A Simple but Effective Algorithm to Model the Competition of Virtual Plants for Light and Space

William Van Haevre and Philippe Bekaert

Expertise Centre for Digital Media
Wetenschapspark 2
Belgium B-3590, Diepenbeek

William.vanhaevre@luc.ac.be, philippe.bekaert@luc.ac.be

ABSTRACT

We present a simple but effective algorithm to simulate plant growth in a realistic way, taking into account illumination, spatial occupancy and the nearby presence of other plants. The basic idea of our algorithm is to simulate light flow during all stages of plant development, by tracing virtual photons emitted by light sources in the same way as it is done in many global illumination algorithms. We augment the L-system, modelling plant growth, with a mechanism for determining predominant illumination directions by analysing the distribution of the direction of incidence of photons hitting plant parts. Based on photomorphogenesis principles found in literature, the same photons are used to “discourage” plants from growing into each other by penalizing the contribution of photons reflected of other plants. At this time, our algorithm is implemented using the very simplest solutions at all stages. Nevertheless, first experiments show that the approach is promising. In future work, the algorithm proposed here will be refined considerably in various ways.

Keywords

L-systems, tree modelling, photomorphogenesis, natural phenomena, light interaction, photons, environment sensitivity, plant representation.

1. INTRODUCTION

The desire to obtain realistic models of trees is evident from the large number of papers covering this subject.

Our goal is to end up with a description of a realistic model in terms of positions, ages, etc… A good shaded, textured and illuminated visual representation of this model can be added afterwards to the needs of the modeller. For research purposes we developed a simple viewer in which branches are only represented by cylinders. More complex visualizations will, in future work, be build on top of this implementation.

First we will present a small review of earlier work, done in this research domain. Next we will point out a few basic observations concerning the behaviour of trees interacting with light or with other trees. As mentioned before we use a tree representation based on L-systems, so a brief overview of these interesting rewriting systems will be given. The rest of this paper presents our new algorithm, which allows trees to interact with light and with each other, by demonstrating how we managed to model and visualize certain of these natural phenomena.

Previous work

Some of the earliest work done in the field of tree modelling, was done by Lindenmayer, who introduced string rewriting systems, later to be known as L-systems. Later work [Lind90], with a contribution of Prusinkiewicz showed how these L-systems turned out to provide a practical approach to model trees. To achieve this, a few extensions were described: making the L-systems stochastic (in order to loose the self-similarity they used to show) and making them context-sensitive. Aono and Kunii

Journal of WSCG, Vol.11, No.1., ISSN 1213-6972
WSCG’2003, February 3-7, 2003, Plzen, Czech Republic.
Copyright UNION Agency – Science Press

[Aono84] developed a different model, capable of producing more complex patterns of branching, and showing some interesting features, such as the attraction towards objects and inhibition of growth due to several factors. Oppenheimer [Oppe86] took yet another approach (visually influenced by the work of Bloomenthal [Bloo85]), using fractals to create tree models. His technique had the regrettable drawback of being limited to a small number of basic trees. Reeves and Blau [Reev85] developed a technique able of creating trees and grasses using a particle system. Recently, Benes and Millan [Bene02] presented an idea based on this work. It allows climbing plants to compete for space using oriented particles, which are able to sense the environment. Using directed random walks and traumatic reiteration their research allows plants to find the best way to grow. De Reffye et al. [Reff88] obtained a procedural model, based on the birth and death of plant components. Using this idea, AMAP, a commercial library of plants was build. Weber and Penn [Webe95] put more emphasis on the overall shape of the generated trees. This was accomplished by specifying a bounding volume and by restricting the tree model to stay within this geometrical primitive. Lintermann and Deussen [Deus98a, Deus99] developed an interactive plant modelling program *xfrog* [Deus98b] using an approach in which components encapsulate data and algorithms to generate the different plant elements. All components have a set of specific parameters to control their behaviour.

Most of the research mentioned here, was inspired by publications covering computer generated plant development made by Prusinkiewicz et al. [Prus93-01]. One of his latest works [Prus01] covers plant representations. The algorithm described expresses local attributes of the model as a function of the relation between their positions and the overall model (location along the stem).

Our research is highly related to work done by C. Soler et al. [Sole01]. Soler et al. present an extension of hierarchical radiosity with clustering, to be used for simulating plant growth taking into account illumination. Their method is supposed to be very accurate, but its implementation takes considerable care and effort. In contrast, the method we present here is extremely simple while still leading to plausible results. In addition, our method also incorporates a morphogenetically based mechanism, described next, that “discourages” plants from growing into each other too much.

