# Modelling, simulation, and visualization of forest ecosystems

***Abstract*— We present a framework for the simulation of forests, where a complex application system simulates a spontaneous afforestation process. Within this virtual environment, trees can be seen to grow over several centuries. The obtained simulation results are used to animate ecosystem development, where trees struggle for survival. The visualization of trees is speeded up so that the models of trees have progressively lower-details proportional to the distance from a certain point of view. The growth of individual trees is also animated, from the development of branch complexity to per-leaf precision to allow a very realistic perception of the emerging ecosystem.**

***Index Terms*— Artificial life, tree modelling, computer animation, natural phenomena, ecosystem simulation, biology.**

## I. Introduction

THE *modelling of trees* has a twenty year tradition in computer graphics. At the beginning, the accent was on modelling the geometrical structures of individual trees. Manual modelling of tree structure and its leaves is a tedious task, because each branch and leaf position, rotation, size and texture must be appointed. Therefore, procedural tree models are used instead and several techniques for procedural models are available today. Different procedural models are based on various types of branching structure construction [1]. These techniques differ in the level of detail [2], [3], [4], the flexibility, and pretentiousness of modelling [5], [6], space [7], [8], and time complexity [7] in addition to the animation ability and representation of the built 3D model. The majority of these models try to determine some visible properties of the final 3D model, such as the rotation of branches around their central axes. These properties are usually biologically inspired by *phyllotaxis*, i.e. the main influence on the tree's architecture [9].

Following increases in computer power, interest in the research of individual tree models has grown up to the research of algorithms for modelling entire ecosystems [10], where the simulation of afforestation had begun to play the key role [11], [12]. By passing a limited number of parameters to the model, many emerging trees grow over the landscape according to the modelled natural laws. The spontaneous development of an ecosystem is achieved using such a model, and simulated for each step, lasting for one year. Simulation results are presented to the user using computer graphics by rendering the 3D ecosystem scene into an image.

The first to treat *landscape visualization* with forests and shrubs were Weber and Penn [6]. Their trees were represented by a procedural geometrical model and rendered with ray-tracing. As this model featured level of detail adaption, they were able to visualize a few thousand trees on a landscape. Trees on this landscape were manually placed and did not reflect any biological laws of spontaneous afforestation, this being one of considerable problems of their approach.

Chiba et al. [13] represented trees using volumetric textures. Image was compiled by ray marching through voxels (3D image elements) and integrating their density and colour. Trees were placed on landscapes using Poisson disc sampling [14], so that a new tree was successfully added only if its randomly selected location was not too close to the location of some other tree. The simulation was simple and fast, but their rendering process took several hours and the trees were of low detail. Using such an approach they created images with up to twenty thousand trees on a landscape.

Deussen et al. [15] built the first ecosystem simulator used in computer graphics to follow natural laws. Their highly detailed tree models were created using L-systems [16] and rendered using *ray casting* or *ray tracing*. With so many ray-traced objects on scene, they came up with the conclusion that a compact procedural model for geometry generation and the *instancing* of created geometrical tree models must be used for objects to fit in the main memory. Visually credible distribution of trees was achieved using two different approaches, one is from global to local, the other is from local to global [17]. The first approach was semi-automatic, the second was automatic, a pure simulation. This simulation included some natural laws of interaction among the trees themselves and the environment. The first such law was a tendency for the same species of trees to grow in clusters, naming such effect as *clustering* [18], *clumping* [17] and *under-dispersion* [19]. The second behaviour principle considered was competition among individual trees in an area of *ecologic neighbourhood* [20], where one tree can influence the growth of others. The involved trees interact by the intersection of two ecological neighbourhood areas. One of them becomes dominated and when a tree is being dominated, its growth is slower or it may even die. This results generally in principal a phenomena of tree distribution known as *self-thinning*, meaning that in the beginning, trees grow without interference, but as density increases, the dominated trees start dying off. They defined the domination of a tree by comparing its power to the powers of competing trees within its range. If a tree had inferior power than a neighbour, it would become dominated. The power depended on the age of the tree, and simplified living conditions at its place of growth.

The existing simplified living condition models in computer graphics only consider the soil moisture. It is true that this is usually a significant environmental impact, but additional, sometimes even more important ones, can be found. This is why we decided to build a new ecological model to consider more parameters. We prepared new models for tree growth, their distribution across landscapes, and the calculation of additional living conditions. We have designed all the mentioned models and implemented them in C++ programming language. For simulation, we used real terrain elevation data to compute living conditions.

