DOI: 10.1111/cgf.14941

Pacific Graphics 2023
R. Chaine, Z. Deng, and M. H. Kim
(Guest Editors)

COMPUTER GRAPHICS *forum*
Volume 42 (2023), Number 7

Interactive Authoring of Terrain using Diffusion Models

J. Lochner[1], J. Gain[1], S. Perche[2], A. Peytavie[2], E. Galin[2] and E. Guérin[3].

[1]University of Cape Town, South Africa
[2]Univ Lyon, Université Lyon 1, CNRS, LIRIS, France
[3]Univ Lyon, INSA-Lyon, CNRS, LIRIS, France

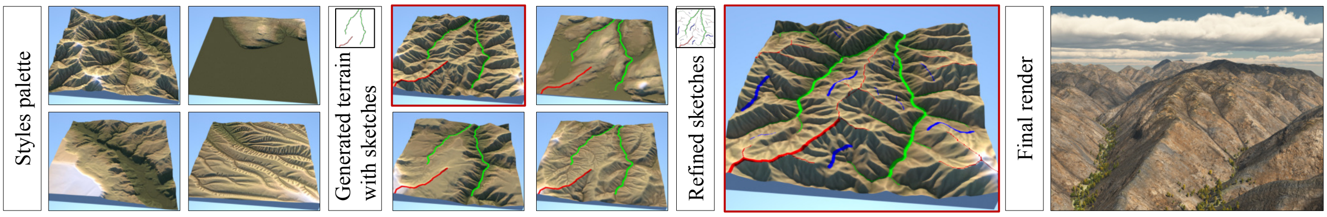

Figure 1: *Our diffusion-based terrain authoring framework empowers users to iteratively combine terrain stylization with feature sketching. Here, an artist first sketches a simple ridge and forked drainage network, then makes their style selection (outlined in red) before adding further details.*

Abstract

Generating heightfield terrains is a necessary precursor to the depiction of computer-generated natural scenes in a variety of applications. Authoring such terrains is made challenging by the need for interactive feedback, effective user control, and perceptually realistic output encompassing a range of landforms. We address these challenges by developing a terrain-authoring framework underpinned by an adaptation of diffusion models for conditional image synthesis, trained on real-world elevation data. This framework supports automated cleaning of the training set; authoring control through style selection and feature sketches; the ability to import and freely edit pre-existing terrains, and resolution amplification up to the limits of the source data. Our framework improves on previous machine-learning approaches by: expanding landform variety beyond mountainous terrain to encompass cliffs, canyons, and plains; providing a better balance between terseness and specificity in user control, and improving the fidelity of global terrain structure and perceptual realism. This is demonstrated through drainage simulations and a user study testing the perceived realism for different classes of terrain. The full source code, blender add-on, and pre-trained models are available.

1. Introduction

The need for authoring tools that enable digital artists to create convincing heightfield terrains suited to a particular aesthetic or functional rôle is widespread in computer graphics applications, such as film, games, training, and simulation. Using scanned real-world terrain often does not suffice, since landforms may be missing or not arranged acceptably. Once authored, a bare heightfield is typically layered with surface detail to depict earth, rock, grass, plants, trees, rivers, and bodies of water, representing a complete natural scene.

The effective authoring of synthetic terrain is a notoriously difficult problem to solve for many reasons. First, achieving perceptual realism is a challenge due to the complexity of the underlying physical processes that interact to shape terrain. For example, shifting tectonic plates, gradual erosive forces, and even natural disasters contribute to terrain formation. Second, the sheer variety of formative processes gives rise to a correspondingly wide range of distinctive landforms, from plains to mountains and every style in-between. Third, authoring tools should afford a balance between design economy and precision so that artists can achieve envisaged results (precision) as tersely and rapidly as possible (economy). Finally, allied to this is the need for a computationally efficient real-time or interactive generation process to enable cycles of iterative design. Ideally, a terrain authoring framework should produce perceptual realism, landform diversity, both economical and precise authoring, and interactive response.

To address this, terrain authoring has historically been built on a foundation of procedural, simulation, or example-based techniques [GGP*19]. Recently, these have been complemented by generative machine learning, where deep neural networks — including Convolutional Neural Networks (CNNs) [ACA18, KSR20] and Conditional Generative Adversarial Networks (CGANs) [GAMA20, ZLZ*22, ZLB*19, ZCZ*20, NJSR22] — are tasked with learning the complex patterns and mutual dependencies between terrain features, given abundant real-life data, and conditioned on user inputs, such as user sketches or style selection. Unfortunately, these strategies suffer from repetition and gridding artifacts with sparse inputs [GAMA20], tend to tackle only specific sub-tasks (e.g., super-resolution) rather than providing an overarching authoring framework, and also, with a few exceptions [LLXT22], focus primarily on mountainous regions, ignoring plains, cliffs, and canyons.

Inspired by evidence of the effectiveness of Denoising Diffusion Probabilistic Models (DDPMs) in conditional image generation [Luo22] with high sample diversity [DN21], we develop a feature-rich terrain authoring framework that performs cascaded applications of diffusion models. Our framework supports interactive authoring through style selection and feature sketching (see Figure 1), followed by a finalizing amplification step. Performance is sufficient to enable near real-time updates ($\sim$ 8Hz) during sketching, with additional iterative quality refinement taking place in the background between strokes. To achieve this, we balance sample quality and inference speed by trading off terrain resolution, number of sampling timesteps, and memory footprint. The resulting synthesized terrains encompass a wide range of landforms and evidence improved structural and perceptual realism. Specifically, we found the results of diffusion to be structurally sound but sometimes missing in fine-scale detail so that users in our perceptual study were able to differentiate between real and generated landscapes in certain cases, but not in others.

2. Related Work

Traditionally, digital terrain generation techniques, as surveyed by Galin *et al.* [GGP*19], have been partitioned into three categories: procedural modeling, geomorphological simulation, and data-driven synthesis. Procedural approaches algorithmically reverse-engineer terrain based on characteristics of the final appearance using a collection of techniques that include constrained multiresolution noise, diffusion, and deformation. Simulation seeks to replicate the physical processes of uplift and erosion involved in landscape formation. However, our focus in this paper is on data-driven methods that exploit the extensive high-quality corpus of scanned real-world terrain data and offer the prospect of effective user control, interactive performance, and perceptual realism.

The earliest data-driven methods repurposed texture synthesis for terrain generation by cutting, reassembling, and joining terrain fragments at a patch or pixel level [LWZ*06, ZSTR07]. Subsequent research built on this foundation to improve: realism by altering patch blending [TGM12] and fixing global drainage [SD21, SD22], performance through GPU acceleration [TGM12, GMM15], and authoring control by extending beyond ridge and valley sketching [GMM15]. Compared to machine learning, the upside is that these methods can operate parsimoniously with as little as a single source terrain, but the downside is that re-assembly is localized and, despite some attempts at improvement [SD21], does not always respect global properties, such as realistic drainage patterns.

Sparse modeling [GDGP16, AAC*17] represents another example-based approach. It is underpinned by compressed sensing theory and relies on a dictionary of reusable terrain patches, which are sampled and blended at run-time to construct a terrain. However, these techniques suffer from the same limitations as texture synthesis methods and lend themselves more to compression and amplification than authoring [GGP*19].

Inspired by the success of image-to-image translation [IZZE17], Guérin *et al.* [GDG*17] were the first to apply machine learning to terrain authoring. They use a conditional generative adversarial network trained on the pairing of real terrains and extracted ridge and valley line maps to enable the interactive generation of plausible terrain from user sketches. This approach is flexible enough to support a range of authoring tools and tasks, including in-filling missing areas of a DEM, user support for painting average elevations or water occupancy [VRGZS20], and upsampling with erosion detail, but suffers from gridding and repetition artifacts when the user inputs are not sufficiently detailed.

A standard CGAN or CNN pipeline generates a single outcome even though two-dimensional user sketches are highly underdetermined. In reality, the same sketched configuration of ridge and valley lines could plausibly lead to various landforms, representing different terrains types or styles. An obvious solution to this stylization problem is to train separate synthesizers for each terrain type, but this inherently limits the number of styles and is a bar to transitional regions that mix styles. Better strategies involve making the training and synthesis conditional on a multi-dimensional style discriminator [ZLZ*22] or selecting variations from a latent space [NJSR22].

Depending on the details of the scanning campaign, Digital Elevation Models (DEMs) of real-world terrain can have varying resolutions, from as fine as 1 to as coarse as 90 meters per pixel. As a consequence, resolution amplification (otherwise known as upsampling or super-resolution) is a common application area for machine learning that has seen various recent experiments in modifying CGAN [ZCZ*20, ZLB*19] and CNN [KSR20, ACA18] architectures.

The vast majority of these data-driven methods have used adaptations of texture synthesis, CNNs, or CGANs specialized for terrain generation. However, in the space of image-to-image translation, of which user-guided terrain synthesis is a specific instance, Denoising Diffusion Probabilistic Models (DDPMs, or simply, diffusion models) have recently emerged as a highly-competitive alternative [HJA20, YZS*22] in terms of task versatility and perceptual image quality. The focus of this paper is thus on exploring the applicability of this class of generative models to terrain synthesis, while at the same time expanding the richness of the authoring toolset.

