Bedřich Beneš
ITESM Campus Ciudad de México, Mexico City
beda@campus.ccm.itesm.mx

A STABLE MODELING OF LARGE PLANT ECOSYSTEMS

Abstract. The focus of this paper is finding a balance of the plant density and the plant spatial distribution in visual models of large plant ecosystems. This is achieved by simulating plant competitions for resources that is sensitive to many constants and initial conditions, and can easily become unstable causing some individuals to die and the others to grow exponentially. We have found that a negative feedback can provide stability able to resolve catastrophes and unstable states. The replication and seeding cause plants of the same species to grow in clusters that guarantees visual plausibility of the resulting models.

Key words: Artificial Life, Clustering, Competition, Computer Graphics, Individual Based Modeling, Large Ecosystem, Virtual Plant, Visual Modeling

1 INTRODUCTION AND PREVIOUS WORK

The computer hardware evolution allows simulation of phenomena that were impossible to do few years ago. This sentence is true anytime but the feasible applications change during the time. One of the ”hot” areas of computer graphics is a simulation of large ecosystems. The techniques balance on the edge between biology and computer graphics but definitively belong to artificial life. The biological part has its justification in rules that are used for plant simulations. The results are aimed to computer graphics so our interested lies in a *visual simulation* and the plant spatial distribution is the emergent phenomenon.

There are two principal approaches to the large ecosystems simulations [6]. The local-to-global, where the ecosystem is simulated by a plant competition for resources. The second one, the global-to-local, could also be called the computer assisted ecosystem simulation, because the computer is used as a tool helping the user in an interactive plant simulation. This paper belongs to the first class and introduces new approach to the efficient and stable simulation.

Realistic modeling of plants has a quite long history in computer graphics and it can be divided into two important classes. The first focuses on an individual plant modeling, whereas the other deals with plant ecosystems.

Three important streams can be distinguished in the individual plant modeling. The first is rooted in the formal description by means of Lindenmayer systems (L-systems), the second is based on particle systems, whereas the last includes interactive techniques. We point the reader to the tutorial [5] for review.

The plant ecosystems were mentioned for the first time in the context of computer graphics in [8]. Forest scenery is modeled using randomly distributed particles and the plant illumination is approximated.

Chiba *et al* [3] use volume textures to simulate spatial distribution of trees and color of their leaves during the development.

Later work of Deussen *at al* [4] uses many *ad hoc* approaches to obtain realistic images of scenes consisting of thousands of plants. A scene is modeled as a hierarchy of tasks. First, the terrain is modeled and the distribution of rivers is described interactively. The plant occupancy is modeled by the competition. The individual plants are designed either by the `xfrog` or by the `cpfg` program. Three approaches to display the scene are used; OpenGL preview, using the `fshade`, or the `rayshade` programs. In all cases the scene is quantized by means of the techniques that are similar to those used for image conversion from the RGB to the index color space. This decreases the number of plant models and instancing decreases this number even more.

Lane and Prusinkiewicz [6] extended L-systems to *multiset L-systems* that are used for global-to-local and local-to-global approaches. The biologically based part of this work, the plant competition, is achieved by an extension of the technique [4].

An interactive approach based on particle systems is introduced by Beneš *et al* [2]. Objects that are generating particles are located in the 3D space and another objects controlling the growth are influencing the motion path by force fields. So called cutters can stop the particles producing ”virtual topiary” effect.

One of the problematic points of the previously mentioned techniques is the finding the parameters needed to keep the system stable. The parameters are sensitive to each other and have relations that are hard to estimate or the influence of the environment is complex. We propose a solution based on the communication with the virtual environment that provides fast and stable solution.

This paper continues with the individual plant model description. The Section 3 describes the plants ecosystems and the simulation algorithm. The following section deals with the plant rendering and the Section 5 describes the results. The last Section 6 concludes the paper.

2 THE MODEL OF ENGLISH DAISY AND GRASS

Our model deals with two plant species, english daisy (*bellis perennis*) and grass, but can be easily extended to arbitrary number of plant species. The more complicated model is the model of the english daisy.