The paper is divided in six sections. The next section describes the simulation of an ecological system. The third section presents models for the calculation of living conditions within an ecosystem, where elevation data is used to calculate living conditions such as slope, soil moisture, and degrees of wind and sun. The fourth section explains the modelling of trees and the visualization of a simulated ecosystem. Discussion takes place in the fifth section. The paper concludes with a summary, and guidelines for future work.

## II. Simulation of an ecological system

We begin the simulation with an empty terrain on which, every year, a different number of trees for each species are added ($\Delta t = 1$ year). We can simulate the lives of several hundred thousand trees over several hundred years. As we mentioned, our simulation determines the distribution of individual trees across a terrain, based on the living conditions within that terrain. These conditions are *height above sea level*, *slope*, *humidity*, and the amount of *sun* and *wind*.

In every simulation step, we execute three sub-steps: the creation of new trees, their growth, and the elimination of dead trees. These three main processes are implemented as software agents, and act stochastically.

The first sub-step of every simulation step, the creation of new trees, is done for each tree species using different parameters, because each tree species has a specific method of replication. Firstly, $\Delta N_{g,s}$ trees for each species to be added is computed in each year. The amount varies according to the number of seeds from this species cumulated over previous year, and the seed germination factor. When we add new trees, we consider the under-dispersion principle by increasing the probability of chosen locations near to existing trees.

The second simulation sub-step, growth regarding trees, depends upon their power. Using $p_{s,p}$ ($p_{s,p} \in [0,1]$), we can denote the power of $p$-th tree from the tree species $s$:

$$p_{s,p} = h_{s,p} p_{persist;s} v_{s,p}, \tag{1}$$

where $h_{s,p} \in [0,1]$ denotes the height of the tree, $p_{persist;s} \in [0,1]$ the persistent clinging to life[1] and $v_{s,p} \in [0,1]$ the suitability of living conditions for a tree with indexes $s$ and $p$.

The height of a tree depends upon its age and factors that have affected its growth until this age. The first factor to consider is *growth obstruction*, caused through domination by more powerful trees within neighbourhood. The second factor is *non-uniform growth intensity* during its lifetime, because trees grow faster in youth than in adolescence. For the latter factor, we defined a polyline to approximate an ideal continuous growth curve for the height of a tree depending on its age. Fig. 1 shows the height of a tree over years without growth obstruction using the ideal continuous curve and its approximated polyline. For each tree species, these polyline control points are defined using *average maturity year of species* ($t_{m;s}$), *average year when species is fully grown* ($t_{a;s}$), *average height of a matured species* ($h_{m;s}$), and *maximum potential age of species* ($t_{f;s}$).

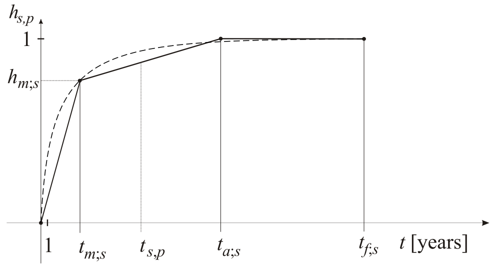


Fig. 1. Height of a species through lifetime without growth obstruction.

The persistent clinging to life of a species, $p_{persist;s}$, is used to distinguish between pioneering species such as shrubs and birches and others such as beeches, spruces, and maples. It is characteristic for the former to retreat if the latter has suitable living conditions in the same area.

The suitability of living conditions ($v_{s,p}$) in our simulation is calculated from the suitability of the considered living condition factors:

$$v_{s,p} = k_{\overline{y};s,p} m_{s,p} w_{s,p} l_{s,p} s_{s,p}, \tag{2}$$

where $k_{\overline{y};s,p} \in [0,1]$ denotes the suitability of height above sea level, $m_{s,p} \in [0,1]$ soil moisture, $l_{s,p} \in [0,1]$ sun, $w_{s,p} \in [0,1]$ wind, and $s_{s,p} \in [0,1]$ slope at growth location. We define the latter by considering *slope harmfulness* ($f_{s;s}$) for each species:

$$s_{s,p} = 1 - (f_{s;s} s_{i,k})^2, \tag{3}$$

where indexes $i$ and $k$ select the patch at the growth location, determining the slope. The suitability of the remaining four living condition factors depends on any deviation from the *optimal living conditions* of a species, denoted as $\overline{y}_{o;s}$, $m_{o;s}$, $l_{o;s}$ and $w_{o;s}$ in the following equations:

