Guided Ecological Simulation for Artistic Editing of Plant Distributions in Natural Scenes

Gwyneth A. Bradbury[1] Kartic Subr[2] Charalampos Koniaris[2]
Kenny Mitchell[2] Tim Weyrich[1]
[1]University College London [2]Disney Research Edinburgh

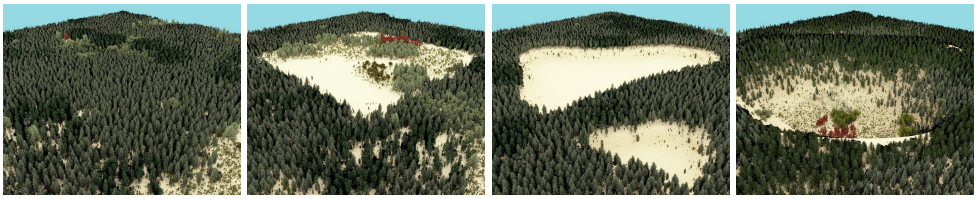


Figure 1. Our algorithm simulates realistic foliage cover for landscapes and allows for several modes of editability. From left: initial burn-in of the simulation; sparsifying a specified region; adding a lake to the landscape; simulating the same species on a different terrain.

Abstract

In this paper we present a novel approach to author vegetation cover of large natural scenes. Unlike stochastic scatter-instancing tools for plant placement (such as multi-class blue noise generators), we use a simulation based on ecological processes to produce layouts of plant distributions. In contrast to previous work on ecosystem simulation, however, we propose a framework of global and local editing operators that can be used to interact directly with the live simulation. The result facilitates an artist-directed workflow with both spatially- and temporally-varying control over the simulation's output. We compare our result against random-scatter solutions, also employing such approaches as a seed to our algorithm. We demonstrate the versatility of our approach within an iterative authoring workflow, comparing it to typical artistic methods.

1. Introduction

With digital distribution of games providing an ever-increasing storage budget, there is a strong trend toward open-world games that feature realistic, large-scale environments. Feature films also require increasingly large digital assets, including artificial landscapes extending as far as entire planets. Creating such expansive natural envi-

 ISSN 2331-7418

Term	Definition
Genotype	The underlying constitution of a species
Phenotype	The observed characteristics of an individual resulting from the interaction of its species genotype with the environment
Abiotic Factors	Physical features of a landscape
Biotic Factors	Biological features of a landscape
Endemic Species	Native or restricted to a certain region

Table 1. Definitions from ecology.

ronments requires tools for efficient authoring, capable of combining scalability with fine-grain artistic control.

Natural landscape cover is a key element of large-scale, digital environments and their appearance is often dominated by the characteristic distribution of the vegetation present. In real-world environments, however, the distribution and morphology of vegetation is the product of a set of *endemic species*, *abiotic factors* (climate, altitude, soil type, etc.) and *biotic factors* (species' adaptation and spatial and biological interaction of individual plants) [Hoffmann and Sgrò 2011; Deussen et al. 1998]. We argue that human observers are sensitive to the characteristic look of ecosystems arising from such complex interplay. We hypothesize that environmental factors can be used to efficiently create more natural vegetation cover for virtual landscapes by modelling plant responses to parameters such as spatial location, height, canopy size, and age of each instance of every species as well as abiotic parameters such as soil quality and rainfall.

Developing a tool which exploits these features and yet provides an intuitive interface for the designer poses unique challenges for digital content creation. Even under the guidance of an expert, manual control of each plant instance is an unmanageable task for large-scale environments.

Other rule-based, procedural generators (for instance, exploiting simple rules that depend on altitude and slope of the terrain [Hammes 2001]) aim to produce ecologically *plausible* results by definition. Creating procedural rules that lead to truly *realistic* plant cover is non-trivial as vegetation cover in reality develops over many years and relies upon complex interaction between the environment and its flora. In an artistic model, however, control is usually limited to changing model parameters up front and manually editing distributions post-generation.

In practice, the manual aspect of virtual landscape creation is tedious and, while commercial tools (see Table 2) bring high functionality to procedural or random-scattering approaches, editing capabilities are limited. Masking of regions and instance-based parameter editing (scale, translation, add and remove, for example) are common. In the context of a large, varied landscape, an artist may need to fine

Tool	Description
UnrealEngine [UnrealEngine 2015]	A game engine and suite of integrated tools for game design and building, simulations, and visualizations.
iToo ForestPack [iToo Software 2015]	A plug-in for Autodesk 3DS max and 3DS max design, designed to give a complete solution for creation of vast surfaces of trees.
CryEngine [CryTek 2015]	A game engine and terrain editor with a sandbox interface incorporating a vegetation tool.
Torque 3D Foliage Replicator [iToo Software 2015]	Foliage placement brush tool within an open source game engine.
L3DT [Bundysoft 2015]	Application for generating terrain maps and textures. Possesses partial environmental attributes (water flooding, water table, salinity map) and attributes map that affects the texture of the terrain.
SpeedTree [Interactive Data Visualisation 2015]	Toolkit used to create 3D animated plants and trees for games, animations, visual effects, and more.

Table 2. Environment modelling tools.

tune a large number of parameters. Furthermore, commercial tools currently support a limited understanding of ecology, restricted to incorporating simple adaption to elevation but not to further environmental factors such as soil type and water availability.

Rule-based, procedural models are an established means of content creation and appraised for their variety of results. A number of works on procedural vegetation modelling have used ecosystem simulation as a means to create more realistic plant distributions in computer graphics [Deussen et al. 1998; Ch'ng 2009b]. Their methods scale well to larger environments and achieve a degree of realism. Similar to other procedural methods, however, artistic control is mostly limited to the definition of starting conditions from which the ecosystem evolves.

Our contributions can be summarized as follows: we propose an approach to combine ecosystem simulation with editing. This allows for global and local edits of plant distributions by directly interacting with a live simulation; we draw from the state-of-the-art in ecological simulations to design a set of operators that allow for iterative artistic control of the vegetation cover while still retaining the ecological realism of the underlying simulation; finally, we expose the time-axis of the simulation. This is central to our workflow, allowing an artist to adjust the path of the simulation through intuitive editing operators that correspond to events in time and changes in biotic and abiotic factors as the simulation progresses.

Our method scales well to large environments (spanning several kilometers) and can be parameterized to create vegetation covers representative of various regions across the world. We analyze the dynamics of our simulation (Figures 9, 14) and

provide guidance for its use in an authoring context, including a number of sample edit sessions (Figures 11, 12, 13). We compare the outputs of our method to more commonly used, noise-based generators, demonstrating the more natural appearance of simulation-based plant distributions (as seen in Figure 10).

The remainder of this work is organised as follows: Section 2 discusses related literature; the simulation model we use is reviewed in Section 3; Section 4 explains how our tools are used to guide and interact with the simulation model; and finally, Sections 5, 6, and 7 discuss our findings.

2. Related Work

2.1. Commercial Tools

Many commercial tools, such as iToo [iToo Software 2015] and UnrealEngine [UnrealEngine 2015], among others (see Table 2), are available for distributing vegetation within landscapes: creative modelling tools that provide a user with fine-grain control but offer little help in ensuring ecological realism, and a set of systems that offer a much higher level of automatism in populating computer-generated landscapes with vegetation. A list of common design tools for virtual landscapes is given in Table 2. Full evaluation of these tools (and others) would be beneficial to the field.

Typical features of these tools allow automated vegetation placement dependent on terrain convexity/concavity and slope direction. Our tools expand on this functionality to include species competition among different plant types and environment factors across the terrain.

For natural environments with many species, editing with these tools is cumbersome and tends to result in scenes with a landscaped aesthetic, not dissimilar to carefully landscaped parks. That is, wild scenes, which have evolved over hundreds of years according to inherently complex and long-term interactions between vegetation and terrain, are much more difficult to create (indigenous forest or sweeping heathland, for example). Results from our algorithm applied to such wild scenes can be seen in Figure 8.

2.2. Point-sampling Approaches

Many modelling tools use point-sampling approaches as a quick and simple way to randomly distribute vegetation instances over a region, and a substantial body of work exists on studying properties of random-sample patterns for use in computer graphics. A prominent class of sample patterns for general instance distributions are blue-noise distributions [Lagae and Dutré 2008]. Wei [2010] presents an adaptation of blue-noise sampling, making it suitable for multiple object classes. The underlying Poisson disk sampling ensures that points remain a minimum distance from each other (while also

being randomly, uniformly distributed). This appears to be a desirable property of plant distributions; however, real-world plant distributions hardly ever exhibit these blue-noise characteristics [Law et al. 2009].

