

Authoring Landscapes by Combining Ecosystem and Terrain Erosion Simulation

GUILLAUME CORDONNIER, Univ. Grenoble Alpes & CNRS (LJK) and Inria

ERIC GALIN, Univ Lyon, Université Lyon 2, CNRS, LIRIS

JAMES GAIN, University of Cape Town

BEDRICH BENES, Purdue University

ERIC GUÉRIN, Univ Lyon, INSA-Lyon, CNRS, LIRIS

ADRIEN PEYTAUVIE, Univ Lyon, Université Lyon 1, CNRS, LIRIS

MARIE-PAULE CANI, Univ. Grenoble Alpes & CNRS (LJK) and Inria

We introduce a novel framework for interactive landscape authoring that supports bi-directional feedback between erosion and vegetation simulation. Vegetation and terrain erosion have strong mutual impact and their interplay influences the overall realism of virtual scenes. Despite their importance, these complex interactions have been neglected in computer graphics. Our framework overcomes this by simulating the effect of a variety of geomorphological agents and the mutual interaction between different material and vegetation layers, including rock, sand, humus, grass, shrubs, and trees. Users are able to exploit these interactions with an authoring interface that consistently shapes the terrain and populates it with details. Our method, validated through side-by-side comparison with real terrains, can be used not only to generate realistic static landscapes, but also to follow the temporal evolution of a landscape over a few centuries.

CCS Concepts: •Computing methodologies → Shape modeling;

Additional Key Words and Phrases: Landscape, Terrain, Vegetation, Erosion, Stochastic, Simulation of Natural Phenomena

ACM Reference format:

Guillaume Cordonnier, Eric Galin, James Gain, Bedrich Benes, Eric Guérin, Adrien Peytavie, and Marie-Paule Cani. 2017. Authoring Landscapes by Combining Ecosystem and Terrain Erosion Simulation. *ACM Trans. Graph.* 36, 4, Article 134 (July 2017), 12 pages.

DOI: <http://dx.doi.org/10.1145/3072959.3073667>

1 INTRODUCTION

Landscape authoring involves modeling the entirety of a natural scene incorporating both vegetation and terrain and is an important part of a designer's workflow in computer graphics. The human visual system has evolved into a highly effective interpreter of the

Author's addresses: Guillaume Cordonnier and Marie Paule Cani, Centre de recherche Inria Grenoble - Rhône-Alpes, 655 avenue de l'Europe, 38 330 Montbonnot-Saint-Martin, France. Éric Galin, Éric Guérin, Adrien Peytavie, LIRIS UMR 5205, Université Claude Bernard Lyon 1, Bâtiment Nautibus, 43 bd du 11 novembre 1918, 69622 Villeurbanne CEDEX, France. James Gain, Department of Computer Science, University of Cape Town Private Bag Rondebosch, Cape Town, 7701, South Africa. Bedrich Benes, Purdue University, High Performance Computer Graphics Lab, Knoy Hall of Technology, 401. N. Grant Street West Lafayette, IN, USA 47907-2021.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2017 Copyright held by the owner/author(s). Publication rights licensed to ACM. 0730-0301/2017/7-ART134 \$15.00

DOI: <http://dx.doi.org/10.1145/3072959.3073667>



Fig. 1. Our landscape authoring framework enables interactive modeling of the mutual interaction between terrain and vegetation, such as vegetation destroyed by rock-slides and plants growing over accretion areas.

natural world and we are capable of rapidly identifying features that we recognize as important or visually plausible. This poses a challenge for designers of virtual landscapes, because errors and inconsistencies are unforgivable to observers.

There has been significant progress in landscape modeling. Methods for virtual terrain date back to early days when the fractal characteristics of mountains were first explored [Mandelbrot and Pignoni 1983]. However, it was soon noticed that terrains are affected by geomorphological processes and cannot be described by fractals alone. Various forms of terrain erosion [Benes and Forsbach 2002; Chiba et al. 1998; Musgrave et al. 1989] and example-based approaches [Zhou et al. 2007] have since been introduced. Vegetation cover has typically been created by treating plants as particles that can be placed on pre-generated terrain by random, simulation [Deussen et al. 1998] or example-based [Emilien et al. 2015] processes. Despite significant progress, there are still many open problems in landscape modeling. Landscapes exhibit huge spatial and temporal variance that is affected by multiple agents acting at different rates and scales. For instance hydraulic erosion, landslides, and lightning can act very quickly and move or remove large amounts of plant matter and soil material within hours, while thermal erosion typically acts slowly and is expressed only over very long time spans. These different rates make it difficult to simulate the spatio-temporal evolution of landscapes, i.e., how a given terrain would look after some period of time. Geomorphological and

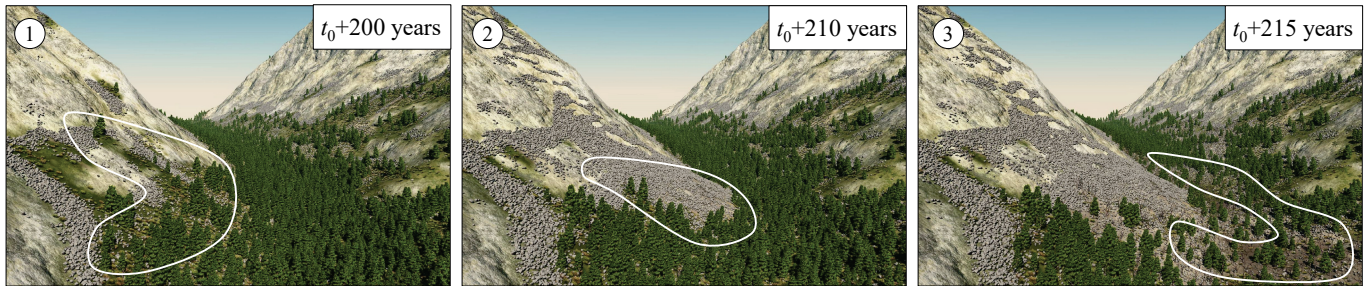


Fig. 2. Our framework combines layered terrain and vegetation data and supports their interlinked simulation, which can be driven by users editing layers or triggering natural events. (1) The user first provides a bare-earth digital elevation map for time step t_0 and our framework simulates interleaved erosion and plant growth, up to $t_0 + 215$ years. (2) In the next time step at $t_0 + 210$ years, a landslide creates boulders that destroy vegetation. One year later, the designer triggers a fire in the valley, which spreads to consume part of the forest. (3) After four more years at $t_0 + 215$, the remaining trees have continued growing, new saplings have germinated and the humus layer is beginning to regenerate. The white loops indicate affected areas.

ecosystemic phenomena also act differently depending on a plethora of factors such as location, slope, temperature, sun exposure, and soil type.

The key observation of our work is that the shape and temporal evolution of landscapes are significantly affected by the interaction between vegetation and erosion, a factor that has thus far not been considered in computer graphics. While vegetation may affect the speed of the geomorphological processes and reduce their effects such as landslides, many of them such as erosion, rock falls, and soil deposited by running water, also have a strong impact on the vegetation.

We present a novel, unified framework for *landscape modeling* that allows for interactive modeling of the mutual interaction between various geomorphological phenomena and their effects on terrain and vegetation. The input is an elevation map, which is modified, textured, and populated with consistent sets of details such as rocks, trees, or bushes by our simulation. The user retains full control over the terrain, vegetation, and their co-evolution during an interactive modeling session; she can tune the control parameters of a wide range of geomorphological phenomena, their interplay, and the weather. The user can also directly edit each layer state, such as the current plant cover, using brushes. As a result, the 3D layered model evolves and exhibits features that are difficult to achieve with other landscape modeling methods (Figure 2).

We claim the following technical contributions: We present an original framework for efficient, *stochastic simulation* of multiple phenomena and their interactions. Simulations are driven by a consistent set of models for the geomorphological and ecosystemic agents that simultaneously modify terrain and vegetation and influence their temporal evolution over time. To our knowledge, ours is the first approach that allows user-controlled authoring of landscapes by using an ecosystem simulator that takes into account the entire cycle of vegetation, from germination through to death, and ultimate reduction to an organic mulch (humus), as well as providing a unified treatment of disturbance events, such as fire, lightning and landslides.

2 PREVIOUS WORK

Solutions to ecosystem simulation and terrain modeling can be usefully classified according to procedural, simulation or interactive approaches. Although a focused review based on this classification is provided here, there are more general surveys of terrain representation [Natali et al. 2013], procedural terrain [Smelik et al. 2014] and plant [Deussen and Lintermann 2006] modeling available.