$$k_{\overline{y};s,p} = \begin{cases} 1, & |\overline{y}_{i,k} - \overline{y}_{o;s}| \leq \Delta\overline{y}_{o;s} \\ 1 - \max\left\{1, |\overline{y}_{i,k} - \overline{y}_{o;s}| f_{\overline{y};s}\right\}, & \text{otherwise} \end{cases}, \tag{4}$$

$$m_{s,p} = \begin{cases} 1, & |m_{i,k} - m_{o;s}| \leq \Delta m_{o;s} \\ 1 - \max\left\{1, |m_{i,k} - m_{o;s}| f_{m;s}\right\}, & \text{otherwise} \end{cases}, \tag{5}$$

$$l_{s,p} = \begin{cases} 1, & |l_{i,k} - l_{o;s}| \leq \Delta l_{o;s} \\ 1 - \max\left\{1, |l_{i,k} - l_{o;s}| f_{l;s}\right\}, & \text{otherwise} \end{cases}, \tag{6}$$

$$w_{s,p} = \begin{cases} 1, & |w_{i,k} - w_{o;s} \leq \Delta w_{o;s} \\ 1 - \max\left\{1, |w_{i,k} - w_{o;s}| f_{w;s}\right\}, & \text{otherwise} \end{cases}. \tag{7}$$

Here, *tolerated deviations* $\Delta\overline{y}_{o;s}$, $\Delta m_{o;s}$, $\Delta l_{o;s}$ and $\Delta w_{o;s}$ are auxiliary parameters for defining the classes of suitable living conditions for species. If a certain living condition is out of the tolerated deviation bound, its harmfulness is considered by diminishing a living condition with it, and a *harmfulness factor*. Latter factors are denoted as $f_{\overline{y};s}$, $f_{m;s}$, $f_{w;s}$, and $f_{l;s}$ for each living condition.

Classes of suitable living conditions can be obtained from some of the existing trees living condition taxonomies. One such was prepared by Ellenberg [21] for Central Europe and

[1] For easier notation, descriptive indexes for main symbol are separated from variable indexes with a colon in this paper.

extended to other areas of the world [22], [23]. Ellenberg divided living conditions within ten classes on a linear scale and classified suitable living conditions for individual species in these classes. Statistically numerated data can be read from a digital library [24] and used in our simulation.

As mentioned, the final, third simulation sub-step is the elimination of dead trees. We distinguish between three causes of death: *natural*, *intolerableness of living conditions*, and *suffocation*. We define natural cause as any type of death caused by some other cause than the remaining two causes. The year, when a tree is going to die because of this cause, is pre-selected at its creation as a uniformly distributed random number between the mentioned years $t_{a;s}$ and $t_{f;s}$ for its species. During each simulation step, we compare this year and the current age of the tree, and eliminate the tree if the years match. As seen, only one comparison must be done for each tree in a single simulation step, making this a modest $O(n)$ algorithm, where $n$ is the number of trees in the ecosystem and the operations counted, are comparisons.

We model the intolerance of living conditions, together with the growth of a tree, so that if living conditions are not satisfied, a seed does not germinate and is forgotten. Once again, this is a modest $O(n)$ algorithm.

Suffocation is caused if a tree is dominated by other neighbour trees. It is caused because of interaction between trees and is much harder to calculate. Generally, certain trees can be dominated by any of the remaining trees. For each tree we would eventually have to check all the remaining trees, resulting in an undesirable $O(n^2)$ algorithm. The optimization approach we took is a localization of the algorithm to lower the order of the improved algorithm.

To find trees nearby, we conduct a reference list on each patch to search for nearby trees much faster. The list contains pointers to trees, which grow in this part of terrain, and is constructed using uniform plane partitioning. For every search, all trees in few nearby patches are tested to determine any dominance of a certain tree. We conducted an experiment, to present improvement in a real case. It was shown that with 600,000 trees in the 299th simulation year, on average only 540 instead of 600,000 tests were made per a tree. For a case with ideally uniformly distributed locations of trees, the complexity order of this algorithm is $O(n \log n)$.

