Pacific Graphics 2024
R. Chen, T. Ritschel, and E. Whiting
(Guest Editors)

Volume 43 (2024), Number 7

Procedural underwater terrain generation with Environmental Objects

paper1524

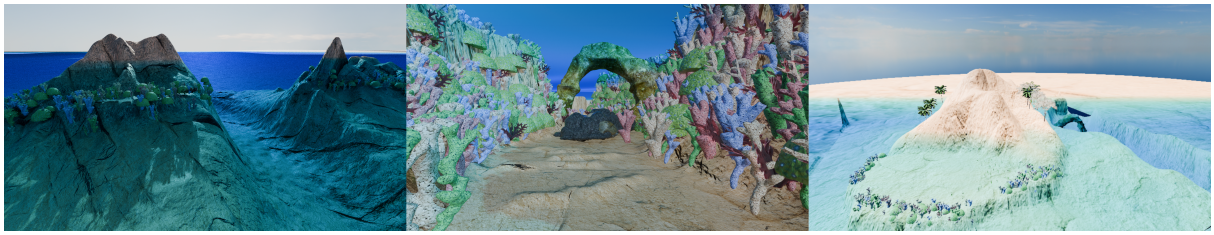

Figure 1: *Our method can produce different scenes including coral islands and canyons at multi-scale using environmental objects to represent terrain features.*

Abstract

Simulating underwater landscape growth is complex due to the interplay of biological, environmental, physical, geological, and human factors. Our method addresses this complexity by introducing a sparse representation of the terrain elements using environmental objects, defined as parametric models based on point, curve, or region skeletons, interact locally to spawn, grow, and die. This approach removes the need for complex interconnections, enabling a parallelizable and scalable generation process. Using fitting functions to incorporate biological rules and focusing on geological plausibility, our method enables interactive underwater landscape simulations. This work's main contributions are the introduction of sparse terrain element representation, the use of fitting functions to simulate biological processes without complex physics, and the development of an interactive simulation framework based on geological events.

CCS Concepts

• ***Computing methodologies*** → *Description logics; Multi-agent systems;* ***Model development and analysis; Simulation types and techniques; Shape modeling;***

1. Introduction

Automated terrain generation is a key component of natural scene digital modeling for animated movies and video games. Many landscapes have been studied and are synthesised with more and more realism. Different processes can be used and combined to achieve these scenes: fractal terrains Musgrave et al., 1989; Prusinkiewicz and Hammel, 1993, erosion simulation Cordonnier et al., 2023; Mei et al., 2007, manual modeling de Carpentier and Bidarra, 2009; Guérin et al., 2022, geological simulation Cortial et al., 2019; Cordonnier et al., 2017, ... The high quality of synthesis for such environment is due to the possibility to observe these environments from many point of views: long-distance gazing, hiking on mountains, remote sensing, aerial imaging, ... Thanks to the quality of the digital modeling, the entertainment industry display often breathtaking land scenes.

Underwater scenes are rarely created in these media for multiple reasons: these environments are not completely understood and mastered as much as land environments because they are difficult to access, we lack the capacity to see them at a larger scale (unlike mountains for example) and the underlying process that forms these landscapes are much more complex to simulate.

These limitations cause animated movies and video games studios to avoid as much as possible underwater environments.

However, these environments are important for the study of biology, geology, and, by extension, robotics. Due to the complexity, limitations and danger of underwater human operations, underwater robots are more and more used for marine environment monitoringMaslin et al., 2021; Williams et al., 2016; Dunbabin et al., 2020; Palmer et al., 2021. Yet the validation process of such underwater robot is expensive, with heavy logistic, and it is often impossible to find the appropriated environment to test the system. Thus, underwater robotics requires simulation capacities, to be able to test the

robot's algorithms in very specific environment and conditions. But for now, roboticians are lacking the capacity to test on realistic virtual scenes, and so only test them on synthetic scenarios that do not correlate with real world terrains. For that, they need to manipulate more precisely the characteristics of the underwater environment, at different scales.

The difficulties to visualize and study the underwater environments on a large scale at the same time as a small scale is an obstacle to the procedural generation and simulation of scenes that are coherent in these two different scales. A solution to this problem could be a bottom-up approach, simulating at the smallest scale the behaviour of all the elements of the environment. Computing such a simulation in order to generate an entire ecosystem is an near-impossible task due to time and memory complexity.

The method we propose provides a way to procedurally generate environments on multiple scales without introducing such complexity by using a sparse representation of the environment using environmental objects. The environmental objects only have access to local values of the environments to spawn, grow and die. The use of local interaction with environment values removes the need for complex interconnections between all elements of the terrain, providing a parallelisable generation process at large to small scales. Defining environmental objects of the terrain as parametric models based on point, curve or region skeletons provides a lightweight representation of the terrain that the user can interact with. The interaction with the simulation process is still a difficult task Smelik et al., 2014. Our method does not aim for a visually realistic generation, but for a plausible terrain depending on geological and biological constraints, guided by the user. Using state of the art modeling of the geometry of the environmental objects of the terrain could achieve realistic results. We illustrate the method through the generation of coral islands and reefs.

Our main contribution are the introduction of a sparse representation of the terrain elements as environmental objects, the use of fitting functions to incorporate biological rules in the simulation process avoiding physic simulation and an interactive simulation based on geological events for underwater landscapes.

2. Related works

Procedural terrain generation has been heavily studied for the last 40 years Galin et al., 2019. Researches in this topic try to find new solutions to compromise between realism, user control and efficiency Gain et al., 2009. Using fractal noise parametrized to resemble real landscape has been an important first step Musgrave et al., 1989 as it's a fast and light solution to generate procedurally the appearance of mountains. The lack of user control pushed newer works toward the use of controlled noise by including real DEM in the process through learning Kapp et al., 2020; Brosz et al., 2007, while the rise of deep learning technologies gave higher control to the user through sketches Guérin et al., 2017; Talgorn and Belhadj, 2018.
By including expert knowledge of tectonic process and subsurface geology, some algorithms tend to get more realistic Patel et al., 2021; Cortial et al., 2019; Michel et al., 2015.
While these algorithms are able to generate large-scale landscapes, the finer details of the terrain is often computed by the use of erosion simulation Cordonnier et al., 2023; Schott et al., 2023; Paris et al., 2019. This process can be expensive in time but results in more plausible surfaces.