Biological aspects

Studies in the field of photomorphogenesis [Kend86] (the branch of biology studying the interaction of trees with light) have revealed the existence of several very interesting kinds of behaviour. Here, we will present a solution for the visualization of two of these special adaptations:

- In their book, Kendrick and Kronenberg state: “*the perception of light direction yields important information enabling organisms to optimise their position in the natural environment by appropriate orientation movements*”. A good example of this, consists of trees, trying to spread their leaves in such a way that the majority of them receives as much of the available light as possible (because certain components of the light spectrum are necessary for photosynthesis which is a chemical reaction that provides energy to the tree). This phenomenon is called “phototropism”: growth movements of plants in relation to the light direction. A tree provides a solution to this need by obtaining a good distribution of its branches and by growing in the direction that receives most of the light (in the hope to find even more light!)
- Trees can “sense” the presence of each other within a certain neighbourhood, sometimes even when there’s a large distance between them. They can try to avoid each other by growing in an opposite direction so they would not end up in each other’s shadow. The adaptation described, results from a biological reaction by the leaves in which they only absorb certain wavelength components of the light spectrum and reflect others. If a tree receives a lot of light from a certain direction, containing only the components of the light spectrum that are supposed to be reflected, and just a few rays that can be absorbed (the so called photobiologically-active portion of the radiant spectrum), a system will be triggered grow away from that direction. This means that the wavelength of the light that is transported by the light beams, functions as some sort of information transport between adjacent trees. The same rules apply for reflections from walls or other objects, because they filter certain components of the light spectrum. The effect is however strongest for light reflected by a tree-part (a leave, a branch…), because the specifically required wavelengths will be missing in the light beams.

2. L-SYSTEMS

The algorithm starts with an L-system, representing the overall grow process to be used by the plants. An L-system is a rewriting system in which matching modules are replaced by a (possible) more complex sequence of modules. The allowed rewritings are performed in parallel and are described in so called production rules. More information on the use of L-systems can be found in [Bour91, Lind90, Kari96, Prus93-96, Smit84].

One of the most interesting applications of L-systems is the creation of fractals (which most of the time have a rather unnatural and synthetic visual representation, because of the self-similarity on every recursive level). Several changes need to be made to get realistic and natural looking tree models. In the next section we will describe a few of the extensions needed to create nice looking plants.

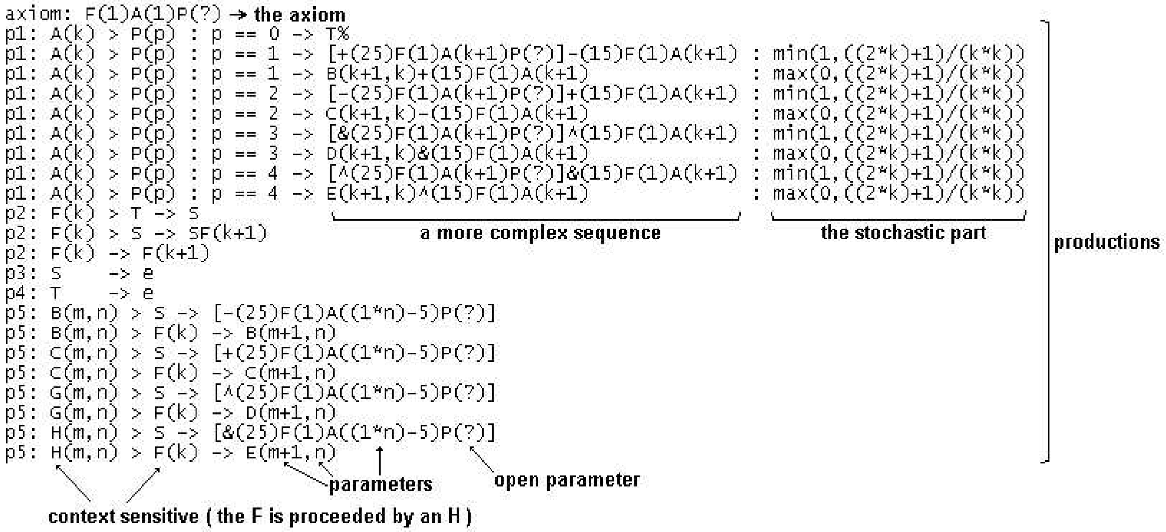
Stochastic, Parametric, Context sensitive, Open L-systems

