

# Procedural and learning-based generation of coral reef islands

Submission1265



**Figure 1:** Given a free hand-drawn sketch of the different regions of an island, our method generate a corresponding height field using neural networks. Subsidence (gradual sinking of land) is controlled by the user in the luminosity channel of the input.

## Abstract

We propose a procedural method for generating single volcanic islands with coral reefs using user sketching from two projections: a top view, which defines the island's shape, and a profile view, which outlines its elevation. These projections, commonly used in geological and remote sensing domains, are complemented by a user-defined wind field, applied as a distortion field to deform the island's shape, mimicking the effects of wind and waves on the long term and enabling finer user control. We then model the growth of coral on the island and its surroundings to construct the reef following biological observations. Based on these inputs, our method generates a height field of the island. Our method creates a large variety of island models to compose a dataset used for training a conditional Generative Adversarial Network (cGAN). By applying data augmentation, the cGAN allows for even greater variety in the generated islands, providing users with higher freedom and intuitive controls over the shape and structure of the final output.

1    **Keywords:** Procedural modeling, Terrain generation, cGAN,  
2    Coral reef, Sketch-based interface

## 3    1. Introduction

4    Simulating the formation of coral reef islands presents significant  
5    challenges due to the complex interplay of geological, environmen-  
6    tal, and biological factors [Hop14]. One major difficulty lies in cap-  
7    turing the long-term subsidence of volcanic islands, which occurs  
8    over millions of years, while concurrently simulating the upward  
9    growth of coral reefs that rely on environmental conditions such as  
10   water depth, temperature, and sunlight. This combination of slow  
11   geological processes and dynamic biological growth is difficult to  
12   replicate accurately in a computational model.

13   The biological aspects of coral growth are inherently tied to en-  
14   vironmental factors. Coral reefs grow only within a specific range  
15   of water depth and sunlight, and their growth patterns are affected  
16   by the health of the reef ecosystem and the availability of resources.  
17   Accurately modeling these biological dependencies in a procedural  
18   system is complex, since these factors are numerous and difficult

19   to generalize. Additionally, the scarcity of data available obstructs  
20   the global understanding of these ecosystems. In a recent high-  
21   resolution mapping of shallow coral reefs [LMK\*24], researchers  
22   estimated the total surface area of this biome to cover less than  
23   0.7% of Earth's area, and more specifically, coral habitats consti-  
24   tute less than 0.2%.

25   Moreover, the need to design plausible reef structures in digi-  
26   tal environments for applications such as scientific visualization,  
27   simulation, or digital entertainment, calls for generative algorithms  
28   that balance realism with controllability. Existing terrain generation  
29   methods, such as Perlin noise-based algorithms or uplift-erosion  
30   models, are often ill-suited for these processes. While they can gen-  
31   erate natural-looking landscapes (such as alpine landscapes, repre-  
32   senting about a quarter of land area [KSBZ14]), they do not ac-  
33   count for the unique geological and biological interactions that  
34   govern coral reef island formation, thus missing coherence. Cap-  
35   turing these dynamics, while also providing user control during the  
36   modeling of a terrain, requires a balance between plausibility and  
37   flexibility, allowing for both accurate, though computationally ex-

38 pensive, simulation of natural processes and intuitive user control  
 39 in interactive time.

40 Despite advances in terrain generation, existing methods struggle  
 41 with user-controlled design of specific island shapes and achieving  
 42 realism without real data. Coral reef islands exemplify this gap:  
 43 we lack datasets to directly train deep learning models, and purely  
 44 procedural methods require expert tuning to mimic their features.

45 To address these issues, we propose to use curve-based modeling  
 46 techniques as an initial step in our approach as a means to  
 47 efficiently create a large and diverse set of training examples for  
 48 a learning-based model. Each synthetic example is represented by  
 49 a terrain height field and a corresponding semantic label map that  
 50 marks different regions, providing structured input-output pairs for  
 51 the learning stage.

52 We trained a conditional Generative Adversarial Network  
 53 (cGAN) as the core of our learning-based approach [MO14;  
 54 IZZE17]. A cGAN is a type of deep learning model that learns  
 55 to generate realistic data based on an input condition or context.  
 56 In our case, the cGAN takes as input the semantic label map of  
 57 an island generated by the procedural step and learns to produce a  
 58 plausible island height field that matches this layout. By training  
 59 on a large variety of examples from our curve-based algorithm, the  
 60 cGAN captures the subtle terrain features and variations characteristic  
 61 of coral reef islands. This approach surpasses the capabilities  
 62 of traditional procedural rules, aided by extensive data augmentation.  
 63 The cGAN can be used on its own to generate new island terrains  
 64 with simplified and more intuitive user inputs through digital drawing,  
 65 and the model will generate a realistic island terrain accordingly.

66 The key contributions are as follows: 1) a novel curve-based pro-  
 67 cedural algorithm for shaping island terrains from top and profile  
 68 views, 2) the training of a deep learning model on synthetic data  
 69 derived from procedural rules, serving as an abstraction layer that  
 70 hides underlying complexity, 3) a demonstration that the cGAN ap-  
 71 proach tolerates imprecise, low-detail user input sketches, broadening  
 72 usability, without the need for cutting-edge network architectures,  
 73 and 4) an insight that procedural generation remains essential  
 74 to produce training data in data-sparse domains such as coral reef  
 75 islands. These contributions collectively show a pathway to blend  
 76 user-driven design with learning-based generation in terrain mod-  
 77 eling, especially for data-sparse domains.

## 79 2. Related work

80 Procedural terrain generation spans a spectrum from purely noise-  
 81 driven algorithms to physics-based simulations and, more recently,  
 82 data-driven methods. In the context of coral reef islands, where  
 83 both long-term geological subsidence and biogenic reef accretion  
 84 interplay, existing approaches fall short in one of two ways: they  
 85 either lack ecological or geological grounding, or they offer insuf-  
 86 ficient authoring control.

87 **Procedural and simulation-based methods** Noise-based tech-  
 88 niques such as Perlin noise [Per85], Simplex noise [Per01], and  
 89 the Diamond-Square algorithm [FFC82] (often extended via fBm  
 90 or multifractal noise [MKM89; EPW\*03]) remain popular for their

91 simplicity and speed. Island shapes are generated by modulating  
 92 noise with radial falloff masks [Ols04], but these methods can-  
 93 not reproduce reef rings, lagoons, or atoll structures in a geo-  
 94 logically coherent manner. They treat terrain purely as a signal-  
 95 processing problem, separated from processes like volcanic subsi-  
 96 dence or coral growth [SKG\*09; GGP\*19].

97 Simulation-based approaches introduce surface deformations  
 98 by modeling erosion [BTHB06; NWD05; MDH07], tectonic up-  
 99 lift [CBC\*16; CCB\*17; SPF\*23], or vegetation-terrain feedback  
 100 ([IECC\*21; CGG\*17]). Hydraulic and thermal erosion capture flu-  
 101 vial networks and slope-driven mass wasting, but they omit under-  
 102 water sedimentation and biogenic carbonate accretion. Tectonic  
 103 and isostatic models excel at orogeny but ignore coral reef dynam-  
 104 ics, while vegetation-based methods do not generalize to marine  
 105 ecosystems. Consequently, none of these simulations jointly cap-  
 106 ture the slow subsidence of volcanic islands and the compensatory  
 107 growth of surrounding reefs on the timescales required for atoll for-  
 108 mation.