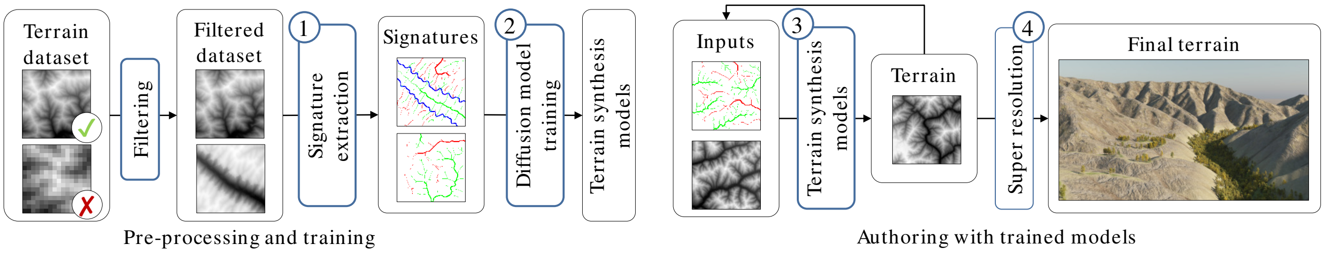

Figure 2: *An overview of diffusion-based terrain authoring. During pre-processing and training, a database of real-world terrains is constructed by filtering and downsampling. Then signatures, consisting of a feature sketch and style vector, are derived for these terrains (1) and used to train an ensemble of diffusion models (2). During iterative authoring, the trained models are used to synthesize terrains based on styles and sketches provided by an artist (3). Once the artist is satisfied, the terrain can be upsampled (4) to create a final high-resolution version.*

3. Overview

Our framework supports authoring in two complementary forms: feature sketching and example-based stylization. At a high level, feature sketching provides users with an intuitive way to specify the type and placement of features on the landscape, while stylization controls the overall appearance of the resulting terrain (using examples for guidance). As demonstrated in Figure 7, users are provided with tools to sketch cliff, valley, and ridge lines at different levels of prominence. This expands significantly on the typical toolset offered by machine learning interfaces. Notably, blank areas in the sketch are nevertheless populated with plausible features, but this can be suppressed by painting with a flattening brush. Furthermore — in the interests of iterative refinement — users are offered a palette of available styles, both before and after sketch input, to control the overall terrain character.

By design, the combination of a style and detailed sketch map provides a signature sufficient for terrain reconstruction, meaning a user-provided terrain can be imported, matched, and then edited. This idea is demonstrated in Figure 3, where localized changes can be made to an existing terrain in a manner that balances realism and user intent.

As is typical in machine learning this is realized by separating the architecture into a pre-processing and training phase, and an online generation phase (see Figure 2). Our data preparation (described in section 4), centers around the training and utilization of CNN-based classifiers to remove terrain that contains recording errors or artificial features, followed by signature extraction and model training.

In terms of models, our framework is built upon two different types of diffusion-based terrain synthesizers (detailed in section 5). First, our *sketch-to-terrain* model $\mathcal{S}$ generates terrain from a signature that combines a sketch and style vector. Second, our *terrain-upscaling* models $\mathcal{U}_1$ and $\mathcal{U}_2$ enhance the small-scale details of terrain in a realistic way, being able to increase terrain resolution from 153m per pixel to 19.1m per pixel and from 19.1m per pixel to 2.39m per pixel, respectively. Thus, in a similar fashion to Ho *et al.* [HSC*22], we cascade these diffusion models (i.e., using the output of one as input to the next) to achieve high-resolution terrain synthesis. As demonstrated in Figure 4, this results in a 64× resolution increase, in a manner that preserves the global structure of the output terrain.

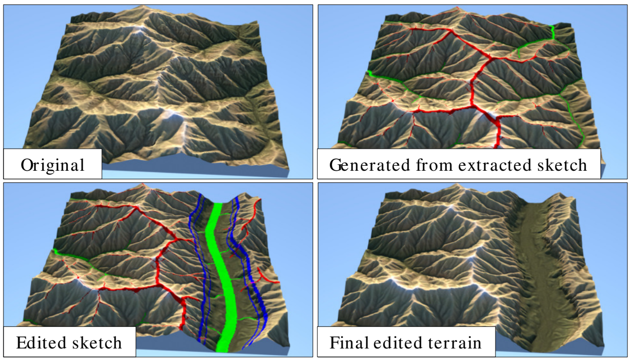

Figure 3: *Inverse terrain modeling: a structural replica of a terrain can be generated from the extracted signature and subsequently edited by the user to introduce or remove features.*

4. Data Preparation

We obtained raw elevation and co-registered satellite data from the United States Geological Survey (USGS). To select suitable regions, we submitted a user-defined mask created by taking into account elevation, gradient, and population-density maps of North America. This approach ensures that the final dataset includes a diverse selection of landforms. The corresponding ~ 6 million tiles (stored individually as 256×256 float arrays) were downloaded at the highest available resolution of ~ 2.39 meters per pixel.

4.1. Preprocessing

Unfortunately, many of the downloaded tiles contain recording errors, making them unusable for training. Failure to remove such data would degrade output quality, with artifacts or sampling errors replicated (or even magnified) in the synthesized terrains [GGP*19].

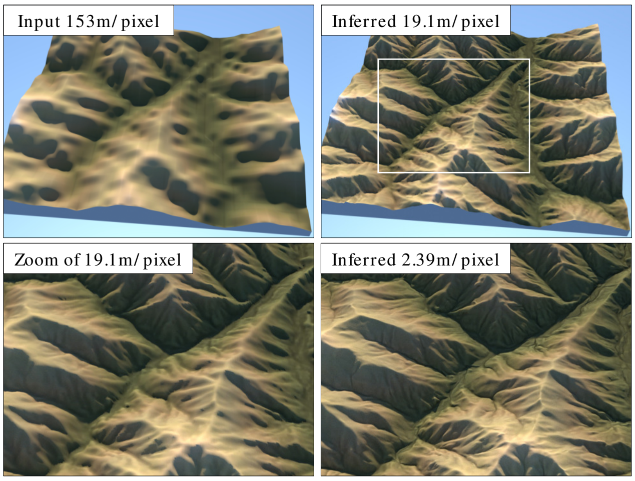

Figure 4: *Super-resolution upsampling from a coarse resolution of* 153*m per pixel to the limits of the original source data at* 2.39*m per pixel.*

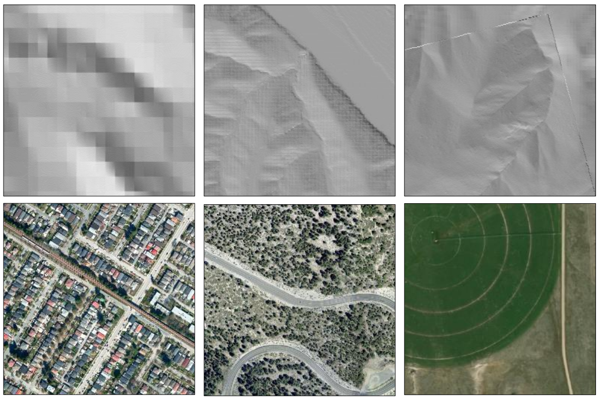

Figure 5: *Categories of excluded source data: [top] recording errors - pixelation, patching, and line-artifacts; [bottom] artificial elements - buildings, roads, and farmland.*

We identify the following classes of derivative data (see Figure 5[top]): *pixelated* (blocky image), *patched* (checkerboard pattern over the image), *line-artifacts* (stitch lines), and *valid* (no recording errors). The process of detecting anomalies is simplified by first converting each heightmap to the gradient domain, which enables more accurate identification of local changes and inconsistencies.

We first performed a combination of algorithmic and manual labeling of approximately 350 000 derivative images. More specifically, we defined a list of simple heuristics to identify abnormal changes in elevation, followed by a manual moderation process to ensure edge-cases were correctly labeled. This initial batch was then used to train a CNN classifier based on EfficientNetV2 [TL21] to filter the remaining tiles. We also removed terrains containing man-made structures (see Figure 5[bottom]), such as buildings, roads, and farmland. To this end, an additional classifier was trained on around 200 000 manually-labeled satellite images. Note that it is specifically for this purpose that we downloaded registered satellite data, which is not used in the remainder of the pipeline.

Filtering out artificial elements was performed iteratively, continually improving and expanding the training set until a satisfactory validation accuracy was achieved. Once again, this was done by manually reviewing subsets of the predictions made by the classifier. Only tiles free from artifacts and artificial structures were included in the final dataset, resulting in 3.84 million usable tiles (covering an area of 1.44 million km^2).

