Clustering in Virtual Plant Ecosystems

Bedřich Beneš
Dept. of Computer Science
ITESM CCM, México D.F.
Bedrich.Benes@itesm.mx

Juan Miguel Soto Guerrero
Dept. of Computer Science
ITESM CEM, México D.F.
mguerrer@itesm.mx

ABSTRACT

The spatial plant distribution in virtual ecosystems is usually modeled by means of artificial life. Each plant is considered as a solitary element competing for its resources and obeying simple rules of behavior. Clustering, that is an important clue for visual perception, is achieved by sowing new plants close to their parents. In nature, plants aggregate to clusters that favor plants of the same kind. We introduce an approach that encompasses plant clusters by implicit curves. The clusters ”behave” as compact units and plants inside the cluster have higher chance to survive. Clusters, in the contrary to individual plants, develop by grouped competition that provides visually plausible results, more convincing than those achieved by the previously published algorithms

Keywords
Virtual ecosystems, visual simulation, procedural modeling, plant competition, clustering

1. The Need for Clustering?

Plant simulation and modeling has almost twenty-year tradition in Computer Graphics (CG). There have been numbers of approaches trying to simulate individual plants, many of them purely *ad hoc*, some justified formally, some techniques aiming for plant development, some for real-time displaying, etc. We will not describe this long history here; the interested reader can find an overview in the book [Prusi90] or in the tutorials [Jones00, Prusi95].

WSCG SHORT Communication papers proceedings
WSCG'2004, February 2-6, 2004, Plzen, Czech Republic.

The increasing computational power has lead the interest of researches from individual plants to realistic modeling and simulation of plant ecosystems. It seems the history repeats itself. Some algorithms simulate natural behavior and situate themselves into artificial life [Beneš02, Lane02]. Another algorithms try to simulate LOD and strive for fast and efficient displaying [Deuss98, Deuss02], some techniques focus interactive design [Cohen03], etc.

In all cases, our main goal (for the purposes of CG) is to obtain a spatial plant distribution that is visually plausible and, in this way, reflects reality.

Visually convenient approaches are based on a simulation of the plant competition for resources; water, space, nutrition, etc. These algorithms are inspired or belong to artificial life where the competition and the absence of global rules lead to convincing results.

In the previous work (see in the next section) the algorithms for plant competition were somewhat simple. A plant is substituted by a circle that corresponds to its ecological neighborhood. It is supposed only one plant can be within this circle. Plants grow and, as collisions occur, plants carry out certain actions. In this way the plant distribution is accomplished as the emergent phenomena of the artificial life algorithm.

The main contribution of our paper is the refinement of the described technique. Instead of considering each plant as a stand alone element, we introduce *clusters* of plants which have the same goal as the individual plant, but collaborate, i.e., support themselves in the competitions. A cluster does not permit any other plant specie to interpenetrate that serves for better spatial plant distribution. The power of each cluster can be interactively set by the user. The clustering algorithm is based on implicit curves in 2D space and can be easily extended to 3D.

We continue with the previous work description. Section 3 describes the original algorithm and the next section goes on with the clustering and competition of clusters. Section 5 discusses the implementation issues and the last section describes possible future work and concludes the paper.

2. Previous Work

Deussen et al [Deuss98] has written probably the first paper about simulations of plant development in the area of CG. A simple algorithm simulates the spatial plant distribution. Plants are seeded randomly or interactively by a user action. Each plant is represented as a point in 2D by its position and a circle corresponds to its *ecological neighborhood*. As plants grow their ecological neighborhoods can eventually interpenetrate. This *collision* is interpreted as a competition for resources and the plants' abilities to compete are compared. The winner is definitively left in the simulation and the weaker plant is either diminished or eliminated. This paper, that is extended here, is an example of artificial life approaches. The system of active individuals, each with the ability of sensing its local environment and able to act, is described. There are no global rules and as the simulation runs some pattern emerges. The pattern, also called *the emergent phenomena*, can be a way of behavior, geometrical pattern, or, as in this case, the spatial plant distribution.

This paper was later formalized and extended by Lane and Prusinkiewicz [Lane02]. So-called *multiset L-systems* are used for the formal description of the plant competition. The above described algorithm for the plant competition is applied with minor changes with respect to the previously mentioned paper [Deuss98].