All the algorithms aim to reproduce plausible relief in terrestrial landscapes, mostly limited to alpine landscapes, but a lack of research can be found in almost all other biomes. Underwater landscapes generation, for example, has been almost completely absent from literature for many reasons: the difficulty of accessing the area, the lack of visibility under water and the complex physics of underwater geology and biology make the algorithms adapted for this environment scarce.
The majority of the ocean floor can be represented as a fractal terrain Mareschal, 1989. While stochastic noise can be sufficient to model the ocean floor, this process won't cover areas with the biggest biomass, near shallower waters such as near coasts and islands.
Due to the impossibility to observe the large-scale and the small-scale of underwater environments, some works related to geology model large structures like the profile shape of the coral reef Bosscher and Schlager, 1992, simulate its surface growth Li, 2021, or use procedural algorithms for single polyp Abela et al., 2015. We however don't have a mix of the different scales, and neither methods take into account the environment such as the topography or the interaction of different terrain elements. This is mainly due to the fact that the evolution time for each scale varies from a span of weeks to thousands of years.

In an ecosystem, any element of the system has an impact on their surrounding. Simulating each physical properties such as shading, heat, humidity may require enormous computation power. By considering these properties as scalar fields surrounding the whole scene, and that elements of the terrain affect locally the scalar fields, we can simplify the computation of the physical properties of the environment Grosbellet et al., 2016; Guérin et al., 2016. This process provides a scalable system from which scene details are rendered in a plausible way. In a similar way, other works represent the wind flow as a composition of local vector fields Wejchert and Haumann, 1991, avoiding complex fluid simulation while providing user control in a lightweight model. We extend these works by incorporating a time-evolution system such that the scene can be dynamic.

3. Method

The overall pipeline of the method is based on simple incremental generation like most rule-based systems. In this type of system, the final state is defined either by reaching equilibrium, or by verifying specific conditions, such as a maximum number of iterations. We define our pipeline in three phases (Figure **??**): the initialization phase that describe the generation and simulation rules, the iterative phase generating populating the terrain with our environmental objects and finally the output.

3.1. Pipeline overview

The generation of the terrain is initialized using an initial height field $\mathcal{H}$ and a water level $\mathcal{L}$. The height field provides variation

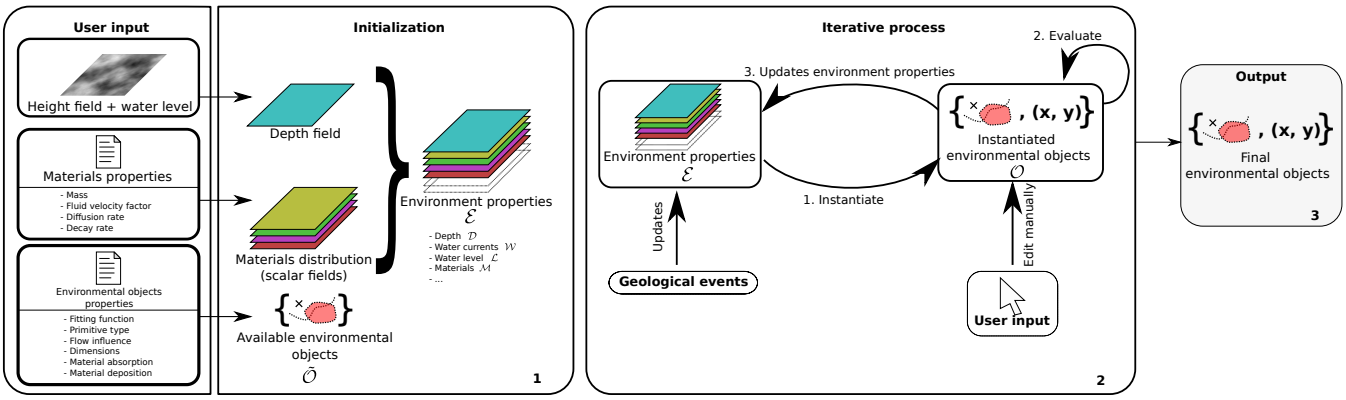

Figure 2: *Overview of the pipeline of the method. The user provides as input an initial height field and sets the water level, as well as a definition of the materials properties and environmental objects properties that will be used in the iterative process. These inputs are initialized as an initial set of environmental objects and scalar fields that represents the environment values. In the iterative loop, new environmental objects are instantiated using the current state of the environment at their optimal position. The existing environmental objects in the terrain reevaluate their fitting function to grow or die and update the environment values locally. At each iteration, geological events can update the environment values, while the user can interact directly with the environmental objects. The result of the whole process is a set of environmental objects which is a sparse representation of the elements of the scene.*

on the depth $\mathcal{D}$, which can influence the generation process of the scene. We set $\mathcal{D} = \mathcal{H} - \mathcal{L}$.
The list of available environmental objects $\tilde{\mathcal{O}}$, representing the different elements that can be present in the scene, are provided with their properties: type, size, generation rules, growing conditions and effects on the environment values (Section **??**).
Finally, different materials can be defined with their properties such as diffusion speed, mass, damping factor and influence from the water currents. Materials distributions are represented as a scalar field $\mathcal{M} : \mathbb{R}^2 \to \mathbb{R}$ and water currents as a vector field $\mathcal{W} : \mathbb{R}^2 \to \mathbb{R}^2$ that can be evaluated by the environmental objects of the scene to simulate their growth and spawn at the most probable position. The environmental properties $\mathcal{E} = (\mathcal{D}, \mathcal{W}, \mathcal{L}, \mathcal{M})$ is composed of depth, water currents, water level and materials distribution information at any point of the terrain (Section **??**).
The definition of environmental objects' properties and environment properties is done with field experts, providing the pertinent parameters required to model the evolution of the terrain elements using expert knowledge (Section **??**). Additional properties can easily be added to the environment properties $\mathcal{E}$ in order to fit to the experts needs, such as atmospheric pressure, humidity, temperature, ...
The generation phase can optionally be started with an initial set of environmental objects present in the scene.