A small body of related work investigates how to drive random-sampling methods with real-world data. Given an existing, labelled point distribution, Öztireli and Gross [2012] present a correlation measure that can be used to generate a similar-looking point distribution from a pairwise correlation function. While it is possible to apply this method to vegetation distributions, ground-truth data is hard to collect. In aerial images, for example, only the highest canopies are observed, while underlying vegetation is obscured.

Although these point-sampling methods are fast to compute, they also suffer from a lack of domain knowledge on global plant distributions, which are strongly linked to abiotic factors such as soil type, rainfall, and temperature parameters. In contrast, work in ecolocical modelling uses domain knowledge to learn from real-world observations [Illian et al. 2009]. This allows similar plant distributions to be synthesized, but an efficient means of artist interaction with such a model is non-trivial.

2.3. Procedural Methods for Vegetation

Procedural approaches are an attractive alternative to manual modelling, with high-quality solutions available for specific landscape features including terrain [Doran and Parberry 2010], river networks [Smelik et al. 2011], plant models [Longay et al. 2012; Stava et al. 2014], vegetation distribution [Deussen et al. 1998; Hammes 2001; Dietrich et al. 2005; Decaudin and Neyret 2004], as well as urban layouts and building façades (see [Raffe et al. 2012] for a comprehensive review). While procedural methods promise an increase in productivity and variety of results, fine-grain control of an intuitive interaction is difficult to achieve.

Recent developments in procedural modelling address some of these interactivity issues, focussing on interactive procedures and control. In an urban context, layers can be used to apply transformation and merging operations using graph cuts [Lipp et al. 2011]. This allows for intuitive manipulation of layouts (drag and drop, translation, and rotation operators, etc.) which always result in valid layouts. In ecological modelling, intuitive artistic direction conflicts with the complex, temporal and spatial interaction of plant species, a problem that our work attempts to address.

In general, while procedural generators can reach a sometimes uncanny degree of realism, intuitive interaction remains a challenge in procedural modelling. We believe that a good modelling interface should offer control on both global and local scales, allowing the user to fix certain aspects while others are freely generated.

2.4. Ecology and Ecosystems

Related literature in the field of biology reports data about the populations of various species collected from real, but specific forests and desert, etc. Pommerening [2002] collects several case studies. Using such data, as well as controlled experiments, some methods develop models for plant and tree growth [Prusinkiewicz 1998] and interaction with a multitude of environmental factors [Hammes 2001]. Computational methods in the field use these models to simulate natural growth patterns. In this paper, we leverage those computational methods.

Methods such as cellular automata [Green 1997] take an Eulerian perspective and partition the domain into a grid of cells. Agent-based methods [Bornhofen and Lattaud 2009; Ch'ng 2011], on the other hand, adopt a *Lagrangian perspective* and simulate the growth by iteratively updating the interactions between individuals scattered in the domain. Finally, procedural methods identify rules and construct grammars to represent growth patterns of plants [Kurth et al. 2012].

Regardless of the type of simulation, a thorough and systematic validation of results remains a challenging problem in ecology [Pommerening 2002]. Ch'ng demonstrates, in a series of work [Ch'ng 2009a; Ch'ng 2009b; Ch'ng 2011; Ch'ng 2013], that even simplified versions of agent-based simulators are valuable for designing realistic environments with vegetation for virtual worlds. However, while such simulations produce realistic results for naturally distributed vegetation, they are difficult to control. Either they are fully deterministic and exhibit low variation without extensive manual intervention or they are stochastic with potentially hundreds of parameters that affect the final spatio-temporal distribution of populations. In this paper, we adapt Ch'ng's agent-based simulator [Ch'ng 2011] to be more suited for and to incorporate guiding of the simulation via user interaction. Further details of the simulation model are detailed in Section 3.

We define the content creation process as a chain of operators, interspersed with user guidance, for producing an ultimate, desired appearance of naturally distributed vegetation cover on landscapes.

3. Review of the Simulation Model

The identification of the distribution of flora spatially, temporally, and across species is a challenging task. The number of individuals of each species that thrive in any location depends on multiple factors such as latitude, temperature, and rainfall. In addition to the above *abiotic* factors, the distribution also depends on *biotic* factors such as mutual shade, competition for resources, and the extent to which nitrates are fixed in the soil. While no explicit models have been proposed for obtaining counts of spatio-temporal populations across multiple species, the typical approach is to resort to iterative simulation.

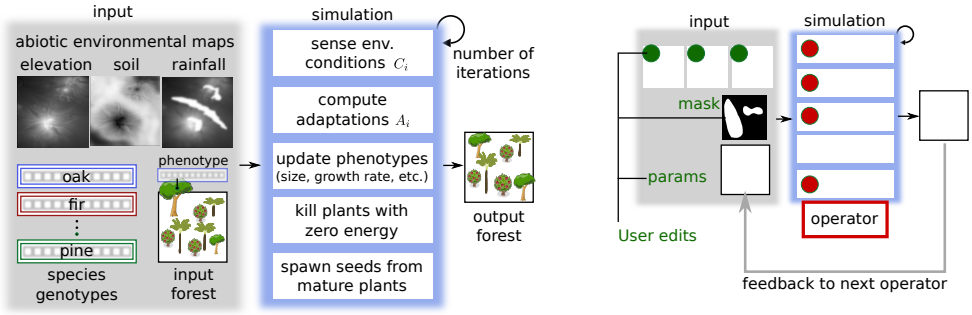


Figure 2. (a) We adapt an individual-based ecological simulator. (b) We propose a novel approach to authoring natural landscapes, by guiding the simulation process via user interaction and editing. As the simulation unravels, rich spatio-temporal statistical distributions are created. Our workflow allows users to guide the simulation, as it proceeds, by either editing the parameters directly or using our operators with simplified parameters.

Our work employs Ch'ng's [2011] state-of-the-art, individual-based, ecological simulator. The general approach is to associate a *genotype* with each species, which consists of parameters that describe its robustness to a number of biotic as well as abiotic factors. Individuals of each species then express this genotype differently, depending on environmental factors, giving rise to observable traits, or *phenotypes* such as height, leaf density, and diameter. The simulation is seeded using an initial distribution of trees and proceeds by iteratively updating the phenotypes of each in accordance with ecological models for growth and propagation. While exact simulation potentially requires infinite parameters, Ch'ng considers multiple environmental factors and ultimately proposes a simplified simulation model for generating virtual environments. In the context of our work, this lends itself particularly well to an artist-directed workflow, where precision may be sacrificed for appearance but the ecological realism of the underlying simulation is maintained.

We increase the functionality of this ecosystem model to allow interactive authoring of complex, natural landscapes. Our adaption of Ch'ng's model is explained in Section 4 and shown in Figure 2.

3.1. Genotype

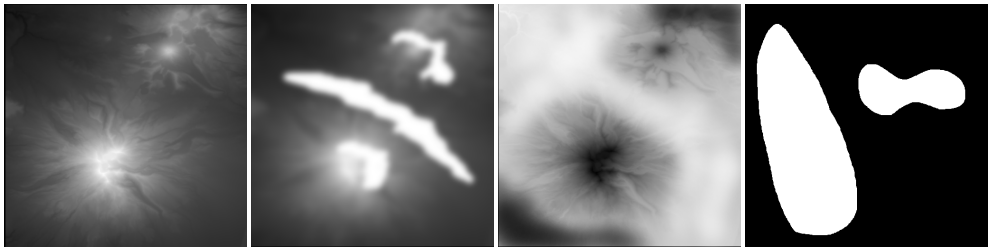
The following traits are associated with each species: minimum, maximum, and mean heights; maximum and mean ages (longevity); minimum, maximum, preferred, and robustness values for both abiotic and biotic factors; age to maturity; number of speeds spawned per seasonal quarter; and seed dispersal radius.

3.2. Phenotype

We simulate the following parameters for each individual: location (coordinates), fitness (this tolerance value is based on robustness values in the plant's genotype), energy level, rate of energy loss, growth rate, height, and canopy size (diameter).

