Evaluating Texture Synthesis and Diffusion Models for Global Terrain Generation

Research Proposal

Christopher Langebrink
University of Cape Town
South Africa
lngchr014@myuct.ac.za

Sam Frost
University of Cape Town
South Africa
frssam005@myuct.ac.za

ABSTRACT

Synthetic terrain generation is used in many applications like film and game design. Terrain authoring is an important aspect of terrain generation as it gives the artist control over the landscape. Terrain generation has been generalised into procedural, simulation and example-based methods. Example-based methods using diffusion models and texture synthesis have shown promise in their effectiveness in producing diverse and realistic landscapes while allowing for interactive user control. This paper proposes using diffusion-based models and texture synthesis to address the issue of visualising and generating digital terrain on a global scale. This paper further presents the methods and procedures for implementing the models.

CCS CONCEPTS

• **Computing methodologies → Shape modeling; Machine Learning**; *Image manipulation, Texturing, Image-based rendering*; • **Applied computing**;

KEYWORDS

Terrain authoring, terrain generation, machine learning, texture synthesis

1 INTRODUCTION

Terrain plays a fundamental role in many multimedia applications such as open-world computer games which require large-scale natural environments. The environments need to be synthesised by a model to create a landscape with high diversity while maintaining the realism of nature. Terrain formation is a complex interplay of gradual erosion and sudden events like landslides, shaping landscapes from featureless plains to rugged mountain ranges. Despite decades of study, terrain generation still faces numerous unresolved challenges due to the diverse scales and processes in shaping Earth's surface [15]. Terrain authoring is difficult due to the complex underlying geomorphological factors that interact and shape the landscape. Terrain authoring tools try balancing precision and efficiency enabling real-time placement of landscape features with high landform diversity and realism [23].

Traditionally procedural, simulation and example-based methods have tried to address this problem. Procedural approaches in terrain generation are phenomenological, aiming to replicate natural phenomena using terrain properties such as fractal characteristics and constructing landscapes based on observations rather than real data inputs [15]. Procedural methods often use fractional Brownian motion [12, 25], subdivision schemes [10], and faulting [11] to reproduce natural phenomena. However, these methods do not provide effective user control and cannot capture the high-level structure of real terrains with a variety of landforms [15]. Simulation techniques emerge from the process of simulating erosion over the landscape. The simulation can come in the form of height-field thermal erosion [34], the movement of tectonic plates [26], Eulerian approaches [2], Lagrangian approaches [20], and Aeolian erosion [3]. However, these simulation-based methods suffer from scalability and do not allow for interactive user authoring as they are computationally expensive [15].

Example-based approaches to terrain generation attempt to solve this problem by using vast amounts of real-world data samples to synthesize terrain. Digital Elevation Models (DEMs), commonly utilized as exemplar terrains, encapsulate the geological history, leading to realistic outcomes in terrain synthesis[16]. A promising avenue to terrain generation is the example-based methods which use diffusion models and texture synthesis. These methods provide performance, effective user control, and realism. However, it is important to consider that the models are limited by the sampling resolution of the input data and any artefacts could cause undesirable results [16]. This paper aims to determine the possibility of using diffusion models and texture synthesis for global terrain generation.

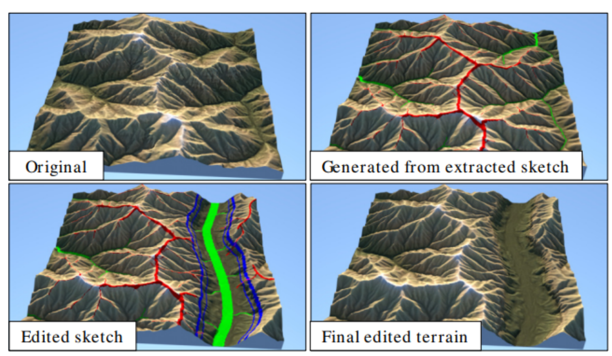

Figure 1: Gain *et al.* [23] showing the use of diffusion models for terrain authoring.

2 RELATED WORK

2.1 Procedural Generation

Procedural generation replicates the effects of geomorphological processes through algorithms and parametric specification. It sculpts terrain using noise, faulting [12], and subdivision schemes [10]. Procedural generation techniques differ from the other two categories of terrain generation procedures as they do not simulate actual geomorphological processes or use real-terrain data as input. Tripkovic [38] recently developed a procedural generation tool to generate planets using a mesh sphere. Hnaidi et al. [17] used a multi-grid diffusion of constraints for the procedural generation of terrain to develop an efficient method for controlling landforms. Procedural generation offers an efficient method for terrain generation, however, the realism provided is unsatisfactory as it fails to adequately illustrate the variety of land formations found in real terrain.