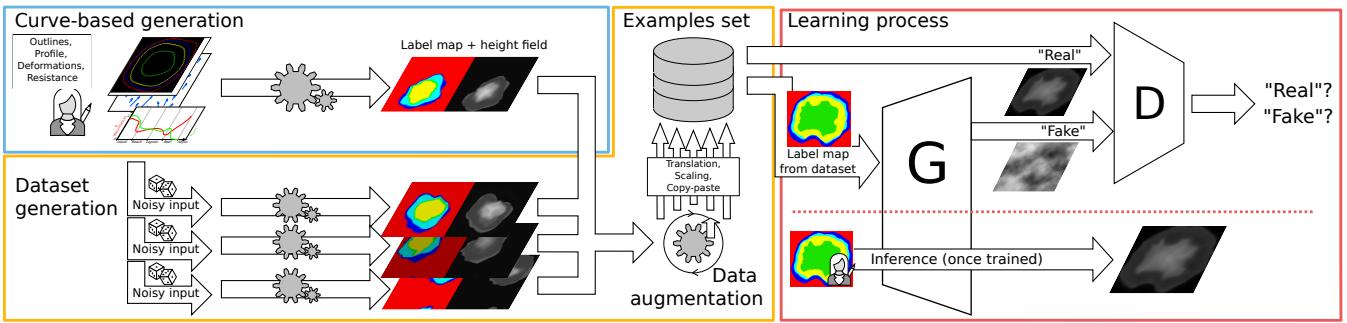
109 **Sketch-based terrain modeling** Sketch-driven interfaces bridge  
 110 user intent and procedural detail. Curve-based systems let users  
 111 draw ridges, valleys, or coastlines that guide surface deformation  
 112 and noise propagation [GMS09; HGA\*10]. Constraint-based meth-  
 113 ods extend this by enforcing absolute elevation or slope values at  
 114 control curves or points, solved via diffusion or fractal interpola-  
 115 tion [GCRR20; TB18], and even gradient-domain editing for slope  
 116 control [GPM\*22]. Semantic approaches encode high-level "terrain  
 117 atoms" from a dictionary of primitives [GGP\*15] or interpret ge-  
 118 ological schematics into 3D models [NVP12]. While these tech-  
 119 niques grant artists fine-grained control, they typically lack ecolog-  
 120 ical constraints and have not been tailored to marine biogeomor-  
 121 phology.

122 **Learning-based terrain synthesis** Deep generative models offer  
 123 a way to learn complex patterns without explicit procedural rules.  
 124 Unconditional GANs have been applied to digital elevation maps of  
 125 mountains [WRMB18] and joint height-texture synthesis [SW19],  
 126 but their reliance on latent noise prevents precise layout control.  
 127 Two-stage pipelines use an initial GAN for heightmaps and a con-  
 128 ditional GAN for textures [BP17] or reverse the order (imagery-to-  
 129 DEM) [PC20], yet still lack the ability to guide authoring.

130 Conditional GANs (cGANs) extend image-to-image translation  
 131 methods such as pix2pix [IZZE17] to terrain, enabling sketch- or  
 132 label map-conditioned generation. Prior work includes sketch-to-  
 133 DEM translation for generic landforms [GDG\*17] and sparse "alti-  
 134 tude dot" conditioning [VMP21], as well as cGANs that invert  
 135 satellite imagery into elevation [Sis22]. However, these models re-  
 136 quire extensive paired real-world datasets which are scarce for coral  
 137 reef islands.

## 138 3. Overview

139 Our method for procedural generation of coral reef islands is com-  
 140 posed of two independent modeling techniques shown in Figure 2.  
 141 A first curve-based algorithm (top-left blue block, presented in Sec-  
 142 tion 4), parametrizes the surface of islands by a top-view sketch  
 143 interface, describing the outlines of its constituent regions and a  
 144 stroke-based wind field deforming them, as well as a profile-view



**Figure 2:** Our pipeline first prompts the user to create single pairs of coral reef island height fields and label maps using our proposed curve-based modeling algorithm (Section 4). The same algorithm is applied multiple times on randomly altered versions of the user input to create a dataset, which we further enhance with data augmentation techniques (Section 5). Finally, we train a conditional Generative Adversarial Network, which can then be used standalone to create height fields from labelled sketches (Section 6).

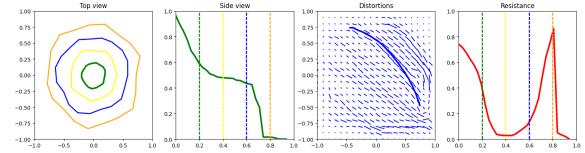
sketch describing for each region its altitude and resistance to deformations. User inputs are given as 1D functions (altitude and resistance), a 1D polar function (island outlines) and a 2D vector field (wind field) as shown in Figure 3, which are convenient to randomize through the use of noise, but limit the user's freedom to model arbitrary shapes.

We propose a second learning-based generation algorithm (right red blocks Figure 2, developed in Section 6), which trains a neural generator  $G$  to transform a labelled image into a height field. Parallelly, a discriminator  $D$  is trained to distinguish height fields provided from the training set against height fields generated by  $G$ . This adversarial training strengthens the outputs of the generator, up to the point where we discard the discriminator and training data, only keeping the generation step (bottom-right). Users are then able to provide unseen label maps and obtain height fields as close as possible to the training dataset. This method for terrain modeling is poorly constrained, however requires a large amount of prior data in its training set, which is not available in our case.

We connect these two algorithms through a process of dataset generation (central orange block Figure 2, described in Section 5) in order to enforce the benefits of each while reducing their limitations. Given the initial curve-based user inputs, we create a dataset composed of similar samples processed by our first algorithm, rasterized into a 2D image of the parametrized regions, and paired with the resulting height field. We then greatly increase the size of the dataset by employing various data augmentation techniques such as translations, scaling and copy-pasting of multiple islands in a single sample. We then obtain a dataset large enough for training our cGAN and enable users to procedurally create coral reef islands with a high level of control.

#### 4. Curve-based island generation

The generation of coral reef island terrains involves a structured process that takes the user's sketches and produces a complete 3D terrain model. This process begins with the creation of the initial height field based on the user's input, followed by the application of wind deformation to introduce natural variations, and concludes



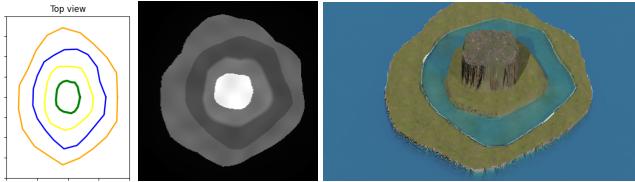
**Figure 3:** The user can interact directly on the island by editing the different canvases in any order. This UI shows, from left to right, the top-view sketch with the different outlines of each regions, the profile-view sketch with the outlines represented by dotted lines, the wind velocity sketch drawn with strokes (last stroke is visible), and the resistance function showing here a high resistance at the top of the island and on the front reef.

with the integration of coral reef features through subsidence and coral growth modeling.

The generation of coral reef islands in our system begins with two intuitive curve-based inputs from the user: a top-view sketch and a profile-view sketch, which define the island's horizontal layout and vertical elevation profile. In addition to these sketches, the user can further refine the terrain by applying wind deformation strokes, which simulate the effects of wind and waves on the island's shape. This combination of sketches and wind inputs gives users precise control over both the island's structure and its natural variations, such as irregular coastlines or concave features. We present the usefulness of these sketches in this section, and describe the technical details in the next section.

##### 4.1. Initial height field generation

The top-view sketch defines the island's outline as seen from above. Using a simple drawing interface, the user delineates concentric boundaries for key regions such as the island itself, the beaches, the lagoon, and the surrounding abyss, around the center of the canvas. Each boundary is represented in polar coordinates, where  $r_p$  is the radial distance from the island's center and  $\theta_p$  is the angular position. Allowing  $r$  to vary with  $\theta$  introduces irregular, natural shapes rather than perfect circles (Figure 4).



**Figure 4:** Using only the outlines from the top-view sketch, each point in the field is assigned a region (island, beach, lagoon, reef, abyss), which later guides its height assignment.

241 canvas. Each stroke is represented as a parametric curve  $C$ , interpreted  
242 as a local wind flow with direction given by the derivative  
243  $C'$ , strength  $S$ , and influence width  $\sigma$ . These strokes simulate wind  
244 and wave erosion effects on the terrain.

