Realism in Data-Based Terrain Synthesis

by

Joshua James Scott

A thesis submitted to the
Victoria University of Wellington
in fulfilment of the requirements for the degree of
Doctor of Philosophy
in Computer Graphics

Victoria University of Wellington
2020

Supervisory Committee

Primary Supervisor: Prof. Neil Dodgson (Victoria University of Wellington)

School of Engineering and Computer Science

Secondary Supervisor: Prof. Peter Andrae (Victoria University of Wellington)

School of Engineering and Computer Science

Abstract

The landscapes on Earth are varied and complex, having been created by innumerable physical processes over millions of years. The creation of artificial terrain that replicates the realism of landscapes on Earth has been a major challenge for computer graphics. Many different approaches have been taken, including approximating the terrain with fractals and splines, simulating the terrain using models from the physical geography, and reconstructing terrain from elements of real-world data. A primary issue in the field of terrain synthesis is the lack of, and evaluation of, realism in synthesized terrain.

This thesis identifies and discusses the flaws of existing data-based methods based on example-based texture synthesis methods. It provides improvements to an existing data-based method using algorithms from the field of geographic information science, and presents a novel algorithm, “terrain-optimization”, based on the example-based texture synthesis technique of texture-optimization. Finally, it discusses a new approach to the experimental evaluation of terrain realism, with the largest experiment conducted to date. The results of this show that each of the tested methods is indistinguishable from reality in certain circumstances and that those circumstances differ for each method tested, and that subjects with a high level of expertise in physical geography are the most qualified for identifying real terrain from synthesized terrain.

Overall, the thesis provides substantial analysis and evidence about the challenges of data-based terrain synthesis while also developing new approaches in the field that perform as well as existing state-of-the-art methods.

Acknowledgments

My complete and utter gratitude to all those who helped me to complete my PhD.

To my supervisors, academic, and professional staff at Victoria University of Wellington. Neil Dodgson, my primary supervisor, for being encouraging, supportive, understanding, and providing his expertise and contacts to help me see my research to completion, even beyond the submission of my thesis. Peter Andreae (Pondy), my secondary supervisor, for giving me direction and guidance regardless of his unfamiliarity with my subject of research. Kevin Norton, Associate Professor in the School of Geography, who volunteered his time and expertise to instruct me in physical geography and organized participants for my research. Lisa Woods, consulting statistician, for her guidance and validation of the statistics in my research. Siyun Jeong, Monoa Taepa, and Tony McLoughlin, professional staff, for their technical and emotional support throughout my enrolment.

To my examiners. Fanglue Zhang, not only an examiner, but a colleague who broadened my knowledge of image synthesis and other areas in computer graphics. Burkhard Wuensche, who curated a wealth of constructive criticism and feedback that helped improve the quality of my thesis post-examination. James Gain, who provided critical information about his own research which helped form the foundation of some of my research in Chapters 3, 4, and 6.

To my friends and colleagues. Benjamin Allen, one of my best friends, who all throughout University and to this day helps guide my learning and understanding of programming and computer graphics. John Lewis (JP), Diana Dean, Andrew Chalmers, and Richard Roberts, my seniors and role-models, who provided

guidance when it came to post-graduate study. Ian Loh, Lohit Petikam, Weng Khuan Hoh, Stephen Thompson, Chris Dean, Thomas Iorns, Dominik Schmid, Ping Liu, Wee Kiat Tay Wan-Duo (Kurt) Ma, Ryan Sumner, and Kieran Carnegie, my friends and colleague, who encouraged me and provided insight into various problems I encountered along the way.

To my family. Angela Wilks, my mother, for her complete emotional and financial support. Andrew Scott, my father, and Jo Wilks, my aunt, who helped edit my thesis for spelling, grammar other mistakes.

And most of all, to my best friend, dearest wife, and love of my life, Georgina Lillian Margaret Scott. Your love, care, and support, carried me throughout the toughest times, and it was a joy to spend my entire time in university alongside you. You mean the world to me, and without you this could not have been possible.

Contents

Supervisory Committee	iii
Abstract	v
Acknowledgments	vii
1 Introduction	1
1.1 Aim and Objectives	3
1.2 Contributions	4
1.3 Thesis Organization	4
2 Background	7
2.1 Terrain Synthesis Background	9
2.1.1 Introduction	9
2.1.2 Realism and Believability	9
2.1.3 The Ideal Terrain Synthesis Algorithm	10
2.1.4 Approximation Methods	11
2.1.5 Model-Based Methods	16
2.1.6 Data-Based Methods	22
2.1.7 Summary	26
2.2 Texture Synthesis Background	26
2.2.1 Introduction	26
2.2.2 Patch-Based Methods	27
2.2.3 Pixel-Based Methods	28
2.2.4 Texture-Optimization Methods	33
2.2.5 Summary	36

2.3	Conclusion	37
3	Flaws of Data-Based Terrain Synthesis	39
3.1	Existing Algorithms	40
3.1.1	Zhou et al. 2007	40
3.1.2	Tasse et al. 2012	46
3.1.3	Gain et al. 2015	50
3.2	Flaws	56
3.2.1	Parameterization	56
3.2.2	Patch Size	60
3.2.3	Feature Identification	63
3.2.4	Thin-Plate Spline (TPS)	71
3.2.5	Interpolation	77
3.2.6	Limitations Using Data	80
3.2.7	User Expertise	87
3.3	Conclusion	88
4	Combining Pit-Removal and Terrain Synthesis	91
4.1	Pit-Removal Algorithms	94
4.1.1	Depression-Filling	94
4.1.2	Depression-Breaching	96
4.1.3	Hybrid Method	99
4.1.4	Summary	101
4.2	Pit-Removal Applications	101
4.2.1	Height-Offset Modification	102
4.2.2	Constraint Addition	105
4.2.3	Flow-Accumulation Appearance-Space	109
4.3	Conclusion	113
5	Terrain-Optimization	115
5.1	Texture-Optimization by Kwatra et al.	116
5.1.1	Basic Algorithm	116
5.1.2	Additions	118
5.2	Terrain-Optimization	122

5.2.1	Core Algorithm	123
5.2.2	Constraints	125
5.2.3	Summary	130
5.3	Terrain Optimization Experiments	131
5.3.1	Expanding the PatchMatch Search Space	132
5.3.2	Pit-Removal	138
5.4	Conclusion	141
6	Subjective Evaluation of Realism	143
6.1	Previous Subjective Evaluations of Realism	144
6.1.1	Tasse et al. Subjective Evaluation	146
6.1.2	Gain et al. Subjective Evaluation	147
6.1.3	Summary	149
6.2	Experimental Design	150
6.2.1	Testing Procedure and Analysis	150
6.2.2	Terrain Visualization	155
6.2.3	Terrain Feature-Bias	166
6.3	Realism Subjective Evaluation	171
6.3.1	Design	172
6.3.2	Subjects	173
6.3.3	Apparatus and Materials	173
6.3.4	Procedure	177
6.3.5	Results	177
6.3.6	Discussion	185
6.3.7	Future Work	189
6.4	Conclusion	191
7	Conclusion	193
7.1	Main Contributions	193
7.2	Future Work	196
7.3	Concluding Remarks	197
	Appendices	199

A Feature-bias Survey	201
B Subjective Evaluation Survey	209
C Expertise Analysis	223
D Subject Evaluation Strategies	227

Chapter 1

Introduction

The creation of realistic artificial terrain has been a challenge for computer graphics for decades. Real terrain is generated by complex physical processes over millions of years and it is a substantial challenge to reproduce plausible virtual terrain. Virtual landscapes that do not exhibit plausible features, or do not adhere to artistic direction, are unfit for practical applications. Realism in terrain generally refers to the terrain's adherence to the physical rules of the world and it is one of the most important aspects of terrain synthesis.

Computer graphics continues to be a critical part of the multi-million dollar film and games industry [139, 134, 133]. Artists are under pressure to produce increasingly high quality visual effects under strict time constraints [115]. Highly detailed assets are rarely reused which results in a constant need for the development of new terrain assets as new projects arise [49, 50]. As a result there are a variety of commercial solutions for synthesizing virtual terrain, but there is currently no consensus on the best synthesis method. In fact, artists often use in-house solutions, which in some cases includes manually modelling the terrain or using scans of real-world terrain. Popular video games the Elder Scrolls, Just Cause, and Fallout series all have large outdoor scenes with tens of square kilometers of terrain [152] modelled by teams of artists. Similar open world games, that feature vast areas of traversable terrain, are continuing to rise in popularity with major companies looking to develop the technology further [142, p. 3]. My

motivation is to research methods that will automate part of the process for generating realistic virtual terrain.

The three core aspects of terrain synthesis for practical applications are realism, control, and speed. Realism refers to how likely the terrain would form due to real-world processes, control is the influence the user has in guiding the synthesis towards the desired result, and speed is how long it takes to synthesize the terrain. This thesis focuses primarily on the realism aspect of synthesizing terrain, with control and speed being important, but considered secondary.

Research in the field initially focused on developing algorithms that approximate the shape of terrain, inspired by Mandelbrot's research that models natural phenomena using fractals [92, 93, 94]. Other approximation approaches followed, including the use of alternative fractal generation techniques [58, 99, 89, 1], distance-based functions [132, 125, 68], and sketching methods [16, 75, 24, 14]. These approximation algorithms were designed to be fast and were especially popular due to the lack of computing power at the time. While later approximation methods allowed a greater degree of control, overall the terrain produced is not realistic as there is little to no consideration of the natural phenomena that shape terrain in the real world. Sketching approaches, in particular, are so unconstrained that they rely almost entirely on the user's knowledge of physical geography to synthesize any kind of realistic structure.

Terrain synthesis methods also include model-based approaches, which synthesize terrain using models derived from physical geography [102, 34, 103]. Some methods in this category synthesize terrain by running simulations at interactive rates [106, 19, 17, 135], and others through the use of high level evaluations [104, 36]. While these methods produced more realistic terrain than the approximation methods, the use of simplified models limits the realism of the terrain that they can produce. For instance, most approaches rely on using over-simplified models of hydraulic and thermal (diffusive) erosion, and ignore other influential forces that shape terrain including glaciers, earthquakes, weather patterns, and animal and human interference. Control is also an issue, as the user can often only change the parameters of the models but cannot specify the shape of

desired terrain. As a result only a limited range of terrain types can be synthesized and the outputs are not realistic and are hard to control.

Data-based methods, that synthesize terrain from existing data, were developed as more real-world elevation data became available [144]. Most of the data-based methods used techniques from example-based texture synthesis algorithms [30, 160, 137, 52, 41, 40, 61] while others use rule-based artificial intelligence techniques [126, 70]. Data-based methods produce more realistic results than model-based methods and allow the user to have a greater degree of control over the synthesis output. However, while the local topology is realistic due to the real-world data provided, the global structure is less realistic than model-based methods as there is no consideration of natural processes. For instance, numerous endorheic basins (an area with no outflow to external bodies of water) commonly occur in a single synthesis as a result of the lack of this consideration, whereas it is very rare to have even a single endorheic basin in a section of real-world terrain.

Overall, current terrain synthesis algorithms are not suitable for producing realistic terrain due to several underlying factors in the design of the algorithm. Many approaches also fail to provide the control or speed required for practical use. Failing to meet these requirements of realism, control, and speed show that they are not ideal as terrain synthesis algorithms and make them less suitable for practical applications.

1.1 Aim and Objectives

My aim is to further the field of terrain synthesis by investigating the flaws in existing algorithms, producing improved methods of terrain synthesis, and creating a procedure to evaluate realism in terrain. The primary focus of this thesis is the realism of terrain synthesis methods, with control and speed as secondary considerations.

In order to achieve this aim my objectives are to categorize existing methods of terrain-synthesis and identify the most promising area of research, implement state-

of-the-art methods in that area, identify their flaws, explore possible improvements, investigate alternative approaches to synthesizing realistic terrain, and investigate appropriate approaches for evaluating realism in synthesized terrain.

1.2 Contributions

This thesis presents the following contributions to the field of terrain synthesis:

- a clear definition of the term “realism” and how it differs from the term “believability”,
- an in-depth critique of the state-of-the-art data-based methods that are based on example-based texture synthesis methods,
- an improvement to the state-of-the-art terrain synthesis method by Gain et al. [61] using a pit-removal algorithm,
- a new approach to data-based terrain synthesis, called terrain-optimization, inspired by the example-based texture synthesis method of texture-optimization,
- a new testing methodology for evaluating the realism of terrain synthesis methods,
- and the results of the largest experimental evaluation of terrain realism to date.

1.3 Thesis Organization

Chapter 2 provides a definition of the ideal terrain synthesis algorithm, defines the objective term “realism” and how it differs from the subjective term “believability”, a review of existing terrain-synthesis methods (including a justification for the direction of research), and a background in texture-synthesis methods that relate to the data-based algorithms explored in the rest of this thesis.

Chapter 3 discusses and identifies the flaws of the state-of-the-art data-based terrain synthesis methods of Zhou et al. [160], Tasse et al. [137], and Gain et al. [61]. It provides an in-depth exploration of the flaws with each algorithm and their effect on data-based terrain synthesis in general.

Chapter 4 investigates various applications of pit-removal algorithms to the terrain synthesis method of Gain et al. [61] to improve the realism of the synthesized terrain.

Chapter 5 proposes a new algorithm called “terrain-optimization” based on a method from the field of example-based texture synthesis. It also experiments with various additions to improve the realism of the synthesized terrain.

Chapter 6 discusses the experimental evaluation of realism in terrain (including previous experiments), and presents a new methodology and results of evaluation for the methods presented in Chapters 3, 4, and 5.

Chapter 7 concludes the thesis with a summary of the research presented and outlines future avenues of research.

Chapter 2

Background

This chapter provides a background on the previous works of terrain synthesis and texture synthesis. I discuss existing approximation, model-based, and data-based terrain synthesis, and conclude that data-based methods are the most promising area for further research. I then discuss pixel-based, patch-based, and optimization-based texture synthesis, which forms the background for most data-based methods and the rest of this thesis.

There are several key observations from this chapter:

Section 2.1.2: The realism of terrain is separate from the believability of terrain. Realism is an objective measure of the likelihood to be created by real processes and believability is the subjective measure of realism. In addition, there is no method to conclusively evaluate the realism of synthesized terrain and only a few studies provide results for the believability of their terrain synthesis algorithm.

Section 2.1.3: The ideal terrain synthesis algorithm has three main objectives: realism, control, and speed. An ideal terrain synthesis algorithm is one that produces terrain that is as real as possible, while meeting user specified constraints, in real-time. While the focus of this thesis is on realism in terrain, control and speed are also considered.

Section 2.1.4: Approximation methods work well to create terrain where control or speed are more important than the realism of terrain. However, they do not

consider the natural processes that shape real terrain and will not produce more realistic terrain than methods that do consider these processes.

Section 2.1.5: Model-based methods can synthesize terrain that is more realistic than approximation methods because they incorporate models from physical geography. However, these physical geography models are an ongoing subject of research as the science of the physical geography on Earth is not yet fully understood and quantified. Model-based methods take the liberty of using simplified versions of these incomplete models in order to have simulations and evaluations completed in a practical time frame on modern consumer hardware. Until more complete physical geography models are formed, and used as part of the algorithm, and the processing speed of consumer hardware improves drastically, model-based methods will not be able to produce suitably realistic terrain for practical use.

Section 2.1.6: Data-based methods are the most promising area of future research because they utilize real-world terrain in their synthesis. To produce realistic terrain, these methods are not bound by the limitation of incomplete geomorphological models, but by the supply of real-world data and the synthesizing algorithm. While these algorithms may be improved by applying physical geography concepts when considering how the synthesized terrain should be arranged, current data-based methods still produce more realistic terrain than both approximation and model-based methods.

Section 2.2: Pixel-based and patch-based texture synthesis methods have been adapted with great success as algorithms for data-based terrain synthesis. Texture optimization is a comparatively good texture synthesis algorithm and to my knowledge has not yet been applied to terrain synthesis.

Finally, this chapter concludes that while data-based methods are the most promising area of research there are many flaws that need to be addressed, and that texture-optimization is an example-based texture synthesis method that is worth adapting to terrain synthesis.

2.1 Terrain Synthesis Background

2.1.1 Introduction

Terrain synthesis in computer graphics has been a subject of research since Mandelbrot suggested that the self-similarity of oceanic coastlines could be approximated using fractals [92]. There are many different approaches to generating terrain and I have classified methods into three broad categories: approximation, model-based, and data-based. Terrain synthesis methods are listed under the category I feel they correlate with most, even though some methods could be classified into more than one category.

2.1.2 Realism and Believability

Realism is an important aspect of terrain synthesis, but in previous research the term “realism” is used ambiguously without a concrete definition. Terms like “real”, “realism”, and “realistic” are often used to describe the visual aesthetics of the terrain instead of adherence to forms created by real-world processes. In addition, the overwhelming majority of terrain synthesis methods have never undergone experimental evaluation of realism - instead they rely on the subjective, informal, and potentially biased evaluation by a few individuals.

To define a measure for realism, one must consider what would be classified as realistic terrain and unrealistic terrain. A simple definition would be to consider that any terrain on planet Earth to be realistic terrain, simply due to its existence. This terrain was formed from an initial condition (the formation of Earth) and shaped over time by real-world processes. Any terrain that has previously existed, will exist in the future, or could exist given reasonable alternative starting conditions, are also products of real-world processes and by extension are also realistic. Conversely, terrain that could not possibly be created through real-world processes would be unrealistic terrain. Terrain that is neither realistic nor unrealistic then falls on a spectrum of realism. With this line of thinking, the realism of terrain with respect to Earth is a measure of how likely that terrain

would have been formed by real-world processes. To clarify the difference between the objective and subjective measure for realism, I have defined the terms of realism and believability.

The realism of terrain is defined as the objective measure of the likely-hood of having been produced by real-world processes on Earth given reasonable initial conditions. Believability is defined to be the subjective measure of realism of a terrain as evaluated by a person.

As there is currently no method of accurately measuring the realism of terrains, it must be done by believability. The quality of this subjective measure relies on the evaluators ability to accurately tell realistic terrain apart from synthesized terrain. This discussion on subjective evaluation of realism is continued in Chapter 6.

2.1.3 The Ideal Terrain Synthesis Algorithm

Artists are under pressure to produce increasingly realistic visual details. Artists are required to produce new assets for each application because highly detailed assets are rarely reused [49, 50]. As with other virtual assets, the shape of terrain must be artist directed to suit the particular application, while maintaining a high degree of realism. There are exceptions where realism is not the goal in synthesizing terrain, but these situations are rare. For example, Olsen et al. [109] aimed to synthesize procedural terrain that was suitable for a particular computer strategy game. The cost associated with creating new terrain assets should also be minimized. Companies in the visual effects industry have thin profit-margins due to the competitiveness of the fixed-bid system, where companies who bid the lowest cost are awarded the visual effects contract [9]. The primary factor in the cost of creation is the amount of time artists spend on developing the assets or effects to meet the specification.

Considering the aim of realistic terrain and artist driven creation process, the definition of an ideal terrain synthesis algorithm can be formed. I define the ideal terrain synthesis algorithm as one that produces terrain that is as real as possible, while meeting user specified constraints, in real-time. This can be broken down into three main objectives for an algorithm: realism, control, and speed.

Realism requires the algorithm to synthesize terrain that is as real as possible, while meeting the constraints. Algorithms that produce terrains that are more realistic are closer to the ideal algorithm. If realism is not required, then a noise or a spline function may suffice instead.

Control allows the user to specify constraints for the algorithm. A limited specification of constraints, or failure to satisfy the constraints, reduces usability as users have less control to guide the algorithm towards an intended result. If constraints are not required, a synthesis algorithm may not be needed and could be replaced by using real-world data instead.

Speed refers to the total time to synthesize a result given the constraints. The longer an algorithm takes to synthesize a result the less ideal it is. If an algorithm takes too long, or never finishes synthesizing, any other method of procuring terrain would be preferred.

With these considerations, terrain synthesis methods in this chapter will be compared and discussed on the basis of the three aspects of the ideal terrain synthesis algorithm. However, this thesis will primarily focus on realism, with control and speed as secondary considerations.

2.1.4 Approximation Methods

Approximation terrain synthesis methods employ algorithms that take a mathematical approach to generating terrain. There is no regard for the natural processes that create real terrain, with the exception of a few basic assumptions such as, “rivers flow downhill”. The resulting terrain approximates “typical-terrain” features like mountains and valleys.

Early research of terrain synthesis in computer graphics was inspired by Mandelbrot [93], who recognized that a wide variety of natural phenomenon can be approximated by using fractals. Generating terrain with fractals has little memory overhead and is efficient to compute. The first published paper on terrain synthesis using fractals was by Mandelbrot [92], which uses fractional Brownian

motion (fBm) to generate an approximation to the height-map for an island. Perlin [114] created an inexpensive method that generates fBm as a multi-dimensional texture with many applications in both procedural texture generation and terrain synthesis. The use of layering noise was later explored by Schneider et al. [128], who combined the Rescale-and-Add method [127] and domain warping to achieve a real-time synthesis of infinite terrain. Layering scaled noise is known as multifractals, and is a popular method for generating large or infinite terrain when realism is not as important as efficiency, such as in the games: Minecraft [2], and No Man's Sky [107, 62].

Needels [105] independently investigated terrain synthesis using a multivariate Gaussian distribution called the modified bivariate normal (MBVN) model. This was later expanded upon by Parry [111] who modified the MBVN to allow asymmetric distributions by changing key parameters. This parametric approach is not trying to model visually correct terrain, but model terrain that is useful for mathematical applications like trajectory calculation. This work influenced Pickover [117] to extend the fractal based generation methods by using modified near-Gaussian distributions to replicate craters that appear commonly in extraterrestrial terrains. Parametric models may be suitable for basic applications where surfaces need to be described mathematically, but they are unsuitable as terrain synthesis algorithms due to the lack of consideration for the underlying processes.

Fournier et al. [58] developed fractal-based methods further by creating the midpoint displacement algorithm which produced a mesh with vertices displaced by fBm using subdivision. Midpoint displacement was later expanded upon by Miller [99] who provided an analysis on triangle-edge and diamond-square subdivision and who introduced square-square subdivision. These methods of subdividing a geometric structure produced visible artifacts compared to true fBm, as the triangle-edge method produces horizontal and vertical patterns due to the exclusion of lateral values when interpolating edges; the diamond-square produces holes or peaks due to the variation of contributions between the square and diamond sub-steps. Lewis's work [89] generalizes the stochastic subdivision methods to remove the artifacts caused by using traditional midpoint displacement

methods. He also generalizes the stochastic models to different power spectra, other than fractal, showing the possibility to generate different styles of terrain, like desert dunes, by altering the parameters. Adams et al. [1] outlined a midpoint displacement method that calculates the height of each vertex in an arbitrary order which is similar to Lewis's [89] approach.