Ecosystem simulation builds on vegetation models dating back to work by Lindenmayer [1968] and Bloomenthal [1985] and the book of Prusinkiewicz and Lindenmayer [1990]. A seminal paper in ecosystem simulation for computer graphics by Deussen et al. [1998] set a Lagrangian trend by describing each plant as a particle and simulating an ecosystem as a competition for resources in which colliding plants are evaluated and the fittest survives. Follow-up research focused on multilevel communities [Lane and Prusinkiewicz 2002], adding virtual agents to work as gardeners [Benes and Espinosa 2003], real-time ecosystem simulation [Benes et al. 2009], combining ecosystems with urban simulations [Benes et al. 2011], greater botanical fidelity [Ch'Ng 2013] and pause-and-restart localized ecosystem editing [Bradbury et al. 2015]. To the best of our knowledge, ecosystem simulation has not been combined with erosion simulation for landscape modeling.

Terrain and landscape modeling methods were first based on procedural approaches and applied noise functions to imitate the fractal structure of terrains [Fournier et al. 1982; Voss 1985]. Subsequent methods combined fractal interpolation with river and valley modeling [Kelley et al. 1988], applied L-systems to river and landscape generation [Prusinkiewicz and Hammel 1993], and deformed fractal terrains to fit certain constraints [Stachniak and Stuerzlinger 2005]. More recently, compactly supported function representations [Génevaux et al. 2015] and sparse procedural representations [Guérin et al. 2016] were proposed to control the placement and distribution of procedurally defined landforms features. The common problem with pure procedural methods is their inability to generate terrains that have been affected by geomorphological agents, such as water or temperature.

Simulation-based approaches use erosion simulation either to further improve existing terrains or to create new terrains from scratch. The seminal work of Musgrave et al [1989] introduced

thermal weathering and hydraulic erosion and was later extended by using: layered data structures [Benes and Forsbach 2001], fully 3D simulation with smoothed particle dynamics [Krištof et al. 2009], or discrete space subdivision [Benes et al. 2006]. Simulations have been used in interactive frameworks [Neidhold et al. 2005; Št'ava et al. 2008] and accelerated by implementation on many-core hardware architectures [Mei et al. 2007; Vanek et al. 2011]. Current work allows for large-scale simulation of erosion at the level of entire mountain ranges [Cordonnier et al. 2016]. While erosion simulation produces terrains that are geomorphologically correct, the large spatial and temporal scales of the simulations (about 100 meters per grid-cell, with time steps in the thousands of years) and the limited user-control prevent those algorithms from being used extensively for terrain authoring.

Various interactive and example-based methods for terrain modeling address the major issues with procedural and simulation-based approaches. One of the first example-based methods [Zhou et al. 2007] drew on a set of digital elevation exemplars to create terrains using texture synthesis. This approach was extended and parallelized by Gain et al. [2015] to allow various forms of interactive user control. Terrain concavity was addressed in an approach [Peytavie et al. 2009] that allows interactive creation of 3D models of caves with boulders. Recent work uses statistical analysis to train brushes that can transfer stochastic distributions of features, such as trees, rocks, houses and roads, from one terrain to another [Emilien et al. 2015] or intersection-free placing of objects [Guérin et al. 2016] controlled by density functions. The work of Gènevaux et al. [2013] builds complex river networks that are used to construct the corresponding terrain. These methods provide good control over the phenomena they address, but do not ensure the geological consistency of results, or a plausible interplay between the morphology of the terrain and the details that populate it.

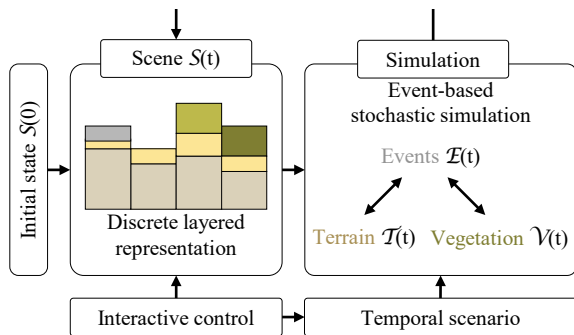


Fig. 3. Framework: The layered scene $S(t)$ is affected by the simulation, which includes an interplay of geomorphological and ecosystem events, vegetation and terrain.

3 METHOD OVERVIEW

Our method computes the temporal evolution of a terrain covered with vegetation under the combined action of various *geomorphological and ecological events*. Environmental effects such as rain, gravity, temperature, wind, fire, and lightning not only directly impact the evolution of the terrain and the development of vegetation, but also

have an indirect impact, as vegetation and the upper terrain horizon [Grunwald 2016] mutually interact in a complex feedback loop. For example, vegetation can prevent rockslides but may equally be destroyed by falling rocks; vegetation absorbs water but flooding may also uproot plants.

Our framework (Figure 3) uses a discrete layered model that unifies the representation of different terrain materials and types of vegetation. Geomorphological and ecological events modify the data stored in the various layers during simulation. These layers collectively represent the state of the scene at each frame.

We use a discrete spatio-temporal simulation. At a given time step, the scene, denoted as $S(t)$ is a set of 2D discrete layers composed of $n \times n$ cells. The simulation process computes the evolution of the scene $S(t + \Delta t)$ from scene $S(t)$ by stochastically applying a number of events to the cells of the terrain.

3.1 Layered landscape model

The layered model is an ordered sequence of layers from which a static landscape representation can be derived at any given time step. Alternatively, a series of simulation results can be used to show the temporal evolution of the landscape over a chosen time span. Specifically, a discrete regular grid of size $n \times n$ cells is combined with a multi-layer ordered data-structure to represent different terrain materials and plant matter in every cell (Figure 4).

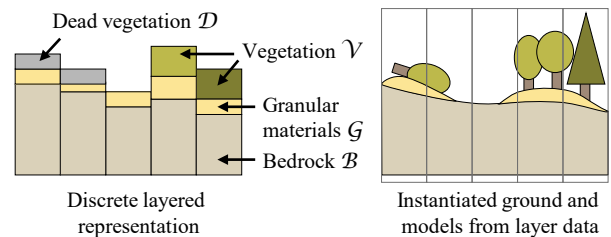


Fig. 4. Representation of the ground and vegetation layers. Bedrock B and granular materials G , consisting of rocks, sand, and humus, define the layers of terrain. Plants are represented using various vegetation layers V , for grass, shrubs and trees. The layers store data used in simulations, such as canopy density and age for vegetation layers, and moisture for granular layers.

Terrain materials sit on top of a bedrock layer B , which defines the base elevation. The broken rock, sand, and humus layers, denoted as R , C and H , represent the respective material thicknesses. In the remainder of the paper, we refer to these layers as *granular material layers* $G = \{R, C, H\}$. The ground elevation, denoted as E , is defined as the sum of the bedrock and the granular material thicknesses: $E = B + G$. The slope of the terrain between two points p and q will be defined $s(q) = (E(p) - E(q)) / \|p - q\|$. In addition, cells are also characterized by values for soil moisture content M , and the average daily duration of direct sun exposure I , both of which are crucial to ecosystem simulation.

Vegetation layers store the density maps and other important parameters for every vegetation type. For trees and shrubs this includes the count, age, and height of plants in each grid cell, while for grass density alone is sufficient. We also model dead vegetation D , which decays into humus and plays an important role in

the ecosystem simulation. Total vegetation density \mathcal{V} is given by the sum of the grass, shrub, and tree densities, weighted by their respective importance.

3.2 Simulation

The novelty of our method lies in a stochastic simulation that supports robust and efficient integration of many phenomena acting on a landscape. In this section, we first describe the main challenges, and then explain our solution.

Challenges: An obvious approach to simulating multiple phenomena would be to jointly simulate the incremental actions of all agents (water flows, while thermal erosion fractures rock and plants grow), on each cell of the terrain, for each simulation frame.

Unfortunately, this strategy works against many of the data-structures and optimizations proposed in previous work, which presuppose a single agent acting in isolation. For instance, water-pile models accumulate water but often yield large water-level disparities between neighboring cells. Even with dampening, the disparity at intermediate stages can introduce significant instabilities when combined with chaotic events, such as rockfalls or plant growth.

This issue can be viewed from another perspective: a large, extensible simulation framework, with many parameters and unknown, and non-linear relationships, is most typically resolved using the forward Euler method. In this case, to achieve stability, the global time step must be chosen with respect to the most ill-conditioned equation in the simulation. This lockstep prevents efficient simulation of the other factors. Therefore, such an approach is limited to simple algorithms and data-structures or fine grids and small simulation steps.

Our solution: Instead of jointly simulating phenomena we successively generate a large number of individual events, which act separately on the landscape. Each event begins in a cell, typically follows a path through other cells (generally with some decay) and ends in bounded time. To generate the desired effect, both the sequence and specifics of individual events are stochastic. More precisely, in our formalism *events* obey the following rules:

- Propagate without backtracking to at most a single adjacent cell. An event may thus spawn a path so long as this does not branch, reverse course or form loops. It is important to note that this restriction only applies to propagation: an event may modify the layers of any number of neighboring cells along the way.
- Involve a bounded volume of material.
- Terminate in bounded simulation time δt , sufficiently short to be negligible with respect to the simulation time step $\delta t \ll \Delta t$.