We have created two level logical model. At the low level the plant is represented as a set of basic geometric primitives located in the 3D space, whereas at the higher level the plant shape depends just on its initial state and the plant age.

The geometry is represented as follows. Leaves are Bézier patches, the center of the flower is the top of a sphere, and its stem is a set of cylinders. The procedure that generates the geometric model has a set of parameters that allows for a generation of the wide variety models (see in the Figure 1). The parameters influence the number of leaves, the angle of the head of the flower, the length of the stem, the angle between the leaves and the center of the flower, etc. The procedure call is a bit complicated and has no biological meaning so we have designed another procedure. This is the second level model of the plant that encapsulates all the parameters and allows the user for better control over the program. Only the initial state and the age control the plant shape.

When the plant is initialized the day of the plant death is also determined. English daisies grow in the ideal conditions for four years. We modulate this time span randomly by 10%. We will see in the Section 3.3 that the fate of the plant also depends on the other plants and external conditions.

The same two level procedural philosophy is followed in the modeling the grass, but there are two important differences. The corresponding geometric model is simpler, because the grass is represented just as one Bézier bi-cubic patch. The annual grasses grow just one year and then die. In the autumn they produce seeds that become dormant in the ground during the winter and then start to grow in the spring. This also depends on the grass vernalization that is easily modeled as a delay of the growth.

Fig. 1 . Different models of the english daisy

3 ECOSYSTEMS

The main aim of the paper is to obtain a realistic spatial plant distribution with variety of plants. We use the local-to-global model typical for artificial life approaches. The scene is described and the plants are planted. Every plant has local rules of behavior and lives its life. The fate of the plant depends on the other plants and the external conditions. A plant can die, grow, and go seed. The emergent phenomenon of the model is the spatial plant distribution.

3.1 The Simulation Algorithm

The ecosystem is represented as a planar area. Planting certain number of plants and setting their age to a nonzero value initializes the model. The initial state is not perfect, but the system converges fast to stability. Then the plants grow and compete for space and resources. The system is controlled by the continuous time approximated by a discrete time-step with $\Delta t = 0.5$ day. During the one time-step the following actions are performed for each plant:

1. plants that should go to seed generate new seeds,
2. plants out of the ecosystem are eliminated,
3. colliding plants are detected,
4. old and colliding plants are eliminated,
5. all plants grow, and
6. the time step is increased, so $t = t + \Delta t$.

The above-described algorithm is quite general and does not include the dependence on the environment. The system allows for exponential growth as displayed on the sequence of images in Figure 2. We will show how the negative feedback is incorporated into the plant competition later in the Section 3.4. First we need to describe how the plant interactions are simulated.

3.2 Clustering

Seeding simulates clustering of the plants. Every plant goes to seed at the end of the year. Seeds are planted within certain radius from the plant so the new plants are aggregated within this area. The competition then eliminates the weaker plants. We can see in the Figure 3 that the plants of the same species keep in clusters although the clusters vary during the time.

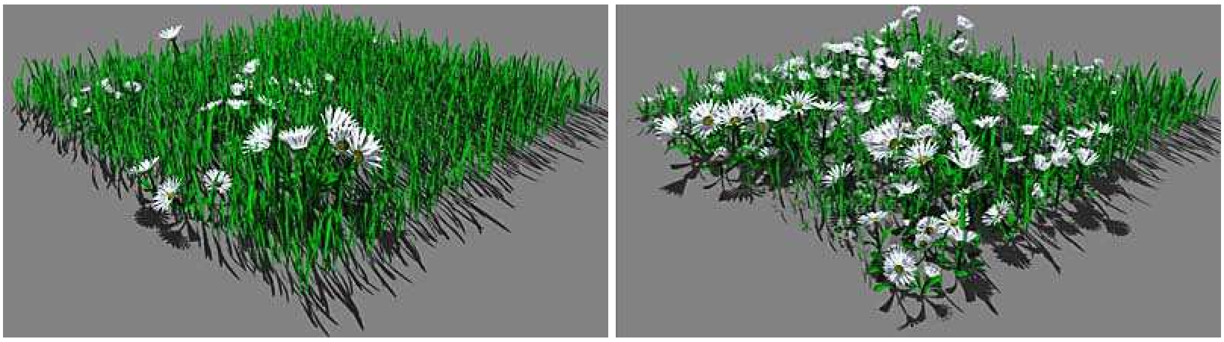

Fig. 2. Local rules can cause an exponential expansion in space. The time is 2 and 11 years