Mandelbrot [94] redeveloped triangle-edge subdivision to remove the creases and self-similarity of traditional fractal methods while introducing new rules for generating river networks. Rivers are defined along the edges of triangles and when subdivided their midpoint displacement is forced to be negative to ensure a downhill flow. While this method does remove some of the artifacts caused by the original triangle-edge subdivision, its river model is unrealistically rugged at lower elevations. Prusinkiewicz and Hammel [119] created an approximation for river generation by using triangle midpoint displacement, modelling the rivers with squig curves. Squig curves outlined by Mandelbrot [93] are line segments that have been iteratively subdivided using the midpoint displacement method and can be used to approximate meandering rivers. A more recent approach by Derzap et al. [47] uses a similar controlled subdivision approach to create planet-scale terrain with basic assumptions about the layout over continent-scale rivers. The method by Ariyan and Mould [5], uses a 2D curve network to describe ridges and valleys (as well as their profiles) and assigns heights along the curves based on a random-walk-based procedural method. The resulting elevations are similar to fBm, but can be controlled with a user-drawn probability distribution. However, as with other approximation methods, the results are not realistic in terms of how true rivers and ridges behave, due to the simplified assumptions made about natural phenomena.

Krten [84] creates terrain using a fault line generation technique. Starting with a flat height-map, an iterative process generates a line that crosses the height-map from one side to the other at random. The heights are raised one side of the line and lowered the other, with the magnitude of the change in height decreasing every iteration. The result is a noisy fractal-like height-map which is then smoothed using a single-constant finite impulse response filter, to emulate erosion and deposition, although there is no specific geomorphological model taken into

account. Building on this idea, Kamal and Uddin [79] create terrain where multiple lines are generated across the area, and the cells (defined by the intersections of these lines) are selectively raised and lowered. The user then controls the iteration count, line count, and cells selected. While fault line generation algorithms use a similar idea to how tectonic plates create terrain, they do not behave in a natural way and fail to deliver realistic results. It may be more suitable for continent-scale terrain synthesis, but it is hard for the user to control.

The challenge of generating terrain with features like cliff faces and overhangs was tackled by Gamito and Musgrave [63] using a vector flow field that modifies the positions of vertices of the terrain using a three-dimensional vector. Fractal noise was used to generate the directions for each vertex allowing the terrain to have sheer cliff faces and overhangs. However, the system does not support caves or arches and the output can result in self-intersecting or knotted terrain. Geiss [64] approached three-dimensional terrain using voxels and outlines a method of generating terrain by evaluating various density functions. The output is difficult to control for users who are inexperienced. The math behind density functions and the synthesized terrain appears to be more suitable for underwater or extraterrestrial terrain as shown in their examples.

A sketching method was proposed by Belhadj and Audibert [16] where the user sketches the desired ridgelines on a height-map. The ridgelines, represented using splines, deform the initial mesh and a simplified physically-based hydraulic method is used to construct the corresponding river network. They proposed the Midpoint Displacement Inverse (MDI) algorithm to fill out the rest of the height-map from the given constraints at points along the splines. The main issue of the MDI algorithm is the estimation for the inverse of the midpoint displacement, where the order of estimation can lead to severe artifacts due to missing data points as the algorithm progresses in a top down fashion. In a subsequent paper Belhadj [15] addresses this by proposing a Midpoint Displacement Bottom Up (MDBU) algorithm for calculating the sparse data before doing a modified midpoint displacement algorithm to fill out the rest of the height-map. Talgorn and Belhadj [136] outlined a GPU implementation and proposed an improvement to the model that relaxes the extrapolation step in the

MDBU algorithm, which reduces the appearance of artifacts in the terrain. Sketching gives a great deal of control to the user but relies heavily on the user's ability to guide the algorithm to produce realistic results. These algorithms will not produce realistic results when there are few or no constraints made by the user.

A distance-based sketching approach by Rusnell et al. [125] allows the artist to place generator nodes over the height-map. The generators are used to create a cost field by evaluating distance from a pixel to each generator using a custom height-profile function. The height of each pixel in the height-map is then determined by traversing the cost field using Dijkstra's algorithm [48]. Golubev et al. [68] builds on this work by using weight functions instead of height-profile functions, and by optimizing the calculation of height values to be independent of the number of generators. Stachniak and Stuerzlinger [132] created a method using stochastic search to bend fractal terrain to user defined constraints. These methods work well for generating terrain to user constraints and developing coarse terrain height-maps quickly, but they do not produce realistic fine detail and the process requires the user to have domain knowledge in order to place the constraints correctly for realistic outputs.

A pure sketching based approach was proposed by Gain et al. [60], where the artist directs the terrain synthesis by drawing the silhouette of the mountain, the base line and its shadow. Using these as a guide the algorithm then generates a fractal mountain constrained within the boundaries. The authors have also outlined how to apply additional fractal detail to the terrain by using a user brush stroke which implicitly defines the parameters needed for the fractal noise. This means the user can draw around a region and then determine the amount of fractal noise to apply by sketching a jagged line inside the boundary. Another sketch-based method by Hnaidi et al. [75] has the artist create control curves to define the shape of the terrain at any point, not just ridges. Diffusion equations are used to solve for the terrain elevation and then fractal noise is applied to increase the fine detail. This was later improved upon in Bernhardt [24] who used a CPU-GPU coupled computation with multigrid solvers to allow the user to sketch and interact with constraints in real time. Becher et al. [14] expanded the curves to work in three-dimensions, allowing the creation of arbitrary terrain surfaces with cliffs, caves and holes.

In summary, approximation terrain synthesis techniques provide a fast and scalable method of generating coarse approximations to terrain. Many of these methods give full control to the user and are able to meet the given constraints exactly. However, the methods either produce unrealistic terrain because they do not consider the effects of real physical phenomena, or they rely almost entirely on the users expertise to create realistic terrain with little assistance. Artifacts are also common in many of these methods as a side effect of using functions and noise which are not suitable for representing the surface of terrain.

2.1.5 Model-Based Methods

Model-based terrain synthesis methods use models from physical geography related disciplines to emulate the forces that shape terrain surfaces. Model-based methods can be further split into two categories: simulation and evaluation. Simulation model-based methods refer to the operation of a model over time, involving an iterative process where the terrain is formed by applying rules of natural phenomena to shape the terrain. Evaluation model-based methods statically calculate the shape of the terrain using statistical models, synthesizing terrain without the need to simulate natural processes over time.

Simulation-based Methods

Musgrave et al. [102] introduced the idea of combining fractal and erosion methods to produce terrain. The algorithm starts with a fractal model of the terrain followed by a simple simulation of hydraulic and thermal erosion. Roudier and Peroche [124] presented a more complex model of erosion that also incorporated chemical dissolution - a process that percolates through the ground and dissolves rock. The algorithm simulates the erosion on a three-dimensional geological representation of terrain with five different material types. Both of these models are simple cellular automata that assume that water in a cell flows entirely towards its lowest neighbour. This was later improved by Chiba et al. [34], who presented a modified model for hydraulic erosion using a velocity field to allow water to flow into multiple neighbouring cells. Beneš and Arriaga [21] modified

Musgrave et al.'s thermal erosion method to produce table mountains. They use two separate height-maps to represent the hard terrain forming the rim-rock and soft terrain for the slopes. These erosion models produce more realistic terrain than fractal-based methods because they more closely resemble real-world processes.

An alternative to the height-map-based erosion was presented by Chiba et al. [35] who used ridgelines to simulate the effects of erosion instead. Starting with some user defined principle ridgelines, a height-map is derived from the existing ridges, and water flow is simulated on the terrain. The existing ridgelines are altered to simulate the effects of erosion and additional derivative ridgelines are added in a recursive refinement process. The first use of parallelization for erosion algorithms was by Beneš [23], where thermal erosion algorithms were computed on multiple CPUs across a network. Having simplified models also reduced the potential realism of the terrain and aspects like transportation of sediment would lead to buildups of landmass in locations that would make the terrain look less realistic.

Nagashima [103] looked at improving existing erosion algorithms by using more complicated weathering models, taking into account different layers of terrain types in the simulation. This was later improved upon by Beneš and Forsbach [22] who used a multi-map approach to represent different materials in columns across the terrain. This improved the simulation of erosion by providing multiple types of solids with different densities, which allowed the erosion process to become more complex and closer to simulating the natural forces that drive the formation of terrain. This column representation is more compact than a voxel grid while still being able to generate caves and overhangs because the system can use air layers. An approach to modelling three-dimensional rocky terrain found especially on coastlines was tackled by Ito et al. [77]; it uses voxels to represent rock structures and a weather simulation to erode them. Using alternative physically-based hydraulic models, Beneš et al. [17, 19] used voxels to store information about water, sediment and rock, performing the hydraulic erosion algorithm in three-dimensions. Petyavie [116] built on the idea of two-dimensional grids of material stacks and allowed users to direct three-dimensional erosion to modify terrain at interactive rates. The method also procedurally generates rock piles in appropriate locations and the final terrain mesh is generated using the

polygonization of the implicit surface defined by the material stacks [27]. While the method does provide interactive editing using three-dimensional erosion, the realism of results depends on the expertise of the user and the underlying algorithms for sediment stabilization and the placement of rock piles are simplified approximations leading to unrealistic detail on the terrain.

Hydraulic erosion for simulation-based algorithms has been a significant focus over the last couple of decades. Beneš [18] presented an erosion technique that separates out erosion simulation into steps and simplifies the parameters for users. However, it is not physically realistic. The algorithm is adjusted with ad-hoc methods that have no physical basis but which are considered, by Beneš, to produce results that look more realistic. Olsen [109] modified Beneš's method and combined it with various initialization methods to produce terrain suitable for application in games. This is a notable outlier of terrain synthesis algorithms where the aim is expressly to generate terrain that prioritizes practicality (for use in a strategy game) over realism. Neidhold et al. [106] investigated ways to merge fluid simulation and erosion algorithms while running at interactive rates. This produced a new algorithm for water transportation that gave the user the ability to define local water sources and modify erosion parameters at run time. This method of water simulation was used by Anh et al. [3] to create a Newtonian physics-based method of hydraulic erosion, implemented on the GPU in order to perform in real-time. Other real time approaches include the work by Beneš [20], which modeled erosion using shallow water equations by Kass et al. [81], where fluids are modelled as non-overlapping waves, and by Mei et al. [97], who focused on real time simulation on the GPU using the virtual pipe model of water transport introduced by O'Brien et al. [108]. These methods mainly focused on improving the performance of the water simulation algorithms, rather than the effect that they have in shaping terrain to be realistic. While most hydraulic erosion algorithms produce visually plausible results, the methods are confined to producing outputs dominated by fluvial erosion processes, which is only a sub-set of terrain types that exist on Earth. Other influential forces that shape terrain include glaciers, earthquakes, weather patterns, and animal and human interference. To simulate more realistic terrain additional materials and processes must be considered, as

without them even synthesized terrains that are dominated by fluvial erosion are still unrealistic. Another issue is the level of user control over the final result, as these simulation methods take parameters and guides given by the user and iterate until a result is achieved.

Combining of thermal and hydraulic models of erosion is a popular approach for simulated model-based terrain synthesis. Št'ava et al. [135] combined the dissolution-based erosion inspired by Beneš [20], the force-based erosion by Mei [97], and thermal weathering adapted from Musgrave et al. [102] on a multi-layered data-structure. Another method by Jákó [78] combined the hydraulic model by Mei et al. [97] and thermal erosion model by Beneš and Forsbach [22] into a real-time erosion model on the GPU. To address the issue of control in simulation model-based methods, the algorithms by de Carpentier [46] and Vanek et al. [145] used a CAD modelling approach where the user can apply erosion using brushes on the terrain. The method by Vanek et al. [145] in particular allowed the user to edit large-scale terrains on the GPU using importance-based mip-maps. The editing includes brushing to apply erosion simulations, and is based on the erosion models used by Beneš and Forsbach [22] and Št'ava et al. [135].

More recent model-based methods include the method by Cordonnier et al. [36], which bases the simulation on the stream power equation [153]. This models the erosion as a combination of terrain uplift and hydraulic erosion, on a graph rather than a grid, allowing the algorithm to efficiently synthesize large-scale terrain covering tens of square kilometers. One of the most complex erosion simulation methods was created by Cordonnier et al. [38], who incorporated vegetation as part of the simulation of erosion, with a layered landscape model. Geomorphological events such as rainfall and running water, temperature variation, lightning, rock-slides, and wild fires were all considered influencing factors. While this is moving closer towards a more complete implementation of models for physical geography, it still neglects the many types and structures of material that ultimately make up terrain, such as complex strata with non-horizontal orientation. Another uplift and erosion method was created by Michel et al. [98], which created landscapes by simulating colliding plates to produce folded terrains. The erosion method by Št'ava et al. [135] is used to add additional small-scale details. Finally,

the most recent simulation approach I found was by Cordonnier et al. [37], which uses a recent geologically-inspired model for the Earth's crust, where uplift and erosion are jointly simulated to synthesize the terrain.

An intelligent agent system was created by Doran [51] as a method of simulation-based terrain synthesis that does not use physically based processes. The agents have a limited lifespan and their actions that modify the terrain are programmed by the user. This requires the user to have expertise and domain knowledge in order to program the agents to correctly shape the terrain.

Simulation methods, especially erosion method, often use models of terrain from the field of physical geography leading to more physically correct terrain. The downside is that they rely on simplified models which neglect many important aspects of terrain modeling, such as the variety of terrain materials, structures, and complex processes that are difficult to model, including wind weathering, glacial erosion, tectonic uplift, vegetation growth, and animal and human interference. Because the process of terrain creation cannot be simulated accurately down to the finest detail, generation methods use rules intended to make output look roughly realistic when assessed visually, usually by someone who is not an expert in geography. In some cases, this has led to methods modifying physically based algorithms for simulation in order to produce more visually appealing results instead of creating realistic terrain. The majority of methods also lack control where users are unable to define the output for specific points of interest. This means the user has to guess how to tune parameters to guide the synthesis towards an acceptable result or modify the final results manually which results in less realistic terrain overall. It is also worth noting that lack of user control is a challenge for simulation-based methods in many other domains of computer graphics and it is not simply unique to challenge of terrain synthesis.

Evaluation-Based Methods

Evaluation-based terrain synthesis methods are similar to simulation-based methods in that they use models from physical geography related disciplines, but the model is not operated over time. Instead of simulating the natural phenomena

through an iterative process, the terrain synthesized is a static evaluation of the user parameters in the context of a physical geographic system.

The earliest approach was Kelley et al. [82], who in 1988 used a physically-based approach in order to model rivers on terrain. The river network was modeled using empirical data of rivers and the terrain surrounding the network was filled out using fractal-based methods. Similarly, Teoh [138] created an approximation method combined with fractals to synthesize terrain. Terrain heights for mountain ridges are specified by the user and the rest is filled out using a combination of sinusoidal functions, linear interpolation and midpoint displacement. Rivers specified by the user are artificially meandered according to user parameters and deltas are created on the coastline by using a sweeping arc from the river. Coastline heights are determined by assigning whether they are beaches or cliffs on the idea that concave coastlines have little erosion and large sediment deposits to become beaches while convex coasts have greater erosion exposing more rock and become cliffs. An approach by Red Blob Games [122] sought to generate terrain represented by a grid of polygons created by relaxing a Voronoi diagram generated from random points. Elevations are assigned to each polygon vertex using a user defined function, rivers are randomly placed on vertices above sea level and flow downhill. Using the combination of elevation and humidity the method assigns terrain biomes to each polygon. This method works well for generating approximations to islands, but is not able to create realistic land-forms in general.

A more recent approach to approximating terrain was achieved by Genevaux et al. [65]. River networks are created using an elevation slope and river slope guide given by the user. Each river's flow and shape is determined by physical models in hydrology and the terrain is generated using a tree of terrain primitives inspired by the Blob-Tree model [155] and using Perlin noise [114]. The rivers are carved into the terrain using a profile determined by their Rosgen classification [123] to produce the final result. This produces terrain that is context sensitive, where features in the terrain conform to realistic principles. However, the terrain synthesis is only an approximation and like most simulation methods only generates terrain dominated by hydraulic processes. This method was improved upon by Zang et al. [159] who

incorporated the Tokunaga self-similar network [42], which aligns closer to real-world terrain morphology.

Evaluation methods use models from physical geography to create terrain without the need for simulation. The main problems stem from the homogeneity of the terrain, where evaluation methods fail to take into account features that cannot be created with a subset of models used for the generation process. Primarily these methods use hydraulic models to create terrain generated by fluvial erosion. This leads to situations where ad-hoc methods are created to emulate the visual appearance of more complicated terrain features (such as adding fractal detail), leading to visually plausible but not realistic terrain.

2.1.6 Data-Based Methods

Data-based terrain synthesis methods use existing terrain data to drive the synthesis of the terrain. The category covers a wide range of techniques that use existing terrain data in any form, including real-world height-maps. The work presented in this thesis builds on several of the principal data-based methods that are outlined below and discussed in more detail in Chapter 3. The texture-synthesis methods that some of these algorithms are based on are also discussed in the next Section (2.2).

Data-based methods have only been explored recently with respect to the history of terrain synthesis. This is due to elevation data becoming more available as a result of an increase in popularity of Geographic Information Systems (GIS) and advances of technology for scanning and storing large terrain data-sets [67]. Saunders [126] introduced the use of genetic algorithms for mutating and combining terrain data. The user supplies data of example terrain and paints the types of terrain they desire on a grid. Patches are extracted from the example data and are mutated and warped to fit onto the painted grid. While this is a novel approach to terrain synthesis, it fails to retain the realism of the supplied terrain. Patches are mutated and arranged in the synthesis without regard for maintaining the realism afforded by the supplied real-world data. An alternative method by Guerin et al. [70] trained a Conditional Generative Adversarial Network, by supplying it with real-world terrains and their

sketched counterparts. The method synthesizes terrain by taking a rough sketch of the main terrain features and synthesizing a height-map using the learned features from the training data. Failure cases can still occur when the user provides a sparse set of constraints, or sketches structures that are not present in the training set. While the method looks promising for the future of neural networks in terrain synthesis, it still fails to produce the same level of realism as the real-world terrain data it was trained with.

The data-based terrain synthesis method by Dachsbacher [43] uses non-parametric sampling, which is adapted from the pixel-based texture synthesis method by Efros and Leung [55]. For a given input that is partially filled with data, the algorithm grows outward from the set data, pixel-by-pixel, based on the data provided. While the authors state this technique can only be used to grow existing input data, fill gaps, or substitute parts of the terrain, it can principally synthesize new terrain from the given example data. A more successful adaptation of pixel-based texture synthesis methods was made by Gain et al. [61], which uses a modified pixel-based texture synthesis technique by Lefebvre and Hoppe [87]. The user defines point, curve, type, and copy-paste constraints that control the height and shape of the terrain in certain areas, and the method performs a parallel hierarchical texture synthesis with example height-map data to produce the final result. By utilizing the GPU the generation finishes in under 100 milliseconds which allows the user to make interactive edits and see the results in real time.

Brosz et al. [30] used example height-map data to synthesize fine detail for terrain. The user provides an example height-map with fine detail features and a coarse height-map with no fine detail as the target. Small-scale features are extracted from the example data by matching patches from the target height-map to the example height-map and taking the relative difference in elevation as the fine detail. This detail is then transferred to the target height-map by using image quilting [54], a patch-based texture-synthesis technique. Zhou et al. [160] used an improved method of image quilting by Wu and Yu [154] as a method of synthesizing terrain. The user supplies example data and a rough sketch of the ridges and valleys as the target output. Patches are extracted from the example height-map along the ridges and valleys, and are stitched together using Poisson image stitching [113] against

the corresponding ridges and valleys in the target. Patches are then placed to fill out the rest of the image in a raster scan order until there are no pixels unfilled. This work was later built upon by Tasse [137], who provided a more efficient algorithm for ridge identification, an alternative patch stitching method that uses Shepard interpolation [129] as well as a GPU implementation for finding the most candidate patches to place. This allowed the algorithm to run much faster and produce less artifacts between stitched patches. The patch-based terrain synthesis method by Cruz et al. [41, 40] looks at creating terrain with patches that contain large chunks for terrain with multiple ridges and valleys. This approach uses an acceleration structure where patches are described using a k-tuple, where each value is the average height of a subsection of the patch. Patch elevations are offset, when placed, to better fit neighbouring patch elevations. Overall, these patch-based methods have problems dealing with terrain features that cannot be identified by the ridgeline detection algorithm [33] and produce poor terrain for those styles as a result. Furthermore, patches are not placed considering the natural processes of underlying structure. This often leads to results that contain features like rivers that do not flow to the sea resulting in endorheic basins which, in the real world, are usually only found in desert environments [143].

It is also difficult for users to fully express ridge profiles when sketching with a top down view as the user can only express the height of a ridge using a gradient of colours. LandSketch [52] is a system developed by dos Passos and Igarashi which implements a first-person sketching system for patch-based terrain synthesis. The user draws curves of the ridge profiles from a first person view, which are then used to determine the patches to select from example data similar to the Zhou et al. [160] algorithm. As an alternative to techniques based on artificial intelligence and texture-synthesis, the method by Parberry [110] calculates an elevation histogram for existing terrain data and creates a height distribution table that converts the distribution of Perlin noise [114] to the same distribution as the input data. The result is slightly more realistic terrain than plain fractal height-maps, but the effect is relatively minimal and the resulting terrain is underwhelming compared to other data-based approaches.

A more recent data-based approach to terrain synthesis was investigated by Genevaux et al. [66] and then extended by Guerin et al. [69]. The terrain data the user provides is decomposed into a Sparse Construction Tree of elevation primitives based on the work in compressive sensing theory [31]. The primitives represent a function of height and the number of unique primitives is kept low making the tree sparse. To generate new terrain, the user provides a rough height-map of the terrain and the algorithm decomposes it into a sparse tree and iteratively increases the detail of the primitives and adds new primitives to increase the detail of the terrain. The downside to this technique is that the user is required to produce a coarse height-map, but in order to achieve the most realistic results, they must have domain knowledge to lay out this height-map realistically before starting the detail enhancing process. Additionally, the detail provided by the algorithm is not context sensitive; where you may have similar coarse structures in different parts of the height-map, the algorithm will provide the same fine detail for both structures when that may not be appropriate for enhancing the realism of the terrain. The work of Argudo et al. [4] address the later issue by combining the terrain primitives with additional contextual information such as mean elevation, slope, global vegetation density, solar irradiance, and upstream drainage area. This contextual information is used to enhance the selection of primitives to better match the context of the low-resolution input and gives greater control to the user. However, it does not address the reliance on a realistic initial input terrain created by a user with domain knowledge.