As is typical in machine learning, GPU memory limitations impose a cap on image size — 256×256 pixels per image in our case. To meet this requires a careful balance between terrain extent (area) and resolution (detail). Accordingly, we chose a terrain extent of roughly 5×5 km^2 since this is large enough to encompass significant features (such as mountains, river gorges, and canyons) while also exhibiting feature diversity and is thus suitable for many terrain applications. By concatenating source DEMs in an 8×8 grid and downsampling to the desired 256×256 pixel resolution we obtain a final extent of $\approx 4.9 \times 4.9$ km^2 and sampling resolution of $\sim$ 19.1m per pixel.

After concatenation and downsampling, we were left with approximately 200 000 terrain tiles. By applying combinations of vertical and horizontal flips, and 90-degree rotations, we were able to enlarge our dataset eightfold, for a total of 1 600 000 terrain images.

4.2. Signature Extraction

It is clearly infeasible to require hand-drawn feature sketches for hundreds of thousands of terrains. Instead, we concentrate on algorithmic derivation of feature sketches from heightmap sources, with a specific focus on ridge and drainage networks, cliff lines, and flat regions.

Drainage networks are detected by simulating water flow over a terrain and identifying regions with high water accumulation. Individual streams are labeled according to their Strahler order. This is a numerical measure of branching complexity [Hor45, Str57] that provides a way to rank the importance of rivers in a network. We then invert the terrain and apply the same algorithm to detect ridge lines. Cliff lines are extracted by applying Canny edge detection [Can86] on the normalized heightmap and then selecting lines that correspond to steep terrain (i.e., high average slope magnitude). Lastly, we mark regions in the terrain as flat if they have a low average slope magnitude.

The process of deriving a feature sketch is subject to several pre- and post-processing steps. First, as suggested by Guérin *et al.* [GDG*17], we apply a light Gaussian blur prior to feature extraction with the intention of making synthesis more robust to imprecise sketches. After extraction, we randomly select features to display weighted by their Strahler order and water accumulation value so as to emulate sketches containing varying extent and concentration of detail. Sketches are further enhanced by widening important ridge, drainage, and cliff lines in the scene (based on Strahler order or steepness). We also perform minor post-processing to fill in gaps, remove small artifacts, and smooth edges.

	Parameter	**Options**	**Selected**
Model	ResNet blocks	2, 3, 4	3
	Base dimension	32, 64, 128, 256	64
	Channel multipliers	[1, 2, 4, 4], [1, 2, 4, 8], [1, 2, 4, 4, 8, 8]	[1, 2, 4, 8]
	Dropout	0, 0.2	0
	Attention head dimension	8, 16, 64	8
	Noise scheduler	`DDPMScheduler`, `DDIMScheduler`, `PNDMScheduler`	`DDIMScheduler`
	Noise beta schedule	linear, cosine	linear
Training	Learning rate	1e-3, 1e-4, 5e-5, 1e-5	1e-4
	Learning rate scheduler	None, `ReduceLROnPlateau`	`ReduceLROnPlateau`
	Optimiser	Adam, AdamW	AdamW
	Batch size	1, 2, 4, 8, 16	8
	Loss function	L1, L2	L2

Table 1: *Hyperparameter tuning, including options considered (column* 2*) and finally selected (column* 3*).*

Finally, each sketch is stored as an RGBA image, with one feature class per channel.

The other component of the terrain signature is a style vector, which is based on histograms of slope and both normalized and un-normalized elevation. Each histogram consists of 16 bins, with appropriate bin ranges selected to enforce a uniform distribution over the training set. These bin ranges are used consistently across all subsequent training and inference. To compute a style vector for an individual terrain, we compute the 3 histograms according to these ranges and concatenate them to form a single 48-dimensional vector.

Training data for the upsampling models are generated in a self-supervised manner: downsampling the original terrain images by a factor of 8 using nearest neighbor interpolation. The models are then trained to predict terrains at the original resolution, given the down-scaled images as input.

5. Diffusion-based Synthesis

Diffusion models draw inspiration from non-equilibrium thermodynamics [SDWMG15], where the core idea is to systematically destroy the structure in a data distribution by iteratively adding noise to a sample and then learning to reverse this process. After training, new samples can be generated by passing pure noise through this learned denoising process.

We use a time-conditioned U-Net [RFB15, DN21], augmented with cross-attention and skip-connections, as the backbone of our architecture (denoted as f_θ in Figure 6). It processes an image, formed from the concatenation of noise and a feature sketch, by progressively lowering the resolution, passing it through a bottleneck, and then upscaling this representation back to its original resolution. As is standard practice, the U-Net is trained with a denoising objective to iteratively remove noise from an image [SHC*21].

With regard to model hyperparameters, we carefully examine the trade-off between sample quality and inference speed. For our *sketch-to-terrain* model $\mathcal{S}$, we focus on providing users with near-immediate feedback during authoring without significantly impacting the realism of the synthesized terrain. Early experimentation revealed that although smaller models lead to a decrease in fine details, they are generally capable of producing terrains with a consistent global structure. This is a reasonable compromise given the availability of amplification techniques to account for the lost detail [ZLB*19, ZCZ*20, ACA18, KSR20]. Details of the chosen hyperparameters and implementation of the method can be found in Table 1 and in the official repository, respectively.

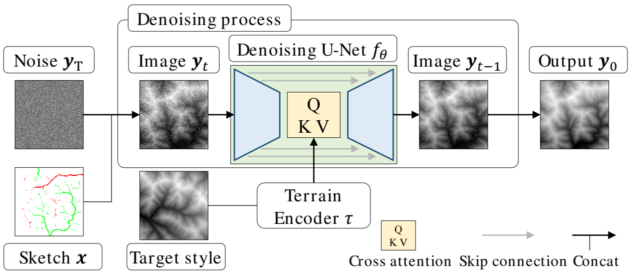

Figure 6: *Our diffusion model architecture supports stylization using a terrain encoder and cross-attention, and feature sketching through concatenation with the input noise image. The denoising process applies a U-Net to iteratively reduce noise and introduce structure.*

As illustrated in Figure 6, the feature sketch, x, is concatenated with the noisy image, y_t, at each timestep t. Aligning the sketch with the noisy terrain enables direct and localized manipulation of terrain features. In contrast, example-based stylization serves as a means of controlling the global aesthetic of the terrain, with synthesized terrains containing similar landform characteristics to those present in the provided examples. This is achieved by forming a single 48-element style vector based on histograms of slope and elevation (both normalized and un-normalized). This process (denoted as τ) ensures that deviations from the norm are accentuated in the generated styles. These style vectors are then passed to the intermediate layers of the U-Net, which utilize a cross-attention mechanism [VSP*17].

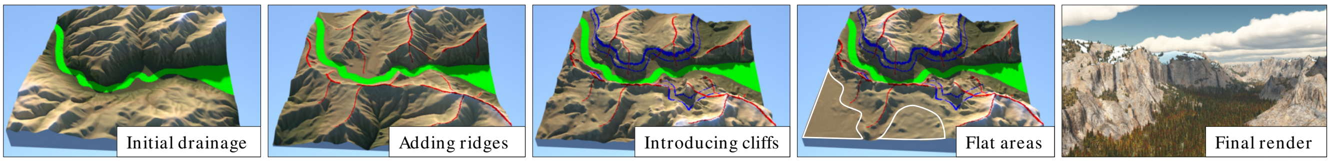

Figure 7: *As part of an authoring sequence an artist sketches drainage (green), ridges (red), cliffs (blue), and a flat area (outlined in white) to produce a landscape reminiscent of the Yosemite National Park.*

For consistency, we use the same architecture for upscaling as the *sketch-to-terrain* model, with the only exception being the type of input image (i.e., low-resolution terrain instead of a feature sketch). We also provide additional guidance through exemplar terrains and their corresponding style vectors, allowing the synthesized terrains to include characteristic features of specific landform classes (e.g., sharp ridge lines). Since upscaling is only performed as a post-processing step, users may select a larger number of sampling timesteps ($250-1000$) to achieve higher quality, without the need for interactive feedback.

5.1. Model and Training Hyperparameters

In terms of tuning, early tests demonstrated that although performance improves with larger models (base dimensions, channel multipliers, and ResNet blocks), this comes at the cost of substantially longer training and sampling times (or even out-of-memory errors). Given our focus on interactive authoring, it is important to select model parameters that balance quality and performance. Accordingly, we tested the parameter options listed in Table 1, by training each model for 72 hours on a machine with an Nvidia® A100 GPU (20GB). We split our dataset into training (90%), validation (5%), and testing (5%) subsets.

In terms of optimal parameters, in keeping with Saharia *et al.* [SHC*21, SCC*22], we did not observe any benefits from dropout, likely because regularisation is not needed due to the volume of source data.

We use a batch size of 8 as the largest value that does not result in out-of-memory errors. We also find that a learning rate of 1e-4 with a `ReduceLROnPlateau` scheduler and an L2 loss produces the best FID outcomes.

As observed by Chen *et al.* [Che23], the appropriate noise scheduler is task-specific, and depends on model size, image size and dataset statistics. We find that a linear (1e-6, 1e-2) `DDIMScheduler` [SME20] provides the best results in our case.