245 The deformation vector at any terrain point  $\mathbf{p}$  is computed as a  
246 sum over all strokes:

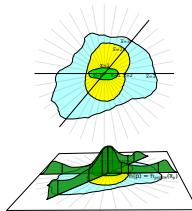
$$\Phi(\mathbf{p}) = \sum_{C \in \text{curves}} S \frac{C'(\mathbf{q})}{\|C'(\mathbf{q})\|} \cdot G_\sigma (\|\mathbf{p} - \mathbf{p}_C^*\|) \quad (2)$$

247 weighted by a Gaussian falloff centered on the closest point  $\mathbf{p}_C^*$   
248 along each curve (Figure 7, top):

$$G_\sigma(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}}. \quad (3)$$

249 To preserve semantic structure across terrain regions, a resist-  
250 tance function  $\rho(\tilde{x})$  modulates the deformation based on terrain  
251 zones such as beach, lagoon, reef, or abyss (Figure 8). The final  
252 deformation vector becomes:

$$\tilde{\Phi}(\mathbf{p}) = (1 - \rho(\tilde{x}_{\mathbf{p}})) \cdot \Phi(\mathbf{p}). \quad (4)$$

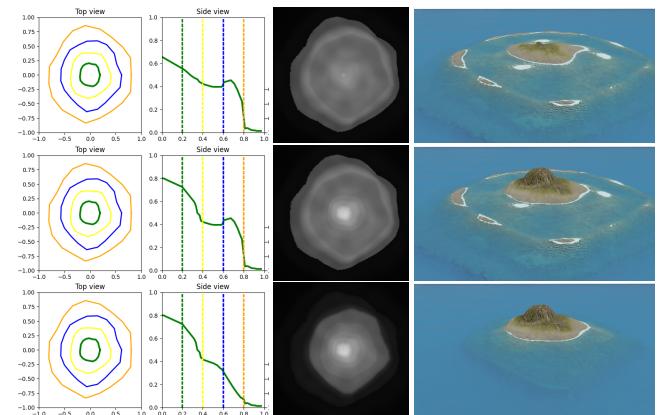


**Figure 5:** Parametric distance  $\tilde{x}$  depending on angle  $\theta$ .

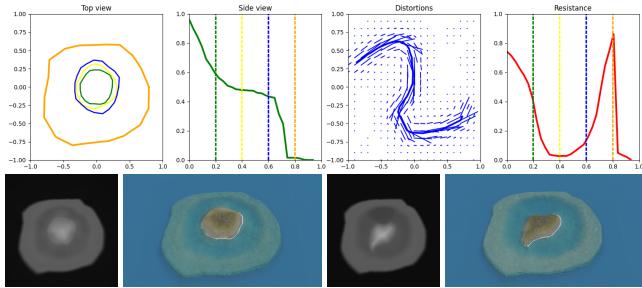
253 This warp is applied to both the height field and the label map,  
254 ensuring consistent semantic deformation. For example, applying  
255 strokes to one side of a circular island creates concave coastlines  
256 while leaving high-resistance regions (e.g., the abyss) unaffected,  
257 simulating the localized impact of natural erosion (Figure 7, bot-  
258 tom).

## 4.2. Coral reef modeling

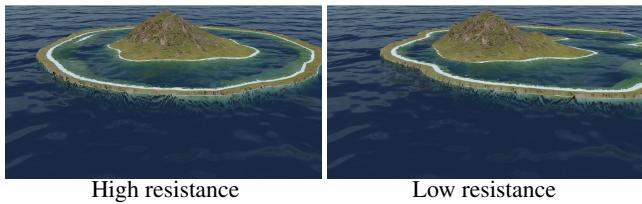
Once the terrain has been generated and deformed by the wind, we  
simulate the long-term geological evolution of coral reef islands



**Figure 6:** Providing a smooth function between each region results in islands with plausible reliefs. We fixed the outlines while editing only the height function in order to produce, from top to bottom, a low island, a coral reef island, and finally an identical island without the reef.



**Figure 7:** A top-view wind vector field is defined from user-provided strokes (top, blue), in association with a resistance function (top, red); a height field is deformed accordingly. Left: original height field and render; right: altered results. The beach and lagoon regions are defined with low resistance, which is visible by having only these regions deformed in bottom results.



**Figure 8:** (Left) An island defined with a high resistance, under a uniform lateral wind velocity field, and (right) the same island with lower resistance on the reef borders, simulating the effect of coastal erosion.

through two parallel processes: the subsidence of the volcanic island and the upward growth of coral reefs. As observed in nature, the volcanic base sinks over time while coral formations grow vertically to remain close to the water surface, following the "keep-up" strategy of reef development.

#### 4.2.1. Subsidence

Subsidence is modeled by uniformly scaling the original terrain height downward, simulating the gradual sinking of the volcanic landmass due to tectonic processes. The user specifies a subsidence rate  $\lambda \in [0, 1]$ , which controls how much the island has sunk. The subsided terrain is computed as:

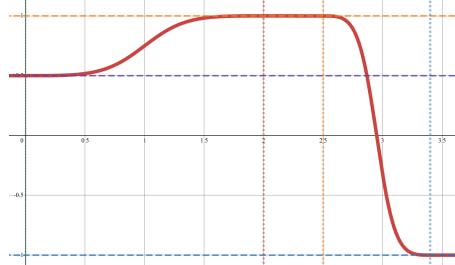
$$h_{\text{subsid}}(\mathbf{p}) = (1 - \lambda) \cdot h_0(\mathbf{p}). \quad (5)$$

This factor is applied uniformly across the island, offering a geologically plausible and computationally efficient approximation of large-scale subsidence.

#### 4.2.2. Coral reef growth

Coral reef growth is modeled independently from the subsiding terrain. The system generates a coral-specific height field  $h_{\text{coral}}(\mathbf{p})$

that remains near the sea surface regardless of the island's vertical shift, reflecting coral growth in biologically viable depth ranges (typically 0-30 meters below sea level).



**Figure 9:** Our model describes reef growth by a piecewise function  $h_{\text{coral}}$  which is flat in the lagoon, the crest and abyss, and follows a smoothstep function as transitions for the backreef and fore reef regions. Zones' anchor heights are represented by horizontal dashed lines; zones' limits are dotted vertical lines.

We define distinct reef zones anchored at specific depths:

- Reef crest near sea level:  $h_{\text{crest}} = -2 \text{ m}$ ,
- Back reef and lagoon:  $h_{\text{back}} = -20 \text{ m}$ ,
- Fore reef sloping to abyss:  $h_{\text{abyss}} = -100 \text{ m}$ .

Each reef subregion is defined over a parametric domain  $x \in [0, 1]$ , with  $x = 0$  the beginning of the reef region and  $x = 1$  its end, directly inputting the parametric distance  $x = \tilde{x} - i_{\text{reef}}$ . For instance:

- Back reef:  $x_{\text{back},\text{start}} = 0$ ,  $x_{\text{back},\text{end}} = 0.5$ ,
- Reef crest:  $x_{\text{crest},\text{start}} = 0.75$ ,  $x_{\text{crest},\text{end}} = 0.8$ ,
- Abyss begins at  $x_{\text{abyss},\text{start}} = 1$ .

We model transition zones between these regions using a smoothstep operator:

$$\text{smoothstep}(x) = 3x^2 - 2x^3. \quad (6)$$

We denote the interpolating function as:

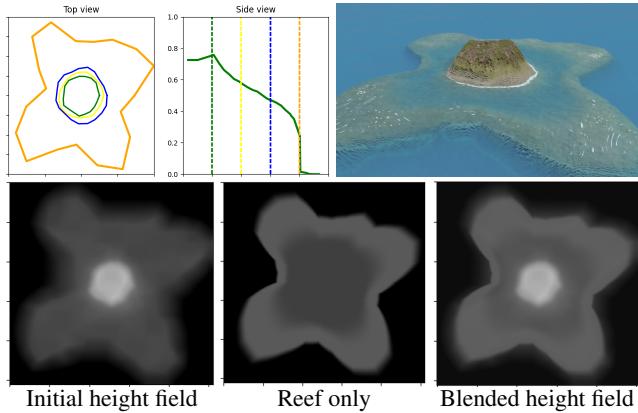
$$S(a, b, x_0, x_1, x) = a + (b - a) \text{smoothstep}\left(\frac{x - x_0}{x_1 - x_0}\right). \quad (7)$$

The complete coral height field, as displayed in Figure 9, is built as a piecewise function:

$$h_{\text{coral}}(x) = \sum_{r \in \text{subregions}} \begin{cases} h_r & \text{if } x_{r,\text{start}} \leq x \leq x_{r,\text{end}} \\ 0 & \text{otherwise} \end{cases} + \sum_{t \in \text{transitions}} \begin{cases} S(h_t, h_{t+1}, x_{t,\text{end}}, x_{t+1,\text{start}}, x) & \text{if } x_{t,\text{end}} < x < x_{t+1,\text{start}} \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

#### 4.2.3. Output

Finally, we merge the island base height field and the coral reef height field by our *ad-hoc* smooth maximum operator  $\text{smax}$  defined



**Figure 10:** Volcano with single vent. Bottom left: the initial height field is computed directly from the user input. Bottom center: the reef height field is output from Equation (8). Bottom right: we blend the two results with Equation (9).