The first rule limits execution time, the second promotes stability, and the third ensures convergence (by making it legal to perform such a decomposition).

This strategy is motivated by the seemingly stochastic nature of many events in real terrains, either because they are triggered at random (e.g., rockslides) or their effects appear chaotic (e.g., the spread of fire). Moreover, the path of such events is often determined by fine-scale features that are obfuscated by the discretization of

a simulation grid. Although a single event always propagates to a single neighboring cell, branching effects, such as rocks scattered during landslides, are achieved by the accumulation of many events in neighboring locations. The overall simulation can be described as follows:

- Choose a random position \mathbf{p}_0 , with uniform probability, from among all terrain cells;
- Choose a random event, again with uniform probability, from all the available events;
- Activate the event at \mathbf{p}_0 and simulate it until it terminates;
- Store the effects (transported material, plant growth) in the relevant terrain layers.

This process is repeated a very large number of times N for each simulation step, which ensures a convergence to plausible results. In our implementation, N is a product of the number of grid cells and different event types ($N > 256 \times 256 \times 10 \approx 650,000$ in all our examples), so that all terrain cells are likely starting points for at least a few events.

The time step of our simulation Δt is set to one year. This represents a balance between the relatively rapid development of vegetation and longer-term impact of erosion, while allowing efficient simulations that span several centuries. In practice, Δt can be reduced over the last few simulation months before exporting final geometry so as to account for seasonal variations in vegetation.

Initialization of the landscape model $S(0)$ is controlled by the user, who can either provide data for the different layers or rely on procedural instantiation from a single input heightfield. In the latter case, the bedrock layer $\mathcal{B}(0)$ is initialized using elevation values from the heightfield, while sand $\mathcal{C}(0)$, rocks $\mathcal{R}(0)$ and vegetation $\mathcal{V}(0)$ are set to zero. The thickness of the humus layer $\mathcal{H}(0)$ is determined by the local slope of the bedrock: $\mathcal{H}(\mathbf{p}, 0) = g \circ s(\mathcal{B}(\mathbf{p}, 0))$ where s denotes the slope of the terrain computed at \mathbf{p} . The function $g : \mathbf{R}_+ \rightarrow \mathbf{R}_+$ is a smoothly decreasing function of the slope that maps $[0, +\infty]$ onto $[0, 1]$ with $g(\tan 30^\circ) = 1/2$ so that there should be half as much humus on 30° slopes as on flat areas. In our implementation, we use a Gaussian $g(s) = e^{-s^2/\sigma^2}$ with $\sigma^2 = 3 \ln 2$. Moisture \mathcal{M} in the granular material layers \mathcal{G} is computed during rainfall simulation and need not be pre-calculated.

3.3 Control

The user is afforded pause-and-edit control over the simulation at several levels of abstraction, with changes stored in the timeline:

- *Environmental parameters* of the scene (rainfall patterns, frequency of lightning strikes, temperature, etc) can be directly and interactively adjusted at any point in the simulation timeline.
- *Layered data* can be locally edited using interactive brushes, by adding or removing plants, or adjusting the thickness of materials, at any given time step. User edits may introduce inconsistencies, such as too much sand or too many trees on steep slopes, but these usually self-correct after a few simulation time steps.
- *Probability maps* allow the user to locally re-weight the chance of any chosen geomorphological events (such as enforcing a lightning strike in a given cell).

- *Environment scenarios* provide functionality to pre-define a script for any of the above interactions (environmental parameters, layer changes, and probability maps) as they evolve over time.

At each simulation step, the user is provided with a real-time visualization of the current state, enabling stop-and-restart editing of the simulation.

4 GEOMORPHOLOGICAL EVENTS

In this section, we show how five significant and influential natural phenomena – rainfall, temperature, lightning, gravity and fire – have been incorporated into our unified framework.

4.1 Rainfall and running water

Rain, and the resulting running water, impact not only terrain shape (by hydraulic erosion) and soil composition (through material transport), but also vegetation growth. As a typical effect, water carves channels in the surface of the terrain and forms accretion cones (Figure 6). There are several factors to consider: the trajectory of runoff is heavily dependent on slope, drainage area and other features; the quantity of runoff is reduced by evapotranspiration off vegetation; and, finally plant roots bind soil and dampen hydraulic erosion.

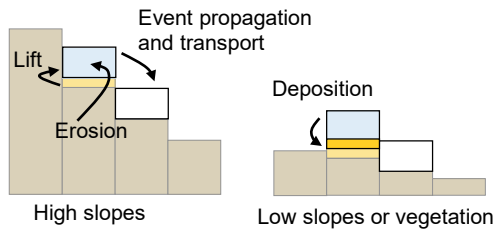


Fig. 5. Runoff events use water as a vector to transport material. They perform erosion, lift, and deposition. Water is also partly absorbed and generates soil moisture.

We simulate all these effects through *runoff events*, which model the progression of surface water: its stochastic course across the terrain, its reduction due to soil absorption, and its ultimate impact in terms of erosion and material transport.

Runoff event simulation: A runoff event is parameterized by the volume of water w it transports. The event starts at a point \mathbf{p}_0 with $w = w_0$, follows a trajectory, and ends when $w = 0$ or a local terrain minimum is reached. Initially, w_0 is set to the total quantity of rain in a cell during a simulation time step Δt , and reduced by the proportion intercepted by plants and released to the atmosphere through evaporation, which depends on plant density.

The event is then simulated iteratively over the terrain grid as follows. Let \mathbf{p}_k be the current position of the runoff. The subsequent position \mathbf{p}_{k+1} is determined by a random choice from among the neighboring cells $\mathcal{N}(\mathbf{p}_k)$ having lower elevation than \mathbf{p}_k , with probability:

$$\rho(\mathbf{p}) = s(\mathbf{p}) / \sum_{\mathbf{q} \in \mathcal{N}(\mathbf{p}_k)} s(\mathbf{q}), \quad (1)$$

Once this random choice is made, the current slope between \mathbf{p}_k and \mathbf{p}_{k+1} is stored, or set to zero if \mathbf{p}_k has no lower neighbor.

The runoff w_k is then reduced by soil absorption, which is inversely proportional to the slope. Absorbed water is added to the local moisture in cell \mathbf{p}_k , stored in layer \mathcal{M} . If this moisture exceeds the maximum holding capacity of the local soil layers (given preset parameters for bedrock, rock, sand, and humus), then the excess is returned to w_k because of soil saturation.

Interactions with the landscape: Inspired by the geology literature [Braun and Sambridge 1997], we account for both hydraulic erosion, which grinds terrain material (bedrock and rocks) to a finer constituency (rocks and sand), and soil transportation, which involves water redistributing small quantities of rock, sand and humus.

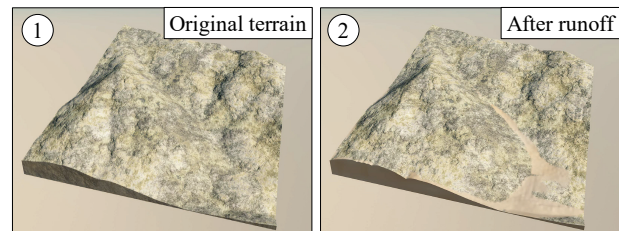


Fig. 6. Results after several runoff events: channels form in the upper part, while accretion cones develop over those already present in the original terrain. One event has been initiated per cell and per year for 200 years.

We use a standard discrete model for erosion [Musgrave et al. 1989], where redistribution acts by lifting or depositing material, depending on a slope threshold (Figure 5). Lift only occurs until carrying capacity is reached, and deposition is inversely proportional to slope. Moreover, bedrock is partially shielded from erosion by intervening soil and vegetation layers. Finally, lift is impeded and deposition enhanced by the presence of vegetation, by virtually reducing the slope according to the amount of vegetation, for the purposes of redistribution.

Once the runoff sequence terminates we approximate the effects of plant transpiration and seepage into groundwater by reducing the moisture at the source \mathbf{p}_0 by a constant amount.

4.2 Temperature

Temperature variation plays an important role in triggering the fracture of bedrock – referred to as *thermal erosion* [Musgrave et al. 1989] – as well as in determining the niche suitability of vegetation. Thermal erosion occurs when water in rock cavities freezes, expands and breaks the bedrock into rocks that may then fall under gravity. Thermal erosion is most influential in regions with high temperature gradients, such as cliffs that are exposed to direct sunlight and cold night-sky radiative transfer. Sand, humus and vegetation act to reduce the impact of temperature variations by shielding the bedrock.