6. Results

With the exception of visualization software, our entire framework is coded in Python. The diffusion models are implemented using Hugging Face's `diffusers` library (with PyTorch backend) [vPPL*22] due to the memory optimizations and additional inference parameters it provides. Our authoring interface uses Blender's® Python API, enabling seamless integration with existing artist workflows. The hypsometric terrain views were rendered in Mitsuba 0.6, while photorealistic landscape images were rendered with *Vue*®.

Training was conducted on a workstation equipped with a partitioned Nvidia® A100 GPU (20GB), while evaluation and testing were performed on a desktop computer equipped with an Intel® Core™ i9-10980XE CPU @ 3.00GHz and NVidia® RTX 3090 (24GB).

6.1. Hydrological Validation

Hydrological consistency is an important criterion for determining not only the geological but also the perceptual realism of a synthetic terrain [RKv*22]. One form of evaluation is to compute and visualize the drainage of a terrain. The upstream drainage area a of point $\mathbf{p}$ is the amount of water that flows through $\mathbf{p}$, and is proportional to the area of the surface where every downstream route passes through $\mathbf{p}$. Figure 8 provides a visual comparison of the drainage of our method against reference procedural, image-based synthesis and erosion simulations, obtained from the online repository of Galin *et al.* [GGP*19].

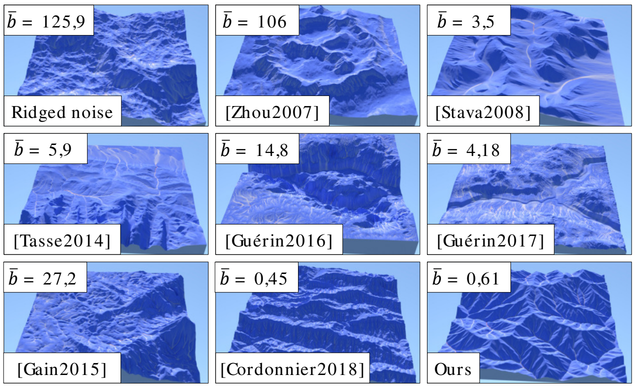

Figure 8: *A comparison of drainage areas for different synthetic terrains: our method produces fewer endorheic pits and greater hydrological consistency (as measured by average breaching volume $\bar{b}$) than previous data-driven approaches. Instances of procedural (ridged noise) and simulation [CCB*18] outputs are included for a basis of comparison. A lower breaching volume, $\bar{b}$, is better.*

As an additional measure, we compute the average breaching volume $\bar{b}$ for each case. This is defined as the volume of material removed by the minimal breaching required to ensure free drainage off the terrain [BLM14] divided by terrain area. Values reported in Figure 8 demonstrate that our method outperforms comparator techniques and is on par with erosion simulation. A notable exception is a method described by Scott and Dodgson [SD21], which specifically performs a complete post-processing multi-resolution breaching to improve the realism of image-based synthesis methods: in that case, the resulting average breaching volume is 0.

6.2. Balancing Quality and Performance

As illustrated in Figures 15 and 16, our *sketch-to-terrain* model is capable of generating high-fidelity terrains faithful to user inputs in the form of feature sketches and style selections. This is complemented by *upscaling* models that introduce high-frequency detail that is cognizant of the low-frequency source (see Figure 9). We also see significant improvement over the current state-of-the-art when generating terrain from sparse input sketches. In these cases, our results show plausible and consistent structures and are free of the artifacts typical of other methods (see Figures 16 and 17).

Next, we consider the necessary trade-off between response rate and output quality. One strength of diffusion models is that they rely on an iterative generation process involving a sequence of sampling timesteps during which output quality improves at a diminishing rate [DN21]. Since the time required for an individual timestep remains relatively constant (at around 40Hz in our case) this allows a time limit to be imposed on synthesis, so long as the user is willing to accept the consequent quality level. Conversely, a quality threshold can be set with the generation time determined during execution.

To explore this dynamic, we compared terrain quality as a function of the number of timesteps using the following standard image distance metrics: Learned Perceptual Image Patch Similarity (LPIPS) [ZIE*18], and Fréchet Inception Distance (FID) [HRU*17]. Despite a clear decline in quality when using fewer timesteps (as shown in Figure 10, particularly at less than 25 timesteps), we, nevertheless, find that the global structure of the terrain is well-maintained (see Figure 12).

Furthermore, we compared our results to the current state-of-the-art deep learning model for *sketch-to-terrain* synthesis [GDG*17] on the basis of LPIPS and FID scores (see Figure 10). Their implementation uses conditional Generative Adversarial Networks (cGANs) for image-to-image translation, based on the Pix2Pix [IZZE17] architecture. As evidenced by the significantly lower FID scores for $t \geq 15$, our model more closely matches the underlying distribution of real terrains. Similarly, our model produces better LPIPS scores for $t \geq 25$, which is indicative of a perceptual image similarity closer to the target terrains.

Our framework supports two modes of fixed-duration operation: real-time response during active sketching (5 timesteps in 0.125s at a rate of 8Hz), and interactive updates once the pen is lifted (up to 250 timesteps in 6.25s). In fact, diffusion iterations can continue in the background during unrelated activities, such as viewing, tool selection, and parameter setting so that the terrain is progressively refined.

6.3. Perceptual Study

We tested the perceived realism of terrains obtained from our framework, the real ground truth data and a cGAN model [GDG*17] with an internet-administered user study ($n = 41$ participants). The comparator cGAN model was trained on exactly the same dataset as the diffusion model using the code from the original paper [GDG*17].

The experiment design employed a 2AFC (two-alternative forced choice) protocol in which participants were required to choose on each trial between two landscapes drawn from real, diffusion-synthesized and CGAN-synthesized sets. Their selection was based on the question: "Which terrain looks more realistic (left or right)?", with the order of presentation (left or right) randomized. The number of pairings of treatments (Real vs. cGAN, Real vs. Ours, and Ours vs. cGAN) was also balanced. The landscapes were fixed at a 256×256 resolution and rendered with a hypsometric texture from an oblique angle and rotated to left and right about the vertical axis over a period of 10 seconds to aid depth discrimination.

To address the known differences in perception between landscape styles and their constituent landforms [SD22, RKv*22] we partitioned the trials into four distinct categories: cliffs, hills, mountains, and flatland (which nevertheless contained some detail, such as coastlines and river courses). In order to avoid selection bias, we first performed k-means clustering on the style vector of the input dataset to group terrains by category (with $k = 4$). Then for each treatment and category, we randomly selected a set of 10 signatures based on the style distance from the centroid of the cluster, with 5 above and 5 below the median distance to provide diversity. The original terrain was retained for the real treatment, while the corresponding signature was used to generate terrain in the case of the diffusion and cGAN models. We avoided performing resolution amplification so as to ensure parity between real and generated data. The extent of terrains was kept constant at $5 \times 5\,\text{km}^2$ across all trials.

Each participant was presented with 26 trials (3 treatment parings with 2 repetitions for each of the 4 landscape categories and 2 additional repeated control pairs). The duplicate controls were chosen at random, with a participant's data excluded due to inconsistency if both controls differed from their original selections. We introduced further randomization by drawing each category-treatment sample without repetition from a pool of 10, meaning that the experiment dataset contained 120 terrains (10 pool elements ×3 treatments ×4 categories).

A demographic questionnaire was completed by each participant beforehand to determine familiarity with real landscapes (regularity of experiencing natural landscapes) and gaming (regularity of playing games depicting outdoor environments).

We obtained participants via convenience sampling with the majority being students and academics. After exclusion due to inconsistency, we analyzed data from $n = 41$ participants of whom 37% hiked and 54% gamed either weekly or monthly (see Table 2). We then undertook a Bayesian statistical analysis partitioned by category with the probability of treatment t (real, ours, or cGAN) modeled as a Bernoulli random variable with associated probability θ_t.