## III. Models for the computation of living conditions in a terrain

The mesh for the terrain is read from a digital terrain model (DEM) [25] and defined using vertices:

$$\mathbf{p}_{i,k} = [\,x_i \quad y_{i,k} \quad z_k\,]^T, \qquad i,k \in [0,99], \tag{8}$$

being sampled for every e.g. 25 m along $x$ and $z$. $y_{i,k}$ denotes elevation of the $(i,k)$-th sample. From vertices, quad-like patches are formed, e.g. 10,000 patches for a sample terrain. From the data of these patches, living conditions are calculated for every patch. Height above sea level $y$ is normalized for later use with other models:

$$\overline{y} = \frac{y - y_{min}}{y_{max} - y_{min}}, \tag{9}$$

where $y_{min}$ is the minimum height above sea level of the terrain, and $y_{max}$ is its maximum. The surface normal of a quad (patch) is denoted by $\mathbf{n}_{i,k}$ and calculated as:

$$\begin{aligned} \mathbf{n}_{i,k} &= (\mathbf{p}_{i+1,k} - \mathbf{p}_{i,k}) \times (\mathbf{p}_{i,k+1} - \mathbf{p}_{i,k}) \\ &= [n_{x;i,k} \quad n_{y;i,k} \quad n_{z;i,k}]^T \end{aligned}, \tag{10}$$

where the first vector spans along neighbouring vertices in the first axis and the second one in the next axis. With $\overline{\mathbf{n}}_{i,k}$, the normalized normal is denoted. Slope is calculated by using $y$ component of $\overline{\mathbf{n}}_{i,k}$:

$$s_{i,k} = 1 - \overline{n}_{y;i,k}. \tag{11}$$

A flat (horizontal) terrain has a slope of zero and a steep (vertical) terrain has a slope of 1. Using advanced algorithms for simulating natural phenomena impact, we determine terrain soil moisture, and average wind and sun distribution for the summer's half-year. These algorithms we created ourselves, but they have been compared with existing referential literature. Fig. 2 displays the distributions of these living conditions across terrain, obtained using our models. Due to this paper's page limit constraint, we are only able to give a shallow explanation of these models here.

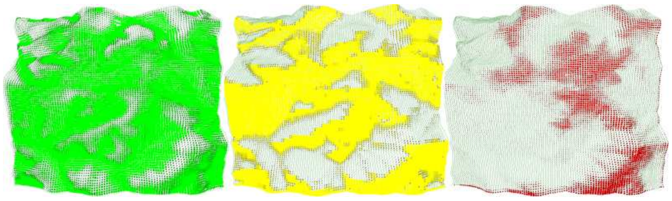


Fig. 2. Intensities of soil moisture, wind, and sun for real terrain data obtained using our models on real terrain data.

### A. Soil moisture

Our liquid flow model is used for soil moisture calculation. Rain flow is its main affecting factor, depending on slope, soil absorption, and terrain topology. Another factor is still water accumulation, such as rivers and lakes. The latter is included by initializing the consisting areas to high moisture levels. Now, let us discuss the main impact factor. We assume that rain fall distribution is uniform over the whole terrain and, therefore, each patch receives the same quantity of water. Because soil can not usually absorb all the taken rain, the rain-water flows from the original rainfall area to other nearby areas, or flows in from other original rainfall areas. All this results in nonuniform soil moisture distribution.

As explained, rain-water flows to or from neighbouring patches with a magnitude depending on their slope, height variation with neighbouring patches, and the soil absorption rate. Locally this means, the moisture received for certain patches equals the immediately absorbed rain and the absorbed part of the water flowing to this patch from higher patches. Soil absorption rate depends on soil structure and rain type (light rain, rain shower). In our simulation, the rain absorption rate is approximated as a portion of absorbed rain compared to all the rainfall, but a absorption rate map can be loaded too.

The obtained moisture distribution is further processed by convolving it with a low-pass digital filter (an efficient moving average FIR filter is used). The filter is executed to soften and widen the moisture accumulation peaks.

### *B. Wind*

Our wind calculation model is based on wind-shelter position determination. To calculate wind shelter areas, our algorithm considers two main principles: wind flow is, in many ways, similar to liquid flow, moving horizontally but because of minor differences in temperature it changes its mass and is, therefore, subject to vertical movement too [26]. Both principles result in wind shelter areas being created behind hills, because wind flows forward horizontally, but does not move down quickly enough. Using wind drop speed and forward motion speed, a total angle of wind occlusion can be defined.

### *C. Sun*

Our sun distribution model is based on astronomical models for the Earth's rotation around the Sun. As the Earth rotates around the Sun and by itself, different areas of the terrain on the Earth receive different amounts of the Sun's radiation. What we do is sample the key Sun positions and cumulate the average Sun radiation, similar, but not quite the same, as in [27]. The latter is mainly dependant on the incidence angles of Sun's rays. Another important factor is hill shadows, preventing the Sun's rays breaking through, as in e.g. some valleys.

