

1 Introduction

Terrain plays a fundamental part in many virtual scenes across a broad range of applications, including games, films, training, and simulation.

For example, open-world computer games often include large-scale natural environments for players to explore, films sometimes require scenic locations that either do not exist or are difficult or dangerous to film, particularly in the science fiction and fantasy genres, and training applications, such as combat, driving and flight simulators, rely on realistic environments to be effective. In such applications a bare earth terrain is usually the first part of the authoring pipeline to be subsequently augmented with props that represent rocks, trees, plants and buildings.

The shape of real-world terrains is complex and varied. Diverse landforms ranging from featureless plains to mountain ranges crumpled by tectonics and scored by erosion can co-exist within a single scene. Terrain formation is a combination of long-term influences, such as gradual erosive forces, and occasional catastrophic events, such as landslides and lightning strikes on bedrock. Furthermore, specific features appear at different scales, spanning tens to hundreds of kilometers in the case of tectonics and glaciation, whereas only evident over a few meters for certain forms of hydraulic and aeolian erosion. It is no wonder then that, despite more than four decades of research, there remain many unsolved challenges in terrain modeling.

A particular challenge is that Computer Graphics applications invariably require iterative authoring of the final terrain shape. Real-world terrains emerge from complex geomorphological processes for which simulation seems like a natural fit. Unfortunately, a direct simulation with only boundary and initial conditions is often at odds with user intent and rarely suffices.

In this paper, we present an overview and critical comparison of terrain generation methods in Computer Graphics. We begin with a consideration of the different terrain representations (including elevation and volumetric models), followed by coverage of terrain generation under three broad headings: procedural generation methods, which rely on algorithms (noise, faulting, subdivision, and the like) that do not directly emulate erosive processes; simulation techniques, which iteratively apply computer simulations of geomorphological processes, and example-based methods, which extract and combine scanned data of real-world terrains.

Finally, we compare existing methods on the basis of a number of practical considerations: the range of different landforms that can be represented (variety); the perceptual or geomorphological accuracy with which they are reproduced (realism); specific limits on feature extent and detail (scale); the types of user control available (authoring), and memory and computation overheads (efficiency). If a generated terrain is used in an application where rendering speed is important, there is a choice among several level-of-detail methods to improve performance, as described in the survey of Peytavie and Galin, 2007. These criteria serve two intended purposes: first, to help developers select a terrain modeling method that best fits their particular application.

For instance, efficiency is vital for terrains generated on the fly (as opposed to pre-designed), while authoring is secondary, since there is no opportunity for direct designer involvement. Another consideration is whether the terrain is to be modeled entirely from scratch, dropped into a cut-out region in a broader terrain, or upsampled from low-resolution input. The most appropriate technique will depend in part on such requirements.

Second, these criteria could be used to guide future research by identifying failings exhibited by current approaches. For ease of reference, Tables 2 and 3 provide a comparison of terrain modeling methods on the basis of achievable landforms variety, underlying method and model, and the supported authoring, scale, and efficiency.

2 Terrain representation

In this section, we present an overview of the underlying models used to represent and synthesize terrains. We focus on definitions of the terrain surface, and not on volumetric representations of the underlying geology. An overview of models for representing terrains and their subsurface geology can be found in Nardi et al., 2013 and a survey of cellular data representations for the entire globe (the digital Earth) in MAAS15.

2.1 Elevation models

The terrain elevation can be defined as a function $h : \mathbb{R}^2 \rightarrow \mathbb{R}$, which is at least C^0 and represents the altitude at any point in \mathbb{R}^2 . Such a definition restricts modeling to terrains without overhangs, arches, or caves. Let T denote the terrain defined over a domain $\Omega \subset \mathbb{R}^2$. The surface area of Ω , expressed in m^2 or km^2 depending on the size of the domain, will be referred to as the extent. In general, and unless stated otherwise, Ω will be a rectangular domain $R(a, b)$ where a and b are points in the plane representing opposite corners of Ω . Let $h : \mathbb{R}^2 \rightarrow \mathbb{R}$ define the elevation of T . The slope is defined as the norm of the gradient:

$$s(p) = \|\nabla h(p)\|$$

The normal at a given point on the terrain is derived from the gradient of the elevation function ∇h by projecting it to \mathbb{R}^3 as $(-\nabla h(p), 1)$, and then normalizing it:

$$n(p) = \frac{(-\nabla h(p), 1)}{\|(\nabla h(p), 1)\|}$$

2.1.1 Function representation

The elevation can be represented by a procedural or closed-form expression of the function h . While this representation is compact in terms of memory and has infinite precision, the evaluation of $h(p)$ at a given point may be computationally demanding. While in theory any suitable function h can be used for representing elevation, the construction of a function for representing large terrains with realistic landforms and characteristics is a complex task.

2.1.2 Discrete heightfields

Discrete heightfields defined over regular grids are undoubtedly the most common representation for terrains. Heightfields consist of a collection of altitudes arranged on a regular 2D grid. They can be captured by remote sensing and are often referred to as Digital Elevation Models (DEMs). A regular square grid is defined by its rectangular region $R(a, b) \subset \mathbb{R}^2$ with n subdivisions. Without loss of generality, we consider square grids of side length a with $(n + 1)^2$ elevation samples. In this representation, only the altitudes z_{ij} are stored at the vertices of the grid $p_{ij} = a + (b - a) \cdot \left(\frac{i}{n}, \frac{j}{n}\right)$, with $(i, j) \in [0, n]^2$ and where \cdot denotes the pairwise product of elements.

The accuracy or precision of the heightfield in the domain Ω is limited to $\frac{a}{n}$ and a continuous surface for the terrain needs to be reconstructed by interpolation of the elevation of grid points z_{ij} . A simple way to reconstruct the surface is to use two linear interpolations for the two triangles

inside a cell, which matches the triangle polygonization of the surface of the terrain. Bi-linear interpolation produces elevation of class C^1 inside the cells, but is only C^0 at their borders. Higher order interpolations, e.g. bi-cubic, define $C^k, k \geq 1$ continuous functions, but may generate elevations outside the range of the inputs z_{ij} . They are more computationally demanding, and require retrieving the values z_{ij} on a larger neighborhood to compute the elevation $h(p)$ of a point p inside a grid cell.

In contrast to functions, this data-oriented representation supports different forms of elevated terrain at the expense of increased memory overheads, as it requires $O(n^2)$ storage. Reducing the accuracy of elevation data by quantization (in general using 8-bit or 16-bit) is one means of reducing memory costs. However, 8-bit quantization does not provide sufficient accuracy for procedural generation and simulation algorithms, and introduces visible step patterns when displayed. Heightfields are the most common data representation for digital terrains in the industry. They are used in many applications such as GIS, authoring tools, erosion simulations and video games. In the film industry, they are also used as bare terrains and usually completed with representations that allow true 3D features. Because heightfields are represented as grids, they also lend themselves to texture synthesis and machine learning methods.

2.1.3 Layered representations

Layered representations encode the different material layers of the terrain as a collection of ordered functions describing thicknesses in a pre-established layer layout. Material layers were proposed in Musgrave et al., 1989a as a way of modeling different sediments, and developed in Cordonnier et al., 2017; Musgrave et al., 1989b to represent different layers of granular materials, such as sand or rocks on top of the bedrock. The layered data representation has been widely used in erosion simulation ever since. Layers can be represented as a set of functions, where the bottom layer represents the elevation of bedrock h_B , and each subsequent layer represents the thickness of other materials, including rocks h_R , sand h_S , or water h_W .

