Simulation and Visualisation of Functional Landscapes: Effects of the Water Resource Competition between Plants

Vincent Le Chevalier, Marc Jaeger, Xing Mei, Paul-Henry Cournède

Month 200X, Vol.21, No.X, pp.XX–XX *J. Comput. Sci. & Technol.*

Simulation and Visualisation of Functional Landscapes: Effects of the Water Resource Competition between Plants

Le Chevalier Vincent[1], Jaeger Marc[2], Mei Xing[3], and Cournède Paul-Henry[1]

[1]*Ecole Centrale Paris, Laboratory MAS, France*

[2]*INRIA-Futurs, DigiPlante Team, CIRAD AMAP, Montpellier France*

[3]*LIAMA, CASIA, China*

E-mail: {vincent.le-chevalier, paul-henry.cournede}@ecp.fr; marcjaeg@gmail.com; xmei@nlpr.ia.ac.cn

Received MONTH DATE, YEAR.

Abstract Vegetation ecosystem simulation and visualisation are challenging topics involving multidisciplinary aspects. In this paper, we present a new generic frame for the simulation of natural phenomena through manageable and interacting models. It focuses on the functional growth of large vegetal ecosystems, showing coherence for scales ranging from the individual plant to communities and with a particular attention to the effects of water resource competition between plants.

The proposed approach is based on a model of plant growth in interaction with the environmental conditions. These are deduced from the climatic data (light, temperature, rainfall) and a model of soil hydrological budget. A set of layers is used to store the water resources and to build the interfaces between the environmental data and landscape components: temperature, rain, light, altitude, lakes, plant positions, biomass, cycles, etc. At the plant level, the simulation is performed for each individual by a structural-functional growth model, interacting with the plant's environment. Temperature is spatialised, changing according to altitude, and thus locally controls plant growth speed. The competition for water is based on a soil hydrological model taking into account rainfalls, water runoff, absorption, diffusion, percolation in soil. So far, the incoming light radiation is not studied in detail and is supposed constant. However, competition for light between plants is directly taken into account in the plant growth model. In our implementation, we propose a simple architecture for such a simulator and a simulation scheme to synchronise the water resource updating (on a temporal basis) and the plant growth cycles (determined by the sum of daily temperatures). The visualisation techniques are based on sets of layers, allowing both morphological and functional landscape views and providing interesting tools for ecosystem management. The implementation of the proposed frame leads to encouraging results that are presented and illustrate

* This work is supported by the National Natural Science Foundation of China under Grant No. 60473110 and by LIAMA-GREENLAB project

simple academic cases.

Keywords landscape visualisation, plant growth models, natural phenomena simulation, water cycle models

1 Introduction

Realistic simulation of ecosystems is a challenging topic, involving bio-physical, ecological, social, economical and human aspects. We focus here on vegetation growth in interaction with resources, mainly temperature and water, over a delimited domain. The target applications are agronomy, forestry and landscape planning, at mid range space and time scales, *i.e.* crops, stands or small landscapes, evolving over a period of several months to several years. Nowadays, process based vegetation models (PBM) and functional structural models (FSPM) [8] become mature. Some of them are currently able to model plant development and production, under constraints from a variable resource environment, especially in terms of light, water and temperature [6]. Nevertheless, in such approaches, the interaction between plant models and resources is limited. Another point is that such models cannot usually be easily extended at crop, plantation and higher spatial levels (landscape). Significant results were gained in the case of homogeneous crops, but competition for resources and heterogeneity in terms of space and time make such extensions not obvious at all.

The proposed work is to define some simple bases to allow plant and environment interactions, both ways. In this paper, we mainly focus on conservative water resources, to be shared among the components of a crop field, plantation or landscape. Such a study is motivated by the fact that the availability and supply of water resources are fundamental issues for human activities. All the elements of the water cycle have to be modeled, not only plants but also soil and climate.

The next section briefly recalls some interesting pioneering works on this topic, while section three introduces the principles of the proposed landscape simulator. Section four details some specific models and their interaction with resources. Section five presents the techniques used to visualise and analyse simulation results. Some study cases are then illustrated before the conclusion.

2 Previous works

Landscape modelling is a difficult subject, mainly because of the complex interactions at various time and space scales. Historically, the first goal was to reach a satisfying visual simulation of landscapes, and thus contributions came from computer graphics specialists. A pioneering study in this domain is the work by J. Hammes [13] introducing for the first time the word *ecosystem* in computer graphics, proposing a multilevel texture-based approach to synthesise such systems.

Relief definition was extensively studied [10], including simulation of long term effects such as erosion [18, 4, 20]. This research area is still very active nowadays, benefiting from modern GPU capabilities [21]. Such capabilities offer a high potential for the local refinement of relief, including vegetation-like features [7]. Various vegetation generation and evolving environmental approaches were also tested, voxel based [11], with L-Systems [9] or even recently with spatial competition [1]. The shortcomings of these approaches are mainly the lack of retro-action between elements of the landscape, as well as unrealistic dynamics. Physical accuracy is not an objective of these studies since only the visual aspect is desired.

Landscape sub-models, on the other hand, are getting more and more complex and accurate. Plant models, soil models, hydrological models [17], are all constantly improving. However, they are rarely integrated together to allow a coherent simulation of the whole landscape.

