Real-time terrain enhancement with controlled procedural patterns

Charline Grenier, Eric Guérin, Eric Galin, Basile Sauvage

DOI: 10.1111/cgf.14992

COMPUTER GRAPHICS *forum*
Volume 43 (2024), number 1, e14992

Real-time Terrain Enhancement with Controlled Procedural Patterns

C. Grenier,[1] É. Guérin,[2] É. Galin[3] and B. Sauvage[1]

[1]Université de Strasbourg, CNRS, ICube UMR 7357, Strasbourg, France
{charline.grenier, Sauvage}@unistra.fr
[2]INSA Lyon, CNRS, UCBL, LIRIS, UMR5205, Villeurbanne, France
eric.guerin@insa-lyon.fr
[3]UCBL, CNRS, INSA Lyon, LIRIS, UMR5205, Villeurbanne, France
eric.galin@liris.cnrs.fr

Abstract

Assisting the authoring of virtual terrains is a perennial challenge in the creation of convincing synthetic landscapes. Particularly, there is a need for augmenting artist-controlled low-resolution models with consistent relief details. We present a structured noise that procedurally enhances terrains in real time by adding spatially varying erosion patterns. The patterns can be cascaded, i.e. *narrow ones are nested into large ones. Our model builds upon the Phasor noise, which we adapt to the specific characteristics of terrains (water flow, slope orientation). Relief details correspond to the underlying terrain characteristics and align with the slope to preserve the coherence of generated landforms. Moreover, our model allows for artist control, providing a palette of control maps, and can be efficiently implemented in graphics hardware, thus allowing for real-time synthesis and rendering, therefore permitting effective and intuitive authoring.*

Keywords: modelling, natural phenomena, geometric modelling, rendering, real-time rendering

CCS Concepts: • Computing methodologies → Shape modelling; Rendering; Texturing; Graphics processors; • Applied computing → Media arts

1. Introduction

Virtual terrains play a significant part in the atmosphere of a virtual scene and are essential in many domains, including the simulation and entertainment industry. Despite decades of research and technical breakthroughs, generating large terrains at a high resolution in real time remains a perennial challenge. Both large-scale structures and detailed landforms participate in the overall realism of a synthetic scene. While several advances for generating large-scale terrains conforming to orometry [AGP*19] or stream power erosion laws [CGG*17, SPF*23] have been proposed, generating plausible details remains a difficult problem. The challenge stems not only from the high-resolution models that are needed to generate details but also from the balance between realism and artistic intent. Particularly, details such as ravines obtained by surface erosion simulations [KMN88] are computationally intensive, which means they do not scale with large terrains, and often do not allow for artistic control. In contrast, user-prescribed landforms are often perceived as unrealistic or necessitate a huge amount of authoring to make them compatible with large-scale landform features. To be effective, terrain generation should combine user control and geomorphological consistency.

The purpose of terrain enhancement, often referred to as augmentation [GDGP16], is to enhance an otherwise low-resolution terrain with high-resolution details. Unfortunately, most existing terrain augmentation techniques require offline pre-processing and do not lend themselves to hardware acceleration implementation.

In this paper, we propose a procedural method for augmenting an input terrain with ravines resembling erosion phenomena, with various scales, amplitude and orientations depending on the characteristics of the terrain, in real time.

By construction, our procedural approach is resolution-independent which allows large amplification factors (up to $\times 32$ in our experiments), not bounded to any spatial range, and yet extremely compact. Moreover, synthesized details automatically adapt to low-resolution features and can be controlled by parameters including frequency, orientation and amplitude. More precisely, we present the following contributions: (1) We introduce

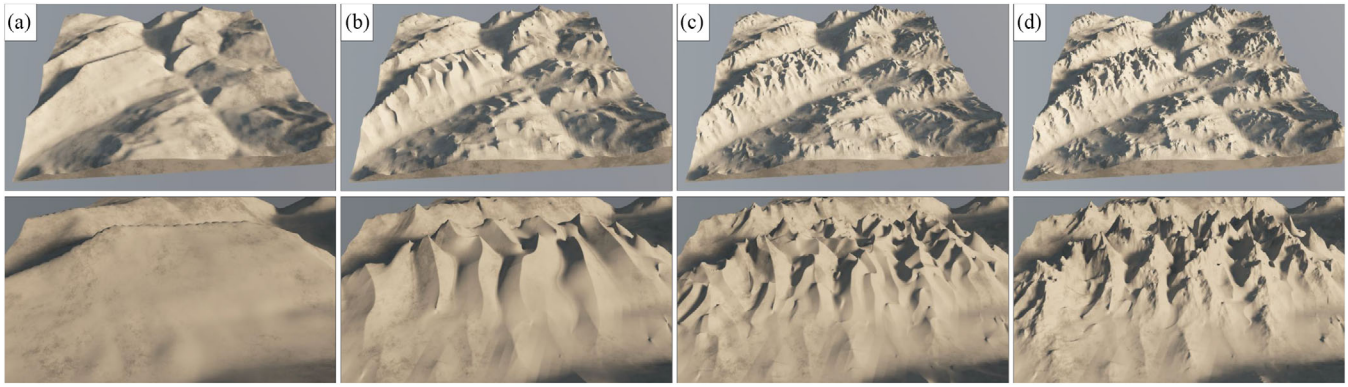


Figure 1: *Our procedural model enhances a low-resolution input terrain with oriented and spatially varying erosion patterns. (a) Input terrain h. (b) Large patterns $r^{(1)}$ are generated consistently with the terrain. (c) Narrow patterns $r^{(2)}$ are added in cascade, based on the same formulation. (d) A fractional Brownian motion (fBm) improves the appearance of crest lines.*

a Phasor noise-based method for generating controllable erosion patterns (Section 3); (2) We describe the generation of cascading patterns, *i.e*. narrow patterns nested in large ones (Section 4); (3) We validate the model using a measure of flow coherence and the comparison with existing methods (Section 5). (4) We provide a parallel and procedural implementation on GPU, which achieves real-time for both generation and rendering.

The originality of our approach is that procedurally generated erosion patterns adapt to the underlying terrain and orient in the direction of the slope. Moreover, the designer can provide low-resolution maps to control details or let the system directly derive information from the input terrain. Overall, our method yields plausible features that do not significantly hinder the hydrogeomorphological properties of the terrain.

2. Related Work

The generation of details onto an existing terrain can be interpreted as a particular case of terrain generation. A review of digital terrain modelling can be found in Galin *et al.* [GGP*19]. Here, we focus on terrain augmentation and enhancement techniques that add relief details to a previously computed topography.