A discrete layer representation consists of an ordered collection of discrete grids and the data structure reduces to a stack of heightfields. This data-structure is a compromise between single-layer heightfields and voxels, for which every layer is further regularly divided. Moreover, the vertical resolution of layered heightfields is potentially infinite, whereas voxels are grid limited. Alternatively, the ordering of the layers can be explicitly defined at every grid position, at the expense of more complex dynamic stack management of the different materials.

2.2 Volumetric models

Elevation alone cannot manifest internal structure. Instead, volumetric representations must be used to capture the folds and faults of different geological strata allowing terrains with overhangs, caves, and arches.

Volumetric models are defined by a function $\mu : \mathbb{R}^3 \rightarrow M$, where $M \subset \mathbb{N}$ denotes the index of the material at any point in space. Single material models use only one type of solid material, i.e. $M = \{0, 1\}$, where 0 refers to air and 1 to bedrock.

2.2.1 Function representations

Function representations define the terrain as an implicit surface extracted from a field function f and defined as:

$$S = \{p \in \mathbb{R}^3 \mid f(p) = 0\}$$

A given elevation function h can be easily converted to a corresponding implicit representation by defining: $f(p) = h(p_{xy}) - p_z$, where p_{xy} denotes the projection of the point p onto the plane, and p_z denotes its elevation. The material function μ can be derived from the field function f as $\mu(p) = 1$ if $f(p) > 0$, i.e. inside the bedrock, and $\mu(p) = 0$, otherwise. The field function $f : \mathbb{R}^3 \rightarrow \mathbb{R}$ should be at least of class C^0 , although functions with higher continuity are often preferred since then implicit surface algorithms that require C^1 can be applied. Although this representation can, in theory, represent various complex landforms with concavities, it is seldom used in practice likely because of the complexity of authoring implicit surfaces and additional costs incurred in reconstructing the surface of the terrain through polygonization.

2.2.2 Voxels

Voxels offer a way to describe volumetric terrains, but at a high memory cost because of their explicit spatial enumeration. Space is partitioned into a 3D regular grid, and each cell is assigned a material index. The data structure can be optimized using compression techniques such as a Sparse Voxel Octrees Laine and Karras, 2011 in order to reduce the memory cost. Further compression can be achieved by generalizing the tree structure to a directed acyclic graph Kähler et al., 2013 and introducing symmetry transforms Vasa et al., 2016.

Voxels are often used for representing terrains involving complex simulation processes, such as erosion or for representing complex and detailed landforms such as arches and caves. Although the material function $\mu(p)$ can be evaluated efficiently, in addition to the disadvantage of their memory cost, their discrete nature does not lend itself to modeling continuous features such as gentle slopes.

2.2.3 Hybrid representations

Hybrid representations are inspired by layered-material representations, and exploit vertical run-length encoding of the different layer stacks to compress layers into intervals of constant material. The uppermost terrain is defined as a continuous implicit surface whose field function f is derived from the discrete layer stacks μ by computing a convolution, denoted by $*$, between a characteristic function of the layer stack and a box filter k . In spirit, this convolution plays the same smoothing role as interpolation does in heightfields. The field function $f : \mathbb{R}^3 \rightarrow [-1, 1]$ is defined as:

$$f(p) = 2(\mu * k)(p) - 1 \quad \text{where} \quad (\mu * k)(p) = \frac{V_M(p)}{V_\Omega}$$

Here, the term $V_\Omega = \sigma^3$ denotes the volume of the cubic compact support Ω , which is a box of side length σ and $V_M(p)$ is the volume of material. Unlike voxels, which are by definition a discrete structure, the implicit surface representation derived from the layered representation generates a smooth continuous surface, at the expense of computationally demanding field functions.

3 Procedural generation

In this review, we call procedural any technique that is not directly related to a physical simulation and that does not use real data as exemplars. Instead of simulating the physical processes that sculpt the terrain such as hydraulic erosion, procedural approaches are phenomenological and aim at directly reproducing the effects of the phenomena. They often rely on the properties of terrains, such as the fractal characteristics, and construct terrain from the observations of the real world without taking real data as input.

Procedural generation methods can be classified into two categories. Large scale terrain generation methods synthesize a terrain over the entire plane \mathbb{R}^2 or a large domain $\Omega \subset \mathbb{R}^2$, focus on the fractal and self similar properties of the relief at different scales, and in general provide only little indirect control over the generated features (Section 3.1). In contrast, procedural landform methods target the synthesis of specific landforms such as rivers, canyons, or hills, operate at a smaller scale and provide several direct or indirect control parameters to tune the resulting shapes (Section 3.2).

Both large scale and landform methods often rely on fractional Brownian motion Mandelbrot and Van Ness, 1968 (fBm), which is one of the most commonly used procedural models due to its simplicity. It can be synthesized in many ways, including random walk, subdivision schemes or additive synthesis of functions Ebert et al., 1998.

3.1 Large scale terrain generation

Those methods build on the observation that some terrains such as eroded mountain ranges or coastlines show dendritic structures that have fractal properties, i.e. self-similarities at several scales. Early works on terrain modeling often rely on fractal models; therefore most large scale terrain generation algorithms are α -dimensional, i.e. the fractal nature of the underlying structures allows the creation of an infinity of details that look similar at every scale.

3.1.1 Subdivision schemes

Subdivision schemes iteratively refine an input terrain in order to introduce more and more details. They are used to produce fractal features, although not every subdivision algorithm follows the statistical fractional Brownian motion process. The recursive midpoint displacement subdivision algorithm Fournier et al., 1982a, 1982b progressively refines an input grid by adding fractal details. Let k be the current level of the grid, new points p^{k+1} are introduced by averaging existing neighbor points p^k and by displacing them with a random value. As the iterations progress, the standard deviation of the random value distribution decreases, and the decreasing factor is directly related to the fractal dimension of the result.

Authors have proposed several modified subdivision and averaging schemes (including, triangle, square, and diamond-square) in order to limit visual directional artifacts **Lew87**; **Man88**; Miller, 1986. The most popular is the diamond-square Fournier et al., 1982b, even though it suffers from local extrema artifacts visible at the first subdivisions locations. The square-square Miller, 1986 scheme solves the problem of local extrema by using a more balanced subdivision. Different types of polygons have also been used in subdivisions with mixed types **DKW94**, which necessitates maintaining both topology and geometric consistency in the subdivision process.

3.1.2 Faulting

The faulting algorithm was introduced in **Man82** and developed in Ebert et al., 1998; Voss, 1991. Starting from a flat terrain, the algorithm iteratively generates random vertical faults as lines: points on either side are displaced upwards or downwards according to the distance to the fault. Let $d(p, \phi_i)$ denote the distance to the faulting line ϕ_i , and g a smooth step function parametrized by a radius of influence R , such as the following quartic:

$$g(r) = \begin{cases} \left(1 - \left(\frac{r}{R}\right)^2\right)^2 & \text{if } r < R \\ 0 & \text{otherwise} \end{cases}$$

The elevation function can be defined by summing the influence of the faults as:

$$f(p) = \sum_{i=0}^n f_i(p) \quad \text{where} \quad f_i(p) = a_i g \circ d(p, \phi_i)$$

The coefficients a_i represent the random vertical displacement applied at every iteration, and are in general decreasing in the same way as recursive subdivision with the Hurst exponent $(\frac{1}{2H}, 0 < H < 1)$ and then $D = 3 - H$ is the fractal dimension. While the line cuts the surface in an arbitrary position and the divided parts will be different in size, it is important that in average the random faults select each side with the same probability, otherwise the terrain will either sink or bloat.

3.1.3 Noise

Noise functions **Per85**; **Per89**; Lagae et al., 2009; Worley, 1996 have been widely studied and used for a variety of natural phenomena, including terrain modeling. A complete description of noise functions is beyond the scope of this paper, we refer the reader to Lefebvre et al., 2010 for an overview of procedural noise functions.