Data-based methods are a promising area of research to synthesize more realistic terrain. By using existing real-world data to create new terrain, the realism of the data provided can be retained to ensure that the output is equally realistic. While providing accurate data remains a fundamental requirement for data-based methods, the availability of data is growing constantly. For instance, 1-arc second digital elevation data is readily available for the majority of the Earth’s surface [7] with some areas scanned up to 1 meter resolution [95]. However, data-based methods are not without flaws. When synthesizing new terrain, retaining the original shape and structures of provided terrain is a major challenge. The original

data may be warped unnaturally, and the layout of the terrain may not reflect the underlying forces in the original data.

2.1.7 Summary

Overall, current terrain synthesis algorithms are not “ideal” because they lack sufficient realism, control, speed, or a combination of the three. In general approximation methods are efficient and many of the approaches have a great degree of control, they produce unrealistic results due to the inconsideration of geomorphology. Model-based methods are more realistic than approximation methods because they consider models developed in the field of geomorphology. However, the models used in the algorithm are too simple to compute completely realistic terrain, and it is difficult for an artist to control the synthesis towards the desired result. Data-based methods, specifically the texture-synthesis inspired algorithms, produce terrain that is locally more realistic than data-based methods and provide a great degree of control and speed, like approximation methods. But they are less realistic globally due to the same inconsideration of geomorphology that approximation methods have.

Data-based methods are the most promising area of research and previous methods show that a fundamental flaw, the inconsideration of geomorphology, is a topic that can be addressed to improve existing methods.

2.2 Texture Synthesis Background

2.2.1 Introduction

The main focus of this thesis is data-based terrain synthesis algorithms that build on methods from example-based texture synthesis. A background of the relevant texture synthesis methods are presented here. Following the state-of-the-art report by Wei et al. [148], I classify them as: patch-based, pixel-based, and texture optimization approaches.

Figure 2.1: A visualization of the patch-based texture synthesis method, image quilting. Patches are regularly placed with the boundary being defined by the minimum cut. (Image by Efros and Freeman [54, p. 342])

2.2.2 Patch-Based Methods

The patch-based terrain synthesis methods of Zhou et al. [160] and Tasse et al. [137] were an initial focus of my work and are discussed in Chapter 3. Both of these methods are based on algorithms from patch-based texture synthesis.

Patch-based texture synthesis methods synthesize a texture by incrementally adding patches from the example and blending or carving the overlapping regions to fit it into the synthesis as shown in Figure 2.1.

One of the first patch-based texture synthesis methods “image quilting” was created by Efros and Freeman [54]. The idea builds on Efros and Leung [55], where larger elements (patches) are used to build up the synthesis instead of pixels. The synthesis progresses in a raster scan order where patches are placed overlapping existing patches in the synthesis. Dynamic programming is used to calculate a minimum cost boundary cut to determine which pixels to keep from the placed patch and the existing synthesis. The patches to place are selected by first compiling a set of patches that minimize the SSD of the overlapping region, then randomly selecting the patch from the set.

Kwatra et al. [86] modified the approach of Efros and Freeman [54] to use graph cuts, modelling the patches as an undirected weighted graph connecting neighbouring pixels in each patch and overlapping pixels between the patches. The weight of each edge is the distance between the pixel values. At least one pixel in one patch connected to a source node and at least one pixel in the other patch is connected a sink node, with infinite weighted edges. A max-flow or min-cut [57] algorithm is used to find a least-cost seam between the patches instead of the dynamic programming approach by Efros and Freeman. Graph cuts are advantageous to the dynamic programming approach as it allows seams to be found in cases where dynamic programming would fail, such as when finding a seam on a patch that overlaps pixels along the entire border region. This improvement allows placement of patches in any order instead of just raster scan order, because it does not rely on the assumption of a border region that does no overlap existing pixels.

To improve the alignment and matching of strong features present in the example texture, Wu and Yu [154] looked at feature matching and deformation. Features, such as edges and ridges, are identified in the example texture using a modified two-pass classification similar to the Canny detector [32]. Patches are then selected based on how the features are aligned. Patches are optionally warped using thin-plate splines (TPS) [53, 28] or, in degenerate cases, by Shepard's method [129]. The warping is performed by constraining features from the patch to align with the corresponding features in the synthesis, and by constraining the bottom and right border pixels of the matching patch to be fixed. The latter prevents the accumulation of deformations from warping the patch too much.

2.2.3 Pixel-Based Methods

The state-of-the-art terrain synthesis method by Gain et al. [61] is a significant influence on my research. This method uses a pixel-based approach to synthesizing terrain.

Pixel-based texture synthesis methods synthesize a texture, pixel by pixel. A value for a synthesized pixel is selected from the example texture that minimizes the

difference between its neighbourhood in the example texture and the neighbourhood in the current synthesis. While there are many definitions of a texture in computer graphics, pixel-based methods use the Markov random field (MRF) theory which models the texture as a realization of a local and stationary process. Locality is a Markov assumption that the texture can be modeled as a probabilistic process in which all the interactions are local. A texture is local if each pixel is predictable from a small set of neighbouring pixels and is independent from the rest of the image. Stationarity is a mathematical property of a stochastic process when all the random variables of that process are distributed identically. A texture is stationary if, for a given window size, any two small subsections of a texture of that window size have the same variance (See Figure 2.2). Depending on the size of the window a texture may be viewed as either stationary or non-stationary. There is no set window size that works for determining stationarity of all textures, as the size of the window is dependent on the content of the texture.

Efros and Leung [55] created one of the first methods to synthesis textures using the MRF model. The center of the synthesis is initialized with a single pixel taken from the example texture. The synthesized region is grown outward by taking an un-synthesized pixel on the boarder of the region and selecting pixel from the example texture whose neighbourhood is most similar to the neighbourhood in the synthesis. The similarity is measured on the Gaussian-weighted Sum of Squared Difference (SSD) between the two neighbourhoods. This process is repeated until the entire texture is synthesized. The only user parameter is the neighbourhood size, which should be roughly equal to the texture element size - the size of the window for the example texture to be considered stationary. If the size selected is too large the output may produce an undesirable result with either many repeating elements or a pattern that is more regular than the example. If the size is too small the output may also produce an undesirable result with little or no structure.

Wei and Levoy [149] built on Efros and Leung's work by proposing a fixed neighbourhood search where the output was synthesized starting from the corner and proceeded in a scan-line fashion as shown in Figure 2.3. This scheme ensured that the number of synthesized pixels for neighbourhood matching was constant, creating a more even pattern distribution, as well as allowing the use of

Figure 2.2: A demonstration of definition of stationarity. (a) An image without the stationary property, as two movable windows at different positions (e.g. a_1 and a_2) are not perceptually similar. (b) An image with the stationary property, as two movable windows at different positions (e.g. b_1 and b_2) are perceived as similar. (Images by Wei and Levoy [149, p. 480])

Figure 2.3: A visualization of a scan-line pixel-based texture synthesis. Each pixel is selected based on the previously synthesized pixels. (Image by Wei and Levoy [149, p. 481])

acceleration methods such as: tree-structured vector quantization, kd -coherence, or k -coherence. Wei and Levoy use a non-Gaussian SSD to measure the similarity between neighbourhoods, but it is not made clear how this makes a difference to the synthesis. Textures can be synthesized in a multi-resolution fashion where the synthesis is performed at a lower resolution to guide the synthesis at higher resolutions. This allows small neighbourhoods to capture large texture elements. Wei and Levoy also introduce an order-independent synthesis scheme that allows the pixels in the synthesis to be computed in parallel.

Ashikhmin [6] explored the use of coherence to find matching neighbourhoods more efficiently. Previously, neighbourhoods in the example were found by randomly, or exhaustively searching the input example texture. Ashikhmin proposed that, because the neighbouring pixels in the example texture are likely to be selected as neighbours in the synthesized texture, coherence could be used to inform the selection of pixels in the synthesis. When selecting a value for an un-synthesized pixel, Ashikhmin's method takes the synthesized pixels in the synthesis neighbourhood and uses their original positions in the example (forward shifted) as candidates for the selection. PatchMatch [10, 11] is an algorithm for efficiently searching for matching neighbourhoods using a combination of random search and propagation. It relies on the concept of coherence in textures and is an important algorithm for many texture-synthesis methods to run efficiently.

Tong et al. [140], and Zelinka and Garland [156, 157] independently explored the idea of precomputed similarity sets. These sets can be used to ensure an efficient search for neighbourhoods similar to the synthesis neighbourhoods. Tong et al. created their candidate set with a k -coherence search, where the k closest matching neighbourhoods are precomputed for every neighbourhood in the example. The neighbourhood candidates for each value in the example can then simply be looked up in this set. Zelinka and Garland found that when computing more than one match per neighbourhood, matches for a given neighbourhood were clustered in the same region. To ensure diversity in matching, Zelinka and Garland required a minimum distance between all of the matches, implemented using Poisson disc sampling. The minimum distance is unspecified in the original paper.

Figure 2.4: A visualization of pixel-based texture synthesis method by Lefebvre and Hoppe [87]. The texture is synthesized from a low- to high-resolution using the process of up-sampling, jittering, and correcting. (Image by Lefebvre and Hoppe [87, p. 779])

Wei and Levoy [150] investigated an order-independent scheme that performs multiple synthesis passes, allowing values to be processed out of order without affecting the result of the synthesis. The method uses Gaussian pyramids for both the example and the synthesis as part of a multi-resolution approach. Lefebvre and Hoppe [87] built on this method by replacing the Gaussian pyramid with a Gaussian stack (each level has the same resolution) that provides a more diverse range of neighbourhoods for matching. The synthesis at the lowest level in the stack is initialized with random samples taken from the example at the same level. Up-sampling, jittering and correction is used until the synthesis reaches the top level in the stack, as shown in Figure 2.3. Up-sampling doubles the size of the image by expanding a single pixel into four pixels, and jittering reduces getting stuck in local minima adjusting the pixels by a small random value. The correction is the process of optimizing every pixel in synthesis by selecting a new pixel from the example whose neighbourhood best matches the current neighbourhood in the synthesis. The correction is performed in four sub-passes which operate on just one of the pixels of each 2×2 region. Notable precomputed k -coherence $k = 2$ uses a minimum 5% image size distance between neighbourhoods. Lefebvre and Hoppe use Principle Component Analysis (PCA) to reduce the number of values in the neighbourhoods that are compared when computing the SSD. This increases the efficiency of the algorithm without visibly degrading the synthesis results.

In a subsequent paper Lefebvre and Hoppe [88] introduced a new scheme that projects the entire example into an appearance-space, where each 5×5 neighbourhood the example is reduced through PCA to a small vector. During the correction phase, instead of projecting each synthesis neighbourhood into this PCA reduced space like their previous method, the example and synthesis neighbourhoods are compiled by looking up the corresponding values in this precomputed appearance-space.

Han et al. [72] introduces the idea of feature graphs with multiple examples and inconsistency correction. Feature graphs are directed graphs, where the nodes are the examples provided by the user and the edges are directed and contain the relationship of scale from one example to another. The synthesis uses the graph as a guide for sampling pixels from the example data and at what scale it should be applied. This allows the user to generate very detailed textures with only a few examples, or infinitely detailed textures using a cyclic graph. Inconsistency correction modifies the values of pixels taken from the example so that they better suit the neighbourhood after being selected as the best candidate during the correction process. The inconsistency correction is used to blend the values taken from different examples in the feature graph to produce a visually consistent result.

2.2.4 Texture-Optimization Methods

Texture-optimization texture synthesis techniques are explored in Chapter 5, as an alternative to pixel-based and patch-based methods, for applications in terrain synthesis.

Texture-optimization is a technique for texture manipulation that has application such as synthesis, reshuffling, hole-filling and interpolation (see Figure 2.5). A MRF-based similarity metric is defined for measuring the quality of the synthesis with respect to the given example. The process of synthesis is then formulated as a minimization of this metric. Typically this is performed using an Expectation Maximization (EM)-like algorithm where the algorithm iterates between, finding a correspondence between the synthesis and the example-texture (maximization M-step), and then optimizing the synthesis using the correspondence (expectation

Figure 2.5: A visualization of texture-optimization texture synthesis method by Kwatra et al. [85]. The algorithm alternates between selecting patches from the example that match the synthesis and using those patches to reconstruct the synthesis, until the synthesis converges. (Image by Kwatra et al. [85])

E-step). This approach refines the entire synthesis, as opposed to previous patch-based and some pixel-based texture-synthesis methods that progressively grow the synthesis.

Kwatra et al. [85] proposed a texture optimization algorithm as an alternative to pixel-based and patch-based methods. Starting with a randomized input, the synthesis is refined in an EM-like algorithm where pixel values are selected that minimize the energy of the local neighbourhood. Each iteration of optimization, the M-step selects neighbourhoods from the example that best match the neighbourhoods of a sparse grid of pixels in the current synthesis. The E-step averages the selected neighbourhoods to determine the new value of each pixel for the synthesis. This averaging can create blurred regions when selected neighbourhoods disagree on the pixel values where they overlap in the synthesis. Kwatra et al. also introduced many extensions including, a multi-scale approach where the neighbourhood size starts large and decreases after each iteration, and a multi-resolution approach where the synthesis is performed at a low-resolution, then is progressively up-sampled and optimized until the final size is reached.

Han et al. [73] proposed a texture optimization that uses discrete values from the example provided instead of averaging values. A k -coherence table is pre-calculated for the example, to find k neighbourhoods for each coordinate that best match the coordinate's surrounding neighbourhood. For the synthesis, instead of just storing pixel values it also stores coordinates into the example from which that pixel value was taken. The M-step is modified so that only a consistently small subset of neighbours are evaluated each iteration. This subset is compiled by taking each neighbour in the synthesis, and using the coordinate values to look up the k -nearest neighbours in the precomputed k -coherence table. The E-step is modified to directly copy the pixel value from the center of the best neighbourhood and the coordinate for where it appears in the example. This algorithm is parallelizable, as both the E-step and M-step are able to be computed independently for each value in the synthesis of each iteration. Because the method discretely selects pixel values instead of the minimizing the energy using least-squares, it has more in common with pixel-based methods like Lefebvre and Hoppe [87] than with other texture optimization methods.

Barnes et al. created an algorithm called PatchMatch [10] that improves the efficiency of finding a correspondence between images based on neighbourhood similarity. With PatchMatch they devised a texture optimization algorithm that runs at interactive rates. It uses a bi-directional similarity metric by Denis et al. [130] for the texture energy function, which creates a more even distribution of visual information synthesized from the example image. The bi-directional similarity metric consists of two terms, completeness and coherence. The completeness term ensures the synthesis contains as much information from the example as possible, and the coherence term ensures the synthesis is consistent with respect to the example, penalizing inconsistent artifacts. The coherence term is identical to the texture energy function of Kwatra et al. [85]. Barnes et al. generalized the PatchMatch algorithm in a subsequent paper [11] to incorporate a k -nearest-neighbour algorithm, to search across scale and rotations in addition to translations. The k -nearest-neighbour algorithm was significant in improving the efficiency of pixel-based methods for precomputing k -coherence tables, and the

scale and rotation search helped decrease the total texture energy of synthesis results due to the ability to find closer matches during the optimization process.

Image melding by Darabi et al. [44] used Poisson image editing to merge images, instead of least squares, during the E-step for calculating the values for the synthesis. First the gradients are calculated and then the values are derived from the gradients. An important note is that the images used are photographs and stored in the Lab colour space, allowing for gain and bias adjustments for small changes in luminosity from HaCohen et al. [71]. While the Poisson approach for optimizing pixel values works well for photos with many soft and blurry features, it is not suitable for texture synthesis due to the blurriness this introduces.

Kaspar et al. [80] provided a texture synthesis method for structured texture synthesis. Instead of the bi-directional similarity metric used by PatchMatch, the method uses a spatial uniformity metric and histogram enforcement. The spatial uniformity metric employs an occurrence map that keeps track of the number of times an element from the example-map is used during synthesis. The patch-distance is then modified to encourage a uniform occurrence map, ensuring a more even distribution of patches from the example being used in the synthesis. The histogram enforcement is used in addition to the spatial uniformity metric, to weight the contributions of patches, so the distribution of colours in the synthesis is more like the distribution of colours in the example.

2.2.5 Summary

Patch-based methods were one of the earliest example-based texture synthesis methods and are relatively simple to implement. However, little research has been done on them in recent years due to them being outperformed in both quality and speed by the state-of-the-art pixel-based and texture-optimization methods. In addition, a subjective study of texture synthesis methods by Kolář et al. [83] rated the texture-optimization method by Kaspar et al [80] to have the best overall quality for synthesizing a wide range of textures.

Pixel-based and texture-optimization methods share similar metrics for optimization, with the primary difference being that pixel-based methods

synthesize discrete pixel values from the example and texture optimization methods synthesize new values that minimize the overall energy with respect to the example. The similarity between the methods suggests that improvements for either type of algorithm are likely to be similar, as seen with type constraints (e.g. multi-example by Han et al. [72] and guided image completion by Barnes et al. [10]) and multi-resolution synthesis (e.g. Wei and Levoy [150] and Kwatra et al. [85]).

2.3 Conclusion

Current terrain synthesis methods are not ideal as defined in section 2.1.3. Approximation methods have little to no consideration for real-world terrain shaping processes and model-based methods rely on models that are too simplified to produce realistic terrain. Data-based methods are the most promising area of research because the realism is drawn from real-world data and there are areas for improvement that incorporate the consideration of geomorphology which would address its greatest weakness. This is addressed further in Chapter 4.

The data-based texture synthesis methods by Zhou et al. [160] and Tasse et al. [137] are based primarily on patch-based method of Wu and Yu [154] and the method by Gain et al. [61] is based primarily on pixel-based methods of Lefebvre and Hoppe [88] and Han et al. [72]. As far as I am aware, there are no methods that have approached terrain synthesis using an approach similar to texture-optimization. With the success of state-of-the-art pixel-based texture synthesis methods for synthesizing terrain Gain et al. it follows that texture optimization is also worth exploring, which I do in Chapter 5.

Chapter 3

Flaws of Data-Based Terrain Synthesis

Data-based methods, as introduced in Chapter 2, use real-world data to drive the synthesis of terrain. Real-world data is readily available in many formats, such as world-wide raster elevation data from sources like the Shuttle Radar Topography Mission [144]. Real-world data is guaranteed to be realistic (simply due to its existence), and the assumption is that by using realistic data to drive the synthesis of the terrain with a data-based method, the synthesized terrain will be more realistic.

The majority of data-based methods reorganize elements of the given data, like a jigsaw puzzle, to satisfy the user constraints. The output should satisfy the user constraints as much as possible, while subject to remaining realistic. As there is no measure for realism, as discussed in Section 2.1.2, the methods attempt to achieve realism for the synthesized terrain by minimizing the deformation of the input elements. The less deformation applied during the process, the more the output terrain resembles the input and the more realistic the output terrain is. However, to be used practically, data-based methods must strike a balance between maintaining the realism of the original data and satisfying the user constraints. There are currently no data-based terrain synthesis methods that can synthesis terrain, where the terrain is as real as possible, while meeting user constraints. The current methods have many flaws that are discussed in this chapter.

This chapter discusses the state-of-the-art data-based methods that are based on texture synthesis techniques (see Section 2.2) and their flaws as terrain synthesis methods. Section 3.1 presents an in-depth explanation of the relevant existing data-based terrain synthesis algorithms, with a brief summary of the flaws that I have identified. Section 3.2 gives a detailed exploration of the flaws and the general problems that need to be addressed when developing data-based terrain synthesis algorithms.

3.1 Existing Algorithms

3.1.1 Zhou et al. 2007

Overview: Zhou et al.'s [160] algorithm (referred to as Zhou) uses a modified image quilting approach [54], based on the texture synthesis method of Wu and Yu [154]. The algorithm takes patches of terrain from the example data that match the ridges and valleys of the user sketch and stitches them together. Empty areas of the sketch are then incrementally filled in with patches from the example-map. A visualization of the algorithm is shown in Figure 3.1.

Technical: The user provides, as input, an example terrain height-map (example-map) and a sketch of the desired height-map (sketch-map). The example-map is the input data that typically consists of real-world terrain. The sketch-map is a simple height-map with raised areas to indicated ridges and lower areas to indicate valleys, but it is not required to have the same elevation range as the example-map.

First, Zhou identifies the ridges and valleys (features) of the inputs. The Profile recognition and Polygon breaking Algorithm (PPA) [33] is used to create a non-cyclic feature-graph of the inputs. The edges of the graph are the curvilinear ridges or valley segments and the nodes are the end-points (degree 1) and branch-points (degree 3 or more) connecting the ridge or valley segments. There are no nodes of degree 2 as these are incorporated as part of the curvilinear edge. Separate feature-graphs are created for both the example-map and the sketch-map.

Figure 3.1: An example of the Zhou et al. algorithm. Feature-graphs are created for both the example-map and sketch-map. Feature-patches are extracted from the example-map using the feature-graph and are placed on the synthesis according to the sketch-map's feature-graph. Finally, patches from the example-map are used to fill in the rest of the synthesis.

Figure 3.2: Examples of feature-patch classifications. (a) Path-feature, with three control points, (b) End-feature, with two control points. (c) Branch-feature, with four control points. (d) An example of control points being determined by where the edges of the feature-graph intersect the inscribed circle. (e) An ambiguous case where there is no intersection between the edge of the feature-graph and the inscribed circle. (f) An ambiguous case where outside edges intersect the inscribed circle.

The user specifies if the feature-graphs are created for ridges or valleys, or for both (two feature-graphs, one for ridges and one for valleys).

Second, feature-patches are extracted from the example-map using its feature-graph created in the first step. A feature-patch is a small height-map and a set of control points. The size of the feature-patch is determined by the spatial scale of the terrain (Zhou et al. use 80×80 pixels in their examples). Each feature-patch is classified as one of the following: a *path-feature*, an *end-feature*, or a *branch-feature*¹, based on the number of control points in the feature-patch (an example of each is shown in Figure 3.2). Path-features (3 control points) are created by traversing the curvilinear edges of the feature-graph in steps that are one-half of the patch size, and extracting the patch centered at that position from the example-map. End-features (2 control points) and branch-features (> 3 control points) are created by extracting the patch centered at end-point and branch-point nodes of the feature-graph respectively. The control points for all feature-patches include the center point of the patch and all the positions where the outgoing graph edges intersect the inscribed circle of the patch, as shown in Figure 3.2d. Zhou et al. does not explain how to handle the case when the edge does not intersect the circle as shown in Figure 3.2e, or when ridges from separate parts of the feature-graph

¹Zhou et al. refers to the feature-patch classifications as path feature, end point, and branch point respectively. I have renamed them for the purpose of providing a clear differentiation between the node in the feature-graph and the feature-patch that was extracted.

intersect the circle as shown in Figure 3.2f, making it an ambiguous detail of the algorithm. The process of creating feature-patches is performed separately for the ridges and valleys with their respective feature-graphs. The result is a set of feature-patches for ridges and/or valleys.