3.3. Environmental Factors

The input to our simulator includes abiotic factors (temperature, elevation, rainfall, and soil fertility) represented as an image map covering the layout of the environment (Figure 3 shows examples) as well as biotic factors (shade and competition for resources from neighbors) represented as a per-species parameter. In combination with individual phenotype values, these lead to a fitness parameter for each instance that depends on its location and neighbors. While abiotic factors can only be altered by user interaction, biotic factors must be updated on each iteration as instances adapt to and grow within their local environment.



(a) elevation (b) rainfall (c) soil fertility (d) selection

Figure 3. The greyscale abiotic maps and selection mask used to demonstrate our operators in Section 4. Authored using commercial image-editing software.

3.4. Simulation

Trees are initialized with a fitness value of 100%, and heights and canopy sizes are drawn from the distribution specified by the corresponding species' minimum, maximum, and mean values. In each iteration of the simulation, we update the parameters in the phenotype, including the height and canopy size, according to the formulae proposed by Ch'ng. Similarly, we compute fitness of individuals by sensing the influence of environmental and neighboring plants. We refer the reader to [Ch'ng 2011] for a full definition of plant adaption to biotic and abiotic factors. Fitness is computed as the products of the adaption of a plant to each abiotic and biotic factor in each time step. When fitness is close to zero, we decrease the energy (using Ch'ngs rate of energy loss) in each iteration. Finally, when the energy in a plant is close to zero, the plant "dies," and its instance in the simulator is deleted.

4. Guiding Ecological Simulation

Vegetation cover in natural landscapes exhibits rich diversity, both in terms of the variety and number of co-existing species, as well as in the distribution thereof and interactions between individuals. The spatial statistics of real natural environments are grossly governed by two correlated factors: the environment and the suitability of each of the species to the environmental conditions. Explicit characterization of these statistics in terms of the input factors is a challenging and open problem. However, individual-based simulators, such as Ch'ng's, which we adapt in this paper, can be seen as iterative solutions to the global equations that coarsely approximate these distributions.

Ch'ng's simulator provides realistic results but control remains unwieldy. For example, introducing a simple constraint, such as creating a user-placed clearing, requires manipulation and tuning of all available abiotic maps. From an artistic perspective, this may also not lead to the desired result since species suited to the alternate abiotics will thrive in the new environment

At the other end of the spectrum, there exist tools that allow fine-grained artistic control at the level of editing individual trees [Deussen et al. 1998; Dietrich et al. 2005]. Such tools scale poorly to large environments with millions of trees (requiring each instance to be hand-tuned). While it is indeed possible to author complex and realistic natural scenes with such tools, authoring large realistic environments can consume many days of editing even for skilled artists. Procedural approaches scale well, but still suffer from lack of directability.

Our main contribution is to conceptualize the authoring process as an iterative progression of user edits interspersed with realistic, ecologically-based simulation. The aim is to enable the user or artist to "watch" the environment grow and evolve, allowing him to pause, undo, redo, or resimulate at any point while also applying simple operators to guide the next phase of simulation. These operators perform ad-hoc modifications and are applied on an artistically or stochastically generated initial forest. We implement most of these spatio-temporal operators in the parameter space of our ecologically-based simulator.

Our workflow allows the following operations:

- Spatially-varying control of overall population density by manipulating simulation speed and environmental conditions.

- Spatio-temporal editing of simulation parameters for realistic blending of simulated and non-simulated regions.

- Modifying the number of plants across a species pool.

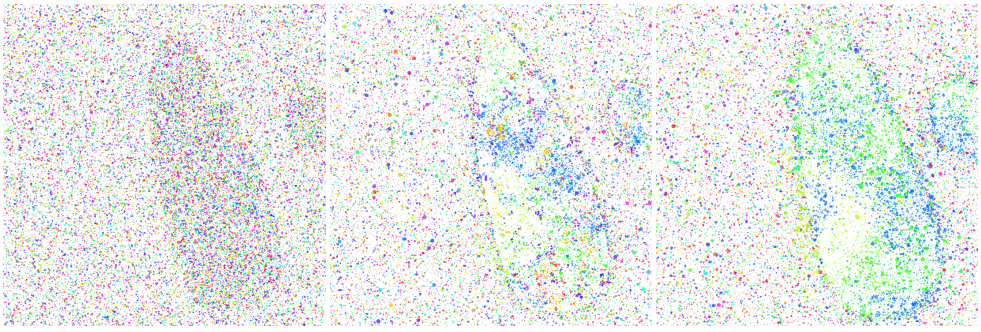
4.1. Local Operators

In this section, we describe our operators. We verify the functionalities of the simulator, present example workflows using our operators, and, finally, we discuss various aspects of the simulation and authoring process.

4.2. Explicit Editing and Cut-copy-paste

The most local edit that we allow is manual editing of phenotypes of individuals. The user may choose to add, move, or delete groups of plants as well as modify their attributes such as height, canopy size, etc. Although this edit operation allows maximum control, it can be tedious, leads to potentially unnatural results, and scales poorly with population size.

Cut-copy-paste operations (see Figure 4) are achieved using a mask and target location. Objects in a masked region are translated to a new region and either merged into or replace the existing objects.

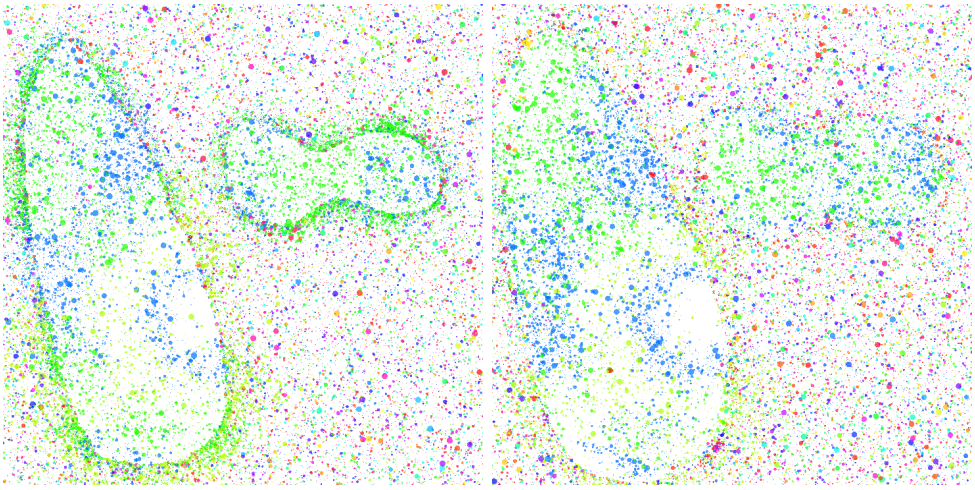


(a) copy-pasted trees (b) with simulation (c) simulation + feathering

Figure 4. Our copy-paste operator provides additional features to run the simulation within the pasted region with or without feathering the boundary. Mask shown in Figure 3 (d).

4.3. Typify Operator

We provide a typify operator (see Figure 5) that applies the simulation to a specified region of the landscape, blending seamlessly with the surroundings. We use the simulation as a spatio-temporal sculpting operator that blends the edit into its local environment. The user selects a region, represented by a binary mask. The time per iteration, or speed, of the simulator is equal to the user-specified value (typically one year) in selected (white) regions and zero in black regions. We run the simulation for a specified number of iterations, feathering the simulated duration of one iteration across the mask's boundary. The smoothness of the transition is controlled by pre-blurring the mask and allowing the user to define the diameter of the blur kernel.



(a) simulation within selection (b) boundary feathering

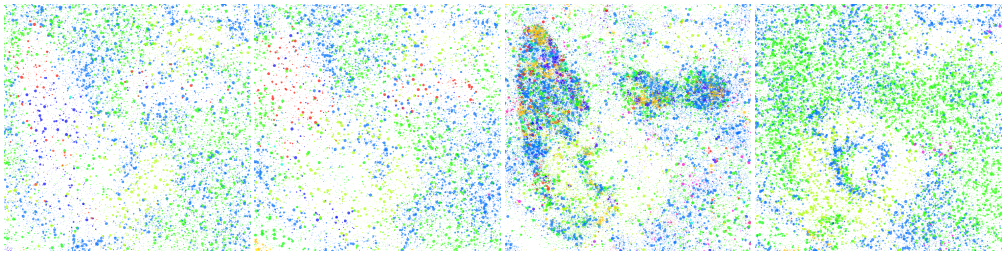
Figure 5. Our typify operator applies the simulation only to a selected region (Figure 3 (d)). (a) Using the binary mask results in an unnatural boundary where no growth can occur outside the region. (b) Feathering the boundary yields a more natural result.