Once the initialization phase is done, the generation begins. The generation process is incremental and its main loop is composed of two different steps: the instantiation of new environmental objects then the update of the environment. At each iteration, new environmental objects can be created at their most fitting locations if possible. The generation rules provided in the initialization phase are used to find the optimal position from stochastic sampling (Section **??**). All environmental objects are evaluating their state analytically using the fitting function provided as input (Section **??**). Once the instantiation step is done, the environment's values are updated by each environmental object by deposing and absorbing some of the available materials (Section **??**) while modifying the water currents (Section **??**) around them. Finally, water currents and height field's gradient displace materials of the terrain at each iteration. During the generation process, the user can alter directly the distribution and shapes of the environmental objects (Section **??**) and perturb the generation process by planning geological events that have impacts on the environment values (Section **??**).

The output of our system is a set of environmental objects disposed in the plane. We do not provide the 3D representation of the environmental objects, letting the user define the rendering method. The figures used in the paper use a mix of implicit surfaces and triangular meshes.

3.2. Environmental objects

Environmental objects are rule-based objects following rules depending on their local environment for evaluation their state in their life cycle. We can see them as a life form in the way that they are created and eroded with time. During their lifetime, they influence their local environment by depositing and absorbing materials around them and influencing the water currents. The environment objects are described spatially as a single point, a parametric curve or a region.

We consider that all environmental objects follow a life cycle of spawning, growing and dying. While many environmental objects of a terrain is not a living being, we assume that evolution of relief, for example, starts at one point in time, grow as the geological factors force it to and is eroded until a point where this environmental object can not be distinguished from the rest of the environment.

Environmental objects are spawn stochastically in the terrain at

the optimal fitting position. This position is determined from a generation rule given by the user for each of the environmental objects, which is dependant on the environment state. Once the environmental object is present in the scene, it will continuously evaluate its fitting function to determine its state in the life cycle. If the evaluation results as less than zero, the environmental object dies and it is removed from the list of environmental objects present in the scene. While the environmental object remains, it will continue influencing its environment, by absorbing and depositing material around it and by influencing the water currents.

3.2.1. Generation rules

Generation rules provides, for each environmental object, a fitting function Γ_{obj} defining the most probable location for an environmental object to spawn. Fitting functions' parameters contains, for every point p, the environmental values $\mathcal{E}_p$ (the amount of each material available $\mathcal{M}(p)$ and the velocity and direction of the water currents $\mathcal{W}(p)$) and information about surrounding environmental objects $\mathcal{O}$ (signed distance from the closest punctual environmental object or curve defining curve- and region-based environmental objects, curvature of the curve, and start and end points of the curve-based environmental objects).

The seed point of a spawning environmental object is defined by a stochastic sampling of the plane. We propose different optimization means to find the optimal fitting position, depending on the environmental object shape.

The spawning position of a punctual environmental object is found at the local maxima of the fitting function from a seed point. The optimisation process simply follows the field's gradient $\nabla\Gamma_{obj}$ until the local maxima is reached.

A region is defined as an isocontour of the field for which the target area A is found. From the seed point, we follow the isolevel of the fitting function $\nabla\Gamma^{\perp}_{obj}$ until a loop is created to define the initial condition of the shape. Using the Active Contours algorithm, we can optimize the region's energy defined as $E = E_{internal} + E_{shape}$ with

$$E_{internal} = \frac{1}{2}\left(\alpha(t)\left\|\frac{dv}{dt}(t)\right\|^2 + \beta(t)\left\|\frac{d^2v}{dt^2}(t)\right\|^2\right). \quad (1)$$

The internal energy $E_{internal}$ force the shape compact while the shape energy E_{shape} force the shape into a specific target. For the environmental objects generated in the following examples, we used a constraint on a target area.

$$E_{shape} = (A-a)^2$$

with a the current area of the shape and A the target area, provided by the user for each shape, with some randomness.

A curve have different generation rules. It can either follow the gradient of the fitting function $\nabla\Gamma_{obj}$, follow the isocontour $\nabla\Gamma^{\perp}_{obj}$, or follow the heat points. While the first two possibilities are trivial, the later can also be optimized using the Active Contours algorithm by optimizing the energy $E = E_{internal} + E_{shape}$ with Equation (**??**). We applied a length constraint on the curves :

$$E_{shape} = (L-l)^2$$

with l the curve's length and L the target length. This algorithm is sensible to the initial shape of the curve, so we start with a straight line following the isolevel at the seed point.

3.2.2. Environment object's evaluation

Environment objects are evaluated at every iteration in order to determine the current state of the life cycle of the environmental object. For punctual environmental objects, this evaluation is applied at its position $\Gamma_{obj} = f(\mathcal{E}_p)$. Curve environmental objects are evaluated along the parametric curve C such that $\Gamma_{obj} = \int_C f(\mathcal{E}_{C(t)})\,dt$. In practice, we compute the evaluation as the average of all control points of the curve $\frac{1}{n}\sum_i^n f(\mathcal{E}_{C_i})$. Region environmental objects are evaluated inside their region Ω as such $\Gamma_{obj} = \int_\Omega f(\mathcal{E}_p)\,dp$. In practice, we compute the average of random points in the region $\frac{1}{n}\sum_i^n f(\mathcal{E}_{p_i})$. When deformations on the environmental object's shape is applied, we apply cage deformation using Green's coordinates in order to keep consistent evaluation points during the whole life cycle of an environmental object.

3.3. Environment values

All environmental objects of an ecosystem has an impact on all the other environmental objects, which would result in an exponentially growing computation effort as the number of environmental objects of the terrain increase. We avoid this problem by considering the environment values as a proxy to allow any environmental object to interact with any other one. Each of the environmental object have a local impact on the environment values without knowledge of neighboring environmental objects. This modification of the environment values can be due to an absorption and deposition of some material $\mathcal{M}$ or an influence on the water currents.

3.3.1. Environment materials

The environment is composed of a scalar field for each of the possible material that can be found in the terrain. The scalar fields represents the availability of the material at any point, but not a height field. Each material is defined with a mass m, a fluid velocity factor ν, a diffusion rate D and finally a decay rate k.