Third, the feature-patches created in the second step are placed on the synthesis, guided by the feature-graph of the sketch-map and previously placed patches. The feature-graph for the sketch-map is traversed by selecting a node (preferably a branch-point) and performing breadth-first traversal of the graph, selecting a feature-patch and then placing it on the synthesis. Edges are traversed by stepping along the curvilinear edges of the feature-graph in steps that are one-half of the patch size, and at each point the set of path-features are used as candidates for selection. For the end-points and branch-point nodes, the end-features and branch-features (preferably of the same degree) are used as candidates for selection. Thin-plate splines (TPSs) [28] are used to warp branch-feature and path-feature candidates to better match the feature-graph of the sketch-map before further selection. This warping aligns the outgoing edges of patches to produce a consistent ridge or valley between neighbouring placed patches. A pair of TPSs are created for a 2-dimensional transformation that maps the control points in the feature-patches to the corresponding control points from the feature-graph at the point of placement. As there is more than one possible alignment for mapping the corresponding control points of the outgoing edges for the patches, different alignments of the mapping between control points are considered as different candidates. For branch-features, only the best k candidates with the lowest TPS deformation energy are selected for further matching. Zhou et al. do not mention how large k should be. End-features only require a simple rotation to be aligned with the feature-graph for placement, so no TPSs are used to warp the patch.

Forth, for each candidate feature-patch, an image graph cut [86] is calculated for the overlapping regions between the candidate and the previously placed patches in the synthesis. Additionally, a feature dissimilarity cost is computed by comparing the uniformly sampled height profile (perpendicular to the feature-graph edge), of the candidate feature-patch and the existing neighbouring patches, at the position the patches join. The candidates are then scored according

Figure 3.3: A combination of a graph cut and Poisson image editing is used to solve the for the elevation values in the overlapping region between patches. (a) The overlapping region Ω lies between the patches A and B , and is the region of elevations being determined. $\delta\Omega$ indicates the location of values used as Dirichlet boundary conditions for the Poisson equation. The dotted line shows the graph cut between the two patches. (b) ΔA and ΔB are the gradient values for the patches A and B respectively. Gradient values along the seam, shown as a black line, are artificially set to zero.

to a weighted combination of the deformation energy of the TPS warping, the graph cut score, and the feature dissimilarity cost. The candidate with the lowest score is selected and placed on the synthesis using the previously calculated graph cut. The seam between the placed patch and the synthesis is removed using Poisson image editing [113]. For the purposes of the Poisson equation, the gradient is calculated for the overlapping region with the gradient values of the seam artificially set to zero and the neighbouring pixels of synthesis used as the Dirichlet boundary condition. The Poisson equation is solved for the overlapping region for the final result of the patch placement (as shown in Figure 3.3). After performing the selection and placement of feature-patches, the result is a partially synthesized terrain with empty areas that are not covered by the feature-graph of the sketch-map.

Fifth, the empty regions of the synthesis from the third step are filled in by using non-feature-patches from the example-map that best fit the existing synthesis. Patch positions at coarsely-spaced intervals are selected for placement. Zhou et al. suggest using intervals of 100 pixels vertically and horizontally, but it is ambiguous

if the size of the non-feature-patches are the same size as the feature-patches. These regions are ranked in descending order by the number of synthesized pixels they contain. Completely synthesized regions are not considered for patch placement. For each region in order, a set of k candidate patches from the example-map are selected that minimize the Sum of Squared Difference (SSD) between the candidate patch and the synthesized pixels in the region. Each patch in the k candidate set is then scored based on a weighted combination of the SSD score and the graph cut cost for merging. The patch with the lowest score is selected and placed into the synthesis using the graph cut and Poisson image editing. After this process, all the empty regions of the synthesis will have been filled and the final synthesis of the terrain is produced.

Flaws: Zhou’s algorithm has several flaws in its design and there are many ambiguous details in the original description. It is important to note that Tasse et al. [137] are the only researchers that have built their algorithm on the work by Zhou et al., and they have stated that they were unable to implement the algorithm of Zhou as described in the paper, due to some of the flaws and ambiguities I have identified². These flaws, some of which are shared with other data-based methods are discussed in Section 3.2:

- The parameters of the algorithm are input dependent, are interdependent, and the effects of changes to the parameters are not made explicit. This is discussed in Section 3.2.1.
- Terrain feature identification for ridges and valleys is performed through the PPA which is not a reliable algorithm to select features for the purposes of terrain synthesis. This is discussed in Section 3.2.3.
- Thin-plate splines are used to deform patches during feature-patch placement, but the process fails in a large percentage of cases. This is discussed in Section 3.2.4.

²This has been confirmed through personal correspondence with Flora Tasse.

- Poisson seam removal modifies the patches of terrain drastically reducing the naturalness of the original patch. It also produces seam artifacts and is discussed in Section 3.2.5.
- If the user provides an example for which features cannot be identified using the PPA, the algorithm will fail to generate a reasonable result. This limitation when interpreting the input data is discussed in Section 3.2.6.
- There is no process for ensuring the terrain is natural, relying on the user's expertise to guide the synthesis. The reliance on user expertise is discussed in Section 3.2.7.

3.1.2 Tasse et al. 2012

Overview: Tasse et al.'s algorithm [137] (referred to as Tasse) expands upon terrain synthesis algorithm presented by Zhou et al. [160]. It introduces an alternative approach for selecting and merging patches and incorporates the algorithm as part of a larger framework that aids the user in creating constraints.

Technical: The user provides, as input, an example terrain height-map (example-map) and an estimation of the desired height-map (target-map). The example-map is the input data that typically consists of real-world terrain. The target-map is a height-map that can be provided directly or created through a terrain sketching interface [60]. In the sketching interface, the user draws $2\frac{1}{2}$ -dimensional silhouettes and boundary curves for ridges and valleys, and the target-map is generated using multi-resolution surface deformation and wavelet noise. The target-map differs from the sketch-map in Zhou because the magnitude of the elevation values in the target-map are instrumental in guiding patch matching and the final deformation.

First, Tasse closely follows Zhou by generating a feature-graph for the example-map, and the target-map, for ridges and/or valleys using the PPA. However, the PPA is optimized for efficiency, where the polygon breaking portion of the algorithm is replaced with a minimum spanning forest algorithm as in

Bangay et al. [8]. This change greatly speeds up the process of calculating the feature-graph while producing equivalent results.

Second, Tasse extracts feature-patches from the example-map using its feature-graph created in the first step. This process differs slightly from Zhou. A feature-patch is a small height-map (with a default size of 80×80 pixels) and a set of control points. Feature-patches are extracted from steps along the curvilinear edges and centered at nodes of the feature-graph. The control points for a feature-patch include the center point of the patch and all the positions where the outgoing graph edges intersect the inscribed circle of the patch. This process is done separately for the ridges and valleys with their respective feature-graphs. Unlike Zhou, Tasse does not have a classification for the patches based on the number of control points they contain. The set of feature-patches is then expanded by creating mirrored and rotated (in increments of $\frac{\pi}{4}$) versions from the initial set feature-patches.

Third, feature-patches created in the second step are placed on the synthesis, guided by the feature-graph of the target-map and previously placed patches. This step differs greatly from Zhou. The feature-graph for the target-map is traversed by selecting a node and performing breadth-first traversal, selecting a feature-patch and then placing it on the synthesis. Edges are traversed by stepping along the curvilinear edges of the feature-graph in steps that are one-half of the patch size. At each point, a target patch is extracted from the target-map in the same way feature-patches are extracted in the second step. All feature-patches are used as candidates and are scored for selection. The score of each feature-patch for the target patch is the weighted sum of the feature dissimilarity, angle difference, noise variance, and overlap area costs. The feature dissimilarity cost is computed by the Normalized Sum of Squared Difference (NSSD)³ of the uniformly sampled height profile along the outgoing edges. If the candidate and the target patch have two outgoing edges, the NSSD between the height profiles orthogonal to the outgoing edges is also added to this cost. The angle difference cost is calculated by the NSSD of the angles of the outgoing paths. The noise variance cost is the NSSD of the

³NSSD refers to the SSD divided by the number of values. This was confirmed by examining the source code provided by Tasse et al.

Gaussian noise variances of the feature-patch and the target patch, computed at multiple resolutions. The overlap area cost is the NSSD of the overlapping region between the feature-patch and the synthesized pixels. In addition, feature-patches that do not have the same number of control points as the target patch are given a large penalty, but the paper does not make it clear how the angle feature dissimilarity and angle difference are calculated in these cases. A set of k candidates with the lowest score are taken, and a graph cut is calculated against the existing synthesized pixels. The feature-patch from this set with the lowest graph cut score is selected and placed onto the synthesis using the graph cut. The seam between the placed patch and the synthesis is removed using a combination of Shepard interpolation [129] and Poisson image editing [113]. When removing this seam, instead of using Zhou’s process of zeroing the gradient values before performing the Poisson seam removal, Shepard interpolation [129] is used to alter the gradient values of the new pixels, smoothing the transition across the seam in the gradient domain. After performing the selection and placement of feature-patches, the result is a partially synthesized terrain with empty areas that are not covered by the feature-graph.

Fourth, non-feature-patches are used to fill in the empty areas of the synthesis from the third step. Non-feature-patches are patch-sized height-maps extracted from the example-map that do not contain ridges or valleys as identified by the PPA⁴. The order of positions for non-feature-patch placement is determined by a modified onion-peel approach similar to the one presented by Criminisi et al. [39]. Positions along the boundary of synthesized and empty pixels are scored based on a sum of synthesized pixels in the region, weighted by the order in which they were synthesized. Positions are selected in descending order for non-feature-patch placement, and the process repeats until all empty areas are synthesized. Patch candidates for non-feature-patch placement are scored on the weighted sum of noise variance and the overlap area costs, with separate weights to those used for feature-patch placement. A set of k candidates with the lowest score are taken from that set and the candidate with the lowest graph cut score is selected to be placed

⁴Creating a set of non-feature-patches is not mentioned in the paper but has been confirmed as the intention of the algorithm through personal correspondence with Flora Tasse. The details of this process are ambiguous.

on the synthesis. The placement is performed in the same manner as feature-patch placement, using a graph cut to place pixels and then Poisson image editing with Shepard interpolation to remove the seam.

Finally, if the target-map was created using the sketching interface, the terrain synthesized in the fourth step is deformed to fit the exact elevations provided in the initial user sketch. This deformation ensures that the elevation constraints specified by the user in the form of the sketch are satisfied. The deformation is performed using a multi-resolution surface deformation.

Critique: Tasse et al. addresses some of the flaws of Zhou et al., including using rotated and flipped patches to avoid the extreme deformations caused by the thin-plate spline, and the use of Shepard interpolation to correct some artifacts during the seam removal process. However, they fail to address some of the more fundamental flaws:

- The issue of parameters is not addressed and the increased number intensifies the flaw. This is discussed in Section 3.2.1.
- Tasse et al. still relies on the use of PPA which does not robustly identify the features in the example height-map. This is discussed in Section 3.2.3.
- The use of Shepard interpolation helps remove artifacts during the seam removal between patches, but the process still greatly modifies the terrain creating unnatural shapes. The final framework also performs a large-scale deformation to warp the terrain to the user sketch which further leads to more unrealistic terrain. This is discussed in Section 3.2.5.
- The use of rotated and flipped patches allows greater use of the data in the example-data than Zhou’s method. However, Tasse still relies on feature identification through PPA and there is no consideration to changing the elevation of a patch to allow features to be located at different elevations. This is a limitation of the use of data and is discussed in Section 3.2.6.

- While the sketch interface provides the user greater control over the shape of the terrain, the method still requires user expertise to synthesize natural terrain. This is discussed in Section 3.2.7.

3.1.3 Gain et al. 2015

Overview: Gain et al. [61] adapted and built on the existing pixel-based texture synthesis methods of Lefebvre and Hoppe [87, 88], and Han et al. [72]. The method starts with a low-resolution synthesis and incrementally up-scales the synthesis while performing corrections to ensure that the pixels in the synthesis, and their neighbourhoods, match the example data and also satisfy the user constraints. A visualization of the algorithm is shown in Figure 3.4.

Technical: The user provides, as input, an example terrain height-map E (example-map) and a set of synthesis constraints. The example-map is a 2D array of elevations and can be either a single example height-map, or a texture atlas of height-maps where individual height-maps have an identifying type. The synthesis constraints include point, curve, type, and copy-paste constraints. Point constraints have a position, elevation, gradient, and radius of effect. Curve constraints are specified as a list of point constraints with separate gradient and radius of effect attributes for the right and left side of the curve. Type constraints are a set of terrain types that can be used for a region of the synthesis and can only be used if the user provides a texture atlas of example-maps as input. Copy-paste constraints are coordinates from the example-map to be used at a set location throughout the synthesis.

First, Gain generates a Gaussian image pyramid of depth $L + 1$ from the example-map E^i , where i is level in the pyramid and E^0 is the highest resolution. In contrast to the Gaussian image stack used by Lefebvre et al. [87], this reduces memory requirements during the synthesis. Gain then generates an appearance-space A^i associated with each level of the example-map pyramid E^i , where each value in the appearance-space is an appearance-vector that describes the region around an elevation in the corresponding example-map. To create the

Figure 3.4: An example of the Gain et al. algorithm with $M = 7$ and $L = 5$. The top three rows of images visualize the image pyramids for the appearance-space, look-up table, and example-map. The bottom two rows visualize the image pyramids for the offset-map and synthesis. The images at each level i are twice the resolution of the previous level $i + 1$, but are not shown to scale.

Figure 3.5: An example of the appearance-space, created from a given example-map. The appearance-space is visualized as four separate images, one for each channel of the appearance-vector.

appearance-space, Gain creates a set of Gaussian weighted 5×5 neighbourhoods, centered on each elevation in the example-map, to produce a 25D neighbourhood-vector for each elevation. Principle Component Analysis (PCA) is used on this set of neighbourhood-vectors to reduce the dimensions to four significant components (see Figure 3.5). Gain then artificially sets the first component of the reduced vectors to the mean of their original Gaussian weighted 5×5 neighbourhood in the example-map, which produces the vectors (appearance-vectors) of the appearance-space. The use of PCA for describing neighbourhoods in an appearance-space was introduced by Lefebvre and Hoppe [88] to reduce computation when comparing neighbourhoods, but the addition of setting the first component to the mean is a modification required for using height offsets, inspired by Han et al. [72]. For each appearance-space, Gain computes a k -coherence ($k = 2$) look-up table C^i of 5×5 regions that are most similar using the PatchMatch algorithm [10]. This look-up table produces a mapping from a coordinate to a set of coordinates, and is used later during the correction phase of the algorithm.

Second, Gain creates image pyramids of depth $M + 1$ for the synthesis S^i and offset-map V^i for each level i , where $M > L$ from the first step (the value of M is

unspecified). Each synthesis S^i in the pyramid is a 2D array of floating-point coordinates that sample the example-map of the corresponding level i . Each offset-map is a 2D array of elevation offset values and is the same size as the synthesis at the same level in the pyramid. The synthesis S^M is initialized with coordinates at the center of the corresponding example-map, as if the example-map pyramid were extended to a depth of M . At the same level the offset-map V^M is initialized with zeros.

Third, starting from level M with the pyramids from the second step, Gain creates a progressively higher resolution synthesis and offset-map. At each level, Gain performs the operation of up-sampling, jittering, and correcting to produce the next level in the pyramid. Up-sampling doubles the resolution for both the synthesis and offset-map and jittering involves taking each coordinate in the synthesis and shifting it by a small magnitude in a random direction. Values in the new synthesis S^i and offset-map $V^i(p)$ for some level i are calculated as

$$\begin{aligned} S^i(p) &= 2S^{i+1} \left(\left\lfloor \frac{p}{2} \right\rfloor \right) + \left(p - 2 \left\lfloor \frac{p}{2} \right\rfloor \right) + rH(p) \\ V^i(p) &= V^{i+1} \left(\left\lfloor \frac{p}{2} \right\rfloor \right), \end{aligned}$$

where $p \in S^i, V^i$, r is the magnitude of the jittering (usually $r = 0.4$), and H is a deterministic function that maps coordinates to pseudo-random offsets in $[-1, 1]^2$. The jittering produces coordinate variation which reduces repetition and helps the correction phase produce more globally optimal results (refer back to Figure 2.4).

The correction phase adjusts the synthesis and height-offset values to fit more coherently with their neighbours, and is performed when operating on pyramid levels $i \leq L$. For every target coordinate p in the synthesis, Gain computes a target neighbourhood N_S of four appearance-vectors by taking a combination of the diagonal values in the synthesis S^i and the values adjacent to the diagonal values, and using their coordinates to index into the appearance-space A^i . Gain then creates a set of replacement candidates for the target neighbourhood N_S by taking the coordinates of the 3×3 region around the target coordinate p and using them to look-up replacement candidates with similar appearance-vectors in the pre-computed look-up table C^i . The candidate neighbourhood N_E of four

appearance-vectors is computed for each replacement candidate q by taking the values of its diagonal neighbours in the appearance-space A^i . A height-offset value h is then calculated to minimize the SSD between the first components of the target neighbourhood N_S and the candidate neighbourhood N_E . h is then added to the first components of candidate neighbourhood N_E . The calculation of this height-offset is similar to the transfer function in Han et al.'s method [72]. Each replacement candidate is then scored on a recalculated SSD between the modified candidate neighbourhood and the target neighbourhood. The candidate with the lowest score is selected as the replacement for the target coordinate in the synthesis. The value in the synthesis is set to the candidate coordinate ($S^i(p) = q$) and the value in the offset-map is set to the corresponding height-offset value calculated for that candidate ($V^i(p) = h$). The correction phase is repeated twice per pyramid level.

Point and curve constraints adjust the height-offset values in the offset-map after the correction phase. The current elevation for each coordinate in the synthesis is determined by sampling the example-map using the coordinate and adding the corresponding height-offset. For point constraints, an adjustment is calculated to match the current elevation to the constraint elevation based on a piece-wise non-linear blending function. This blending function calculates smaller adjustment values for positions further away from the constraint and current elevations that are closer to the constraint elevation. The adjustment is then added to the height-offset in the offset-map. Instead of calculating the effect of multiple constraints on a pixel individually, the effect is calculated by weighting the parameters of the adjustment calculation to be inversely proportional to the constraint distance to the target pixel. This means the adjustment only needs to be calculated once per coordinate, as opposed to once per constraint per coordinate. Curve constraints extend the operation of the point constraints where the values for calculating the adjustment are interpolated along the curve using Hermite curves. Type constraints limit the set of possible candidates during the correction phase and can only be used when the user provides a texture atlas as input for the example-map. Constrained coordinates have a set of types from the example-map atlas. When compiling the set of replacement candidates during the correction

phase, any candidates whose type does not match any of the specified constraint types are rejected from consideration. If all the candidates are rejected, nine new candidates are selected at random that match the constraint types. Unconstrained pixels are free to match any type of terrain. Copy-paste constraints simply set coordinate values in the synthesis that cannot be changed. It essentially allows the user to copy parts of the example-map that will remain in the output after the synthesis process. A multi-resolution approach is taken for constraints, where higher the resolution of the synthesis, the more locally the constraints are enforced. Both height (point and curve) and type constraints are not enforced at all for the last two levels of up-sampling and correction to avoid interfering with the fine level of detail that is created through the correction process.

Finally, once the highest resolution of the synthesis is reached, the final height-map is reconstructed by using the coordinates in the synthesis to sample the elevations from the example-map and adding the height-offsets from the offset-map.

Critique: Gain et al. has adapted pixel-based texture synthesis methods expertly to the field of terrain synthesis. The method produces results in real-time for a variety of terrain types, but fails to address several of the issues regarding the differences between terrain and textures, resulting in some fundamental flaws that need to be addressed:

- The height-offset used to improve the matching between neighbourhoods distorts the original elevations, leading to unnatural terrain in some cases. The drawbacks of interpolation are discussed in Section [3.2.5](#).
- The limitation imposed by the PCA restricts the neighbourhoods to being strictly oriented with the original example-map, whereas Tasse is able to use flipped and rotated patches from the source data. Not using the data to its full extent is discussed in Section [3.2.6](#).
- Gain relies heavily on the user providing many carefully placed constraints to guide the synthesis toward a natural result, which may not be possible, depending on the user's expertise. This issue is discussed in Section [3.2.7](#).

3.2 Flaws

Zhou, Tasse, and Gain described in Section 3.1 are the state-of-the-art data-based methods. They are able to produce terrain that is more realistic than the majority of approximation and model-based approaches, but they all suffer from flaws that reduce the realism of the output. These flaws are discussed in this section.

3.2.1 Parameterization

Zhou and Tasse both require the user to specify a large number of parameters to synthesize terrain. The primary issue is that these parameters have complex interactions with each other making it difficult to tune one parameter at a time. The parameters are also dependent on the height-maps used as inputs, so one set of parameters is not optimal for all inputs. This leads to a trial and error approach to selecting the parameters and owing to the complexity of the interaction it is impossible to find the ideal setting which leads the user to settle on a terrain output that is less realistic than one generated with the ideal parameters.

If the parameters used for the PPA are included (assuming that the same parameters are for both the example-map and sketch-map/target-map) and any normalization factors are assumed that to be included in their respective weights, then Zhou has a total of 13 tunable parameters and Tasse has a total of 15, as listed in Tables 3.1 and 3.2. Due to the ambiguous description of the algorithm by Zhou et al., the actual parameters used in the original implementation of Zhou may differ. The parameters used in Tasse were confirmed by inspecting the source code provided by the authors. The parameters for PPA are discussed in Section 3.2.3 and the parameter for patch size is discussed in Section 3.2.2.

To discuss the problem of complex parameterization I will use the weighting of feature-patches, in both Zhou and Tasse, as an example. When these parameters are altered, the effect that it has on the output terrain is not clear until the terrain has been synthesized. As a sum of weighted costs is used to select candidate feature-patches, a change to a single weight affects the relative influence of the

X	Name	Description
-	PPA grid interval	Space between samples on the grid, for the PPA.
-	PPA profile length	Length of sample profile, used to identify ridges or valleys.
-	PPA features type	Whether the PPA is run for ridges, valleys, or both.
k	Candidate set size	Number of branch-features with the lowest deformation energy, or number of non-feature-patches with the lowest overlap SSD, selected before comparing the total costs.
-	Feature-patch size	Size of feature-patches.
α_d	Feature deformation weight	Weight for TPS deformation cost for feature-patches.
α_g	Feature-graphcut weight	Weight for graphcut cost for feature-patches.
α_f	Feature profile weight	Weight for feature profile difference for feature-patches.
-	Feature profile count	Length of the feature profile for feature-patches.
-	Non-feature-patch size*	Size of non-feature-patches
-	Non-feature position increment	Spacing between the placement of non-feature-patches.
-	Non-feature overlap weight*	Weight for overlap cost for non-feature-patches
-	Non-feature-graphcut weight*	Weight for graphcut cost for non-feature-patches.