Introducing details in terrains can be achieved by various techniques, often categorized into erosion simulation, data-driven approaches, or procedural methods. Contrary to tectonic erosion simulations that shape the relief of mountains after reaching an equilibrium between uplift and erosion [CBC*16, SPF*23], surface erosion methods apply a few iterations of physically inspired hydraulic or thermal erosion [MKM89] to improve the overall appearance of synthetic terrains. Variants rely either on smooth hydraulic particle systems [KBKŠ09] or shallow water approximations [MDH07, BTHB06, ŠBBK08]. Hydraulic erosion, probably the most common technique to add naturalistic details, is used intensively in the industry as a post-processing step to reproduce characteristic features such as ravines and sediment deposits at the bottom of hills or mountains. Erosion simulation is a global, iterative and computationally expensive process that cannot be employed in real time.

Among the numerous example-based approaches mostly dedicated to terrain generation [ZSTR07, GMM15, GDGP16], sparse modelling [GDGP16] stands out as a way to amplify an input low-resolution terrain with high-resolution patches from a dictionary. To avoid the lack of global structure consistency intrinsic to the local patch-matching process algorithm, Argudo *et al.* [AAC*17] extended the neighbourhood of patches. Machine learning is also a viable option that can directly infer high-resolution terrain models from their low-resolution versions [GDG*17, ZLB*19]. As for erosion simulation, example-based methods are computationally intensive and do not lend themselves to real-time applications.

In contrast, procedural approaches are computationally efficient: they allow for local elevation computation and lend themselves to parallel implementation. Fractal Brownian motion, traditionally calculated as a sum of noises at different frequencies and amplitudes [EMP*02], is commonly used to add surface details to an underlying low-resolution terrain. Even though their properties can be modulated spatially, depending on the slope or the altitude, to form a multi-fractal signal, most existing noise-based variants and combinations fail at synthesizing natural-looking landforms. The essential explanation is a lack of structure and topography foundation of noise-basis functions.

To address some of those limitations, Parberry [Par15] introduced exponential noise to better approximate the relief and slope spectra of digital elevation models. Directional ridged noise and erosive noise [DB09] introduced the concept of procedural directional noise adding details to a low-resolution elevation, and adapting surface details according to the slope. Still, the employed ridge noise does not synthesize structured details and is not adapted to terrain augmentation. Closer to our approach is the procedural model for dendritic patterns [GBG*19] which generates river networks. Unfortunately, it remains computationally intensive and does not lend itself to implementation on graphics hardware.

In the texture synthesis context, patterns that exhibit sharp features with random shapes have been generated for long using procedural noises, such as cellular noise, or a combination of a smooth random signal with a non-smooth transfer function [EMP*02]. The

latter approach has recently been improved in terms of shape and spatial control [TEZ*19, GSDT22]. Here, we build upon the Phasor noise [TEZ*19], which lends itself to producing elongated ripples with a sharp profile that conforms to designing erosion patterns. More specifically, we control spatial variations (frequency, orientation and profile) according to the characteristics of the terrain (height and gradient). We leverage the formulation of Grenier *et al.* [GSDT22] to remove unwanted discontinuities in the Phasor noise and compute terrain gradients. The actual computation is based on underlying Gaussian fields, for which several generation algorithms exist. We exploit the capability of some sparse convolution algorithms to produce spatial variations [CSM14, LSD21].

3. Model

We propose an enhancement model for terrains represented by a heightfield. Let h denote the elevation function that represents the low-resolution terrain, which maps from $\mathbf{p} \in [0, 1]^2$ to $h(\mathbf{p}) \in \mathbb{R}$. This low-resolution elevation is enhanced using another function that represents the details, called r. The final elevation is

$$e(\mathbf{p}) = h(\mathbf{p}) + r(\mathbf{p}). \tag{1}$$

Our contribution is a definition of r as a spatially varying procedural function. Through r, we aim to represent hydraulic erosion patterns, while fulfilling four conditions: (a) they contain sharp features; (b) they exhibit some randomness; (c) they are consistent with the low-resolution terrain, especially in terms of flowing properties; (d) they are controllable. To achieve this, we rely on an extension of the Phasor noise [TEZ*19, GSDT22], defined as

$$r(\mathbf{p}) = a(\mathbf{p})\, H \circ G_{f(\mathbf{p}),\theta(\mathbf{p})}(\mathbf{p}) \tag{2}$$

Recall that $\circ$ represents the composition operator. The next sections detail the terms of equation (2): G is a stochastic Gaussian field that provides randomness; H encodes the sharp transverse profile of the erosion patterns; $a(\mathbf{p})$, $f(\mathbf{p})$ and $\theta(\mathbf{p})$ are spatially varying maps that control the magnitude, spacing and orientation of the patterns, respectively. To ensure consistency between the erosion patterns r and the low-resolution terrain h, we make the three control maps depend on the altitude h, but also on the gradient ∇h. We also provide means to introduce artistic control through user maps (see Figure 2).

3.1. Background

The procedural Phasor noise [TEZ*19] is defined as a composition of a 2π-periodic profile p with a phase field ϕ:

$$\psi(\mathbf{p}) = p \circ \phi(\mathbf{p}) \tag{3}$$

Intuitively, it represents a field of waves, as shown in Figure 3. p encodes the transversal shape of the waves (*e.g.* sine, square, triangle, *etc.*), by associating a height $p(\phi)$ to any phase $\phi \in]-\pi;\pi]$. $\phi(\mathbf{p})$ defines a phase for any position $\mathbf{p}$, which can be interpreted as a position in the wave: we arbitrarily place the bottom of the wave at $\phi = 0$ and the crest at $\phi = \pm\pi$. The phase field is defined as the argument of a complex Gaussian field, *i.e.* the realization of a 2D Gaussian stochastic process. It is fully controlled by its power spectrum distribution (PSD). In this work, we use the simplest distribution, known as the bi-lobe, where the energy is concentrated on one single frequency. We denote (f, θ) the polar coordinates of the frequency (see Figure 3a). The stochastic Gaussian fields exhibit waves at frequency f, oriented according to θ.