4.4. Local Editing of Abiotic Maps

A set of abiotic maps are grayscale images provided to the simulator that specify spatially varying environmental conditions, such as rainfall, soil fertility, elevation (altitude), and temperature. The maps can be hand-painted by artists and used to control the simulation. Replacement maps can be incorporated at any point as the artist guides the simulation according to what has already happened. These conditions, in combination with species' genotypes, determine the endemic species that will thrive locally as well as the particular spatio-temporal distribution of the plant instances. One natural and realistic method of controlling the distribution across species as well as the plant densities is to edit these maps. For example, painting a certain region of the rainfall map with a brighter value amounts to increasing the rainfall locally (this may also be considered as increasing the moisture content of the soil, such as near to a lake or river). This, in turn, increases the density of plants that thrive under wet conditions in that region and decreases the densities of those that require drier conditions.

4.5. Sparsify and Densify Operator

Editing of abiotic maps is a natural tool for biasing the population towards (or away from) containing a higher number of individuals of certain species. To simplify the specification of spatial non-uniformity we provide a single overarching sparsify (or densify) operator which restricts (or favors) the population growth uniformly across species within the marked regions (see Figure 6). We achieve this by using the



(a) sparsify (b) + feather (c) densify (d) + feather

Figure 6. Our sparsification/densification operator allows us to create clearings or dense clusters, with or without boundary feathering, of plants and trees using a simple selection tool. The density of plants in the unselected region (black) is unaffected. Note that abiotic factors (elevation, rainfall, and soil) are respected by all our operators.

input mask to control the abiotic and biotic parameters independently as follows. For each plant or tree, we read the spatially varying sparsification control weight $w \in [0, 1]$ from the mask image and a global sparsification factor $w_{\max} > 0$. For densification, w is inverted. We then scale the adaptations of the plant to abiotic factors and the current environmental conditions for biotic factors so that plants in the black (zero) regions of the mask are unmodified. For the white (one) regions of the mask that specify w, we scale the adaptations of the plant to abiotic factors by $\gamma = 1 + w(1/w_{\max} - 1)$, which controls the plants lifetime. Similarly, for the biotic factors, we scale the current environmental conditions by $\gamma = 1 + w(w_{\max} - 1)$, since a higher value of γ corresponds to sparser growth.

This scaling of the adaption parameter for each plant affects its growth such that better adapted plants thrive more. As the parameter scales uniformly across species, all species thrive, and the result is a population that is more dense if the densify operator is used. Under sparsification, all plants are uniformly less adapted to the environment and the result is less dense.

4.6. Global Species-mapping Operator

While local operators allow fine-tuning of the spatio-temporal distributions, identifying an initial set of species for achieving a specific type of landscape is a challenging problem. An additional problem, at the global scale, is to suitably map artistically authored species to the appropriate 3D model from a database of plant and tree models. In this section, we describe how we address both of these problems.

Since there is little data available on the genotype of real plants (i.e., the location, age, height, canopy size, and species in existing forests), we constructed a scheme for mapping genotypes to climate zones analogous to Koppen-Geiger climate zones [Peel et al. 2007]. We fix the pool of species whose genotypes are supplied to our simulator and run simulations with different values for the abiotic factors. For each combination

of abiotic factors, we identify the corresponding Koppen-Geiger climate zone and observe which of the species from our pool survive. The surviving set of species is then associated with the input climate zone. This allows users to choose a "preset" climate zone for the initial set of plants or trees. Figure 8 depicts the sets of species that we identified from a random species pool. We then labelled the species with the corresponding climate zones. Also shown in the figure, is a further fine control of the abiotics to introduce spatial variation.

We use plant models from the XFrogPlants database [XFrog 2015], consisting of 600 species from multiple world regions and climate zones, and we map each species to its nearest neighbor in the 3D model database with respect to mean height, mean age, and canopy size, as well as climate-zone preference. Categorizing models by region allows us to restrict models to those from a desired region or climate. While this is a simplistic means of achieving such a mapping, it already provides visually consistent environments. We expect future work to improve upon this mapping operation.

5. Results

5.1. Implementation and Visualization

We implemented our simulator in C++ and generated all the results on an Intel Core i7 (1.6G Hz) with 16 GB RAM. Since each iteration of the simulation feeds on the output of the previous iteration, the simulation is difficult to parallelize. We mitigate

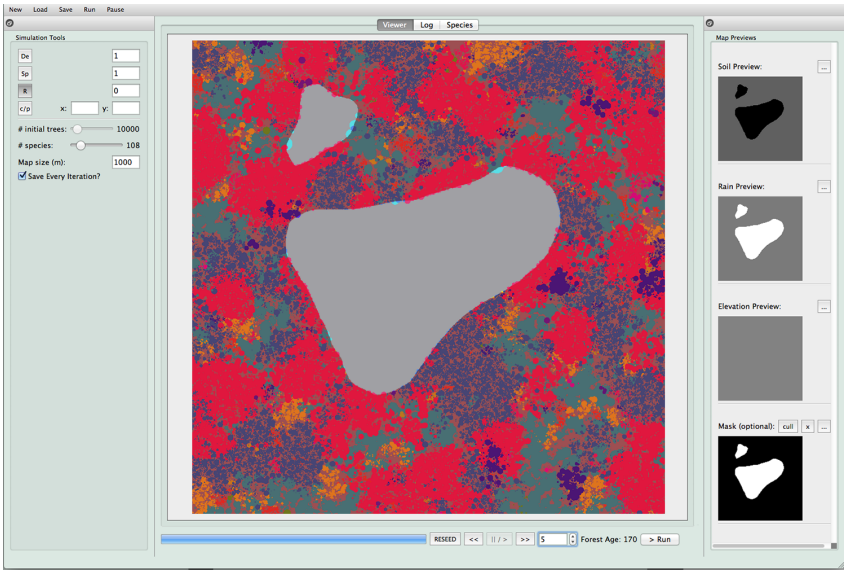


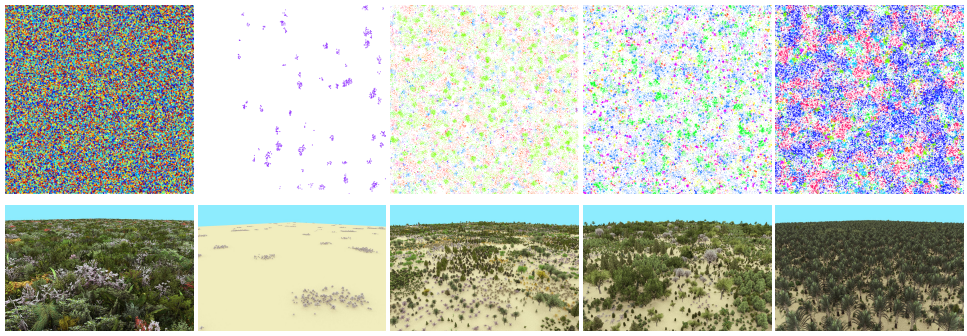
Figure 7. Our user interface.

this by performing basic optimization, such as pre-computation and lazy updates of abiotic adaptations, as well as by using approximate nearest neighbors [Muja and Lowe 2009a; Muja and Lowe 2009b] to prune searches for computing local biotic conditions, such as shade and competition for space. On average, each iteration of our simulator functions at about 400 K trees per second. This could be improved by better parallelization of instance updates or by vectorizing the adiabatic feature maps.

Our user interface (shown in Figure 7) provides a basic visualization of the simulation, showing a color-mapped scatter plot of the emerging distribution. We later render the scene using Blender's Cycles Render Engine [Blender 2015]. The simulation can typically generate over 100,000 tree instances, which may be a problem for some rendering applications, but this is mitigated with varied level-of-detail rendering.