The stability issues were discussed in [Beneš02]. The previously described technique can be difficult to control and plant specie can easily extinct or infest. An environmental feedback can solve this problem. Plants that are not very frequently presented in an ecosystem are supposed to have good conditions to survive, because they have enough resources. The infest plants are weaker since they spend resources intensively. The fitness function used in the plant competitions is modified according to these facts. The system is surprisingly stable and tends to recuperate ecological catastrophes even when 95% of plants of one specie are eliminated. The problem of this technique is that it uses global, statistical, information. No locality is used, as well as no information about the plant distribution.

Virtual agents interacting with ecosystems were recently published [Beneš03a, Beneš03b]. A description language defines behavior of agents. As the simulation runs, agents enter the ecosystem and perform certain actions; they can seed new plants, kill old ones or weeds, water them, etc. The emergent phenomena are fields, virtual gardens, animal paths, etc.

Wang tiles were used for geometric texturing with precomputed illumination in [Cohen03]. Wang tiles are squares with color assigned to each edge that are connected with the matching color. The stochastic alignment leads to a simulation of semi regular patterns, the precomputed geometry and illumination allows their efficient rendering.

3. Simulation Algorithm

The basic simulation algorithm [Deuss98, Lane02] provides a visually plausible spatial plant distribution. This is achieved by means of plant competition. In the original algorithm [Deuss02, Lane02] the competition is simulated in the projected 2D space, where each plant is represented as a circle. The inner part of this area corresponds to the ecological neighborhood of the plant. The most important property is that this space cannot be occupied by any other plant.

As plants grow there is a possible intrusion of the ecological neighborhood of one plant by another one. This is referred to as *collision* or *competition* and the following action is taken. The plant viability (denoted by v_i see details in [Beneš02]) is calculated according to its age, amount of nutrition available, etc. Both viabilities are compared and the winner is determined. The weaker plant is either diminished, i.e., its size is reduced in such a way not to interpenetrate the ecological neighborhood of the strong one, or it is eliminated from the simulation. The processes of thinning and succession are simulated in this way (see [Deuss98] for details).

The stability of this system can be small, some plants can extinct and some can infest, because the local rule cannot assure the overall stability. An environmental feedback is introduced by means of the total amount of plants in the ecosystem. We follow the description of [Beneš02]. If the number of plants of one specie is high, this specie has smaller chance to survive and looses in the competitions easily. This is achieved by scaling the plant viability function by a parameter that is related to the total number of plants.

Let's suppose a system with n plant species indexed by $k = 1, 2, \ldots, n$. We measure the area covered by plants of each species, i.e., the area of their ecological neighborhoods. Let's denote these areas by a_i. The corresponding modified viability functions are denoted by $v'_k$and have form

$$v'_k = \frac{\sum_{i=1, i \neq k}^{n} a_i}{\sum_{i=1}^{n} a_i} v_k \tag{1}$$

4. Clustering

4.1. Combined Neighborhoods

The problem with the above-described algorithm is that it does not simulate nor reflect the grouping of the plants as normally observed in nature. Seeding forms the groups; young plants are sown near their parents so the grouping appears in the natural way. The difference is that a group of plants has different ability to compete compared with a standalone plant. Using just the ecological neighborhood as the key factor in the plant competitions does not reflect this at all.

We have extended the concept of ecological neighborhood to groups of plants in the following way. Plant competition occurs in two different concepts - within a plant specie and with the other plants. If two plants of the same specie meet their ecological neighborhoods are combined, if plants of different kind meet, their neighborhoods are repelled as shown in Figure 1

Inside a cluster, i.e., within the plants of the same specie, the competition is calculated in the usual way as described in Section 3. The second case, the cluster competition, follows in Section 4.3.

4.2. Implicit curves

An intuitive way of modeling joining of different areas of influences can be optimally

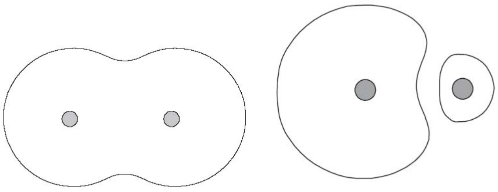

Figure 1: Ecological neighborhoods of two plants of the same specie are combined (left). If the species are different their respective neighborhoods repulse (right)

achieved by *implicit objects* [Bloom97]. Since we deal with a 2D problem, we focus on implicit curves and the areas they form.

