2003－ＣＧ－113　　（10）
2003／11／25

Generating Realistic Spatial Distribution of Flowers in Plant Ecosystem using Artificial Life System *Avida*

Rawitat PULAM * DongSheng CAI †

Abstract

Generating realistic spatial patterns of Plant Eco-System (PES) has a lot of important applications in computer graphics (CG) modeling of outdoor scenes and landscapes, as well as other potential areas such as a study of theoretical ecology of plant population. However, the evolution of PES, including the complexities and noises in the environment, is indeterministic and therefore make this a difficult and challenging task. Also, pervious works had focused on either generating plant distributions for CG applications or studying spatio-temporal processes in real plant communities. In this research, we propose an approach based on computer simulation using an adaptive and self-organizing Artificial Life system *Avida*, as a realistic natural PES texturing engine, merging together CG and theoretical studies, and thus enable us to gain more understanding of PES and generate realistic CG scene at the same time.

人工生命システム Avida を用いたリアリスティックな植物エコシステムにおける花の空間分布生成

Abstract

リアリスティックに植物エコシステム (Plant Eco-System, PES) の花の空間パターンを生成することは屋外風景や地形の Computer Graphics (CG) モデリングにおいて、また、植物人口の理論エコロジー学においても非常に重要である。しかしこれは、PES の非決定的な進化、環境の複雑さやノイズが存在するため、困難な問題である。また、先行研究は、CG か植物人口の理論エコロジー学のいずれかのみに着目していた。そこで本研究では自然な PES テクスチャ生成方法として、適応的な自己組織化人工生命システム Avida を用いたコンピュータシミュレーションに基づく手法を提案する。本手法により、PES のより深い理解とリアリスティックな CG の生成を同時に行うことが可能になる。

1 Introduction

Synthesis of realistic images of terrains covered with vegetation with realistic spatial distribution is challenging and important problem in CG, as well as understanding the underlying processes and natures of how these natural textures are formed in real plant communities is a challenging and important problem in theoretical ecology.

Several research works had been undergoing in both fields. However, most of them fall into two categories:

- Generating spatial distributions of plants in PES by methods that yield no insight of how there are generated in reality. Example of such works are [6] and [9].
- Study the spatio-temporal processes using statistical data collected from real fields and/or modeled them mathematically (mostly using differential equations), without generating spatial pattern. Example of such works are [5], [7], and [8].

However, this created a paradox: how a spatial pattern can be generated, with all its natural spirits such as high environmental entropy and noises, without modeling real plant communities, and how the underlying processes of pattern formation be understood without really generating the pattern?

In this research, we propose a framework for binding them together using bottom-up, synthesis approach with might yield new insights into theoretical ecology and, at the same time, generating realistic spatial distributions for CG applications.

2 Spatial Pattern Formation in Plant Communities

PES are three-dimensional entities. However, as the vertical dimension is the height of the horizontally-

[1]筑波大学 大学院 理工学研究科
Master's Program in Science and Engineering.
University of Tsukuba.
rawitat@coins.tsukuba.ac.jp
[2]筑波大学 電子・情報工学系
Institute of Information Sciences and Electronics.
University of Tsukuba.
cai@is.tsukuba.ac.jp

arranged plant bodies, therefore *spatial pattern* in PES is referring to the two-dimensional projection of the plant bodies onto the earth's surface. Even though the vertical dimension plays an important role as plants often compete in vertical growth which result in death of some individuals and hence effect the horizontal pattern; the height is still largely determined by the biochemical constraints. The horizontal dimensions, however, have no such constraint.

Understanding spatial patterns in PES had been a long-standing challenge in theoretical ecology and biology. This is due to several natures of the plant communities themselves. For instance [8],

- Most patterns are scale-dependent. There seems ot be range of *correct* scales at which the pattern is the most interesting. This is because individual plant species differ markedly in intensity and range of their aggregation.
- Their dynamic nature. Even though the general features of spatial patterns tend to be quite persistent, the patterns almost remain frozen.
- Noises in environment. In PES, many factors for indeterministic and there are always a lot of environmental noises, both within a single PES itself and between several PES. As a result, patterns are very vary. Patterns found in two similar communities are almost differ [7]. Nevertheless, they are statistically self-similar.