3 Architecture of functional landscapes

The idea behind *functional landscapes* is to simulate the multiple interactions taking place within ecosystems at the landscape level, with tools to visually investigate these interactions and the resulting quantitative variability. The notion was first introduced in [15] and has been the subject of further reflexions described in this work. However, such definition is to be considered in a very restrictive way. The term 'functional' did not (in [15]) include any bio-physical, social, nor economical evolution except vegetation development and its interaction with the water cycle simulation.

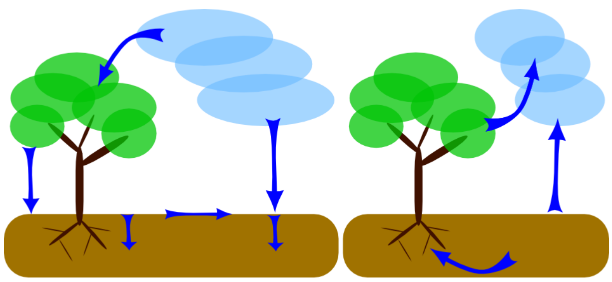

Figure 1: The water cycle. On the left, the descending part of the cycle: water falls on the soil and vegetation, and is finally absorbed into the soil. On the right, the ascending part: water is absorbed through roots and released in atmosphere during plant transpiration, or even directly evaporated from the soil.

3.1 Resources

Our approach focuses on the evolution of *resources* across the landscape. Resources are physical quantities that follow conservative laws. Components of the landscape compete for those resources, because they are in limited availability. Resources are thus one of the main ways for components to interact.

The water resource was our primary concern, because on one hand water plays a key role for plant growth and on the other hand its cycle is a well-known phenomenon, described for example in [19, 2] (see figure 1). Moreover, it offers nice challenges in modeling, since it covers a wide range of motions with various time scales. It is, as such, a good test case to derive the proper architecture for functional landscapes. However, the proposed architecture could be used for other resources as well: light, nutrients, carbon, *etc.*

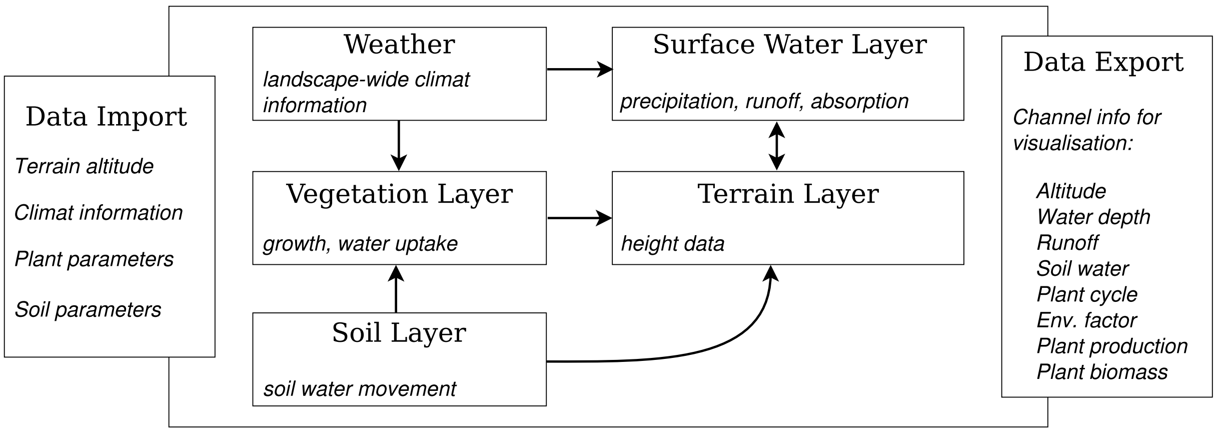

Figure 2: Links and relationships between the different elements of the simulator. The simulator is split in several *layers*. Each of them is responsible for running specific models. Arrows represent data fluxes that are established inside the architecture to allow it to function.

3.2 Layers

In our study, modularity is a key factor. Our objective is not only to make several models of different parts of the landscape interact, but also to use different models for the same part without impacting the functioning of the rest of the landscape. To achieve this goal, we isolate each model inside its own *layer* (see figure 2).

To reflect the spatial variability of the landscape, layers are subdivided into *cells* that contain the information needed by the models. In our implementation, most layers are decomposed into a regular grid of cells. However, the water layer for example uses a cell graph that is not regular.

Layers exchange information and resources. Since their spatial discretisation is not necessarily the same, interpolation methods are required in order for them to communicate.

3.3 Model specifications

Models should be adapted in order to run within our framework. Specifically, they should become resource-oriented, to communicate meaningfully with other models. They should not assume that resources are always available. Functioning under a shortage of resources is a key point for the simulation of competition. Functioning with an excess of resources is generally less problematic, but must be ensured as well.

4 Landscape sub-models

4.1 Plant Model

4.1.1 The Functional Structural Plant Model

The GreenLab plant model is used in our simulation. It is a functional-structural model, which means that it combines both functional growth and structure development, interacting together. Details can be found in [8].
The relatively low number of parameters and an advanced mathematical formalisation allows calibration of the model and thus reproduction of the behaviour of existing plants. The basis of the model is a structural factorisation of the plant architecture. This idea has led to an enormous improvement in computing time. It made the individual plant growth simulation very fast, and makes the model suitable for landscape simulations with individual plants.
Of course, the full model is a bit too complex for our needs. We choose to implement a simple model to experiment with the impact of the environment at the landscape scale. Specifically, most of the structural part is omitted.