As mentioned, because of the emergence of shadows, some rays can not penetrate into some areas. To determine, if a certain patch is in shadow, a similar algorithm was used as in the determination of wind shelter positions. In this case, traversal is done from a current patch towards the Sun's location. The height of the patches is used during traversal of patches towards $\mathbf{p}_n$. If the central point $\mathbf{c}_{i,k;n}$ of some $n$-th patch forms a smaller incidence angle with the central point $\mathbf{c}_{i,k}$ of the starting patch, this patch is preventing the Sun's rays from radiating onto the starting patch. As soon as some occluding patch is found, the search is finished for the current sample. The mentioned incidence angle is defined in the same as for wind calculation. If a certain patch is in shadow for some Sun's sampled position, that sample's radiation is diminished according to the ambient light coefficient parameter controllable by the user of the model.

## IV. Ecosystem landscape visualization

In our application, we can analyse simulation results quantitatively using 2D graphs, e.g. population size of each species, average age of species or the clustering of species during simulation. Fig. 3 shows graphs for the first-mentioned case, where simulation lasted for several hundred simulation years. Initially, there are enough resources for all species, so their population sizes grow exponentially. Shrubs are a pioneering species so they propagate at a fastest pace. After 200 years, the terrain has long been saturated and only the number of willows changes slightly. Species population sizes are now fairly stabilized, with a total population size of over 600,000 trees in the ecosystem. To test simulation stability, we enforce an ecological catastrophe in the year 300 by killing all trees and not changing any simulation random generators. The ecosystem recovers with similar population sizes for each species as before the catastrophe. In year 600, we trigger another catastrophe by killing only $99,9\%$ of trees and eliminating all shrubs and willows during this progress. But the ecosystem recovers once again. As can be seen, shrubs get eliminated again, but the willows also. They become extinct because other competitive species populate former growth areas of the willows and they do not have enough seeds to recover.

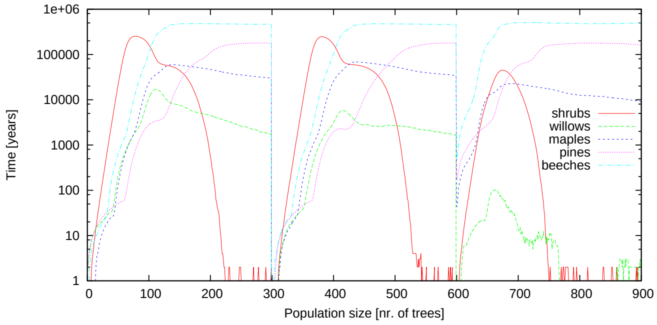


Fig. 3. Exponential propagation of trees, stabilization and recovery after catastrophes in years 300 and 600.

Computer graphics is indispensable for analysis of the visual credibility of integrated models, allowing us to synthesize realistic images of simulated ecosystems [5], [28], [13], [10]. We visualize each simulated year by drawing a terrain surface and adding individually personalized geometrical models of trees at their growth locations.

A photo-realistic image is rendered in order to visualize the distribution of trees across a terrain. At the tree locations for the image, our procedural models are used to instantiate the adopted and personalized geometrical models, for more realistic visualization. Procedural models are obtained using our interactive modeller part in order to create these geometrical models. Using the modeller, related data for parametrizable procedural model is acquired by the design of graphs and the use of interactive dialogues, which immediately affect the displayed image of a tree. This certainly eases modelling and allows rapid model creation. As the user parametrizes the procedural model with a limited set of intuitive parameters, the underlying procedural model calculates and instantiates a geometrical model for a tree. The 3D position, size, orientation, and texture are calculated in order to create a geometrical model, for each of up to over thousand branch segments and several ten thousand leaves. A procedural model can also be designed within an ecosystem with other trees, for easier perception. The ergonomics of the modeller enables us to obtain a new procedural model within twenty minutes or less. In addition, the procedural model enables a fairly controllable, but flexible, animation. Each leaf and each branch segment can be animated in real time by changing some