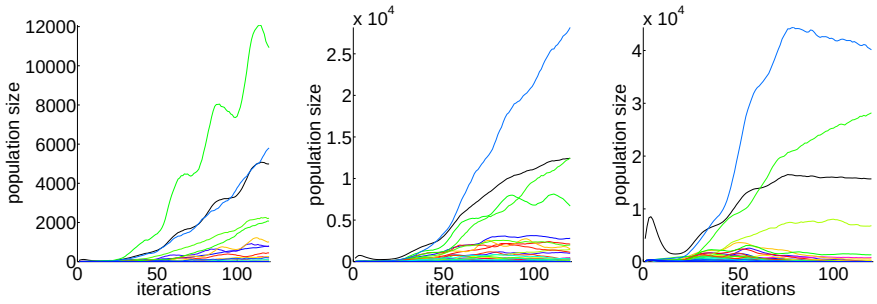
5.2. Validation

First, we supplied our simulator with the genotypes of 150 randomly generated, diverse species and varied the abiotic factors to match known biomes such as desert, boreal forest, temperate, and tropical environments. Figure 8 visualizes the input forest (randomly scattered plants with randomly chosen phenotypes) and the output of our simulator under each of the above conditions. This example illustrates the complex interplay between the abiotic factors resulting in non-trivial distributions of the population across canopy sizes, space, time, and species. The iterative nature of our simulator helps achieve this, propagating biotic factors, such as shade and competition, for space among neighbors.



(a) (b) (c) (d) (e)

Figure 8. One of our global operators allows easy selection of preset abiotic factor combinations that correspond to Koppen-Geiger climate-zone classifications. Model selection is similarly based on the selected climate zone. These can then be combined with spatially varying modulating maps for the independent abiotic factors. We show that using just two abiotic maps already introduces a complex interplay between the species. (a) Random initialization; (b) desert; (c) boreal forest; (d) temperate; (e) tropical.



(a) 1 K initial trees (b) 10 K initial trees (c) 60 K initial trees

Figure 9. Analysis of the effect of initial number of trees over a 120-year simulation. Plots are shown of the total population (thick black lines scaled down for visualization) and the populations of each of the species.

Figure 9 plots species counts over time for different numbers of starting instances in the same region. The figure suggests that the population stabilizes within 100 iterations only for large numbers of starting instances.

5.3. Comparison

Figure 10 compares results from our simulator after 10 years (third column) and 100 years (last column) against uniform random distributions (first column) and Poisson disk sampling (second column). We initialized our simulation with random samples on the left half and Poisson disk samples on the right. Our simulation was run with constant abiotic factors within the domain. Several problems are immediately apparent with the naïve stochastic methods: first, they contain a "regular" look that leads to unrealistic vegetation cover; second, noise models are insufficient for representing the distribution of the population across the many species; finally, when abiotic factors such as temperature and elevation are known, it is unclear how to determine parameters that lead to realistic appearance when using noise distributions. Even with sophisticated methods such as multi-class blue noise, this problem of selecting parameters according to biotic and abiotic variation is a challenging and open problem.

5.4. Sample Workflow

Figures 11, 12 and 13 show sample workflows using the tools our simulator provides. Both simple and complex abiotic maps are used and altered, and the sparsify and densify operators are demonstrated.

Note that even though the abiotic distributions contain sharp boundaries, the effect of the simulator is to mute these, creating natural distributions, even at these boundaries. This can be seen in Figures 11 and 13.

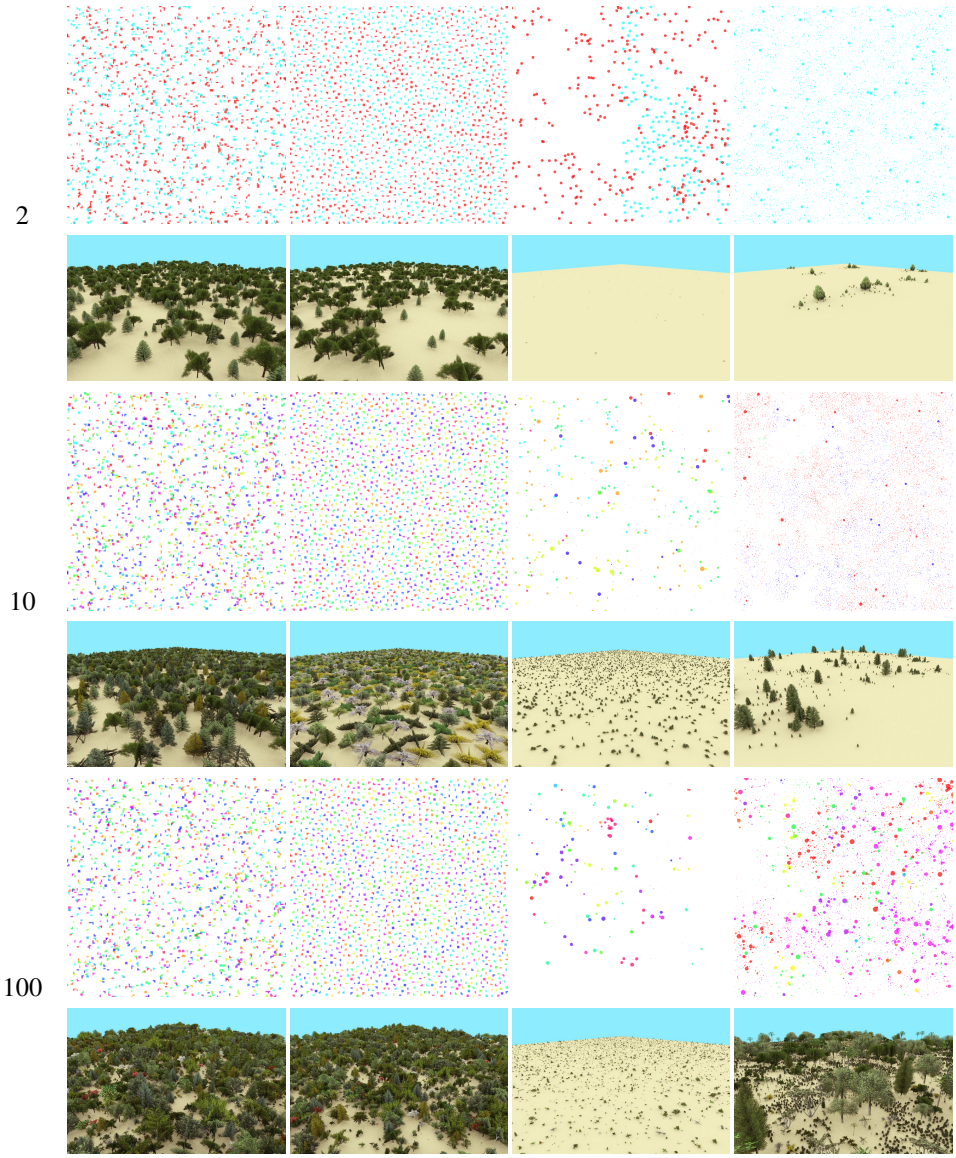
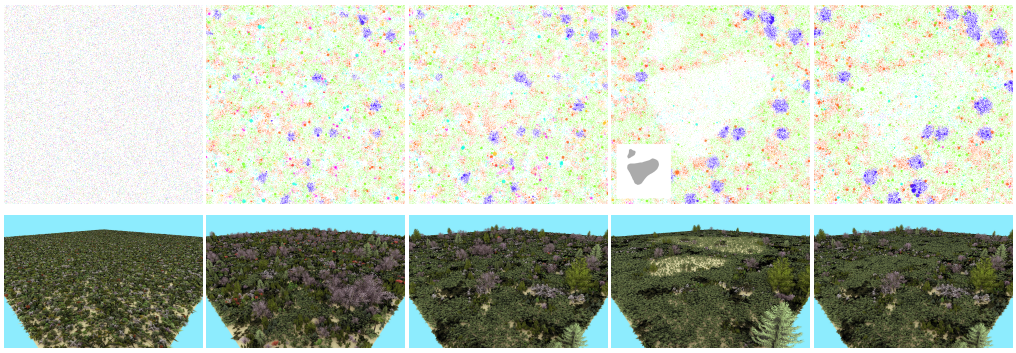
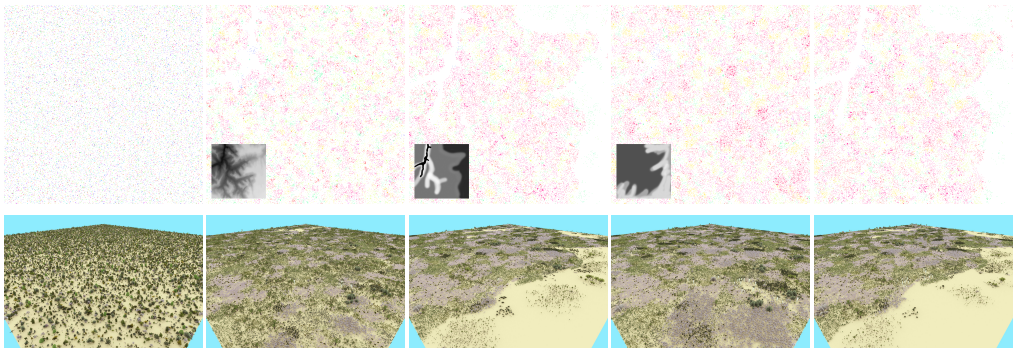


Figure 10. Comparison against random and blue noise for different species pool sizes. (a) Uniform, random noise, (b) Multi-class blue noise, (c) Ours after 10 years, (d) Ours after 100 years. Our method is shown after 10- and 100-year simulation periods. Plots represent the full $500\ \mathrm{m}^2$ area and renders show a $50\ \mathrm{m}^2$ crop.