Table 3.1: The symbols, names, and descriptions of the parameters used in Zhou. Ambiguous parameters have been marked with a (*).

X	Name	Description
-	PPA grid interval	Space between samples on the grid, for the PPA.
-	PPA profile length	Length of sample profile, used to identify ridges or valleys.
-	PPA features type	Whether the PPA is run for ridges, valleys, or both.
k	Candidate set size	Number of feature and non-feature-patches with the lowest total cost, selected before comparing the total graphcut scores.
-	Feature-patch size	Size of feature-patches.
α_f	Feature dissimilarity weight	Weight for dissimilarity between outgoing edges of feature-patches.
α_a	Feature angle difference weight	Weight for angle difference cost for feature-patches.
α_n	Feature noise variance weight	Weight for noise variance cost for feature-patches.
α_o	Feature overlap weight	Weight for feature overlap cost for feature-patches.
-	Feature profile count	Length of the feature profile for feature-patches.
-	Non-feature-patch size	Size of non-feature-patches
β_n	Non-feature noise variance weight	Weight for noise variance cost for non-feature-patches
β_o	Non-feature overlap weight	Weight for overlap cost for non-feature-patches.
α	Shepard interpolation exponent	Shepard interpolation power parameter for patch merging.
d_ϕ	Shepard interpolation distance	Shepard interpolation distance for patch merging.

Table 3.2: The symbols, names, and descriptions of the parameters used in Tasse.

other weights in the sum. As such, the magnitude of the change must not only take into consideration the magnitude of the other weights, but also the magnitude of the costs that are weighted. There is no method to estimate the magnitude of this change to get the desired result. As an example, if the user prioritizes the feature dissimilarity cost for feature-patch selection in order to improve the matching of ridgelines and valley-lines, it is unclear how much to alter the weights to ensure that feature dissimilarity is the primary component of feature-patch selection, without neglecting the other weighted components. Too small a change, and the algorithm will produce a similar result. Too large a change, and the algorithm will neglect other important costs, like the graph cut score (Zhou) and feature overlap cost (Tasse). Both of these help match similar patches together, reducing the appearance of image artifacts, and neglecting them leads to less realistic terrain.

Another problem is adjusting the parameters based on the input of the example-map and sketch-map to produce the most realistic output. When selecting the best candidate for patch placement, the weights for each cost must consider the range of possible values for the associated cost. For instance, an example-map with a high variance in elevations will have a larger range of possible values for costs, than an example-map with low variance in elevations. Zhou et al. mention that a normalization of each cost is incorporated into the respective weights (this is also true for Tasse et al.'s algorithm). The problem with normalization is the estimation of the upper bound that acts as the divisor of the costs. For feature dissimilarity cost in Zhou and feature overlap area cost in Tasse, the upper bound is simply calculated as a multiple of the SSD between the lowest and highest elevation of the example-map. The issue is more complicated, with the deformation energy cost in Zhou and the noise variance cost in Tasse, where the upper bound is unlimited and requires the implementer of the algorithm to rely on an approximate divisor. This is especially difficult to estimate when the choice of divisor requires intimate knowledge of the TPS algorithm and signal processing, along with the divisor being dependent on the initial input. Zhou and Tasse do not provide a method of correctly calculating the divisors for these components and leave it for the user to estimate and incorporate it into their weight parameters.

As discussed in Section 2.1.2 there are no measures for the naturalness of terrain. This means the parameters for a given example-map and sketch-map input cannot be automatically determined by an algorithm to give the most realistic result. Additionally, the large search space of parameters and their complex interactions make it infeasible to generate a set of terrain outputs that cover the full range of possible inputs, as the set would be too large. Therefore, the user must find a suitable set of parameters using a trial and error approach. The user inputs a set of parameters, generates the terrain with the algorithm, and then evaluates the realism of the output, repeating the process until a terrain height-map of acceptable realism (according to the user) is synthesized. This presents us with two concrete problems: the repeated synthesis of terrain and the user's expertise in evaluating terrain. There is an issue here of how much time is needed to run each iteration, in both the synthesis time and human evaluation time. There is no guarantee that an acceptable result will be generated within a certain number of iterations, so with a limited time allocated to synthesizing, the trial and error approach may end early with an unacceptable and unnatural result. The user's expertise in evaluating the terrain for each iteration also limits the naturalness of the terrain. An incorrect evaluation of whether one terrain output is more natural than another leads to a sub-optimal selection of parameters. This issue of user expertise is discussed in further detail in Section 3.2.7.

In summary, the realism of terrain cannot be objectively measured, the large parameter space means you cannot generate terrain for the entire range of parameters, and the parameters are dependent on both the example-map and sketch-map. Because of this, the user must find the parameters using a trial and error approach. This leads the user to pick parameters that produce sub-optimal natural terrain for the given example-map and sketch-map.

3.2.2 Patch Size

Patches are used in Zhou and Tasse for copying features of terrain from the example-map to synthesize the output. The parameter *patch size* refers to the dimensions of the square patch extracted from the example data. A constant sized patch at a single

resolution is not suitable for extracting a variable sized feature from an example-map, which affects the output negatively.

Zhou and Tasse are primarily based on the patch-based texture synthesis method by Wu and Yu [154]. In patch-based texture synthesis, the window size (equivalent to the patch size in Zhou and Tasse) needs to be large enough to capture a section of the image for which the stationary property holds for that image. That is for a given window size, any two window sized sections of the image must be visually similar. This works well for homogeneous textures whose repeating features are uniformly distributed, but the features in terrain height-maps vary in size, shape, and distribution, due to the heterogeneous nature of terrain (see Figure 3.6). While one patch size might capture one type of feature accurately, the same patch size might be too large for a different type of feature, capturing more than intended. The patch size might also be too small for a single feature, resulting in a patch that has captured only part of a feature, or worse, a featureless patch. This flaw is even more severe when the patch is restricted to be square, as it will not correctly extract features that are wider than they are high (or vice versa), or are an irregular shape as shown in Figure 3.6e.

Finding the correct patch size for use in Zhou and Tasse is dependent on the example-map provided. As discussed previously in Section 3.2.1 the process of finding the ideal size requires the user to run the generation process multiple times in a trial and error approach. This also applies to finding the correct patch size because there is no intuitive way to select the best square patch size for the algorithm. One approach, used by Cruz et al. [40, 41] requires the patch size to be larger than every feature in the example-data, ensuring that a patch will not be too small to contain each terrain feature. However, this severely restricts the artistic control of the algorithm and is not suitable for Zhou and Tasse because they require a patch to contain a single feature of terrain in each feature-patch, so it can be more closely matched to the user sketch-map/target-map.

An alternative solution is to use a multi-resolution approach to synthesis. This approach is used by the majority of state-of-the-art texture synthesis, as it allows the synthesis to deal with textural structures that vary in size. However, for current patch-based methods to use multi-resolution approach, it would require

Figure 3.6: Patches of a fixed square size are unable to consistently capture terrain elements of a height-map that contains variably sized features. (a) Shaded relief map with highlighted sections of patches extracted from the ridgelines. (b) Patch size correctly captures a terrain feature. (c) Patch size is too large to capture a single feature and instead captures additional features. (d) Patch size is too small to capture the entire feature. (e) A square patch is not the right shape to capture this complex feature.

fundamental changes to the algorithm, as they currently rely on patches being only the size of a single feature in accordance with the feature graph.

The choice of patch size is critical for extracting single elements of terrain for feature-patches in Zhou and Tasse. However, many terrains appear to require variable, non-square, and non-axis aligned patch sizes for these elements to be captured correctly. Because of this Zhou and Tasse work as intended with a limited range of possible terrains as the example-map, for which a single square patch size is appropriate. The use of other terrain data as the example-map would cause the methods to produce a less natural result, due to the unintended placement of patches that contain less than an entire terrain element, or more than one terrain element.

3.2.3 Feature Identification

Zhou and Tasse both use the PPA for ridge and valley identification to create feature-graphs for both the example-map and sketch-map/target-map. The feature-graphs are used to extract feature-patches from the example-map and unused areas are used to fill in non-feature areas. However, the PPA does not separate features and non-features reliably enough for this purpose, which leads to the incorrect classification during synthesis. This results in inconsistent placement of patches and reduces the realism of the synthesized terrain.

The PPA is an algorithm for detecting ridge and valley-lines on a height-map as shown in Figure 3.7. The algorithm has three inputs: a height-map, grid interval μ , and profile length φ . Grid points are created on the height-map at vertical and horizontal intervals equal to the grid interval. The PPA performs *profile recognition* and considers each target grid point to determine whether it is a feature node by analyzing its profile in cardinal and ordinal directions: N-S, NE-SW, E-W, and NW-SE. The profile length φ determines the number of grid points used in the profile, centered on the target. The target is a feature node if there is one grid point lower than the center, on both sides of the profile. Every adjacent, and diagonally adjacent feature node is linked by an edge, with a weight equal to the sum of the elevations of the feature nodes it connects. For any pair of diagonal edges that overlap, the edge with the lowest weight is removed. The result is a graph of

Figure 3.7: An example of the result of the PPA. (a) Shaded relief map. (b) PPA result with ridgelines shown in red and valley-lines shown in blue.

weighted edges that connects feature nodes. The PPA then performs *polygon-breaking* by removing the lowest weighted edges until the graph is acyclic. An equivalent but more efficient polygon-breaking approach is Kruskal's algorithm, which is used in Tasse for the purposes of implementing the PPA. After polygon-breaking, the graph is reduced by iteratively removing feature nodes of degree one, as well as their edges. This is repeated a number of times equal to half of the profile length ($\lfloor \frac{\varphi}{2} \rfloor$) used for profile recognition. Finally, the graph is smoothed by setting each of the feature nodes positions to be the weighted average of itself and its neighbours in the graph (weighted by elevation). The algorithm can be adapted to create a graph for valley-lines by inverting the profile recognition to look for values that are higher than the center point and inverting the weight of the edges. The process of PPA is illustrated in Figure 3.8.

The grid interval φ is dependent on the height-map provided; both the spacing of the samples in the height-map, and the type of data in the height-map, affect the optimum value of the parameter (see Figure 3.9). If the grid size is specified in terms of samples then it is inversely proportional to the spacing of the sample, whereas, if the spacing (the real-world distance between the samples) of the height-map is

Figure 3.8: The process of the PPA using a grid interval $\mu = 20$ and profile length $\varphi = 7$. (a) Target recognition: ridge points are identified. (b) Target connection: adjacent ridge points are connected. (c) Polygon breaking: edges are removed to remove cycles. (d) Branch reduction: nodes of degree 1 are incrementally removed.

(a) $\mu = 10$ (b) $\mu = 20$ (c) $\mu = 30$ (d) $\mu = 40$

Figure 3.9: An example of the difference between different grid intervals $\mu \in \{10, 20, 30, 40\}$, with a profile length $\varphi = 7$.

increased the grid interval must be decreased, and vice versa. The grid interval is also dependent on the type of terrain presented in the height-map. The grid interval must be less than the frequency of the ridgelines, otherwise not all the ridgelines will be detected during the feature detection stage, because they will be located between the grid points. The grid interval must also be large enough to omit the classification of small convex features of the terrain including noise in the original data.

The profile length φ is the number of grid points that are used in determining whether a grid-point is a ridge target (see Figure 3.10). A profile length $\varphi = 3$ is pure shape recognition and recognizes all convex points as ridges, whereas a profile length $\varphi \geq 5$ can also recognize flat topped ridgelines, and grid-points close to the ridgelines as ridge targets. The larger the profile length, the more ridge targets that fit the classification of a ridgeline. With this increase comes a dramatic increase in the number of irrelevant ridgelines that are pruned in the later part of the algorithm.

While the PPA works well in GIS for analyzing the structure of a height-map (its original intention is to draw lines on a map for later subjective evaluation by experts), it does not reliably identify and separate features from non-features for the purposes of terrain synthesis. Issues in this space include the difference in parameters for use in terrain synthesis, the terrain features that are not detected, the decimation of long strings, and the identification of noise as features.

The difference in parameters of the PPA for use in GIS, versus the use in terrain synthesis, contribute to the undesired results of Zhou and Tasse. Zhou et al. do not provide the exact PPA parameters they used for synthesizing the terrain in their papers, so I was unable to replicate their exact results. However, the original implementation of Tasse, provided by Tasse et al., shows that the authors used a default grid interval of 5 pixels. For a typical height-map with a spacing of 30m per pixel, this equates to a grid interval of 150m. In the original paper for the PPA, the authors use a 0.01° grid interval on height-map of Taiwan, which is approximately 1025m at $23^\circ N$. This is an order of magnitude more than the value used by Tasse et al. As I discussed earlier, ridgelines can only be reliably detected if the grid interval is smaller than the ridgeline frequency. This means that the original intention of the PPA authors, was to capture large-scale features on the magnitude of kilometers,

Figure 3.10: An example of the difference between different profile lengths $\varphi \in \{3, 5, 7, 9\}$, with a grid interval of $\mu = 20$.

Figure 3.11: An example of the PPA where the length of a ridge is shortened, leading to misclassification. The green square indicates what patch would be extracted as an end-feature for the Zhou and Tasse algorithms. (a) After polygon-breaking but before branch reduction. (b) After branch reduction, the patch no longer contains the entire end of the ridgeline. (c) After smoothing, the patch is moved even further away from the original position of the end node.

whereas it is being used by Zhou and Tasse to capture much smaller features. This difference contributes to the identification of small-scale artifacts and noise as features, discussed later in this section.

The PPA does not identify major terrain features that do not have a prominent ridgeline, such as stepped features like cliffs and conical features like volcanoes. Stepped features can fail the feature detection stage because the profile on one side is lower than the center point, but the profile on other side may be the same elevation or slightly higher. Conical features are identified as features, but there will be few if any neighbouring feature points that will be linked with edges - the few edges that are created will be decimated during the branch reduction stage, leaving no ridgelines behind.

Long ridgelines can also be unintentionally shortened during the branch reduction process, leading to a misclassification of the end point features. The purpose of the branch reduction process is to remove small branches of ridges from the larger ridgeline that were detected due to the overestimation of feature points during the profile recognition. However, long ridgelines with few branches are unnecessarily decimated, shortening the length of the ridgeline. The result is that the end point

moves up the ridgeline further than it should, as shown in Figure 3.11. When features are extracted from the feature-graph, end-feature-patches will be extracted from a position that should be a path-feature. Additionally, in the case of Tasse, the position where the end point should be is left unidentified as a feature, resulting in the area being used as a non-feature instead.

Noise present in real-world height-maps may be unintentionally identified as features using the PPA. Typically, these height-maps have undergone pre-processing to remove artifacts and other inaccuracies from the data collection process, but they often still contain noise and small elevation variations that are difficult to remove automatically. While these differences in elevation are small relative to the features surrounding them, they are large enough that the PPA may identify them as features through the profile recognition. Because the profile recognition requires only one grid point to be lower than the center on either side, flat and featureless regions can become candidates for features due to the variations in elevation caused by noise. In a small number of cases, these features persist after the branch reduction step leading to Zhou and Tasse using these featureless regions as feature-patches. This effect is amplified when the grid interval is decreased, as a shorter grid interval is more sensitive to elevation variations in the height-map, making it more likely that noise is identified as a feature.

Overall, the PPA provides a poor analysis of the features (and implicitly the non-features) for use in terrain synthesis. When stepped and conical terrain features are not classified as features, they are incorrectly used as non-features for hole-filling. This leads to the placement of patches that contain strong terrain features in areas that the user has specified for there not to be any features. Additionally if the stepped and conical terrain features in the sketch-map are not identified as features, this leads to the placement of non-features where the user has specified for there to be features. The classification of features and non-features is also affected by features being classified as non-features, due to branch reduction on long ridgelines and featureless regions being classified as features due to noise. Thus, Zhou and Tasse will place patches in a way that does not match the user specified constraints. Both Zhou and Tasse do not perform any guidance of the

layout of the synthesis to ensure that it is natural, so they rely on the expertise of the user to provide a sketch-map that will generate a natural layout for the terrain (discussed further in Section 3.2.7). Because the features are not correctly identified in the example-map and the sketch-map/target-map, the user’s control over the artistic direction is decreased and less realistic terrain will be synthesized.

Feature identification is a topic of research primarily in the field of GIS and there is little to no focus on providing a method that performs feature identification for use in terrain synthesis. The use of the PPA for feature identification, in Zhou and Tasse, leads to the synthesis of less realistic terrain due to the limitations of its application. More recent methods by Bangay et al. [8] and Zhang et al. [158] are similar to the PPA and produce a graph with features that are more resistant to branch reduction and provided feature identification that is affected less by noise in the example. They both suffer from the lack of a robust method for detecting stepped and conical features, which would have similar problems in Zhou and Tasse with the poor classification of feature and non-feature terrain.

3.2.4 Thin-Plate Spline (TPS)

Zhou uses thin-plate splines to warp feature-patches and uses the deformation energy of the warping as part of the score for selecting the best feature-patch to place in the synthesis. Two TPSs are used to map the feature-graph for the sketch-map to the control points in the feature-patch, one for each dimension. With only three control points for path-feature-patches, the TPS warps the patch drastically in the majority of cases. This causes Zhou to produce artifacts leading to the synthesis of unrealistic terrain.

The name “thin-plate spline” is a reference to the physical analogy of bending a thin metal plate [28]. The deflection of the bend for any given point (x, y) is calculated as a function constructed from the control points $\{P_i \mid 0 \leq i < n\}$ and their corresponding deflection values $\{v_i \mid 0 \leq i < n\}$. The thin-plate spline

function is defined as

$$f(x, y) = a_1 + a_2x + a_3y + \sum_{i=1}^n w_i U(|P_i - (x, y)|), \quad (3.1)$$

where the first three terms, a_1 , a_2x , and a_3y , are coefficients that correspond to a linear function (affine transformation) and the last term corresponds to the bending forces applied by the n control points P_i (non-affine transformation). U is the TPS kernel function and is defined as $U(r) = r^2 \log r$. w_i are the coefficients for the n control points and all w_i form the vector W . The unknown coefficients a_1 , a_2 , a_3 , and W , are calculated by solving the equation

$$L^{-1}Y = \begin{bmatrix} W \\ a_1 \\ a_2 \\ a_3 \end{bmatrix}. \quad (3.2)$$

The matrix L is defined as a composite of matrices such that

$$L = \begin{bmatrix} K & P \\ P^T & O \end{bmatrix}, \quad (3.3)$$

where K and P are defined as

$$K = \begin{bmatrix} U(r_{11}) & U(r_{12}) & \dots & U(r_{1n}) \\ U(r_{21}) & U(r_{22}) & \dots & U(r_{2n}) \\ \vdots & \vdots & \ddots & \vdots \\ U(r_{n1}) & U(r_{n2}) & \dots & U(r_{nn}) \end{bmatrix}, P = \begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ \vdots & \vdots & \vdots \\ 1 & x_n & y_n \end{bmatrix}, \quad (3.4)$$

where r_{ij} is the Euclidean distance between two control points P_i and P_j , x_n and y_n are the x and y values of control point P_n , and O is a 3×3 matrix of zeros. The

column vector Y is defined as

$$Y = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (3.5)$$

where $v_n = f(x_n, y_n)$ is the deflection value for the control point P_n .

The TPS is used to warp a feature-patch that more closely matches the desired feature-graph. The deflection values for the TPS used in Zhou are the differences in the x and y values between the control points in the feature-patch and the control points in the feature-graph of the sketch-map. The most common feature-patch in Zhou is the path-feature-patch which has exactly three control points. Using TPS warping on path-feature-patches leads to dramatic warping due to the near collinear alignment of the control points in the majority of patches. In order to solve equation (3.2) for the TPS coefficients, there must be three or more control points and the control points must not be collinear. When exactly three control points are used, the solution to equation (3.2) forces all the w_i coefficients to zero, making the function linear as demonstrated below. For the case where there are only three points,

$$P = \begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{bmatrix}, K = \begin{bmatrix} 0 & d_{12} & d_{13} \\ d_{21} & 0 & d_{23} \\ d_{31} & d_{32} & 0 \end{bmatrix}, \quad (3.6)$$

$$\begin{bmatrix} K & P \\ P^T & O \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \quad (3.7)$$

Because there are only three points, the bottom half section of the matrix comprises three equations and three unknowns w_1 , w_2 , and w_3 , which can be solved independently.

$$\begin{bmatrix} P^T & O \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \Rightarrow P^T \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (3.8)$$

The general solution is that $w_1 = w_2 = w_3 = 0$. Substituting this back into Equation 3.7 gives

$$P \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}, \quad (3.9)$$

which can be solved in the usual way

$$\begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = P^{-1} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}. \quad (3.10)$$

Figure 3.12: An example of a 2D deformation using TPSs with a near-collinear case of three points. (a) Checkerboard texture with three near-collinear control points. (b) Shifting a control point one direction, by a small amount, causes severe compression of the texture. (c) Shifting a control point in the other direction, by a small amount, causes severe stretching of the texture.

If (x_1, y_1) , (x_2, y_2) , and (x_3, y_3) are collinear then P is singular and there is no solution. If the three points are close to collinear, then the system of equations is unstable leading to dramatic distortions, as shown in Figure 3.12.

The majority of patches used in Zhou et al. have only three control points. For example, the half-life symbol, used as a target for the terrain shown in Figure 3.13, contains 5 end features (two control points), 96 path features (three control points), and 1 branch feature (4 or more control points)⁵. Zhou et al. mention that degenerate cases can occur when the three control points are collinear, but does not provide a solution to the collinear or near-collinear cases [160, p. 5]. Further, the implication they make is that such cases are rare, whereas I found that they are so common that it is the cases that are not near-collinear that are rare.

The deformation energy of the TPS warping is proportional to WKW^T [28, p. 570], which is zero when all the components of W are zero. Path-feature-patches have three control points and thus have zero deformation energy, which means there is no additional cost for dramatic distortions in the near-collinear cases. This means Zhou may select heavily distorted patches for placement, as long as the other associated costs for the near-collinear candidate are lower than the other candidates.

⁵Numbers are approximate and are based on my implementation of Zhou.

In the texture synthesis paper, that Zhou is based on, Wu and Yu [154] require that the bottom and right edges of a patch are constrained in the warping to ensure that the patch remains square and that the deformations do not accumulate. This could be a possible solution to ensure that there are always more than three points and that collinear cases do not occur. However, this would restrict patches to be aligned with the synthesis, limiting the possibility of using rotated data (another flaw discussed in Section 3.2.6).

These examples show that the use of thin-plate splines for the deformation of feature-patches, as presented by Zhou et al., significantly decreases the realism of the terrain. Instead of TPSs, Tasse uses a precomputed set of rotated patches and uses the angle difference of outgoing edges in the feature-graph to help determine the best feature-patch to fit. This alternative allows Tasse to select a good fit for the feature-graph without distorting the data, leading to a synthesis with more natural terrain.