We use a subset of L-systems that covers a very wide range of possible inputs. The main properties of the L-systems we work with include: they are stochastic, can handle parameters, are context sensitive and they are what we call "open". Each of these properties will be explained very briefly:

- *Stochastic*: This means, that the choice of the production to use when several productions have a good match, depends on a certain probability.
- *Parametric*: Modules, used to describe the productions, can have 0 or more parameters. These parameters are relevant while matching the modules.
- *Context sensitive*: Within a sequence of modules every single module is aware of the modules next to it (this means that the context in which the module is used is known and taken into account).
- *Open*: (term introduced by Radomir Mech [Prus96a]). There is an interaction between the L-system and the program that uses this L-system. After a sequence of modules has rewritten itself, a second pass is done to update specific parameters as needed.

Figure 2.1 shows the L-system we used to generate several of the images in this paper. An example of each of the four properties mentioned above is being pointed out.

The growth of each tree starts with an axiom, which is the start sequence of the L-system. In every growth phase the current sequence of modules is first replaced by a new one: 1. according to the rewriting productions, 2. using the appropriate probabilities when several productions meet the requirements, 3. making the required parameters to match and 4. taking into account the context sensitivity. In a second stage, the requested parameters are updated. This process, when repeated several times, leads to a more complex sequence of modules, which can be interpreted as drawing commands (ex. F represents a sequence of commands to draw an internode, A is an apex, ...). This leads to a visualization also known as turtle-interpretation. More information on how to create and use these L-systems can be found in [Prus93-96].



```
axiom: F(1)A(1)P(?)
p1: A(k) > P(p) : p == 0 -> T%
p1: A(k) > P(p) : p == 1 -> [+(25)F(1)A(k+1)P(?)]-(15)F(1)A(k+1) : min(1,((2*k)+1)/(k*k))
p1: A(k) > P(p) : p == 1 -> B(k+1,k)+(15)F(1)A(k+1)               : max(0,((2*k)+1)/(k*k))
p1: A(k) > P(p) : p == 2 -> [-(25)F(1)A(k+1)P(?)]+(15)F(1)A(k+1) : min(1,((2*k)+1)/(k*k))
p1: A(k) > P(p) : p == 2 -> C(k+1,k)-(15)F(1)A(k+1)               : max(0,((2*k)+1)/(k*k))
p1: A(k) > P(p) : p == 3 -> [&(25)F(1)A(k+1)P(?)]^(15)F(1)A(k+1) : min(1,((2*k)+1)/(k*k))
p1: A(k) > P(p) : p == 3 -> D(k+1,k)&(15)F(1)A(k+1)               : max(0,((2*k)+1)/(k*k))
p1: A(k) > P(p) : p == 4 -> [^(25)F(1)A(k+1)P(?)]&(15)F(1)A(k+1) : min(1,((2*k)+1)/(k*k))
p1: A(k) > P(p) : p == 4 -> E(k+1,k)^(15)F(1)A(k+1)               : max(0,((2*k)+1)/(k*k))
p2: F(k) > T -> S
p2: F(k) > S -> SF(k+1)
p2: F(k) -> F(k+1)
p3: S    -> e
p4: T    -> e
p5: B(m,n) > S -> [-(25)F(1)A((1*n)-5)P(?)]
p5: B(m,n) > F(k) -> B(m+1,n)
p5: C(m,n) > S -> [+(25)F(1)A((1*n)-5)P(?)]
p5: C(m,n) > F(k) -> C(m+1,n)
p5: G(m,n) > S -> [^(25)F(1)A((1*n)-5)P(?)]
p5: G(m,n) > F(k) -> D(m+1,n)
p5: H(m,n) > S -> [&(25)F(1)A((1*n)-5)P(?)]
p5: H(m,n) > F(k) -> E(m+1,n)
```

Figure 2-1: Example of a stochastic, parametric, context sensitive, open L-system, used to model trees. More information van be found in the main text.

Environment sensitivity

The most important aspect of the L-systems we use is that they are open. It gives us the opportunity to include certain decisions during the rewriting process. As can be seen in figure 2.1, module P can take a parameter p. Each value from 0 to 4, which is updated in the second stage, gives a different way of replacement for that module. This means that an interface is provided between the L-system and the software that uses this L-system.