Implicit function has the form

$$F(P) = 0,$$

where P is the set of points of the same potential. The value of the potential in the point P is given by a set of n elements (called *the skeleton*), the influence factor c_i, and the blending function F_i. The equation that provides this value is

$$F(P) = \sum_{i=1}^{n} c_i \, F_i(r_i), \tag{2}$$

where r_i is the distance of the point P from the $i-$th element of the skeleton. This function assigns a value to each point P in the space; it describes a field depending on the skeleton and the blending function

$$F_i(r_i) = -\frac{4}{9}\frac{r_i^6}{R^6} + \frac{17}{9}\frac{r_i^4}{R^4} - \frac{22}{9}\frac{r_i^2}{R^2} + 1.$$

The R is the maximal distance of influence of the i-th element of the skeleton.

The function (2) provides the value in each point in the space. Finding isosurfaces, in general, is a complicated task. Here we deal with the 2D case, so we evaluate the function in every point P of a predefined 2D grid.

The advantage is that it provides the power of each cluster in every point that is inside the area defined by the implicit function. The inner part of the cluster are all points P such that

$$F(P) > 0$$

The size of the grid depends on the area of influence of the smallest plant. To sample under the Nyquist limit, this should be at least 1/2 of its size. In practice, for ecosystems made up of hundred of thousands of plants, this leads to the grid resolution of approximately 20k cells.

An example in Figure 2 demonstrates formation of the groups by means of implicit curves. Plants of the same species are sown close to their parents. The clusters are recomputed and the cluster formation appears as the results of this successive aggregation.

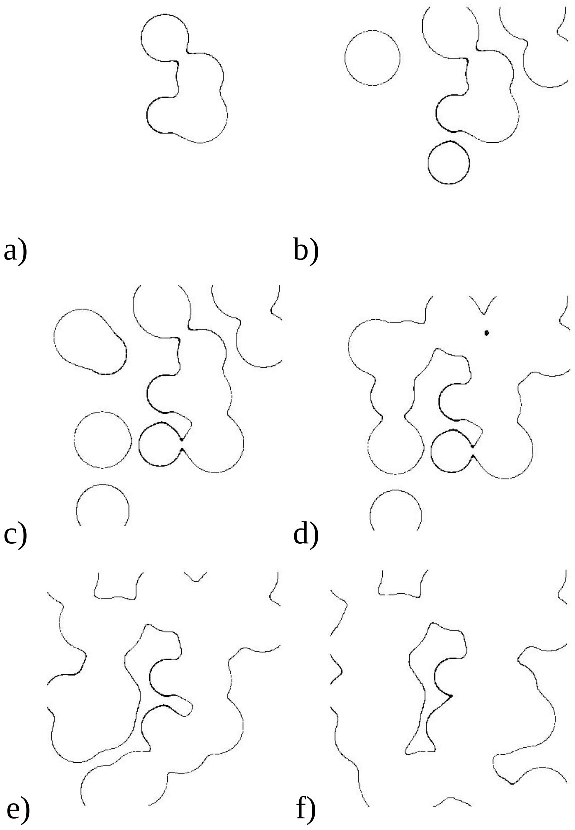

Figure 2: Cluster formation

4.3. Cluster Competitions

To evaluate cluster capacity to compete we need to determine the area of the plant ecological neighborhood of each plant specie. To do this we divide plants into two groups - foes and friend - and calculate the implicit curves.

Each plant has its viability depending on the age, nutrients, etc. This viability should correspond to the area of influence of the plant and is therefore represented as the parameter c_i in the equation (2); strong plants increase the size of the cluster. The c_i is negative for the plants that do not belong to the same group and positive to the tested plant specie. Figure 3 shows an example from the simulation.

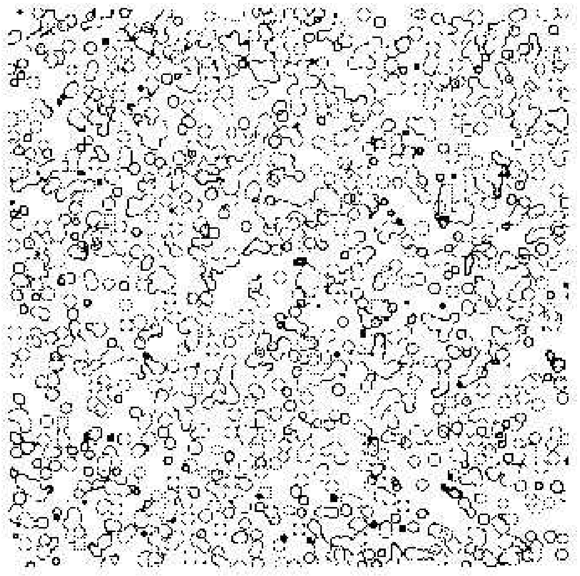

Figure 3: Isocurves showing clusters of one plant specie

Implicit curves are computed in this way for each plant specie and the resulting extents are counted. The cells inside the area formed by the implicit curve, i.e., cells with $F(P) > 0$, are counted. A simple scan-line algorithm calculating *in* counter for each scanline could be also implemented. Cells with nonzero counter are the cells that are inside the area formed by the implicit curve.