318 noise  $\eta$  such that the final contour, profile, and resistances, are de-  
319 fined as

$$\begin{aligned} r(\theta) &= r^* + \eta_{\text{contours}}(\theta) \\ h_{\text{profile}}(\tilde{x}) &= h_{\text{profile}}^*(\tilde{x}) \cdot \eta_{\text{profile}}(\tilde{x}) \\ \rho(\tilde{x}) &= \rho^*(\tilde{x}) \cdot \eta_{\text{resistance}}(\tilde{x}) \end{aligned}$$

320 Finally, a wind velocity field is randomly generated as a final in-  
321 put for the curve-based modeling algorithm, introducing deforma-  
322 tions on the resulting islands. The realistic nature of wind is ignored  
323 for the generation of the wind strokes in order to provide complex-  
324 ity and variety in the results. We generate a random number  $n$  of  
325 strokes and their path by uniformly sampling a random number  $m$   
326 of points. The spread and intensity of each stroke is also random.

327 Once all inputs are set, we generate an example for multiple lev-  
328 els of subsidence  $\lambda \in [0, 1]$  to obtain a height field incorporating the  
329 coral reef modeling and the associated label map.

330 We use the Hue component to encode the labels directly from  
331 the parametric distance  $H = |\tilde{x}|$  and encode the subsidence rate  
332 into the Value component  $V = \lambda$ . Moreover, we purposefully left  
333 the Saturation component unchanged at this stage, reserving space  
334 for potentially including another parameter in the future.

#### Data augmentation

335 To enhance the variety of the dataset and improve the model's  
336 ability to generalize, we apply several data augmentation tech-  
337 niques in addition to the usual affine transformations (rotation, scal-  
338 ing, and flipping):

340 *Translation:* Since the original algorithm always centers the is-  
341 land, we translate the islands within the image to remove this con-  
342 straint (Figure 11, leftmost). This ensures that the cGAN can gen-  
343 erate islands in any position within the frame. By wrapping the island  
344 on the borders, the model learns that data can appear on the edges  
345 on the image.

346 *Copy-paste:* In some cases, we combine multiple islands into a  
347 single sample, with only a maximum intersection over union (IoU)  
348 of 10%, allowing the abysses to overlap but without obtaining other  
349 region types to merge. Height fields blending is done through the  
350 smooth maximum function from Equation (9), and the label blend-  
351 ing uses the usual max operator. The regions not covered by any  
352 island are assigned the abyss ID, which is very close to the min-  
353 imal height, meaning that in this case specifically, the subsidence  
354 factor (Value channel) is close to irrelevant (Figure 11, rightmost).

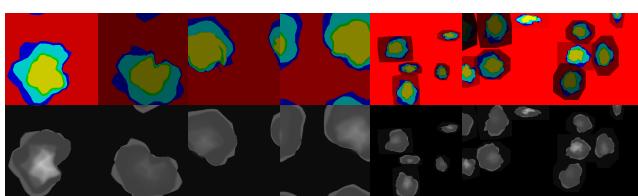
355 All augmentation techniques are applied both to the height field  
356 and the label map simultaneously to ensure consistency between  
357 the input (the label map) and the output (the height field).

#### 6. Learning-based generation

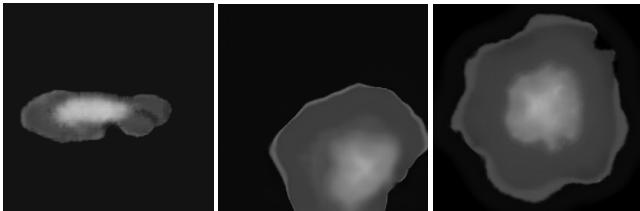
358 In this section, we introduce the use of a conditional Generative  
359 Adversarial Network (cGAN), specifically the pix2pix model pro-  
360 posed by [IZZE17], to enhance the island generation process by  
361 increasing terrain variety and flexibility. While the curve-based  
362 modeling algorithm can create numerous island examples, cGAN  
363 provides additional flexibility in generating more complex terrain

## 5. Data generation

364 The dataset is created using the procedural curve-based modeling  
365 algorithm, with altered input parameters. For each generation, the  
366 top-view and profile-view sketches use an initial layout. Each out-  
367 line of the top-view sketch is defined as a centered circle of random  
368 radius  $r_{\min} \leq r^* \leq r_{\max}$ . We add another deformation based on fBm



**Figure 11:** Examples from a dataset with increasing data augmen-  
tation.



**Figure 12:** After the first epoch (left), grid artifacts similar to a low-resolution image are visible, but are being corrected during a second epoch (middle) where erosion patterns appear in the height fields (right).

365 without the rigid constraints of the curve-based algorithm that stem  
366 from our initial assumptions based on coral reef formation theory.

367 The training was performed using a batch size of 1, and optimized  
368 using the Adam optimizer with a learning rate of 0.0002 and  
369 momentum parameters  $\beta_1 = 0.5$  and  $\beta_2 = 0.999$ . The loss function  
370 combined a conditional adversarial loss and an L1 loss, where the  
371 L1 loss was weighted by a factor  $\lambda = 100$ . The generator follows  
372 a U-Net architecture with encoder-decoder layers and skip connec-  
373 tions, while the discriminator was based on a PatchGAN, classi-  
374 fying local image patches. These parameters are directly adopted  
375 from the original paper.

376 During inference, our model, with a dataset of approximatively  
377 10 000 examples representing about 30 minutes of training, still  
378 outputs grid artifacts, visible in Figure 12. After the second epoch,  
379 visual artifacts are already rare. This training time seems long but,  
380 at the best of our knowledge, is the shortest training time for terrain  
381 generation with a learning-based approach in the literature, mostly  
382 due to the synthetic nature of our dataset.

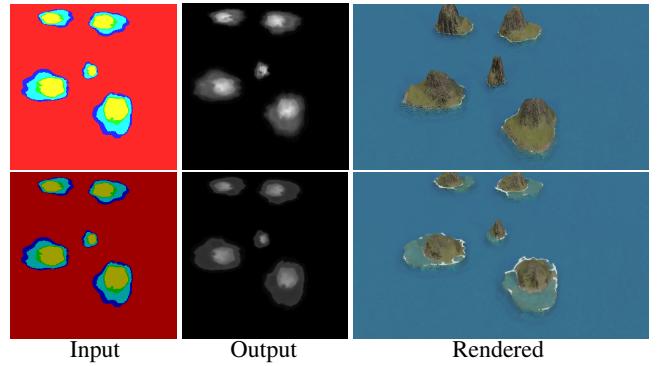
383 Once the model is trained, the user colors a  $256 \times 256$  RGB im-  
384 age to sketch the different regions. Each region is given a specific  
385 color based on the HSV encoding used for the dataset generation.  
386 The examples presented in this paper use red for abysses, blue for  
387 reefs, cyan for lagoons, green for beaches and yellow for moun-  
388 tains. The subsidence factor presented in Section 4.2.1 is control-  
389 lable through the luminosity of the input image.

390 The Python script for the initial island dataset generation is unop-  
391 timized and takes about 2.5s per island of size  $256 \times 256$  as we do  
392 not perform parallelization here. Implementing an optimized C++  
393 version of the initial generation process reduces this execution time  
394 to 50ms per generation.

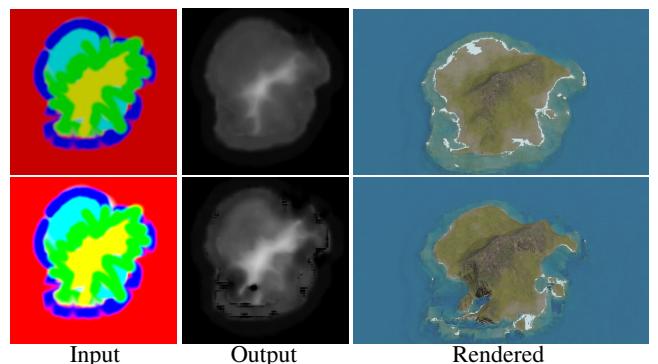
395 On the other hand, the inference time of our deep learning model  
396 for a single input image of dimension  $256 \times 256$  remains constant  
397 regardless of scene complexity. Using the NVIDIA GeForce GTX  
398 1650 Ti GPU with Python 3.10 and PyTorch version 2.5.1+cu121,  
399 the inference time measured is 5ms (std 1.1ms).