2.2 Simulation-based Techniques

Simulation-based techniques mimic actual geomorphological processes primarily in the form of erosion phenomena [15]. Prominent erosion phenomena used in simulation techniques include thermal, hydraulic, and tectonic erosion. Thermal erosion techniques are caused by gravity and the thermal expansion of water under the Earth's surface [28]. Eulerian [4] or Lagrangian [20] approaches can be used to compute hydraulic erosion which simulates the water's movement against bedrock. Tectonic erosion uses tectonic plate movement to simulate mountain range formation [7, 8, 27]. While simulation techniques offer geological accuracy due to the underlying physics, they are computationally expensive and do not scale well. User control is also a notable issue with simulations [15].

2.3 Example-based Techniques

Example-based approaches utilise real-world terrain data in the form of DEMs. DEMs are collections of altitude values stored in a two-dimensional grid [15]. These approaches have achieved high performance, high terrain realism, and effective user control. Although, they rely on the quality of the input data.

2.3.1 Machine Learning. Machine learning techniques train models using large amounts of input data. These models require a substantial amount of time for pre-processing but are incredibly efficient during terrain generation. Guérin et al. [16] trained two competing neural networks that created and assessed terrain using a conditional generative adversarial network (CGAN). The CGAN, inspired by the success of image-to-image translation [19], uses a generator which creates new example terrains and a discriminator which compares the generated terrains to real DEMs. However, the CGAN methodology for terrain generation is prone to produce small grid artefacts [16]. Lochner et al. [23] were able to show the effectiveness of diffusion-based terrain authoring techniques. The Denoising Diffusion Probabilistic Models (DDPMs) [18, 39] or simply diffusion models are influenced by principles from non-equilibrium thermodynamics [36], which is the concept of systematically disrupting the organization within a dataset. This disruption is achieved by iteratively introducing noise to a sample and then learning to reverse this process. Once the model is trained, it can generate new samples by applying the denoising method to pure noise inputs[23].

2.3.2 Texture Synthesis. Texture synthesis techniques reassemble continuous regions of input images to create a realistic output texture that resembles the characteristics of the input data. There is a wide variety of texture synthesis techniques including pixel-based and patch-based approaches. Patch-based techniques combine small patches of terrain taken from real input terrain data [9]. Zhou et al. [40] developed a patch-based algorithm that took DEMs as input and attempted to match them to a user sketch. Poisson editing and graph cuts are used to erase similar sets of edges [21, 32]. Tasse et al. [37] improved on the work by Zhou et al. [40] to allow for a technique that produced more efficient and realistic results. The improvement in realism was achieved by removing noticeable patch seams [15]. Tasse et al. [37] added a Shepard gradient interpolation process to the patch merging technique. Gain et al. [14] developed a parallel pixel-based terrain synthesis technique that improved efficiency and user control over existing works. Improved user control was achieved by a remarkable interface with constraints, a copy-paste feature, and sketching and painting tools. The synthesis averaged times of 63ms and 151ms for resolutions of 512^2 and 1024^2 [14].

3 RESEARCH QUESTIONS

The broad research problem is that there are no complete algorithms and authoring techniques for efficiently generating realistic, adaptable, and well-scaled Earth-like planets where the user can sketch landscape features onto a globe. The project includes aspects drawn from O. Borg's honours project (2023) which explored the generation of planets [5]. Section 6 addresses the optimal outcomes ensuring that the research questions are answered. The objective is to improve upon the limitations of existing terrain generation techniques such as PlanetAI (O. Borg's honours project) and Gain et al. [14] while adapting them for global generation. Namely the lack of available data to train the generative models, super-resolution, and using a single model trained on multiple higher-resolution zoom levels and adapting upsampling, jitter, and correction techniques to deal with planetary generation. The generation of these planets will be accomplished via two methods of terrain generation: diffusion models and texture synthesis. Specifically, the following research questions will be addressed:

- Does a texture synthesis method allow for realistic generation of Earth-like planets using a pixel-based method? (EarthSynth)
- How effectively can diffusion models be adapted to generate a realistic and well-scaled Earth-like planet? (PAID)
- How effectively can our evaluation procedures assess and compare the efficiency, realism, adaptability, and scalability of the above two methods using common assessment procedures? (PlanetEval)

4 PROCEDURES AND METHODS

Our work will utilize part of O. Borg's sketch-to-terrain interface [5], however, adaptations and extensions will be made to incorporate a global sketching interface, diffusion model and texture synthesis model. EarthSynth and PAID will utilize data from Google Earth Engine which provides the DEMs necessary for the approaches. The data processing pipeline will be adjusted to account for the new data.

4.1 Systems

Our project's systems will be designed with methods that are utilised in the following order:

(1) Globe Sketching Interface
(2) Tile Generator
(3) Sketch-to-DEM using Diffusion model / Texture Synthesis
(4) Global Terrain Creation

Methods 1, 2, and 4 will integrate a shared interface and models used to capture and visualise the user sketches. These methods will be described as the Shared Methods. The main section, method 3, will include the algorithms specific to each approach. The project is designed to optimise the two methods' comparison by using largely similar core methods. Therefore, the comparison is solely attributable to the specific approaches (diffusion models or texture synthesis).