An alternative solution to calculating the deformation cost of path-features could involve creating a 2D affine transformation, extracting the scaling factors using singular value decomposition, and analyzing the magnitude of the scaling component. With the values of a_1, a_2, a_3 for both x and y patch deformations, a 2D affine transformation can be created,

$$T = \begin{bmatrix} a_{x2} & a_{x3} & a_{x1} \\ a_{y2} & a_{y3} & a_{y1} \\ 0 & 0 & 1 \end{bmatrix}, \quad (3.11)$$

that is used to transform a position in the feature-patch (path-feature) to a position in the synthesis. If the warping of a patch is defined by the distortion of its shape, the relative effects of rotation and translation can be ignored. Dropping the translation component, a 2×2 transformation matrix T' is left over. This transform can be

decomposed into a rotation R_1 , a nonuniform scale (composed of s_x and s_y), and another rotation R_2 [26], such that

$$T' = \begin{bmatrix} a_{x2} & a_{x3} \\ a_{y2} & a_{y3} \end{bmatrix} = R_1 \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} R_2. \quad (3.12)$$

This decomposition enables the use of the scaling components to evaluate the relative deformation for the feature-patch. However, further experimentation would need to be conducted to evaluate if this is an effective replacement to the problem of TPS for path-feature placement, and what function of s_x and s_y would best evaluate the deformation of a path-feature.

3.2.5 Interpolation

Zhou, Tasse, and Gain, all use a form of data interpolation where the original elevations of the real-world data are modified for the synthesis of the final output. Zhou and Tasse both use a form of Poisson image editing [113] for removing seams between patches and Gain uses a transfer function to produce height-offsets for minimizing the error of each coordinate. However, this modification can fundamentally change the original data and in some cases can lead to unrealistic terrain.

While both methods use a form of Poisson image editing to remove seams, Zhou artificially modifies gradient values along the seam and Tasse uses a combination of Poisson image editing and Shepard interpolation [129] to blend the edges together. The values that are altered during this seam removal process are in the overlapping region of the patch that is placed and the current synthesis. The Poisson blending technique works well in general to remove the artifacts left behind after the graph cut, but there are some complications such as the improper use of the blending technique and cases where the overlapping region is small. Zhou's method of artificially setting the gradient values along the seam leaves behind seam artifacts due to the improper handling of gradient values in the gradient domain (see Figure 3.13). Tasse et al. solves this by using Shepard

Figure 3.13: Shaded relief map of the terrain provided by Zhou et al. [160], highlighting the artifacts synthesized as part of the patch placement process. (a) A subsection of the height-map provided by Zhou et al. as an example of their algorithm with sections highlighting placement artifacts. (b) seam artifacts between non-feature-patches. (c) Seam artifacts between feature-patches. (d) Seam artifacts between a complex interaction of feature-patches.

Figure 3.14: A visualization of various situations for overlapping regions for seam removal. (a) The desired situation for overlapping, where the region is proportionally large compared to the patch area. (b) Situation where there is a narrow overlap between the patches. Seam removal still occurs but cannot satisfactorily smooth dramatic differences in elevation between the patches due to the small overlapping area. (c) Non-overlapping patches. Seam removal does not occur, leading to dramatic seam artifacts in the synthesis.

interpolation to blend the values in the gradient domain, and shows an improvement over just using the Poisson image editing, or just Shepard interpolation in the height domain.

A major flaw of only altering the elevations in overlapping regions using Poisson seam removal, is that it fails to produce realistic results when the altered region is narrow, as shown in Figure 3.14. And in cases where patches are placed adjacent to synthesized pixels (not overlapping), there is no graph cut or seam removal, leaving behind an artifact. Zhou et al. tries to address this problem by requiring that feature-patches are placed with a large overlap, and non-feature-patches are placed in order of the number of overlapping pixels. Tasse uses a similar method of feature-patch placement, but the non-feature-patches are placed using a modified onion-peel approach [39]. While both of these improve the results by reducing the number of small or non-overlapping regions, they do not completely eliminate the issue.

An alternative method of blending elevations from the example-map is presented by Gain et al. who use an adaptation of the transfer function from Han et al. [72] to add a height-offset to each pixel in the synthesis. The height-offset is calculated as the value that minimizes the SSD between the target neighbourhood and the candidate neighbourhood. A typical output of Gain has many coherent regions where neighbouring values in the synthesis are also neighbouring coordinates of

the example-map. For these regions, the height-offset is almost constant. The problem with a height-offset is that terrain that appears at a relatively low elevation in the example-map can be synthesized at a relatively high elevation in the synthesis. It does not make sense from a geomorphological stand-point that low-lying features, such as those found in flood plains, should appear at higher elevations. This offset is insensitive to the underlying processes that would generate that terrain and produces terrain that is uncharacteristic of the example-map. Gain et al. state in their paper that their algorithm does not consider the underlying geomorphology [61, p. 114] and instead Gain relies on the artistic direction to create realistic terrain. The issue of relying on artistic direction to create realistic terrain is discussed further in Section 3.2.7.

The interpolation of data during the synthesis of terrain in data-based methods is required to synthesize new terrain to satisfy the user specification. Without some form of interpolation, artifacts would be left between patches (Zhou and Tasse), or the synthesis would be an incoherent organization of data from the example-map (Gain). On the other hand, without the need to match user specified constraints, a method could simply reproduce the example terrain in its entirety without changing the terrain at all. Because interpolation is needed to match the user specification, and it directly affects the realism of terrain, the error introduced by interpolation must be minimized with respect to the likeness of the terrain that exists in the example. This means elevations may be interpolated, as long as the likeness of that region of terrain is similar to a region of terrain in the example, in order to preserve the existing realism of the example data.

3.2.6 Limitations Using Data

Zhou, Tasse, and Gain do not make full use of the data provided, which limits the naturalness of the synthesized output. All data-based methods require existing data to synthesize new terrain, and as such, the output is limited to replicating only parts that exist in the data provided. There are two limitations that an algorithm may have when using the data provided: being transform-limited, and being search-limited. Transform-limited is when there exists a transform that would

match the supplied data to the user constraints, but a method is unable to perform it. For example, if the supplied data contains a longitudinal river and the user constraints require a latitudinal river, then a rotation is required to transform the data to meet the constraints. If a method is not able to perform this rotation, then it is an example of the method being transform-limited. Search-limited is when supplied data that matches the user constraints exists, but the algorithm is unable to find it. For example, if the supplied data contains a strata volcano and the user has constrained the synthesis to produce a conical shape but the method cannot find the feature in the supplied data to match, then it is an example of the method being search-limited.

An obvious limitation of purely data-based methods is that they cannot produce realistic terrain that is not contained within the data provided. The data-based methods I have discussed are based on example-based texture synthesis methods, which finds elements of the data, and replicates and rearranges the elements to synthesize the terrain. A small data-set with few terrain elements constrains the synthesis and may lead to obvious visual repetition of terrain features due to the lack of variety. A large data-set with a variety of different features and structures increases the search space for elements and may reduce the algorithm's ability to find a particular element. It is to the benefit of a data-based method to utilize the data provided as fully as possible by: having a full range of transformations (include non-integer translation, rotation, and flipping, with consideration for offsetting the heights), to avoid being transform-limited, and a robust method of searching for features to avoid being search-limited.

However, with regards to being transform-limited, special consideration needs to be made as to what transforms are acceptable for application of terrain synthesis. Whether a transformation is acceptable is based on the effect it has on the data. Data-based methods rely on maintaining the realism of the source data in order to ensure that the output is realistic. If the realism is not maintained with respect to the source data, that is, if terrain elements from the source data are modified in a way that they are no longer like any terrain element in the source data, then data-based methods cannot ensure that the output is realistic.

Figure 3.15: An example that terrains exhibit rotational and reflectional symmetry, while others do not. (a) Mount Sundoro and Mount Sumbing, Central Java, Indonesia. The radial drainage patterns around the stratovolcanoes exhibit rotational and reflectional symmetry about the crater of each volcano. (b) Serranía del Aguara, Tarija, Bolivia. The straight mountain range shown here exhibits no rotational symmetry and only one axis of reflection symmetry that is perpendicular to the mountain range.

Figure 3.16: An example that similar terrain features, such as post-glacial threshold mountains, can exist at different absolute (relative to sea-level) elevations. (a) Mount Jackson, Colorado, United States of America, with an elevation range of approximately 2433–4138m (b) Mount Franklin, Canterbury, New Zealand, with an elevation range of approximately 398–2241m.

Let us consider the affine transforms: translation, rotation, reflection, shearing, and scaling, as well as the transformation of height-offsetting. Translation is fundamental for the process of the texture-synthesis methods that Zhou, Tasse, and Gain are based on. Rotation and reflection depend entirely on the data provided as many real-world terrains contain a strong component of alignment or may not have an axis of symmetry, as shown in Figure 3.15. Rotating or reflecting terrain elements in this case is not appropriate because the transformed elements would be unlike the terrain in the source data. This does not exclude rotation or reflection from being used entirely, just that it depends on the source data being used (whether terrain elements have a strong alignment component or an axis of symmetry), as well as the values of rotation and the axis of reflection that are being used. Scaling and shearing terrain elements is not appropriate, as it does not respect the terrain data and the underlying processes that would have led to its creation. For example, if a terrain feature, such as an alluvial river, is sheared or scaled, the width and/or length of the river is altered, but not its elevation. This undermines the power-law relationship between the river's slope, depth, and width. If the fundamental shape of the river feature has been changed, it may no longer reflect an accurate result of real-world physical processes that would lead to its creation. As such, scaling and shearing are not appropriate transformations in data-based terrain synthesis for any real-world terrain. Height-offsetting terrain elements is similar to the case of rotation and reflection, in that it depends on the structure of the source data and the value of transformation being used. Similar terrain elements, such as the ridges of the post-glacial threshold mountains shown in Figure 3.16, do not strictly appear at different absolute (relative to sea-level) elevations, but typically at elevations that are relative to the terrain system as a whole. In the case of these ridges, they will appear at the higher range of elevations for the height-map and further up the drainage network than other elements. If the relative elevation is not respected, then ridges may be placed too low, or floodplains too high, due to the underlying principles of physical geography. Therefore, while it may be appropriate to use height-offsets as a way of transforming the data to synthesize new terrain, adherence to the underlying principles must be taken into account to prevent features being synthesized at unsuitable elevations relative to the structure of the surrounding terrain.

Zhou and Tasse both use patches as terrain elements. Zhou transforms the patches using TPSs allowing the data to be transformed (and deformed) freely. This is not an acceptable method for transforming patches, because the TPS produces affine transformations that contain elements of scaling and shearing, as well as non-affine transformations which are inappropriate for transforming terrain⁶. On the other hand, Tasse uses discretely rotated and reflected patches for transforming terrain elements, which are acceptable for certain types of terrain, as discussed earlier. Neither Zhou nor Tasse use height-offsets to transform terrain elements.

Gain uses single elevation values and their neighbourhood as terrain elements. Due to the restrictions of the pixel-based texture-synthesis approach that Gain is based on, only translation and height-offsets can be used to transform the data. Gain et al. mention that lack of rotation and reflection is one of the major limitations of their method and suggest their focus on memory reduction allows for a range of orientations for the data to be provided in addition to the original orientation. This solution, however, is limited to allowing only discrete rotations and reflections of the data. The inclusion of the height-offset transformation is necessary for the functioning of elevation constraints, but may lead to the situations described earlier, where terrain features are not placed at correct elevations with respect to their surroundings.

In terms of search-limited algorithms, Zhou and Tasse both categorize sections of the example-map as terrain features using the PPA algorithm, and then match these sections to the features in the sketch-map. In addition, Tasse categorizes non-feature sections of the example-map as regions that were not identified as features using the PPA. As discussed in Section 3.2.3, the PPA does not reliably identify terrain features for the purposes of terrain synthesis, and thus Zhou and Tasse are search-limited when it comes to finding patches in the example-map for the purposes of synthesis. For example, if a terrain element is not identified as a feature through the PPA, it will not be considered as a candidate for placement during the feature-patch part of the synthesis. Because the terrain element exists and it is possible to be transformed

⁶The mechanisms of shearing in geomorphology produce fundamentally different results to geometrical shearing.

to a suitable location, but both Zhou and Tasse are unable to locate it as a candidate, they are classified as search-limited.

Gain uses a k -coherence look-up table, constructed by the PatchMatch algorithm, to search for terrain elements. The PatchMatch algorithm essentially performs an optimized global search for elements that are most self-similar within the example-map, but because it is precomputing a $k = 2$ look-up table at the start of the synthesis rather than computing the most similar element every step of the synthesis, it is possible that the correction for a single element during synthesis may be found through the small selection of values selected from the k -coherence look-up table. For instance, if the example-map contains two distinctly different types of terrain that split the map in two, the values contained in the k -coherence look-up table for an element of one type of terrain are unlikely to reference an element of the other type of terrain contained in the same example-map. This is mitigated in two ways: by ensuring that subsequent values in the k -coherence table are at minimum distance away from previous values, to ensure the diversity of candidates, and a multi-resolution synthesis, where the approximately correct terrain elements are selected early on in the synthesis. With this consideration it is unlikely that Gain is search-limited for real-world terrain, but an in-depth exploration k -coherence tables would need to be conducted to confirm this.

To avoid being transform-limited, an algorithm must be able to support transforming terrain elements using rotation, reflection, and height-offsets. However, to ensure that the data is not being transformed in a way that reduces the overall realism of the synthesized terrain with respect to the data provided, the algorithm must ensure that they are being transformed only in the appropriate circumstance, or should have these transforms as optional for the user to decide when is appropriate (relying on the user's expertise). To avoid being search-limited, an algorithm must employ a reliable global search mechanism that is not dependent on the data or parameters provided by the user.

3.2.7 User Expertise

Like many terrain synthesis methods, Zhou, Tasse, and Gain do not guarantee that the synthesized terrain is structurally realistic and instead rely on the user's expertise to provide the constraints that guide the synthesis toward a realistic result. Currently in the industry, the majority of users who want to synthesize terrain are artists who are typically inexperienced in the field of physical geography. This lack of expertise is not accounted for in the design of the terrain synthesis algorithms. The issue is that if an algorithm relies on the expertise of the user to drive the synthesis towards a more realistic result and the user does not have that expertise, then the algorithm will not reliably synthesize realistic terrain.

As discussed in Section 3.2.1, Zhou and Tasse require a trial and error approach in the generation of terrain where the user alters parameters until the desired realism is reached. In the case that the user lacks expertise in evaluating the realism of terrain, it is likely that they will be unable to decide if one terrain is more realistic than another, or if terrain has reached an acceptable level of realism (expertise in physical geography is correlated with accurately identifying real terrain, as discovered and discussed in Chapter 6). In a trial and error approach this leads to the termination of the process with a terrain that is less realistic than what would be possible if the user was more experienced in evaluating the realism of terrain.

In addition to Zhou and Tasse, Gain also requires the user to have expertise. Gain et al. state in their paper

"There is no explicit consideration of geomorphology, which may lead to unrealistic global drainage patterns. ... The responsibility for detecting and correcting such situations is placed on the user." (Gain et al. [61, p. 114])

In Gain, the user specifies the point, curve, type, and copy-paste constraints to control the synthesis. With these parameters, the user has more control over the synthesis of the terrain, but is required to have more experience, both with physical geography and the method to ensure that the method produces a more realistic result.

The two issues concerning user expertise is the lack of experience with a terrain synthesis method and the lack of expertise in the field of physical geography. The former issue, lack of experience with a terrain synthesis, can be solved by developing a method that is intuitive and gives a great degree of control to the user. This issue of intuitive control is outside the scope of this thesis. The latter issue, the lack of expertise in the field of physical geography, can be addressed by designing the algorithm to compensate for the lack of expertise. For example, as Gain et al. mention, they do not explicitly consider geomorphology in the design of their algorithm. However, if their method borrowed techniques from the field of physical geography to ensure that the structure of the terrain synthesized was more in line with the natural processes that formed the original terrain data, then the user would not have to be as experienced in order to synthesize a more realistic result. A terrain synthesis method that mitigates the knowledge required by the average user would be useful and is explored further in Chapters 4 and 5.

3.3 Conclusion

Data-based methods improve on approximation and model-driven methods because the realism of terrain is taken from real-world data and the synthesis is shaped to fit the user constraints, providing a balance of realism and artist direction. However, the current state-of-the-art data-based algorithms that are based on texture synthesis methods have some major flaws that need to be addressed in order to produce more natural terrain.

Zhou and Tasse are built on the patch-based texture synthesis methods and primarily rely on the placement of feature-patches to match user constraints. Through the replication of the algorithms I have found that the resulting terrains are acceptable in only a small number of cases as both Zhou and Tasse suffer from major flaws that inhibit the naturalness of the terrain. I believe that an algorithm for patch-based terrain synthesis would need to be designed from the ground up if it is to avoid the flaws outlined in this chapter. For that reason, I present no further work based on Zhou and Tasse in this dissertation, instead returning to the texture synthesis literature and designing a new algorithm. Currently there are no terrain

synthesis methods that are based on the texture synthesis method of texture-optimization. I explore the use of texture-optimization for synthesizing terrain in Chapter [5](#).

Gain is built on pixel-based texture synthesis methods and relies on the guidance of neighbourhood matching to match user constraints. The results are acceptable in most cases but the algorithm suffers severely with inexperienced users who cannot guide the synthesis to produce a natural output. I address this issue and expand upon the method further in Chapter [4](#).

Chapter 4

Combining Pit-Removal and Terrain Synthesis

The primary downside to the current data-based methods, that are based on texture synthesis techniques, is the lack of consideration for the global structure of the synthesized terrain with regards to how real-world terrain is formed. As discussed in Section 3.2.7, these methods rely on the user providing the appropriate constraints to ensure that the structure of the terrain is realistic, and make no effort to ensure that non-expert users can synthesize realistic terrain.

The surface flow of water plays a large role in shaping the topography of landscapes. Areas of terrain where the surface flow of water does not drain into an external water body are named endorheic basins (also known as ‘pits’ in GIS) and rarely appear naturally in height-maps. This is due to the effect of hydraulic erosion and gravity-driven sediment transport, where the matter is eroded and transported down-slope until it reaches a sink (e.g. the ocean). This process fills in existing basins and prevents new basins from being created, as matter would have to be transported up-slope against gravity. There are rare cases where pits occur naturally in terrain due to underground transport networks or where recently created pits such as sinkholes and craters have not yet been filled through the erosion process. There are also areas where the rate of evaporation is greater than rate of precipitation, reducing the effect of the erosion and transport of sediment.

Figure 4.1: A visualization of large endorheic basins around the world. Major endorheic basins are shown in dark gray, major endorheic lakes are shown in black, and continental divides are shown with dark lines. (Image used under a Creative Commons license (CC BY-SA 4.0). Image created by user AcidSnow at Wikimedia Commons)

This leads to basins being filled at a slower rate than they are being created. While there are areas on earth where endorheic basins cover thousands of square kilometers, subsections of those areas do not always meet the definition of an endorheic basin. For example, much of Central Asia has a large region of endorheic basins (see Figure 4.1), but subsections of that area do not contain endorheic basins themselves, as shown in Figure 4.2. From this, a reasonable assumption can be made that a synthesized terrain with more endorheic basins would be less realistic than a synthesized terrain with fewer endorheic basins, and that an algorithm could produce more realistic terrain if it minimized the number of endorheic basins it produced in its synthesized terrains.

In GIS the removal of pits (endorheic basins) is of great interest due to their role in interfering with calculations and simulations involving surface flow. Various pit-removal algorithms have been developed to efficiently remove pits while minimizing the change to the original height-map. These algorithms can be adapted to existing terrain synthesis techniques and provide a way to reduce pits in synthesized terrain to make them more realistic.

Figure 4.2: A subsection of the endorheic region in Central Asia showing, in yellow, the drainage basin of the Caspian Sea. Any area of terrain that drains into the Caspian Sea, but does not contain the sea in its entirety, is not an endorheic basin because it drains to an external water source. This demonstrates that subsections of endorheic basins are not necessarily endorheic basins themselves. (Image used under a Creative Commons license (CC BY-SA 4.0). Image created by user Redgeographics at Wikimedia Commons)

This chapter details my exploration of adapting an existing terrain synthesis algorithm to reduce endorheic basins using algorithms from the field of GIS. Section 4.1 discusses the three main types of pit-removal algorithms. Section 4.2 explores applications of pit-removal to the data-based terrain synthesis method of Gain et al. [61].

4.1 Pit-Removal Algorithms

Pit-removal algorithms are used for processing elevation data to ensure that there is always a down-sloping path from any point to a sink (by default, the edge of the data). When calculating the surface flow direction for elevation data that contains pits, the minima of the pit has an undefined flow direction which is problematic for many surface flow calculations and simulations. A terrain height-map T has no pits if, for every pixel $p_0 \in T$, there exists a connected path of adjacent pixels $(p_0, \dots, p_n)$ where $p_n \in S$ for a set of sink pixels S , and $T(p_i) > T(p_{i+1})$. The sink set S is typically the set of pixels on the border of the height-map but can also include bodies of water, sinkholes, etc.

There are a three main approaches for pit-removal: depression filling, depression breaching, and hybrid methods.

4.1.1 Depression-Filling

Depression-filling algorithms remove pits by increasing the elevations inside the pit region to ensure that all elevations have a down-slope to a sink. I implemented the state-of-the-art “Improved Priority-Flood” algorithm described by Barnes et al. [13] (referred to as the fill-algorithm) which performs depression-filling taking into account flow along flat areas. An example of the algorithm’s output can be seen in Figure 4.3.

The fill-algorithm takes, as input, a height-map and a set of pixels as the sinks. The output is a height-map with elevations greater than or equal to the input and without pits. The fill-algorithm uses an open-queue (a priority queue with

Figure 4.3: An example of depression-filling. (a) Source terrain height-map synthesized using Perlin noise. (b) Output from the depression-filling algorithm. (c) Difference between the other two images, where the yellow-red scale shows the increase in elevation from minimal to substantial.

ascending order of elevation), a pit-queue (a plain queue), and a closed-set (a plain set). The open-queue acts as a list of pixels that still need to be processed in an ascending order of elevation, which simulates the flooding process from the specified sinks. The pit-queue is used instead of the open-queue when a pit is encountered during the flood-fill process. Pixels in the pit-queue are explored in a breadth-first manner to ensure that the subsequent drainage is not dictated by the previous values of elevation, but by the spread of the area of the queue from its outlet. The closed-set keeps track of pixels to ensure that they are not processed more than once. First, the open-queue and the closed-set are initialized with the sink pixels. Next, the fill-algorithm polls a pixel p from the pit-queue, or the open-queue if the pit-queue is empty, and iterates over each neighbouring pixel p_n . If the neighbouring pixel is already in the closed-set it is ignored, otherwise it is added to the closed-set. If the neighbouring pixel p_n is a higher elevation than the pixel p , $T(p_n) > T(p)$, it is added to the open-queue, but if it is of equal or lower elevation, its elevation is modified to be slightly higher than the pixel p (next representable value above $T(p)$) and it is added to the pit-queue. This process is repeated until both the pit-queue and open-queue are empty.