In our case, each of the four values represents a direction to grow a new part at a certain position of the current tree. Our algorithm computes such a value, based on available lighting and occupancy information, in order to make plants grow towards a specific direction.

3. TREE INTERACTION

We address three sorts of interaction in which the plants participate:

Object interaction

In [Prus94b] a good solution can be found to simulate the interaction of a tree with surrounding objects. The approach presented starts by checking for an intersection between a new branch and the objects in the scene. If an intersection is found, the branch will be cut off and a pruning signal is propagated downwards to the lower branches. Recursively other parts will be pruned and other growth directions can be tested. This approach leads to a behaviour illustrated in figure 3.1. A similar idea can be used to force a tree to grow inside a certain primitive. This results in a more synthetic shape often found in topiary gardens (figure 3.2).

In our algorithm, we use a similar idea to prevent the competition of several branches for the same space. The geometry of our trees is sorted into an octree data structure (figure 3.3). Each time a new branch is added, the octree is checked to see if there is any space left for a new branch to grow. If not, the new branch is cut off and a corresponding pruning signal is sent. The octree data structure is also used to speed up ray-tree intersection testing. Because of this, it is better to use two octrees: one for the surrounding objects and one for the tree.

Light Interaction

Light interaction is one of the most important aspects of plant growth, because it provides a plant with the necessary material to survive, by photosynthesis (a chemical process in the leaves that requires light to function).

Use of Photons

First of all we need an easy to use model to represent the illumination in the scene. To simulate nature, we decided to position a light source into the scene that emits virtual photons. Each of the photons is assigned a direction and an amount of energy. The photons are traced through the scene, taking into account that with every intersection a new direction is calculated and a portion of the energy can be absorbed. The number of photons emitted depends on the number of tree intersections. (For instance:

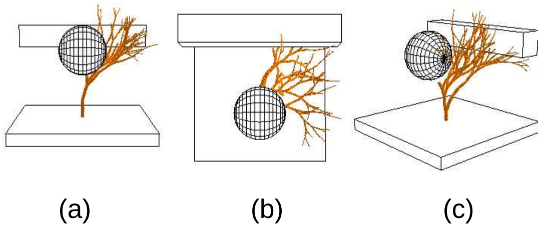


Figure 3-1: Trees interacting with objects in their surrounding environment: (a) front view, (b) top-view, (c) front-right view.

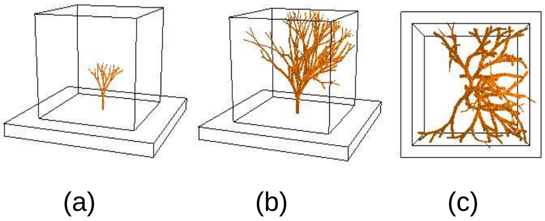


Figure 3-2: A synthetic shape based on an enclosing primitive: (a) after 4 rewritings, (b) after 10 rewritings, (c) top-view, after 15 rewritings.

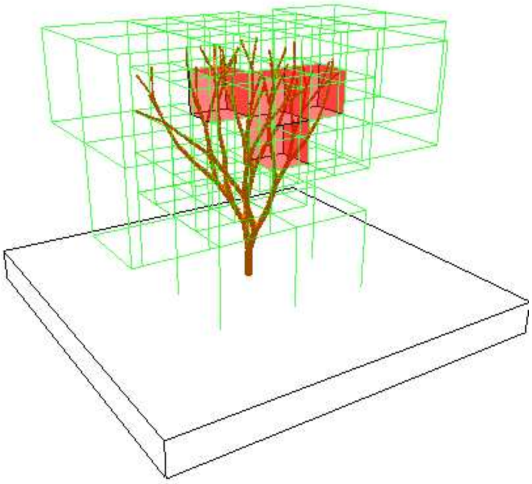


Figure 3-3: The use of an octree as a partial solution to the problem of branches competing for space. (Filled cubes indicate that no more space is available.)

keep shooting photons until you have an average of n photon hits per tree part).

Photon count

Intersections with branches receive a special treatment. Every branch in the tree should provide a number of possible growth directions. (For example: we took 4 growth directions per branch in the L-system in figure 2.1). A good example to retrieve such directions is to represent each branch internally with its surrounding cylinder. On this cylinder several vectors can be calculated, to become possible growth directions. For some applications it might be necessary not to use a predefined amount of allowed growth directions. The use of the L-systems should than be altered. Instead of having one single parameter indicating the new direction, the angle itself should be exchanged between the software and the L-system.