4.1.2 Plant model implementation

We adapt the GreenLab production equation given in [12]:

$$dQ = E\,dt\,\alpha\S_p\left(1-\exp\left(-\beta\,\frac{S}{S_p}\right)\right) \qquad (1)$$

where dt is the time step (daily in our case), dQ is the biomass produced through photosynthesis over this time step, E is the environment factor (detailed below), S is the surface area of the photosynthetic leaves, α and β are empirical model parameters estimated from experimental data and S_p is a characteristic surface area which integrates the competition for light with neighbours. In the simple case of homogeneous stands or crops, S_p can be chosen as the inverse of the density, [5].
This equation is not the classical form of GreenLab because we transposed it to the calendar (physical) time. The computation of plant production is usually synchronised with the development cycle. This one is not determined by the calendar time but rather by the *thermal time* ([12]): new organs appear rhythmically at integer values of:

$$c = \int_0^t \frac{\max(0, T - T_b)}{T_g} d\tau \qquad (2)$$

where t is the current time, T_b is the base temperature below which development pauses, T_g is the number of degree-day necessary to complete a growth cycle. Synchronising the growth cycle with calendar time can be difficult, with possible side-effects in days when a cycle ends. We adopted an approach that is valid when the cycle duration is at least several time steps.

At the end of each cycle, the biomass produced is allocated to the organs of the plant according to their demands. The demand of the organs follows a bell-shaped curve during their functioning time.

The parameter E is the primary way of interaction between plants and their environment. We choose to make it dependent on water conditions and temperature:

$$E = E_0 E_T E_w \tag{3}$$

E_0 depends on the photosynthetically active incident radiation (PAR) and is chosen constant in this preliminary study, E_T depends on temperature, and E_w is a function of the soil water content.

The temperature effect on photosynthesis (also known as *biological efficiency*) is represented by a simple bell-curve between the minimum and maximum temperature for photosynthesis, with a maximum of 1 at the optimal temperature.

The water effect is simulated thanks to a root model. It allows communication between the soil model and the plant model and covers two processes. First, roots sense the amount of water in soil, allowing the plant to react to water depletion. Second, they take some water from the soil. This latter aspect has been the subject of several studies such as [24, 25], but seldom implemented with a FSPM.

The simplest model for the first process is to introduce an environment factor linearly related to the soil water content W (see [27] for applications of this model). However, this is not satisfactory considering the soil model outlined in section 4.2. Indeed, the same water content can lead to great differences for the plant according to the type of soil. That is because different soils have varying water retention capacity; it is harder for the plant to extract water from a soil with a very fine texture. That could explain some observed non-linearities as well.

For this reason, we propose a model for the environment factor that is based on the suction of the soil ψ:

$$E_w = \frac{\exp(\nu\psi) - \exp(\nu\psi_{min})}{\exp(\nu\psi_{max}) - \exp(\nu\psi_{min})} \tag{4}$$

where ψ_{min} is the soil suction at wilting point, ordinarily taken at $\psi_{min} = -150m$ regardless of the plant species, and ψ_{max} is the soil suction at field capacity, $\psi_{max} = -3m$ with the same hypothesis of independence from plant species. The parameter ν represents how the plant adapts to the soil water content. For a specific soil, there is a value of ν giving the linear relationship.

The water uptake itself is restricted to the first layer of soil. Uptake of plants is defined on a cell of soil, containing plants at a density δ, as described in [27]:

$$dW = \delta \frac{dQ}{wue} \tag{5}$$

where wue is the water use efficiency of the plant.

It is to be noted that, even with the varying E_w, the plant could attempt to take more water than what is available. The problem is purely numerical, and is handled by first checking that the amount of water available stays positive, and changing the production dQ accordingly if it is not the case.

4.2 Soil water movement

This section describes the model chosen to represent water movement inside the soil, under the surface of the terrain.

4.2.1 Richards equation

The fundamental equation describing the evolution of soil water content, in unsaturated conditions, is the Richards equation ([26]), given below in its one-dimensional form:

$$\frac{\partial \theta}{\partial t} = \frac{\partial}{\partial z}\left(K(\psi)(\frac{\partial \psi}{\partial z} - 1)\right) \tag{6}$$

where θ(dimensionless) is the volume water content of the soil, ψ(m) is the suction of the soil (negative in unsaturated soil), z(m) is depth, and $K(\psi)$(m.s^{-1}) is the soil hydraulic conductivity.

The behaviour of this equation depends on the two functions $K(\psi)$ (conductivity curve) and $\theta(\psi)$ (water retention curve). Several empirical models have been proposed to fit the measured behaviour of actual soils (see [16] for an overview). The best models give complex nonlinear equations, and thus some special care must be taken to ensure an accurate numerical resolution.

We choose the simplest version of the model, because the object of this work is to study rather the interaction of models than their inner complexities.

4.2.2 Linear form

Certain forms of the conductivity and water retention curves make the Richards equation linear. They allow fast and accurate resolution, while still appropriately representing the phenomenon (as used in [26]).

Let us first define:

$$\theta_e = \frac{\theta - \theta_r}{\theta_s - \theta_r} \tag{7}$$

where θ_s is the value of θ in a saturated soil, and θ_r is a fitting parameter, that can be interpreted as the residual water content when the soil is subject to an extreme depression.

Inspired by [16], we model the water retention curve as follows:

$$\theta_e(\psi) = \begin{cases} \exp(\lambda(\psi - \psi_d)) & , \quad \psi < \psi_d \\ 1 & , \quad \psi \geqslant \psi_d \end{cases} \tag{8}$$

where λ is a fitting parameter, and ψ_d is the air-entry value.