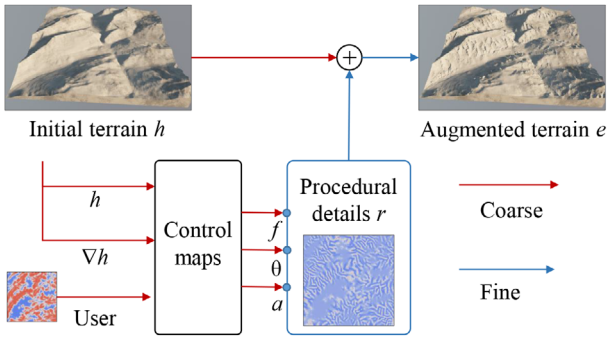


Figure 2: *Our procedural model adds fine resolution structured patterns r onto an existing, coarse resolution, terrain h. A few maps (f, θ and a) control the details according to the input terrain and to the user's desire.*

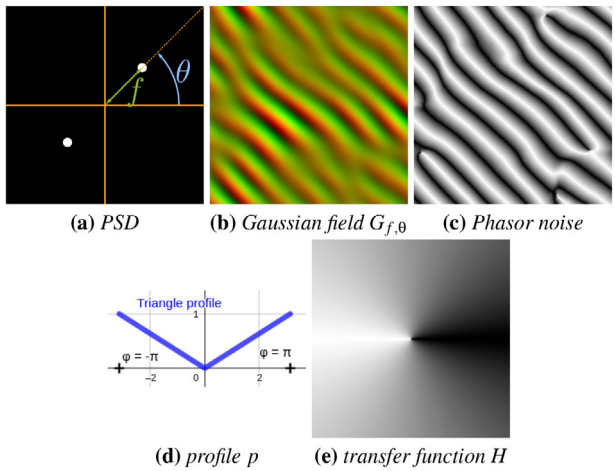
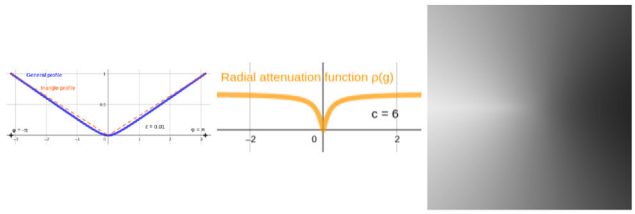


Figure 3: *Phasor noise. The power spectrum distribution (PSD) (a) fully controls a complex Gaussian field (b). The profile function (d) (triangle profile here) defines the transversal shape of the waves. The resulting field of profiled wave field (c) can be equivalently expressed as (i) the composition of the profile p with the argument of the Gaussian field $G_{f,\theta}$, or (ii) the composition of a 2D transfer function H (shown in (e), equal to p ∘ atan2) with the Gaussian field $G_{f,\theta}$ itself [GSDT22].*

In practice, for a fixed (f, θ), the computation of the phase field $\phi_{f,\theta}$ relies on the procedural generation of the Gaussian field $G_{f,\theta}$ using a local and parallel algorithm [LLDD09, GSV*14, GLM17, HN18]:

$$\phi_{f,\theta}(\mathbf{p}) = \text{atan2} \circ G_{f,\theta}(\mathbf{p}) \tag{4}$$

Here atan2 has two parameters, and the complex field is regarded as a vector field.



(a) *angular profile p* **(b)** *radial attenuation* ρ **(c)** *transfer function H*

Figure 4: *The transfer function H combines angular and radial components, as in Equation (8). (a) The angular profile p (blue, solid line) is a modified triangle profile (orange, dashed line) with a slight rounding at the bottom. (b) The radial attenuation removes the discontinuity in the middle of H.*

Grenier *et al.* [GSDT22] introduced a more generic noise model, called colour-mapped noise vector field. In this context, the Phasor noise can be reformulated as

$$\psi(\mathbf{p}) = H \circ G_{f,\theta}(\mathbf{p}) \tag{5}$$

$H = p \circ \text{atan2}$ denotes a 2D transfer function (a.k.a. colour-map). In the particular case of Phasor noise, H has no radial component, *i.e.* it is invariant by scaling. However, the model allows for more variety in H and, in this paper, we leverage it in two ways. First, we add a radial component to H, in order to remove singularities (Section 3.2). Second, we compute its derivatives in order to add smaller patterns recursively (Section 4) and to compute the shading.

It is then possible to introduce smooth spatial variations, by making f and/or θ smooth functions of the position $\mathbf{p}$. The core intuition of our work is to imitate ravines created by erosion. p controls the transversal profile of the ravines, θ controls their orientations, and f controls their scale and spacing.

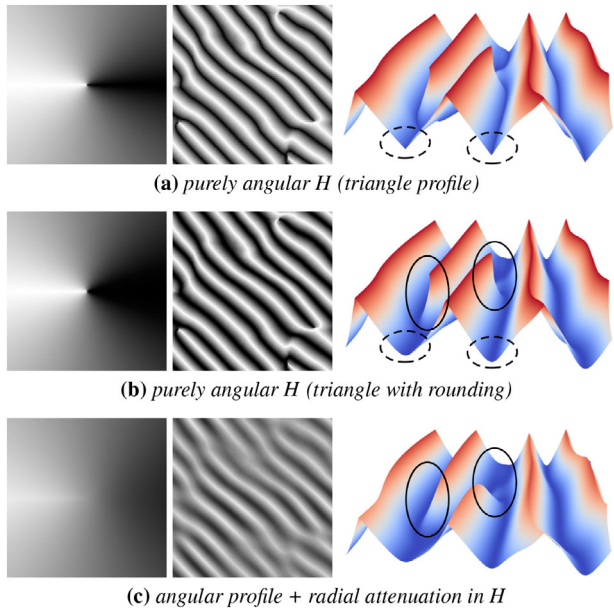
3.2. Profile design

The general profile of the ravines can be set arbitrarily, according to the desired appearance. Examples shown throughout the article were generated using a rounded triangle profile curve as illustrated in Figure 4a:

$$p(\phi) = \sqrt{(1+2\sqrt{\epsilon})\left(\frac{\phi}{\pi}\right)^2 + \epsilon} - \sqrt{\epsilon} \quad \text{with } \epsilon \ll 1. \tag{6}$$

As shown in Figure 5b compared to Figure 5a, the rounding produces a more natural appearance, similar to the effect of the streaming of water that transports and deposits sediments on the bottom of valleys.

The Phasor noise produces singularities in the phase field, which creates unwanted geometric singularities (see Figure 5b, right). This is due to the discontinuity of the arctangent, which creates a singularity at $H(\mathbf{0})$, even if p is continuous. While it is possible to remove some of them by using optimization techniques [TTZ*20], this strategy does not conform to our objectives. First, the optimization algorithm is not compatible with a purely parallel and procedural implementation, and second, it removes geomorphologically significant bifurcating patterns between ravines. We leverage the 2D transfer



(a) *purely angular H (triangle profile)*

(b) *purely angular H (triangle with rounding)*

(c) *angular profile + radial attenuation in H*

Figure 5: *Design of the profile of ravines: transfer function H (left); wave field H ∘ G in greyscale (middle) and heightfield (right). (a) The pure triangle profile is too sharp. (b) A slight rounding softens the bottom of the ravines. (c) A radial attenuation removes the discontinuities at forks.*

function formulation [GSDT22], and introduce a radial attenuation function ρ (see Figure 4b). It vanishes at $\mathbf{0}$, increases quickly, and tends to one. In practice, any function that fits these constraints can be used, we chose

$$\rho(\mathbf{g}) = \frac{2}{\pi}\text{atan}(c\|\mathbf{g}\|) \quad \text{where } c \gg 1. \tag{7}$$

Then we use ρ as an interpolating weighting function, so that $H(\mathbf{0})$ equals the average of the profile, denoted as $\bar{p}$:

$$H(\mathbf{g}) = \rho(\mathbf{g}) \cdot p \circ \text{atan2}(\mathbf{g}) + (1 - \rho(\mathbf{g})) \cdot \bar{p}. \tag{8}$$

As shown in Figure 5c, the discontinuity is removed and the forks are smoothed. Equations (6)–(8) are combined to define H in Equation (2).

