P.W., Lindenmayer, L., 1990. With Hanan, J.S., Fracchia, F.D., Fowler, D.R., de Boer, M.J.M., Mercer, L. *The Algorithmic Beauty of Plants*, Springer-Verlag, New York.
- Prusinkiewicz, P.W., Hammel, M., Hanan, J.S., Mèch, R., 1997. Visual models of plant development. In: Rozenberg, G., Salomaa, A. (Eds.), *Handbook of Formal Languages*. Springer-Verlag, Berlin, pp. 535–597.
- Renault, O., Magnenat-Thalmann, N., Thalmann, D., 1990. A vision-based approach to behavioural animation. *Visualisation and Computer Animation* 1, 18–21.
- Room, P.M., Maillette, L., Hanan, J.S., 1994. Module and metamer dynamics and virtual plants. *Advances in Ecological Research* 25, 105–157.
- Rosenthal, J.P., Kotenan, P.M., 1994. Terrestrial plant tolerance to herbivory. *Trends in Ecology and Evolution* 9 (4), 145–148.
- Thaler, J.S., 1999. Jasmonate-inducible plant defences cause increased parasitism of herbivores. *Nature* 399, 686–688.
- Webb, B., 2000. What does robotics offer animal behaviour? *Animal Behaviour* 60 (5), 545–558.
- Willmer, P.G., Hughes, J.P., Woodford, J.A.T., Gordon, S.C., 1996. The effects of crop microclimates and associated physiological constraints on the seasonal and diurnal distribution patterns of raspberry beetle (*Byturus tomentosus*) on the host plant *Rubus idaeus*. *Ecological Entomology* 21, 87–97.
- Wilson, L.T., Waite, G.H., 1982. Feeding patterns of Australian *Heliothis* on cotton. *Environmental Entomology* 11, 297–300.
- Zalucki, M.P., 1983. Modelling egg laying in the Monarch butterflies, *Danaus plexippus*, L. *Researches on Population Biology* 25, 353–365.