In our framework, when a *thermal stress event* is launched at a random position \mathbf{p} , we first estimate the variation between daytime and nighttime temperatures denoted as ΔT using the elevation $\mathcal{E}(\mathbf{p})$ and illumination $\mathcal{I}(\mathbf{p})$. This ΔT value is then damped in proportion to the local density of vegetation $\mathcal{V}(\mathbf{p})$ and thickness of sand and

humus $\mathcal{G}(\mathbf{p})$. The resulting value is used as the probability that a given quantity of local bedrock \mathcal{B} will fracture into rocks \mathcal{R} :

$$f(\mathbf{p}) = k\Delta T s(\mathbf{p}) / (1 + k_{\mathcal{G}}\mathcal{G}(\mathbf{p}) + k_{\mathcal{V}}\mathcal{V}(\mathbf{p}))$$

The coefficients k , $k_{\mathcal{G}}$ and $k_{\mathcal{V}}$ are constants and $s(\mathbf{p})$ is the maximum local slope. Since bedrock is often approximated as a granular material over large time scales [Densmore et al. 1998], we assume that the landslide effect is limited by a critical slope. The quantity of falling rock is computed from the difference between the local slope and this critical slope. By design, rocks are created in-place, and their fall is triggered and simulated according to gravity events (Section 4.4). The effect of temperature on local vegetation will be discussed further in Section 5.

4.3 Lightning

While it is well-known that lightning destroy trees, recent research in geomorphology [Knight and Grab 2014] has demonstrated that bedrock struck by lightning is blasted into rocky material. A single strike may break down tons of bedrock and eject rocks up to several meters from the point of impact, resulting in a volume of up to $10\,000m^3$ being moved per square kilometer per 100 years.

We model this phenomenon through *lightning strike events*. At a random point of impact \mathbf{p}_0 , the probability of damage is a function of elevation and the exposed character of the location, which we evaluate using local curvature of the terrain elevation \mathcal{E} :

$$l(\mathbf{p}) = k_{\mathcal{L}} \min(1, e^{k_{l_c}(\nabla \mathcal{E}(\mathbf{p}) - k_{l_s})})$$

The coefficient $k_{\mathcal{L}}$ is the maximum probability that a lightning strike hits the cell at position \mathbf{p} , k_{l_s} is the minimum curvature for which this probability is achieved, and k_{l_c} is a scaling factor.

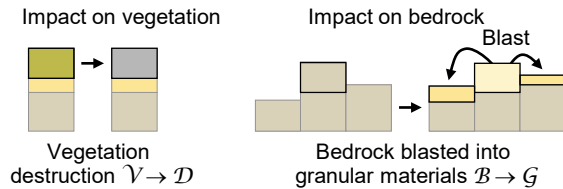


Fig. 7. Effect of a lightning strike: vegetation is destroyed and part of the bedrock layer disintegrates into rock material, which is spread to neighboring cells.

If the lightning strike occurs, we proceed as follows (Figure 7). Lightning destroys local vegetation, if present, and we reduce the number, and aggregate height and age of trees in the vegetation layer \mathcal{V} and increase the density of dead vegetation \mathcal{D} accordingly. Furthermore, in conducive conditions of high global temperature and low rainfall lightning has a chance of initiating a *fire event* at the same location (section 4.5). Otherwise, a constant amount of bedrock material \mathcal{B} is removed and spread as granular material \mathcal{G} (rocks and sand) in the neighboring cells, while taking their relative elevation into account.

4.4 Gravity

We use separate events (*rock-slide*, *sand-slide* and *humus-slide*) to represent the collapse under gravity of different granular materials, since each has a specific friction angle. As with runoff, a slide event starts at a random position \mathbf{p}_0 , propagates along a random slope-dependent trajectory (see Equation (1)), and terminates when the local slope falls below the friction angle. At each step, the amount of sliding material is computed as a random proportion of the material column that sits above the friction angle. We also add a contribution proportional to the curvature of the surface, to simulate a form of diffusion known as the hill-slope process on granular material [Braun and Sambridge 1997].

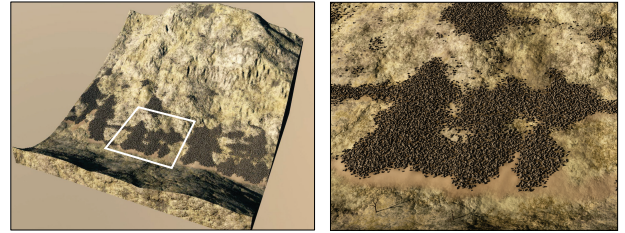


Fig. 8. The effects of thermal stress and material slides. Bedrock fractures into rocks and sand, which slide under gravity. Here, rock is dark gray and sand is yellow. Note that the friction angle for sand is set lower than for rock.

Material slide effects are dampened by vegetation since the roots of trees form a stabilizing lattice on slopes. This is captured by a vegetation-dependent increase in the friction angle. Conversely, rockslides also damage vegetation, which we model by decreasing plants in the vegetation layers \mathcal{V} and increasing the dead vegetation \mathcal{D} , accordingly: vegetation is destroyed in proportion to the volume of falling rocks.

4.5 Fire

Fire as a disturbance event is one of the primary causes of deforestation in hot and dry ecosystems. In our system, it also serves as a useful control tool for users.

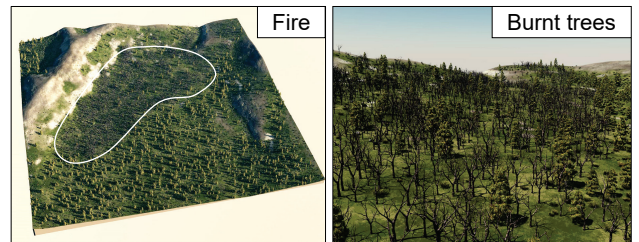


Fig. 9. A fire set by the user and advected by a north wind, with subsequent regrowth over a few years.

We note that fires spread more strongly upwards and in the prevailing wind direction [Yassemi et al. 2008]. Accordingly, a *fire event* at \mathbf{p}_0 starts with a seeding process igniting fires in neighboring cells,

whose intensity is a function of local temperature, moisture, and vegetation density. The number, distance and direction of nearby cells damaged by the fire depends on this intensity and the prevailing wind (a constant user-defined 2D vector in our implementation). In particular, the fire spread direction follows a normal distribution centered on the wind direction, with variance related to the inverse of fire intensity, narrowing when the wind is strong. Next, vegetation is destroyed at the seeded locations in proportion to fire intensity and local slope. The cycle then begins again with one of the seeded locations chosen as the potential source of a new fire event. A fire dies out when there is no more vegetation to fuel it or if the fire intensity has dropped below the level required to ignite new trees.

5 ECOSYSTEM EVENTS

In simulating vegetation, our point of departure from previous CG ecosystems is a consideration of the cyclic interaction between plants and soil. Plants rely on soil moisture but also impact it in various ways. For instance, due to evapotranspiration some rainfall is intercepted by tree canopies and by litterfall (the detritus of fallen leaves and dead plants) and evaporates before reaching the ground, while water is also drawn up from the soil and transpires through leaves. Conversely, litterfall decays into humus, which changes the constituency of soil and improves its moisture holding capacity. We explicitly account for these effects, as well as the impact of vegetation on erosion (see Section 4.1).

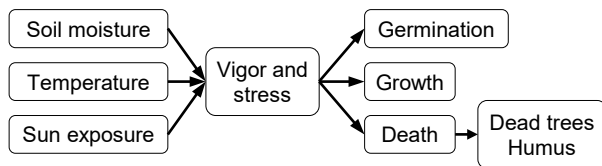


Fig. 10. Ecological events use monthly soil moisture, temperature, and sunlight exposure to derive yearly stress and vigor values, which drive plant germination, growth, and death (contributing to the dead tree and humus layers).

Most ecosystem simulations in computer graphics [Benes and Espinosa 2003; Bradbury et al. 2015; Ch’Ng 2013; Deussen et al. 1998; Lane and Prusinkiewicz 2002] model individual plant specimens using a circular footprint to determine competition for resources. While this is viable for small-scale simulations (up to 1km^2) it becomes computationally costly at larger scales due to the correlation between terrain area and numbers of plants. Instead, we adopt a cell-based Eulerian approach, as favored by botanists and ecologists in their Dynamic Global Vegetation Models (DGVM) [Foley et al. 1996; Sato et al. 2007; Sitch et al. 2003]. While plants are treated in aggregate it is still possible to incorporate competition for sunlight and soil moisture.

Another feature borrowed from the DGVM literature is our grouping of species into Plant Functional Types, representing plants with similar response to environmental conditions. A broad separation between tree, shrub and grass layers is sufficient in our case but

the architecture supports finer categories, should differentiation between, for instance, evergreen and deciduous or needle-leaved and broad-leaved plants be required for a particular application.