3.3. Orientation, frequency and amplitude control

Here, we investigate the control of spatial variations through three control maps: orientation, frequency and amplitude. Note that the maps must vary smoothly with respect to the location $\mathbf{p}$, otherwise, the elongated erosion patterns we are looking for would be broken. Depending on the degree of control or automation desired, these maps can be either provided by the user or computed on the fly from the terrain characteristics. Here, we explain the motivations for these three maps.

The orientation map. $\theta(\mathbf{p})$ determines the direction orthogonal to the ravines and crests lines. Ravines result from erosive processes

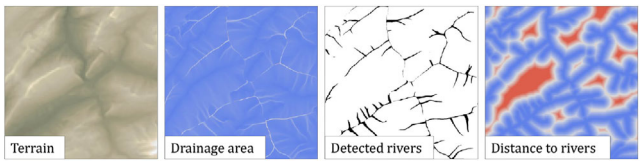


Figure 6: *Automatic amplitude extraction: given an input elevation, we compute the corresponding drainage area map, detect rivers as cells whose drainage is greater than a user-defined threshold and finally define the amplitude of details as a function of the distance to the river network.*

and globally follow the direction of the slope of the terrain. To obtain a smooth varying control map, we cannot rely directly on the gradient of the input terrain. Instead, we use a smoothed version of the terrain $\tilde{h}$ and its gradient $\nabla\tilde{h}$:

$$\theta(\mathbf{p}) = \text{atan2}(\nabla\tilde{h}(\mathbf{p})) + \frac{\pi}{2}. \tag{9}$$

The frequency map. $f(\mathbf{p})$ influences the spacing between ravines. Inferring a frequency based on the local properties of the terrain is not straightforward. From our experiments, we observed that a constant value $f(\mathbf{p})$ yields visually satisfying results. In our experiments, we set it manually when authoring the terrain features. More important is the ratio when cascading the features, as we explain in Section 4.

The amplitude map. $a(\mathbf{p})$ defines the depth of the ravines. Amplitude can be totally controlled by using a user input map. If we seek full automation, we make the following observations: first, the patterns appear on slopes and not near rivers, and second, they depend on the drainage, *i.e.* the amount of streaming water. In practice, we adopt a simple yet efficient technique to infer the presence of drainage patterns automatically, as illustrated in Figure 6. The following pre-processing step is applied to the initial low-resolution heightfield. We compute the drainage area, detect the rivers by applying a threshold value and compute a distance transform to obtain the skeleton [PDG*19]. This distance transform is directly used as amplitude map $a(\mathbf{p})$.

3.4. Crest lines

So far, the model generates patterns consistent with the terrain (Figure 1b). However, steep crest lines still exhibit unrealistic smooth regular shapes. Unlike valleys, which are smoothed by hydraulic erosion and sediment deposits, real crests exhibit irregular sharp features. We improve the appearance by adding a fractional Brownian motion $fBm(\mathbf{p})$ locally on the crests. To discriminate whether $\mathbf{p}$ is close to the crest or the centre of the valley, we compute the phase of $G_{f,\theta}(\mathbf{p})$ and apply a smooth step function s that vanishes under $\pi/4$ and equals to 1 above $3\pi/4$. Finally, we complete the details r by adding the field

$$s(|\text{atan2} \circ G_{f,\theta}(\mathbf{p})|, \pi/4, 3\pi/4) fBm(\mathbf{p}). \tag{10}$$

As shown in Figure 1d, the final appearance is significantly improved.

4. Cascading Patterns

Real terrains exhibit cascading erosive patterns, where narrow patterns are nested into large ones. To improve the realism, we propose to add a second level of detail by cascading our method (see Figure 1). Let $r^{(1)}$ denote the first level of (large) ravines defined by Equation (2), and controlled by the maps $f^{(1)}$, $\theta^{(1)}$ and $a^{(1)}$. We define a second level of (narrow) ravines defined by the same equation but different control maps $\theta^{(2)}$, $f^{(2)}$ and $a^{(2)}$.

4.1. Orientation

The orientation $\theta^{(2)}$ should follow the slope of $h + r^{(1)}$. Thus, we define

$$\theta^{(2)}(\mathbf{p}) = \text{atan2}\big(\nabla\tilde{h}(\mathbf{p}) + \nabla r^{(1)}(\mathbf{p})\big) + \frac{\pi}{2}. \tag{11}$$

While $\nabla\tilde{h}$ can easily be pre-computed, the challenge here consists in calculating $\nabla r^{(1)}$ because $r^{(1)}$ is computed procedurally. To achieve this, we assume that $a^{(1)}$ is smooth enough to be considered locally constant when deriving Equation (2):

$$\nabla r^{(1)}(\mathbf{p}) = \mathbf{J}(\mathbf{p})^T \nabla H \circ G_{f,\theta}(\mathbf{p}) \tag{12}$$

where $\mathbf{J}$ is the Jacobian of $G_{f,\theta}$. The gradient ∇H is pre-computed and stored, as it does depend neither on $\mathbf{p}$ nor on the control maps. The Gaussian field $G_{f,\theta}$ is computed procedurally for $r^{(1)}$, so it is available for free. We now derive a procedural formula for the Jacobian $\mathbf{J}$.