Each environmental object in the terrain is a source and a sink of materials. It is the main mean of communication between environmental objects as it allows them to interact with their surrounding environment. We define the amount of deposed material with $D_{\mathcal{M}}$ and $A_{\mathcal{M}}$ the amount of material deposed and absorbed by the environmental object and $\gamma(t) \in [0,1]$ a factor related with the current state of the environmental object, which state that more material will be displaced when the environmental object is fully formed than when it was just spawn:

$$\int_0^t \gamma(t)\,(D_{\mathcal{M}} - A_{\mathcal{M}})\,dt$$

The deposition and absorption around an environmental object is defined using the Gaussian kernel distance computation from the skeleton.

The scalar field for the material $\mathcal{M}$ is displaced by using a warp operator ω, taking into account the water flow $\mathcal{W}$ and the terrain

slope $\nabla\mathcal{H}$. We unified the warp with m the mass of the material and ν a influence factor of the fluid on the material:

$$\omega(p,t) = m\nabla\mathcal{H}(p,t) + \nu\mathcal{W}(p,t)$$

The materials are also dispersed at a diffusion rate D, for which we can use the advection-diffusion-reaction equation to evaluate the distribution after a time t

$$\frac{\partial\mathcal{M}}{\partial t}\omega\nabla\mathcal{M} = D\nabla^2\mathcal{M} - k\mathcal{M} \tag{2}$$

We solve (**??**) numerically using Euler integration

$$\begin{aligned}\mathcal{M}(p,t+dt) = \mathcal{M}(p,t) + dt(D\nabla^2\mathcal{M}(p,t) - k\mathcal{M}(p,t) \\ - \omega(p,t)\nabla\mathcal{M}(p,t))\end{aligned} \tag{3}$$

The introduction of the decay rate k in the equation allows for the reach of a steady-state, where we can consider the simulation stable. As the user updates the state of the simulation manually, we observe the reach of this steady state before continuing the iterative steps.

3.3.2. Water currents

We define our water currents as a vector field defined as

$$\mathcal{W}(p) = \mathcal{W}_{\text{user}}(p) + \mathcal{W}_{\text{simulation}}(p) + \mathcal{W}_{\text{objects}}(p)$$

With $\mathcal{W}_{\text{user}}$ a user-defined vector field, $\mathcal{W}_{\text{simulation}}$ an analytical solution inspired by a wind flow simulation Paris et al., 2020, and $\mathcal{W}_{\text{objects}}$ the water flow alteration computed from the environmental objects. The component $\mathcal{W}_{\text{simulation}}$ is terrain-induced. Given an input flow direction a, we modify the vector field by warping it with the terrain gradient smoothed at multiple scales :

$$\mathcal{W}_{\text{simulation}}(p) = \sum_{i=0}^{i=n} c_i\omega_i \cdot v$$

with $v = (a(1+k_w h(p))$ and $h(p)$ the depth at point p and k_w a scaling factor, used to simulate the Venturi effects. $\omega_i \cdot v$ is the warping operator at scale i with a coefficient c_i defined as

$$\omega_i \cdot v = (1-\alpha)v + \alpha k_i \nabla\tilde{h_i}^{\perp}(p) \qquad \alpha = \left\|\nabla\tilde{h_i}(p)\right\|$$

with k_i a deviation coefficient, α the slope of the smoothed terrain and $\nabla\tilde{h_i}^{\perp}(p)$ the orthogonal vector of the smoothed terrain. As proposed by the authors, we used two scaling levels $n = 2$ with gaussian kernels of radii 200m and 50m with weights 0.8 and 0.2 and deviation coefficients k_0 and k_1 of 30 and 5.

$\mathcal{W}_{\text{objects}}$ is a deformation field defined as the accumulation of flow primitives Wejchert and Haumann, 1991. Kelvinlets are applied on each environmental objects to deflect the water flow. We use the scale and grab formulations of the regularized Kelvinlets brushes Goes and James, 2017, denoted as $s_\varepsilon(r)$ and $g_\varepsilon(r)$ respectively to simulate obstruction and diversion, are defined as

$$s_\varepsilon(r) = (2b-a)\left(\frac{1}{r_\varepsilon^3} + \frac{1}{2r_\varepsilon^5}\right)(sr)$$

$$g_\varepsilon(r) = \left[\frac{a-b)}{r_\varepsilon}I + \frac{b}{r_\varepsilon^3}rr^t + \frac{a\varepsilon^2}{2r_\varepsilon^3}I\right]f$$

Figure 3: *Starting from a coral colony developed around a canyon (left), the user edits the shape of the canyon, resulting in a different configuration of the scene, killing the corals that ends too deep in the water (center) and the development and growth of new corals at the previous location of the canyon (right).*

with $a = \frac{1}{4\pi\mu}$ and $b = \frac{a}{4(1-\upsilon)}$ provided μ a shear modulus and υ a Poisson ratio provided for each Kelvinlet, $r = p - q$ for p the evaluation position and q the center point of the Kelvinlet, $r_\varepsilon = \sqrt{\|r\|^2 + \varepsilon^2}$ the regularized distance, ε a radial scale for the deformation field, s a scaling factor and f the force vector of the grab operation. Deformations defined on curves use $q = C(p)$ with $C(p)$ the closest point on the curve from the point p and $f = C'(p)$. We can then define $u_o(p) = s_\varepsilon(q-p) + g_\varepsilon(p-q)$.
Finally, we can retrieve the velocity field from the objects:

$$\mathcal{W}_{\text{objects}}(p) = \sum_{o\in\mathcal{O}} \lambda_o u_o(p)$$

3.4. User interaction

The user can guide the generation process. The use of simple shapes as environmental objects facilitate the edition of the simulation, as we can interactively add, remove or modify environmental objects, or focus the generation process in a restricted area. Interaction with the environment values is also provided as geological events, that the user can invoke during the simulation. While the direct interactions on the environmental objects are instantaneous, as a the geological events are active on a given duration.

3.4.1. Direct interactions on the environmental objects

The interactive nature of our simulation enables the user to modify the state of the terrain by manipulating directly the environmental objects of the scene. We assume the modifications applied between two iterations of the simulation.