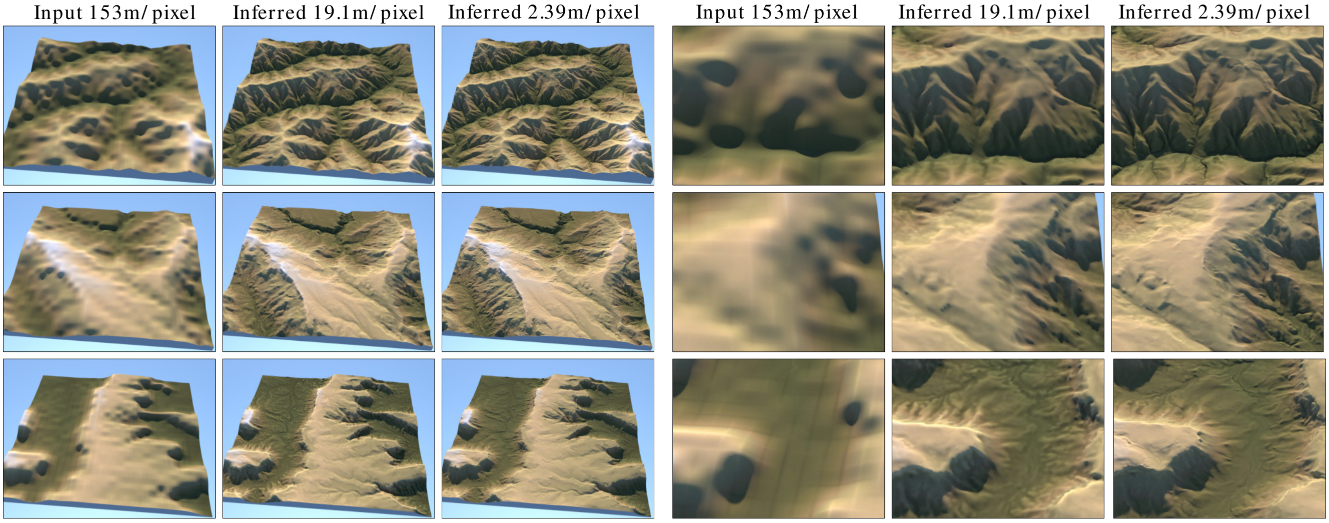

Figure 9: *Cascaded application of our upscaling models $\mathcal{U}_1$ (153 → 19.1 m/pixel) and $\mathcal{U}_2$ (19.1 → 2.39 m/pixel), allowing for* 64× *super-resolution.*

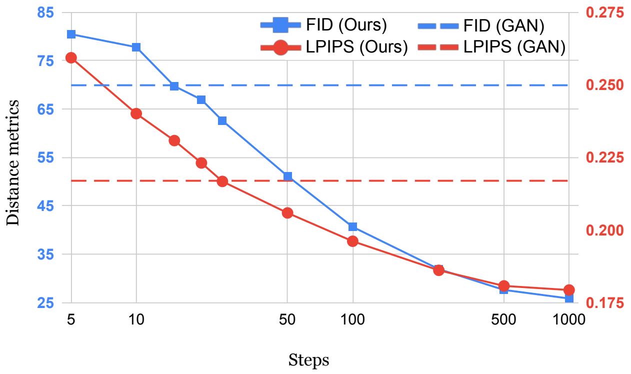

Figure 10: *A graph of FID and LPIPS quality error metrics as a function of the number of timesteps. We indicate the current state-of-the-art deep-learning approach with dashed horizontal lines.*

For statistical significance, we derived the credible intervals corresponding to 95% of each probability density function (see Figure 14), implying that if the intervals for two treatments in a category do not intersect then they credibly represent distinct distributions. For the purposes of inspection, Figure 11 shows the terrains most and least favored by participants grouped by treatment and category.

Our analysis indicates that users are able to reliably distinguish real from generated terrains in the case of hills, but not plains, cliffs, and mountains (see Figure 14). In fact, our method outperforms cGANs for plains and hills and, surprisingly, even real terrains in the case of plains. Our tentative conclusion is that these differences are primarily due to viewers favouring terrains with well-defined erosion detail (as borne out by Figures 11 and 13). Such detail is visible because we use a fractional Laplacian to enhance high frequencies in our rendering. The cGAN approach lacks any discernible notion of style and this impacts its performance in certain categories. It also has a tendency to produce small grid artifacts [GDG*17]. While this clearly harms the realism of terrain upon close inspection, such artifacts can pass for plausible detail at medium viewing distances or when viewed for a short amount of time. Finally, it is worth noting that these categorical differences support the view [SD22] that perceptual experiments really should treat terrain types separately.

6.4. Limitations

Based on the success of sketching and stylization, we attempted to extend the authoring toolset to include elevation constraints: with the idea that users would be able to specify the elevation of key features, such as mountain peaks and river junctions. Unfortunately, we found it impossible to achieve alignment with the base terrain, while balancing control and ease-of-use. We also attempted to incorporate *elevation-to-satellite* translation, as a means of texture synthesis. Although the generated textures exhibited well-defined feature lines and shadows, present in the satellite imagery, they were particularly noisy and lacked a coherent global structure.

Diffusion models have shown marked success in text-to-image generation tasks [RDN*22, SCS*22]. This prompted us to experiment with terrain synthesis conditioned on free-form textual descriptions. Unfortunately, the results were deeply disappointing: our crowdsourced labeling resulted in generic landform descriptions and vaguely specified placement leading to poor quality inference. Our conclusion is that non-expert language is insufficiently precise in this context, especially when compared to a combination of stylization and sketching.

Finally, the performance of our framework approaches but does not cross the threshold of real-time response (20Hz) and indistin-

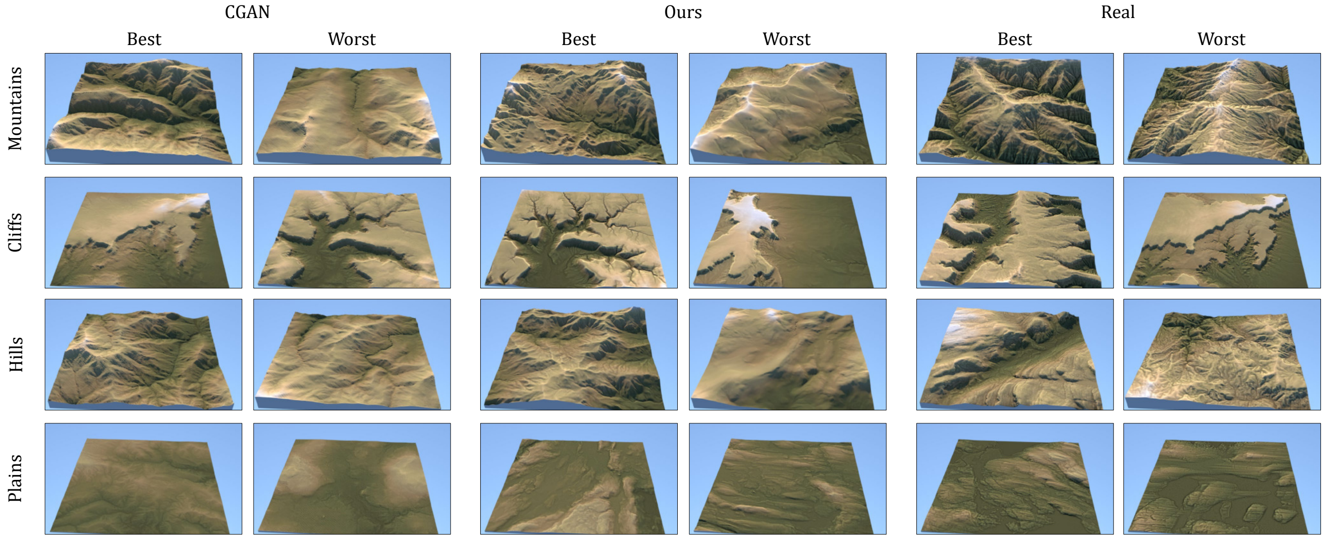

Figure 11: *Terrains selected as most and least realistic in our perceptual study, by category (mountains, cliffs, hills, and plains) and method (CGAN, ours, and real).*

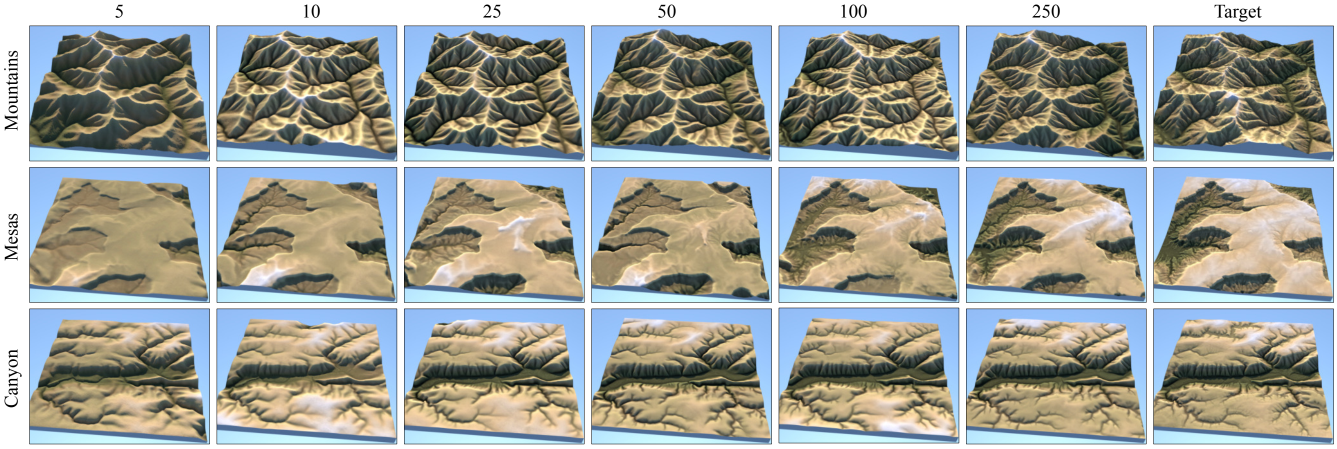

Figure 12: *Sketch-to-terrain generation at different sampling timesteps.*

guishability from real terrains across all landform classes. Another issue, common in the use of diffusion models for iterative authoring, is flickering and inconsistent outputs after minor adjustments to the canvas. Although increasing the detail of the input sketch and fixing the random seed during authoring may improve consistency, further research and experimentation is required. This may include, for example, conditioning on previous frames to provide a more coherent authoring experience. It is also worth noting that this is a highly active area of research and the inevitable improvements to come can be carried over to our system.