## 400 7. Results

401 The resulting model for coral island generation enables a high  
402 control-level from a user perspective as the unconstrained paint-  
403 ing allows for complex scenarios while producing in real-time the



**Figure 13:** An identical label map yields similar height fields over multiple inferences from the model, even after modifying the subsidence factor (visible in the luminosity of the input image).



**Figure 14:** User applied fuzzy brushes to draw the label map, resulting in some pixels that are inconsistent with the dataset and unlogical island layouts: abyss regions (red) are found between beach (green), lagoon (cyan) and reef regions (blue). The neural model ignores the layout inconsistencies, and partially over-brightness as long as the pixel isn't completely white.

404 resulting height fields. In this paper we used the software Blender  
405 to provide renders directly from the output height fields. As our  
406 pix2pix model is trained to output  $256 \times 256$  images, the resolution  
407 of the 3D models is limited by this architecture.

408 Using deep-learning-based models, most constraints from our  
409 initial assumptions are lifted (radial layout, isolated islands, ...).  
410 The control over the overall shapes of the islands regions are given  
411 through digital painting, here using the GIMP software. Each pixel  
412 of the image is encoded in HSV, with the region identifier encoded  
413 in the Hue channel. The user may increase or decrease the subsi-  
414 dence level of the island by modifying the Value channel over the  
415 whole image (see Figure 13).

416 Since the model relies on pixel-level statistics rather than strict  
417 class values, it is robust to noise caused by compression, anti-  
418 aliasing from brush tools, or resizing artifacts introduced by im-  
419 age editors. The example displayed in Figure 14 presents a sketch  
420 for which the outlines of the regions are at the same time blurry  
421 and with layouts that are not expected (such as the small red re-

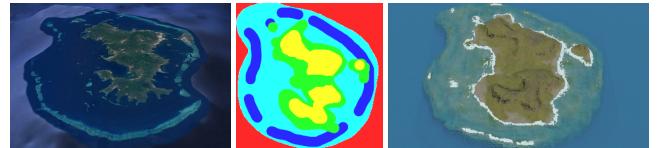
422 gions inside the southern lagoon region or the adjacency of beach  
 423 regions directly with the abyssal region) on the top figure and show-  
 424 ing highlight clipping on the bottom figure. The learned model does  
 425 not include inconsistencies and results in plausible 3D models. We  
 426 can note that the input label map isn't require to be hand-drawn,  
 427 as a simple Perlin noise can produce it randomly, as shown in the  
 428 examples of Figure 17.

429 The tolerance over the input values may be used to provide even  
 430 more control about the transitions between two regions. Figure 15  
 431 shows an example of input map with regions that are leaking over  
 432 neighboring regions, and the introduction of new hue values nonex-  
 433 istent in the dataset (light green and dark green) but are the inter-  
 434 polated hue value of mountain regions and beach regions.

435 Since the curve-based procedural phase included low random-  
 436 ness, the output of the cGAN is limiting its unpredictability, mean-  
 437 ing that small input changes results only in minor changes on the  
 438 output, preventing unexpected results. Figure 13 shows the result  
 439 of an input map with only a variation on the subsidence level, the  
 440 resulting height fields are very similar. Adding the real-time com-  
 441 putation of outputs, it becomes possible to construct progressively  
 442 a landscape and correct small mistakes to intuitively design islands  
 443 inspired by real-world regions. A creation of an island similar to  
 444 Mayotte, presented in Figure 16 was hand-made in just a few min-  
 445utes.

446 **Limitations** While this approach brings significant advantages,  
 447 there are also some limitations to consider. The reliance on a syn-  
 448 thetic dataset means that the cGAN inherits some biases and limita-  
 449 tions of our initial curve-based algorithm. This could limit the true  
 450 diversity of the terrains that the model can generate, as the output  
 451 is confined by the patterns present in the training data. Addition-  
 452 ally, the cGAN model's internal logic lacks transparency, offering  
 453 limited user control over the generation process once the model has  
 454 been trained. Moreover, the training time is far from real-time.

455 **Future work** Further improvements could be made to the syn-  
 456 thetic dataset. Incorporating more complex geological processes,  
 457 such as wave erosion or tidal influences, could lead to even more  
 458 realistic terrains. Additionally, refining the way islands are blended  
 459 in multi-island samples, or adding more diverse input conditions  
 460 (e.g., different geological settings), could help the model general-  
 461 ize better and produce more varied and dynamic landscapes. While  
 462 the current model allows for rapid terrain generation, adding more  
 463 options for users to interact with the cGAN, such as tweaking pa-



**Figure 16:** Comparison between real (left, from Google Earth) and synthetic (right) islands of Mayotte. The label map (center) is hand-drawn to match approximatively its real-world counterpart.

464 rameters like wind strength or island size, could enhance the flex-  
 465 ibility of the system. Many other neural network models could be  
 466 exploited to increase the possibilities, such as newer variants of  
 467 cGANs [PRHK19], or models with style transfer functionalities  
 468 [GEB15; ZCZ\*20] in order to change the overall aspect of a ter-  
 469 rain [PPB\*23a; PPB\*23b], use text-to-images models [RBL\*21;  
 470 RKH\*21] to generate height fields from a verbal prompt, or super-  
 471 resolution models [DLHT14] to increase the definition of details in  
 472 the final output [GDGP16].

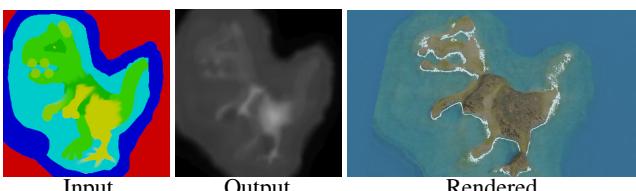
## 473 8. Conclusion

474 We presented a novel approach to generating coral reef island ter-  
 475 rains by combining traditional procedural methods with deep learn-  
 476 ing techniques. We first developed a procedural generation algo-  
 477 rithm capable of creating a wide variety of island terrains using  
 478 top-view and profile-view sketches, wind deformation, subsidence,  
 479 and coral reef growth simulation. By applying these methods, we  
 480 were able to produce realistic terrains based on geological pro-  
 481 cesses, capturing key features of coral reef islands such as beaches,  
 482 lagoons, and reefs.

483 To further enhance flexibility and realism in the generation pro-  
 484 cess, we incorporated a conditional Generative Adversarial Net-  
 485 work (cGAN), using the pix2pix model to translate island feature  
 486 label maps into height maps. The cGAN model helped to overcome  
 487 some of the constraints inherent in the procedural algorithm, such  
 488 as radial symmetry and fixed island positioning. Using data aug-  
 489 mentation techniques, we trained the cGAN on a synthetic dataset  
 490 to produce varied and realistic island terrains.

