

Fast Large-Scale Ecosystem Simulation

C. Louis-Itty¹, E. Guérin¹, A. Peytavie², and E. Galin²

¹INSA Lyon, CNRS, LIRIS, France

²Université Lyon 1, CNRS, LIRIS, France

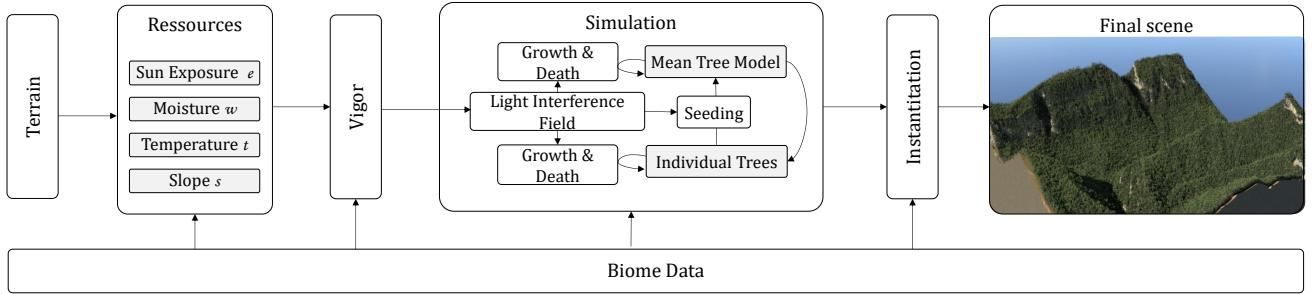


Figure 1: An overview of the ecosystem simulation: natural ressources are used to compute the vigor of the different species, that is used alongside the Light Interference Field to compute the growth of the individual trees and of the mean tree model. Our implementation on graphics hardware provides a factor of acceleration of more than 50k compared to existing methods

Abstract

Vegetation makes virtual worlds come alive. Given the number and the complexity of the interactions that factor in the development of an ecosystem, and given the need for increasingly large virtual worlds, there is a need for a fast and realistic simulation method that can process large terrains and compute the vegetation growing on them. We propose a grid-based method that simplifies the simulation of the smaller vegetation by using a mean tree model, and computes the interactions between individuals more easily using a convolution-like operation. This results in a method that is scalable, operates on graphics hardware and outperforms most ecosystem simulation methods by orders of magnitude.

CCS Concepts

•Computing methodologies → Agent / discrete models;

1. Introduction

The creation of realistic, natural ecosystems is a perennial challenge that has been addressed for decades in Computer Graphics. The existing solutions have limitations: they usually account for a limited number of factors and are expensive, thus severely limiting the size of the terrain they can be applied on.

Ecosystems involve complex interactions, as they are the result of a competition for ressources between individuals. The ressources needed by a plant to grow properly are numerous, including access to sunlight, water and soil nutrients, while having suitable air and soil temperatures. Those are called abiotic factors. They contrast with biotic factors, which stands for the effects of other plants, animals, insects, and other living organisms. As a result, resource availability for a plant can be limited by both resource availability

and the competition with neighboring individuals, making realistic simulations complex to build.

To avoid expensive simulations, procedural generation is used as an easy way to generate vegetation for large areas very quickly [Ham01, AD06, WHDS13, dNFP18, ENCC*21]. The abiotic factors are used to compute a density value for each species. Then, the densities are sampled to get positions for the individual plants. Those methods allow easy artistic authoring and quick iterations, but suffer from the lack of consideration for the biotic factors. For instance, the tendency of individuals of the same species to grow in clusters is generally not reproduced. Moreover, it can only provide a snapshot of the ecosystem at a given time, without any growth history, which makes it impossible to observe phenomena such as succession or self-thinning.

To get more accurate results, every individual can be simulated. The degree of precision may vary: some simulate the growth of the entire trees using branch modules [MHS^{*}19], while others use approximations, typically representing the tree by its height and its crown radius [DHL^{*}98, LP^{*}02, BE03, AD05, Ch^{*}11, PGG^{*}24]. Those methods are similar in that every time step, individual plants grow using the resources available to them. Therefore, the biotic factors are naturally taken into account. This method is more expensive because of the amount of data needed and updated at each time step, and because the interactions with neighboring plants are expensive to compute.

To avoid simulating on the entire terrain, other methods use point process statistics to reproduce the results of a sandbox simulation on a larger terrain. The input data can be a single simulation containing a good variety of environmental conditions [EVC^{*}15] or several simulations using a set of environmental conditions computed by a clustering of the terrain [GLCC17]. The issue with those methods is that the reconstruction of the point processes are made using expensive and iterative algorithms.