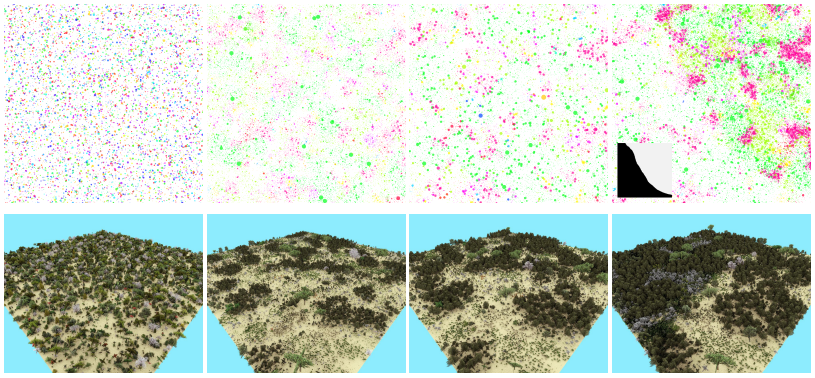
(a) (b) (c) (d) (e)

Figure 11. Workflow demonstrating changes made to the abiotic maps and the sparsify operator. (a) Initial state; (b) burn-in (120 years); (c) adapt to new abiotics, allowing different species to dominate; (d) sparsify region by changing the abiotic maps (inset) kills off certain species in the altered region; (e) grow for a further 10 iterations, species adapted to the new abiotic conditions thrive in the new environment. Rendered from the NW corner.



(a) (b) (c) (d) (e)

Figure 12. Workflow demonstrating different operators applied over more complex abiotic maps. (a) Initial state; (b) burn-in (120 years), initial soil map (inset); (c) adapt to new soil map (inset); (d) adapt to new rain map (inset); (e) adapt to both. Scene rendered from the SE corner.



(a) (b) (c) (d)

Figure 13. Workflow demonstrating the densify operator. (a) Initial state; (b) burn-in (120 years); (c) adapt to new, uniform abiotics; (d) densify NE side. Rendered from NW.

6. Observations and Discussion

6.1. Simulation Phases

We observed that the simulator typically progresses in three phases, as can be seen in Figure 14. The phases are described below.

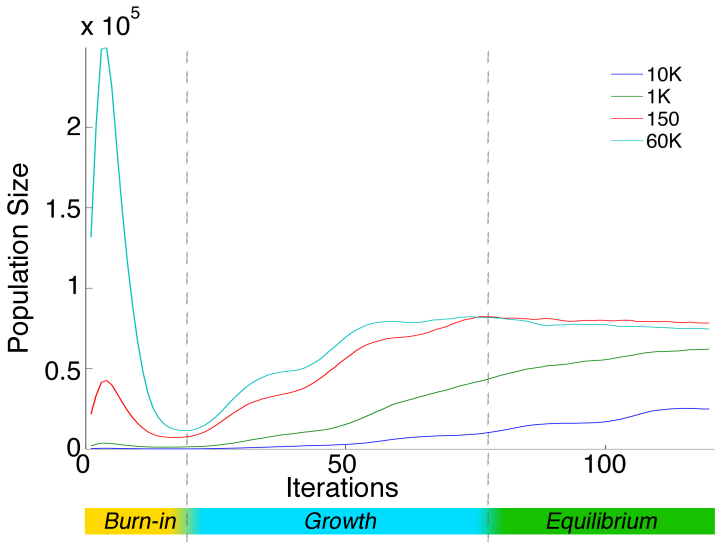


Figure 14. Comparison of the total population size over a 120 year simulation when using different numbers of initial trees (numbers in legend). Different phases of growth are highlighted with a gradient scale (as there is some variation between experiments).

Burn-in In this *burn-in phase*, plants that are shaded by neighbors and plants that are not strong enough compared to their neighbors are killed. These biotic factors are different from hardcore processes (minimum radius constraint), because they depend on the robustness of the species (determined when the species bank is randomly generated—but could additionally be user-controlled), the size and nature of the neighbors, and spatially-varying abiotic factors.

Growth Plants grow to maturity and spawn seeds that settle in the clearings created by the dead plants. This phase corresponds to a steep growth of the overall population.

Equilibrium This is an equilibrium phase where the ecosystem typically behaves as a *dynamical system*.

The simulated phases correspond well to growth in natural ecosystems [Bornhofen and Lattaud 2009].

6.2. Equilibrium/Stability of Simulation

Not every simulation results in an equilibrium phase (initial plants quickly die out, for example, if the species supplied cannot survive in the specific abiotic conditions). In another scenario, if the genotype lists a high age to maturity and a low lifespan, plants are likely to die out before they spawn seeds. Another case, where we observed a perishing population, was when there were too few species with low diversity. Typically, we find that an equilibrium is attained, even with randomly generated species pools, provided that a large number of species (about 100) are included. Figure 15 demonstrates the effect of initialization on reaching this equilibrium. In effect, the system benefits from larger numbers of initial instances.

6.3. Editability during Simulation

Our key contribution is the ability for users to guide the simulation by performing edit operations between iterations. For example, a user may begin by choosing a million randomly distributed plants and trees in an environment determined from abiotic control maps and then run a few iterations of the typification operator to adjust the overall look. Once this is done, in a few seconds, they may perform interactive edits on a subregion that is particularly important. The set of preferred species in that locale may be controlled by editing abiotic maps, and the spatial distribution may be controlled by supplying masks. After running a few iterations (less than a second), the user is provided with feedback and can iteratively perform edits that guide the simulation process. The workflow is therefore a sequence of edit operators interspersed with the simulation, providing control to the user to guide the simulation to achieve a desired "look." (See Figures 11, 12, 13).