## 491 References

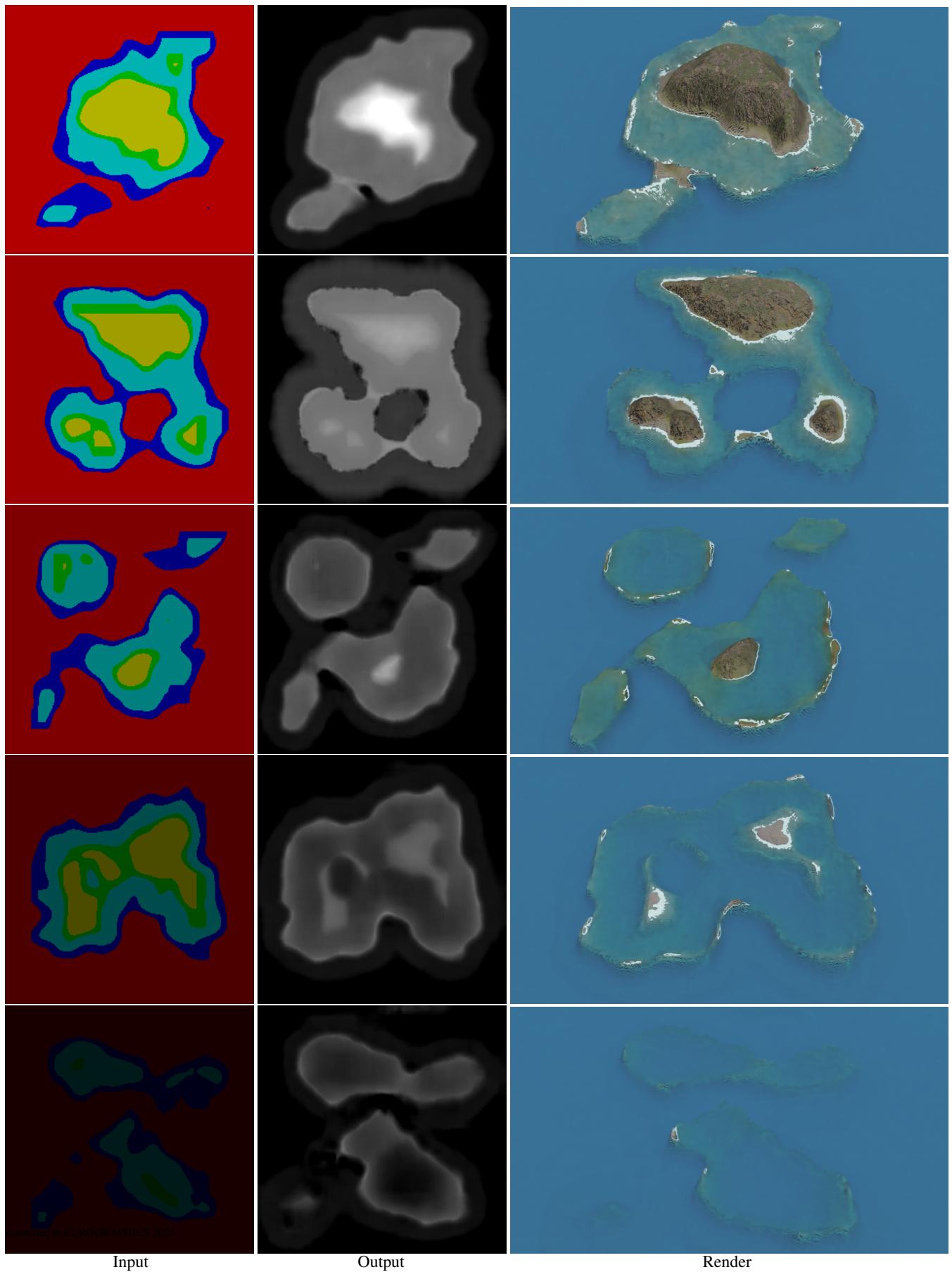
- [BP17] BECKHAM, CHRISTOPHER and PAL, CHRISTOPHER. "A step towards procedural terrain generation with GANs". (July 2017). URL: <http://arxiv.org/abs/1707.03383> 2.
- [BTHB06] BENEŠ, BEDŘICH, TĚŠÍNSKÝ, VÁCLAV, HORNYŠ, JAN, and BHATIA, SANJIV K. "Hydraulic erosion". *Computer Animation and Virtual Worlds* 17 (2 2006), 99–108. ISSN: 15464261. DOI: 10.1002/cav.77 2.
- [CBC\*16] CORDONNIER, GUILLAUME, BRAUN, JEAN, CANI, MARIE-PAULE, et al. "Large Scale Terrain Generation from Tectonic Uplift and Fluvial Erosion". *Computer Graphics Forum* 35 (2 May 2016), 165–175. ISSN: 0167-7055. DOI: 10.1111/cgf.12820. URL: <https://onlinelibrary.wiley.com/doi/10.1111/cgf.12820> 2.



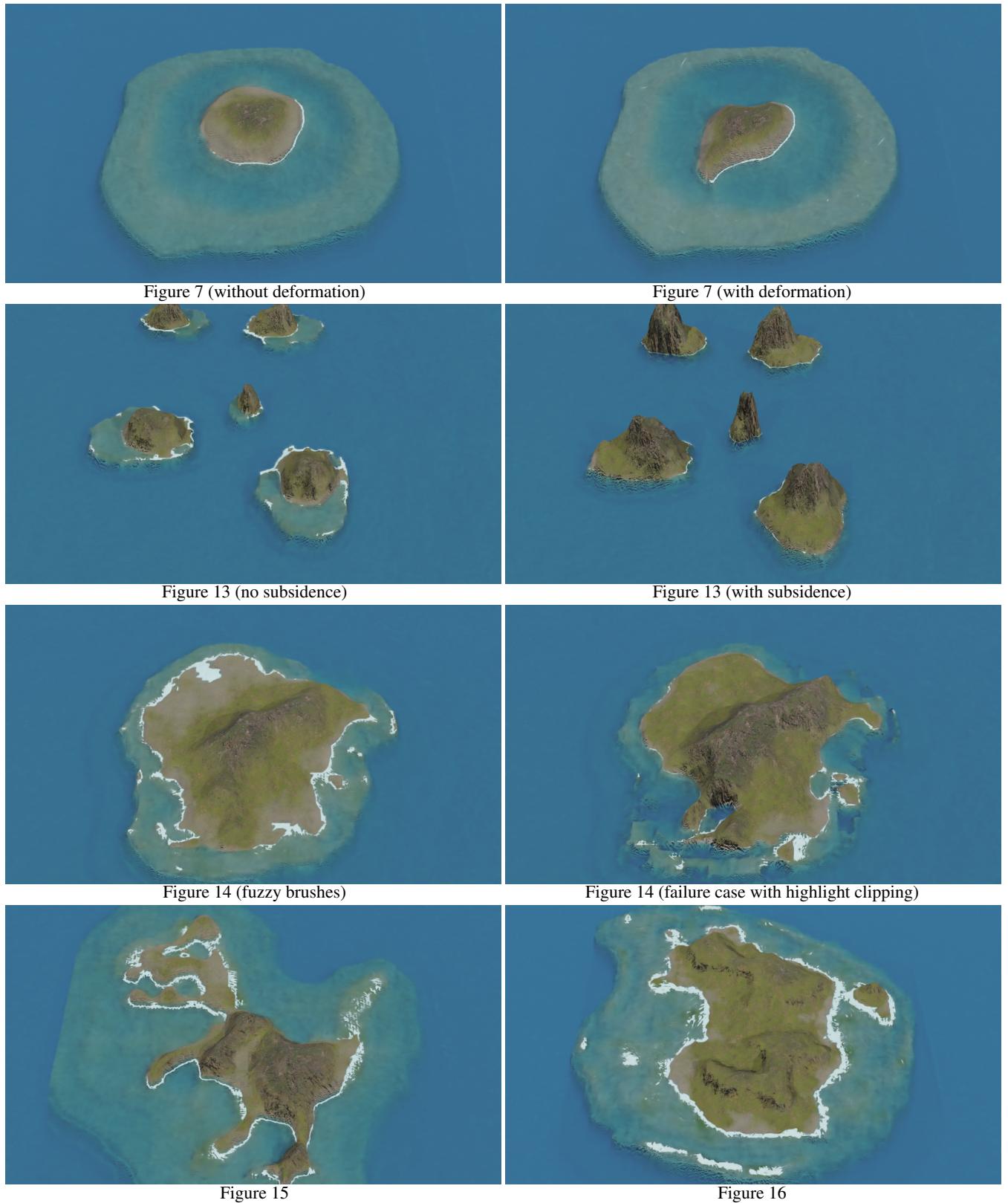
**Figure 15:** Without constraints on the generation, the user may use unrealistic layout and the neural network will however output a plausible result. Here new colors has been added (pistachio), but the network naturally consider it as a transition from mountain to beach.