7. Conclusion

We have shown that diffusion models form a sound basis for authoring terrains, with an appropriate balance of interactive response and both perceptual and structural realism. Our diffusion-based authoring framework enables artists to select terrain styles from a palette and also sketch features that include cliffs, ridges, drainage, and flat areas. This provides expressivity in terms of the achievable terrain diversity, economy in the extent to which terse input leads to rich output, and precision in that the combination of a style and detailed feature sketch effectively represents an accurate terrain signature.

Nevertheless, there is an opportunity to take diffusion-based terrains further in future work. We would like to extend authoring to encompass accurately registered surface detail derived from satellite imagery and elevation controls at feature points, such as river junctions and mountain peaks. In this regard, recent advances in latent diffusion [RBL*22] offer a promising avenue.

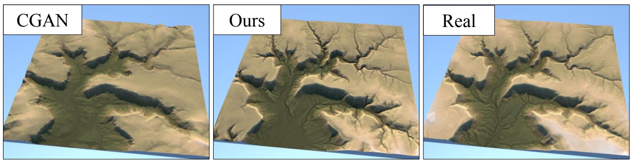

Figure 13: *A side-by-side comparison of cGAN, ours, and real terrain based on the same canyon signature. All three of these terrains appeared by chance in the perceptual experiment due to a lack of diversity in the cliff category. Within its category the cGAN canyon scored worst, ours performed best, and the real was intermediate (see Figure 11).*

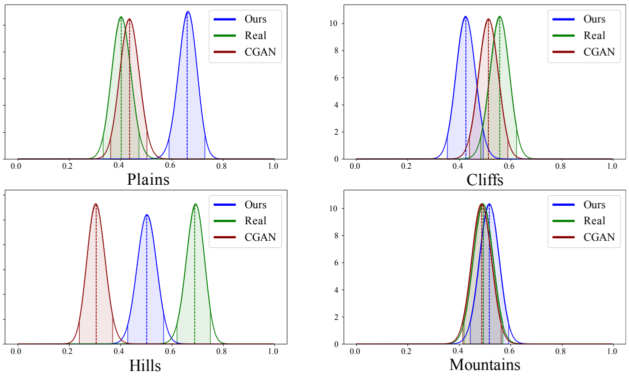

Figure 14: *Posterior beta distributions for θ_{ours}, θ_{CGAN} and θ_{real} from our perceptual study, grouped by terrain category. Dashed lines indicate the mean with a higher score meaning that the treatment in question is selected more frequently. The shaded portions show the 95% credible interval, with no overlap implying a statistically-significant difference in distributions.*

8. Acknowledgments

We are grateful to Oscar Argudo for providing advice and code for Bayesian analysis and Daniel Lochner for his input on the use of text labelling for terrain generation. This work is funded by the National Research Foundation of South Africa (Grant Number: 129257) and the project AMPLI ANR-20-CE23-0001, supported by Agence Nationale de la Recherche. Computations were performed using facilities provided by the University of Cape Town ICTS High-Performance Computing team.

References

[AAC*17] ARGUDO O., ANDUJAR C., CHICA A., GUÉRIN E., DIGNE J., PEYTAVIE A., GALIN E.: Coherent multi-layer landscape synthesis. *The Visual Computer 33*, 6 (2017), 1005–1015. 2

[ACA18] ARGUDO O., CHICA A., ANDUJAR C.: Terrain super-resolution through aerial imagery and fully convolutional networks. *Computer Graphics Forum 37*, 2 (2018), 101–110. 2, 5

[BFM06] BARLOW J., FRANKLIN S., MARTIN Y.: High spatial resolution satellite imagery, DEM derivatives, and image segmentation for the detection of mass wasting processes. *Photogrammetric Engineering & Remote Sensing 72*, 6 (2006), 687–692. 12

[BLM14] BARNES R., LEHMAN C., MULLA D.: Priority-flood: An optimal depression-filling and watershed-labeling algorithm for digital elevation models. *Computers & Geosciences 62* (2014), 117–127. 7

[Can86] CANNY J.: A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, 6 (1986), 679–698. 4

[CCB*18] CORDONNIER G., CANI M.-P., BENES B., BRAUN J., GALIN E.: Sculpting mountains: Interactive terrain modeling based on subsurface geology. *IEEE Transactions on Visualization and Computer Graphics 24*, 5 (2018), 1756–1769. 6

[Che23] CHEN T.: On the importance of noise scheduling for diffusion models. *arXiv preprint arXiv:2301.10972* (2023). 6

[DN21] DHARIWAL P., NICHOL A.: Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems 34* (2021), 8780–8794. 2, 5, 7

[GAMA20] GUÉRIN E., AYDIN O., MAHDAVI-AMIRI A.: *Manual of Digital Earth*. Springer Singapore, 2020, ch. Artificial Intelligence, pp. 357–385. 2

[GDG*17] GUÉRIN É., DIGNE J., GALIN E., PEYTAVIE A., WOLF C., BENES B., MARTINEZ B.: Interactive example-based terrain authoring with conditional generative adversarial networks. *Acm Transactions on Graphics 36*, 6 (2017), 1–13. 2, 4, 7, 8

[GDGP16] GUÉRIN E., DIGNE J., GALIN E., PEYTAVIE A.: Sparse representation of terrains for procedural modeling. In *Computer Graphics Forum* (2016), vol. 35, Wiley Online Library, pp. 177–187. 2

[GGP*19] GALIN E., GUÉRIN E., PEYTAVIE A., CORDONNIER G., CANI M.-P., BENES B., GAIN J.: A review of digital terrain modeling. *Computer Graphics Forum 38*, 2 (2019), 553–577. 2, 3, 6, 12, 13

[GMM15] GAIN J., MERRY B., MARAIS P.: Parallel, realistic and controllable terrain synthesis. In *Computer Graphics Forum* (2015), vol. 34, Wiley Online Library, pp. 105–116. 2

[HJA20] HO J., JAIN A., ABBEEL P.: Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems 33* (2020), 6840–6851. 2, 13

[Hor45] HORTON R. E.: Erosional development of streams and their drainage basins; hydrophysical approach to quantitative morphology. *Geological society of America bulletin 56*, 3 (1945), 275–370. 4

[HRU*17] HEUSEL M., RAMSAUER H., UNTERTHINER T., NESSLER B., HOCHREITER S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems 30* (2017). 7

[HSC*22] HO J., SAHARIA C., CHAN W., FLEET D. J., NOROUZI M., SALIMANS T.: Cascaded diffusion models for high fidelity image generation. *J. Mach. Learn. Res. 23* (2022), 47–1. 3

[HWL*22] HU G., WANG C., LI S., DAI W., XIONG L., TANG G., STROBL J.: Using vertices of a triangular irregular network to calculate slope and aspect. *International Journal of Geographical Information Science 36*, 2 (2022), 382–404. 13

[IZZE17] ISOLA P., ZHU J.-Y., ZHOU T., EFROS A. A.: Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2017), pp. 1125–1134. 2, 7

[KSR20] KUBADE A. A., SHARMA A., RAJAN K. S.: Feedback neural network based super-resolution of DEM for generating high fidelity features. In *IGARSS 2020 - 2020 IEEE International Geoscience and Remote Sensing Symposium* (2020), pp. 1671–1674. 2, 5

[LLXT22] LI S., LI K., XIONG L., TANG G.: Generating terrain data for geomorphological analysis by integrating topographical features and conditional generative adversarial networks. *Remote Sensing 14*, 5 (2022), 1166. 2

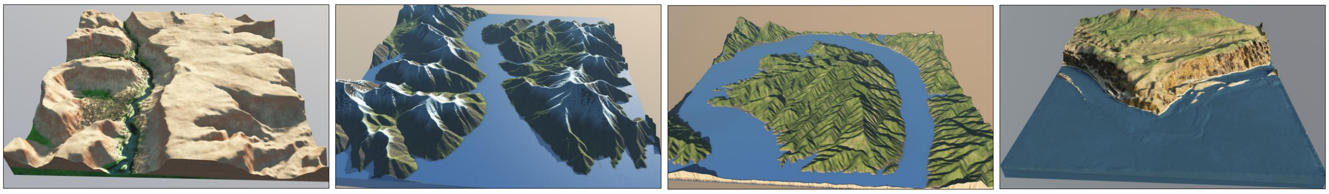

Figure 15: *Examples of terrains authored by non-expert users in under a minute. From left to right: (1) a canyon with a narrow river flowing through it, (2) a snow-capped mountain range with a fjord, (3) a large island and surrounding lake, and (4) a cliff-dominated coastline with rocks protruding from the sea.*