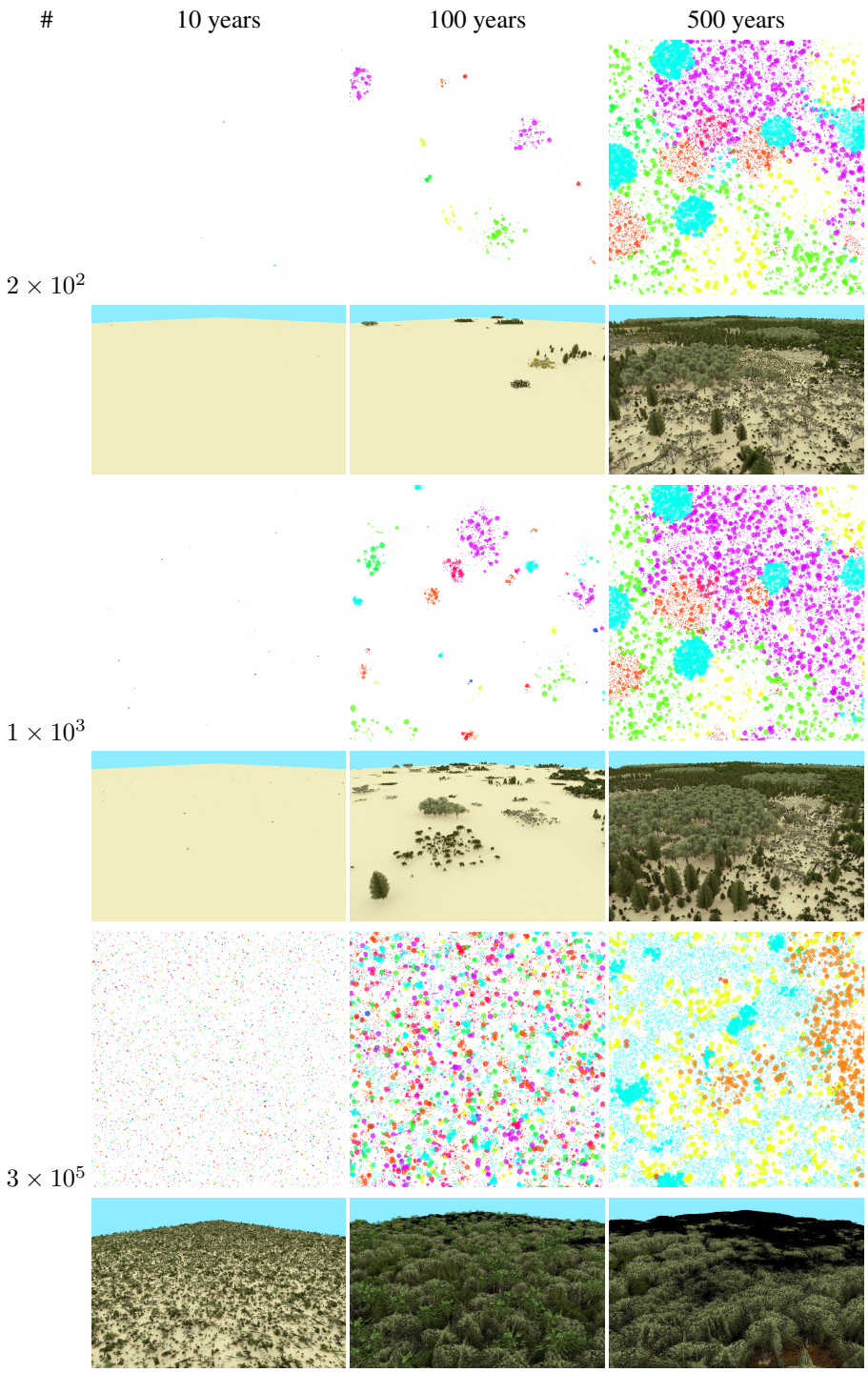


Figure 15. Effect of initialization: the system stabilization rate is a function of the number of initial instances, stabilizing more quickly and completely with a larger number of initial instances (the number of initial instances is indicated in left column, scale: $250\ \mathrm{m}^2$).

6.4. Initial Distribution of Vegetation

Our workflow can be viewed as complementary to existing methods that generate vegetation cover, because it can be applied to any initial set of plants or trees. We experimented with different initializations, such as random sampling, multi-class blue noise, and pre-authored forests. Choosing too few initial plants causes unnatural clusters of plants, each of a particular species, and requires a long burn-in phase. Choosing too many initial plants also results in a long burn-in phase since the simulation takes longer to stabilize, but it does result in more natural-looking distributions.

6.5. Simulation Time-step

Ch'ng's simulator may be viewed as an iterative solution to a complex set of differential equations that approximate the distribution of plants across space and species. While it is hard to fathom the nature of the governing equation, as the time associated with each simulation iteration is reduced, we could imagine that the simulation approaches the correct solution to the approximate equations. Although larger time steps result in faster simulation, inaccuracies due to the higher-order effects (larger scope than immediate neighbors) are noticeable. We found that using a one-year time-step was a good compromise between efficiency and accuracy.

6.6. Extending Abiotic Factors

Although we consider four different abiotic factors in this paper, our work supports a potentially large number of additional factors and extending to include additional factors is straightforward. The procedure involves including L (lower value), U (upper value), p (preferred value), and b (resilience) for each additional factor in the genotype of every species. A second change is to compute adaptations for each of these factors (for every tree) and additionally multiply the fitness of the individual in every iteration with the adaptations.

6.7. Limitations

We make the following assumptions in our adaptation of Ch'ng's simulator

- Although Ch'ng's recent model [Ch'ng 2013] encompasses animals, we restrict ourselves to biotic factors that are due to flora. We do not consider factors such as seed dispersion by animals, nitrates added to the soil by animal waste, etc.

- We consider a limited set of abiotic factors in our demonstration. It is, however, trivial to extend this set (see above).

- We restrict our simulations to species that are less resilient to crowding, since the distributions of such species are more interesting (resilient species such

as grass tend to be more evenly distributed and can be solved using common rendering optimization techniques [Decaudin and Neyret 2004], for example).

7. Conclusion

In summary, our work demonstrates a novel approach to vegetation authoring for virtual environments. In contrast to commercially available tools, our method recognizes species' properties and simulates placement according to simplified ecological rules, which results in a more natural arrangement of species' instances. This iterative process leads to a more natural overall arrangement of species, and the user can interact with and guide the simulation to a desired aesthetic.

In comparison to both SpeedTree [Interactive Data Visualisation 2015] and iToo [iToo Software 2015], we offer a new means of editing large landscapes according to artist direction while maintaining the underlying ecological realism of the simulation. Furthermore, compared to iToo, we require no support objects (splines and polygonal geometry) for different distributions but rather only a handful of grayscale maps corresponding to the whole scene.

We present editing operators that are both global and local: combining ecological simulation with a versatile editing framework that encourages direct interaction. Global operators allow the simulation to be parametrized and to create vegetation cover representing different biomes, while local operations create different effects on a smaller area within the scene.

This work would benefit from detailed user studies comparing results to similar output using commercial tools such as Unreal Engine 4 in order to substantiate and develop claims made here.

To expand our work, further calibration is still required. Calibration of the abiotic and biotic parameters to real-world environments would better allow a user to create a determined environment type rather than having to hone in more iteratively on the desired result. Our work would also benefit from user testing to demonstrate its application to a typical production pipeline.

Acknowledgements

This work was supported by the UCL EngD VEIV Centre for Doctoral Training as well as by funding from both Disney Research and the Engineering and Physical Sciences Research Council (EPSRC).

References

BLENDER, 2015. Cycles Render Engine. Blender Reference Manual. URL: http://www.blender.org/manual/render/cycles/. 41

BORNHOFEN, S., AND LATTAUD, C. 2009. Competition and evolution in virtual plant communities: a new modeling approach. *Natural Computing 8*, 2, 349–385. URL: http://dx.doi.org/10.1007/s11047-008-9089-5. 33, 46

BUNDYSOFT, 2015. Large 3D Terrain Generator. Homepage, L3DT. URL: www.bundysoft.com/L3DT/. 30

CH'NG, E. 2009. A behavioral agent model for synthesizing vegetation distribution patterns on 3D terrains. *Applied Artificial Intelligence 23*, 1 (Jan.), 78–102. URL: http://dx.doi.org/10.1080/08839510802573616. 33

CH'NG, E. 2009. Ground cover and vegetation in level editors: learning from ecological modelling. In *Proceedings of 14th International Conference on Computer Games: AI, Animation, Mobile, Interactive Multimedia, Educational & Serious Games (CGAMES)*, University of Wolverhampton, School of Computing and Information Technology, Wolverhampton, UK, 100–105. 30, 33

CH'NG, E. 2011. Realistic placement of plants for virtual environments. *IEEE Computer Graphics and Applications 31*, 4 (July), 66–77. URL: http://dx.doi.org/10.1109/MCG.2010.42. 33, 34, 35

CH'NG, E. 2013. Model resolution in complex systems simulation: Agent preferences, behavior, dynamics and n-tiered networks. *Simulation 89*, 5 (May), 635–659. URL: http://dx.doi.org/10.1177/0037549712470582. 33, 48

CRYTEK, 2015. CryEngine homepage. URL: www.cryengine.com/. 30

DECAUDIN, P., AND NEYRET, F. 2004. Rendering Forest Scenes in Real-Time. In *Proceedings of Eurographics Symposium on Rendering*, Eurographics Association, Aire-la-Ville, Switzerland, A. Keller and H. W. Jensen, Eds., 93–102. URL: http://hal.inria.fr/inria-00537458. 32, 49

DEUSSEN, O., HANRAHAN, P., LINTERMANN, B., MĚCH, R., PHARR, M., AND PRUSINKIEWICZ, P. 1998. Realistic modeling and rendering of plant ecosystems. In *Proceedings of SIGGRAPH*, ACM, New York, 275–286. URL: http://doi.acm.org/10.1145/280814.280898. 29, 30, 32, 36

DIETRICH, A., COLDITZ, C., DEUSSEN, O., AND SLUSALLEK, P. 2005. Realistic and interactive visualization of high-density plant ecosystems. In *Proceedings of the First Eurographics Conference on Natural Phenomena*, Eurographics Association, Aire-la-Ville, Switzerland, NPH'05, 73–81. URL: http://dx.doi.org/10.2312/NPH/NPH05/073-081. 32, 36