We also make the hydraulic conductivity simply proportional to θ_e:

$$K(\theta) = K_0(\theta_s - \theta_r)\theta_e \tag{9}$$

and Richards equation (6) becomes:

$$\frac{\partial \theta_e}{\partial t} = \frac{K_0}{\lambda}\frac{\partial^2 \theta_e}{\partial z^2} - K_0\frac{\partial \theta_e}{\partial z} \tag{10}$$

which is a classical advection-diffusion equation.

This equation is then solved by finite differences in several layers of soil that store the water resources used for example by plants. It is also used to compute a maximum depth of saturation achievable during precipitation events, thus allowing to model water absorption. The water that is not absorbed flows on the soil surface as described in the next section.

4.3 Surface runoff

Water runoff is a key point in hydrological simulations, since it is a phenomenon that effects the repartition of water resources over the whole landscape. It is relatively well understood, and many examples of simulations exist. Our approach was based on the idea of *contributing area* as described in [3]. However, we adapted the model to the specificities of our problem, as shown in the rest of this section.

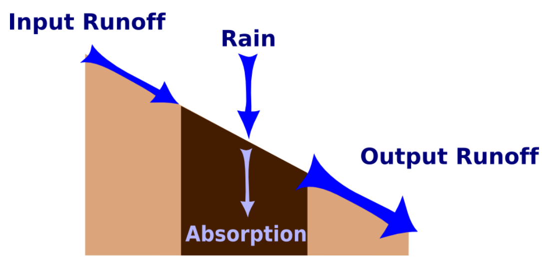

Figure 3: Water movements involved in runoff.

4.3.1 Specificities

Water runoff on the surface is clearly one of the quickest processes that we attempt to simulate in our landscapes. In a typical time step of one day, water can run over long distances. Moreover, the computing load of a detailed resolution of fluid dynamics is prohibitive. We want to know how the water is distributed in the soil after runoff. Thus we use a simplified model, mostly neglecting the depth of flow. Interestingly, such an approach is also chosen in research about erosion [14]. Water flow on the surface is integrated over the whole time step, and the runoff is computed at each point of the landscape during this step.

4.3.2 Implementation basis

The underlying terrain is used to first build a graph of water cells. At the beginning of the simulation, this graph is an image of the regular grid that is used in most digital elevation models (DEM). However, some cells of this basic grid can be merged under certain conditions. Thus, the graph used for the runoff simulation is based on the terrain grid but can differ. The goal of the algorithm is essentially to determine how water flows from one cell to another, following neighbourhood relationship materialised by the edges of the graph.

Runoff quantity can be modeled thanks to a very simple balance equation at the level of each cell:

$$R_o = \max(0, R_i + r - A - E_v) \qquad (11)$$

where R_i is the input runoff, arriving in the cell, R_o is the output runoff, r is the rain, A is absorption in the cell, and E_v is evaporation, as illustrated on figure 3.

The quantity R_o is then redistributed among the neighbours that are lower than the cell, according to the slope (*i.e.* the lowest cell gets most of the water). There are other approaches for the distribution, and the paper [22] gives a good overview of them. We have chosen the basic proportional model because even though it is simplistic, it gives realistic results.

Given this equation, it becomes clear that the problem is recursive. If we want to compute the output runoff of one cell, we have to know the input runoff. This input runoff is the sum of the output runoffs from higher neighbouring cells. A recursive algorithm was thus implemented, it computes runoff by going upslope, against the flow.

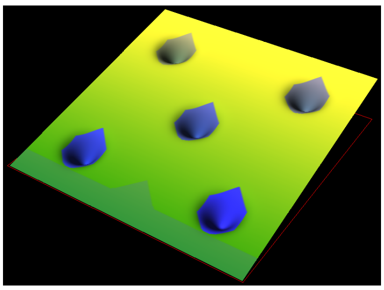

Figure 4: Cell flags on a simple slope. The terrain is colored from green to yellow according to altitude. The blue color indicates how the cells are flagged. All the water flowing out of similarly colored cells end up in the same pit. Some cells do not have a flag, when the water flowing out arrives in more than one pit. A pit's outlet is the lowest cell on the edge of the area flagged with the pit's ID.

4.3.3 Lakes

The main problem with the algorithm described occurs in cells that have no lower neighbours towards which the water could flow. We call those particular cells *pits*. Water arriving in pits through runoff cannot flow out and therefore accumulates. Pits are thus the starting points of lakes over the landscape. Designing algorithm to detect and fill lakes is a complex problem, and several strategies can be adopted. Other authors have designed algorithms that actually fill the lakes (see [23] for example). The problem is that their objective was not entirely similar to ours, entitling them to hypotheses that we cannot make. In particular, the assumption is made that the lake will fill until it overflows. This is mostly true when large time period are considered, but over a day, a lake could be only partially filled, without overflowing. It depends on the amount of water that contributes to this lake.

The algorithm we have designed proceeds iteratively in 3 phases. First, an outlet point is associated to each pit, according to topological properties of the underlying terrain, thanks to flags that are set on every cell to indicate their relationship with the pits (see figure 4). Second, the recursive runoff algorithm is executed, along with a computation of the reserve available in each lake at every altitude up to its outlet's height. Third, lakes' overflows are detected. When a lake overflows, it invalidates the data computed during the runoff phase, and thus the algorithm iterates the 3 phases until no overflow happens. The result of a simple runoff simulation as shown in Figure 5.