Translating an environmental object is trivial, we simply requires to evaluate the state of the environmental objects at a translated position. The deformation of environmental objects can be applied on curve and region environmental objects by updating the control points of the skeleton and recomputing the resulting implicit surfaces. The evaluation positions used for region environmental objects are displaced by applying a cage deformation of the 2D shape using the Green coordinates of points in the shape. After the alteration of the region, evaluation points should be keeping a similar distribution than before, avoiding unexpected results during the interaction. By modifying an environmental object, the environment values may change, which can result in the destruction of the now incompatible environment objects in the scene (Figure **??**).

As long as a non-zero fitting function is defined in the terrain,

new environmental objects can be forced by the user at any point of the simulation.

Control over the region of the terrain that should be updated can be given by adjusting all fitting functions through a scalar field $\lambda : \mathbb{R}^2 \to \mathbb{R}$ such that the fitting function $\Gamma_{obj}(p)$ of any new environmental object is evaluated as $\Gamma^*_{obj}(p) = \lambda(p)\Gamma_{obj}(p)$. This is especially useful in the planning of robotic simulations as we can first generate the overall shape of our terrain and secondly focus the generation process around the areas that may be visited by the robot, avoiding useless simulations and computer power. Figure **??** shows an example of colonization of the coral polyps that we limited manually into an annulus.

Our water current simulation is modeled as a simple vector field. As such, the user is able to interact with it at any moment of the simulation, allowing for the death of sensible environmental objects while it will guide the simulation into a new landscape. By modifying the water currents, the user also modifies the transport rate of materials at this position. The modification of currents is given as a stroke, a parametric curve C for which we evaluate $\Delta\mathcal{W}_{\text{user}}(p)$ just as for curved environment objects (Section **??**).

3.4.2. Geological events

A configuration file can define in advance the different events that should be triggered during the simulation. This can be useful to generate landscapes that are close to some existing locations. Multiple geological events can be triggered either as sudden or continuous environmental changes. These changes play a huge role in the morphology of landscapes. We define events with a starting point and an ending point, such that at any time of the simulation we can compute the progress of the event as $t_e \in [0,1]$.

Water level changes are important events that shape the underwater landscapes. As previously submerged environmental objects get elevated above water level, flora and fauna terrain elements dry and die. Deprived from the living part of the elements, everything is more affected by terrestrial erosion. By updating the value of the depth $\mathcal{D}$ evaluated in the fitting functions, any environmental object that is sensible to the depth will be impacted automatically, that may be causing death (Figure **??**). The modification of the water level is defined as

$$\mathcal{D}(p) = \mathcal{D}_0(p) + \sum_{e \in events} \Delta\mathcal{D}_e t_e$$

with $\Delta\mathcal{D}_e$ the amount of water rising or lowering during an event. We assumed a linear evolution of the water level during an event. This allows to evaluate the depth at any point in space and in time.

Subsidence and uplift are the main geological events that create or destroy islands in the long term. These events are simulated as a simple factor on the height field of the generated terrain (Figure **??**). Subsidence is not always uniform in the terrain. As such, the user can provide a position q at which the subsidence is the strongest, the amount of subsidence applied $\Delta\mathcal{H}_e$ and a standard deviation σ for which we can then compute at any point in space and time of the simulation the height of the terrain

$$\mathcal{H}(p) = \mathcal{H}_0(p) \cdot \sum_{e \in events} \frac{G(\|p-q\|)}{G(0)} \Delta\mathcal{H}_e t_e$$

Figure 4: *Lowering the water level by a few meters caused most of the coral objects to satisfy $\Gamma_{obj} \leq 0$, causing their death. Since the water level (blue) decrease slowly, new coral objects spawn progressively at a lower altitude.*

Figure 5: *Simulating subsidence on a part of the terrain (brown area) cause the depth value to change locally, resulting in the death of coral objects that find themselves too deep to survive. Here two subsidence events are triggered in parallel.*

with $G(x)$ the Gaussian function

$$G(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{x^2}{2\sigma^2}\right)$$

Storms are factors of the geomorphology of coral reefs Vila-Concejo and Kench, 2016; Oron et al., 2023 and coasts Domínguez et al., 2005; Cowart et al., 2010. Due to the extreme wind and wave velocities coasts are highly eroded in a short time period and the more fragile corals near the water surface are broken, possibly causing breaches in the reefs and spreading polyps in the currents direction. While there are many factors at play to understand the apparition of storms and the hydrodynamics affecting it, we simplified the model of storms to the user as a single epicenter q with a wind velocity v and a standard deviation σ representing the spread around the epicenter (Figure **??**). The computation of water currents are then computed as

$$\mathcal{W}_{\text{user}}(p) = \mathcal{W}^*_{\text{user}}(p) + \sum_{e \in events | t_e \in [0,1]} v \frac{G(\|p-q\|}{G(0)}$$

In this case, we did not include the linear factor t_e as storms are usually conserving a constant force for the time of the few weeks or months of their occurrence.

with ΔT_e the change of heat during an event, T_0 the temperature at the water surface, and c a very small factor.

The framework can easily be extended as the geological event system stays similar for all events. Including higher level simulations in the event system can be added, such as the simulation of

Figure 6: *The result of a storm localized on one side of the island (red area) modifies the result of the evaluation of environmental objects around its epicenter for a short period of time. Most of the coral objects died from the event, except few environmental objects less sensible to water currents strength.*

tectonic activity, the use of fluid dynamics for tsunami events, the integration of human activity, ...

4. Expert knowledge integration

The definition of the fitting functions of the environmental objects are inspired by the biological and geological factors that rule the evolution of underwater landscapes. The main factors are depth, light, water currents and biodiversity. External events have direct and indirect repercussions on the biodiversity of underwater environments. Coral islands are complex bio systems in which fauna, flora and geology are mixed together.