Each of the growth directions of a branch has an associated counter that represents how much light arrives from that direction. Once a photon hits a branch, the incoming light ray is compared to each of the available growth directions. By taking a dot product of the incoming ray direction and the growth direction we get an estimation of how important the contribution of the ray is to that direction and we add it up to the corresponding counter.

After evaluating all normal vectors with all incoming photons, we get a global overview per branch of the incoming light. The normal corresponding to the counter with the highest value, becomes the new growth direction for that specific branch.

The most important improvement resulting from this approach is that we can model trees and plants that are subjected to indirect illumination. This means that light rays that reach them through reflections on objects still can influence trees, which are blocked from a light source by an object. An example of this is given in figure 3.4. This image shows a tree growing around a wall to reach for more light. (Initially, it grows towards the reflection on the right wall)

Interaction with neighbours

The previous section described a method to discover a posteriori that certain regions of space are overloaded with plant parts. From biological experiments it turns out that plants also anticipate to this problem by growing away from each other in an early phase [Kend86]. We present a simple but effective algorithmic solution to simulate this behaviour.

Special Light rays

In this case, again, the solution lies within the light rays. The reason why two trees tend to grow away from each other, results from a reaction to certain rays, originating at one tree and intersecting another tree, which only contain those particular components of light that could not be absorbed and thus can not be used for photosynthesis. Those photobiologically inactive parts of the light spectrum are of no use to the leaves and will be reflected again. Light beams containing the useful light components will only arrive from those parts of the scene where no other plants are situated.

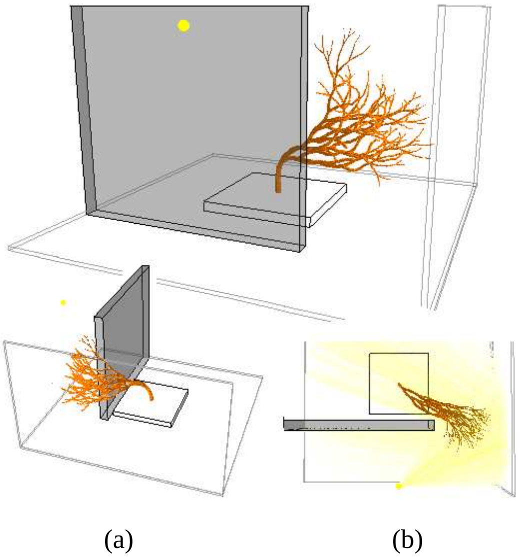

(a) (b)

Figure 3-4: A tree growing in an environment with indirect illumination: (top) the tree in a random view, (a) a different view, (b) top-view with illumination shown. Note, how the tree first grows towards the reflection on the wall and bends to the light source afterwards.

A possible way to handle this in a plant simulation is to take into account the different wavelengths of the emitted light. In this way reflections and absorptions depend on the material properties of the scene, which demands a much more detailed description of the scene. To preserve a sufficient and simple illumination model, a different approach can be taken.

The idea is very intuitive. Whenever a light ray is emitted from a certain tree part (after a reflection) a flag is set, to indicate that this ray has the potential to be a “tree-to-tree” ray. Every time a light ray strikes a tree part, this flag is checked to conclude if this ray has originated on another tree part. In case of such a particular ray, the intersection with a second tree is handled in a different way. (Figure 3.5 illustrates the rays that are emitted from a certain part of a tree and also strike another part)

Instead of increasing the counters (associated with the corresponding growth directions) we decrease the

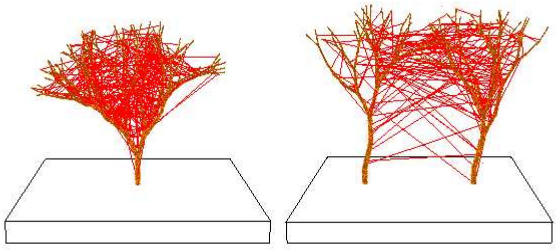


Figure 3-5: Rays preventing trees from growing in each other's shadow by exchanging information about their positions: (a) within a single tree (b) between to adjacent trees.

probability for that specific direction to become the new growth direction. This can be done by storing a new counter for each growth direction, which indicates the penalty for growing in that direction. At the end, when the decision needs to be made about what the new growth direction should be for that branch, all corresponding counters are added together. We specifically use the term "tree parts" because this idea is also applicable between the branches of one single tree.