4.1.1 Shared Methods.

- Globe Sketching Interface: This interface will be developed to allow users to sketch onto a globe. These sketches will then be used to generate the terrain for the planet. Basic tools allowing for feature allocation and elevation determination will be used. An extension allowing for more tools for sketching is possible.
- Tile Generator: This method will decompose the spherical sketch of the planet on the globe onto a 2-dimensional surface. This will use UV mapping [6] to represent the sketches on a 2D surface without distortion. Once the user sketch is on a single plane, the tile generation algorithm will split the sketch into defined regions of tiles which the sketch-to-DEM model can use to generate the output DEM tiles.
- Global Terrain Creation: This method will take the terrain generated by the diffusion models or texture synthesis methods and apply it back to the globe. The tile merger model seamlessly integrates adjacent tiles to construct a planet-sized terrain. Without the merging model, seams would be visible leading to a deterioration of the realism of the terrain. O. Borg implemented alpha blending [22], Poisson blending, and a Graph Cut algorithm [9, 21]. The methodology requires 4 times as many tiles to be generated to ensure sufficient overlap for the merging system[5]. Investigating using an extra column and row in the tiling grid system could improve the number of tiles generated to 1.9 times.

4.1.2 Specific Methods.

Sketch-to-DEM. PAIN and EarthSynth will separately implement the Sketch-to-DEM model using diffusion models and texture synthesis respectively. This method will generate a DEM based on differing precision scales. The precision of each zoom level is represented as km^2 per pixel, this is the real-world distance that each pixel would resemble. This gives an estimation as to how fine-grained the resulting terrain will be represented at each zoom level. The precision differences at each zoom level are increased

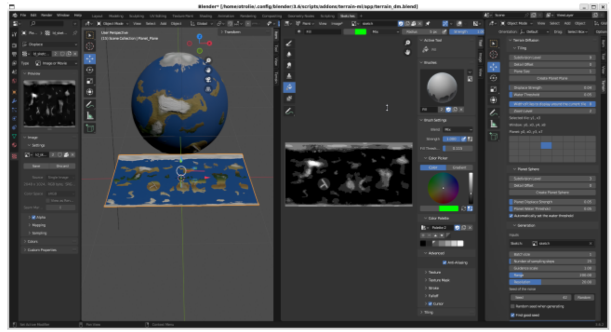

Figure 2: O. Borg's [5] implementation of a blender interface used to author planets.

by a factor of 2. The initial precision of 40,075 km (the Earth's circumference) at zoom level 0 is set to 2048x1024 pixels which is the optimum resolution for the sketch-to-DEM model to produce a realistic output. Each zoom level represents the resolution of the rendered output terrain of the full planet. Each tile of the model is 256x256 pixels as this is the size of the input and output.

PAID. The PAID (Planet Authoring Interface using Diffusion Models) method will implement the sketch-to-DEM model once the user has specified general regions of elevation and land formations. The methods will use the diffusion models described in section 2.3.1 to synthesize terrain. PAID consists of a data processing pipeline and diffusion model training. The system developed by O. Borg [5] implements each of these pipelines however adaptations will need to be made to integrate a super-resolution model and the training of a single model on multiple zoom levels.

The super-resolution model will create high-resolution images from low-resolution inputs by iteratively refining the image details. This process allows for producing high-quality results with fine-grained control over the enhancement process. The enhancement allows for sketch up-scaling for the first few zoom levels providing control over individual zoom levels while necessitating terrain generation solely for higher zoom levels. An approach to investigate is using a single model trained on multiple zoom levels to decrease the training complexity. A single encoder will inform the model of the zoom level for the input and output data. PyTorch [31], a tensor flow library for machine learning, and Hugging Face Accelerate [13], were used in O. Borg's [5] deep learning pipeline. PAIN will utilize the developed architecture however further work and adaptations to their pipelines will need to be implemented to accommodate for a super-resolution model and the single encoder.

EarthSynth. The texture synthesis methods of EarthSynth will provide the ability to generate Earth-like planets from user sketches. EarthSynth will implement the sketch-to-DEM algorithm that generates elevations at various precision levels. A comprehensive synthesis framework will allow for the efficient generation of realistic, adaptable, and well-scaled planets. The texture synthesis framework of Gain et al. [14] provides an elegant example of a pixel-based texture synthesis algorithm used for terrain generation. Although developed in 2015, it remains one of the leading frameworks for

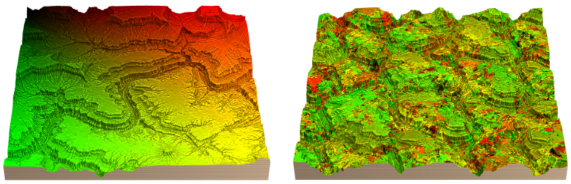

Figure 3: Gain *et al.* [14] showing the use of texture synthesis models for terrain generation.

texture synthesis usage for terrain generation. Scott and Dodgson's [35], although impressive, is still eclipsed by Gain et al's [14] implementation in various aspects. EarthSynth will follow the general structure of a texture synthesis method which is as follows: (1) Input Texture (2) Feature Extraction (3) Modelling (4) Synthesis (5) Refinement (6) Output Terrain.