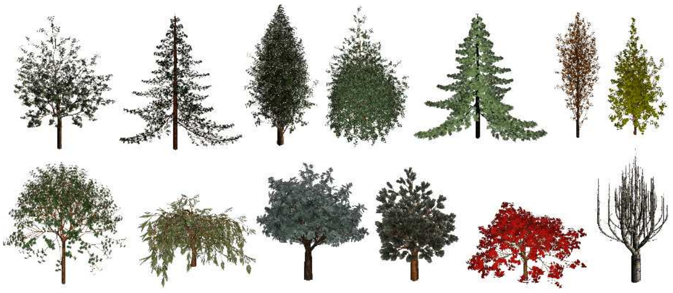


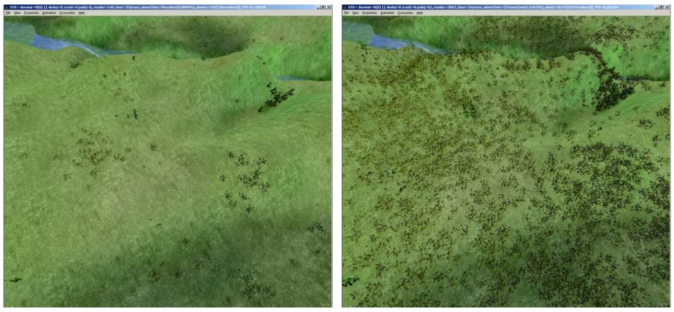
Fig. 4. Trees obtained using our modeller.

adjustable procedural model parameters, e.g. age of a tree. Those animation effects already included in our modeller are the growth of a tree and the sway of the tree in the wind. Fig. 4 displays some trees, which were visualized using our procedural models. As seen, coniferous and foliage trees can be obtained, which have very different branching structures and leaf distribution.

For the visualization of complete ecosystems, we upgraded our tree modeller with three major improvements and integrated it into the ecosystem simulator, where the tree-visualization part was used to visualize a full range of different trees in the terrain. The first improvement is the use of culling for the removal of objects from the scene, so that the geometry is never computed for culled trees. The second improvement is the level of detail reduction for distant trees. Dependent on reduction degree, less smaller branches are drawn. The appearance of simplified trees differs little, but visualization process speedup is by up to several hundred times. Using these two improvements, we can already afford real-time animation of several thousand trees. The third improvement pushes visualization performance even further. According to the above reduction degree, differently-complexed geometrical objects are chosen for branch segment and leaf representation. Fully detailed branch segment and leaf representations are drawn with a cone and a quad. A simplified branch segment and leaf are drawn using a line and a vertex.

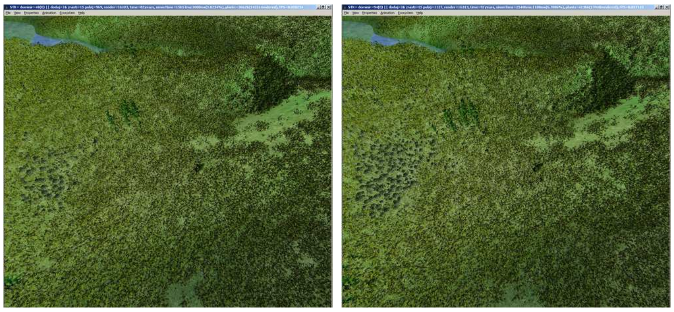
### A. Synthesized images

We present some rendered images, which are synthesized from one of the simulation runs. Four tree species are included: beech, maple, pine, and shrubs. Simulation begins with an empty terrain with no trees. The trees populate the landscape, as a natural consequence if people stop mowing grassland (Fig. 5a). The first few years mostly include these, because shrubs can grow and replicate quickly (Fig. 5b). As they are weak in their fight for space against new dominating species, they retreat and the terrain is populated by larger foliage plants (Fig. 6a). Later, spruces start to spread from the right side (Fig. 6b). After approximately 180 simulated years the terrain is saturated with all species used in simulation. Each of them acquires a suitable area for living conditions, where it can dominate other species. Obtained clustering of species is seen, but some trees from another species are still contained within the environed areas making the image more authentic.



a b

Fig. 5. a) Beginning of simulation, b) People stop mowing grassland, shrubs happen to appear.



a b

Fig. 6. a) Simulation after 60 years: terrain is saturated, bigger tree species begin to prosper and replace shrubs, b) Results after 80 years: pines start to penetrate massively, whereas beech and maple are already distributed finitely across the terrain according to living conditions, and their interaction.

### B. Architectural configuration

Architectural configuration of the built application is shown in Fig. 7. EcoMod application is composed of both tree modelling component and ecosystem modelling component. Each component is finely granulated and they interact with each other at top level by the latter using constructed procedural models of the first to create geometrical models for growing trees. As seen from the figure, user parameter input is saved in the databases of procedural model and ecological model parameters. The links of ecological models with their visual representations using procedural geometrical models, are saved in the former database.