As with other events, *ecosystem events* are generated at random in a given cell. Each of them accumulates growth, death and germination for a particular plant layer (Figure 10). This is based on the aggregation of a monthly suitability function that considers local temperature, soil moisture and sun exposure to arrive at a value for plant viability $V \in [-1, 1]$, for all types of plants (trees, shrubs, and grass, in our implementation). Negative values represent stresses to the ecosystem in response to extreme conditions, such as drought and frost, while positive values indicate proportional opportunities for growth and germination.

The response of plants is modeled using a piece-wise linear hat-like function $v(c)$ that captures the influence of a given climatic condition c [Gain et al. 2017]:

$$v(c) = \begin{cases} -1 & \text{if } c < E_{\min}, c > E_{\max} \\ (c - E_{\min}) / (I_{\min} - E_{\min}) & \text{if } E_{\min} \leq c < I_{\min} \\ 1 & \text{if } I_{\min} \leq c \leq I_{\max} \\ (c - I_{\max}) / (E_{\max} - I_{\max}) & \text{if } I_{\max} < c \leq E_{\max} \end{cases},$$

where E_{\min}, E_{\max} are the bioclimatic limits outside of which a particular plant type cannot survive [Sitch et al. 2003], $[I_{\min}, I_{\max}]$ is the ideal range for a plant, and c is a monthly bioclimatic value for a given cell. We use a separate such function $v_i, i = \{\text{temperature, moisture, sunlight}\}$ for each combination of plant type and bioclimatic condition. Ultimately, $V = \min_i(v_i)$, since viability is constrained by the weakest resource. For instance, a plant cannot flourish if it is over- or under-watered even if temperature and sunlight conditions are ideal, a principle known in Ecology as Leibig’s law of the minimum [Cade et al. 1999].

The monthly bioclimatic inputs (c_i) are calculated as follows: soil moisture is derived by distributing a cell’s yearly moisture value (layer \mathcal{M}) in proportion to monthly rainfall patterns; temperature is provided by the user as an average monthly value (\bar{t}_i), which is reduced according to cell altitude at a lapse rate of 6.5°C per 1000 m: $\theta_i(\mathbf{p}) = \bar{\theta}_i - 0.0065 \mathcal{G}(\mathbf{p})$; sun exposure \mathcal{I} is calculated, based on latitude and compass direction, by intersecting ray’s from the sun’s position along its trajectory with the terrain. This captures terrain self-shadowing and provides average daily hours of direct sunlight per month for a cell.

Each plant layer in the cell, with the exception of grass, is encoded by the number of plants (n), sum of plant heights (h) and sum of plant ages (a). This allows an average plant specimen to be derived and hence an estimate of the density of plant coverage in a cell to be computed, as:

$$d = n\pi(r \cdot h/n)^2/w^2,$$

where w is the width of a cell and r is the ratio between a plant’s canopy radius and height.

In addition to its effect on the various geomorphological events in Section 4, the density of plants is useful to the ecosystem simulation itself in two respects. First, it provides a proxy for competitive pressure within plants of the same functional type. For instance, if $d > 1$ there is more than complete canopy coverage in a cell and self-thinning is mandated due to plants encroaching on each other. Second, it can be used to account for the shading of subordinate

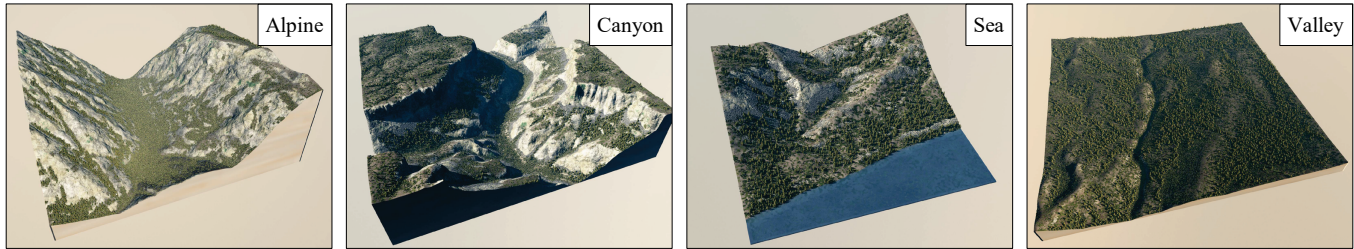


Fig. 11. Results from different initial heightfield inputs produced using our simulation after 300 years of evolution. From left to right: an Alpine landscape from the U.S. Rockies, a portion of the Grand Canyon, a Mediterranean landscape, and a forested valley.

plants, such as shrubs shaded by trees, by reducing sun exposure in proportion to density.

Since the expected interval between vegetation events in a given cell is one year (based on Δt), we use the monthly viability to derive yearly values for vigor and stress and feed these into simple growth, germination and death processes. Vigor is the average viability during the growing season (when the average monthly temperature is at least 5°C), while stress is the average of the four worst viability values, but is only considered if it is negative.

For seeding and germination, the event framework could support propagation to neighboring cells, in a similar fashion to fractured rock ejected by a lightning strike (section 4.3). However, propagation processes are complicated by different forms of seed dispersal and variable delays in germination. For instance, in the case of obligate seeding forest fires are needed to spur germination. Instead, we make the simplifying assumption that an existing bank of seeds is present in the soil and model germination by adding a number of plants as seedlings to the cell in proportion to the vigor and available space ($1 - d$), but only if there is no stress. For woody plant types we use an establishment rate of 0.24 saplings per m^2 [Prentice et al. 1993]. For growth, we scale a constant yearly increase in plant height for each plant according to the vigor value. In the last process, plant death acts by removing average plant specimens from a cell into a litter pool according to self thinning, environmental stress and senescence rules. First, to avoid over-saturation plants are removed to enforce $d < 1$. Next, bioclimatic limits are accounted for by removing plants in proportion to the stress value. Finally, if the average age is above a threshold a small random proportion of plants are removed to represent death by old age.

It makes little sense to consider grass as individual plants, even in aggregate. So, instead we translate grass vigor directly into a density value and ignore issues of individual grass clump growth and death.

The final phase in an ecosystem event is to model the breakdown of dead plant matter \mathcal{D} into humus \mathcal{H} . [Higgins et al. 2007] provides a conversion from tree height to biomass, which when combined with an average green wood weight (m in kg / m^3) allows a derivation of bio-volume (b) from average plant height (\bar{h}), as: $b = 0.52 \bar{h}^{2.55} / m$

Roughly 30% of litter fall is converted to humus in a given year [Sitch et al. 2003], while the remainder rots away and is released as carbon dioxide.

6 IMPLEMENTATION

Our system has been implemented in C++ and uses OpenGL for rapid previsualization and *Vue* for rendering photorealistic landscape images. All simulations were performed on a desktop computer equipped with an Intel Core i7 CPU clocked at 2.5 GHz. We did not use any graphics hardware acceleration.

We use elevation maps from real terrains downloaded from the (U.S. Geological Survey). The output can be either a single, static landscape, or several frames representing its evolution over time. In both cases we use our layered model to enhance the landscape with procedural detail, as follows: we provide a terrain heightfield, obtained by stacking the sand \mathcal{C} and humus \mathcal{H} layers on top of the bedrock elevation \mathcal{B} , and a surface texture, computed from the thickness of sand and humus and the density of grass \mathcal{V}_g . Plant geometry is instantiated from models stored in an atlas, in accordance with the vegetation layers \mathcal{V} , which define local plant data such as density, size, and age, for every cell. After scene geometry has been generated, it is exported to *Vue* for final rendering. The final rendering smooths the layers, and bedrock, sand, humus, and grass are rendered in order from the topmost layer to the bottommost. All geometric elements (rocks, shrubs, and trees, both living and dead) are instantiated.

To explore the possibility of acceleration, we have implemented a multi-threaded CPU version of our framework, where each thread handles a set of events. We solve race conditions, where two events converge on the same cell, using atomic instructions. Several cells can be locked iteratively during the propagation of an event, but, to avoid deadlock, the same thread cannot lock more than one cell at the same time. With this approach we achieve a speed-up factor of 4 on a 12-thread machine.

7 RESULTS AND DISCUSSION

The size of a cell in all our simulations is set to $10 \times 10 m^2$. This allows the capture of medium-scale erosion and ecosystem features, while allowing the simulation and authoring of landscapes up to $10 \times 10 km^2$ in extent at interactive rates.