EarthSynth will then set up a framework based on Gain et al.'s [14] implementation but with necessary adjustments to deal with the increased scale. The necessary permissions will be attained as necessary. As such, EarthSynth will consist of a pre-generated coarse-to-fine pyramid of exemplar data, then upsampling and correction algorithms executed at run-time. Upsampling algorithms provide finer resolutions for synthesis and correction algorithms move coordinates towards better matches with neighbouring pixels in the exemplar [14]. The PatchMatch algorithm is widely used for finding the best matches between the exemplar and output neighbourhoods and will be utilised by EarthSynth [1].

Unlike Gain et al.'s [14]implementation which used GPUs, PlanetSynth will only use the CPU. This will affect the performance capabilities for larger exemplars but will reduce hardware requirements. Gain et al.'s [14] jitter phase that adjusts coordinates to improve visuals and Scott and Dodgson's [35] pit-removal algorithm that improves realism are potential additions to EarthSynth.

4.2 Evaluation

The evaluation procedures will be kept constant between the PAID and EarthSynth methods. This allows for comparable results and strengthens any findings. Evaluations of each approach will be conducted separately after which a comparison of the overall methodology (PlanetEval) will occur.

4.2.1 Realism. Realism refers to how accurately the generated terrain mimics features found in real landscapes. This is important in terrain generation as it encompasses factors like physical plausibility, visual authenticity, scale, detail, diversity and variation of the landscape features. Viewer perception (perceptual realism) or qualitative assessment (geomorphological realism) can be used to ascertain the realism of the generated terrain however the measurements often contradict due to selective feature bias [15]. For this project, the realism evaluation will be accessed using static renderings where we can perform side-by-side comparisons of generated and real terrain. A qualitative evaluation will assess whether the generated terrain contains artefacts, line stitches, landmark features and texturing [15]. Gain *et al.*[14] performed a user study in which 25 subjects evaluated the perceived realism of 60 pairs of terrain images however, performing a user study may be infeasible due to time limitations. The Perceived Terrain Realism Metric (PTRM) proposed by Rajaskaran *et al.* [33] could be used as a qualitative evaluation to estimate the human-perceived realism of a terrain.

4.2.2 Efficiency. Time efficiency is integral to terrain authoring as it allows for an interactive environment. Time efficiency can be classified as the time between a user input and the terrain's update. Galin et al [15] propose an efficiency evaluation method in which the efficiency is categorised into four general groups: real-time (less than 1/3 of a second per update), interactive (less than 3 seconds per update), seconds (greater than 3 seconds but less than a minute per update), and minutes (anything greater than a minute per update). PAID and EarthSynth will be categorised into one of the general groups.

Space Efficiency is worth examining, especially during the preprocessing stages. Where vast amounts of memory are needed to train example-based methods [15]. We have access to UCT's HPC for the pre-processing stage of the PAID method. Typically lower terrain resolutions of $512^2 - 1024^2$ pixels require $1 - 4$ MB of memory however, higher zoom levels of the model require image sizes of 32768x16384 pixels, equivalent to 4 GB of memory which affects the detail and quality of the generated terrain. Memory profilers provided by Nvidia-smi [29] and Pytorch [24] will be used to examine memory efficiency.

4.2.3 User Adaptability. An important aspect of terrain generation is how effectively an interface provides user control over the placement of landscape features. An authoring interactivity test will be conducted to determine whether the generated terrain resembles the structure of the user sketch, adhering to general elevation constraints and the placement of landscape features. A task completion test will be performed in which the user will be asked to sketch specific landscape features such as mountain ranges, river networks and general elevation regions using the global sketching interface. An analysis of the Mean Squared Error (MSE) will be used to determine the accuracy of the elevation constraints, comparing the average squared difference between the user-specified elevation and the generated elevation. The specific criteria we will assess will be as follows:

- Placement of features
- Type of feature
- Complexity of feature authoring
- Effect on other evaluation procedures

4.2.4 Scalability. Scalability determines how the model performs at each of the zoom levels. This metric can analyse the feasibility of increasing the zoom levels. O. Borg [5] showed that with the first 2 zoom levels at an image size of 512x256 pixels and 1024x512 pixels, the model did not perform and could not produce realistic outputs. Scalability will show the possibility of increasing the image sizes to 65536x32768. At this increased resolution size, the realism of the output terrain would be greater, but it may be too computationally expensive to perform. For each zoom level, the output terrain will be assessed for realism and efficiency. This will show the model's scalability.

5 ETHICAL ISSUES

This project is completed under the guidance of a University of Cape Town supervisor and is conducted as academic work. All efforts have been made to avoid infringements on ethical, professional, and legal standards.