We compute the complex Gaussian noise using a procedural Gabor noise [LLDD09], *i.e.* , a weighted sum of Gabor kernels:

$$G_{f,\theta}(\mathbf{p}) = \sum_{\mathbf{c}} w(\mathbf{c}) e^{-\frac{\|\mathbf{p}-\mathbf{c}\|^2}{2\sigma^2}} e^{i\,\xi\,\cdot\,(\mathbf{p}-\mathbf{c})}, \tag{13}$$

where $\xi = 2\pi f[\cos(\theta), \sin(\theta)]^T$, and $\mathbf{c}$ are random positions. Then, the vector of complex derivatives $\widetilde{\mathbf{J}}$ is computed as the weighted sum of the complex gradient:

$$\widetilde{\mathbf{J}}(\mathbf{p}) = \sum_{\mathbf{c}} w(\mathbf{c}) e^{-\frac{\|\mathbf{p}-\mathbf{c}\|^2}{2\sigma^2}} e^{i\xi\,\cdot\,(\mathbf{p}-\mathbf{c})} \left(\frac{-(\mathbf{p}-\mathbf{c})}{\sigma^2} + i\xi\right) \tag{14}$$

and the 2×2 real-valued Jacobian is given by

$$\mathbf{J}(\mathbf{p}) = \begin{bmatrix} Re(\widetilde{\mathbf{J}}(\mathbf{p})) \\ Im(\widetilde{\mathbf{J}}(\mathbf{p})) \end{bmatrix}. \tag{15}$$

These formulas are used not only for computing $\nabla r^{(1)}$ but also $\nabla r^{(2)}$, which is needed for shading at the rendering stage.

4.2. Frequency

Our model takes inspiration from Horton's law [Hor45]: the numbers of streams of different orders in a given drainage basin tend to approximate a geometric series in which the ratio is the bifurcation ratio. This provides us with a hint on the ratio between two ravine frequencies of different orders, *i.e.* between $f^{(1)}$ and $f^{(2)}$. In our experiments, we used a ratio $f^{(2)} \approx 3f^{(1)}$. The same relation can be

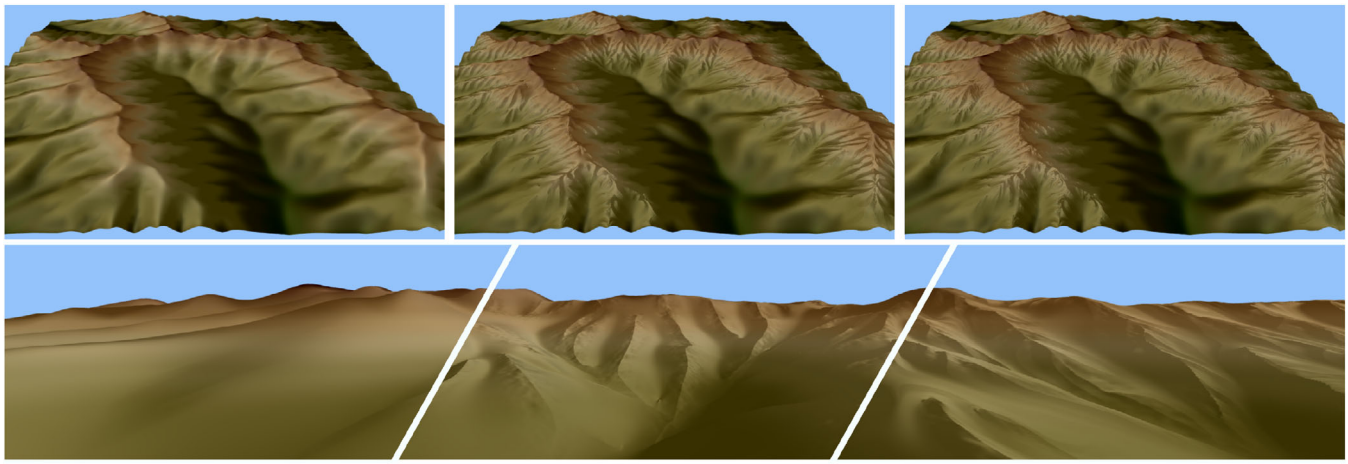


Figure 7: *Global view (top) and panoramic close-up (bottom) of a* $20 \times 20\,\mathrm{km}^2$ *terrain. We amplified the initial low-resolution terrain (left image) with one (centre image) and two (right image) levels of ravines, along with ridged noise over the crests. The input terrain results from a tectonic erosion simulation [SPF*23] and has a* 512^2 *resolution; the output has a* 4096^2 *resolution, a* $\times 8$ *amplification factor.*

used to determine the initial frequency $f^{(1)}$: we evaluate the initial frequency of mountains in the input terrain and apply a ratio.

4.3. Amplitude

To decide about the amplitude map $a^{(2)}$, we use two geomorphological arguments. First, $r^{(1)}$ and $r^{(2)}$ should appear at similar locations and be roughly proportional. Second, $r^{(2)}$ should vanish at the bottom of $r^{(1)}$ because the water streams here. Our solution is

$$a^{(2)}(\mathbf{p}) = \lambda r^{(1)} \tag{16}$$

The parameter λ is a proportional factor empirically set to $\lambda = 0.2$, and $r^{(1)} = a^{(1)} H \circ G^{(1)}$ contains both the spatial variations $a^{(1)}$ and the vanishing at the bottom encoded in the function H.

5. Results

We implemented a parallel version of our terrain amplification method on graphics hardware. We provide the code, consisting of a web page containing WebGL objects and shaders, as supplementary material. Figure 7 shows a direct snapshot of the WebGL implementation. It illustrates the ability of our method to augment large terrains with plausible details. Figures 8–11 illustrate amplified ray-traced terrains.

5.1. Performance

To facilitate experimentation, accessibility and portability, we implemented our method on WebGL2.0. The pipeline consists of three stages. **(1) Initialization**. The input textures are stored in 8-bit images. The initial low-resolution terrain h and its gradient ∇h have a 400^2 pixel resolution; the amplitude control map a has a 100^2 pixel resolution; the frequency is a scalar variable directly in the code. So the total input storage is less than 1MB. **(2) Generation**. Intermediate computations are done in Frame Buffer Objects (FBO) and stored in 2048^2 pixel textures with floating-point precision. A first pass computes all details $r^{(1)}$, $r^{(2)}$, $\nabla r^{(1)}$, $\nabla r^{(2)}$ and fBm. A second pass computes the final terrain e and its gradient ∇e. **(3) Rendering**. A 512^2 pixel vertex plan is enhanced with displacement and normal maps defined from e and ∇e.