For detail discussions on these topics, see [8].

In this research, we take an bottom-up approach in generating spatial patterns in PES, that is: use the internal processes operate in the system and let the *emergence* and *self-organization criticality* (SOC), which are ones of the ubiquitous properties of complex systems, bring out the spatial patterns.

There are two major internal processes operate in the PES that could generate the spatial patterns seen [8]:

- Natality (associated with dispersal): While plant individuals themselves usually do not move, new individuals typically establish at a distance from the mother individual. Dispersal distance that is associated with establishment is highly variable between species and often highly species-specific.
- Interactions between individuals: Because plants are immobile, they interact only with their immediate neighbors. Interaction here refers to effects, both positive and negative, that one plant individual may exert on another individual. The majority of interactions are due to resource competition.

For detail discussions on these topics, see [8]

According to these natures of plant communities and internal processes, we found that artificial life systems (section 3) are suitable as tools for modeling and studying plant communities, and hence as well as generating the spatial patterns.

3 Artificial Life

In the past ten years, computers have become powerful enough to enable us a new way of studying the evolution of life. Rather than following the traditional approach of trying to capture properties of while populations in mathematical models, this new approach models a large number of *individual* self-replicating entities which are competing against each other for resources required for replication. In simpler words, *artificially synthesizing* self-replicating programs that live in a virtual world and study them as they evolve. These systems are called *Artificial Life* (ALife). Examples of such systems are Tom Ray's *Tierra* system [11], which pioneered this approach of study, and California Institute of Technology's *Avida* system [2], used in this research.

Also, these ALife systems are not merely *simulations*. As the programs replicate themselves, they *recreate* the conditions necessary for evolution [14]. In other words, they adjust, self-organize themselves into the environment in which they exist for a better replication. Therefore, these systems are *strong* ALife systems (computer programs as actually living in some respects), not *weak* ALife systems (computer programs as useful simulation of real life) [13]. However, we will be using a strong ALife system *Avida* to study natural phenomena (in this case, spatial patterns in plant communities) as an potential alternative to using weak ALife systems suggested in [10].

3.1 The *Avida* System

The world in *Avida* is an $N \times M$ grid with the topology of a torus. Each position in the grid hold either an empty, unoccupied space, or a self-replicating string (a *program*). Each string is a segment of code written in a language similar to Intel 80x86 assembly language with supplementary instruction sets that allow self-replication, running on a configurable virtual computer [2].

The heart of the *Avida* system is the artificial energy metabolism of the organisms. The primary resource, without which no program can survive, is CPU time [3]. CPU time is distributed in *time slices* to each organism in the population. Each individual receives the same amount of *default* CPU time. Extra *bonus* CPU time is given out as a reward for those programs that have developed computational mechanisms to accomplish certain tasks. In current implementation of *Avida*, these tasks are logical operations on binary numbers, with up to three inputs [2][3]. As a result, individuals that learned to perform such tasks are given more CPU time (in a lingo of *Avida*, more *merits*) and hence replicate better.

Individual programs in *Avida* learn to perform tasks by learning to adapt to the environment in which they exist. This is done through evolution of the programs in Darwinian manner: each program in *Avida* is subjected to mutations, which is the key to adaptation. Programs that adapted better to the environment are said to have higher *fitness* (that is, they fitted better), and therefore have higher merits. (In *Avida*, fitness is calculate by the merit of an individual divide by the time required for its replication).

An *Avida* world, the embeddedness of the individuals and the scope of local interactions are graphically illustrated in figure 1. According to the figure, we can

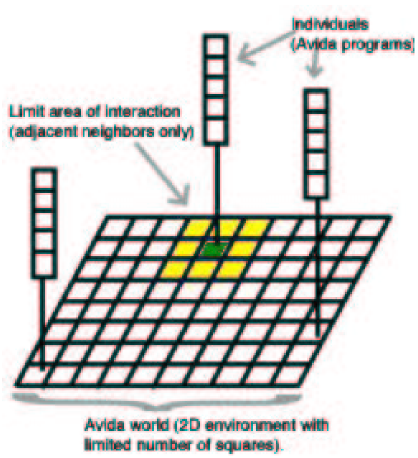

Figure 1: Graphical illustration of *Avida* world.

clearly see why *Avida* is suitable for modeling plant communities and we can instantly draw close analogies between *Avida* system and real plant ecosystems. This will be the subject of section 4.