Noise functions have been proposed as basis functions for representing infinite terrains, i.e. defined over \mathbb{R}^2 , as a set of scaled and warped noise functions Ebert et al., 1998; Musgrave et al., 1989a. By adding several noises at different scales and amplitudes, it is possible to build a function that locally resembles a real terrain. Fractional Brownian motion, also sometimes referred to as turbulence and denoted as t , can be implemented as a combination of multiple steps of noise each with a different frequency and amplitude. In the context of procedural generation, the variation in frequency from a step to the next is called lacunarity, whereas the variation in amplitude from a step to the next is called gain or persistence.

Let n denote a smooth noise function that maps from \mathbb{R}^2 to $[-1, 1]$ interval; without loss of generality we consider that this fundamental function has a primary frequency of 1; which means that it interpolates values or gradients defined at every integer position. The turbulence function t is defined by summing the contributions of noises with varying frequencies and amplitudes:

$$t(p) = \sum_{i=0}^{o-1} a_i n(\phi_i p)$$

where a_i refer to the different amplitudes, ϕ_i to the different frequencies. The number of terms o is often referred to as the number of octaves, even if the frequencies are not multiples of 2. In

general, the amplitudes and frequencies are defined as a geometric series: $a_i = a_0 p^i$ and $\phi_i = \phi_0 l^i$ where a_0 and ϕ_0 are the base amplitude and frequency, $l \in [0, 1]$ is the lacunarity, and $p \in [0, 1]$ the persistence. The persistence defines how the amplitude decreases in the successive octaves. Values $p \approx 1$ produce very jagged terrains, whereas values $p \approx 0$ drastically limit the impact of the successive frequencies.

Exponential noise was proposed in Parish, 2015 to better reproduce the slope distribution observed in real terrains. The exponential slope distribution spectrum is obtained by analyzing real terrain data, and used to constrain the generation process using gradient noise whose gradient samples match the multi-fractal spectrum **vLJ95**; Parish, 2014, 2015.

The smoothness of the noise function n prevents the creation of crests and ridge lines that can be found in mountainous terrains. Ridge noise was therefore introduced with a view to generating sharp features such as crests or ridges, and simply defined as $r(p) = 1 - |n(p)|$. Although the absolute value was already present in the definition of the turbulence function in **Per85**, the idea of turning it upside-down to produce crests on the top of mountain ranges was introduced in Ebert et al., 1998.

The fractal property of these noise functions is uniform, i.e. they generate mono-fractals, which does not conform to real landscapes. Multi-fractal terrains can be obtained by modifying the sum of octaves so that the amplitude of higher frequencies a_{k+1} should be weighted by a function α according to value of the previously computed octaves Ebert et al., 1998. This can be obtained by using the following recurrence relation:

$$t_{k+1}(p) = \alpha(t_k(p))a_{k+1}n(\phi_i p) + t_k(p) \quad \text{where} \quad t_0(p) = a_0 n(\phi_0 p)$$

Lower elevations $t_k(p)$, computed at step k , scale down higher frequencies in order to smooth valleys, whereas higher values will boost high frequencies to enhance the mountains peaks with small details. Multi-fractal generate terrains that are not uniform featuring smooth areas in plains and ridged landforms in mountains.

Summing the same scaled noise function n can lead to grid artifacts that can be avoided by applying warping functions made of rotations and translations. This concept is easily formalized by the means of an affine transformation applied to p , i.e. using $n(R_i p + v_i)$ where R_i is a 2×2 random rotation matrix and v_i is the translation vector.

Warping Generally, warping can be used to deform the domain and break the monotonicity and regularity of noise. Formally, a continuous warping function may be defined as: $\omega : \mathbb{R}^2 \rightarrow \mathbb{R}^2$; and warped terrains are computed by evaluating $n \circ \omega(p)$ instead of $n(p)$. The warping function ω may be defined as a sum of scaled displacement functions created from noise. Low frequency vector offsets are commonly used for this purpose. Limited erosion effects can be approximated **dCB09** by using warping functions based on scaled noise, oriented in the direction of the gradient of the underlying terrain. Domain warping still only provides a coarse approximation of erosion, and simulation techniques are needed to obtain more realistic effects.

3.2 Landform generation

Local procedural techniques aim at shaping specific landforms such as rivers, cliffs or canyons without relying on simulations or synthesis from exemplars. They operate by the means of geometric control parameters such as features curves or anchor points. Contrary to large scale generation approaches that are often dimensionless, local methods introduce the notion of spatial dimension, which is crucial for reproducing specific structures and patterns.

3.2.1 Controlled subdivisions

Because fractal subdivisions only provide a limited control to the user, several authors have tried to prescribe features such as rivers, crests, or any landforms in the generation process.

River networks with consistent watersheds were created by constraining the mid-point displacement algorithm Kelley et al., 1988 to generate rivers. Subdivision rules were applied directly to river trajectories Perrin and Hugli, 1993 and later extended to entire planets to generate large scale watersheds Dupuy et al., 2011.

Another way to control the generation process is to constrain the fractal reconstruction Bel07; BA05a; BA05b with a subdivision algorithm to respect important features of the terrains like crests or river trajectories. Recently, these methods were adapted and implemented on graphics hardware Thévenin and Bousseau, 2018.

Hnaidi Hnaidi et al., 2010a proposed to break the systematic nature of fractal subdivision rules of a Projected Iterated Function System by inserting details at each step. Artists can thus combine the ease of use of subdivision schemes and the controllability of free forms. Ariyan et al. AM15 used a formalism based on subdivision of planar networks using a path planning algorithm. These planar curves are then assigned heights using altitude profiles and the final heights are computed using interpolations.

3.2.2 Feature-based construction

Feature-based construction methods come in two categories: curve-based terrain generation techniques which rely on generating feature curves to diffuse terrain characteristics and synthesize a variety of landforms, such as ridge lines, coastlines or rivers, and primitive based construction trees that take their inspiration from implicit surface models and combine specific landforms primitives by blending or warping operators.

Curve-based models The work of Gain et al. GMS09 represents one of the earliest sketch-based interfaces for interactive modeling of terrains from control curves. The user specifies landforms, such as mountains and valleys, through a variety of sketched curves, including: baselines (representing the projection of a crest onto the ground plane), elevations (capturing the vertical shape of a crest), and boundaries (limiting how far the landform extends to either side of the baseline). The elevations can be sketched as silhouettes from a particular viewpoint, with additional controls to indicate how they fit relative to existing landforms. Each set of baseline, elevation and boundary curves is submitted to a multi-resolution deformation process which ensures that the terrain is constrained to fit the curves. One unusual aspect is that the noise attributes of the sketched elevation curve are analyzed and used to add corresponding wavelet noise to the local terrain being deformed. This means that a jagged mountainous silhouette will induce similar characteristics in the surrounding terrain. The technique can also be used to build terrains from scratch or modify existing landscapes.

Rusnell et al. Rusnell et al., 2009 used a technique to blend feature curves defined by the combination of an elevation curve (defining crests) and a monotonically decreasing cross-sectional profile (defining slope on either side of the crest). The influence of features can be varied between a simple normalized sum and a weighting that favors feature curves locally. The key aspect that differentiates this from similar Euclidean distance-based schemes is that distance is calculated efficiently on the grid using a shortest-path algorithm. Every grid-node is connected edge-wise to its eight neighbors and the shortest path is used to determine distance to a feature curve, whose grid-vertexes serve as generator nodes for the algorithm.