Depression-filling is the most efficient pit-removal technique, processing each point of elevation only once, which makes it ideal for large height-maps used in GIS. It works well for pits with a small area and regions that are completely flat that need a flow direction, as there is only a small change in elevation values and the effects are barely noticeable. However, it does not work well for pits with a large area as the result is an often undesirable flat plane that spans the area of the pit.

4.1.2 Depression-Breaching

Depression-breaching (also known as carving) algorithms decrease the elevations around pit regions to ensure that all elevations have a down-slope to a sink. I implemented a variant of the algorithm by Soille [131] (referred to as the breach-algorithm) which determines the path of the channel by the flood order of the height-map.

Figure 4.4: An example of depression-breaching. (a) Source terrain height-map synthesized using Perlin noise. (b) Output from the depression-breaching algorithm. (c) Difference between the other two images, where the green-blue scale shows the decrease in elevation from minimal to substantial.

The breach-algorithm takes, as input, a height-map and set of pixels S as the sinks. The output is a height-map with elevations less than or equal to the input and without pits. The breach-algorithm uses an open-queue (a priority queue with ascending order of elevation), a closed-set (a plain set), and a flow-map F (a map of pixel to pixel). The open-queue acts as a list of pixels that still needs to be processed in an ascending order of elevation, which simulates the flooding process from the specified sinks. The closed-set keeps track of pixels to ensure that they are not processed more than once. The flow-map is a map of neighbouring pixels that represents the direction that a pixel should flow downhill. It is used to produce a path from any pixel to a point of lower elevation, or a sink pixel. First, the open-queue and the closed-set are initialized with the sink pixels. Next, the fill-algorithm polls a pixel p from the open-queue and iterates over each neighbouring pixel p_n . If the neighbour pixel p_n is already in the closed-set it is ignored, otherwise it is added to the open-set, closed-set, and flow-map ($F(p_n) = p$). If the neighbour pixel p_n is a lower elevation than the pixel p ($T(p_n) < T(p)$), the process of breaching occurs. Starting with the pixel $p_0 = p$, the breach-algorithm creates a path $P = (p_0, \dots, p_k)$, where each sequential pixel in the path is found using the flow-map $F(p_i) = p_{i+1}$ and this continues until an elevation lower than $T(p_n)$, meaning $T(p_i \in P) \geq T(p_n)$, or a sink pixel is reached, meaning $p_k \in S$. The elevation of each pixel in this path $p_i \in P$ is then lowered to the next representable value below the neighbour pixel $T(p_n)$ and the previous pixel in the path $T(p_{i-1})$. This process is repeated until the open-queue is empty.

Breaching is generally regarded as better than filling in GIS [90, p. 3] but is less efficient than the fill-algorithm due to backtracking when carving channels. Breaching algorithms are well suited for shallow pits that cover a large area, as the breaching channel alters a minimal number of pixels to remove the pit. However deep pits cause drastic modification to elevation levels when breaching, where the channel incises the surrounding terrain deeply. These deep pixel-wide channels are undesirable in many cases, especially when the channels are carved from pits created by noisy data, producing an appearance of a river network where there is none. There are other approaches to breaching, such as the least-cost-path method by Lindsay and Dhun [91], but they are developed for more specific reasons, like

breaching on fine-resolution height-maps of heavily altered landscapes, and are unsuitable for the general case of terrain synthesis.

4.1.3 Hybrid Method

Hybrid methods combine the techniques of depression-filling and depression-breaching to remove pits, with the intention of minimizing the downsides to each algorithm. I implemented the selective breaching mode for the hybrid approach by Lindsay [90], which breaches channels if the cost is lower than the user-defined thresholds and fills the rest of the pits.

The hybrid-algorithm takes, as input, a height-map, a set of pixels as the sinks, and the breaching thresholds: maximum channel depth m_d and the maximum channel length m_l . The output is a height-map without pits. First, the hybrid algorithm performs constrained breaching, which processes the pixels in the same way as the breach-algorithm (Section 4.1.2), but with the following modification. When the breach path P is calculated for a given neighbour pixel p_n , the path P is evaluated against the thresholds before any breaching occurs. If the depth of the channel breached is greater than the maximum breach dept, $\max_{p_i \in P}(T(p_i) - T(p_n)) > m_d$, or the size of the path is greater than the maximum channel length, $\|P\| > m_l$, then no breaching (lowering of elevation) occurs. This may result in some pits not being removed as part of the breaching process. Second, if there are still some pits unresolved, the fill-algorithm (Section 4.1.1) is used to remove the rest of the pits.

The selective breaching approach was implemented because it retains larger depressions and more closely models the impact of internal drainage [90, p. 10]. In general, hybrid methods reduce the area affected by depression filling (causing flat regions) and therefore increases the naturalness of the terrain. However, due to the additional parameters for the maximum channel depth m_d and the maximum channel length m_l , tuning is necessary to achieve the best results for the pit-removal for a given terrain. This parameter tuning requires an understanding of the trade-offs involved.

Figure 4.5: An example of hybrid-method. (a) Source terrain height-map synthesized using Perlin noise. (b) Output from the depression-breaching algorithm. (c) Difference between the other two images, where the yellow-red scale and green-blue shows the increase and decrease in elevation respectively, from minimal to substantial.

4.1.4 Summary

Many practitioners in the GIS community continue to use depression-filling algorithms for pit-removal, despite evidence that depression-breaching and hybrid methods are more aligned with the causes of depression artifacts and affect the overall height-map to a lesser extent [90, p. 3].

The algorithms are geared towards removing pits for GIS applications, such as surface flow simulations. They are not designed to make the terrain more realistic nor are they designed for terrain synthesis. Nevertheless, they do provide a solution to the presence of pits in terrain synthesis. In the next section, these algorithms and ideas are applied to an existing terrain synthesis method in three different ways.

4.2 Pit-Removal Applications

In my experiments I have adapted the concepts of pit-removal and flow and integrated them into the method by Gain et al. [61].

From a GIS perspective, pits can be the result of natural terrain features (e.g. sinkholes, craters, endorheic water bodies), unnatural terrain features (e.g. bridges, embankments, and dams), and artificial data (e.g. random noise and data collection artifacts). As such, the removal of unwanted pits is the primary goal of these pit-removal algorithms, not realism. Thus careful consideration needs to be taken to ensure that, when adapting these methods to remove pits for terrain synthesis methods (in an attempt to increase realism), the overall realism is not reduced due to unintentional features, including flat-planes from the fill-algorithm, or deep and thin valleys from the breach-algorithm.

This section presents three applications of GIS techniques to Gain et al.'s algorithm: height-offset modification, constraint addition, and flow-based appearance space. I compare all three pit-removal algorithms for the height-offset modifications, use the depression-breaching method for the constraint addition approach, and use flow-accumulation in the appearance space, as modifications to Gain et al.'s algorithm.

4.2.1 Height-Offset Modification

Height-offset modification, is an adaptation of the algorithm by Gain et al. [61], where the offset-map is adjusted to ensure that the algorithm synthesizes height-maps that are free of pits.

I call this algorithm Gain-pit and it is modified from the Gain algorithm in the following way. During the correction phase of the algorithm at each pyramid level, an extra step is taken to remove pits for the current synthesized terrain. After each iteration of correction, where candidates are selected to replace values in the synthesis and offset-map, the terrain is reconstructed using the coordinates in the synthesis to sample the elevations from the example-map and adding the height-offsets from the offset-map. Next, one of the pit-removal algorithms from Section 4.1 is used to remove the pits from the reconstruction R , to produce a new pit-free height-map R' . The difference between the reconstruction and the pit-free height-map, $R' - R$, is then added to the current offset-map. This ensures that if Gain-pit reconstructs the synthesis using the modified offset-map, it will result in the same pit-free height-map R' . This process is repeated after each correction iteration until the final synthesis level is reached.

I ran the algorithm on several terrains with all three pit-removal algorithms. An example of the difference between each pit removal algorithm is shown in Figure 4.6. The breach-algorithm appears far superior for removing pits than the fill-algorithm and only slightly better than the hybrid-algorithm. This is due to flat-area artifacts produced by the latter two algorithms during the synthesis process. The pit-removal algorithm is run many times during the synthesis because the correction process introduces new pits in each iteration. This compounds the flat regions produced by the fill-algorithm and produces an overall rise in the center to the terrain relative to the sides. Because the correction and de-pitting are also executed at low-resolutions, small flat regions, created by filling the pits, are enlarged due to up-sampling. The flat regions may be adjusted through the correction process, but in my experiments the correction process is not enough to avoid producing terrain that does not exist in the example.

Figure 4.6: A comparison between the original Gain algorithm and the results of the Gain-pit algorithm using the three different pit-removal algorithms. (a) The original Gain algorithm. (b) Gain-pit using the fill-algorithm. (c) Gain-pit using the breach-algorithm. (d) Gain-pit using the hybrid-algorithm.

Figure 4.7: A comparison between the original Gain algorithm and the results of the Gain-pit for constraints that imitate real terrain. (a) The original Gain algorithm (b) Gain-pit using the breach-algorithm. (c) Difference between the two images, where the yellow-red scale and green-blue shows the increase and decrease in elevation respectively, from minimal to substantial.

Compared to the original Gain algorithm, Gain-pit ensures that there are no endorheic basins, but pit-removal has little to no effect on the terrain synthesized if the constraints are specified in such a way that few pits form during the synthesis process. For instance, Figure 4.7 shows terrain synthesized by Gain and Gain-pit, with constraints to imitate the structure of the real terrain. There is only a slight difference in elevations between the two synthesized terrain images, and it is barely noticeable in a visual side-by-side comparison. The difference between the elevations reveals that, although the majority of pixels have been modified in some way, only a few regions have been lowered significantly due to the addition of breaching. However, when there are fewer constraints or less realistic constraints, the pit-removal does have a visible effect and ensures that the terrain is laid out more realistically in the context of eliminating the presence of endorheic basins.

One issue of using the breach-algorithm for Gain-pit is the drastic valley that can be introduced as a result of carving. This is a known side effect of the pit-removal algorithm and while performing the breaching at a low-resolution and at progressively increasing resolution can smooth out the valley, it still remains a problem for the final few layers of synthesis. This could be addressed by not performing the pit-removal at the highest resolutions of the synthesis, but this would result in some pits being left in the output. An alternative could use the breached paths as a guide for deforming the terrain indirectly through constraints instead of directly modifying height values. This is explored in the following section.

4.2.2 Constraint Addition

Instead of carving through the terrain, which may leave large and unnatural terrain features such as deep thin valleys, I attempted to provide a more natural way for the valleys to be placed to remove pits. As the breaching algorithm produces a path, it lends itself naturally to being used as a path-constraint for Gain et al.'s algorithm. Instead of carving the terrain by modifying the offset-map directly, constraints are added at each level of the synthesis to help reduce the pits in a more realistic fashion.

Figure 4.8: An example of Gain-constraint. (a) The results of the Gain-constraint algorithm. (b) A small breach-constraint that severely affects the synthesis. (c) Breach-constraints overlapping with the user defined constraints.

I call this algorithm Gain-constraint and it is modified in the following way. The Gain-constraint algorithm uses additional input parameters c_i , the control interval, and c_r , the control radius-of-effect, which aid in automatic generation of control points for curve constraints. During the correction phase of the algorithm at each pyramid level, an extra step is taken to add additional constraints to the synthesis. Before each iteration of correction, the terrain is reconstructed using the coordinates in the synthesis to sample the elevations from the example-map and adding the height-offsets from the offset-map. The breach-algorithm is used to remove the pits in the reconstructed height-map. In addition to the queue and map used by the breach-algorithm, the Gain-constraint algorithm also uses a breach-map, M_b , that maps each pixel to a breach path that runs through it. During the breach-algorithm, when a path $P = (p_0, \dots, p_k)$ has been constructed to remove a pit, each pixel in the path is mapped to the path in the breach-map, $M_b(p_i \in P) = P$, overwriting the previous mappings. This breach-map then provides a mapping from each pixel to the last breach path that ran through it. After the breach-algorithm has been performed, the breach paths are used to create curve constraints. The path P_c for each curve constraint is constructed from a corresponding breach path P , such that pixels $p_i \in P_c$ only if $M(p_i) = P$. This ensures that the curve constraints do not overlap and are only created for the paths that completely remove the pits in the height-map. Curve constraints with a path of less than the minimum path length are discarded (I discarded $\|P_c\| < 5$) to remove the small insignificant constraints. The control points along the curve are created by sampling the position along the path at an interval set by the control interval c_i . The left and right radius-of-effect for the control points are set to the constraint radius-of-effect c_r , and the left and right gradients are calculated by sampling elevations perpendicular to the path. After the curve constraints are constructed they are added to the set of constraints for the current iteration of the correction process. This is done as described by Gain et al. [61, p. 111], by adding the appropriate weights to the pre-computed constraint coefficients. This process of calculating additional constraints is run at every correction iteration. At each level the user defined constraints are applied as well. The result is a height-map that is not necessarily free of pits but has a more defined structure for overland flow.

I ran the algorithm on several terrains but was unable to achieve satisfactory results. There were several serious issues I encountered while testing this algorithm: the excessive number of constraints at higher resolutions, the drastic influence of pits, and the interference with existing constraints.

In my experiments, at a resolution of 1000×1000 pixels the synthesized terrains averaged approximately 10000 constraints at the highest synthesis level. This was the driving factor behind my decision to (1) remove constraints below a certain length, as there were many paths that were only a few pixels long, and (2) limit the constraints to be calculated at the same level as the user-defined constraints, as lower resolutions do not produce as many breach paths. However, reducing the number of breach constraints and the pyramid levels at which they are calculated reduces the overall effectiveness of the pit removal process, resulting in some pits left in the final terrain output.

Even though the number of breach constraints is reduced by removing shorter paths, the less important constraints still impact the terrain synthesis in a significant way. Small deep pits create constraints that affect an area much larger than they would if they were just carving using the height-offset modification (as seen in Figure 4.8b). While the curve constraint parameters for the shape, gradients and height values can be calculated trivially, selecting the control points along the curve and the radius-of-effect are too complex to calculate automatically and currently require the user to specify set values for all curves. This is difficult for the user to optimize, especially as large control radius-of-effect c_r values intensify the effect of small breach constraints (creating large pits as seen in Figure 4.8b), and small values decrease the effect of long and deep constraints that need to carve wider valleys through the terrain.

In addition, breach paths sometimes cut through existing user constraints. Breach paths are calculated based on the elevation of the reconstruction for the current level, and can often overlap with existing user defined curve constraints. This creates a conflict where the breach path constraints overlap with the user constraints, but provides a different elevation for the overlapping region. While situations where the user is specifying constraints that produce endorheic basins should be rare, it is still a concern. Figure 4.8c shows that the addition of constraints along the breach

path may conflict with these user defined constraints, resulting in a blurry synthesis that does not remove the endorheic basin entirely. For overlapping constraints Gain et al. [61, p. 108] specifies that when a user creates intersecting curve constraints a control point is used at the location of the junction, to ensure that the curves do not interfere with other constraints on the correct value of the elevation. This does not apply for breach path constraints as the shared height value will neither satisfy the original user constraint, nor the breach path trying to remove pits. The conflict between the breach paths and existing constraints would need to be addressed if this method is to produce more realistic terrain.

4.2.3 Flow-Accumulation Appearance-Space

As an alternative to removing the pits directly, I experimented with taking structural information about the drainage networks in the example-map and using it in the appearance-space that drives the correction process for the synthesis. The idea is that, if the structural information about the layout of the example map is embedded as a heuristic for the correction, the algorithm will synthesize a structure more like the example-map, but not necessarily guarantee that pits will be removed.

I call this algorithm Gain-flow and it is modified from the Gain algorithm in the following way. During the construction of the appearance-space for the example-map, at each level of the pyramid, the algorithm computes a flow-accumulation map, a map that provides the accumulation of the flow from all the cells that flow downhill through a given cell. This is done by first computing a directional flow map using a D8 flow method by Barnes et al. [12], which removes pits in the height-map (using a depression-filling approach) and models flow direction from each cell to its steepest down-slope neighbour. Gain-flow then computes the flow-accumulation for each cell as the sum of the flow-accumulation of the cells that flow into it, where cells that have no upstream cells have a flow accumulation of 1. Figure 4.9 shows a visualization of the d8 flow directions and the flow-accumulation for an example map.

Next, the weighted log of the flow-accumulation-map is used to construct the appearance-space for the example map. The weight ensures that the values of the

Figure 4.9: An example of the flow over a height-map. (a) A visualization of the 8-direction flow of the example-map converted to the Lab colour-space, where the four cardinal directions converted to colours: north (yellow), south (blue), east (red), and west (green). (b) A visualization of the log of the flow-accumulation-map, calculated from the flow directions.

Figure 4.10: A comparison between the original Gain algorithm and the results of the Gain-flow algorithm using the same constraints. (a) The original Gain algorithm. (b) The Gain-flow algorithm.

flow-accumulation-map can be relatively balanced against the elevation values in the example-map and the log evens out the range of values produced by the flow-accumulation. In the original algorithm of Gain, the appearance-space is calculated by taking the Gaussian weighted 5×5 neighbourhoods around each pixel in the example-map, producing a 25d neighbourhood-vector, and using PCA to reduce this to a 4d appearance-vector. In Gain-flow, the neighbourhood-vectors are calculated as two Gaussian weighted 5×5 neighbourhoods, one from the elevations in the example-map and one from the flow-accumulation-map, producing a 50d neighbourhood-vector. The process then continues as normal, with the neighbourhood-vector being reduced through PCA to four significant components for the appearance-vector.

I ran the algorithm with several different example-maps, target-maps, and weights for the flow-accumulation-map. I found that the addition of the flow-accumulation-map creates a decidedly different but not necessarily more realistic output. An example shown in Figure 4.10 provides some insight into the difference between the outputs of the two methods. The Gain-flow result (Figure

Figure 4.11: A comparison of the reconstructed flow-accumulation and the real flow-accumulation from the Gain-flow result in Figure 4.10b. (a) Flow-accumulation reconstructed using the coordinates from the synthesis into the flow-accumulation-map from the example-map. This gives an insight into what the algorithm was guided towards creating, based on the introduction of flow-accumulation into the appearance space. (b) Flow-accumulation calculated from the synthesized height-map. Notice that it differs significantly in terms of connectivity and how it drains to the edge, but the diagonal directionality is similar.

4.10b) exhibits more of the structural qualities of the original example-map, especially the fault bounded mountain ranges with strong parallel structural control, running diagonally from SW to NE. Figure 4.11a shows a reconstruction of the flow-accumulation from the example-map using the coordinates from the synthesis. This reconstruction shows how integrating the flow-accumulation-map into the appearance space influences the correction to create structures of flow inline with the example - even the diagonal structure is similar to the example. Yet, it is clear that the reconstructed flow accumulation is not connected and does not flow outwardly to the edges of the height-map, especially when directly compared to the correct flow-accumulation calculated for Gain-flow in Figure 4.11b.

While the addition of flow-accumulation in the appearance space adds a heuristic for reconstructing similar structures from the example-map, there is more research required to properly explore this application. It is clear that simply adding a weighted flow-accumulation map is not sufficient for synthesizing more realistic terrain, and that it does nothing to aid the issue of endorheic basins in synthesized terrain. One potential avenue of future research could be to investigate the integration of both flow direction and accumulation, similar to the recent self-tuning texture-optimization approach by Zhou et al. [161], which integrates source guidance channels into the energy function as a target for optimizations. These source guidance channels include information such as a scalar progression map and an orientation field, which could equate to using a flow-accumulation-map and a flow-direction-map for the application of terrain synthesis. However, further experimentation would be necessary to confirm if this is applicable to synthesizing realistic terrain.

4.3 Conclusion

As discussed in this chapter, the presence of endorheic basins in synthesized terrains reduces the overall realism of the terrain. The prevalence of these basins is also exemplary of the fact that current data-based methods lack the ability to ensure that the structure of the synthesized terrain is as realistic as the data provided, without relying on the expertise of the user to guide the synthesis

towards a more realistic result. I explored the use of pit-removal algorithms from the field of physical geography to address this issue.

Three main approaches to pit removal were discussed and three applications to the Gain et al. [61] algorithm were explored. Height-offset modification (Gain-pit), which altered the offset-map with the results of the pit-removal algorithms to directly ensure that there were no pits during the synthesis. Constraint addition (Gain-constraint), which used the breach paths created by depression breaching to add additional constraints to remove pits during synthesis. And flow-accumulation appearance-space (Gain-flow), which added the modified results flow-accumulation to the appearance-space to guide the synthesis.

After synthesizing many different results for the three algorithms, my personal analysis is that Gain-constraint and Gain-flow, as presented, produce terrain that is insufficiently realistic, while Gain-pit (with depression-breaching) produces the most realistic results. I therefore use Gain-pit in the comparative user-study in Chapter 6.

Chapter 5

Terrain-Optimization

This chapter details my experiments creating the terrain-optimization algorithm. As discussed in Chapter 2 most of the data-based terrain synthesis algorithms are based on methods of texture synthesis. While the patch-based and pixel-based approaches have been adapted to synthesizing terrain there is currently no research that I have found that applies texture-optimization techniques to the field of terrain synthesis. The current state-of-the-art methods for synthesizing terrain from example data include the methods by Tasse et al. [137] and Gain et al. [61] which were inspired by patch-based and pixel-based texture-synthesis methods respectively (see Section 3.1). Texture-optimization algorithms perform comparably well to other example-based texture synthesis methods as evidenced from a subjective evaluation of texture synthesis methods by Kolář et al. [83]. It follows that if other texture synthesis methods are worth investigating for adaptation to terrain synthesis, then texture-optimization is worth exploring as well.

This chapter explores the adaptation of the texture-optimization approach to the application of terrain synthesis. Section 5.1 outlines a basic texture-optimization algorithm presented by Kwatra et al. [85] and discusses how well it performs for texture synthesis and its limitations. Section 5.2 presents my algorithm, terrain-optimization, which is primarily based on the Kwatra et al.'s algorithm. Section 5.3 details and discusses the results of my experiments to improve the terrain-optimization algorithm.

5.1 Texture-Optimization by Kwatra et al.

The algorithm by Kwatra et al. [85] (referred to as Kwatra) is an optimization-based approach to texture synthesis. Kwatra et al. offer a core approach to texture-optimization with some optional additions designed to improve the result of the synthesis. The core algorithm synthesizes a texture using an EM-like approach where the algorithm alternates between finding a correspondence between the current synthesis and the example-texture, and re-synthesizing the output using the correspondence it has found. Both of these steps aim to minimize a texture-energy function, the sum of SSD between the patches of the synthesis and the corresponding patches in the example-texture. The additions to the core algorithm include:

- generalizing the texture-energy function to include other measures of texture energy,
- using robust optimization to reduce the effect of outlying data,
- multi-level synthesis to aid with synthesizing larger texture structures,
- controlling the synthesis using weighted soft-constraints,
- and generating animated textures using flow.