4 Artificial Life Framework for Modeling Plant Communities

4.1 Natural Plant Community and *Avida*

From the natures of natural plants communities discussed in section 2 and of ALife system *Avida* in section 3, we can see a lot of conceptual similarities between them, as listed briefly in here:

- Mobility of individuals. Both are immobile.
- Local area interaction. Immediate neighbors for plants, Moore neighborhood (adjacent sites) for *Avida* programs.
- Spatial patterns are two-dimensional. "Vertical" dimension is for plant's height, as for program's length (see figure 1). Plant's height effect the horizontal patterns as it effects plant's ability to get more sunlight and results in death of nearby plants, *Avida* program's length effects the ability of getting CPU time for better replication, and hence results in replacing its adjacent site.
- Plants adapt themselves to the environment through evolution and natural selection. *Avida* programs adapt themselves to the environment through evolution and a mechanism akin to natural selection.

Also, there are various other results from previous *Avida* and *Tierra* experiments that agreed well with data collected from real field studies. For example,

- Relationship between mean plant mass and maximum population density in plant populations obey a so-called power-law distribution [12] (figure 2), which is an evidence of existence of SOC in natural plant communities. In ALife, power-law distributions of genotype abundance (relationship between programs' size and number of species in a limited-space environment). were reported from various studies of self-replicating ALife programs, using various ALife systems [4].

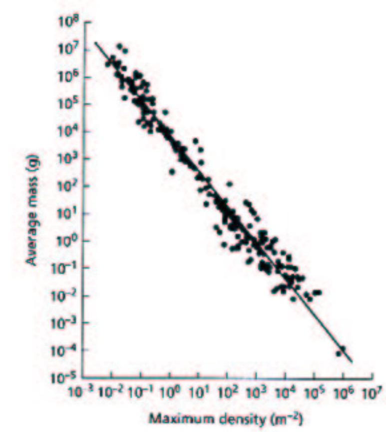

Figure 2: Relationship between mean plant mass and maximum population density in 251 population of plants. *Source:* [12]

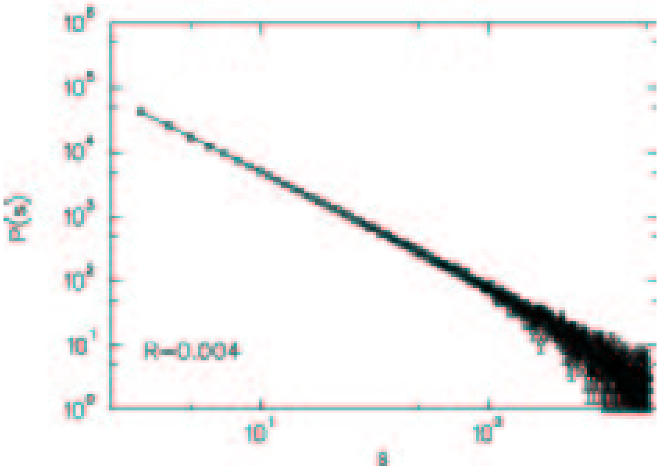

Figure 3: Abundance of genotype size plotted against number of species with mutation rate = 0.004 in *Avida*. Note that while the slope changes according to the mutation rate, the power law characteristic is ubiquitous. *Source:* [4].

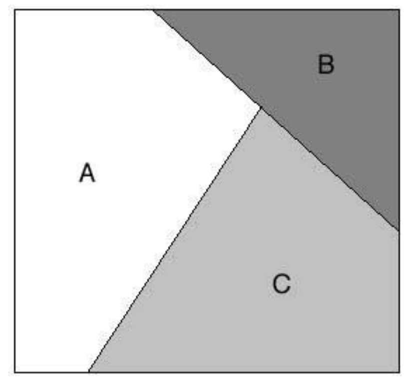

Figure 4: Example of spatial distribution of transformable resources in *Avida*

From these conceptual similarities, it is therefore possible and reasonable to construct a model based on *Avida* for modeling natural plant communities and generating their spatial patterns.