A specific approach for modeling and generating canyon landscapes was introduced in **DCPSB14**. Given an input heightfield generated by using a sum of procedural noises, the authors layer the terrain into terraces by applying clamping functions. Mesas are isolated by creating paths as curves between valleys to create non connected mountain ranges. The trajectory of the river is then generated by using a shortest path algorithm between control points minimizing the elevation range between control points. Finally, the foothills are smoothed.

Hnaidi et al. Hnaidi et al., 2010b created terrains from prescribed feature curves representing the skeleton of crests, rivers and major stream landforms. Feature curves are 2D splines with additional information attached along the curve, describing the elevation and the slopes on each side of the curve. This resulting vector-based model is compact in terms of memory (10kB). The surface is reconstructed using a modified diffusion equation taking into account noise parameters and the slope. Noise parameters and gradient constraints are propagated across the cells by a standard diffusion equation. The obtained constraints maps are then used in a second pass in order to guide the diffusion in particular areas with slopes. Details are generated using a noise map synthesized using the diffused noise parameters, and terrain is obtained by adding details to the smooth diffused terrain. The multi-grid implementation of the diffusion process on graphics hardware allows for interactive generation rates. While the maximum resolution of the synthesized terrain was constrained by the available memory at that time, i.e. 2048×2048, it could probably be higher with recent hardware.

Construction trees The general approach consists in locally modifying the elevation of the terrain by using different kinds of models. This can be achieved by using compactly supported primitive function f_i representing specific landforms over their domain Ω_i , and combining them together to construct complex terrains G  nevaux et al., 2015, using either procedural generation approaches Galin et al., 2013 or example-based sparse synthesis Galin et al., 2016.

G  nevaux et al. G  nevaux et al., 2015 proposed a hierarchical construction tree combining primitives representing a variety of landforms. The primitives at the leaves of the tree represent landforms at different scales such as hills, mountains or mountain ranges, valleys, riverbeds, and are implemented as elevation functions $h_i(p)$ associated to a compactly supported weighting function $\alpha_i(p)$ over their domain Ω_i . The primitives can be hierarchically combined using different operators such as carving, blending or warping to apply deformation effects. The major contribution is a hierarchical model that allows to model and control the placement of landforms features.

G  nevaux et al. Galin et al., 2013 proposed to fully generate a terrain from its river-network. The river network grows over the input terrain by using a grammar that satisfies Horton-Strahler properties. The Horton-Strahler number quantifies the branching complexity of the geometric graph representing the drainage network Horton, 1945. The Holto-Strahler number is computed on every edge of the graph by using the maximum uphill edges values and by incrementing in cases of equality. Leaves of the graph (river sources) are initialized to one. Each river of the network is labeled with respect to the empirical Rosgen classification of watercourses Rosgen, 1994 so that it embeds useful geometrical information. The elevation of the terrain is derived from the propagation of the elevation along the trajectories of the rivers and are computed using a hierarchical terrain construction tree.

3.2.3 Volumetric procedural terrains

Volumetric procedural methods are not very present in the literature. Gamito et al. Gamito and Musgrave, 2001 introduced implicitly-modeled terrains by defining a field function $f : \mathbb{R}^3 \rightarrow \mathbb{R}$

and by warping space along the vertical axis to generate overhangs. Recall that f is derived from h as $f(p) = h(p_{xy}) - p_z$. Instead of defining the elevation h as a function of points in the domain $\Omega \subset \mathbb{R}^2$, the warped surface of the terrain is therefore implicitly defined as:

$$S = \{p \in \mathbb{R}^3 \mid f \circ \omega^{-1}(p) = 0\}$$

The warping function $\omega^{-1} : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ deforms space horizontally and transforms steep parts of the elevated terrain into a concave surface resembling cliffs with overhangs. In practice, the warping function can be defined as a procedural displacement function based on a sum of 3D noise functions, smoothly clamped to a given region of influence in space. Let $\alpha : \mathbb{R}^2 \rightarrow [0, 1]$ denote the compactly supported function defining the region of influence, we have:

$$\omega^{-1}(p) = p + \alpha(p) \sum_{i=0}^{n-1} n(\phi_i p)$$

Peytavie et al. Peytavie et al., 2009a allow the modeling of 3D features like overhangs by using void as a material layer. The corresponding implicit representation can be obtained from a layer stack model to smooth the surface.

3.2.4 Analysis

To avoid the systematic fractal aspect of recursive subdivisions, constraint generation methods rely on user-provided constraints to adapt the subdivision process locally to create the desired landforms. User-controlled perturbations in the subdivision process were also proposed as a means to obtain terrains with different fractal characteristics.

In contrast, local methods and more specifically feature-based modeling provide a more intuitive way of authoring terrains: a complete discussion will be presented in Section 6.4. The main challenge of terrain modeling from features stems from the difficulty to propagate a sparse information, i.e., specific landforms such as peaks or rivers, over an entire terrain, e.g., generating hills between mountain ranges and plains. Several processes such as thermal diffusion, or shortest path computation have been proposed to address this problem. In those approaches, the global structure of the terrain often results from the authoring process during the creation. Because the placement of many features can be a tedious task for the user, research has also focused on the automatic creation of structuring features such as rivers.

4 Simulation

While procedural approaches are phenomenological, i.e., focus on the generation of a particular phenomenon, simulation techniques model the causes and the effects that result from a simulation process. Simulations can be thought of as complex systems where the landform patterns and structures emerge from the interaction of the simulated elements.

Most of the simulations for terrain modeling are based on erosion. Erosion is the action of surface processes that remove material from one location and then transport it to another location, possibly outside the domain. Note that erosion is different from weathering, which involves no movement, but is rather a modification of physical properties and surface aspect.

Erosion can be described as a three-step process: the material is eroded and detached from the underlying terrain, then transported by the agent, and eventually deposited at a different location. Erosion processes include, but are not limited to, rainfall and surface runoff, river and stream erosion, coastal and sea erosion, glacial erosion, wind erosion, and mass movement.

4.1 Thermal erosion

Thermal erosion combines thermal weathering and mass movement, representing the downward movement of rocks and sediments on slopes, mainly due to the force of gravity. It is caused by water present inside cracks and small intrusions of the material boundary. As the temperature changes, the different thermal expansion of water and the material causes the material to break and fall.

4.1.1 Heightfield thermal erosion

Thermal erosion was introduced in Musgrave et al., 1989a as a proxy for a group of various erosional processes other than hydraulic erosion. Note that thermal weathering is one of the many causes of rock breakage, large-scale landslides and the accretion of granular material, all of which results in a perceived regularity in the slope angle of many mountain and hill slopes.

The transportation is caused by gravity and relies on the concept that the deposited granular material has an inner friction that stops the movement when a so-called talus angle has been reached. Let θ denote the repose angle, also referred to as the talus angle, of the deposited (granular) material. The equation is:

$$\frac{\partial h}{\partial t}(p) = \begin{cases} -k(s(p) - \tan \theta) & \text{if } s(p) > \tan \theta \\ 0 & \text{otherwise} \end{cases}$$

The talus angle can be used as a parameter to control the slope of the screes and cones that form at the bottom of the eroded landforms. The talus angle varies within $30^\circ - 45^\circ$ degrees interval for earth forms.

Thermal erosion can be simulated as a relaxation process: at each time step the slope of the terrain $s(p_{ij})$ is computed: if it is lower than $\tan \theta$ no erosion occurs, otherwise a certain amount of material proportional to $\tan \theta - s(p)$ is eroded and removed from the bedrock layer and distributed to the neighboring cells. When operating with a layered model, the amount of removed bedrock is converted to silt. The algorithm stops when there is no more material to move.