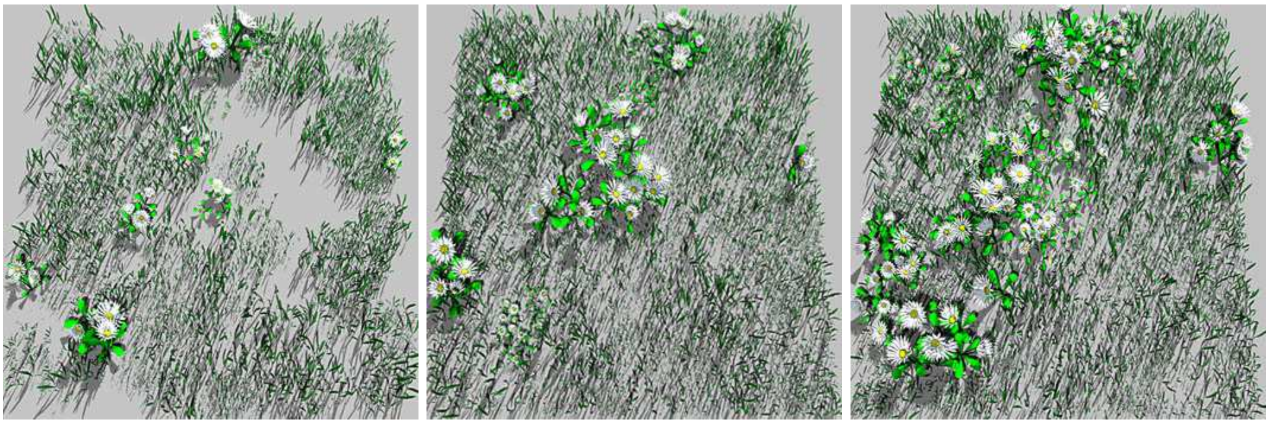

Fig. 3 . An example of a the plant clustering after 500, 1000, and 2000 days

3.3 Plant Interactions

The plant-environment and plant-plant interactions can be simulated at different levels of precision. The detailed interactions of the plant modules (leaves, buds, and internodes) are difficult to compute using the up-to-date computers therefore simplifications are widely used. We extend the interaction model [4, 6] in the following way.

Each plant is represented as a circle corresponding to its ecological neighborhood (see in Figure 5). It is supposed that no other plant can be located within this radius. As the plants grow their neighborhoods interpenetrate and the weaker plants should be eliminated.

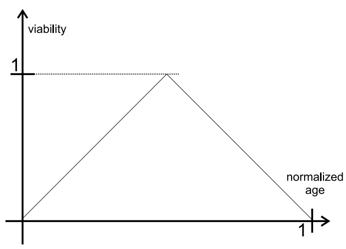

Fig. 4. The viability function

Let's suppose two plant neighborhoods are colliding. The question which should be eliminated and which continues to grow is solved using so called *viability* function denoted by $v(t)$:

$$v(t) = \begin{cases} \overline{age} & \text{if } \overline{age} < \frac{1}{2} \\ 1 - \overline{age} & \text{otherwise} \end{cases} . \quad (1)$$

The $\overline{age}$ is the normalized age computed from the actual age, the time of $born$ and $death$:

$$\overline{age} = \frac{age}{die - born}.$$

The viability is a normalized function returning the plant ability to survive and depends only on the plant's age. The viability is high for mature plants and low for the young and the old plants. We simplify this relationship by a piecewise linear function (1) as shown in the Figure 4. The plant that has higher viability stays in the simulation.

3.4 Influences of the Environment

One of the practical problems is the stability of the system. After testing the system with different rules without an environmental feedback we have found that it leads to an unstable state, as can be seen in the Figure 2, where the daisies and the grass grow and occupy more and more space. One of the ways to avoid this situation is to simulate simulation via environment.