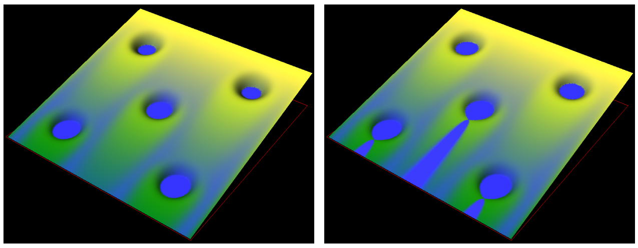

Figure 5: Runoff simulation on a simple slope with holes. Blue color visualises the runoff value, overflooding can be seen on the right.

4.4 Environmental models

Temperature and precipitation are considered as data for our simulator. There is no retro-action from plants to temperature, for instance. The simulator requires daily values for temperature and precipitation at each point of the landscape. We established algorithms to procedurally generate data at this level of detail, based on real data corresponding to the region of Montpellier, France, in 2005, available from the Internet. These real values were monthly and not dependent on the position in the landscape. From these, daily values were procedurally generated. The daily values could also be modulated according to the position in the landscape, taking into account simple effects such as a vertical temperature gradient.

Here, the goal was not to work with accurate weather conditions models, but just to allow spatial and temporal heterogeneity that may be obtained from real measurements or specific models.

5 Visualisation techniques

With a daily time step for the updating of layer interactions, visualisation and analysis tools are necessary. A classical one-year simulation output set is composed by 8 to 10 channels, at the resolution of the digital elevation model (thus 0.1 to 2 million cells), for 365 days; it thus leads to a total storage of 0.2 to 6 Gigabytes. Specific user oriented exports are the classical way to analyse and visualise simulation results. In the simulation process, at each step, the simulator dumps selected channels on the disk, for later off-line visualisation and analysis.

5.1 Exploring simulation output

For each simulated day, a binary multichannel layer record is dumped on the disk. This record is a table of the spatialised cells, each of them containing the scalar values of the user selected channels. First value stands for the terrain altitude, while others are open to user selection. Each multichannel record header specifies the number of channels, an optional scaling factor and a name. Typical records used in this paper contain 5 to 8 channels: terrain elevation, water depth, water contents in soil, runoff, temperature, plant biomass production, cumulated biomass, plant cycle. Simple ASCII files describing day per day environmental conditions are also exported: precipitation level, average temperature.

5.2 Layer visualisers

The proposed visualiser is a real time multi-channel geometrical mesh visualiser. It aims at interactively visualising any combination of channels (from a single one to four), combining geometrical aspect, color mapping and color blending for a given simulated day. Simple classical 3D navigation and navigation through day and time is implemented. Simple illumination is used.

Each channel defines a geometrical mesh, a color table and a mapping function. In practice, each channel is dedicated to a selected layer property (terrain altitude, temperature, water contents in soil, etc). Color tables are classical look-up tables, built from 256 or 4096 entries with simple colors.

Mapping functions specify how channel scalar values are mapped to the color table entries, *i.e.* using constant, linear, exponential, logarithmic, positive or negative function to affect a given scalar value to its color-map entry (thus its color).

The visualiser combines and blends the four channel (geometrical aspect, color table, and mapping functions). The final geometry is defined from a single or from two channels. In this case, the final geometry can result from a logical operation between geometrical meshes, for instance water flood heights over terrain heights (in order to see rivers and lakes, see Figure 5). It can also be processed as two separate geometrical components, a classical mesh built from scalar altitudes, supporting for each cell simple graphical primitives (lines, or spheres) the sizes (length, or radius) of which are defined from the second channel scalar values (see Figure 8). Similar principles are applied to color table definition and mapping functions. Two specific look-up tables are chosen from the four selected channels, as well as one or two mapping functions. Finally, thanks to transparency capabilities of modern computer graphics, a blending mode is selected by the user to choose or combine mapping functions. As a result, a functional view of the scene is defined by a geometry built from one or two channels, a combination of color tables from one or two channels and a blending linear transparency function on the color mapping functions.

5.3 Channel indicators

Not many analysers have been developed up to now, we should rather speak of indicators. Two timescales are considered, the current day, corresponding to the current view in the visualiser, and the year.

At the day level, for each selected channel, the scalar property distribution on the full scene can be visualised on an histogram and a cumulated histogram. It also displays average, standard deviation, extrema, and median value. Statistical dispersion is computed, main distribution orientation deduced. Those values can be exported for further external analyses.

At the year level, curves corresponding to the four channel properties are displayed, resulting from the daily average value over the scene. Rainfall histogram is drawn in background, for visual comparison. User can switch to a detailed layer view, focusing on extrema, average and standard deviation curves of a specific channel, as shown in Figure 8.

6 Results

We present here some simulation results corresponding to simple cases. These examples cannot be considered as validations, but, beside their academic interest, they were used for consistency check (conservation of water resource, comparisons with simple crop models, *etc.*).

For the sake of clarity, we present here some results with very few changes in environmental conditions. Despite the fact that the simulator is able to manage variable local properties for vegetation and soil we consider here an even-aged and spatially homogeneous plant population (with a given constant density) all over the terrain. Soil parameters (except altitude) are also chosen constant, as well as rainfall level all over the terrain. We also consider a vertical, constant and spatially homogeneous incident light radiation. This last condition is quite restrictive on the yearly basis of our simulations, but this article mostly focuses on the water resources and further work will integrate a detailed light model. However, variations of E_0 in Equation (3) over the year would be very easy to implement to illustrate season effect.