4.1. Environmental objects description

We have represented with environmental objects some geologic elements, animal elements and flora elements. The low island is most often raised in a circular shape as the process mainly appear around a hot spot under the ground. The evolution of an island into a coral island requires that the environmental conditions are sufficient for coral development: corals will grow slightly below the water surface as waves will break its growth and at a shallow depth (around 3m to 30m deep) in order for light to reach it. As coral grow and die, the skeleton is transformed into porous limestone, providing shelter to surrounding animals and reducing the impact of water erosion on the island. Corals drop polyps that are transported by the water flow and when they stick to a hard surface, as a rock or the reef itself, the coral may grow and colonize the area. As subsidence cause the island to lower, the living part of the coral reef keep growing toward light, which lead to a reef that is constantly close to the water level without reaching it due to wave erosion. The survival of reefs depends on the equilibrium between coral growth and and erosion. Eroded parts of the reef falling in the sheltered part of the reef accumulates, ending up by forming a lagoon. An island formed by a hot spot will inevitably subside in time, until it is completely flatten. As the coral reefs keep growing, only the lagoon remain, resulting in an atoll.
In this work we we integrate the biological and geological knowledge in the fitting functions of the environmental objects we want to generate. We represent the islands as regions that can be appearing with a uniform distribution. From the formulation of the region description (**??**), we mostly create circular islands. The coral elements, environmental objects described as a single point, have a fitting function that take into account the depth of the ground, the amount of sand, fresh water and polyps in their environment, as well as the strength of water currents. Each coral species have different living conditions, but we reduced our work to soft coral which are sensible to water strength and stony corals that are more resistant to erosion. Reefs are formed as coral's skeleton are transformed into calcareous stone, describing then as an environmental object representing multiple others.

4.2. Simplifications

The environmental factors simulated are greatly simplified as the real processes are in a very small time scale, that computer simulation are not able to simulate in interactive time. The use of environmental objects aim to represent a plausible results, while avoiding modeling the smaller scale events. Examples of simplifications are the geometry and material of each environmental object, which have an influence on the water currents through friction, the water currents represented as stationary flows, while the water flow dynamics are a complex system that may change completely at two different times of the day, the animal influence on the reefs that they transform by the ingestion and deposition of sediments, ...

5. Results

Our method provides a way to generate scenes at different scales. We demonstrate this capacity with the generation of a large scene of an island (Figure **??**) after what we focused the generation process in a canyon (Figure **??**), then a small-scale visualization of coral colonies (Figure **??**). In the examples, we rendered the environmental objects as a implicit tree or as individual meshes. The island, lagoons, reefs, canyons and sand ripples as implicit surfaces

A canyon scene can be generated using our method. The water flow is affected by the curve of the canyon such that the currents are oriented in the direction of the curve's tangent.In this example, we force the position of arches to be inside the canyon. The arches deposits a material "rock deposit", which is the main element of the fitting function of the Rock object. The "rock deposit" is slightly affected by water currents, but its mass make it highly affected by gravity. As such, rocks will spawn underneath arches. In reality, an arch is often created as part of a large coral boulder that sees the calcareous bottom part detached by the water currents, often resulting in an arch surrounded by big rocks and smaller rocks from the erosion of the first rocks. As such, we define an environmental object "Arch" with a fitting function $\Gamma_{arch}(p) = 5 - d(canyon - p) * \|\mathcal{W}(p)\|$, an environmental object "Rock" using $\Gamma_{rock}(p) = \mathcal{M}_{rock\_deposit}(p)$ and Pebble using $\Gamma_{pebble}(p) = \mathcal{M}_{smaller\_rock\_deposit}(p)$. Finally, sand ripples are simply described as curves appearing where there is a lot of sand available: $\Gamma_{ripple}(p) = \mathcal{M}_{sand}(p)$. Following these simple rules, Figure **??** shows the emergence of details in the scene.

In this example we defined three different types of corals, coralA, coralB and coralC, to illustrate the possibility to model behaviours

from the choice of fitting functions. Each of the coral types deposits a material "coral polyp" and "coral polyp A" ("coral polyp B" and "coral polyp C" respectively). By considering a fitting function that minimize the ratio $\frac{\text{coral polyp}}{\text{coral polyp A}}$, we can see an emergent behavior of the three types of coral fighting for the space colonization. Figure **??** shows the result of this simulation at three different interations. At the border between two colonies, none of the colonies make progression due to the amount of coral polyp specific from the other colony.

6. Discussion

The proposed method aims to generate plausible landscapes using simplified versions of the evolution of an ecosystem and of the 3D representation. The biological realism of the result is highly correlated to the amount of simplification and assumptions, while the visual realism is completely dependent to the geometric functions used for the 3D modeling of the environmental objects. While proposing a flexible method that propose a generic approach for terrain generation, a close collaboration with fields experts and with graphists is needed to achieve optimal results.

Most simulation algorithm's quality depends on the size of the time step used, but with the introduction of a decay rate in the materials properties, we limit the influence of time steps by considering that steady-state are reachable. The material deposition and absorption on punctual environmental objects can be seen as a Dirac function δ centered at their position resulting in the advantage that material displacement function can use the definition of the diffusion equation instead of the advection-diffusion-reaction equation. This equation allowing us to evaluate the state of the material $\mathcal{M}$ without intermediate steps, but this is not applicable with curve- and region-based environmental objects.

7. Conclusion

We have proposed a method to generate terrains procedurally using sparse representations. This representation, the environmental objects, enables to introduce expert knowledge by the mean of the fitting functions that rule the environmental objects life cycle, but also to integrate the user in the loop during the generation process. We reduced the terrain resolution limitations by defining the environment objects as parametric elements. Thanks to the sparse representation based on single points, curves and regions, we allow for direct manipulation of the environmental objects of the scene by the user which, thanks to the environment steady state consideration, also enables to include these interactions in the automatic simulation process.
Integrating environmental properties in the fitting function of environmental objects allows the user to guide the generation through geological events. Our method enables each environmental object of the scene to influence the environment locally, reducing the need of computations while also retrieving environment values locally, which result in a parallelizable life-like simulation process. The genericity of the environment properties definitions should be sufficient for plausible generation of other landscape types as long as expert knowledge can be translated to environmental object's formalism.

We limited our work to the use of 2D scalar fields as they are more easily differentiable, interpretable and lighter than volumetric representations. However, future works include using 3D representations of the terrain and the environment to generate 3D terrains, including cavities, sub-terrestrial areas and the interior of coral structures.