We simulate the influence of the environment as a negative feedback. The function (1) has the same form for all plants in the system. We have modified it in order to incorporate the external conditions.

Usually overcrowding of one plant species cause the resources to run out so the weaker plants die faster. This causes the other plants to have more space and to grow extensively. One plant species dominates the area for some time period and then it is swapped. This process is called the succession [6].

Let's suppose a system with two species denoted by 1 and 2. To include the negative feedback we measure the area covered by plants of each species. Let's denote these areas by a_1 and a_2. The corresponding modified viability functions for the species 1 and 2 are denoted by $v_1(t)$ and $v_2(t)$ and have form

$$v_1(t) = \frac{a_2}{a_1 + a_2} v(t) \qquad v_2(t) = \frac{a_1}{a_1 + a_2} v(t), \tag{2}$$

where the $v(t)$ is the original viability function (1).

The negative feedback scales the original viability function depending on the area of the plants and is incorporated in two ways (let's follow the equation (2)). First, increasing the area of the other plant species a_1 increases $v_2(t)$. Second, decreasing the area of the plants a_2 increases $v_2(t)$ as well. It means that the favorable conditions protect small and young plants. On the other hand if the conditions are not good for one species, the first that die are the young and the old plants.

The equations (2) can be extended for n plants denoted by $k = 1, \ldots, n$ in the form

$$v_k(t) = \frac{\sum_{i=1, i\neq k}^{n} a_i}{\sum_{i=1}^{n} a_i} v(t) = \frac{a_1 + a_2 + \ldots + a_{k-1} + a_{k+1} + \ldots + a_n}{a_1 + a_2 + \ldots + a_n} v(t).$$

4 RENDERING

Plants are rendered in three ways. Circles representing the ecological neighborhood with color intensity corresponding to the age are in the Figure 5. This is used just for orientation to see how the plants are clustered and developing.

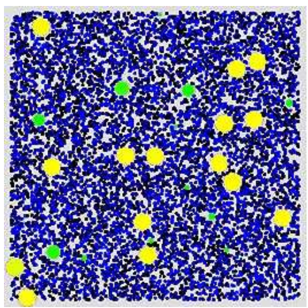

Fig. 5. An area 1m^2, 30 daisies, and 10^4 grass blades

We also use the OpenGL preview to display the plants. This helps to detect possible collisions of the plants and to check the basic orientation and relations of the plant sizes.

The most important is the photorealistic rendering. The Persistence of Vision ray-tracer is used. We use an instancing to avoid an extensive memory usage. Each plant consists of the same parts that are scaled and rotated. These objects are in the memory once and the plants are composed of them using instancing and transformations. The model of a plant is quite complicated and the composition is time consuming, so the rendering is slower than in the case of simple using one model per plant. The file representing the model of one daisy takes 100kB of memory and one grass blade 0.6kB. A scene having 12 daisies and 6448 grass blades would take in non-instanced form 4.7MB of space. The real size of the file is 2.39MB that gives compression factor 51%. We have observed that larger scenes lead to better compression.

5 RESULTS

We are interested in two things. The first is the system's stability that is reflected in the plants' ability to recuperate from some catastrophes. The second aspect is the clustering i.e., ability of the plants of the same kind to keep close together that is important for the visual plausibility of the scene. As we have mentioned above the stability is achieved by the negative feedback between the plant species whereas putting the seeds close to the plant generates the clustering.

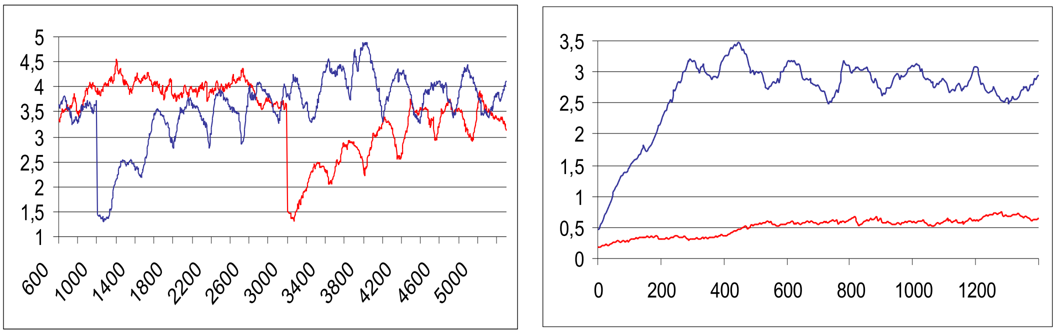

Fig. 6. (Left) Recuperation after the ecological catastrophes that occurs in the day 1000 and 3000. (Right) An incorrectly initialized system reaches its stability fast