More generally, the proposed and developed simulation and visualisation tools are not restricted to all these specific constraints which are mainly related to the availability of consistent data.

We used simple synthetic terrains and digital elevation models downloaded from `http://www.helensimage.com/tg.htm`.

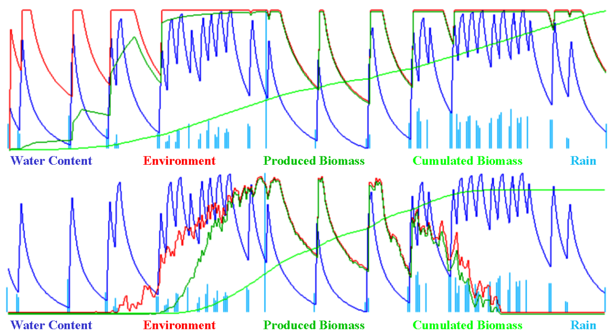

Figure 6: Temporal effect of temperature. Various quantities, averaged over the landscape, are plotted as functions of time: precipitation in light blue, soil water content in dark blue, environment factor in red, produced biomass in dark green, cumulated biomass in light green. Top: no temperature variation — Bottom: temperature varies according to time.

6.1 Temperature effect

Temperature mainly affects plant growth cycle as detailed in section 4.1.2. Therefore, temperature variations, both in time and space, impact access to water resources whose level changes from one location to another and over time. The proposed application was used to illustrate such effects on simple cases, with the same raining conditions (see 4.4).

1. Constant temperature both in space and time: the reference to be compared with. Temperature curve, rainfall, growth cycle curve and cumulated biomass are shown in Figure 6.

2. Changing temperature according to time: temperature corresponding to the raining condition are used (see 4.4). The resulting curves are given in Figure 6.

3. Changing temperature according to situation: a classical model modulates locally the temperature according to altitude. On the example, the local variation is quite low (1°C over the altitude range of the terrain). The curves are very similar to the previous case. However, this small change of temperature impacts significantly the spatial repartition of the growth. This simple example shows the potential applications to the study of global warming.

6.2 Effect of plant density

The following simulation example illustrates the effect of vegetation density. Even though the incident light is supposed constant and homogeneous in our simulation, the effects of competition for light can still be illustrated since they are included in the GreenLab production equation (1) through the parameter S_p. However, to be able to account for the differences between south and north facing slopes, a more detailed radiation model has to be implemented.

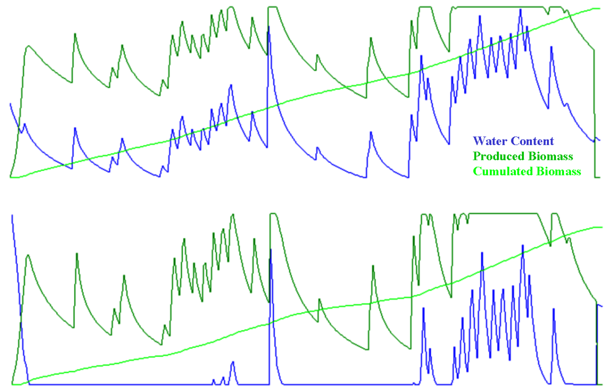

Figure 7: Effects of varying plant density. Plant density is respectively 1 plant/m^2 (top) and 15 plants/m^2 (bottom). The curves are averages over the landscape: soil water in blue, produced biomass in dark green, cumulated biomass in light green. Final biomass produced was only multiplied by 9 between the two cases. This can be explained by water shortage, which is readily apparent from the dark blue curves.

Population density also affects the use of water: as density increases, biomass production increases proportionally. It remains true until:

1. Water becomes a limiting factor. Biomass production is then limited by the available water in soil.

2. Leaf Area Index (LAI) becomes a limiting

factor. This is due to the plant growth model, and more precisely the S_p factor (see 4.1.2). When plants get closer, this surface diminishes, and biomass production is affected.

The first effect depends on precipitation conditions and their variations with time. It is thus not always seen during the simulation. The second effect does not depend on time in our implementation. It could be made dependent on plant growth, though.

6.3 Full landscape simulation

The simulations of plant population growth are run on a pseudo-realistic terrain model. They illustrate how temperature and water interplay may induce temporal and spatial variability. Results are shown in Figure 8 and commented below.

1. No water limitation: plant growth is purely driven by temperature and almost uniform, except for some steep parts of the terrain are not able to absorb enough water, which explains the fact that growth is weaker in those areas. The spatial histogram is representative of the homogeneity.

2. Water stress: precipitation was artificially lessened. It amplifies the slope effect. Since steep areas cannot absorb as much water, plant biomass production is smaller. Water stress happens in summer, but plant growth continues because the soil is alimented in water from below. A lake still forms in the crater and saturates the soil underneath. The main effect of the water limitation is to increase the spatial variability of growth.

3. Water Limitation and temperature variation: compared to the former case, temperature variation according to height adds temporal variability to the growth; plants in high areas start their growth later in the year. This is readily apparent in the curves of the total biomass produced. The first plants start growing at day 38, the last start at day 147.