References

Abela, R., Liapis, A., & Yannakakis, G. N. (2015). A constructive approach for the generation of underwater environments. *Proceedings of the FDG workshop on Procedural Content Generation in Games*.

Bosscher, H., & Schlager, W. (1992). Computer simulation of reef growth. *Sedimentology*, *39*, 503–512. https://doi.org/10.1111/j.1365-3091.1992.tb02130.x

Brosz, J., Ranchordas, A., Araújo, H., & Jorge, J. (2007). Terrain synthesis by example. *Communications in Computer and Information Science*, *4 CCIS*. https://doi.org/10.1007/978-3-540-75274-5

Cordonnier, G., Galin, É., Gain, J., Beneš, B., Guérin, É., Peytavie, A., & Cani, M.-P. (2017). Authoring landscapes by combining ecosystem and terrain erosion simulation. *ACM Transactions on Graphics*, *36*. https://doi.org/10.1145/3072959.3073667

Cordonnier, G., Jouvet, G., Peytavie, A., Braun, J., Cani, M.-P., Benes, B., Galin, E., Guérin, E., & Gain, J. (2023). Forming terrains by glacial erosion. *ACM Transactions on Graphics*, *42*, 14. https://doi.org/10.1145/3592422Ãf

Cortial, Y., Peytavie, A., Galin, É., & Guérin, É. (2019). Procedural tectonic planets. *Eurographics Computer Graphics Forum*, *38*, 1–11. https://doi.org/10.1111/cgf.13614

Cowart, L., Walsh, J. P., & Corbett, D. R. (2010). Analyzing estuarine shoreline change: A case study of cedar island, north carolina. *Journal of Coastal Research*, *265*, 817–830. https://doi.org/10.2112/jcoastres-d-09-00117.1

de Carpentier, G. J. P., & Bidarra, R. (2009). Interactive gpu-based procedural heightfield brushes. *Proceedings of the 4th International Conference on Foundations of Digital Games*, *8*, 55–62. https://doi.org/10.1145/1536513.1536532

Domínguez, L., Anfuso, G., & Gracia, F. J. (2005). Vulnerability assessment of a retreating coast in sw spain. *Environmental Geology*, *47*, 1037–1044. https://doi.org/10.1007/s00254-005-1235-0

Dunbabin, M., Manley, J., & Harrison, P. L. (2020). Uncrewed maritime systems for coral reef conservation. *2020 Global Oceans 2020: Singapore - U.S. Gulf Coast*. https://doi.org/10.1109/IEEECONF38699.2020.9389173

Gain, J., Marais, P., & Straßer, W. (2009). Terrain sketching. *Proceedings of I3D 2009: The 2009 ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, *1*, 31–38. https://doi.org/10.1145/1507149.1507155

Galin, E., Guérin, E., Peytavie, A., Cordonnier, G., Cani, M.-P., Benes, B., & Gain, J. (2019). A review of digital terrain modeling. *Computer Graphics Forum*, *38*, 553–577. https://doi.org/10.1111/cgf.13657

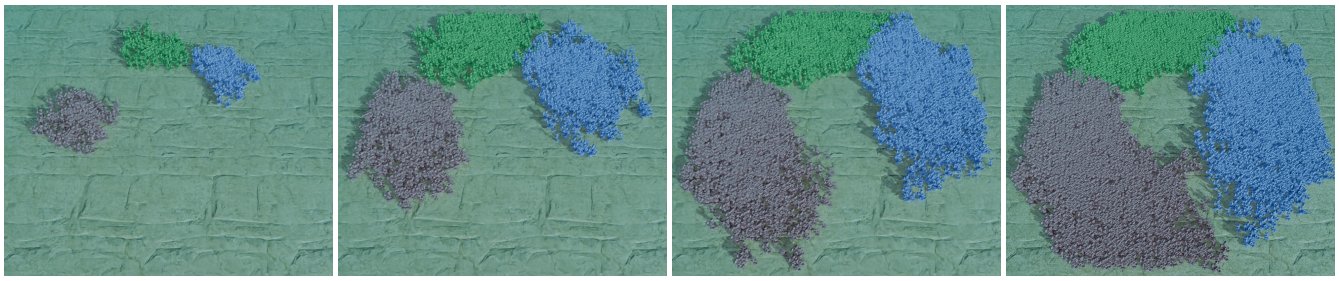

Figure 7: *Three colonies of coral (red, blue, green) restricted to an annulus the middle section of the terrain fighting for the space.*

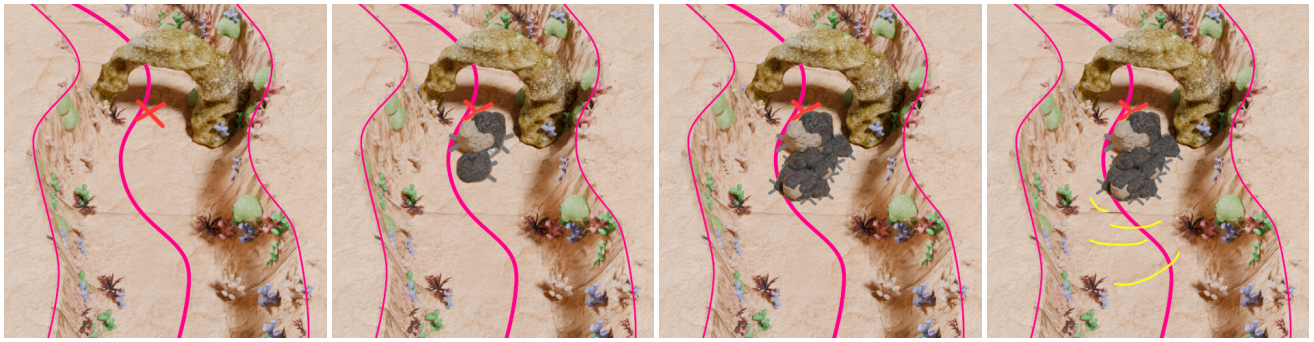

Figure 8: *Evolution of a canyon scene at different iterations of the simulation. The apparition of an arch causes the spawning of rocks, pebbles, and finally some deposition of sand at the bottom of the canyon, spawning ripples.*