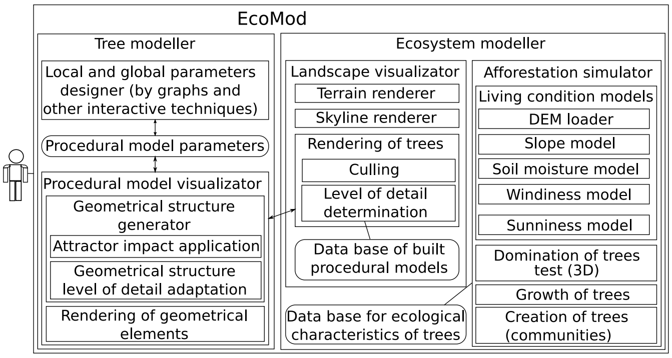

EcoMod
Tree modeller
Local and global parameters designer (by graphs and other interactive techniques)
Procedural model parameters
Procedural model visualizator
Geometrical structure generator
Attractor impact application
Geometrical structure level of detail adaptation
Rendering of geometrical elements
Ecosystem modeller
Landscape visualizator
Terrain renderer
Skyline renderer
Rendering of trees
Culling
Level of detail determination
Data base of built procedural models
Data base for ecological characteristics of trees
Afforestation simulator
Living condition models
DEM loader
Slope model
Soil moisture model
Windiness model
Sunniness model
Domination of trees test (3D)
Growth of trees
Creation of trees (communities)

Fig. 7. Architectural configuration of the application components and included algorithms.

## V. Discussion

A new research topic has arisen for the *modelling of ecosystems*, which combines computer graphics and artificial life in a new way. A cheaper *computer animation* of natural phenomena is achieved by combining both scientific topics. As seen again, computer animation is still and once again the leading economical driver of research into artificial life.

The distribution of trees just *emerges* within the simulation, as this emergence is a principal phenomena of *artificial life*. Simulation is driven by programmable agents which mediate within the development. They are the keepers of rules within the ecosystem, such as the growth of individual trees, competition between neighbouring trees, the dying of trees, the manner of reproduction, and the disposal of living conditions.

Let us stress, what are main contributions of our algorithms in relation to existing solutions:

- efficient integration of a procedural tree-model modeller tool (subject of human-computer interaction), ecosystem simulator (subject of artificial life) and forestry landscape visualization (subject of computer animation) in a single application,
- consideration of several living conditions besides soil moisture, such as height above sea level, wind, and sun, for visualization of ecosystems,
- immediate visualization of a simulated ecosystem during simulation,
- ability to shape a new procedural tree model as a real landscape at a selected DEM location, and preview the built model within it,
- construction of algorithms for the calculation of living conditions for visualization of ecosystem simulation, from real DEM terrain data,
- efficient generalization of an ecological neighbourhood area collision detection of domination from 2D into 3D space,
- use of living condition factors from Ellenberg taxonomy, in forest development simulation for visualization of ecosystems.

## VI. Conclusion

This paper describes an elaborated simulator for landscape afforestation when determining tree distribution. It considers several environmental properties and combines computer graphics with artificial life. The application is used in computer animation for the synthesis and analysis of natural environments. A flexible and adaptable procedural 3D model is used to visualize trees. Behaviour of the presented tree distribution model was also confirmed. The number of trees in the ecosystem increases exponentially, trees do grow in communities, and permanently emerging pattens are biologically inspired and natural looking.

The interdisciplinary spirit of this research enables its findings in algorithms to be used in several different scientific fields, besides computer science, such as biology, ecology, forestry, and pedology. Much more future work can still be done on this topic in all research areas. Simulation could be additionally optimized and run parallel to several computers.

## References

[1] P. Prusinkiewicz and A. Lindenmayer, *The Algorithmic Beauty of Plants*. Springer-Verlag, 1990.

[2] M. Aono and T. Kunii, "Botanical tree image generation," *IEEE Computer Graphics and Applications*, vol. 4, no. 5, pp. 10–34, may 1984.

[3] J. Bloomenthal, "Modeling the mighty maple," in *SIGGRAPH '85 Conference Proceedings (San Francisco, CA, 22–26 July 1985)*, B. A. Barsky, Ed., 1985, pp. 305–311.

[4] W. Reeves, "Approximate and probabilistic algorithms for shading and rendering structured particle systems," *Proceedings of SIGGRAPH'85*, pp. 313–322, 1985.

[5] M. Holton, "Strands, gravity, and botanical tree imagery," *Comput. Graph. Forum*, vol. 13, no. 1, pp. 57–67, 1994.

[6] J. Weber and J. Penn, "Creation and rendering of realistic trees," *Proceedings of SIGGRAPH '95*, pp. 119–128, 1995.

[7] P. E. Oppenheimer, "Real time design and animation of fractal plants and trees," *Computer Graphics*, vol. 20, no. 4, pp. 55–64, 1986.