Average timings for different types of events and for a complete scene are reported in Table 1. Performance is related to the scene resolution $n \times n$, the number of events and their overall complexity. Certain events have only a local extent, such as lightning strikes

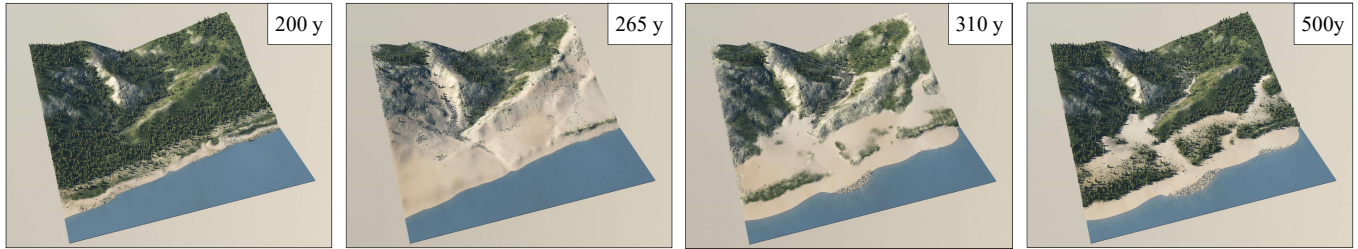


Fig. 12. Evolution of the simulation in response to user edits. Initial state: the user paints sand along the beachfront and humus elsewhere. After vegetation growth at 200y the user layers sand across the lower half of the terrain, reduces precipitation, and destroys vegetation with fire. At 265y he increases rainfall. Finally, at 310y he adds humus to promote forest regrowth and waits until 500y.

and ecosystem events. In contrast, events such as gravity (Section 4.4) and rain (Section 4.1), with hydraulic erosion, may propagate changes in the layered data-structure across many cells.

Event type	$n = 128$	$n = 256$	$n = 512$	$n = 1024$
Rain	0.0695	0.43	3.92	36.5
Gravity	0.0067	0.027	0.08	0.43
Temperature	0.0016	0.0067	0.027	0.12
Lightning	0.0007	0.0027	0.01	0.05
Ecosystem	0.027	0.085	0.36	0.94
Total time	0.11	0.59	4.77	38

Table 1. Average performance (in s) for different events over the course of a simulation step Δt , and the total, as a function of terrain width n (in number of cells).

The worst case complexity is $O(n^4)$, where n is the terrain width, because the number of events per simulation step is proportional to n^2 , and of these the most computationally demanding events trace a path with no cycles, spawning an upper bound of n^2 cells. In practice, events follow a discrete curve on the terrain, so the impact is limited to $(n^2)^{1/2} = n$ cells on average.

We have found that execution is dominated by the runoff event, which exhibits average complexity of $O(n)$ for every cell and represents 75% of the computation for a $2.5 \times 2.5 \text{ km}^2$ landscape. The overall average complexity varies from $O(n^2)$ for terrains with limited runoff to $O(n^3)$ for terrains with long channels.

User Control: A variety of control mechanisms and achievable landscapes are showcased in Figures 2, 11, and 12. The first result in Figure 2 shows a landscape designed by triggering specific landslide and fire events, followed by periods of natural landscape evolution. Figure 11 demonstrates varied outcomes obtained simply by changing the initial 256×256 bedrock heightfield. An example of interactive control during a 15 minute editing session is provided in Figure 12. Here, the user has modeled drought and sand-dune growth over a 128×128 subregion of an oceanic landscape.

Generic framework: The logical subdivision into independent tasks makes our framework flexible. It simplifies the main simulation loop

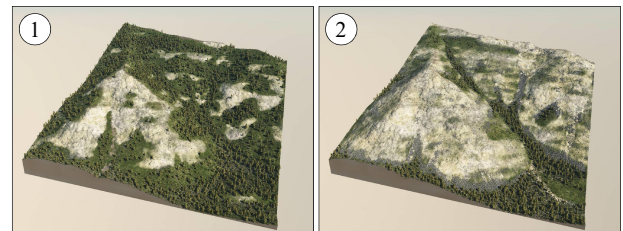


Fig. 13. Impact of vegetation on soil erosion: vegetation protects the soil, damps erosion, and blocks rockfalls. (1) For 150 years the landscape has vegetation growth without erosion. (2) Conversely, over the same time span erosion affects the second terrain but without any vegetation. Finally, vegetation and erosion are combined for a further 150 years in both landscapes and the outcome shown in (1) and (2). The visual impact of erosion is highlighted in (2) by the sparser vegetation: the erosion removed soil, carved steep slopes and triggered rock falls that prevented vegetation growth.

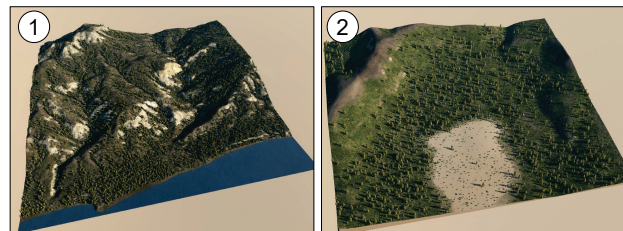


Fig. 14. Effect of soil type on vegetation: (1) hilltops receive less moisture, (2) user-painted sand slows vegetation growth.

(Section 3), making it both scalable and extensible: it is easy to add events to the simulation because each event is a function that takes as arguments a starting position, the terrain layers, and a few other simulation parameters, such as the main wind direction and strength. The relative importance of each event type can be easily adjusted to obtain a wide variety of landscapes (Figure 15).

Simulation quality: Our framework enhances terrain erosion by interlinking multiple geomorphological phenomena, including hydraulic erosion, material transport, shifts and slides due to gravity, and bedrock fracture from thermal erosion and lightning.

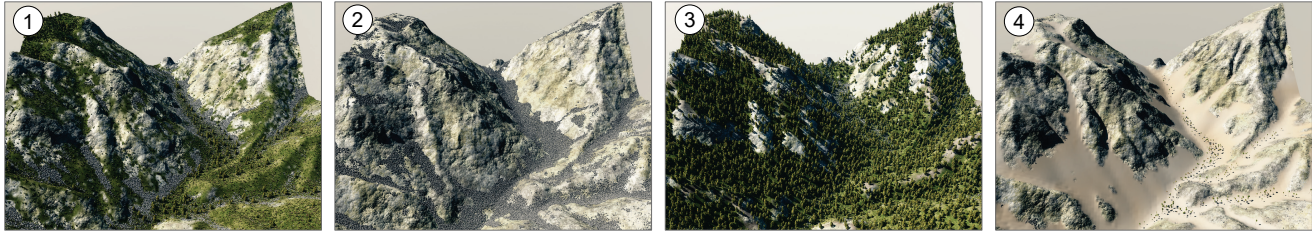


Fig. 15. A terrain used to generate a variety of landscapes: (1) with default parameters, (2) a high altitude rocky region, (3) a dense forest on a fertile ground, and (4) a sand-filled desert with drought-adapted vegetation at the lower elevations.

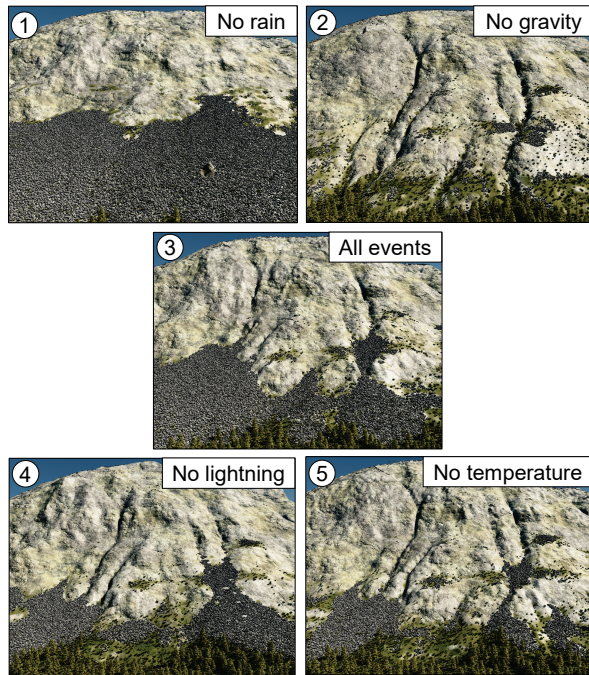


Fig. 16. Disabling the events with respect to the complete simulation rendered in (3) shows their impact: (1) rain, (2) gravity, (4) lightning, and (5) temperature.

The results of erosion of a typical simulation (100–1,000 years) are subtle but distinctive; the final variation of height is on the order of a few meters. The visual consequences are mainly:

- Bedrock destruction evidenced by small channels and fractured cliff-faces (Figure 18.(3)).
- Correct buildup of materials (fallen rocks, sand, and humus) in both screes and accretion areas (Figure 18.(2)).
- That vegetation suppresses erosion and therefore softens slopes. This becomes visually salient when slopes are denuded by fire (Figure 13). However, the effects of vegetation on erosion are subtle in most scenes.