- 504 [CCB\*17] CORDONNIER, GUILLAUME, CANI, MARIE-PAULE, BENEŠ, 568  
 505 BEDŘICH, et al. "Sculpting Mountains: Interactive Terrain Modeling 569  
 506 Based on Subsurface Geology". *IEEE Transactions on Visualization 570*  
 507 and Computer Graphics
- 508 24 (2017). DOI: 10.1109/TVCG.2017.2689022. URL: [http://www.ieee.org/publications\\_standards/publications/rights/index.html2.572](http://www.ieee.org/publications_standards/publications/rights/index.html2.572). 573
- 510 [CGG\*17] CORDONNIER, GUILLAUME, GALIN, ÉRIC, GAIN, JAMES, 574  
 511 et al. "Authoring landscapes by combining ecosystem and terrain ero- 575  
 512 sion simulation". *ACM Transactions on Graphics* 36 (4 2017). ISSN: 576  
 513 15577368. DOI: 10.1145/3072959.3073667. URL: <https://hal.archives-ouvertes.fr/hal-01518967/file/authoring-landscapes-combining.pdf2.577>. 578
- 514 [DLHT14] DONG, CHAO, LOY, CHEN CHANGE, HE, KAIMING, and 580  
 515 TANG, XIAOOU. "Image Super-Resolution Using Deep Convolutional 581  
 516 Networks". (Dec. 2014). URL: <http://arxiv.org/abs/1501.00092v8.582>. 583
- 517 [ECC\*21] ECORMIER-NOCCA, PIERRE, CORDONNIER, GUILLAUME, 584  
 518 CARREZ, PHILIPPE, et al. "Authoring consistent landscapes with 585  
 519 flora and fauna". *ACM Transactions on Graphics* 40 (4 2021). ISSN: 586  
 520 15577368. DOI: 10.1145/3450626.3459952. URL: [https://www-sop.inria.fr/reves/Basilic/2021/ECCMMBC21/Authoring\\_Consistent\\_Landscapes\\_with\\_Flora\\_and\\_Fauna.pdf2.587](https://www-sop.inria.fr/reves/Basilic/2021/ECCMMBC21/Authoring_Consistent_Landscapes_with_Flora_and_Fauna.pdf2.587). 588
- 521 [EPW\*03] EBERT, DAVID S., PEACHEY, DARWYN, WORLEY, STEVEN, 589  
 522 et al. *Texturing and Modeling, A Procedural Approach*. Vol. Third edi- 590  
 523 tion. Morgan Kaufmann, 2003. ISBN: 978-1-55860-848-1 2. 591
- 524 [FFC82] FOURNIER, ALAIN, FUSSELL, DON, and CARPENTER, LOREN. 592  
 525 "Computer rendering of stochastic models". *Communications of the 593 ACM* 25 (6 June 1982), 371–384. ISSN: 0001-0782. DOI: 10.1145/594  
 526 358523.358553. URL: <https://dl.acm.org/doi/10.1145/358523.358553.2>. 595
- 527 [GCRR20] GASCH, CRISTINA, CHOVER, MIGUEL, REMOLAR, INMAC- 596  
 528 ULADA, and REBOLLO, CRISTINA. "Procedural modelling of ter- 597  
 529 rains with constraints". *Multimedia Tools and Applications* 79 (41–42 598  
 530 2020), 31125–31146. ISSN: 15737721. DOI: 10.1007/s11042-020-09476-3 2. 599
- 531 [GDG\*17] GUÉRIN, ÉRIC, DIGNE, JULIE, GALIN, ÉRIC, PEYTAVIE, 600  
 532 ADRIEN, et al. "Interactive example-based terrain authoring with 601  
 533 conditional generative adversarial networks". *ACM Transactions on Graphics 602*  
 534 36 (6 2017). ISSN: 15577368. DOI: 10.1145/3130800.3130804 2. 603
- 535 [GDGP16] GUÉRIN, ÉRIC, DIGNE, JULIE, GALIN, ÉRIC, and PEYTAVIE, 604  
 536 ADRIEN. "Sparse representation of terrains for procedural modeling". 605  
 537 *Computer Graphics Forum* 35 (2 2016), 177–187. ISSN: 14678659. DOI: 606  
 538 10.1111/cgf.12821 8. 607
- 539 [GEB15] GATYS, LEON A., ECKER, ALEXANDER S., and BETHGE, 608  
 540 MATTHIAS. "A Neural Algorithm of Artistic Style". (Aug. 2015). URL: 609  
 541 <http://arxiv.org/abs/1508.06576v8.610>. 611
- 542 [GGP\*15] GÉNEVAUX, JEAN-DAVID, GALIN, ÉRIC, PEYTAVIE, 612  
 543 ADRIEN, et al. "Terrain Modelling from Feature Primitives". (2015). 613  
 544 DOI: <https://doi.org/10.1111/cgf.12530.2.614>. 615
- 545 [GGP\*19] GALIN, ERIC, GUÉRIN, ERIC, PEYTAVIE, ADRIEN, et al. "A 616  
 546 Review of Digital Terrain Modeling". *Computer Graphics Forum* 38 (2 617  
 547 May 2019), 553–577. ISSN: 0167-7055. DOI: 10.1111/cgf.13657. 618  
 548 URL: <https://onlinelibrary.wiley.com/doi/10.1111/cgf.13657.2.619>. 619
- 549 [GMS09] GAIN, JAMES, MARAIS, PATRICK, and STRASSER, WOLF- 620  
 550 GANG. "Terrain sketching". *Proceedings of I3D 2009: The 2009 ACM 621  
 551 SIGGRAPH Symposium on Interactive 3D Graphics and Games* 1 (212 622  
 552 2009), 31–38. DOI: 10.1145/1507149.1507155 2. 623
- 553 [GPM\*22] GUÉRIN, ERIC, PEYTAVIE, ADRIEN, MASNOU, SIMON, et al. 624  
 554 "Gradient Terrain Authoring". *Computer Graphics Forum* 41 (2 May 625  
 555 2022), 85–95. ISSN: 0167-7055. DOI: 10.1111/cgf.14460. URL: 626  
 556 <https://onlinelibrary.wiley.com/doi/10.1111/cgf.14460.2.627>. 627
- 557 [HGA\*10] HNAIDI, HOSSAM, GUÉRIN, ERIC, AKKOUCH, SAMIR, et 628  
 558 al. "Feature based terrain generation using diffusion equation". *Com- 629  
 559 puter Graphics Forum* 29 (7 Sept. 2010), 2179–2186. ISSN: 0167-7055. 630  
 560 DOI: 10.1111/j.1467-8659.2010.01806.x. URL: <https://onlinelibrary.wiley.com/doi/10.1111/j.1467-8659.2010.01806.x.2>. 631
- 561 [Hop14] HOPLEY, DAVID. *Encyclopedia of modern coral reefs : struc- 632  
 562 ture, form and process*. Springer, Credo Reference, 2014. ISBN: 9789048126385 1. 633
- 563 [IZZE17] ISOLA, PHILLIP, ZHU, JUN-YAN, ZHOU, TINGHUI, and 634  
 564 EFROS, ALEXEI A. "Image-to-Image Translation with Conditional Ad- 635  
 565 versarial Networks". *2017 IEEE Conference on Computer Vision and 636  
 566 Pattern Recognition (CVPR)*. IEEE, July 2017, 5967–5976. ISBN: 978-637  
 567 1-5386-0457-1. DOI: 10.1109/CVPR.2017.632. URL: <https://arxiv.org/abs/1611.07004%20http://ieeexplore.ieee.org/document/8100115/2.638>.
- 568 [KSZB14] KÖRNER, CHRISTIAN, SPEHN, EVA M., BUGMANN, HAR- 638  
 569 ALD, and ZURICH, ETH. *Mountain Systems*. Tech. rep. 2014. URL: 639  
 570 <https://www.researchgate.net/publication/238663492.1>. 640
- 571 [LMK\*24] LYONS, MITCHELL B., MURRAY, NICHOLAS J., KENNEDY, 641  
 572 EMMA V., et al. "New global area estimates for coral reefs from 642  
 573 high-resolution mapping". *Cell Reports Sustainability* 1 (2 Feb. 643  
 574 2024), 100015. ISSN: 29497906. DOI: 10.1016/j.crsus.2024.100015 1. 644
- 575 [MDH07] MEI, XING, DECAUDIN, PHILIPPE, and HU, BAO GANG. "Fast 645  
 576 hydraulic erosion simulation and visualization on GPU". *Proceedings - 646  
 577 Pacific Conference on Computer Graphics and Applications (2007)*, 47– 647  
 578 56. ISSN: 15504085. DOI: 10.1109/PG.2007.27. URL: <https://xing-mei.github.io/files/erosion.pdf2>. 648
- 579 [MKM89] MUSGRAVE, FOREST KENTON, KOLB, CRAIG E., and MACE, 649  
 580 ROBERT S. "The synthesis and rendering of eroded fractal terrains". *Pro- 650  
 581 ceedings of the 16th Annual Conference on Computer Graphics and In- 651  
 582 ternational Techniques, SIGGRAPH 1989* (1989), 41–50. DOI: 10.1145/ 652  
 583 74333.74337. URL: <https://citesearx.ist.psu.edu/viewdoc/download?doi=10.1.1.27.8939&rep=rep1&type=pdf2>. 653
- 584 [MO14] MIRZA, MEHDI and OSINDERO, SIMON. "Conditional Genera- 654  
 585 tive Adversarial Nets". (Nov. 2014). URL: <http://arxiv.org/abs/1411.1784.2>. 656
- 586 [NVP12] NATALI, MATTIA, VIOLA, IVAN, and PATEL, DANIEL. "Rapid 657  
 587 visualization of geological concepts". *Brazilian Symposium of Computer 658  
 588 Graphic and Image Processing* (2012), 150–157. ISSN: 15301834. DOI: 659  
 589 10.1109/SIBGRAPI.2012.29 2. 660
- 590 [NWD05] NEIDHOLD, B., WACKER, M., and DEUSSEN, OLIVER. 661  
 591 "Interactive physically based fluid and erosion simulation". *Natu- 662  
 592 ral Phenomena* (2005), 25–32. ISSN: 18160867. URL: <http://graphics.uni-konstanz.de/publikationen/Neidhold2005InteractivePhysicallyBased/> 663  
 593 <http://Neidhold2005InteractivePhysicallyBased.pdf2>. 664
- 594 [Ols04] OLSEN, JACOB. "Realtime procedural terrain generation". 665  
 595 *Department of Mathematics And Computer Science* (2004), 20. 666  
 596 URL: [https://pdfs.semanticscholar.org/5961/c577478f21707dad53905362e0ec4e6ec644.pdf%5Cnhttp://www.tsinghua.edu.cn/PRREC/terrain\\_generation.pdf%5Cnhttp://web.mit.edu/cesium/Public/terrain.pdf2](https://pdfs.semanticscholar.org/5961/c577478f21707dad53905362e0ec4e6ec644.pdf%5Cnhttp://www.tsinghua.edu.cn/PRREC/terrain_generation.pdf%5Cnhttp://web.mit.edu/cesium/Public/terrain.pdf2). 667
- 597 [PC20] PANAGIOTOU, EMMANOUIL and CHAROU, ELENI. "Procedural 668  
 598 3D Terrain Generation using Generative Adversarial Networks". (Oct. 669  
 599 2020). URL: <http://arxiv.org/abs/2010.06411.2>. 670
- 600 [Per01] PERLIN, KEN. "Noise Hardware". *Real-Time Shading SIGGRAPH Course Notes* (2001) 2. 671
- 601 [Per85] PERLIN, KEN. "An image synthesizer". *ACM SIGGRAPH Computer Graphics* 19 (3 July 1985), 287–296. ISSN: 0097-8930. DOI: 10. 672  
 602 1145/325165.325247. URL: <https://dl.acm.org/doi/10.1145/325165.325247.2>. 673