- **Data**. The data that will be used in this project will be data from Google Earth Engine. The machine learning and texture synthesis approaches will utilise terrain data as DEMs. This data is freely available and under no copyrights. Additional pre-processed data may also be used from previous honours projects/students from which the required permissions will be attained.
- **Code**. The code will combine original code, open-source libraries, and publicly available software. The necessary permissions will accompany additional code taken from previous years' projects.
- **Intellectual Property and Licensing**. All Intellectual Property (IP) created during this project is the property of the University of Cape Town as set out in the UCT Intellectual Property Policy [30]. The code created will be open-source so that the methods and results can be used for further scientific and academic development.
- **Referencing**. The project and all related documents will adhere to the Association for Computing Machinery (ACM) Proceedings Format for referencing. This includes all images, figures, graphs and other data representations used.
- **User Studies**. User studies or experiments will not be necessary for this project as the project members can assess any user-related feedback needed. We believe that user studies would be outside the scope of this project. Therefore, no ethical clearance will be needed.

6 ANTICIPATED OUTCOMES

6.1 System

Specific difficulties regarding the balance between scalability and performance will be overcome. The trade-off between performance and scalability will be explored in the PAID and EarthSynth methods. Multiple precision levels combat this by allowing for separate levels to be run in parallel. This reduces the performance constraints created by increased scalability.

6.1.1 PAID. The PAID methods will be able to generate realistic synthetic terrain with a variety of land formations. The diffusion model should provide the user with interactive authoring where updates to the user sketches are shown in close to real-time. The architecture should provide 6 zoom levels of increasing precision. Furthermore, the model should be computationally efficient with a super-resolution model and a single model trained on multiple zoom levels. The major design challenges to overcome are the implementation and training of the aforementioned models in constrained time.

6.1.2 EarthSynth. The EarthSynth methods will largely revolve around the software and main feature: the synthesis framework. If it can generate realistic terrains then it will be largely successful. Adaptability and scalability are important aspects too. Speed efficiency will be key because EarthSynth will only be run on a laptop CPU. This will need to be overcome for EarthSynth to succeed. Features such as the globe sketching interface, tile generator, and the global terrain creation algorithm will also be important to EarthSynth's success.

6.2 Research

6.2.1 PAID Hypothesis: The PAID methods will be developed to generate realistic terrain, as assessed by our realism evaluation procedures, with different land formations and increasing precision levels.

6.2.2 EarthSynth Hypothesis: EarthSynth is a collection of adaptable texture synthesis methods to generate realistic Earth-like planets, as assessed by our realism evaluation procedures, and can be achieved on a suitable laptop.

6.2.3 PlanetEval Hypothesis: Comparing the PAID and EarthSynth methods via the evaluation procedures shows that PAID has high efficiency and realism and EarthSynth has high realism and low hardware requirements.

6.3 Impact

The results of our project will allow for improvements in graphic design scenarios such as training, simulation, games, and films. These fields will be aided by the ability to create realistic Earth-like planets. The implementation of these methods on a global scale has been under-researched and our project will aim to provide insight into this area.

6.4 Key Success Factors

The project's success relies on the evaluation criteria for the PAID and EarthSynth methods as described in Section 4.2. The systems should provide an efficient interface in which the output of the generated terrain is realistic and scalable. The terrain authoring techniques should allow for user adaptability where the user can specify the variety of land formations resembling the sketch on a global interface. Further critical comparison of PAID against EarthSynth will show the utility of diffusion models or texture synthesis for global terrain generation and authoring.

7 PROJECT PLAN

7.1 Risks

We recognize the risks of shared method contributions in our large-scale project. This approach is necessary due to the similarity of methods across sections. Should one member leave, the other will incorporate the shared methods into their workload. The risks and risk management strategies are presented in Table 3.

7.2 Timeline

The milestones, deliverables, work allocation and timing of the project timeline will be presented in Gantt Chart 4.

7.3 Resources Required

This project will require stable, fast internet access and electricity to develop our software. Open-source code and free software will be used in conjunction with our code. The UCT High-Performance

Computing (HPC) facility will be utilised to train our models for the machine learning component of the project. Evaluation and implementation of procedures will require powerful PCs. Data from Google Earth Engine will be used to acquire input terrain data and train models.

7.4 Deliverables

Table 1 shows project deliverables and dates. The deliverables are related to the milestones in Gantt Chart 4.

7.5 Work Allocation

The work allocation for each group member is shown in Table 2. The work is evenly shared with Christopher focusing on the machine learning element and Sam on the texture synthesis element.