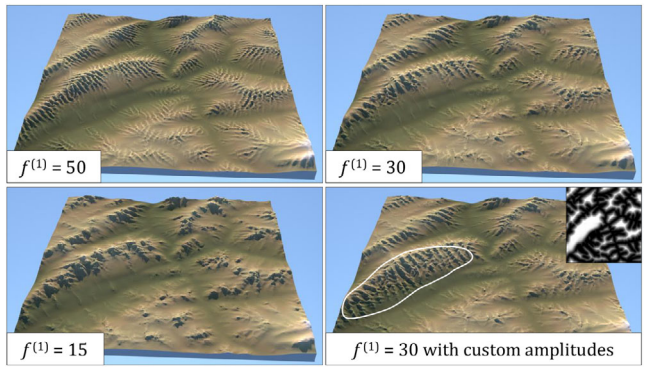


Figure 8: *Control is achieved by tuning several parameters. Here we show the impact of the first-level frequency and the use of a custom map for the amplitude. We also show how a custom amplitude map can be used to locally modify the amount of detail.*

In this pipeline, the initialization is computed once, the generation is computed on demand when the details need updating, and the rendering is computed at each frame. We carried out comparative analyses on different GPUs: NVidia GTX 980; NVidia RTX A2000; Intel mesa UHD graphics 630; AMD Radeon R9 200. For all the GPUs, the interface reaches 60 FPS. The initialization takes less than 1.7 ms; the generation takes less than 1.2 ms; the rendering stage is standard, and the overhead is due to displacement and normal maps only. This validates the real-time capability, however these measures have been done directly in a web browser, so they lack precision. For a more precise measure, we generated 100 times the details with a resolution of 2048^2 pixel on NVidia GTX 980.

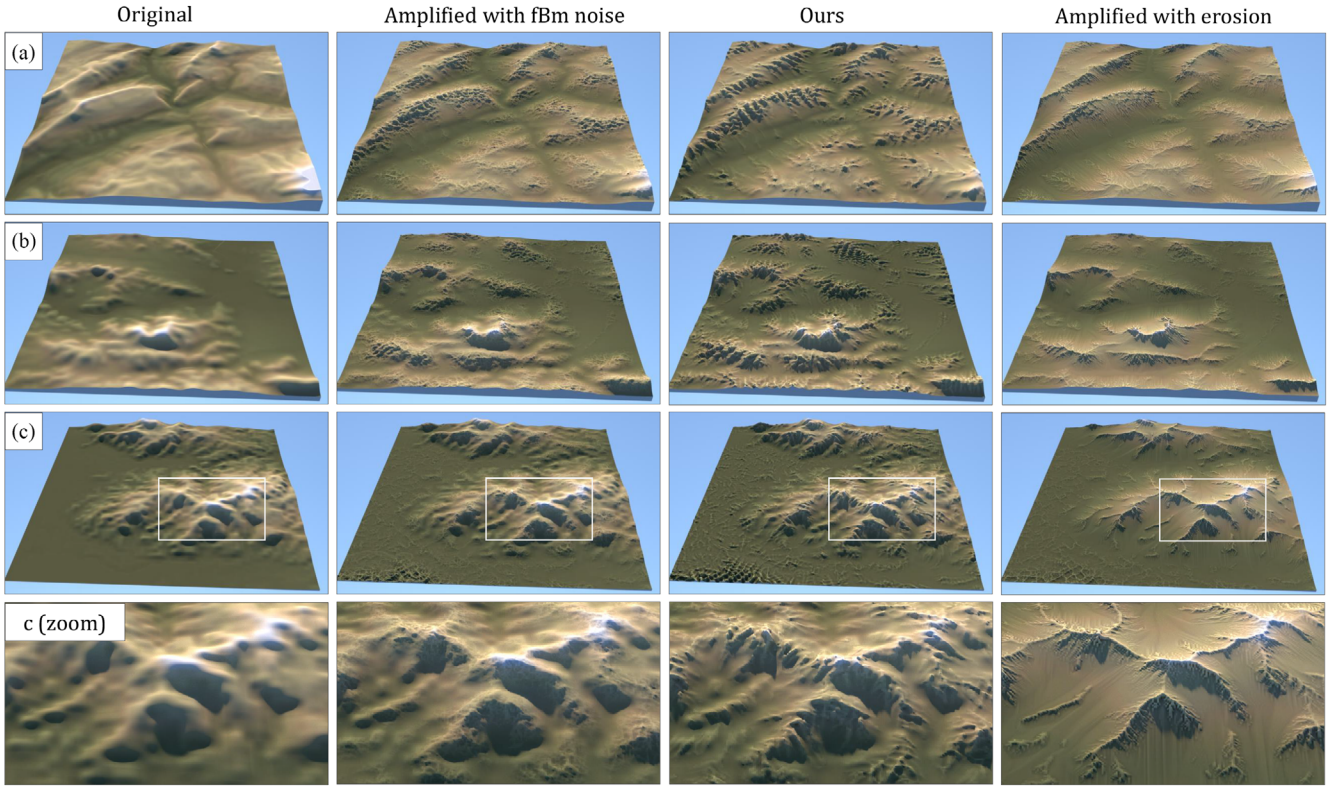


Figure 9: *Comparison of our amplification algorithm, standard fBm noise and erosion simulation: rows a, b and c show different input heightfields, and the last row displays closeups.*

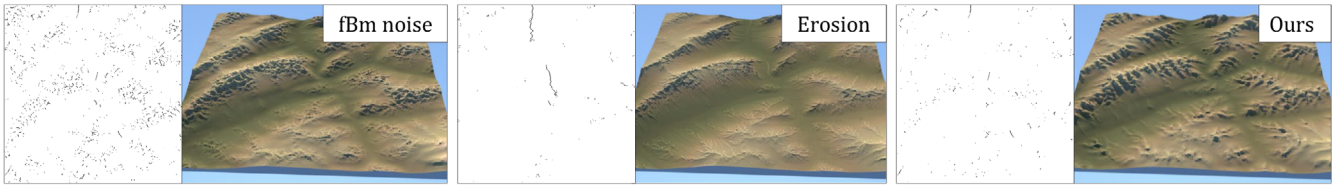


Figure 10: *Comparison of the flow quality using breaching maps: black cells indicate removed bedrock to make water flow. Fractional Brownian motion (left) creates many local pits; erosion simulations (centre) reduce the number of pits but can create inconsistent drainage in valleys because of the sediment deposition process. Our method (right) also limits flowing inconsistencies, mainly caused by the ridged noise on crests.*

Our method takes 4 ms for 100 generations. As a comparison, a 6-octaves Fractional Brownian motion takes 3 ms.

5.2. Control

Our method authorizes user control through parameters and maps that can be automatically computed or assigned to default values. One of the most influencing parameters is the base frequency that defines at which scale the augmentation algorithm introduces the first level of details. Figure 8 illustrates the dramatic impact of this parameter on the landforms created over the base terrain, starting from small erosion patterns ($f^{(1)} = 50$) to large mountain crests ($f^{(1)} = 15$). Another central parameter is the amplitude map, which defines the elevation range of added landforms. Section 3.3 proposes a simple yet effective way to initialize this map, relying on the distance to rivers. This can be further modified by the user to locally add details on specific areas, as illustrated in Figure 8 (bottom right).