Although each event type in isolation has little impact on a terrain, their significance is heightened by combined interaction. Compared to a combined simulation (Figure 16), an absence of erosion and

sedimentation linked to rain prevents rock from being carved and soil from being transported into channels. Without gravity there are no rockslides to fill lower erosion pockets. A lack of lightning and temperature simulation prevents the erosion of exposed and steep slopes, respectively. The importance of fire in hot, dry landscapes is depicted in Figure 17.



Fig. 17. Effect of fires on a warm, dry landscape: no fire for 200 years (left), compared with regular fires every other year (right).

In terms of ecosystem simulation, our terrain erosion models capture visual outcomes not considered in previous work. First, our rainfall model accounts for moisture absorbed by different soil layers with differing moisture retention and these layers are distributed by erosion forces in a consistent fashion. For example, in Figure 14.(1) hilltops receive less moisture and thus less plant cover. Most importantly, our model considers humus generation, displacement, and exploitation by vegetation. For example, in Figure 14.(2) a pile of sand painted by the user to override humus retards vegetation growth. Second, erosion destroys plants through catastrophic events. For instance, rockfalls crush and bury trees, an effect that is characteristic of alpine forests.

Real terrain data with different layers (humus, rock) is not readily available, making comparison difficult. This is exacerbated by temporal simulations that run anywhere from 100 to 1,000 years. Instead, we validate our results by comparing them with real phenomena illustrated by photographs. Figure 19(1) shows the destruction of vegetation by rock-slides, which we reproduce in Figure 18(1). Figure 19(2) has trees growing over accretion areas, and is mimicked in Figure 18(2). Finally, Figure 19(3) is a granular terrain, deeply eroded due to an absence of vegetation, which is simulated in Figure 20 by replacing bedrock with only granular materials.

Limitations: One limitation of our framework is the computational cost of event simulation. Unfortunately, parallelizing the

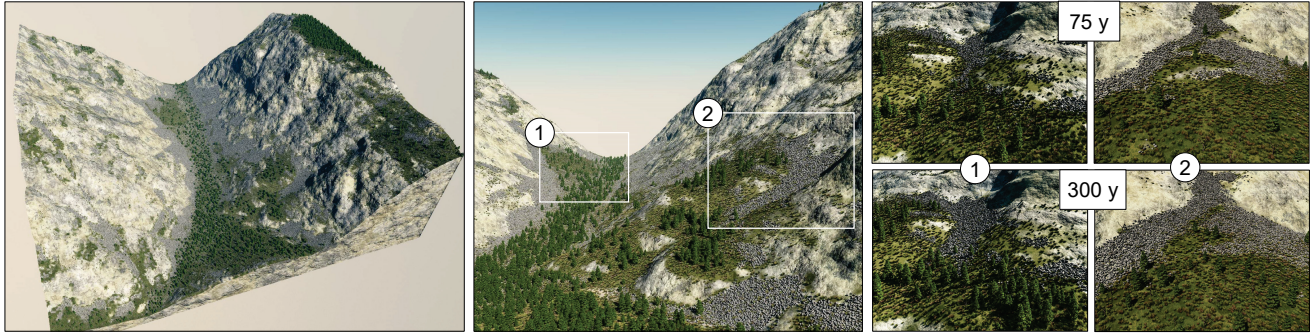


Fig. 18. An example of combined erosion and ecosystem simulation. Images show in detail: (1) vegetation destroyed by falling rock and (2) plant growth on accretion zones, after 75 years and 150 years.

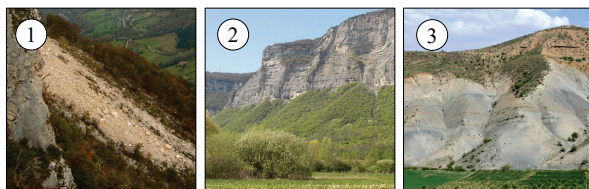


Fig. 19. Photographs of real-world landscapes show: (1) vegetation cleared by an avalanche of rocks, (2) vegetation growing on accretion cones, (3) Terrain erosion accelerated by an absence of plants (© Creative Commons).

simulation is non-trivial, because events occur in sequence and may overlap spatially. In the interests of responsiveness, we limit users to interacting with coarser-scale landscapes ($n = 128$ to 256). One way to compensate for this is to store the events and user modifications and re-run them with an off-line upsampled simulation. Having many combining events increases both the number of simulation parameters and the complexity of their interdependence. These parameters tend to be hard to tune and are not artist-friendly.

Another limitation is the discrete nature of the simulator. While a discretized grid is more efficient, a continuous domain would be more appropriate for phenomena that act at different scales. In the same vein, we do not account for accurate fine-scale detail, so the method is less suited to close-up views. In future work, detailed geometry such as piles of rocks and boulders or fallen branches could be added using mass instancing [Guérin et al. 2016].

Validation is a challenge common to all but the simplest simulation methods. While we included real images for comparison, it is difficult to quantify how closely the results match corresponding effects in nature. Our evaluation is only visual.

8 CONCLUSION

We have presented a novel landscape editing framework, which, for the first time, enables the simulation of complex interactions between a variety of phenomena, ranging from vegetation lifecycles and terrain erosion to lightning and fire disturbance. This is achieved with a layered landscape model, which stores terrain materials, vegetation densities, and other resources, and is evolved over time



Fig. 20. Shielding effects of vegetation: a time lapse of simulations on slopes with granular materials, (left) without vegetation to damp soil erosion, and (right) protected by vegetation.

through a large number of stochastic events. We avoid a complex, joint simulation over all phenomena by launching these events in random order and restricting them to interact only through the landscape layers. This reduces computation time, simplifies the interactions between different phenomena, and enables interactive user control. Our editing tools enable the user to over-paint any of the landscape layers at runtime, throughout the simulation, thereby combining realism and control.

Future work could include adding other events to broaden the range of simulated phenomena to cover wind (affected by vegetation

and topography), sand dunes, grazing herds, and different rock, soil and plant types. It would also be interesting to incorporate an extra layer for bodies of water, such as large rivers and lakes, updated using shallow water fluid simulation, or a snow layer affected by sunlight and avalanches. One important open problem is an evaluation mechanism for further comparing the proposed model against real landscapes.

ACKNOWLEDGMENTS

This work is supported by the advanced grant #291184 EXPRESSIVE from the European Research Council (ERC-2011-ADG_20110209), the project PAPAYA P110720-2659260, funded by the Fonds National pour la Société Numérique, the project HWD ANR-16-CE33-0001, the NSF grant CDS&E 1608762, and NSF EEC-1606396.