Table 1: Project Deliverables

Deliverable	**Date**
Literature Review Final Submission	18-March
Project Proposal Presentations	22-April to 25-April
Project Proposal Final Submission	30-April
Ethics Applications Deadline	6-May
Project Progress Demonstration	22-July to 26-July
Complete Draft of Final Paper	23-August
Project Paper Final Submission	30 August
Project Code Final Submission	9-September
Final Project Demonstration	16-Sep to 20-Sep
Poster	27-September
Website	4-October
School of IT Showcase	22-October

Table 2: Work Allocation

Christopher - Machine Learning	**Sam** - Texture Synthesis
Generate DEM dataset from Google Earth Engine data	Generate DEM dataset from Google Earth Engine data
Implement Tile Generator	Implement a Globe sketching interface
Extend O. Borg's tiling system to use super-resolution	Implement a synthesis framework for global terrain synthesis

REFERENCES

[1] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. 2009. PatchMatch: A randomized correspondence algorithm for structural image editing. *ACM Trans. Graph.* 28, 3 (2009), 24.

[2] Bedrich Benes. 2007. Real-time erosion using shallow water simulation. (2007).

[3] Bedřich Beneš and Toney Roa. 2004. Simulating desert scenery. (2004).

[4] Bedřich Beneš, Václav Těšínsk'y, Jan Hornyš, and Sanjiv K Bhatia. 2006. Hydraulic erosion. *Computer Animation and Virtual Worlds* 17, 2 (2006), 99–108.

[5] Oliver Borg. 2023. Planet Authoring with Generative AI. University of Cape Town. https://projects.cs.uct.ac.za/honsproj/cgi-bin/view/2023/borg_brann_hitge.zip/assets/Papers/PlanetAI.pdf

[6] Zhiqin Chen, Kangxue Yin, and Sanja Fidler. 2022. Auv-net: Learning aligned uv maps for texture transfer and synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 1465–1474.

[7] Guillaume Cordonnier, Jean Braun, Marie-Paule Cani, Bedrich Benes, Eric Galin, Adrien Peytavie, and Éric Guérin. 2016. Large scale terrain generation from tectonic uplift and fluvial erosion. In *Computer Graphics Forum*, Vol. 35. Wiley Online Library, 165–175.

[8] Guillaume Cordonnier, Marie-Paule Cani, Bedrich Benes, Jean Braun, and Eric Galin. 2017. Sculpting mountains: Interactive terrain modeling based on subsurface geology. *IEEE transactions on visualization and computer graphics* 24, 5 (2017), 1756–1769.

[9] Leandro Cruz, Luiz Velho, Eric Galin, Adrien Peytavie, and Eric Guérin. 2015. Patch-based terrain synthesis. In *International Conference on Computer Graphics Theory and Applications*. 6–pages.

[10] AR Dixon, GH Kirby, and Derek PM Wills. 1994. A data structure for artificial terrain generation. In *Computer Graphics Forum*, Vol. 13. Wiley Online Library, 37–48.

[11] Rae A Earnshaw. 1985. *Fundamental algorithms for computer graphics: NATO advanced study institute directed by JE bresenham, RA Earnshaw, MLV Pitteway*. Springer.

[12] David S Ebert, F Kenton Musgrave, Darwyn Peachey, Ken Perlin, and Steve Worley. 2002. *Texturing and modeling: a procedural approach*. Elsevier.

[13] Hugging Face. 2024. Hugging Face Accelerate. https://huggingface.co/docs/accelerate/en/index

[14] James Gain, Bruce Merry, and Patrick Marais. 2015. Parallel, realistic and controllable terrain synthesis. In *Computer Graphics Forum*, Vol. 34. Wiley Online Library, 105–116.

[15] Eric Galin, Eric Guérin, Adrien Peytavie, Guillaume Cordonnier, Marie-Paule Cani, Bedrich Benes, and James Gain. 2019. A review of digital terrain modeling. In *Computer Graphics Forum*, Vol. 38. Wiley Online Library, 553–577.

[16] Éric Guérin, Julie Digne, Eric Galin, Adrien Peytavie, Christian Wolf, Bedrich Benes, and Benoît Martinez. 2017. Interactive example-based terrain authoring with conditional generative adversarial networks. *ACM Trans. Graph.* 36, 6 (2017), 228–1.

[17] Houssam Hnaidi, Eric Guérin, Samir Akkouche, Adrien Peytavie, and Eric Galin. 2010. Feature based terrain generation using diffusion equation. In *Computer Graphics Forum*, Vol. 29. Wiley Online Library, 2179–2186.

[18] Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *Advances in neural information processing systems* 33 (2020), 6840–6851.

[19] ZhuJY IsolaP, T Zhou, et al. 2017. Image to imagetranslation withconditional-adversarialnetworks. *ProceedingsoftheIEEEConferenceonComputerVisionandPatternRecognition. Hawaii, USA* 1125 (2017), 1134.

[20] Peter Krištof, Bedrich Beneš, Jaroslav Křivánek, and Ondrej Št'ava. 2009. Hydraulic erosion using smoothed particle hydrodynamics. In *Computer graphics forum*, Vol. 28. Wiley Online Library, 219–228.

[21] Vivek Kwatra, Arno Schödl, Irfan Essa, Greg Turk, and Aaron Bobick. 2003. Graphcut textures: Image and video synthesis using graph cuts. *Acm transactions on graphics (tog)* 22, 3 (2003), 277–286.