Roudier et al. Roudier et al., 1993 introduced bedrock resistance for both hydraulic and thermal erosion. The concept consists in approximating the underlying geology of the terrain by defining a volumetric function $\rho : \mathbb{R}^3 \rightarrow \mathbb{R}$ representing the resistance of the different materials. The erosion equation is then modified as:

$$\frac{\partial h}{\partial t}(p) = -\rho(p)(s(p) - \tan \theta)$$

In general, the resistance is constant along a vertical line which allows it to be defined by a simpler function $\rho : \mathbb{R}^2 \rightarrow \mathbb{R}$.

4.1.2 Specific landforms

Thermal erosion was used to simulate table mountains (mesas) in **BA05c**. The thermal erosion breaks rimrock (bedrock) into falling rocks that behave like granular material forming the typical accretion cones on the hillside of table mountains.

A variant of thermal erosion was proposed for small-scale volumetric simulation of cliff retreat in Iben et al., 2003. This algorithm procedurally generates faults around voxel groups that represent rock blocks and perform a discrete simulation of the detachment and falling of blocks. A set of blocks at the surface is chosen depending on sliding conditions, and removed from the system. Although not considering a time step, or a stochastic time-dependent probability of block breakage, this approach relies on a repose angle that generates talus, and therefore is a variant of thermal erosion.

4.1.3 Cliffs and overhangs

The previously described thermal erosion algorithms operate on heightfield data-structures and do not allow for rock detaching and falling from cliffs, which would lead the formation of concave shapes, such overhangs or even caves. Peytavie et al., 2009a presented an approximation of 3D thermal erosion by using a hybrid implicit-surface and layerstack representation.

Blocks detaching from vertical cliffs or steep slopes are computed using a Voronoi decomposition of space to define their boundary. The bedrock material is then converted into sand and rock material layers, which are stabilized into accretion piles. Piles of rocks and stones can also be generated from the granular material layers, as described in Peytavie et al., 2009b.

While this approach allows for generating complex caves and cliffs with sharply sculpted overhangs, the volumetric model is limited to $\approx 1 \times 1 \times 1 \text{ km}^3$ scenes.

4.1.4 Analysis

Thermal erosion has been widely adopted in Computer Graphics both because of the simplicity of the governing equation, which allows for a straightforward implementation, even on graphics hardware, and because of its intuitive parametrization using the talus angle. It correctly creates accretion cones of fallen rocks, sand, and granular materials in both screes and accretion areas. Furthermore, it is an important complementary step to hydraulic erosion methods, which carve deep channels between high crests. In this context, thermal erosion gives a measure of the spacing between erosive features.

Thermal erosion has several limitations however. First, it is usually implemented as an iterating scheme that approximates forward Euler time stepping and this is physically accurate only for small time steps. Although this is of lesser importance when thermal erosion is applied as a post-processing step, this issue prevents its use when simulating large scale erosional features Cordonnier et al., 2016.

Second, fallen rocks do not only provide a fundamental visual improvement when represented at their rest state at the bottom of cliffs, they also have a non negligible erosive effect during their fall. Existing algorithms regard the fallen material as granular. However, larger blocks of material can fall, and subsequently break into rocks, which would end up making the simulation significantly more complicated and even harder to control. This effect can be approximated with more general approaches to debris flow erosion, as used in geology Stumpf and Dietrich, 2003. In addition to the breakage of rock, these model the erosive effect of falling rocks along their trajectory. Third, is the

issue of perceived regularity: even when the talus angle is set with random perturbations, slopes look unnaturally similar. Geologists prefer hillslope erosion that averages a variety of lower scale processes into a diffusion equation:

$$\frac{\partial h}{\partial t}(p) = -k\Delta h(p)$$

While this equation does not take into account the material layers introduced by thermal erosion, it constructs more visually irregular transitions between slopes, especially when combined with fluvial erosion **BS97**.

4.2 Tectonic

A lot of attention in Geology and Geomorphology has been focussed on the simulation of crust deformation driven by the compression of tectonic plates McCammon, 1992. This process results in the continuous vertical raising of mountain ranges – a phenomenon called uplift, which is countered by various forms of erosion that shape mountains Tucker and Hancock, 2010. In contrast, the simulation of tectonics for terrain modeling has received less attention in the Computer Graphics community. This can be explained by the complexity of geological phenomena described by differential equations that are computationally demanding. Moreover, it is difficult to set the parameters of the simulation so as to control the terrain generation process and follow the intent of the user.

4.2.1 Tectonics and stream power erosion

Michel et al. **MEC15** procedurally generate the topography of a mountain range based from a user-defined sketch of peaks and rivers. This input is used to build a procedural approximation of tectonic plates whose speeds wrap a noise based fold map. Terrain elevation is obtained by combining the fold map with a watershed obtained by diffusing a slope value away from the user-sketched rivers. The relief is then eroded to generate the mountains by using a hydraulic erosion method.

Large scale terrain generation from tectonic uplift and fluvial erosion was addressed in Cordonnier et al., 2016. This paper introduces the Stream Power erosion law derived from geomorphology Whipple and Tucker, 1999 to computer graphics, which relates the erosion rate at a given point p to the drainage area $A(p)$ (which approximates the water flux by integrating the precipitation rate over the upstream catchment domain), the local slope $s(p)$ and the tectonic uplift $u(p)$:

$$\frac{\partial h}{\partial t}(p) = -kA(p)^m s(p)^n + u(p)$$

The mountain elevation results from the combination of a progressive uplift and the subsequent erosion. The uplift is used as an input in the form of a map of the rate of change of elevation, and the simulation is solved implicitly in linear time.

This work was extended in Cordonnier et al., 2018 with a geologically-coherent uplift derived from the relative movement of the tectonic plates prescribed by the user. The global mountain range uplift is computed by considering the crust as an incompressible viscous material. Folds are added procedurally and their wavelength is approximated by considering the crust as a stack of layered sheets of rocks with discontinuous physical properties. The terrain is finally eroded using

an improved version of the algorithm presented in Cordonnier et al., 2016 that takes into account the characteristics of the different bedrock strata. The method generates a layered 2.5D model with folds and faults.

4.2.2 Analysis

Tectonic-based simulations attempt to reproduce large scale erosion effects, taking into account the uplift of the bedrock balanced by different types of erosion described in geology. Similar to other mesh or grid-based techniques, those approaches are limited in scale range. Existing techniques produced realistic mountain ranges with dendritic river networks covering over ≈ 100 km-wide domains, at a precision of ≈ 100 m. Although the simulation can be controlled by interactively moving the tectonic plates, controlling the generation of the small-scale features remains difficult.

4.3 Hydraulic Erosion

Hydraulic erosion occurs when the motion of water against the bedrock surface produces mechanical detachment. Hydraulic erosion encompasses a number of mechanical erosional processes such as abrasion, corrasion (not to be confused with corrosion) and saltation. Chemical erosion (more often called chemical weathering) is distinct from mechanical erosion but is also part and parcel of hydraulic erosion. It changes the composition of rocks, transforming them when water interacts with minerals to create various chemical reactions.

The existing hydraulic erosion algorithms in Computer Graphics vary in the way they describe the fluid simulation (which in general also relates to the underlying models and data structures) and time and space scales.

Water movement acting to detach, transport and deposit material can be described by the Navier-Stokes equations Chorin and Marsden, 1988 that have been studied in Computer Graphics for a long time Bridson, 2008. Fluid movement can be computed either by Eulerian approaches or Lagrangian methods.

4.3.1 Eulerian approaches

These methods rely on a discrete grid-based model that represents the input scene as well as the fluid. The simulation calculates the pressure, the velocity, and the amount of fluid in each discrete cell. Musgrave et al. Musgrave et al., 1989a presented a complete hydraulic simulation model by simulating material detachment, transport and deposition between heightfield cells. The erosive power of a given amount of water in a cell is a function of its volume and the amount of sediment already carried in the water. When the sediment capacity of fluid is reached, the material is deposited on the ground.