- 634 [PPB\*23a] PERCHE, SIMON, PEYTAVIE, ADRIEN, BENES, BEDRICH, et  
635 al. "Authoring Terrains with Spatialised Style". *Computer Graphics Forum* 42 (7 Oct. 2023). ISSN: 14678659. DOI: 10.1111/cg.14936 8.
- 637 [PPB\*23b] PERCHE, SIMON, PEYTAVIE, ADRIEN, BENES, BEDRICH, et  
638 al. "StyleDEM: a Versatile Model for Authoring Terrains". (Apr. 2023).  
639 URL: <http://arxiv.org/abs/2304.09626> 8.
- 640 [PRHK19] PARK, JOSEPH, REDWINE, JED, HILL, TROY D., and KOTUN,  
641 KEVIN. "Water resource and ecotone transformation in coastal ecosystems". *Ecological Modelling* 405 (Aug. 2019), 69–85. ISSN: 03043800.  
642 DOI: 10.1016/j.ecolmodel.2019.04.015 8.
- 644 [RBL\*21] ROMBACH, ROBIN, BLATTMANN, ANDREAS, LORENZ, DO-  
645 MINIK, et al. "High-Resolution Image Synthesis with Latent Diffusion  
646 Models". (Dec. 2021). URL: <http://arxiv.org/abs/2112.10752> 8.
- 648 [RKH\*21] RADFORD, ALEC, KIM, JONG WOOK, HALLACY, CHRIS, et  
649 al. "Learning Transferable Visual Models From Natural Language Su-  
650 pervision". (Feb. 2021). URL: <http://arxiv.org/abs/2103.00020> 8.
- 652 [Sis22] SISODIA, YOGENDRA. "GAN-Generated Terrain for Game As-  
653 sets". *Indian Journal of Artificial Intelligence and Neural Network-  
654 ing* 2 (6 Oct. 2022), 1–3. ISSN: 25827626. DOI: 10.54105/ijainn.F1060.102622. URL: <https://www.ijainn.latticescipub.com/portfolio-item/f1060102622/> 2.
- 657 [SKG\*09] SMELIK, RUBEN M., KRAKER, KLAAS JAN DE, GROENEWE-  
658 GEN, SASKIA A, et al. "A survey of procedural methods for terrain mod-  
659 elling". *Proceedings of the CASA workshop on 3D advanced media in  
660 gaming and simulation (3AMIGAS)* (2009) 2.
- 661 [SPF\*23] SCHOTT, HUGO, PARIS, AXEL, FOURNIER, LUCIE, et al.  
662 "Large-scale Terrain Authoring through Interactive Erosion Simulation".  
663 *ACM Transactions on Graphics* 42 (5 Oct. 2023), 1–15. ISSN: 0730-  
664 0301. DOI: 10.1145/3592787. URL: <https://dl.acm.org/doi/10.1145/3592787> 2.
- 666 [SW19] SPICK, RYAN and WALKER, JAMES ALFRED. "Realistic and  
667 textured terrain generation using GANs". *Proceedings - CVMP 2019:  
668 16th ACM SIGGRAPH European Conference on Visual Media Pro-  
669 duction*. Association for Computing Machinery, Inc, Dec. 2019. ISBN:  
670 9781450370035. DOI: 10.1145/3359998.3369407 2.
- 671 [TB18] TALGORN, FRANÇOIS XAVIER and BELHADJ, FARÈS. "Real-  
672 time sketch-based terrain generation". *ACM International Conference  
673 Proceeding Series* (March 2021 2018), 13–18. DOI: 10.1145/3208159.  
674 3208184. URL: [https://www.researchgate.net/profile/Fares-Belhadj/publication/325327322\\_Real-Time-Sketch-Based-Terrain-Generation/links/6054c4b4a6fdccbfeaf0a8b2/Real-Time-Sketch-Based-Terrain-Generation.pdf](https://www.researchgate.net/profile/Fares-Belhadj/publication/325327322_Real-Time-Sketch-Based-Terrain-Generation/links/6054c4b4a6fdccbfeaf0a8b2/Real-Time-Sketch-Based-Terrain-Generation.pdf) 2.
- 679 [VMP21] VOULGARIS, GEORGIOS, MADEMLIS, IOANNIS, and PITAS,  
680 IOANNIS. "Procedural Terrain Generation Using Generative Adversarial  
681 Networks". *European Signal Processing Conference*. Vol. 2021-August.  
682 European Signal Processing Conference, EUSIPCO, 2021, 686–690.  
683 ISBN: 9789082797060. DOI: 10.23919/EUSIPCO54536.2021.  
684 9616151 2.
- 685 [WRMB18] WULFF-JENSEN, ANDREAS, RANT, NICLAS NERUP,  
686 MØLLER, TOBIAS NORDVIG, and BILLESKOV, JONAS AKSEL. "Deep  
687 Convolutional Generative Adversarial Network for Procedural 3D Land-  
688 scape Generation Based on DEM". *Interactivity, Game Creation, De-  
689 sign, Learning, and Innovation*. Springer, Jan. 2018, 85–94. DOI: 10.  
690 1007/978-3-319-76908-0\_9 2.
- 691 [ZCZ\*20] ZHU, DI, CHENG, XIMENG, ZHANG, FAN, et al. "Spatial in-  
692 terpolation using conditional generative adversarial neural networks".  
693 *International Journal of Geographical Information Science* 34 (4 Apr.  
694 2020), 735–758. ISSN: 13623087. DOI: 10.1080/13658816.2019.  
695 1599122 8.



**Figure 17:** Starting from random Perlin noise, transformed into label maps with increasing subsidence, we can generate a large variety of results.



**Figure 18:** High-resolution version of figures found in this paper.