5.3. Comparison and validation

The functions $r^{(1)}$ and $r^{(2)}$ can be computed in parallel and in real time on graphics hardware, while their location is controlled through the amplitude map. In the case of terrain amplification, only noise-based methods [EMP*02] meet similar requirements. Figure 9 compares low-resolution terrains (first column) enhanced with standard fractional Brownian motion using the ridge noise (second column) and our method (third column). For a fair comparison, we used the same amplitude map. Ridge noise adds isotropic details whose

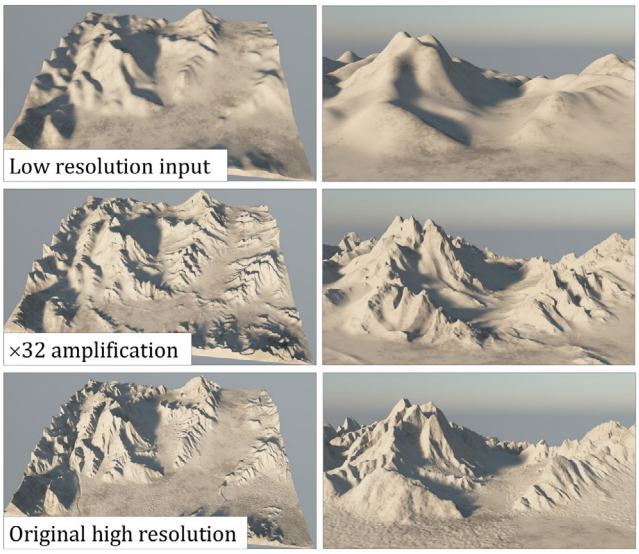


Figure 11: *Example of detail enhancement involving a ×32 amplification factor. While our method cannot recover original details, it adds plausible features.*

structure or patterns do not align with the slope. In contrast, our method adapts the ridges and ravines to the slope of the underlying low-resolution terrain.

Figure 9 also shows a comparison with features obtained by erosion simulation (fourth column) which is interactive yet not real-time. When applied on a smooth input terrain, erosion algorithms tend to carve the steep slopes with small ravines and fill valleys with sediments, but not create ravines at different spatial frequencies. The latter are obtained when high-resolution details are present in the input. For the sake of comparison, as a pre-processing step before erosion, we amplified the low-resolution terrain with noise-based high-resolution details localized according to our amplitude map. Finally, the erosion process significantly re-shapes the terrain. In contrast, our method guarantees that the low-resolution topography is preserved.

We tried to assess the plausibility of our approach by comparing synthesized details to real terrains. Starting from a high-resolution digital elevation model (Figure 11) as ground truth reference (bottom), we down-sampled it to a low-resolution version (top) and applied amplification strategies (middle) with a high amplification factor of ×32. Even though the details do not exhibit as complex features and patterns, our method preserves the major landforms of the initial high-resolution terrain and amplifies it with plausible relief details, particularly on some initially round hills (right column). This is due to the automatic computation of the amplitude map: authoring could improve the matching with the characteristic of the low-resolution terrain. High-frequency details in flat regions do not result from erosion and are not reproduced.

Another validation consists in analysing the coherence of the flow that our method produces. To do so, we use the breaching algorithm and evaluate how much bedrock needs to be removed to guarantee that every drop of water eventually flows outside the domain [SPF*23]. Table 1 and Figure 10 compare this amount for different methods: fBm, erosion simulation, and ours. Unsurprisingly, the simulation gives better results since it is meant to favour flowing. Noise introduces a lot of local pits in the terrain and thus is less adapted to naturally let the water flow. Our method gives intermediate results between the two.

Table 1: *Amount of bedrock removed by the breaching algorithm to guarantee a consistent water flow onto the terrain (Figure 9).*

Terrain	fBm	Erosion	Ours
a	7.4	2	2.1
b	8.9	4.8	6.8
c	128	25	86

5.4. Limitations and extensions

Controlling the impact of spatial frequencies and the ratio between ravine frequencies would be worth investigating. In this work, we focused on spatial variations in terms of orientation θ and amplitude a. Variations of frequencies f remain a not fully exploited capability. Exploring variants in the profile curve would also probably allow for more diverse ravine landforms. However, this would impact the transfer function H, which introduces challenges for storage and derivation. Undoubtedly, such control tools may extend the range of effects and improve artistic creation. However, a complete user study with professional artists would be necessary to validate the effectiveness of those controls, which is beyond the scope of this work.

The proposed method operates on a heightfield representation and therefore inherits the classic limitations of function-based elevation representations which do not lend themselves to modelling overhangs or even steep slopes where considerable distortions may emerge. Not adapted to representing table mountains featuring plateaus and cliffs, it is ill-suited to amplify sedimentary valleys or flood plains with meandering rivers: in that case, the proposed gradient-aligned noise function replicates linear patterns over almost flat regions.

Finally, contrary to computationally intensive erosion simulation methods, our method cannot generate a global coherent river network, a common limitation of all function-based elevation models relying on a fully procedural and parallel algorithm.

In its current development, our method generates geometric details procedurally in real time. Filtering techniques would be advantageous for better integration in the rendering pipeline. While readily usable for albedo data [GSDT22], adapting existing algorithms to elevation remains challenging because of visibility and masking issues.

6. Conclusion

We have introduced a procedural way to generate oriented erosive patterns on terrains. This is done using an improved Phasor noise guided by a series of control maps and parameters. These control

maps can be either totally automated from the low-resolution terrain, or provided by the user for more accurate control over the process. The implementation demonstrates the efficiency of the method. Our terrain enhancement has an advantage over classical fBm noise: thanks to its alignment with initial terrain slopes, it does not break the water flowing. Compared to the expensive erosion simulation, it also has the benefit of preserving the original terrain features.

Acknowledgements

This work has been funded by the projects ReProcTex (ANR-19-CE33-0011-01), Ampli (ANR-20-CE23-0001), and EOLE (ANR-23-CE56-0008-02) from the Agence Nationale de la Recherche.

References

[AAC*17] Argudo O., Andujar C., Chica A., Guérin E., Digne J., Peytavie A., Galin E.: Coherent multi-layer landscape synthesis. *The Visual Computer 33*, 6–8 (2017), 1005–1015.

[AGP*19] Argudo O., Galin E., Peytavie A., Paris A., Gain J., Guérin E.: Orometry-based terrain analysis and synthesis. *ACM Transactions on Graphics 38*, 6 (2019), 1–12.

[BTHB06] Beneš B., Těšínský V., Hornyš J., Bhatia S. K.: Hydraulic erosion. *Computer Animation and Virtual Worlds 17*, 2 (2006), 99–108.