The left part of the Figure 6 shows an example of a double ecological catastrophe. The x-axis shows the days of the simulation and the y-axis the total area that is covered by the plants of given species. There are two different kinds of daisies competing for space and resources. As we can see between the day 600 and 999 the system is stable maintaining almost the same average area of both plant species. In the day 1000 we have killed 90% of daisies of one kind. The red curve immediately increases as the other daisies take the space but this causes the run out of the resources. As we can see from the blue curve, the plundered daisies recuperate quite fast and after one thousands days the system is stable again. The same ecological catastrophe occurs with the other daisies in the day 3000 of the simulation and the recuperation follows the same scheme.

Another example, in the right part of the Figure 6, reflects a development of the ecosystems from the Figure 7. The system is initialized incorrectly but it finds the stability fast (300 days).

An example of clustering can be seen in the top view on the Figure 3. The area is not totally covered in the day 300 but the plants are aggregated. This aggregation continues during the time but the clusters are changing positions. The close-up images of the same scene are in the Figure 7.

6 CONCLUSIONS

An efficient approach for visually plausible stable models of large ecosystems is presented. We have shown that a negative feedback can give stability of the system that behaves according to ecological rules. The system is able to recover from ecological catastrophes in a realistic way. The clustering is achieved by seeding plants within certain distance from the plants. This simple technique provides visually plausible realistic results.

The efficient two-level instancing, where we keep in the memory just basic objects and compose them using transformations, helps to render very huge scenes efficiently.

Animations can be found at http://paginas.ccm.itesm.mx/˜beda/research.html

Fig. 7. The close-up images of the scene from the Figure 3. The images in resolution 800×600 were rendered on a dual CPU Pentium IV 2GHz for ten minutes. The scene consists of 5.10^3 grass blades and 10^2 daisies and produced more than one million objects in the memory

REFERENCES

[1] B. Beneš. Visual Simulation of Plant Development with Respect to Influence of Light. In *Computer Animation and Simulation'97*, Springer Computer Science, pages 125–136. Springer–Verlag Wien New York, 1997.

[2] B. Beneš and E. Espinosa. Using Particles for 3D Texture Sculpting. *The Journal of Visualization and Computer Animation*, 5:3–15, 2001.

[3] N. Chiba, K. Ohshida, K. Muroaka, and S. Nobuji. Visual Simulation of Leaf Arrangement and Autumn Colors. *The Journal of Visualization and Computer Animation*, 7:79–93, 1996.

[4] O. Deussen, P. Hanrahan, B. Lintermann, R. Měch, M. Pharr, and P. Prusinkiewicz. Realistic modeling and rendering of plant ecosystems. In *Proceedings of SIGGRAPH'98*, Annual Conference Series 1998, pages 275–286, 1998.

[5] H. Jones. Modelling of Growing Natural Forms. In *Eurographics'00 Tutorials*. Springer–Verlag, 2000.

[6] B. Lane and P. Prusinkiewicz. Generating Spatial Distribution for Multilevel Models of Plant Communities. In *Proceedings of Graphics Interface'02*, volume I, page to appear, 2002.

[7] R. Měch and P. Prusinkiewicz. Visual Models of Plants Interacting With Their Environment. In *Proceedings of SIGGRAPH '96*, volume 30(4) of *Annual Conference Series 1996*, pages 397–410, 1996.

[8] W. Reeves and R. Blau. Approximate and Probabilistic Algorithms for Shading and Rendering Structured Particle Systems. In *Proceedings of SIGGRAPH '85*, volume 19(3) of *Annual Conference Series*, pages 313–322, 1985.