REFERENCES

- Bedrich Benes, Michel Abdul-Massih, Philip Jarvis, Daniel G. Aliaga, and Carlos A. Vanegas. 2011. Urban ecosystem design. In *Proc. of 3D*. New York, NY, USA, 167–174.
- Bedrich Benes, Nathan Andryscio, and Ondřej Štáva. 2009. Interactive Modeling of Virtual Ecosystems. In *Proceedings of Eurographics Workshop on Natural Phenomena*. 9–16.
- Bedrich Benes and Enrique David Espinosa. 2003. Modeling virtual ecosystems with the proactive guidance of agents. In *Proc. of CASA*. 126–131.
- Bedrich Benes and Rafael Forsbach. 2001. Layered Data Representation for Visual Simulation of Terrain Erosion. In *Proc. of SCCG*, Vol. 25(4). IEEE Computer Society, 80–86.
- Bedrich Benes and Rafael Forsbach. 2002. Visual Simulation of Hydraulic Erosion. In *Proc. of the WSCG*. 120–132.
- Bedrich Benes, Václav Těšínský, Jan Hornýš, and Sanjiv K. Bhatia. 2006. Hydraulic erosion. *Computer Animation and Virtual Worlds* 17, 2 (2006), 99–108.
- Jules Bloomenthal. 1985. Modeling the mighty maple. *Proc. of Siggraph* 19 (1985), 305–311.
- Gwyneth A Bradbury, Kartic Subr, Charalampos Koniari, Kenny Mitchell, and Tim Weirich. 2015. Guided Ecological Simulation for Artistic Editing of Plant Distributions in Natural Scenes. *Journal of Computer Graphics Techniques* 4, 4 (Nov. 2015), 28–53.
- Jean Braun and Malcolm Sambridge. 1997. Modelling landscape evolution on geological time scales: a new method based on irregular spatial discretization. *Basin Research* 9, 1 (1997), 27–52.
- Brian Cade, James Terrell, and Richard Schroeder. 1999. Estimating effects of limiting factors with regression quantiles. *Ecology* 80, 1 (1999), 311–323.
- Norishige Chiba, Kazunobu Muraoka, and Kunihiko Fujita. 1998. An Erosion Model Based on Velocity Fields for the Visual Simulation of Mountain Scenery. *The Journal of Visualization and Computer Animation* 9 (1998), 185–194.
- Eugene Ch'Ng. 2013. Model resolution in complex systems simulation: Agent preferences, behavior, dynamics and n-tiered networks. *Simulation* 89, 5 (May 2013), 635–639.
- Guillaume Cordonnier, Jean Braun, Marie-Paule Cani, Bedrich Benes, Eric Galin, Adrien Peytavie, and Guérin Eric. 2016. Large Scale Terrain Generation from Tectonic Uplift and Fluvial Erosion. *Comp. Graph. Forum* 35, 2 (2016), 165–175.
- Alexander L Densmore, Michael A Ellis, and Robert S Anderson. 1998. Landsliding and the evolution of normal-fault-bounded mountains. *Journal of geophysical research: solid earth* 103, B7 (1998), 15203–15219.
- Oliver Deussen, Pat Hanrahan, Bernd Lintermann, Radomír Měch, Matt Pharr, and Przemysław Prusinkiewicz. 1998. Realistic Modeling and Rendering of Plant Ecosystems. In *Proc. of Siggraph (SIGGRAPH '98)*. ACM, 275–286.
- Oliver Deussen and Bernd Lintermann. 2006. *Digital design of nature: computer generated plants and organics*. Springer Science & Business Media.
- Arnaud Emilien, Ulysse Vimont, Marie-Paule Cani, Pierre Poulin, and Bedrich Benes. 2015. WorldBrush: Interactive Example-based Synthesis of Procedural Virtual Worlds. *ACM Trans. Graph.* (2015).
- Jonathan A. Foley, I. Colin Prentice, Navin Ramankutty, Samuel Levis, David Pollard, Steven Sitch, and Alex Haxeltine. 1996. An integrated biosphere model of land surface processes, terrestrial carbon balance, and vegetation dynamics. *Global Biogeochemical Cycles* 10, 4 (1996), 603–628.
- Alain Fournier, Don Fussell, and Loren Carpenter. 1982. Computer rendering of stochastic models. *Commun. ACM* 25, 6 (1982), 371–384.
- James Gain, Harry Long, Guillaume Cordonnier, and Marie-Paule Cani. 2017. EcoBrush: Interactive Control of Visually Consistent Large-Scale Ecosystems. *Comp. Graph. Forum* 36, 2 (2017), 105–116.
- James Gain, Bruce Merry, and Patrick Marais. 2015. Parallel, Realistic and Controllable Terrain Synthesis. *Comp. Graph. Forum* 34, 2 (2015), 105–116.
- Jean-David Gènevaux, Eric Galin, Eric Guérin, Adrien Peytavie, and Bedrich Benes. 2013. Terrain Generation Using Procedural Models Based on Hydrology. *ACM Trans. Graph.* 32, 4, Article 143 (2013), 13 pages.
- Jean-David Gènevaux, Eric Galin, Adrien Peytavie, Eric Guérin, Cyril Briquet, François Grosbellet, and Bedrich Benes. 2015. Terrain Modelling from Feature Primitives. *Comp. Graph. Forum* 34, 6 (2015), 198–210.
- Sabine Grunwald. 2016. *Environmental soil-landscape modeling: Geographic information technologies and pedometrics*. CRC Press.
- Eric Guérin, Julie Digne, Adrien Peytavie, and Eric Galin. 2016. Sparse representation of terrains for procedural modeling. *Comp. Graph. Forum* 35, 2 (2016), 177–187.
- Eric Guérin, Eric Galin, François Grosbellet, Adrien Peytavie, and Jean-David Gènevaux. 2016. Efficient modeling of entangled details for natural scenes. *Comp. Graph. Forum* 35, 7 (2016).
- Steven I Higgins, William J Bond, Edmund C February, Andries Bronn, Douglas IW Euston-Brown, Beukes Enslin, Navashni Govender, Louise Rademan, Sean O'Regan, Andre LF Potgieter, and others. 2007. Effects of four decades of fire manipulation on woody vegetation structure in savanna. *Ecology* 88, 5 (2007), 1119–1125.
- Alex D. Kelley, Michael C. Malin, and Gregory M. Nielson. 1988. Terrain simulation using a model of stream erosion. *ACM Trans. on Graph.* (1988), 263–268.
- Jasper Knight and Stefan Grab. 2014. Lightning as a geomorphic agent on mountain summits: Evidence from southern Africa. *Geomorphology* 204 (2014), 61–70.
- Peter Kristof, Bedrich Benes, Jaroslav Krivánek, and Ondřej Štáva. 2009. Hydraulic Erosion Using Smoothed Particle Hydrodynamics. *Computer Graphics Forum* 28, 2 (2009).
- Brendan Lane and Przemysław Prusinkiewicz. 2002. Generating spatial distributions for multilevel models of plant communities. In *Proc. of Graphics Interface*. 69–80.
- Aristid Lindenmayer. 1968. Mathematical models for cellular interaction in development. *Journal of Theoretical Biology Parts I and II*, 18 (1968), 280–315.
- Benoit B Mandelbrot and Roberto Pignoni. 1983. *The Fractal Geometry of Nature*. W.H. Freeman and Company, San Francisco.
- Xing Mei, Philippe Decaudin, and Bao-Gang Hu. 2007. Fast Hydraulic Erosion Simulation and Visualization on GPU. In *Proc. of Pacific Graphics*.
- F Kenton Musgrave, Craig E Kolb, and Robert S Mace. 1989. The synthesis and rendering of eroded fractal terrains. *Proc. of Siggraph* 23, 3 (1989), 41–50.
- Mattia Natali, EM Lidal, J Parulek, I Viola, and D Patel. 2013. Modeling terrains and subsurface geology. *Proc. of Eurogr. State of the Art Reports* (2013), 155–173.
- Benjamin Neidhold, Markus Wacker, and Oliver Deussen. 2005. Interactive physically based Fluid and Erosion Simulation. In *Proceedings of the First Eurographics Conference on Natural Phenomena*. 25–33.
- Adrien Peytavie, Eric Galin, Jérôme Grosjean, and Stéphane Mérillou. 2009. Arches: a Framework for Modeling Complex Terrains. *Comp. Graph. Forum* 28, 2 (2009), 457–467.
- I Colin Prentice, Martin T Sykes, and Wolfgang Cramer. 1993. A simulation model for the transient effects of climate change on forest landscapes. *Ecological modelling* 65, 1 (1993), 51–70.
- Przemysław Prusinkiewicz and Mark Hammel. 1993. A Fractal Model of Mountains with Rivers. In *Proc. of Graphics Interface*, Vol. 30(4). 174–180.
- Przemysław Prusinkiewicz and Aristid Lindenmayer. 1990. *The Algorithmic Beauty of Plants*. Springer-Verlag, New York.
- Hisashi Sato, Akihiko Itoh, and Takashi Kohyama. 2007. SEIB-DGVM: A new Dynamic Global Vegetation Model using a spatially explicit individual-based approach. *Ecological Modelling* 200, 3–4 (2007), 279–307.
- Stephen Sitch, Benjamin Smith, I Colin Prentice, Almut Arneth, A Bondeau, W Cramer, JO Kaplan, Samuel Levis, W Lucht, M Thonicke Sykes, and others. 2003. Evaluation of ecosystem dynamics, plant geography and terrestrial carbon cycling in the LPJ dynamic global vegetation model. *Global Change Biology* 9, 2 (2003), 161–185.
- Ruben M. Smelik, Tim Tuteneel, Rafael Bidarra, and Bedrich Benes. 2014. A Survey on Procedural Modelling for Virtual Worlds. *Comp. Graph. Forum* 33, 6 (2014), 31–50.
- Szymon Stachniak and Wolfgang Stuerzlinger. 2005. An Algorithm for Automated Fractal Terrain Deformation. In *Proceedings of Computer Graphics and Artificial Intelligence*, Vol. 1. 64–76.
- Ondřej Štáva, Bedrich Benes, Matthew Brisbin, and Jaroslav Krivánek. 2008. Interactive terrain modeling using hydraulic erosion. In *Proceedings of the SCA Eurographics Association*, 201–210.
- Juraj Vaneek, Bedrich Benes, Adam Herout, and Ondrej Stava. 2011. Large-Scale Physics-Based Terrain Editing Using Adaptive Tiles on the GPU. *Comp. Graph. and App., IEEE* 31, 6 (2011), 35–44.
- Richard F Voss. 1985. Random fractal forgeries. In *Fundamental algorithms for computer graphics*. Springer, 805–835.
- Shahram Yassemi, Suzana Dragičević, and Margaret Schmidt. 2008. Design and implementation of an integrated GIS-based cellular automata model to characterize forest fire behaviour. *ecological modelling* 210, 1 (2008), 71–84.
- Howard Zhou, Jie Sun, Greg Turk, and James M Rehg. 2007. Terrain synthesis from digital elevation models. *IEEE Trans. on Vis. and Comp. Graph.* 13, 4 (2007), 834–848.