[CBC*16] Cordonnier G., Braun J., Cani M.-P., Benes B., Éric Galin, Peytavie A., Éric Guérin: Large scale terrain generation from tectonic uplift and fluvial erosion. *Computer Graphics Forum 35*, 2 (2016), 165–175.

[CGG*17] Cordonnier G., Galin E., Gain J., Benes B., Guérin E., Peytavie A., Cani M.-P.: Authoring landscapes by combining ecosystem and terrain erosion simulation. *ACM Transactions on Graphics 36*, 4 (2017), 134.

[CSM14] Charpenay V., Steiner B., Musialski P.: Sampling gabor noise in the spatial domain. In *Proceedings of the 30th Spring Conference on Computer Graphics* (2014), pp. 79–82.

[DB09] De Carpentier G. J., Bidarra R.: Interactive GPU-based procedural heightfield brushes. In *Proceedings of the 4th International Conference on Foundations of Digital Games* (2009), pp. 55–62.

[EMP*02] Ebert D. S., Musgrave F. K., Peachey D., Perlin K., Worley S.: *Texturing and Modeling: A Procedural Approach* (3rd edition). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2002.

[GBG*19] Gaillard M., Benes B., Guérin E., Galin E., Rohmer D., Cani M.-P.: Dendry: A procedural model for dendritic patterns. In *I3D'19: Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games* (Montreal, Quebec, Canada, 2019), ACM.

[GDG*17] Guérin E., Digne J., Galin E., Peytavie A., Wolf C., Benes B., Martinez B.: Interactive example-based terrain authoring with conditional generative adversarial networks. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia 2017) 36*, 6 (2017), 228:1–228:13.

[GDGP16] Guérin E., Digne J., Galin E., Peytavie A.: Sparse representation of terrains for procedural modeling. *Computer Graphics Forum (Proceedings of Eurographics 2016) 35*, 2 (2016), 177–187.

[GGP*19] Galin E., Guérin E., Peytavie A., Cordonnier G., Cani M.-P., Benes B., Gain J.: A review of digital terrain modeling. *Computer Graphics Forum 38*, 2 (2019), 553–577.

[GLM17] Galerne B., Leclaire A., Moisan L.: Texton noise. *Computer Graphics Forum 36*, 8 (2017), 205–218.

[GMM15] Gain J. E., Merry B., Marais P.: Parallel, realistic and controllable terrain synthesis. *Computer Graphics Forum 34*, 2 (2015), 105–116.

[GSDT22] Grenier C., Sauvage B., Dischler J.-M., Thery S.: Color-mapped noise vector fields for generating procedural micro-patterns. *Computer Graphics Forum 41* (2022), 477–487. ISSN: 1467-8659.

[GSV*14] Gilet G., Sauvage B., Vanhoey K., Dischler J.-M., Ghazanfarpour D.: Local random-phase noise for procedural texturing. *ACM Transactions on Graphics 33*, 6 (Nov. 2014), 195:1–195:11. (Proceedings of SIGGRAPH Asia'14).

[HN18] Heitz E., Neyret F.: High-performance by-example noise using a histogram-preserving blending operator. *Proceedings of the ACM on Computer Graphics and Interactive Techniques 1*, 2 (2018), 1–25.

[Hor45] Horton R. E.: Erosional development of streams and their drainage basins; Hydrophysical approach to quantitative morphology. *GSA Bulletin 56*, 3 (Mar. 1945), 275–370. ISSN: 0016-7606.

[KBKŠ09] Krištof P., Benes B., Křivánek J., Šťava O.: Hydraulic erosion using smoothed particle hydrodynamics. *Computer Graphics Forum 28*, 2 (2009), 219–228.

[KMN88] Kelley A. D., Malin M. C., Nielson G. M.: Terrain simulation using a model of stream erosion. *Computer Graphics 22*, 4 (1988), 263–268.

[LLDD09] Lagae A., Lefebvre S., Drettakis G., Dutré P.: Procedural noise using sparse gabor convolution. *ACM Transactions on Graphics (TOG) 28*, 3 (2009), 1–10.

[LSD21] Lutz N., Sauvage B., Dischler J.-M.: Cyclostationary Gaussian noise: Theory and synthesis. *Computer Graphics Forum (Proceedings Eurographics) 40*, 2 (2021), 239–250.

[MDH07] Mei X., Decaudin P., Hu B.-G.: Fast hydraulic erosion simulation and visualization on GPU. In *PG'07: 15th Pacific*

Conference on Computer Graphics and Applications (2007), pp. 47–56.

[MKM89] Musgrave F. K., Kolb C. E., Mace R. S.: The synthesis and rendering of eroded fractal terrains. In *SIGGRAPH '89: Proceedings of the 16th Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 1989), Association for Computing Machinery, pp. 41–50.

[Par15] Parberry I.: Modeling real-world terrain with exponentially distributed noise. *Journal of Computer Graphics Techniques (JCGT) 4*, 2 (May 2015), 1–9. ISSN: 2331-7418.

[PDG*19] Peytavie A., Dupont T., Guérin E., Cortial Y., Benes B., Gain J., Galin E.: Procedural riverscapes. *Computer Graphics Forum 38*, 7 (2019), 35–46.

[ŠBBK08] Šťava O., Benes B., Brisbin M., Křivánek J.: Interactive terrain modeling using hydraulic erosion. In *Proceedings of Symposium on Computer Animation* (Dublin, Ireland, 2008), pp. 201–210.

[SPF*23] Schott H., Paris A., Fournier L., Guérin E., Galin E.: Large-scale terrain authoring through interactive erosion simulation. *ACM Transactions on Graphics 42* (2023), 1–15.

[TEZ*19] Tricard T., Efremov S., Zanni C., Neyret F., Martínez J., Lefebvre S.: Procedural phasor noise. *ACM Transactions on Graphics 38*, 4 (July 2019), 1–13.

[TTZ*20] Tricard T., Tavernier V., Zanni C., Martínez J., Hugron P.-A., Neyret F., Lefebvre S.: Freely orientable microstructures for designing deformable 3D prints. *ACM Transactions on Graphics (TOG) 39*, 6 (2020), 1–16.

[ZLB*19] Zhao Y., Liu H., Borovikov I., Beirami A., Sanjabi M., Zaman K.: Multi-theme generative adversarial terrain amplification. *ACM Transactions on Graphics 38*, 6 (Nov 2019), 1–14.

[ZSTR07] Zhou H., Sun J., Turk G., Rehg J. M.: Terrain synthesis from digital elevation models. *IEEE Transactions on Visualization and Computer Graphics 13*, 4 (2007), 834–848.

Supporting Information

Additional supporting information may be found online in the Supporting Information section at the end of the article.

Supporting Information