To illustrate that this simple technique produces plausible results, we virtually planted 4 trees close to each other and added a light source coming from up left. Figure 3.6 shows the results. Seen from the top, with the trees standing close to each other, it looks as if we are looking at one single tree, because the branches have positioned themselves to an almost uniform distribution. If we take a closer look (by increasing the mutual distances between the trees, illustrated on the right), we can observe that each of the trees has taken into account the positions and growing behaviour of the other tree-parts in the scene. Lines were added to visualize an estimation of the space occupied by each of the trees.

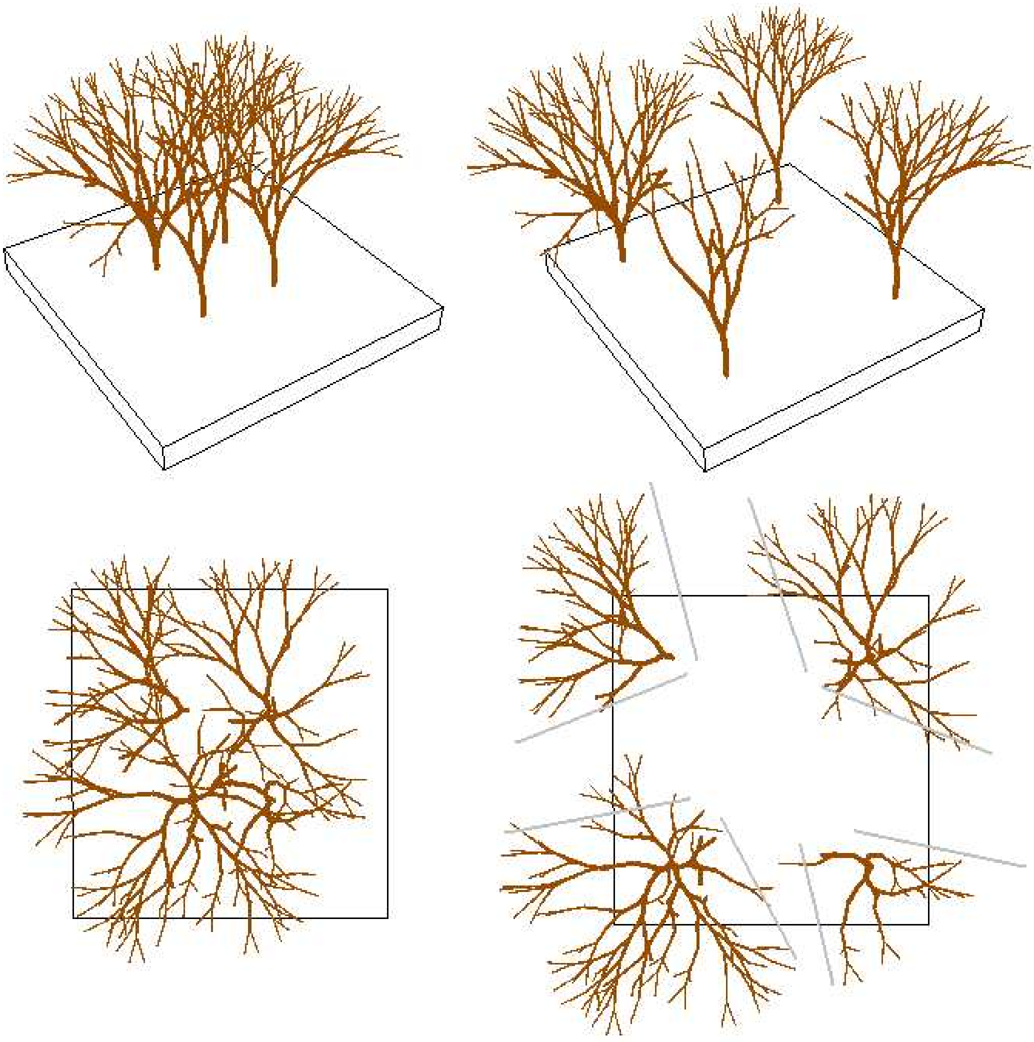


Figure 3-6: 4 trees competing for space: (up) normal views, (down) top views, (left) on the positions were we planted them, (right) pulled away from each other.