In the field of Ecology, simulations are commonly used. As can be expected, there exists a much greater variety of methods than in Computer Graphics. The crucial difference is that the expected results are not the same: what is studied is either the amount of biomass or a measure of density such as fractional plant coverage. As such, many simulators cannot be reused in Computer Graphics to produce realistic results since positions for individual plants are expected. However, there also exists methods, used for small or medium sized terrains that simulate individuals. They are more detailed, and can provide inspiration for improvement in Computer Graphics simulations.

Since the goal is to be more efficient, we took inspiration from the landscape-scale ILAND simulator [SRSS12, SSR^{*}12]. The idea is to approximate the smaller plants (with $h < 4m$) with a mean tree model: for every species, only a single individual per species per $4m^2$ cell is simulated, to represent the small vegetation. Using statistics, the number of individual in each cell can then be approximated. In this method, the simulation of a lot of individuals is avoided. Moreover, other interactions like seeding and shading are based on the same grid.

Our contributions include:

1. A fast method for the simulation of large ecosystems: we reduce the complexity of the simulation by using statistics for trees smaller than the $4m$ limit. This accelerates the computation of the seeding and shade made by the individuals.
2. The parallelization of the method on GPU, allowing for an acceleration of 4 orders of magnitude compared to the CPU base.

2. Method

At the core of the method is the simulation of individual trees, but based on a grid made of 2m wide cells. The trees are very simplified, and represented simply by a height and a species. We define mature individuals as the trees with a height greater than 4 meters. A first optimization is that we only simulate a single mature tree per cell. This makes sure that the data structures match the underlying grid.

The younger trees ($h < 4m$) are much more numerous in comparison to more mature ones. A simplification is the use of a Mean Tree Model: we only simulate the mean tree for a given species in a certain area. As a result, the underbrush is simulated with only a single tree per species, per 2m-wide cell. This means that the computational resources and memory can be focused on the mature individuals. Besides, the larger individual have a greater impact on visual realism of the ecosystem. For the smaller trees, the number of distinct individuals can be estimated easily from the average individual using a statistical rule [Rei33, PB05].

2.1. Ressources

The basis for a plant simulation is the combination of the available natural resources and the environmental conditions. They will be the primary determining factor in the growth of both the individuals and the ecosystem at large. The abiotic factors we use are the temperature t , the soil moisture w , the sunlight exposure e (hours of sun per day) and slope s . As is common in computer graphics, the slope is used as an easy proxy for the soil texture. Like in [KGG^{*}20], the abiotic factors are combined together to form the vigor $v_i \in [-a, 1]$. a is the maximum stress level. Generally, we use $a = 0.2$. The vigor tells if a particular species will be well-adapted (vigorous) or not:

$$v_i(p) = \min(f_t \circ t(p), f_w \circ w(p), f_e \circ e(p), f_s \circ s(p))$$

with f_t, f_w, f_e and $f_s \in [-a, 1]$ are response functions of the plant to the different abiotic factors. They take the form $f = g \circ d$, with:

$$g(d) = (1 + a)e^{(d/r_s)^{4.5} \ln(0.2)} - a \quad \text{with } d(x) = |x - c_s|$$

where the parameter r_s indicates at which distance from the ideal condition c_s the maximum stress a is reached.

A significant difference to previous methods is that the natural resources used to compute the vigor are not adjusted according to the biotic factors (competition of the tree crown for light and of the roots for access to water and nutrients). Instead, we use an estimation of the impact mature trees have around them, called the Light Interference Field (LIF), computed using the same 2m grid. We compute it by combining kernels centered on every mature individual using simple stamping, similarly to a convolution operation. The values of the kernels are scaled with a factor s , using the tree height h and the canopy height z (approximated by the height of the tallest tree in a 10m-wide cell): $s = h/z$.

In [SRSS12], the kernels are precomputed for different tree height and shapes into a database by integrating the shadows caused by the tree over all the sun positions in the year. We observed their shape is similar to 2D gaussian functions and opted to use a procedurally generated gaussian kernel. The resulting value $\ell \in [0, 1]$ is used to adjust the vigor v_i by multiplying the two, simulating the competition for resources.

2.2. Biological processes

The process of creating and spreading the seeds of the mature trees is simulated in our method. For each species, the seeds are not sim-

ulated individually, they are rather represented by probabilistic dispersion kernels. A probability field is computed by combining the kernels, placed around mature trees in a similar way to the LIF, with the difference that the aggregation is made by addition instead of multiplication. This probability is combined with the vigor v and the LIF ℓ to form the probability of a seed germinating and creating a new sapling. Each year, a russia roulette selection can be used to select which species (if any) will propagate and create a sapling, for every cell:

$$p = p_{seed} \times v_i \times \ell$$

The next, and most important process to simulate is the growth of the trees. The process is similar for mature trees and saplings. We use the same growth function that was used in ILAND for the smaller vegetation [SSR*12]. The growth increment g is calculated in several steps: first, a theoretical height h_{th} is estimated using a parameterized growth curve. Then, for the saplings, the potential growth g_{pot} (representing the amount of growth under ideal condition) is weighted with the vigor and LIF to obtain the actual growth g :

$$\begin{aligned} h_{th} &= h_{max} \left(1 - \left(1 - \left(\frac{h_t}{h_{max}}\right)^{\frac{1}{3}}\right) e^{kg}\right)^3 \\ g_{pot} &= h_{th} - h_t \quad g = g_{pot} \times v_i \times \ell \end{aligned}$$