4.2 Modeling Resources Competition and Adaptation in Plant Ecosystems with *Avida*

Using *Avida* for modeling resources competition is suggested and thoroughly discussed in [3]. We will briefly review it here.

As described in section 3, adaptation activity in *Avida* is geared towards the evolution of computational genes. These computations ("tasks") are logical operations such as i AND j, i XOR j, NOT i OR j (NOR), etc. Genes that learned to perform these tasks will get more merits and thus considered better-adapted or fitter to the environment. These tasks, therefore, can be considered as *resources* that individuals compete against each other, both between and within species, to get.

Let us imagine, a simple environment in which there are only three different possible tasks, like ones introduced above. Then, we associate resource A with AND, B with XOR, and C with NOR. We can now load up the world with these resources, as well as controlling and limiting them. Every time an organism performs an AND operation, a certain amount of resource A is depleted, and similarity for the other resources. The consequences of such scenario are very important as computational genes can only evolve in regions where the corresponding resource is present. Thus local differences in resource abundances will lead to different genes evolving in different areas.

At the same time, performance of a computation in the presence of the enabling resource might transform the resource rather than use it up. In such a model, we could have

$$\begin{array}{c} A \xrightarrow{AND} B \\ B \xrightarrow{NOR} C \\ C \xrightarrow{XOR} A \end{array}$$

Imagine we start an experiment in which three resources A, B, and C, are distributed spatially as in figure 4. If resources are continuously renewed, but not transformed, specialists will evolve in the three different habitats, and no species can invade the other. On the other hand, if the resources become scare and if they are transformed through usages, things become very different as the dynamics of spatial pattern formations will be emerge as the individuals performing tasks.

5 Results

In this section, we show a spatial pattern of programs in *Avida* taken during an *Avida* run. *Avida* was configured to analogically reflect a real plant community as much as possible. For the result showed here, the environment size was set to 40×40 sites. A birth method for a new genotype is set to replace the oldest and least adapted, least fitted neighborhood. Genotypes also die when the age limit is reached. To allow adaptation, copy mutation rate was set to 0.0075, insert and delete mutation were set to 0.005. Note that the latter is not important, they help speed up the evolution as suggested in [1]. Also, for the full complexity of environment, a full set of computational tasks, which the individuals have to learn to perform in order to adapt to the environment, was used. The spatial pattern generated in *Avida* is showed here in figure 5, while a corresponding graphically rendered images are showed in figures 6 and 7.

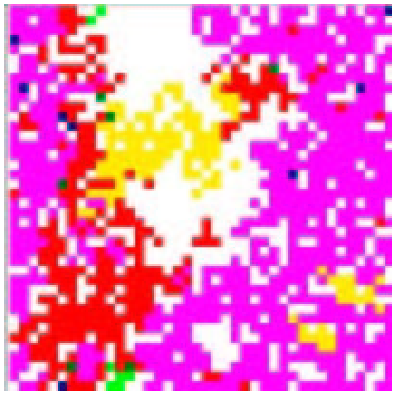

Figure 5: Spatial distribution in *Avida*

.

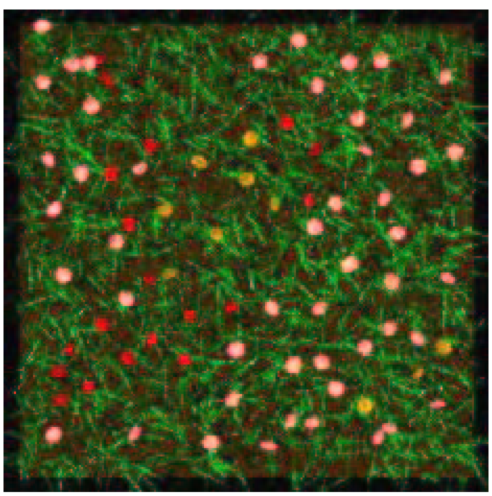

Figure 6: Plant community with spatial distribution pattern generated by*Avida* (figure 5). Some genotypes with small distributions and low fitness/merits were left out as they will die as "seeds" and never make it to a real plant.