The interested thing resulting from using the implicit curves is that the area of the ecological neighborhood depends on the plant viability as well as on the proximity of other plants or clusters. If there is a very strong plant close to another one, the second has smaller chance to survive. This is captured by influencing the sizes of the neighborhoods as shown in the Figure 1.

4.4. Improved Algorithm

The algorithm for the ecosystem simulation works as follows

1. The clusters are calculated for each specie.
2. The local competitions inside each cluster are performed.
3. Clusters are recalculated and their areas are determined.
4. The viability functions are scaled.
5. Competitions between different plant species are calculated.

5. Implementation and Results

The entire program is implemented in C++ and uses OpenGL with GLUT. OpenGL is used for preview, the realistic images were raytraced in Persistence of Vision ray tracer. We use instancing to get reasonable timing.

The graphics card is not really necessary, because the majority of the work is performed on the CPU. Since the 2D area is overcrowded we do not use any kd-trees or 2D variant of BSPs. Regular subdivision into $n \times n$ cells works quite fine, since each cell has approximately the same amount of elements, so their load is well balanced.

We have tested the system on an IBM PC running on Intel Pentium 4 on 1.8GHz with 1.2GB of memory.

The application does not run interactively. Testing a scene with 0.5 millions of plants of ten years of simulation with $\Delta t = 1/2$ day takes more than two hours. The application uses the computer memory very intensively, simply because we deal with really huge amount of data.

One of the interesting things is the way we can compare the quality of the resulting scenes. We want to simulate realistic behavior of plant ecosystems, so the algorithms should lead to realistic scenes. An example in the Figure 4 and 5 show a scene that was obtained using the technique described here. As clearly visible, the plants tend to stay in clusters and they propagate as a colony. A single plant that abandons the cluster has smaller (but not zero) probability to survive.

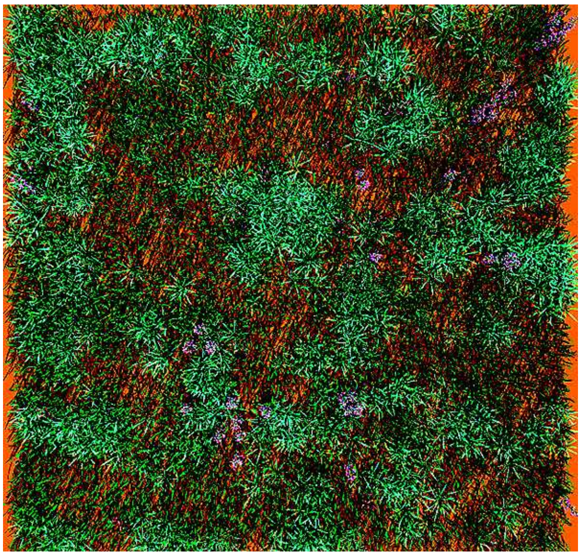

Figure 4: Example of an ecosystem with plant clusters

The plant propagation is an interesting thing that is achieved thank to the plant clustering. The new seeds are integrated into the cluster only if they are close to the cluster boundary, i.e., to the implicit curve. If this is not the case, the plant is abandoned in the space of the hostile plants of another plant species.

6. Conclusions and Future Work

We present an approach for modeling plant competition by means of clusters. The previous techniques considered a plant as a solitary element competing for its space and life with the other plants. The approach introduced here groups plants of the same specie into cluster which boundary formed by an implicit curve. Areas cut out by implicit curves are compared and fed back into the system when the viability functions are evaluated. This makes the system globally stable and the plants clusters are only changing their positions and sizes. Newly sown plants are integrated into the clusters if they are not far. If there is no connection with the group they behave as in the previous case i.e., fighting alone for their lifes.