Light vs. Neighbour influence

Of course, care must be taken when we say that we just add the corresponding counters together. It depends on the sort of plant we are dealing with whether there is a larger attraction towards the light sources, or a larger repulsion against adjacent plants. By normalizing the counters (dividing al 4 counters by the largest one, and that for both kind of counters) we can simply combine the counters proportionally to the relation between attraction and repulsion.

Figure 3.7 gives a visualization of the counters: (a) the tree seen from a random view (b) the attraction to the light sources, positioned left from the tree. (c) the repulsion between the branches of that same tree. (All counters being normalized).

Mathematically the weighted counters can be calculated as follows:

$$C_i = \alpha \cdot \frac{A_i}{A_{max}} - \beta \cdot \frac{R_i}{R_{max}}$$

where **C**$_i$ is the resulting counter for growth direction i, **A**$_i$ is the counter indicating the attraction to light for that specific branch, **R**$_i$ representing the repulsion against other tree parts. **α** and **β** are numbers indicating how to weigh attraction and repulsion for the tree. In the end, the direction corresponding to the C$_i$ with the largest value indicates the new growth direction for the branch. When using an angle as parameter to the L-system and not a predefined number, an intermediate value could be calculated.

Of course, our approach does not only apply to the specific L-system, shown in figure 2.1. Other kinds of trees or plants can be modelled as well using these algorithms. (Figure 3.8)

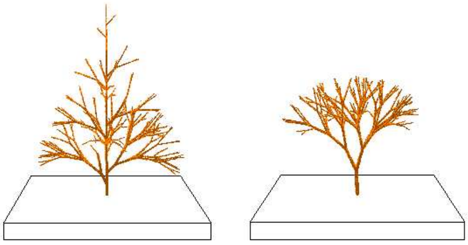


Figure 3-8: Different kinds of trees.

4. CONCLUSIONS

In this paper we proposed a simple but effective extension to L-systems taking into account environment illumination and spatial occupancy. Our technique is based on an emission of photons, which intersect with tree-parts. Each intersection results in an update of a few counters associated with the available growth directions. The accumulated counters are used to guide the process of plant part development.

Inevitably, a number of problems still remain to be solved. First of all, we make very simple and ad-hoc assumptions about how plants react to light: how must attraction and repulsion be weighed exactly, and do plants in reality react in a linear way to incident light intensities? We hope to find a solution for these questions in morphogenesis literature. Second, many aspects of the algorithm presented here can be considerably refined. In particular, we will experiment with more advanced irradiance estimation techniques for finding the predominant illumination direction, for instance in the spirit of the photon mapping method [Jens01]. In addition, a different approach will be taken to describe plant growth, in the style of [Deus99]. Finally, we are looking forward to exploit data available for real plant species.

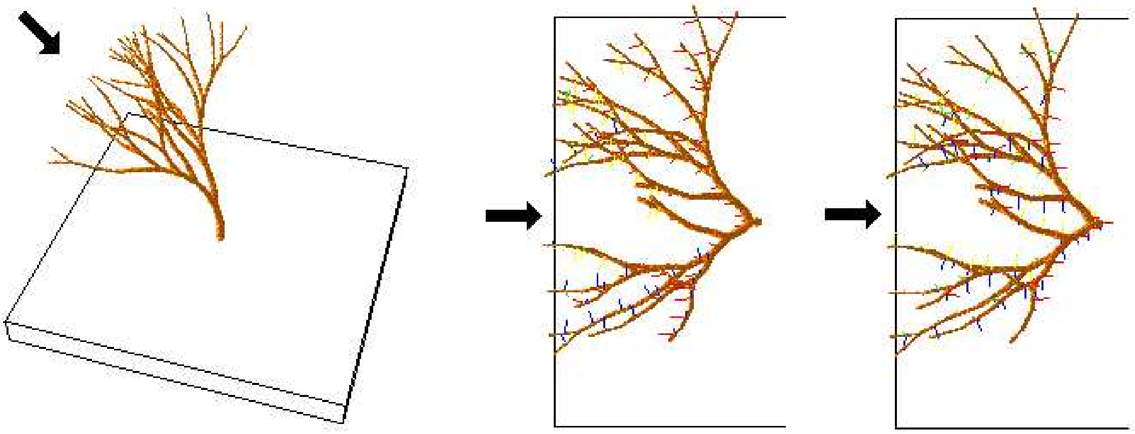


Figure 3.7: Attraction towards light and neighbour repulsion visualised on the same tree: (left) the tree seen from a random view, (middle) top view of the same tree, with a visualization of the growth directions used for attraction, (right) top view, but with the growth directions for repulsion drawn. (The black arrow indicates the incoming light direction)

5. REFERENCES

[Aono84] Aono M. and Kunii L. T., Botanical Tree Image Generation." IEEE Computer Graphics and Applications, Vol 4 Nr 5, pp. 10-34, 1984.