[8] D. Strnad and N. Guid, "Modeling trees with hypertextures," *Comput. Graph. Forum*, vol. 23, no. 2, pp. 173–188, 2004.

[9] D. Reinhardt, E.-R. Pesce, P. Stieger, T. Mandel, K. Baltensperger, M. Bennett, J. Traas, J. Friml, and C. Kuhlemeier, "Regulation of phyllotaxis by polar auxin transport," *Nature*, no. 426, pp. 255–260, 2003.

[10] O. Deussen, C. Colditz, M. Stamminger, and G. Drettakis, "Interactive visualization of complex plant ecosystems," in *Proceedings of the conference on Visualization '02*, 2002, pp. 219 – 226.

[11] A. Guisan and N. E. Zimmermann, "Predictive habitat distribution models in ecology," *Ecological Modelling*, no. 135, pp. 147–186, 2000.

[12] J. Keppens and Q. Shen, "Granularity and disaggregation in compositional modelling with applications to ecological systems," *Applied Intelligence*, vol. 25, no. 3, pp. 269–292, 2006.

[13] N. Chiba, K. Muraoka, A. Doi, and J. Hosokawa, "Rendering of forest scenery using 3D textures," *Journal of Visualization and Computer Animation*, vol. 8, no. 4, pp. 191–199, 1997.

[14] R. L. Cook, "Stochastic sampling in computer graphics," *ACM Transactions on Graphics*, vol. 5, no. 1, pp. 51–72, Jan 1986.

[15] O. Deussen, P. Hanrahan, B. Lintermann, R. Mech, M. Pharr, and P. Prusinkiewicz, "Realistic modeling and rendering of plant ecosystems," in *Proceedings of SIGGRAPH '98*, 1998, pp. 275–286. [Online]. Available: http://doi.acm.org/10.1145/280814.280898

[16] P. Prusinkiewicz, M. Hammel, J. Hanan, and R. Měch, "L-systems: From the theory to visual models of plants," in *Plants to Ecosystems*, ser. Advances in Computational Life Sciences, M. T. Michalewicz, Ed. P.O. Box 1139, Collingwood 3066, Australia: CSIRO Publishing, Feb 1997, vol. 1, ch. 1, pp. 1–27.

[17] B. Lane and P. Prusinkiewicz, "Generating spatial distributions for multilevel models of plant communities," in *Proceedings of the Graphics Interface 2002 (GI-02)*. Mississauga, Ontario, Canada: Canadian Information Processing Society, may 27–29 2002, pp. 69–80.

[18] B. Benes and J. M. S. Guerrero, "Clustering in virtual plant ecosystems," in *WSCG Proceedings*, February 2004.

[19] B. Hopkins, "A new method for determining the type of distribution of plant individuals," *Annals of Botany*, vol. XVIII, p. 213–226, 1954.

[20] M. R. T. Dale, *Spatial Pattern Analysis in Plant Ecology*, ser. Cambridge Studies in Ecology. Cambridge University Press, Cambridge, UK, 1999.

[21] H. Ellenberg, H. Weber, R. Dull, V. Wirth, W. Werner, and D. Paulissen, "Zeigerwerte von pflanzen in mitteleuropa," *Scripta Geobotanica*, no. 18, pp. 1–248, 1991.

[22] M. O. Hill, D. B. Roy, J. O. Mountford, and R. G. H. Bunce, "Extending ellenberg's indicator values to a new area: algorithmic approach," *Journal of Applied Ecology*, no. 37, pp. 3–15, 2000.

[23] J. E. Lawesson, A. M. Fosaa, and E. Olsen, "Calibration of ellenberg indicator values for the faroe islands," *Applied Vegetation Science*, no. 6, pp. 53–62, 2003.

[24] L. Tichý, "Juice, software for vegetation classification," *Journal of Vegetation Science*, no. 13, pp. 451–453, 2002.

[25] Z. Li, Q. Zhu, and C. Gold, *Digital Terrain Modeling*. San Francisco: CRC Press, 2005.

[26] M. P. van Lieshout, "Time dependent energy calculations," in *Wind flow and Trees*. British Wind Energy Association, 2004.

[27] K. Zaksek, T. Podobnikar, and K. Ostir, "Solar radiation modelling," *Computers & Geosciences*, vol. 31, no. 2, pp. 233–240, 2005.

[28] R. Mech and P. Prusinkiewicz, "Visual models of plants interacting with their environment," in *Proceedings of SIGGRAPH '96*, 1996, pp. 397–410. [Online]. Available: http://doi.acm.org/10.1145/237170.237279