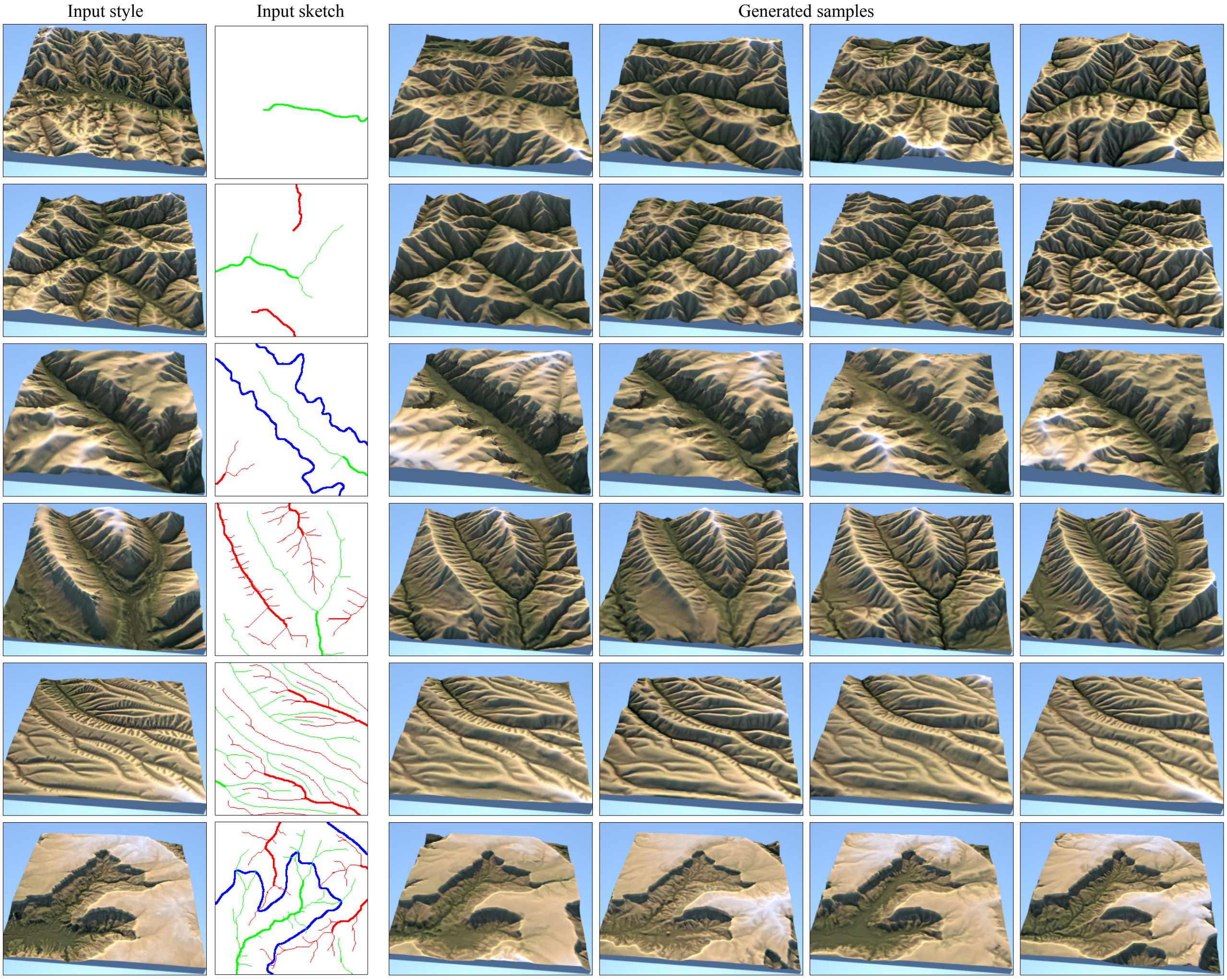

Figure 16: *Sketch-to-terrain sample diversity, showing progressively less diversity as sketches become more detailed.*

[Luo22] Luo C.: Understanding diffusion models: A unified perspective. *arXiv preprint arXiv:2208.11970* (2022). 2

[LWZ*06] Li Q., Wang G., Zhou F., Tang X., Yang K.: Example-based realistic terrain generation. In *International Conference on Artificial Reality and Telexistence* (2006), Springer, pp. 811–818. 2

[NJSR22] Naik S., Jain A., Sharma A., Rajan K.: Deep generative framework for interactive 3d terrain authoring and manipulation. *arXiv preprint arXiv:2201.02369* (2022). 2

[Paw19] Pawluszek K.: Landslide features identification and morphology investigation using high-resolution DEM derivatives. *Natural Hazards 96*, 1 (2019), 311–330. 12

[RBL*22] Rombach R., Blattmann A., Lorenz D., Esser P., Ommer B.: High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision*

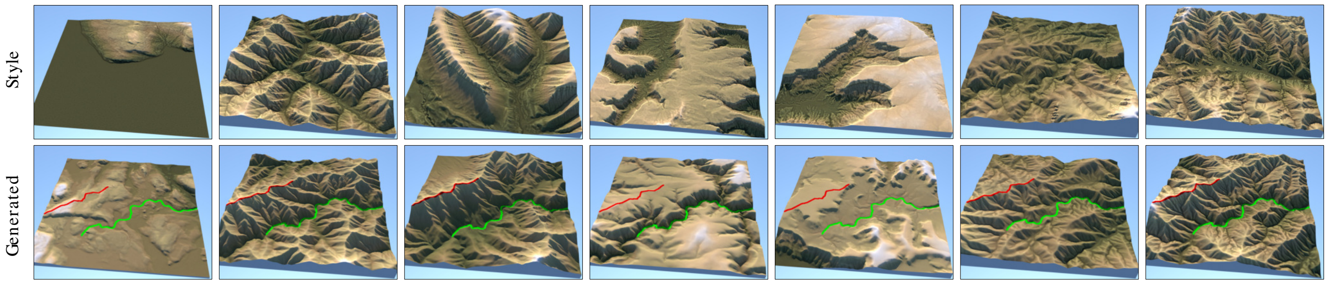

Figure 17: *A single sketch can lead to varied generated terrains by using the style conditioning.*

and Pattern Recognition (2022), pp. 10684–10695. 9

[RDN*22] Ramesh A., Dhariwal P., Nichol A., Chu C., Chen M.: Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125* (2022). 8

[RFB15] Ronneberger O., Fischer P., Brox T.: U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention* (2015), Springer, pp. 234–241. 5

[RKv*22] Rajasekaran S. D., Kang H., Čadík M., Galin E., Guérin E., Peytavie A., Slavík P., Benes B.: Ptrm: Perceived terrain realism metric. *ACM Transactions on Applied Perception 19*, 2 (jul 2022). 6, 7

[SCC*22] Saharia C., Chan W., Chang H., Lee C., Ho J., Salimans T., Fleet D., Norouzi M.: Palette: Image-to-image diffusion models. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Proceedings* (2022), pp. 1–10. 6

[SCS*22] Saharia C., Chan W., Saxena S., Li L., Whang J., Denton E., Ghasemipour S. K. S., Ayan B. K., Mahdavi S. S., Lopes R. G., et al.: Photorealistic text-to-image diffusion models with deep language understanding. *arXiv preprint arXiv:2205.11487* (2022). 8

[SD21] Scott J. J., Dodgson N. A.: Example-based terrain synthesis with pit removal. *Computers & Graphics 99* (2021), 43–53. 2, 7

[SD22] Scott J. J., Dodgson N. A.: Evaluating realism in example-based terrain synthesis. *ACM Transactions on Applied Perceptions* (2022). 2, 7, 8

[SDWMG15] Sohl-Dickstein J., Weiss E., Maheswaranathan N., Ganguli S.: Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning* (2015), PMLR, pp. 2256–2265. 5

[SHC*21] Saharia C., Ho J., Chan W., Salimans T., Fleet D. J., Norouzi M.: Image super-resolution via iterative refinement. *arXiv preprint arXiv:2104.07636* (2021). 5, 6

[SME20] Song J., Meng C., Ermon S.: Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502* (2020). 6

[Str57] Strahler A. N.: Quantitative analysis of watershed geomorphology. *Eos, Transactions American Geophysical Union 38*, 6 (1957), 913–920. 4

[TGM12] Tasse F. P., Gain J., Marais P.: Enhanced texture-based terrain synthesis on graphics hardware. *Computer Graphics Forum 31*, 6 (2012), 1959–1972. 2

[TL21] Tan M., Le Q.: Efficientnetv2: Smaller models and faster training. In *International Conference on Machine Learning* (2021), PMLR, pp. 10096–10106. 4

[vPPL*22] von Platen P., Patil S., Lozhkov A., Cuenca P., Lambert N., Rasul K., Davaadorj M., Wolf T.: Diffusers: State-of-the-art diffusion models. https://github.com/huggingface/diffusers, 2022. 6