The growth of mature trees is slightly more complex. First, the tree's own contribution is taken out of the LIF value ℓ . Additionally, a carbon pool C represents the reserves used to survive for short periods of time with no ressource intake (during droughts for example). The carbon pool is filled with the incoming ressources, and the excess is used to contribute to the actual growth increment g :

$$C_t = C_{t-1} + v_i \times LIF$$

$$g = g_{pot} \times \max(0, C_t - C_{max})$$

Finally, plant mortality is essential to ensure the natural selection of species fit to grow in their particular environment. The death probability μ of each individual is computed based on age and ressource strain in the same way as was done by [PGG*24]:

$$\begin{aligned} \mu_a &= 1 - (\hat{p}^{1/a_{max}}) \\ \mu_r &= \max(-C, 0) \\ \mu &= \min(\mu_a + \mu_r, 1) \end{aligned}$$

The age mortality μ_a makes sure that $\hat{p} = 1\%$ of individual reaches their species' maximum age. The ressource strain μ_r penalizes trees that have overdrawn their carbon pool (and thus have unsufficient ressources to sustain long-term growth).

3. Implementation on Graphics Hardware

A significant benefit of this method is that all the data is grid-based, and that all the interactions between individuals are based on a combination of kernels similar to a convolution.

The GPU implementation is straightforward: all the different data is stored in textures or texture arrays. Data relating to mature individuals is separated in two textures that store the height of the trees, and the species index that indicates if there is an individual in the cell, and which species it belongs to.

The underbrush is stored in a texture array: for each species, a texture indicates the height of the average sapling. Other textures are used to store intermediate results, such as the Light Interference Field and canopy height, or seed probability and vigor, stored in texture arrays.

Every distinct step or operation is done in a compute shader. For each iteration of the simulation, the different shaders are run in succession to calculate the intermediate textures, and to handle the growth processes for the saplings and the mature trees, as well as the seeding step.

4. Results

Our method makes simplifications that drastically reduce the amount of data and computations necessary for the simulation to run. Additionally, the interactions between individuals is based on the underlying grid, making their implementation on the GPU very efficient. As a result, the simulation runs much faster than the previous methods. As a comparison base, we have implemented both CPU (single threaded) and GPU versions. The acceleration factor is about 4 orders of magnitude using GPU parallel computing capabilities. Both versions are not fully optimized and could be made more efficient.

The simulations were executed on a desktop computer with an Intel Core i9-13900K processor (with 24 physical cores clocked at 3GHz), a Nvidia RTX 4070 GPU (with 12GB of dedicated memory), and 32GB of RAM.

We have two different implementations of the method. The first runs on the CPU, without any parallelization. The second uses OpenGL and compute shaders for entire simulation to accelerate it. For the tests, we used the same scene that was used in [PGG*24]: a 2km × 2km plot of land in the Pyrenees mountains in northern-eastern Spain. The underlying grid then is 1000 cells wide. The simulations are ran for 300 years:

Terrain Size	Mature Trees	CPU	GPU	Acceleration
2km × 2km	600k	21 min	87 ms	≈ ×14k
4km × 4km	2400k	85 min	88 ms	≈ ×57k

Table 1: Comparison of the time needed to simulate 300 years with the CPU and the GPU implementation

5. Conclusion

We have introduced a method that allows the simulation of vast ecosystems in less than a second. There is room for improvement of the method with a better simulation of the environment, a better growth functions or the simulation of additional biological processes like decay. Moreover, the efficiency allows for larger areas to be simulated, or the possibility to consider learning simulation parameters to obtain the desired results.