A geological representation for modeling the strata of the bedrock was introduced in Roudier et al., 1993 that considers the characteristics of the different materials during the erosion process: the hydraulic erosion is the more intense as the rock is the softer.

These approaches were extended in Benes and Forsbach, 2002 by introducing an algorithm for hydraulic erosion in which water dissolves soil, transfers it, and deposits at different locations by gravitational settling. This approach generates layers of smooth material deposited in floor bed and in pools of water after drying.

Nagashima et al. Nagashima, 1998 proposed a modified and improved physically-inspired erosion model and focused on the generation of valleys. The method combines a layer-based geological representation of the different strata of the bedrock with thermal erosion and a specific stream erosion process.

Neidhold et al. Neidhold et al., 2005 combined fluid simulation with erosion and analyzes the acceleration or deceleration of the fluid to erode the bedrock or deposit sediments. They also use the sediment capacity Musgrave et al., 1989a to deposit material in still water. The authors report fast simulation allowing this approach to be used at interactive speeds. A parallel implementation was later proposed in ASA07.

Benes et al. Benes et al., 2006b combined a 3D fluid simulation with sediment transportation and deposition to simulate hydraulic erosion. The underlying data structure for representing the terrain was a voxel grid, with data representing the variable amount of material effectively stored in every cell. The movement of the fluid is computed on a grid, and the force of the fluid applied to the material boundary defines the importance of erosion. When the force is larger than the material resistance, some material is eroded and transported by the moving fluid. When the fluid slows down, it deposits the material in the corresponding voxel cell.

Hydraulic erosion has been also used for generating rivers, and an approach inspired by self-organizing systems was proposed in Peytavie and Merillou, 2013. Here, the simulation relies on self-organization and emergent phenomena to synthesize coastlines and terrains with the fractal features characteristic of hydraulic erosion.

Benes et al. Benes et al., 2006a combined shallow water simulation with hydraulic erosion. This approach allows the user to place water sources, control the flow of water, and edit the underlying terrain. The shallow water simulation, however, operates on a layered heightfield representation and does not allow for the simulation of volumetric erosion phenomena such as cliff overhangs produced by river erosion.

4.3.2 Performance challenges

Fluid simulation itself is computationally expensive and becomes even more demanding when combined with erosion. Several approaches propose technical implementation improvements that speed up computations, in general by exploiting graphics hardware as most fluid simulation computations can be performed in parallel. Mei et al. Mei et al., 2007 demonstrated the effectiveness of parallel methods and achieved interactive feedback. Št'ava et al. Št'ava et al., 2008 further improved the method by using a multi-layered representation and increasing the discretization of the domain beyond 1024^2 while preserving interactive feedback. Subsequently, Vanek et al. Vanek et al., 2011 addressed memory limitations by tiling the terrain and offloading the erosion into rectangular blocks that are swapped to the main memory of the computer.

A more general stochastic simulation of different events that impacts terrain at shorter geological timescales (≈ 1000 years) was introduced in Cordonnier et al., 2017. The method combines the joint effects of erosion and vegetation by considering atomic events such as water drop (hydraulic erosion), rock fall, lightning strikes, and vegetation related events such as plant seeding, growth and death, as well as forest fires. An event is randomly chosen, and follows a simple cell by cell path (no loop, no interruptions, no branches) progressively transforming the multiple layers of materials that compose the terrain: bedrock, granular material (sand, humus, rock), vegetation densities and resources such as moisture or illumination.

4.3.3 Lagrangian approaches

Particle-based approaches for computational fluid dynamics model the flow as a collection of particles that move under the influence of hydrodynamic and gravitational forces. Eulerian methods obtain the solution relative to a fixed grid, whereas the Lagrangian framework define the flow in terms of the concentration of advected particles.

General terrain erosion methods distribute particles across the domain to approximate rainfall, and then simulate their movement and effect on the terrain by eroding bedrock into sediments and then lifting, transporting, and eventually depositing these sediments.

Water particles were introduced in Chiba et al., 1998 to compute the erosive forces exerted by the flowing water down the terrain to simulate erosion. They introduced the concept of velocity fields that are calculated from motion of water and used it to erode the underlying terrains. Smoothed Particle Hydrodynamics were used in Krivánek et al., 2009. The fluid particles move and when they hit the terrain they erode it. Each moving fluid particle then transports material that changes its mass; when the speed of the particles is slow, it deposits the material.

4.3.4 Specific landforms

The automatic generation of meandering rivers was addressed in Kurniawan and Warren, 2012, 2013. It is an approximation inspired by erosion that uses particles that carry sediment and generate meandering riverbeds. The method aims at generating meanders and carves the riverbed in the ground. The resolution depends on the type of the river, and as for most simulations the system is limited to small parts of rivers.

4.3.5 Analysis

Although erosion simulations are often considered the most realistic way of eroding terrains because the underlying physics capture emerging phenomena such as channels or accretion cones, several fundamental problems remain.

Scale range Many hydraulic erosion algorithms, some Eulerian methods simulating the dynamics of fluids and almost all Lagrangian methods, perform the simulations at a small time and space scale, and then implicitly scale up the results to approximate large scale terrains. This is intrinsically flawed. While generating visually convincing effects, these approaches are not realistic as the equations of the corresponding phenomena are not linear. The scale of the simulated features heavily depends on the underlying terrain representation. Such simulations can efficiently model scales of a certain size, but using the same approach to simulate much larger or much smaller phenomena is either impossible or computationally intractable.

Thus, depending on the simulated phenomena, simulation can become very time consuming, which makes it ill-suited to the detailed and precise modeling and generation of terrains with large extent.

Control A second problem is the control of these methods. While erosion simulations are more physically or geologically accurate than procedural approaches, they are notoriously difficult to control. The user is usually left with defining the initial conditions of the simulation and waiting for the generated results. If the result is not satisfactory, the initial conditions need to be changed. Moreover, they do not scale well and designing a large-scale simulation is a very difficult task.

It is common to use hydraulic and thermal erosion as a beautification processes to enhance procedurally generated terrain with sedimentary valleys and erosion landmarks, such as gorges and

ravines. However, this neglects a crucial element: mountain ranges result from continuous uplift countered by various forms of erosion. The form of the initial input terrain thus plays a crucial role in the realism of the output. Ideally, the input prior to erosion should be geomorphologically sound and encode the effects of uplift.

4.4 Other erosion phenomena

Compared to the large amount of work that dealt with hydraulic and thermal erosion, other phenomena have not received much attention by authors despite their potentially big impact on landforms generation. Phenomena that have seldom or never been addressed in computer graphics include aeolian, coastal, lightning, glacial, and karst erosion. Glacial erosion is a phenomenon that applies at smaller time scales than fluvial erosion and that induces the creation of U-shape valleys and hanging valleys. Coastal erosion and karst are phenomena that have a volumetric impact on the terrain and that are highly related to the subsurface geology.

Aeolian erosion is caused by winds that erode, transport, and deposit materials; it is more intense where vegetation is sparse since sediments are unconsolidated. Aeolian processes encompass the effects of wind that erode terrains by deflation (the removal of loose, fine-grained particles by the turbulent action of the wind) and by abrasion (the erosion of surfaces by the grinding and sandblasting action of sand particles).

While simulating the formation of sand dunes has received a lot of attention in physics, it has not been a focus in Computer Graphics. Onoue Onoue and Nishita, 2000 borrowed the saltation equation and adapted simulations from physics into a procedural method to generate sand ripples. This approach was further extended in Bendels and Rumpf, 2004 by taking into account the collision of sand with obstacles. Those methods rely on discrete layered heightfields and therefore have a limited extent.