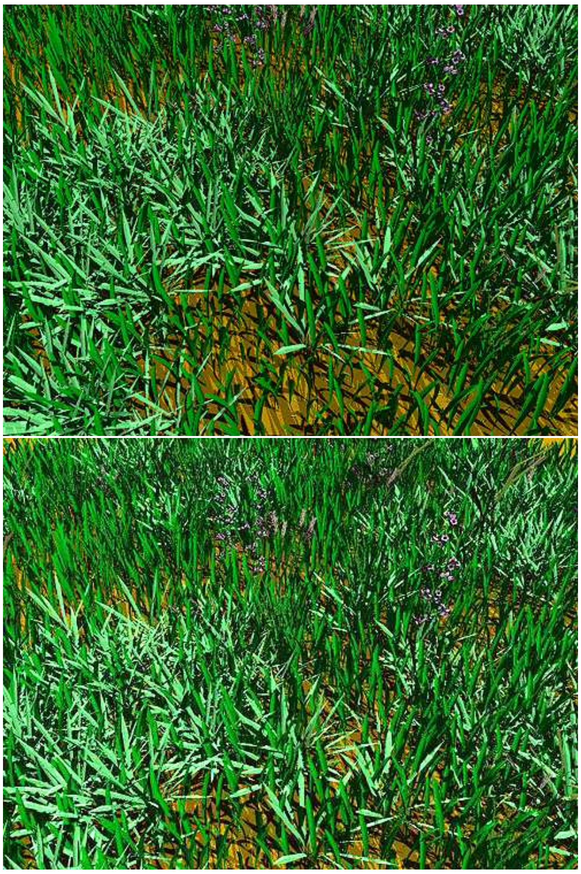

Figure 5: Closeups of the previous scene

Clustering of the artificial life elements seems as a new approach; we have not found this mentioned in literature. We believe that this concept presents a fundamentally new approach. One can imagine 3D application to a group of competing flocks of birds, or a single animal competing with a cluster of hostile entities. There are possible extensions of the ”boids” [Reyno87] that are not perceiving the leader independently, but as a cluster of objects.

Apparently, there are many issues to solve. The most common objection is: How is this everything related to nature? Are there any proofs this is biologically correct? Our approach is biologically inspired, giving visually plausible results. This is usually enough in CG when we want to have tools resembling reality for computer animation, videogames, etc. The above-asked questions are still waiting to be answered.

7. REFERENCES

[Beneš02] B. Beneš. A Stable Modeling of Large Plant Ecosystems. In *Proceedings of the International Conference on Computer Vision and Graphics*, pages 94–101. Associoation for Image Processing, 2002.

[Beneš03a] B. Beneš, J.A. Cordóba, and J.M. Soto. Modeling Virtual Gardens by Autonomous Procedural Agents. In *IEEE Proceedings of the Theory and Practice of Computer Graphics 2003*, IEEE Computer Society, 2003.

[Beneš03b] B. Beneš and E. D. Espinosa. Modeling Virtual Ecosystems with Proactive Guidance of Agents. In *IEEE Proceedings of the Computer Animation and Social Agents 2003*, IEEE Computer Society, 2003.

[Bloom97] Jules Bloomenthal. *Introduction to Implicit Surfaces*. Morgan Kaufmann, 1997.

[Cohen03] M.F. Cohen, J.Shade, S.Hiller, and O. Deussen. Wang Tiles for Image and Texture Generation. In *Computer Graphics Forum*, volume 22(4) of *Eurographics Proceedings Series*, Eurographics, 2003.

[Deuss98] 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. ACM Press, 1998.

[Deuss02] O.Deussen, C.Carste, M.Stamminger, and G.Drettaksis. Interactive Visualization of Complex Plant Ecosystems. In *Proceedings of IEEE Visualization'02*, pages 120–128. IEEE Computer Society, 2002.

[Jones00] H. Jones. Modelling of Growing Natural Forms. In *Eurographics'00 Tutorials*. Springer–Verlag, 2000.

[Lane02] B. Lane and P. Prusinkiewicz. Generating Spatial Distribution for Multilevel Models of Plant Communities. In *Proceedings of Graphics Interface'02*, volume I, pages 69–80, IEEE Computer Society, 2002.

[Prusi90] P. Prusinkiewicz and A. Lindenmayer. *The Algorithmic Beauty of Plants*. Springer–Verlag, New York, 1990. With J.S.Hanan, F.D. Fracchia, D.R.Fowler, M.J.de Boer, and L.Mercer.

[Prusi95] P. Prusinkiewicz, M. James, R. Měch, and J. Hannan. The Artificial Life of Plants. In *SIGGRAPH '95 Course Notes*, volume 7, pages 1–38. ACM SIGGRAPH, 1995.

[Reyno87] C. Reynolds. Flocks, Herds and Schools: A Distributed Behavioral Model. In *Proceedings of SIGGRAPH '87*, volume 21(3) of *Annual Conference Series*, pages 25–34. ACM Press, 1987.