[22] Lin Liang, Ce Liu, Ying-Qing Xu, Baining Guo, and Heung-Yeung Shum. 2001. Real-time texture synthesis by patch-based sampling. *ACM Transactions on Graphics (ToG)* 20, 3 (2001), 127–150.

[23] Joshua Lochner, James Gain, Simon Perche, Adrien Peytavie, Eric Galin, and Eric Guérin. 2023. Interactive Authoring of Terrain using Diffusion Models. In *Computer Graphics Forum*, Vol. 42. Wiley Online Library, e14941.

[24] Maxim Lukiyanov. 2021. Introducing PyTorch Profiler - the new and improved performance tool. https://pytorch.org/blog/introducing-pytorch-profiler-the-new-and-improved-performance-tool/

[25] Benoit B Mandelbrot and John W Van Ness. 1968. Fractional Brownian motions, fractional noises and applications. *SIAM review* 10, 4 (1968), 422–437.

[26] Kenneth R McClay. 2012. *Thrust tectonics*. Springer Science & Business Media.

[27] Elie Michel, Arnaud Emilien, and Marie-Paule Cani. 2015. Generation of folded terrains from simple vector maps. In *Eurographics 2015 short paper proceedings*. The Eurographics Association, 4.

[28] F Kenton Musgrave, Craig E Kolb, and Robert S Mace. 1989. The synthesis and rendering of eroded fractal terrains. *ACM Siggraph Computer Graphics* 23, 3 (1989), 41–50.

[29] Nvidia. 2023. System Management Interface SMI. https://developer.nvidia.com/nvidia-system-management-interface

[30] University of Cape Town. 2011. Intellectual Property Policy. https://uct.ac.za/sites/default/files/media/documents/uct_ac_za/190/Policy_Intellectual_Property_2011.pdf

[31] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems* 32 (2019).

[32] Patrick Pérez, Michel Gangnet, and Andrew Blake. 2023. Poisson image editing. In *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*. 577–582.

[33] Suren Deepak Rajasekaran, Hao Kang, Martin Čadík, Eric Galin, Eric Guérin, Adrien Peytavie, Pavel Slavík, and Bedrich Benes. 2022. PTRM: Perceived terrain realism metric. *ACM Transactions on Applied Perceptions (TAP)* 19, 2 (2022), 1–22.

[34] Pascale Roudier, Bernard Peroche, and Michel Perrin. 1993. Landscapes synthesis achieved through erosion and deposition process simulation. In *Computer Graphics Forum*, Vol. 12. Wiley Online Library, 375–383.

[35] Joshua J Scott and Neil A Dodgson. 2021. Example-based terrain synthesis with pit removal. *Computers & Graphics* 99 (2021), 43–53.

[36] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. 2015. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*. PMLR, 2256–2265.

[37] Flora Ponjou Tasse, James Gain, and Patrick Marais. 2012. Enhanced texture-based terrain synthesis on graphics hardware. In *Computer Graphics Forum*, Vol. 31. Wiley Online Library, 1959–1972.

[38] Filip Tripkovic. 2023. Agent Based Terrain Generator: Cruthú. Master's Thesis. Malmö University, Sweden.

[39] Ling Yang, Zhilong Zhang, Yang Song, Shenda Hong, Runsheng Xu, Yue Zhao, Wentao Zhang, Bin Cui, and Ming-Hsuan Yang. 2023. Diffusion models: A comprehensive survey of methods and applications. *Comput. Surveys* 56, 4 (2023), 1–39.

[40] Howard Zhou, Jie Sun, Greg Turk, and James M Rehg. 2007. Terrain synthesis from digital elevation models. *IEEE transactions on visualization and computer graphics* 13, 4 (2007), 834–848.

A RISK MATRIX

Table 3: Risk Matrix

Risk	Description	Likelihood	Impact	Management Strategies
Load-shedding	South Africa is prone to high levels of load-shedding which might impact the efficiency of work at home.	Very high	Medium	Organise the daily schedule around the load-shedding times. Utilise OneDrive and GitHub to make regular backups of data, work, and code. The HPC provided by UCT will be used for the project which has backup power to continue heavy computations during load-shedding.
Scope creep	The project scope may expand beyond its original requirements, objectives, and deliverables. New features, functionalities, or tasks may be introduced during the project without sufficient evaluation of the impact on the timeline.	Medium	Medium	Establish a clear set of objectives, requirements, and deliverables in the timeline. If new functions need to be added to the project, ensure adequate time to incorporate the features otherwise reconsider the viability of adding the new features. Constantly manage the project's scope with regular meetings with the project supervisor.
Time estimations	Time estimations are given with a Gantt chart where activities are planned during the project timeline. However, time estimations are often inaccurate, and the implementation of methods takes longer than expected.	High	Medium	Proper time management is essential to adapt to the inevitability of taking longer than expected. Bi-weekly team meetings to check team members' progress using the Gantt chart will ensure team coordination and that features are completed within time or allow for further extensions.
HPC access	The High-Performance Computing facility will be utilized to perform computationally expensive model training for machine learning. The HPC often has limited access due to its necessity.	Medium	Very high	Develop the correct methods, models and core functionality of the machine learning aspect of the project to ensure that it minimizes the computational complexity. Apply for early access to the HPC. Possibly purchase a powerful desktop to run the model on a smaller dataset.
Model failure	The model might not produce the expected results for the project.	Medium	Low	Research and apply the techniques of previous models. If failure does occur, ensure it is understood and fixed or explain the failure as part of the project paper.
Clean data	The exemplar images necessary for training the machine learning algorithm need to be clean without artefacts and human interaction.	Medium	High	Use terrain images from credible sources with a large landform diversity. The variety of images needs to have a high resolution. Ensure the images are clear of human impact by developing a model to find the correlation between the terrain and the human impact index.