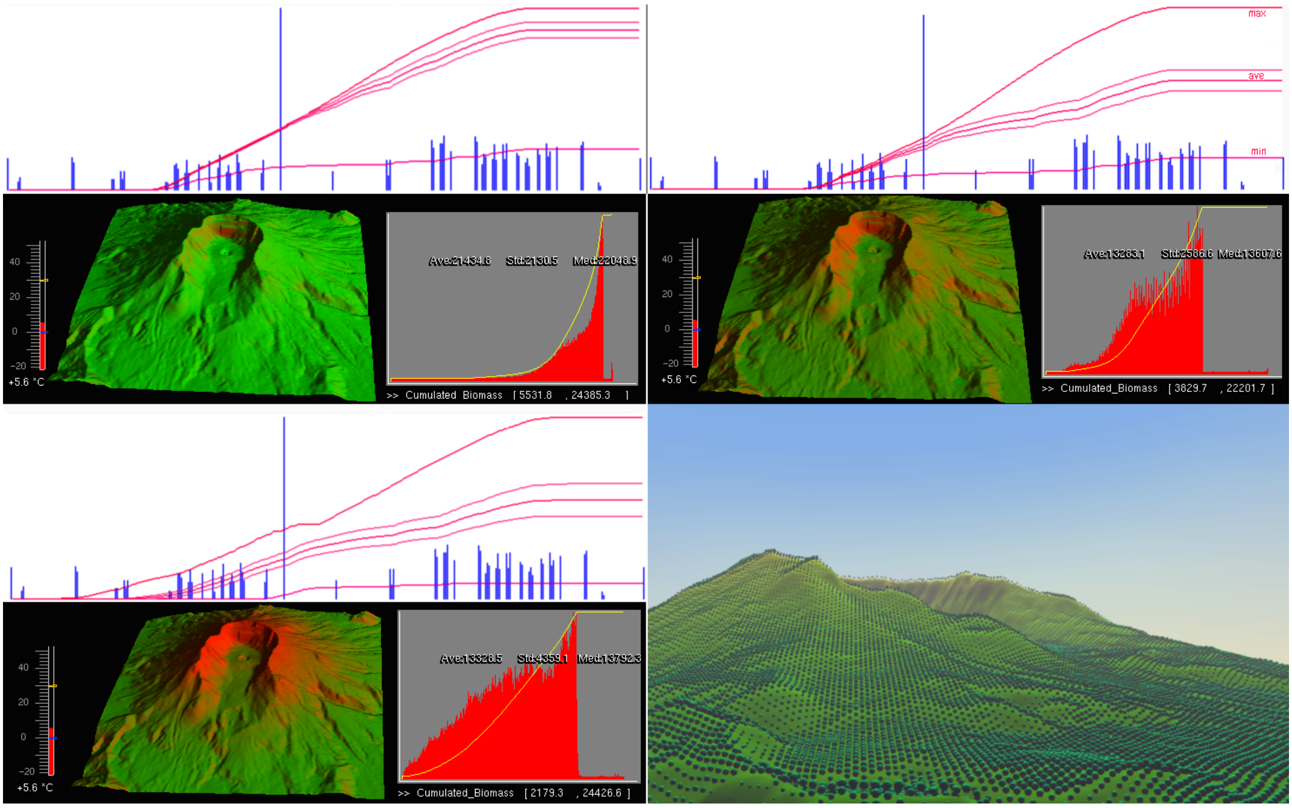

Figure 8: Full Simulation. From top to bottom, left to right: no limitation, water limitation, water limitation and temperature variation, semi-realistic visualisation. Below each set of curves (corresponding to those described in section 5.3) is a view showing spatial variability (high biomass in green, low biomass in red). Spatial histograms of cumulated biomass production are also represented, with a gray background.

7 Conclusion and further work

A simple dynamic water cycle model has been proposed, providing a spatialised water resource potential. Plant biomass production models, even very simple ones, can easily interact with climatic conditions (rain, temperature, light) and impact locally the water resources. Basic tests and experiments show realistic behaviors and a good communication between models.

On the simulations, both the cumulated and daily produced biomass show local spatial heterogeneity due to water resource competition, despite constant environmental conditions. With the proposed approach, local climatic resource variations can be easily introduced. We plan to implement light condition heterogeneity in relation with season and terrain local orientation, interacting with the plant model.

But one of the hardest problem is that of side-effects; since we study a limited portion of the Earth's surface, with generally unknown boundary conditions, it is difficult to ensure that borders do not have a far-reaching influence on the results. This problem is still under study and is crucial for validation.

From this preliminary work, we did also learn the complexity of defining the appropriate models for each biophysical process at the appropriate levels of description and of interaction with resources. It requires exchanges between scientists from various domains, underlining the need of advanced analysing and visualisation tools for mutual understanding and knowledge sharing.

The synchronisation model is also being modified to function accurately with smaller plant cycles. This is part of the ongoing reflexion about simulation architecture. Smaller cycles also open new potentials in terms of realistic rendering. We are currently working on such rendering, using a sky dome and changing illumination parameters based on sunlight properties. We are also implementing a rain particle system and an adaptive fog defined from rain level, temperature and time of the day. Exhaustive 3D visualisation of simulated plants at organ level is an objective of our work, but our current focus is a new geometrical representation of the biomass production to obtain more realistic plant shapes from their wood volumes and leaf areas only.

So far, soil erosion and sedimentation models are not implemented at the small time scales considered in this preliminary study and drastic natural hazard events are ignored. Nevertheless terrain elevation and soil properties evolution models will soon be implemented at a slower update rate.