Note here that, there exist some small group of genotypes in *Avida* (figure 5) that are not in the rendered images. This is because these genotypes were not fitted well to the environment and died out in a few time steps. This scenario is akin to seeds of plants that cannot grow in a particular environment and hence never make it to the plant community as they died in seeds.

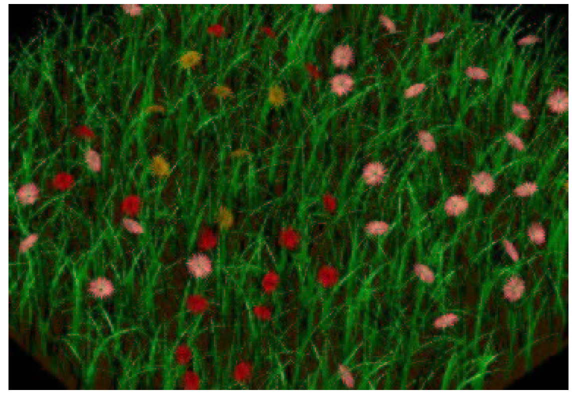

Figure 7: Figure 6 with rotation and zoom.

6 Discussion and Future Research

In this research, we proposed an approach of using an auto-adaptive, evolvable, self-organizing Artificial Life system to generate spatial patterns of plant communities for CG applications and as a tool for studying them theoretically. The spatial pattern generated looks realistic, but we still cannot judge the realness of the pattern yet as the statistical analysis had not been performed.

For future works, we've therefore planned to do the followings: 1) Statistically analyze the generated patterns comparing with natural data using various analysis methods used in real plant ecology research. 2) Extend the concept of fitness landscape in *Avida* to application in vertical plant growth. 3) Experiment with other Artificial Life systems to see the ubiquity of the behavior.

References

[1] Adami, C. *Introduction to Artificial Life*, Spring-Verlag, 1998.

[2] Adami, C., Brown T. *Evolutionary Learning in the 2D Artificial Life System 'Avida'*, Artificial Life IV, p377-381, The MIT Press, 1994.

[3] Adami, C. *Ab Initio Modeling of Ecosystems with Artificial Life*, Natural Resource Modeling 15, p133-145, 2002.

[4] Adami, C., Seki, R., Yirdaw, R. *Critical Exponent of Species-Size Distribution in Evolution*, Proceeding of Artificial Life VI, Los Angeles, June 27-29, 1998.

[5] Dieckmann, U., Herben, T., Law, R. *Spatio-Temporal Processes in Plant Communities*, Interim Report, International Institute for Applied Systems Analysis (IIASA), Austria, 1997.

[6] Deussen, O. em et al., *Realistic Modeling and Rendering of Plant Ecosystems*, Proceeding of SIGGRAPH 98, Orlando, Florida, July 19-24, 1998.

[7] Greig-Smith, P., *Pattern in Vegetation*, J. Ecol., Vol. 67, p755-779, 1979.

[8] Herben, T., Hara, T. *Spatial Pattern Formation in Plant Communities*, Morphogenesis and Pattern Formation in Biological Systems - Experiments and Models, Springer-Verlag, 2003.

[9] Lane, B., Prusinkiewicz, P. *Generating Spatial Distributions for Multilevel Models of Plant Communities*, Proceeding of Graphics Interface, Calgary, Alberta, May 27-29, 2002.

[10] Miller, G. F. *Artificial Life as Theoretical Biology: How to do real science with computer simulation*, Cognitive Science Research Paper 378, University of Sussex, Brighton, UK, 1995.

[11] Ray, T. *An Approach to the Synthesis of Life*, Artificial Life II, p371-408, Addison-Wesley, 1991.

[12] Silvertown, J., Charlesworth, D. *Introduction to Plant Population Biology*, 4th ed, Blackwell Science, 2001.

[13] Sober, E. *Learning from Functionalism - Prospects for Strong Artificial Life*, Artificial Life II, p749-765, Addison-Wesley, 1991.

[14] Taylor, T., Hallam, J. *Studying Evolution with Self-Replicating Computer Programs*, Forth European Conference on Artificial Life, p550-559, The MIT Press, 1997.