Figure 9: *A simple coral island is generated using an island, a lagoon, reefs coral polyps, beaches, trees and algae environmental objects. Trees appear on beaches and algae grow in the lagoon's sand.*

Goes, F. D., & James, D. L. (2017). Regularized kelvinlets: Sculpting brushes based on fundamental solutions of elasticity. *ACM Transactions on Graphics*, *36*, 401–411. https://doi.org/10.1145/3072959.3073595

Grosbellet, F., Peytavie, A., Guérin, É., Galin, É., Mérillou, S., & Benes, B. (2016). Environmental objects for authoring procedural scenes. *Computer Graphics Forum*, *35*, 296–308. https://doi.org/10.1111/cgf.12726

Guérin, E., Peytavie, A., Masnou, S., Digne, J., Sauvage, B., Gain, J., & Galin, E. (2022). Gradient terrain authoring. *Computer Graphics Forum*, *41*, 85–95. https://doi.org/10.1111/cgf.14460

Guérin, É., Digne, J., Galin, É., & Peytavie, A. (2016). Sparse representation of terrains for procedural modeling. *Computer Graphics Forum*, *35*, 177–187. https://doi.org/10.1111/cgf.12821

Guérin, É., Digne, J., Galin, É., Peytavie, A., Wolf, C., Beneš, B., & Martinez, B. (2017). Interactive example-based terrain authoring with conditional generative adversarial networks. *ACM Transactions on Graphics*, *36*. https://doi.org/10.1145/3130800.3130804

Kapp, K., Gain, J., Guérin, E., Galin, E., & Peytavie, A. (2020). Data-driven authoring of large-scale ecosystems. *ACM Transactions on Graphics*, *39*. https://doi.org/10.1145/3414685.3417848

Li, W. (2021). Procedural modeling of the great barrier reef. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *13017 LNCS*, 381–391. https://doi.org/10.1007/978-3-030-90439-5_30

Mareschal, J. C. (1989). Fractal reconstruction of sea-floor topography. *Pure and Applied Geophysics PAGEOPH*, *131*, 197–210. https://doi.org/10.1007/BF00874487

Maslin, M., Louis, S., Dejean, K. G., Lapierre, L., Villéger, S., & Claverie, T. (2021). Underwater robots provide similar fish biodiversity assessments as divers on coral reefs. *Remote Sensing in Ecology and Conservation*, *7*, 567–578. https://doi.org/10.1002/rse2.209

Mei, X., Decaudin, P., & Hu, B. G. (2007). Fast hydraulic erosion simulation and visualization on gpu. *Proceedings - Pacific Conference on Computer Graphics and Applications*, 47–56. https://doi.org/10.1109/PG.2007.27

Michel, E., Emilien, A., & Cani, M.-P. (2015). Generation of folded terrains from simple vector maps. *Eurographics 2015 short paper proceedings*, 4–8. https://doi.org/10.2312/egsh.20151019

Musgrave, F. K., Kolb, C. E., & Mace, R. S. (1989). The synthesis and rendering of eroded fractal terrains. *Proceedings of the 16th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1989*, 41–50. https://doi.org/10.1145/74333.74337

Oron, S., Akkaynak, D., Tchernov, B. N. G., & Shaked, Y. (2023). How monster storms shape fringing reefs: Observations from the 2020 middle east cyclone. *Ecosphere*, *14*. https://doi.org/10.1002/ecs2.4602

Palmer, M. R., Shagude, Y. W., Roberts, M. J., Popova, E., Wihsgott, J. U., Aswani, S., Coupland, J., Howe, J. A., Bett, B. J., Osuka, K. E., Abernethy, C., Alexiou, S., Painter, S. C., Kamau, J. N., Nyandwi, N., & Sekadende, B. (2021). Marine robots for coastal ocean research in the western indian ocean. *Ocean and Coastal Management*, *212*. https://doi.org/10.1016/j.ocecoaman.2021.105805

Paris, A., Peytavie, A., Guérin, E., Dischler, J.-M., & Galin, E. (2020). Modeling rocky scenery using implicit blocks. *The Visual Computer*, *36*, 2251–2261. https://doi.org/10.1007/s00371-020-01905

Paris, A., Peytavie, A., Guérin, É., Argudo, O., & Galin, É. (2019). Desertscape simulation. *Computer Graphics Forum*, *38*, 47–55. https://doi.org/10.1111/cgf.13815

Patel, D., Natali, M., Lidal, E. M., Parulek, J., Brazil, E. V., & Viola, I. (2021). Modeling terrains and subsurface geology. *Interactive Data Processing and 3D Visualization of the Solid Earth*, 1–43. https://doi.org/10.1007/978-3-030-90716-7_1

Prusinkiewicz, P., & Hammel, M. (1993). Fractal model of mountains with rivers. *Proceedings - Graphics Interface*, 174–180.

Schott, H., Paris, A., Fournier, L., Guérin, E., & Galin, E. (2023). Large-scale terrain authoring through interactive erosion simulation. *ACM Transactions on Graphics*, *42*, 1–15. https://doi.org/10.1145/3592787

Smelik, R. M., Tutenel, T., Bidarra, R., & Benes, B. (2014). A survey on procedural modelling for virtual worlds. *Computer Graphics Forum*, *33*, 31–50. https://doi.org/10.1111/cgf.12276

Talgorn, F. X., & Belhadj, F. (2018). Real-time sketch-based terrain generation. *ACM International Conference Proceeding Series*, 13–18. https://doi.org/10.1145/3208159.3208184

Vila-Concejo, A., & Kench, P. (2016, August). Storms in coral reefs. In *Coastal storms: Processes and impacts* (pp. 127–149). Wiley Blackwell. https://doi.org/10.1002/9781118937099.ch7

Wejchert, J., & Haumann, D. (1991). Animation aerodynamics. *Proceedings of the 18th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1991*, *25*, 19–22. https://doi.org/10.1145/122718.122719

Williams, S. B., Pizarro, O., Steinberg, D. M., Friedman, A., & Bryson, M. (2016). Reflections on a decade of autonomous underwater vehicles operations for marine survey at the australian centre for field robotics. *Annual Reviews in Control*, *42*, 158–165. https://doi.org/10.1016/j.arcontrol.2016.09.010