[Bene02] Benes B., Millan E. U., Virtual Climbing Plants Competing for Space, Computer Animation 2002, pp.33-42, 2002

[Bloo85] Bloomenthal J., Modeling the Mighty Maple, SIGGRAPH '85, pp. 305-311, 1985.

[Bour91] Bourke P., L-System User Notes, 1991.

[Deus98a] Deussen O., Hanrahan P., Lintermann B., Mech R., Pharr M. and Prusinkiewicz P, Realistic modeling and rendering of plant ecosystems, SIGGRAPH '98, pp. 275-285, 1998.

[Deus98b] Deussen O. and Lintermann B., xfrog 2.0, www.greenworks.de, 1998.

[Deus99] Deussen O. and Linterman B., Interactive Modeling of Plants, IEEE Computer Graphics and Applications, Jan/Feb 1999.

[Jens01] Jensen H. W., Realistic Image Synthesis Using Photon Mapping, 1st edition, AK Peters, 2001.

[Kend86] Kendrik, R.E. and Kronenberg G.H.M. Photomorphogenesis in plants, Kluwer Academic Publishers, pp. 377-469 & pp. 659-680, 1986.

[Lind90] Lindenmayer A., Prusinkiewicz P., The Algorithmic Beauty of Plants, Springer-Verlag, 1990.

[Oppe86] Oppenheimer P. Real Time Design and Animation of Fractal Plants and Trees. SIGGRAPH '86, pp. 55-64, 1986.

[Prus93a] Prusinkiewicz. P. Modeling and Visualization of Biological structures, proceedings of Graphics Interface '93, 1993.

[Prus93b] Prusinkiewicz. P. Hammel M. and Eric Mjolsness, Animation of Plant Development, proceedings of SIGGRAPH '93, pp. 351-360, 1993.

[Prus94a] Prusinkiewicz P., Hammel. M., Language restricted iterated functions, Koch constructions and L-systems, SIGGRAPH '94 course notes, 1994.

[Prus94b] Prusinkiewicz P. and Mark James and Mech. R. Synthetic topiary, proceedings of SIGGRAPH '94, pp.351-358, 1994.

[Prus95a] Prusinkiewicz P., Hammel M., Hanan J. and Mech. R. "L-systems: from theory to visual models", proceedings of the 2nd CSIRO symposium on computational challenges in life science, 1996.

[Prus95b] Prusinkiewicz P., Hammel M., Hanan J. and Mech R.. "The artificial life of plants", SIGGRAPH '95 course notes, 1995.

[Prus96a] Prusinkiewicz P. and Mech. R. "Visual Models of Plants Interacting with Their Environment", SIGGRAPH '96, pp. 397-410, 1996.

[Prus96b] Prusinkiewicz P., Hammel M., Hanan J. and Mech R.. "Visual models of plant development", Springer-Verlag, 1996.

[Prus96c] P. Prusinkiewicz and Lila Kari, "Subapical bracketed L-systems", Lecture notes, Springer-Verlag, 1996.

[Prus01] Prusinkiewicz P., Mundermann L., Lane B., Karwowski R., The use of positional information in the modelling of plants, SIGGRAPH '01, 2001.

[Reev85] Reeves W.T. and Blau R., Approximate and probabilistic Algorithms for Shading and Rendering Structured Particle Systems, SIGGRAPH '85, pp. 313-322, july 1985.

[Reff88] De Reffye et al., Plant models Faithful to Botanical Structure and development, Computer Graphics, SIGGRAPH '88, pp. 151-158.

[Smit84] Smith, A.R., "Plants, Fractals, and formal languages", SIGGRAPH '84 proceedings, 1984

[Sole01] Soler C., Sillion F., Blaise F., De Reffye P., "A physiological plant growth simulation engine based on accurate radiant energy transfer.", technical report Nr 4116 feb. 2001.

[Webe95] J. Weber and J. Penn, "Creation and Rendering of Realistic Trees", SIGGRAPH '95, pp. 119-128, 1995.