[VRGZS20] Valencia-Rosado L. O., Guzman-Zavaleta Z. J., Starostenko O.: Generation of synthetic elevation models and realistic surface images of river deltas and coastal terrains using cgans. *IEEE Access 9* (2020), 2975–2985. 2

[VSP*17] Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez A. N., Kaiser Ł., Polosukhin I.: Attention is all you need. *Advances in neural information processing systems 30* (2017). 5

[YZS*22] Yang L., Zhang Z., Song Y., Hong S., Xu R., Zhao Y., Shao Y., Zhang W., Cui B., Yang M.-H.: Diffusion models: A comprehensive survey of methods and applications, 2022. 2

[ZCZ*20] Zhu D., Cheng X., Zhang F., Yao X., Gao Y., Liu Y.: Spatial interpolation using conditional generative adversarial neural networks. *International Journal of Geographical Information Science 34*, 4 (2020), 735–758. 2, 5

[ZIE*18] Zhang R., Isola P., Efros A. A., Shechtman E., Wang O.: The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2018), pp. 586–595. 7

[ZLB*19] Zhao Y., Liu H., Borovikov I., Beirami A., Sanjabi M., Zaman K.: Multi-theme generative adversarial terrain amplification. *ACM Transactions on Graphics 38*, 6 (2019). 2, 5

[ZLZ*22] Zhang J., Li C., Zhou P., Wang C., He G., Qin H.: Authoring multi-style terrain with global-to-local control. *Graphical Models 119* (2022), 101122. 2

[ZSTR07] Zhou H., Sun J., Turk G., Rehg J. M.: Terrain synthesis from digital elevation models. *IEEE Transactions on Visualization and Computer Graphics 13*, 4 (2007), 834–848. 2

Appendix A: Terrain Representation

A terrain's elevation can be defined by a continuous function $h : \mathbb{R}^2 \to \mathbb{R}$, where the height/altitude of the terrain at a point (x,y) is given by $h(x,y)$. The primary limitation of such a definition is its inability to represent terrains that contain overhangs, arches, or caves, since only one height value can be associated with each point in the function's domain [GGP*19]. However, due to its simplicity, terrain is most often represented in this manner.

A Digital Elevation Model (DEM) is a 3D representation of a terrain's surface, created from elevation data. DEMs are most commonly defined as heightmaps, where altitudes are arranged on a regular 2D grid. As a result, they can be stored as grayscale images, where the pixel value represents the altitude at a point.

While heightmaps remain the predominant form of terrain surface representation, in many cases, such as landslide detection algorithms [BFM06, Paw19], it is advantageous to store terrain slope as opposed to raw height values. Mathematically, a terrain's slope is defined as a two-dimensional vector field, whose components are the first-order partial derivatives of the elevation function, h, given by $\mathbf{g} = \nabla h = (\partial h/\partial x, \partial h/\partial y)$. These gradient/derivative images can also be stored as images with two channels being used (one for each component of the vector field), where the pixel value represents the angle of the slope at a point.

In practice, elevation images are stored at a bit-depth of 16 (values between 0 and 65535), while derivative images use 8 bits for each channel (values between 0 and 255). These approaches to structuring height data play an important role in transferring geographical knowledge to computer-processable information [HWL*22] and have found many uses in texture synthesis and machine learning methods [GGP*19].

Appendix B: Diffusion Models

The forward diffusion process, q, is modelled as a Markov chain, where Gaussian noise is added to a data point $y_0 \equiv y$ over T timesteps:

$$q(y_{t+1} \mid y_t) = \mathcal{N}(y_{t-1}; \sqrt{\alpha_t} y_{t-1}, (1-\alpha_t)I) \quad (1)$$

$$q(y_{1:T} \mid y_0) = \prod_{t=1}^{T} q(y_t \mid y_{t-1}) \quad (2)$$

The hyper-parameters α_t determine the variance of the noise added at timestep t and are chosen such that by timestep $t = T$, y_T is virtually indistinguishable from Gaussian noise.

A useful property of this process is that we can sample y_t at an arbitrary time step t in a closed form by reparameterising the forward process. Let $\gamma_t = \prod_{i=1}^{t} \alpha_i$, then

$$q(y_t \mid y_0) = \mathcal{N}(y_t; \sqrt{\gamma_t} y_0, (1-\gamma_t)I) \quad (3)$$

The posterior distribution of y_{t-1} given (y_0, y_t) can be derived as

$$q(y_{t-1} \mid y_0, y_t) = \mathcal{N}(y_{t-1} \mid \mu, \sigma^2 I) \quad (4)$$

where $\mu = \frac{\sqrt{\gamma_{t-1}}(1-\alpha_t)}{1-\gamma_t} y_0 + \frac{\sqrt{\alpha_t}(1-\gamma_{t-1})}{1-\gamma_t} y_t$ and $\sigma^2 = \frac{(1-\gamma_{t-1})(1-\alpha_t)}{1-\gamma_t}$.

Learning

The goal of the reverse process, p, is to recover the target image y_0 from a noisy image $\widetilde{y}$,

$$\widetilde{y} = \sqrt{\gamma} y_0 + \sqrt{1-\gamma}\varepsilon, \varepsilon \sim \mathcal{N}(0, I) \quad (5)$$

To this end, the neural network $f_\theta(x, \widetilde{y}, \gamma)$ is parameterised to condition on an input image x (e.g., a user sketch), a noisy image $\widetilde{y}$, and the current noise level γ. The network is then trained to predict the noise vector ε by optimising the objective function:

$$\mathbb{E}_{(x,y)} \mathbb{E}_{\varepsilon,\gamma} \left\| f_\theta(x, \underbrace{\sqrt{\gamma} y_0 + \sqrt{1-\gamma}\varepsilon}_{\widetilde{y}}, \gamma) - \varepsilon \right\|_p^p \quad (6)$$

Algorithm 1 Training a denoising model f_θ

```
repeat
    $(x, y_0) \sim p(x, y)$                ▷ Sample conditional and target images
    $\gamma \sim p(\gamma)$                      ▷ Select amount of noise to add
    $\varepsilon \sim \mathcal{N}(0, I)$                     ▷ Sample Gaussian noise
    Take a gradient descent step on
        $\nabla_\theta \| f_\theta(x, \sqrt{\gamma} y_0 + \sqrt{1-\gamma}\varepsilon, \gamma) - \varepsilon \|_p^p$
until converged
```

Inference

Since f_θ is trained to estimate ε given a noisy image $\widetilde{y}$ and y_t, y_0 is approximated by rearranging the terms in equation 5 to get

$$\hat{y}_0 = \frac{1}{\sqrt{\gamma_t}} \left(y_t - \sqrt{1-\gamma_t} f_\theta(x, y_t, \gamma_t) \right) \quad (7)$$

This estimate, $\hat{y}_0$, is then substituted into the posterior distribution of $q(y_{t-1} \mid y_0, y_t)$ in equation 4 to parameterise the mean of $p_\theta(y_{t-1} \mid y_t, x)$ as

$$\mu_\theta(x, y_t, \gamma_t) = \frac{1}{\sqrt{\alpha_t}} \left(y_t - \frac{1-\alpha_t}{\sqrt{1-\gamma_t}} f_\theta(x, y_t, \gamma_t) \right) \quad (8)$$

The variance of $p_\theta(y_{t-1} \mid y_t, x)$ is set to $(1-\alpha_t)$, a default given by the variance of the forward process [HJA20].

Finally, with this parameterisation, each iteration of the reverse process can be computed as

$$y_{t-1} \leftarrow \frac{1}{\sqrt{\alpha_t}} \left(y_t - \frac{1-\alpha_t}{\sqrt{1-\gamma_t}} f_\theta(x, y_t, \gamma_t) \right) + \sqrt{1-\alpha_t}\varepsilon_t \quad (9)$$

where $\varepsilon_t \sim \mathcal{N}(0, I)$.

Algorithm 2 Inference in T timesteps

```
$y_T \sim \mathcal{N}(0, I)$                         ▷ Start with pure Gaussian noise
for $t = T, \dots, 1$ do                      ▷ Perform iterative refinement
    $z \sim \mathcal{N}(0, I)$ if $t > 1$, else $z = 0$
    $y_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left( y_t - \frac{1-\alpha_t}{\sqrt{1-\gamma_t}} f_\theta(x, y_t, \gamma_t) \right) + \sqrt{1-\alpha_t} z$
end for
return $y_0$                                  ▷ The denoised image
```

Appendix C: User study demographics

		Gaming				
		A	**B**	**C**	**D**	Total
Hiking	**A**	3	1	1	4	9
	B	2	7	1	7	17
	C	2	1	3	5	11
	D	0	3	0	1	4
	Total	7	12	5	17	41

Table 2: *Demographics in our perceptual study pertaining to regularity of experiencing nature (hiking) and playing games in outdoor environments (gaming). Possible responses:* ***A)*** never or rarely *(less than once a year),* ***B)*** occasionally *(a few times a year),* ***C)*** often *(every month), or* ***D)*** usually *(every week or almost).*