B GANTT CHART

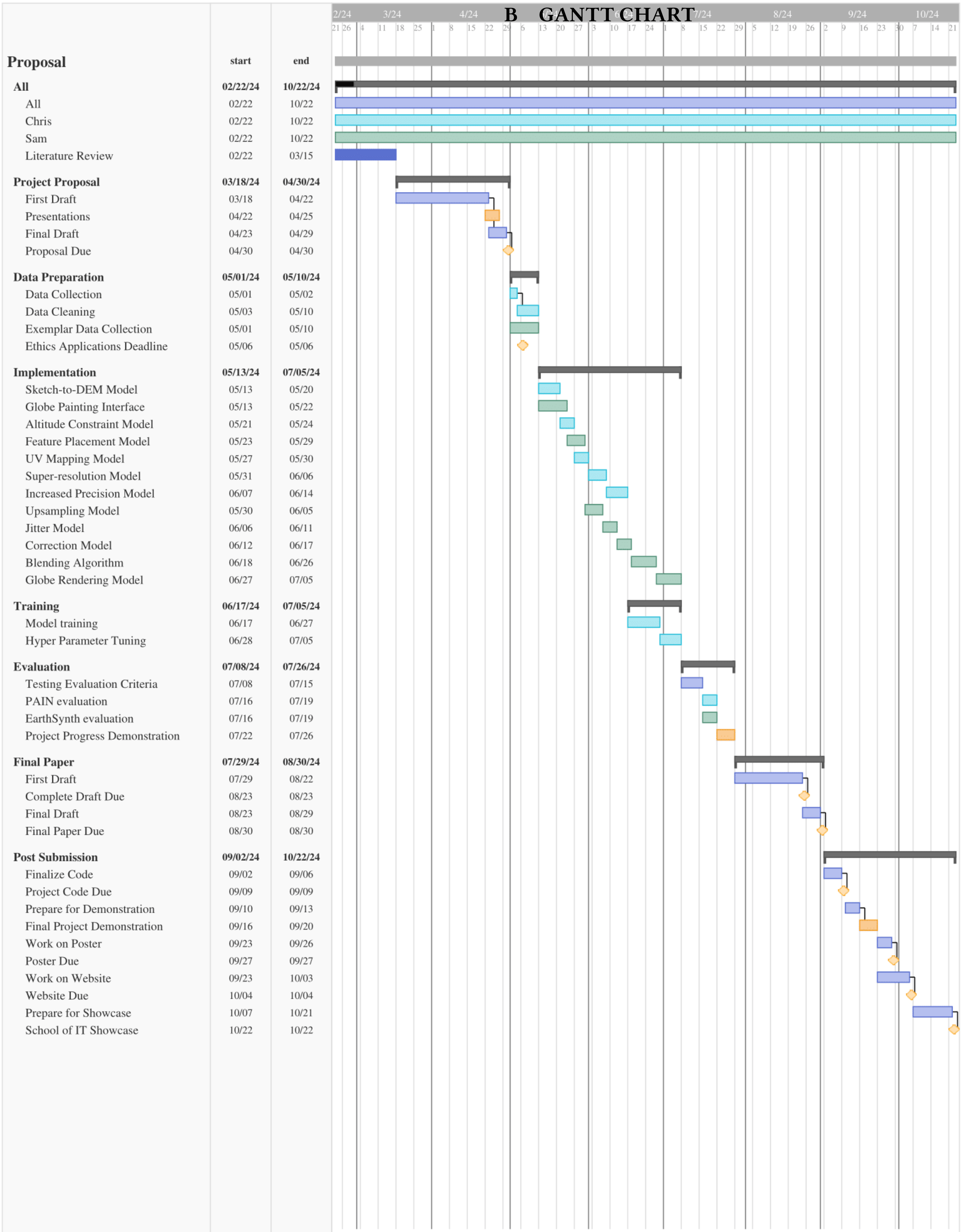

Figure 4: Gantt Chart showing project timeline. Joint tasks are shown in dark blue. Chris in light blue and Sam in green