References

[1] M. Alweis, O. Deussen. Modeling and visualization of symmetric and asymmetric plant competition. In *Eurographics Workshop on Natural Phenomena, Dublin*, pages 83–88, 2005.

[2] J. Beauchamp. L'eau et le sol. `http://www.oleiculteur.com/L'eau%20et%20le%20sol.htm`.

[3] K. J. Beven and M. J. Kirkby. A physically based variable contributing area model of basin hydrology. *Hydrological Sciences Bulletin*, 24(1):43–69, 1979.

[4] Chiba, N., Muraoka, K., and Fujita, K. An erosion model based on velocity fields for the visual simulation of mountain scenery. *The Journal of Visualization and Computer Animation*, 9(6):185–194, 1998.

[5] Cournède, P.H., Mathieu, A., Houllier, F., Barthélémy, D., de Reffye, P. Computing competition for light in the Greenlab model of plant growth: a contribution to the study of the effects of density on resource acquisition and architectural development. *Submitted*, 2007.

[6] Cournède P.-H., Kang M.-Z., Mathieu A., Barczi J.-F., Yan H.-P., Hu B.G., de

Reffye P. Structural factorization of plants to compute their functional and architectural growth. *Simulation*, 82(7):427–438, 2006.

[7] Dachsbacher, C. *Interactive terrain rendering : towards realism with procedural models and graphics hardware*. PhD thesis, University Erlangen-Nürnberg, Computer Sciences Institute, 2006. 162 p.

[8] P. De Reffye, B.-G. Hu. Relevant choices in botany and mathematics for building efficient dynamic plant growth models: Greenlab case. In *Proceedings of PMA03, Beijing*, pages 87–107, 2003.

[9] Deussen, O., Hanrahan, P., Lintermann, B., Měch, R., Pharr, M. and Prusinkiewicz, P. Realistic modeling and rendering of plant ecosystems. In *SIGGRAPH'98 Proceedings*, pages 275–286, New York, NY, USA, 1998.

[10] Fournier, A., Fussell, D., and Carpenter, L. Computer rendering of stochastic models. *Commun. ACM*, 25(6):371–384, 1982.

[11] N. Greene. Voxel space automata: modeling with stochastic growth processes in voxel space. In *SIGGRAPH '89:*, pages 175–184, New York, USA, 1989.

[12] Guo, Y., Ma, Y.T., Zhan, Z.G., Li, B.G., Dingkuhn, M., Luquet, D. , De Reffye, P. Parameter optimization and field validation of the functional-structural model greenlab for maize. *Annals of Botany*, (97):217–230, 2006.

[13] J. Hammes. Modeling of ecosystems as a data source for real-time terrain rendering. In *DEM 2001, Digital Earth Moving First International Symposium*, volume 2181, pages 98–105. Springer Velag, LNCS, 2001.

[14] A.D. Howard. A detachment limited model of drainage basin evolution. *Water Resources Research*, 30(7):2261–2285, 1994.

[15] Jaeger, M. and Teng, J. Tree and plant volume imaging - an introductive study towards voxelized functional landscapes. In *Proceedings of PMA03, Beijing*, pages 169–181, 2003.

[16] K. Kosugi, J.W. Hopmans, J.H. Dane. *Water Retention and Storage - Parametric Models*, volume 5, pages 739–758. Soil Science Society of America, 2002.

[17] Liu, Y.B., Gebremeskel, S., De Smedt, F., Hoffmann, L. and Pfister, L. Predicting storm runoff from different land use classes using a gis-based distributed model. *Hydrological Processes*, 20(6):533–548, 2006.

[18] Musgrave, F.K., Kolb, C.E., and Mace, R.S. The synthesis and rendering of eroded fractal terrains. *Computer Graphics*, 23(3):41–50, 1989.

[19] Musy, A. Cours d'hydrologie générale, EPFL. `http://hydram.epfl.ch/e-drologie/`.

[20] B. Neidhold, M. Wacker, O. Deussen. Interactive physically based fluid and erosion simulation. In *Eurographics Workshop on Natural Phenomena, Dublin*, pages 25–32, 2005.

[21] Schneider, J., Boldte, T., and Westermann, R. Real-time editing, synthesis, and rendering of infinite landscapes on GPUs. In *Vision, Modeling and Visualization 2006*, 2006.

[22] D.G. Tarboton. A new method for the determination of flow direction and upslope areas in grid digital elevation model. *Water Resources Research*, 33(2):309–319, 1997.

[23] G. Tucker, N. Gasparini, S. Lancaster. An integrated hillslope and channel evolution model as an investigation and prediction tool. Annual report, Child model, 1997.

[24] N. Varado, I. Braud, P.J. Ross. Development and assessment of an efficient vadose zone module solving the 1D Richards' equation and including root extraction by plants. *Journal of Hydrology*, 2005.

[25] J.A. Vrugt, J.W. Hopmans, and J. Simunek. Calibration of a two-dimensional root water uptake model. *Soil Science Society of America Journal*, 65(4):1027–1037, 2001.

[26] T.P. Witelski. Motion of wetting fronts moving into partially pre-wet soil. *Advances in Water Resources*, 28:1133–1141, 2005.

[27] Wu, L., Le Dimet, F.-X, Hu, B.G., Cournède, P.-H., De Reffye, P. A water supply optimization problem for plant growth based on greenlab model. *ARIMA Journal*, pages 194–207, Nov. 2005.