DORAN, J., AND PARBERRY, I. 2010. Controlled procedural terrain generation using software agents. *IEEE Trans. Computational Intelligence and AI in Games 2*, 2 (June), 111–119. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5454273. 32

GREEN, D. 1997. Modelling plants in landscapes. In *Plants to Ecosystems – Advances in Computational Life Sciences*, M. Michalewicz, Ed. CSIRO Division of Information Technology, Melbourne, Australia, 85–96. URL: http://www.researchgate.net/publication/246854314_Modelling_plants_in_landscape. 33

HAMMES, J. 2001. Modeling of ecosystems as a data source for real-time terrain rendering. In *Proc. Intl. Symp. on Digital Earth Moving*, Springer-Verlag, London, DEM '01, 98–111. URL: http://dl.acm.org/citation.cfm?id=647811.738117. 29, 32, 33

HOFFMANN, A. A., AND SGRÒ, C. M. 2011. Climate change and evolutionary adaptation. *Nature 470*, 7335, 479–485. 29

ILLIAN, J., MLLER, J., AND WAAGEPETERSEN, R. 2009. Hierarchical spatial point process analysis for a plant community with high biodiversity. *Environmental and Ecological Statistics 16*, 3, 389–405. URL: http://dx.doi.org/10.1007/s10651-007-0070-8. 32

INTERACTIVE DATA VISUALISATION, 2015. SpeedTree Animated Trees and Plants Modelling and Rendering. URL: www.speedtree.com. 30, 49

ITOO SOFTWARE, 2015. Forest pack 4. URL: www.itoosoft.com/forestpack.php. 30, 31, 49

KURTH, W., KNIEMEYER, O., AND SLOBODA, B. 2012. Forest structure, competition and plant-herbivore interaction modelled with relational growth grammars. *Lesnicky Casopis - Forestry Journal 58*, 2, 75–91. URL: http://www.degruyter.com/view/j/forj.2012.58.issue-2/v10114-011-0012-0/v10114-011-0012-0.xml, doi:10.2478/v10114-011-0012-0. 33

LAGAE, A., AND DUTRÉ, P. 2008. A comparison of methods for generating Poisson disk distributions. *Computer Graphics Forum 27*, 1 (Mar.), 114–129. URL: http://www.researchgate.net/publication/220506140_A_Comparison_of_Methods_for_Generating_Poisson_Disk_Distributions. 31

LAW, R., ILLIAN, J., BURSLEM, D. F. R. P., GRATZER, G., GUNATILLEKE, C. V. S., AND GUNATILLEKE, I. A. U. N. 2009. Ecological information from spatial patterns of plants: insights from point process theory. *Journal of Ecology 97*, 4, 616–628. URL: http://dx.doi.org/10.1111/j.1365-2745.2009.01510.x. 32

LIPP, M., SCHERZER, D., WONKA, P., AND WIMMER, M. 2011. Interactive modeling of city layouts using layers of procedural content. *Computer Graphics Forum (Proc. Eurographics) 30*, 2 (Apr.), 345–354. URL: http://doi.wiley.com/10.1111/j.1467-8659.2011.01865.x. 32

LONGAY, S., RUNIONS, A., BOUDON, F., AND PRUSINKIEWICZ, P. 2012. Treesketch: Interactive procedural modeling of trees on a tablet. In *Proceedings of the International Symposium on Sketch-Based Interfaces and Modeling (SBIM)*, Eurographics Association, Aire-la-Ville, Switzerland, SBIM '12, 107–120. URL: http://dl.acm.org/citation.cfm?id=2331067.2331083. 32

MUJA, M., AND LOWE, D. G. 2009. Fast approximate nearest neighbors with automatic algorithm configuration. In *VISAPP International Conference on Computer Vision Theory and Applications*, INSTICC Press, 331–340. 41

MUJA, M., AND LOWE, D. G., 2009. FLANN, Fast library for approximate nearest neighbors. URL: http://www.cs.ubc.ca/research/flann/. 41

ÖZTIRELI, A. C., AND GROSS, M. 2012. Analysis and synthesis of point distributions based on pair correlation. *ACM Transactions on Graphics 31*, 6 (Nov.), 170:1–170:10. URL: http://doi.acm.org/10.1145/2366145.2366189. 32

PEEL, M. C., FINLAYSON, B. L., AND MCMAHON, T. A. 2007. Updated world map of the Köppen-Geiger climate classification. *Hydrology and Earth System Sciences Discussions 4*, 2, 439–473. URL: https://hal.archives-ouvertes.fr/hal-00305098/document. 39

POMMERENING, A. 2002. Approaches to quantifying forest structures. *Forestry 75*, 3, 305–324. URL: http://www.researchgate.net/publication/31197575_Approaches_to_quantifying_forest_structures. 33

PRUSINKIEWICZ, P. 1998. Modeling of spatial structure and development of plants: a review. *Scientia Horticulturae 74*, 1, 113–149. URL: http://www.researchgate.net/publication/239553350_Modeling_of_spatial_structure_and_development_of_plants. 33

RAFFE, W. L., ZAMBETTA, F., AND LI, X. 2012. A survey of procedural terrain generation techniques using evolutionary algorithms. In *Proc. of IEEE Congress on Evolutionary Computation*, IEEE, 1–8. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6256610. 32

SMELIK, R., TUTENEL, T., DE KRAKER, K., AND BIDARRA, R. 2011. A declarative approach to procedural modeling of virtual worlds. *Computers & Graphics 35*, 2 (Apr.), 352–363. URL: http://linkinghub.elsevier.com/retrieve/pii/S0097849310001809. 32

STAVA, O., PIRK, S., KRATT, J., CHEN, B., MECH, R., DEUSSEN, O., AND BENES, B. 2014. Inverse procedural modelling of trees. *Computer Graphics Forum 33*, 6, 118–131. URL: http://dx.doi.org/10.1111/cgf.12282. 32

UNREALENGINE, 2015. UnrealEngine 4 homepage. URL: www.unrealengine.com/. 30, 31

WEI, L.-Y. 2010. Multi-class blue noise sampling. *ACM Transactions on Graphics (Proc. SIGGRAPH) 29*, 4 (July), 79:1–79:8. URL: http://doi.acm.org/10.1145/1778765.1778816. 31

XFROG, 2015. XFrogPlants Model Database. URL: www.xfrog.com. 40

Index of Supplemental Materials

A video of example simulations is included in "Guided Ecological Simulation.mov" available on youtube (https://youtu.be/JL1TisRgbKA).

Author Contact Information

Gwyneth A. Bradbury
Computer Science Department
University College London
Gower Street
WC1E 6BT, UK
gwyneth.bradbury@gmail.com
gwyneth-bradbury.com

Kenny Mitchell
Disney Research
15 South College Street
Edinburgh
EH8 9AA, United Kingdom
kenny.mitchell@disneyresearch.com

Kartic Subr
William Perkin Building, Room 2.24
School of Engineering & Physical Sciences
Heriot-Watt University
Edinburgh
EH14 4AS, United Kingdom
kartic@gmail.com

Charalampos Koniaris
Disney Research
15 South College Street
Edinburgh
EH8 9AA, United Kingdom
ckoniaris@disneyresearch.com

Tim Weyrich
Computer Science Department
University College London
Gower Street
WC1E 6BT, United Kingdom
t.weyrich@csucl.ac.uk
www.cs.ucl.ac.uk/staff/T.Weyrich

Gwyneth A. Bradbury, Kartic Subr, Charalampos Koniaris, Kenny Mitchell, Tim Weyrich, Guided Ecological Simulation for Artistic Editing in Natural Scenes, *Journal of Computer Graphics Techniques (JCGT)*, vol. 4, no. 4, 28–53, 2015
http://jcgt.org/published/0004/04/02/

Received:	2014-12-11		
Recommended:	2015-05-22	Corresponding Editor:	Natalya Tatarchuk
Published:	2015-11-19	Editor-in-Chief:	Marc Olano

© 2015 Gwyneth A. Bradbury, Kartic Subr, Charalampos Koniaris, Kenny Mitchell, Tim Weyrich (the Authors).
The Authors provide this document (the Work) under the Creative Commons CC BY-ND 3.0 license available online at http://creativecommons.org/licenses/by-nd/3.0/. The Authors further grant permission for reuse of images and text from the first page of the Work, provided that the reuse is for the purpose of promoting and/or summarizing the Work in scholarly venues and that any reuse is accompanied by a scientific citation to the Work.