Spheroidal erosion on voxel grids was proposed in Benes et al., 2006a; Jin et al., 2010 for approximating the effects of wind erosion on rocks. The local curvature and accessibility of the surface indicates its exposure to the environment and causes this to be eroded faster. Positive curvature erodes edges and creases, whereas negative curvatures causes protrusions and holes. This approach simulates efficiently so called Goblin structures commonly carved in sandstones such as in Goblin Valley State Park, Utah. An extended version of spheroidal erosion was adapted for eroding triangular meshes and generating Goblins but also concavities and overhangs Talton and Johnson, 2010. This method, however, does not allow the creation of arches or caves.

Lightning may have an erosive effect on exposed terrains parts, where a single strike can destroy a large amount of bedrock and project rocks within several meters. Cordonnier et al. Cordonnier et al., 2017 included the effects of lightning as an erosive agent: strikes are scattered randomly on the terrain, with a higher probability at higher locations with negative terrain curvatures. Bedrock is destroyed, and material is added in a rock layer in the grid cells near the impact, similarly to transport of the outcome of thermal erosion.

5 Conclusion

In this paper, we have presented a comprehensive overview of digital terrain modeling methods in Computer Graphics, categorized into procedural generation, simulation, and example-based approaches. Each of these techniques offers unique advantages and faces specific challenges.

Procedural generation methods are advantageous for their computational efficiency and ability to produce a wide variety of terrain types with minimal input. However, they often lack realism and user control, which can limit their applicability in scenarios requiring specific geological features or high levels of detail.

Simulation techniques, on the other hand, provide a higher degree of realism by modeling the physical processes that shape real-world terrains. These methods can generate highly detailed and geologically accurate terrains but are computationally expensive and can be difficult to control. Additionally, the scale of the simulated features is often limited by the resolution of the underlying data structures.

Example-based approaches leverage real-world data to produce highly realistic terrains, benefiting from the rich detail and natural variations present in scanned terrain models. However, the quality of the generated terrain is heavily dependent on the input data, and these methods can struggle with scalability and the integration of disparate terrain features.

Our comparison of existing terrain modeling methods highlights several key factors that influence their suitability for different applications, including the variety of achievable landforms, realism, scale, authoring control, and computational efficiency. By understanding these factors, developers can better select the appropriate terrain modeling technique for their specific needs.

Looking forward, there are several promising research directions that could address the current limitations of terrain modeling methods. Hybrid approaches that combine the strengths of procedural, simulation, and example-based techniques could offer improved realism and control. Advances in machine learning and data-driven methods also hold potential for enhancing the scalability and accuracy of terrain models.

Ultimately, the continued development of terrain modeling methods will enable more realistic and interactive virtual environments, enhancing applications in gaming, film, simulation, and beyond. By building on the foundations established in this survey, future research can further push the boundaries of what is possible in digital terrain modeling.

References

- Bendels, G. H., & Rumpf, M. (2004). Time discretization for energetic variational approaches to free boundary problems. *Interfaces and Free Boundaries*, 9–24 (cit. on p. 17).
- Benes, B., & Forsbach, R. (2002). Layered data representation for visual simulation of terrain erosion. *Proceedings of Spring Conference on Computer Graphics*, 80–86 (cit. on p. 14).
- Benes, B., Forsbach, R., & Owens, J. D. (2006a). Interactive terrain modeling using hydraulic erosion. *WSCG*, 79–86 (cit. on pp. 15, 17).
- Benes, B., Tesinsky, V., Horny, J., & Bhatia, S. K. (2006b). Hydraulic erosion using smoothed particle hydrodynamics. *Proceedings of Eurographics Workshop on Natural Phenomena*, 39–46 (cit. on p. 15).
- Bridson, R. (2008). Fluid simulation for computer graphics. *SIGGRAPH Courses*, 151–173 (cit. on p. 14).
- Chiba, N., Muraoka, K., & Fujita, K. (1998). An erosion model based on velocity fields for the visual simulation of mountain scenery. *The Journal of Visualization and Computer Animation*, 9(4), 185–194 (cit. on p. 16).
- Chorin, A. J., & Marsden, J. E. (1988). A mathematical introduction to fluid mechanics. *Springer-Verlag*, 320–324 (cit. on p. 14).

- Cordonnier, G., Benes, B., & Cani, M.-P. (2016). Large scale terrain generation from tectonic uplift and fluvial erosion. *Proceedings of ACM SIGGRAPH / Eurographics Symposium on Computer Animation*, 1–12 (cit. on pp. 12–14).
- Cordonnier, G., Cani, M.-P., & Benes, B. (2018). Geologically coherent uplift-erosion for large scale terrain generation. *ACM Transactions on Graphics*, 120:1–120:12 (cit. on p. 13).
- Cordonnier, G., Guérin, E., & Galin, E. (2017). Interactive generation of time-evolving, eroded terrains for real-time applications. *ACM Transactions on Graphics*, 224:1–224:15 (cit. on pp. 3, 15, 17).
- Dupuy, J., Galin, E., Gonzato, J.-C., & Kerautret, B. (2011). Terrain generation using procedural models based on hydrology. *Proceedings of Eurographics Workshop on Natural Phenomena*, 1–12 (cit. on p. 8).
- Ebert, D. S., Musgrave, F. K., Peachey, D. R., Perlin, K., & Worley, S. (1998). Texturing and modeling: A procedural approach. *SIGGRAPH Courses*, 472 (cit. on pp. 5–7).
- Fournier, A., Fussell, D., & Carpenter, L. (1982a). Computer rendering of stochastic models. *Communications of the ACM*, 25(6), 371–384 (cit. on p. 5).
- Fournier, A., Fussell, D., & Carpenter, L. (1982b). Stochastic modeling of terrain. *Proceedings of ACM SIGGRAPH*, 16, 263–268 (cit. on p. 5).
- Galin, E., Devillers, O., Guérin, E., & Peytavie, A. (2016). Sparse representation of terrains. *ACM Transactions on Graphics*, 106:1–106:12 (cit. on p. 9).
- Galin, E., Guérin, E., & Gènevaux, O. (2013). Procedural generation of roads. *Proceedings of Eurographics Workshop on Natural Phenomena*, 15–24 (cit. on p. 9).
- Gamito, M. N., & Musgrave, F. K. (2001). Modeling and sculpting 3d clouds. *Proceedings of the Graphics Interface Conference*, 145–152 (cit. on p. 9).
- Gènevaux, O., Guérin, E., Peytavie, A., & Galin, E. (2015). Terrain generation using procedural models based on hydrology. *ACM Transactions on Graphics*, 143:1–143:13 (cit. on p. 9).
- Hnaidi, H., Guérin, E., & Akkouche, S. (2010a). Feature-based terrain generation using diffusion equation. *Proceedings of Eurographics Workshop on Natural Phenomena*, 25–32 (cit. on p. 8).
- Hnaidi, H., Guérin, E., & Akkouche, S. (2010b). Sketch-based terrain modeling using feature-based data structures. *Computer Graphics Forum*, 283–291 (cit. on p. 9).
- Horton, R. E. (1945). Erosional development of streams and their drainage basins; hydrophysical approach to quantitative morphology. *Geological Society of America Bulletin*, 275–370 (cit. on p. 9).
- Iben, H. N., Foley, T., Mitchell, J. L., & Carpendale, S. (2003). Interactive cliffs. *Proceedings of the Graphics Interface Conference*, 111–118 (cit. on p. 12).
- Jin, X., Fan, R., Benes, B., & Browning, R. (2010). Interactive control of boiling lava using smoothed particle hydrodynamics. *Computer Animation and Virtual Worlds*, 91–100 (cit. on p. 17).
- Kähler, R., Sbalzarini, I. F., & Arnold, A. (2013). Adaptive and parallel particle-in-cell simulation. *Proceedings of the Workshop on Molecular Graphics and Visual Analysis of Molecular Data*, 13–20 (cit. on p. 4).
- Kelley, A. D., Malin, M. C., & Nielson, G. M. (1988). Terrain simulation using a model of stream erosion. *Computer Graphics (Proceedings of ACM SIGGRAPH)*, 263–268 (cit. on p. 8).
- Krivánek, J., Beneš, B., Kolíbal, Z., & Štáva, O. (2009). Fast hydraulic erosion simulation and visualization on gpus. *Proceedings of Eurographics Workshop on Natural Phenomena*, 57–64 (cit. on p. 16).
- Kurniawan, A., & Warren, J. (2012). Detail-preserving simplification of polygonal models. *Computer Graphics Forum*, 1817–1825 (cit. on p. 16).