Figure 2: Largescale view of a $2\text{km} \times 2\text{km}$ terrain with 600k trees, simulated in 87 ms using our model



Figure 3: Closeup of the vegetation generated on a $2\text{km} \times 2\text{km}$ terrain, showing the high density in the valleys

223 *Methodologies, Tools and Applications*. IGI Global, 2011, pp. 418–465.
224 2

225 [DHL*98] DEUSSEN O., HANRAHAN P., LINTERMANN B., MĚCH R.,
226 PHARR M., PRUSINKIEWICZ P.: Realistic modeling and rendering of
227 plant ecosystems. In *Proceedings of the 25th Annual Conference on*
228 *Computer Graphics and Interactive Techniques* (New York, NY, USA,
229 1998), SIGGRAPH '98, Association for Computing Machinery, pp. 275–
230 286. 2

231 [dNFP18] DO NASCIMENTO B. T., FRANZIN F. P., POZZER C. T.: Gpu-
232 based real-time procedural distribution of vegetation on large-scale virtual
233 terrains. In *2018 17th Brazilian Symposium on Computer Games and Digital Entertainment (SBGames)* (2018), IEEE, pp. 157–15709. 1

235 [ENCC*21] ECORMIER-NOCCA P., CORDONNIER G., CARREZ P.,
236 MOIGNE A.-M., MEMARI P., BENES B., CANI M.-P.: Authoring con-
237 sistent landscapes with flora and fauna. *ACM Transactions on Graphics*
238 40, 4 (2021), 1–13. 1

239 [EVC*15] EMILIEU A., VIMONT U., CANI M.-P., POULIN P., BENES
240 B.: Worldbrush: interactive example-based synthesis of procedural virtual
241 worlds. *ACM Transactions on Graphics* 34, 4 (2015), 106–1. 2

242 [GLCC17] GAIN J., LONG H., CORDONNIER G., CANI M.-P.: Eco-
243 brush: Interactive control of visually consistent large-scale ecosystems.
244 *Computer Graphics Forum* 36, 2 (2017), 63–73. 2

245 [Ham01] HAMMES J.: Modeling of ecosystems as a data source for real-
246 time terrain rendering. In *Digital Earth Moving* (Berlin, Heidelberg,
247 2001), Westort C. Y., (Ed.), Springer Berlin Heidelberg, pp. 98–111. 1

248 [KGG*20] KAPP K., GAIN J., GUÉRIN E., GALIN E., PEYTAVIE A.: Data-driven authoring of large-scale ecosystems. *ACM Transactions on Graphics* 39, 6 (2020), 1–14. 2

251 [LP*02] LANE B., PRUSINKIEWICZ P., ET AL.: Generating spatial dis-
252 tributions for multilevel models of plant communities. In *Graphics inter-
253 face* (2002), vol. 2002, pp. 69–87. 2

254 [MHS*19] MAKOWSKI M., HÄDRICH T., SCHEFFCZYK J., MICHELS
255 D. L., PIRK S., PAŁUBICKI W.: Synthetic silviculture: multi-scale mod-
256 eling of plant ecosystems. *ACM Transactions on Graphics* 38, 4 (2019),
257 1–14. 2

258 [PB05] PRETZSCH H., BIBER P.: A re-evaluation of reineke's rule and
259 stand density index. *Forest science* 51, 4 (2005), 304–320. 2

260 [PGG*24] PEYTAVIE A., GAIN J., GUÉRIN E., ARGUDO O., GALIN
261 E.: Deadwood: Including disturbance and decay in the depiction of dig-
262 ital nature. *ACM Transactions on Graphics* 43, 2 (2024), 1–19. 2, 3

263 [Rei33] REINEKE L. H.: Perfecting a stand-density index for even-aged
264 forests. 2

265 [SRSS12] SEIDL R., RAMMER W., SCHELLER R. M., SPIES T. A.: An individual-based process model to simulate landscape-scale forest
266 ecosystem dynamics. *Ecological Modelling* 231 (2012), 87–100. 2

268 [SSR*12] SEIDL R., SPIES T. A., RAMMER W., STEEL E. A., PABST
269 R. J., OLSEN K.: Multi-scale drivers of spatial variation in old-growth
270 forest carbon density disentangled with lidar and an individual-based
271 landscape model. *Ecosystems* 15 (2012), 1321–1335. 2, 3

272 [WHDS13] WEIER M., HINKENJANN A., DEMME G., SLUSALLEK P.:
273 Generating and rendering large scale tiled plant populations. *Journal of
274 Virtual Reality and Broadcasting* 10, 1 (2013). 1

References

212 [AD05] ALSWEIS M., DEUSSEN O.: Modeling and visualization of sym-
213 metric and asymmetric plant competition. In *Paper / Eurographics Work-
214 shop on Natural Phenomena* (2005), pp. 83–88. 2

215 [AD06] ALSWEIS M., DEUSSEN O.: Wang-tiles for the simulation and
216 visualization of plant competition. In *Computer Graphics International
217 Conference* (2006), Springer, pp. 1–11. 1

218 [BEP03] BENES B., ESPINOSA E.: Modeling virtual ecosystems with the
219 proactive guidance of agents. In *Proceedings 11th IEEE International
220 Workshop on Program Comprehension* (2003), pp. 126–131. 2

221 [Ch'11] CH'NG E.: An artificial life-based vegetation modelling ap-
222 proach for biodiversity research. In *Green Technologies: Concepts,*