- Kurniawan, A., & Warren, J. (2013). Feature-preserving adaptive mesh simplification. *Computer Graphics Forum*, 251–260 (cit. on p. 16).
- Lagae, A., Lefebvre, S., Drettakis, G., & Dutré, P. (2009). *A survey of procedural noise functions*. Computer Graphics Forum. (Cit. on p. 6).
- Laine, S., & Karras, T. (2011). Efficient sparse voxel octrees. *IEEE Transactions on Visualization and Computer Graphics*, 204–220 (cit. on p. 4).
- Lefebvre, S., Sylvain, G., & Crassin, C. (2010). Parallel controlled level of detail for procedural models. *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on High Performance Graphics*, 123–131 (cit. on p. 6).
- Mandelbrot, B. B., & Van Ness, J. W. (1968). Fractional brownian motions, fractional noises and applications. *SIAM Review*, 422–437 (cit. on p. 5).
- McCammon, J. W. (1992). Computation of potential of mean force from brownian dynamics trajectories. *Computer Physics Communications*, 64, 85–92 (cit. on p. 13).
- Mei, X., Dehne, F., & Hortsch, M. (2007). Parallel r-tree search on the gpu. *Proceedings of the International Conference on Computational Science*, 311–318 (cit. on p. 15).
- Miller, G. S. (1986). The definition and rendering of terrain maps. *Proceedings of ACM SIGGRAPH*, 39–48 (cit. on p. 5).
- Musgrave, F. K., Kolb, C. E., & Mace, R. S. (1989a). The synthesis and rendering of eroded fractal terrains. *Proceedings of ACM SIGGRAPH*, 41–50 (cit. on pp. 3, 6, 11, 14, 15).
- Musgrave, F. K., Kolb, C. E., & Mace, R. S. (1989b). *The synthesis and rendering of eroded fractal terrains*. ACM SIGGRAPH. (Cit. on p. 3).
- Nagashima, Y. (1998). Computer simulation of erosion. *The Visual Computer*, 1–17 (cit. on p. 15).
- Nardi, F., Loiacono, I., Petrangeli, A., & Vollmer, D. (2013). Modeling geology and geomorphology processes. *Journal of Geographic Information Systems*, 31–40 (cit. on p. 2).
- Neidhold, B., Wacker, M., & Deussen, O. (2005). Interactive physically based fluid and erosion simulation. *Proceedings of Eurographics Workshop on Natural Phenomena*, 25–32 (cit. on p. 15).
- Onoue, K., & Nishita, T. (2000). Virtual sandbox. *Proceedings of the ACM symposium on Virtual reality software and technology*, 85–92 (cit. on p. 17).
- Parish, Y. (2014). Modeling and sculpting 3d clouds. *ACM Transactions on Graphics*, 145–152 (cit. on p. 7).
- Parish, Y. (2015). Interactive terrain generation with multi-resolution feature control. *Computer Graphics Forum*, 37–48 (cit. on p. 7).
- Perrin, F., & Hugli, H. (1993). Modeling and generating realistic landscapes with river networks. *Proceedings of Eurographics Workshop on Natural Phenomena*, 31–40 (cit. on p. 8).
- Peytavie, A., & Galin, E. (2007). Procedural generation of roads. *Proceedings of Eurographics Workshop on Natural Phenomena*, 15–24 (cit. on p. 1).
- Peytavie, A., Galin, E., Merillou, S., & Grosjean, J. (2009a). Arches: A framework for modeling complex terrains. *Proceedings of Eurographics Workshop on Natural Phenomena*, 45–52 (cit. on pp. 10, 12).
- Peytavie, A., Galin, E., Merillou, S., & Grosjean, J. (2009b). Procedural generation of rock piles using a rockfall process. *Proceedings of Eurographics Workshop on Natural Phenomena*, 69–76 (cit. on p. 12).
- Peytavie, A., & Merillou, S. (2013). Sketch-based terrain modeling using feature-based data structures. *Computer Graphics Forum*, 283–291 (cit. on p. 15).
- Rosgen, D. L. (1994). A classification of natural rivers. *CATENA*, 169–199 (cit. on p. 9).

- Roudier, P., Perrin, F., & Pouliquen, J. (1993). Modeling and generating realistic landscapes with river networks. *Proceedings of Eurographics Workshop on Natural Phenomena*, 31–40 (cit. on pp. 11, 14).
- Rusnell, B., Musgrave, F. K., & Ebert, D. S. (2009). Feature-based terrain generation using diffusion equation. *Proceedings of Eurographics Workshop on Natural Phenomena*, 25–32 (cit. on p. 8).
- Št'ava, O., Benes, B., Brisbin, M., & Krivánek, J. (2008). Interactive terrain modeling using hydraulic erosion. *Proceedings of Eurographics Workshop on Natural Phenomena*, 47–56 (cit. on p. 15).
- Stumpf, R., & Dietrich, P. (2003). Manual of applied geology for engineers. *Gustav Fischer Verlag*, 24–38 (cit. on p. 12).
- Talton, J. O., & Johnson, D. E. (2010). Sph for non-deterministic natural phenomena simulation. *Proceedings of Eurographics Workshop on Natural Phenomena*, 47–54 (cit. on p. 17).
- Thévenin, P., & Bousseau, A. (2018). Efficient parallel simplification of point-sampled surfaces. *Proceedings of Eurographics Workshop on Natural Phenomena*, 23–30 (cit. on p. 8).
- Tucker, G. E., & Hancock, G. R. (2010). Modelling landscape evolution. *Earth Surface Processes and Landforms*, 28–45 (cit. on p. 13).
- Vanek, J., Benes, B., Horny, J., & Št'ava, O. (2011). Large-scale hydraulic erosion simulation and visualization. *Proceedings of Eurographics Workshop on Natural Phenomena*, 47–56 (cit. on p. 15).
- Vasa, L., Marroquim, R., & Grinspun, E. (2016). Level of detail rendering for integral surfaces. *Proceedings of ACM SIGGRAPH / Eurographics Symposium on Computer Animation*, 47–56 (cit. on p. 4).
- Voss, R. F. (1991). Random fractals: Self-affinity in noise, music, mountains, and clouds. *Physica D*, 38, 362–371 (cit. on p. 6).
- Whipple, K. X., & Tucker, G. E. (1999). Dynamics of the stream-power river incision model: Implications for height limits of mountain ranges, landscape response timescales, and research needs. *Journal of Geophysical Research: Solid Earth*, 104(B8), 17661–17674 (cit. on p. 13).
- Worley, S. (1996). A cellular texture basis function. *Proceedings of ACM SIGGRAPH*, 291–294 (cit. on p. 6).