The following Section 5.1.1 is a description of the basic algorithm and Section 5.1.2 discusses the additions mentioned in the paper in application to texture synthesis.

5.1.1 Basic Algorithm

Kwatra takes, as input, the example texture and outputs a similar synthesized texture. Additional inputs are necessary for the additions to the algorithm and are described later.

First, the algorithm creates a correspondence-map, which acts as a map between patches p_i (of size $N \times N$) in the synthesis P to patches q_i in the example-texture

Q that minimize the texture energy function. Kwatra only stores a sparse grid of patches in the correspondence-map, where the patches are spaced $\frac{N}{4}$ apart (in a sparse grid pattern), in order to reduce the number of redundant mappings between neighbouring patches. The correspondence-map is initialized by selecting random patches q_i from the example-texture.

Second, using the correspondence-map from the first step, Kwatra begins the iterative process of optimizing the synthesis P to minimize a texture energy function with respect to the example texture Q . The texture energy function can be described as:

$$E_t(P) = \sum_{i \in P} \|p_i - q_i\|^2, \quad (5.1)$$

where p_i is a patch centered around pixel $i \in P$ and q_i is the corresponding patch in Q which is stored in the correspondence-map. The optimization is performed by alternating two steps: the E-step which minimizes the energy function by calculating new values for each patch p_i in the synthesis, and the M-step which minimizes the energy function by selecting the best patch q_i for each patch p_i in the correspondence-map.

The E-step calculates the new values of the synthesis with regard to the current correspondence-map. Because the patches are of size $N \times N$ and are calculated for a sparse grid of values that are spaced $\frac{N}{4}$ apart, the patches p_i overlap. The new value for every pixel in the synthesis is then calculated from the contributions of the overlapping patches q_i from the correspondence-map. As the optimization process minimizes the energy function of the synthesis with respect to the example, the new value is equivalent to the process of averaging the corresponding pixel values in overlapping patches q_i . The result of the E-step is that when the difference between contributing patches is large, the new value is a compromise between the different patches, sometimes resulting in a blurry region. This allows the M-step to find new patches that best match the compromise made.

The M-step selects new patches q_i for the correspondence-map with regard to the patches p_i for the current synthesis. A search is performed to find the patch q_i

corresponding to the patch p_i of the synthesis that minimizes the texture energy function. Kwatra uses a tree-based hierarchical k-means clustering algorithm that recursively searches for better matches in the example. Kwatra et al. suggest that only a good match is needed as opposed to a perfect match, so long as the texture-energy function continues to decrease as a result of the optimization process. Following the previous E-step, the M-step selects an arrangement of patches from the example-texture that best match the current synthesis. Blurry regions in the synthesis aid the M-step in selecting neighbouring patches that are more consistent with each other.

Finally, the process of EM-like optimization is finished after a few iterations (they use 3–5 iterations) and the final synthesis is used as the output.

5.1.2 Additions

The core Kwatra algorithm can be altered with different approaches to improve results. The following is a brief description of each addition described by Kwatra et al., as well as a discussion of its effects on the synthesis with supporting evidence from other research. The description and discussion of the flow-guided synthesis addition is omitted because it extends the synthesis to generate a sequence of textures as a function of time, making it irrelevant to the context of synthesizing a single texture.

Generalized Energy Function

The energy function for texture-optimization can be generalized to include other measures of texture-similarity for use in the optimization process. Other measures of energy can be added to the energy function as a weighted sum or can replace the function entirely as long as they are optimizable with respect to the synthesis. That is, as long as an energy function allows the E-step of the algorithm to select new values of the synthesis that minimize the energy function, that energy function

can be used. Kwatra et al. give the example of adding gradient-based energy to the energy function such that

$$E_t(P) = \sum_{i \in P} \|\mathbf{p}_i - \mathbf{q}_i\|^2 + \mu \sum_{i \in P} \|\Delta \mathbf{p}_i - \Delta \mathbf{q}_i\|^2, \quad (5.2)$$

where μ is a relative weighting coefficient and Δ is the differential operator (the exact operator is not specified). In this case, minimizing the function with respect to the synthesis requires solving a linear system similar to the Poisson equation [113].

This extension is further explored by Darani et al. [44] who applied the technique for hole-filling in images (non-textures) with positive results. However Kaspar et al. [80, p. 4] acknowledge that while this technique is suitable for hole-filling for most images, it is not suitable for texture synthesis because it increases the blurring of details and does not add any significant improvement over using the core energy function.

Robust Optimization

The texture energy function equation (5.1) used in the E-step of the texture-optimization is susceptible to outliers negatively affecting the result of the optimization. In the case where $\mathbf{q}_i$ is not close to $\mathbf{p}_i$, but many other overlapping patches $\mathbf{q}'_i$ are close to their respective $\mathbf{p}'_i$, all of the synthesis pixels in $\mathbf{p}_i$ are undesirably affected by the outlier. To solve this, Kwatra et al. suggest using a robust energy function, where the square term $\|\mathbf{p}_i - \mathbf{q}_i\|^2$ in (5.1) is replaced with $\|\mathbf{p}_i - \mathbf{q}_i\|^r$ for some $r < 2$. This, however, makes the equation more computationally expensive to use and so Kwatra et al. recommend that the energy function can then be viewed as a weighted least squares problem and solved using iteratively reweighted least squares (IRLS) :

$$E_t(P) = \sum_{i \in P} \omega_i \|\mathbf{p}_i - \mathbf{q}_i\|^2. \quad (5.3)$$

Before applying each E-step of the algorithm, for each patch p_i , a weight, $\omega_i = \|p_i - q_i\|^{r-2}$, is calculated based on the current closest patch from the example q_i . Furthermore, Kwatra et al. suggest using a Gaussian kernel within the distance function to ensure that pixels closer to the center of the patch have a greater effect on the synthesis than pixels that are further away from the center.

In my experiments, the robust optimization has little to no effect when the algorithm is optimizing a dense grid of patches for the synthesis, where every pixel $i \in P$ is used for optimization, as opposed to pixels spaced $\frac{N}{4}$ apart in a sparse grid. This indicates that using a dense grid of patches for the optimization process is equivalent to the robust optimization for the purposes of texture-synthesis. Furthermore, current state-of-the-art methods [44, 80] for texture-optimization also do not use this robust optimization, instead using optimization of dense grids of patches in the synthesis. This suggests that the robust optimization does not provide additional benefits when using dense grids. One possible explanation for this similar result is that, with a dense grid, adjacent patches p_i in the synthesis are likely to have corresponding patches q_i (in the correspondence-map) that are also adjacent (and thus are similar) in the example. This redundancy may provide a similar effect to the robust-optimization, where the few outlying patches that do not agree on the pixel value are outweighed by the numerous redundant patches that do agree on the pixel value. For a patch size of 16×16 there is a maximum of 16 patches q_i that overlap a single pixel using a sparse grid spaced $\frac{N}{4}$ apart, whereas there is a maximum of 256 patches using a dense grid. Therefore, on average, a single outlier patch will affect a greater proportion of a pixel value in a sparse-grid approach (a minimum of $\frac{1}{16}$) than in a dense grid approach (a minimum of $\frac{1}{256}$).

Multi-Level Synthesis

Multi-level synthesis is introduced to help synthesize larger texture structures from the example by synthesizing at a coarse resolution then up-sampling and optimizing it at progressively increasing resolutions (multi-resolution) and using varied patch sizes during optimization (multi-scale). The multi-resolution

approach is similar to multi-resolution synthesis found in the pixel-based methods [150, 87, 88], which helps reduce local minima of the texture energy function when optimizing large textures. The multi-scale approach uses decreasing patch sizes for successive optimization for each level of the multi-resolution synthesis. Using larger patch sizes first has a similar effect to multi-resolution, where the large-scale structures are optimized first and smaller details are refined later. Kwatra et al. used three resolution levels and three patch sizes of $N \in \{32, 16, 8\}$.

Multi-resolution works well for texture synthesis and is used by all subsequent texture-optimization approaches that I have found, such as the state-of-the-art image melding [44] and self-tuning texture-optimization [80] methods. Multi-scale however is not used in these state-of-the-art methods, likely because large patch sizes are computationally expensive to optimize, especially when using a dense grid of patches for the synthesis.

Controllable Synthesis

Controllable synthesis is the process of texture synthesis that can produce an output controlled by specified constraints. Kwatra et al. introduce the concept of adding an additional term to the texture energy function that represents the control constraints. The example used by Kwatra et al. is to add a soft constraint, where desired pixel values are specified at particular locations. The texture energy function can then be expressed as

$$E_t(P, C) = \sum_{i \in P} \|p_i - q_i\|^2 + \lambda \sum_{j \in C} (P(j) - C(j))^2, \quad (5.4)$$

where C is the set of constrained pixel values and λ is a relative weighting coefficient. The E-step is modified to optimize the synthesis with a new system of linear equations or, in this case, simply taking the weighted average of the contribution from the overlapping patches q_i and the constrained pixel values. The M-step is modified by looking for patches that are consistent with the control criteria. Specifically, in the case of soft-constraints for each p_i , the algorithm looks

for a q_i that minimizes $\|q_i - p_i\|^2 + \lambda \|q_i - p_i^c\|^2$, where p_i^c is the constrained pixel values for the given neighbourhood p_i . In this way, the algorithm finds a balance in selecting patches that are close to both the current synthesis and the specified constraints.

A major issue with this approach is that the synthesis for each iteration is obtained by blending the constrained pixel values into the synthesis, which can lead to excessive blurring when the values of the constraints are at odds with the patches q_i selected during the M-step. One solution presented by Han et al. [73] is to use a discrete optimization for the E-step - a single value is copied the contributing patch q_i that minimizes the texture-energy function, instead of averaging the values of the contributing patches q_i . This approach is not seen in subsequent texture-optimization methods, likely because it is closer to a pixel-based texture synthesis approach rather than a texture-optimization approach. Using discrete optimization limits other aspects of the algorithm such as not allowing the optimization of the synthesis with respect to the gradient with a generalized energy function.

An alternative method for controllable synthesis is hard-constraints, which set the values of certain pixels to be constant. In this case the M-step remains unchanged and the E-step is modified so that it does not optimize the constrained pixels. This is largely used for the purpose of hole-filling, such as in algorithms by Welexer et al. [151] and Darabi et al. [44], where the constrained pixels must remain the same and the hole is optimized to fit in with the rest of the image.

5.2 Terrain-Optimization

I introduce a new algorithm called terrain-optimization that adapts the method of Kwatra et al. [85] and other texture-optimization techniques to synthesize terrain. More assumptions can be made to inform this adaptation, because terrain synthesis is a more specialized application than general texture synthesis. This section presents the terrain-optimization algorithm and a discussion of the results.

Figure 5.1: An example of synthesizing terrain using the terrain-optimization algorithm. (a) Perlin noise used as the initial target terrain T . (b) Terrain synthesized using the core terrain-optimization algorithm.

5.2.1 Core Algorithm

The user provides, as input, a source terrain height-map (source-map S) and an initial estimation of elevations for the desired height-map (target-map T). The source-map S provides the real terrain features that will be used to synthesize a new terrain height-map, and the target-map T is used to guide the algorithm to produce terrain in line with artistic direction. The final synthesized terrain R is a reconstruction of patches extracted from the source-map S .

First, the algorithm creates a Gaussian image pyramid of depth L , for both the source-map S^k and the target-map T^k , and a correspondence-map C^k for each pyramid level k that maps the patches of the current synthesis R^k to patches in source-map S^k . The highest resolution image in the Gaussian pyramid is at level $k = 0$ and the lowest resolution image is at level $k = L$. The lowest level in the Gaussian pyramid for the target-map T^L is used as the initialization for the synthesis R^L . The correspondence-map C^k maps patches r_p (of size $N \times N$) centered on every pixel $p \in R^k$ to a patch s_p in the corresponding source-map S^k .

The values stored in this map are transforms, where the correspondence-map contains a mapping between the center pixel p of r_p and the coordinate $[x, y]$ of the center of s_p , such that $C^k(p) = [x, y]$. The values in the correspondence-map are initialized by finding the best patches in S^L that match the current synthesis P^L , in the same way as the M-step described below.

Second, after initialization, starting at the lowest level, $k = L$, of the Gaussian pyramids, the algorithm begins the process of optimization and up-sampling. The optimization is an iterative process, minimizing the energy function of the synthesis R^k (with respect to S^k) and uses an EM-like approach. The energy function is similar to the texture energy function proposed by Kwatra et al. [85]. The energy of a synthesized terrain is the sum of SSD between each patch r_p in the synthesis R^k and its closest patch s_p in the source-map S^k . This function takes the form

$$E_t(R^k) = \sum_{p \in R^k} \|r_p - s_p\|^2. \quad (5.5)$$

The E-step of the optimization calculates the new values of the synthesis R^L from the patches in the current correspondence-map C^k , reconstructing the synthesis with patches from the source-map. The new value for every pixel in the synthesis is the mean of values from the overlapping patches s_p , formalized as

$$R^k(p) = \sum_{q \in Q} \frac{s_{p+q}(n_b - q)}{N^2}, \quad (5.6)$$

for each pixel $p \in P^k$, where $Q = [n_a, \dots, n_b]^2$ are the coordinate offsets for each patch, N is the square patch size (typically $N = 7$), and $n_a = \lfloor \frac{-N}{2} \rfloor$ and $n_b = \lfloor \frac{N-1}{2} \rfloor$ are the patch offsets on opposite sides of the patch center.

The M-step of the optimization finds new patches $s_p \in S^k$ that best match each patch r_p in the current synthesis, that minimize the texture energy function. As an alternative to the tree-based method used in Kwatra, the algorithm uses the PatchMatch algorithm [10], which more efficiently finds a corresponding patch that approximately minimizes the energy function. The correspondence-map is then

updated with the new mapping of $\mathbf{r}_p$ to its closest match $\mathbf{s}_p$ found using the PatchMatch algorithm, such that

$$C^k(p) = \arg \min_{\mathbf{s}_p} (\|\mathbf{r}_p - \mathbf{s}_p\|). \quad (5.7)$$

This EM-like optimization process is computed μ times (I use $\mu = 2$, as a balance between realism and speed) before the correspondence-map C^k is up-sampled to the next level C^{k-1} . The new values in the up-sampled correspondence-map are computed as

$$C^{k-1}(p) = 2C^k\left(\left\lfloor \frac{p}{2} \right\rfloor\right) + \left(p - 2\left\lfloor \frac{p}{2} \right\rfloor\right). \quad (5.8)$$

After the correspondence map has been up-sampled, the optimization and up-sampling process repeats on level $k - 1$. The optimization and up-sampling is repeated until the optimization at $k = 0$ is complete, and the final output is the reconstructed synthesized terrain P^0 .

5.2.2 Constraints

The terrain-optimization algorithm has three types of constraints that can be used for artistic direction: soft constraints, hard constraints, and type constraints.

Soft Constraints

Soft constraints for the terrain-optimization algorithm are similar to the soft constraints proposed by Kwatra et al. [85] which guide the optimization process to synthesize values closer to, but not exactly the same as, the constrained values. Soft constraints are based on the soft constraint additions to the texture-optimization algorithm by Kwatra et al.

To use soft constraints, the target-map provided by the user becomes the soft-constraint target for each pixel. More specifically, the target-map T^k is used as the soft-constraint for the synthesis R^k at each pyramid level k . I found it sufficient

Figure 5.2: A comparison between the effect of different soft weight functions w . (a) No weights, $w = 0$ (no soft constraints, initialization only). (b) Exponential weighting, $w = \alpha \frac{2^k - 1}{2^L - 1}$. (c) Linear weighting, $w = \alpha \frac{k}{L}$. (d) Constant weighting $w = \alpha$.

to implement soft-constraints by combining the constrained pixel values in the soft constraints with the reconstruction in a weighted sum, replacing Equation (5.6) such that

$$R^k(p) = wT^k(p) + (1 - w) \sum_{q \in Q} \frac{s_{p+q}(n_b - q)}{N^2}, \quad (5.9)$$

where $w = \alpha^{\frac{2^k-1}{2^L-1}}$. The α term (typically $\alpha \leq 1$) controls the maximum blending weight at the lowest resolution in the pyramid, level L .

This weighting term w ensures that the effect of the soft constraints is halved at each successive level of the synthesis. At low resolutions (where k is larger) the soft constraints drive the general shape of the terrain, ensuring that large-scale structures are consistent with the artistic direction. At high resolutions (where k is smaller), the large-scale structures are inherited from the lower resolutions and the terrain details are almost entirely driven by reconstructing patches from the source-map, ensuring a greater degree of realism while being consistent with the general layout of the artistic direction.

Alternative weighting such as constant weighting $w = \alpha$, or a linear weighting $w = \alpha \frac{k}{L}$, influence the synthesis too much at the higher resolutions of synthesis and reduce the overall realism of the output. A comparison of the different weighting schemes is shown in Figure 5.2. With no weights, only the lowest level initialization reflects the constraints so the final result is dominated by the terrain structure from the source-map. The exponential weighting gives a good balance between the constraint and the structure from the source-map. The linear and constant weighting give too little scope for the source-map terrain and so have dramatically lower believability.

Hard Constraints

Hard constraints for the terrain-optimization algorithm allow the user to specify the values of pixels that are guaranteed to be in the final synthesis, with the rest of the synthesis forming around the constrained pixels. This was inspired by the

Figure 5.3: An example of using hard constraints. (a) Result of the synthesis using soft constraints, where some ridgelines are disconnected. (b) The input for the hard constraint synthesis, viewed as shaded relief. The hard constraints are shaded in yellow and the other regions are adjusted to connect the ridgelines and flatten the corners. (c) Result of the synthesis using hard constraints.

work of Wexler et al. [151], which addressed hole-filling in images with a texture-optimization approach. Hole filling is where there is a (mutable) region in an image that needs to be synthesized to fit with its (immutable) surroundings. Given that some pixels are immutable when hole filling, it follows that patches centered on immutable pixels should have a greater contribution to the pixels that are mutable, as it is the mutable pixels that need to conform with their surroundings. Wexler et al. achieved this by using a weighted mean when calculating the value for each pixel from the contributing patches.

To use hard constraints with the terrain-optimization algorithm, the user sets the values for the hard constraints in the input target-map T^k and provides an additional target-mask T_m that masks the hard constraints for pixels p . This mask is down-sampled exclusively (AND-ing the down-sampled values) to produce T_m^k for each pyramid level k . The E-step of the algorithm is modified to take into account the hard constraints by replacing Equation (5.6) with

$$R^k(p) = \begin{cases} T^k(p) & \text{if } T_m^k(p), \\ \frac{\sum_{q \in Q} w_{p+q} s_{p+q}(n_b - q)}{\sum_{q \in Q} w_{p+q}} & \text{otherwise,} \end{cases} \quad (5.10)$$

where $w_x = \gamma^{-d(x)}$ is a constraint weight for the pixel x with the Euclidean distance $d(x)$ from the pixel x to the nearest masked pixel in T_m^k , and γ controls the weighting distance falloff (I use $\gamma = 1.3$ default, as that is what Wexler et al. [151] recommend in their paper). This weighting controls the difference in contribution of patches s_p that are on or close to the hard constraints and patches that are further away from these constraints. A higher value of γ increases this difference, giving greater influence to patches that are on, or close to, the constrained pixels. This weighting provides a smooth transition from constrained to unconstrained regions of the synthesis.

Using hard constraints, the terrain-optimization algorithm supports copy-and-paste functionality similar to Gain et al. [61], where subsections of the source-map or other terrain are pasted and used as hard constraints for the synthesis. This increases the usability of the algorithm where artists may want specific terrain features or layouts present in the synthesized terrain, and do not have to rely on the soft-constraints reproducing the desired result.

Type Constraints

Type constraints, for the terrain-optimization algorithm, allow the user to have greater control over the synthesis by restricting the terrain used for parts of the desired output. This was inspired primarily from the “texture by numbers” approach by Hertzmann et al. [74] which was later explored with texture-optimization using the PatchMatch algorithm [10], and by the pixel-based terrain synthesis method of Gain et al. [61].

To use type constraints with the terrain-optimization algorithm, the user provides, as input, a source-type-map S_t , and target-type-map T_t . The source-type-map S_t gives type ids to each pixel in the synthesis and is used to identify the different patches that can be used for the synthesis given the type constraints. In the target-type-map T_t each pixel has a set of type ids that restrict what patches can be used from the example-map. The source-type-map S_t and target-type-map T_t are down-sampled to produce S_m^k and T_m^k , respectively, for each pyramid level k . The source-type-map S_t is down-sampled using the nearest-neighbour method, ensuring that each pixel only has a single type id, and the target-type-map T_t is

down-sampled inclusively (taking the union of the down-sampled values). The E-step of the algorithm is modified to take into account the type constraints by replacing Equation (5.7) with

$$C^k(p) = \arg \min_{s_p} (\|r_p - s_p\| \mid S_t^k(s_p) \in T_t^k(p)) . \quad (5.11)$$

This is implemented by ensuring that the PatchMatch algorithm only provides patches s_p for patches r_p such that the type id of the center of s_p specified in the source-type-map S_t^k , is also in the set of type ids in the target-type-map $T_t^k(p)$ ($S_t^k(s_p) \in T_t^k(p)$). I implemented this by creating a distance-map M_i^k for each source type id i , where the values for every pixel $p \in S^k$ are mapped to the closest pixel q that have the type id i . This is formalized as

$$M_i^k(p) = \min_q (\|p - q\| \mid S_t^k(q) = i) . \quad (5.12)$$

The PatchMatch algorithm is altered, so that when searching for a candidate patch that best matches the given patch r_p , the distance-maps can be used to speed up the search. Specifically, assuming the PatchMatch algorithm is considering a candidate patch x centered on some pixel $q \in S^k$ as a match for the given patch r_p , if the type id of the pixel q does not match any type id specified for the target pixel p ($S_t^k(q) \notin T_t^k(p)$), then the distance-maps M_i^k for $i \in T_t^k(p)$ can be used to find the nearest suitable replacement pixel for q as the center of a valid candidate patch. Through experimentation, I have found this optimization necessary when type ids for some value i are sparse in the source-type-map. Without it, the PatchMatch algorithm often fails to find more suitable candidate patches s_p as a best match for the given patches r_p , as they are stuck in a local minimum with respect to the energy function (5.5).

5.2.3 Summary

The core terrain-optimization algorithm produces terrain that is approximately as realistic as the current state-of-the-art example-based methods and contains a

number of constraint options (soft, hard, and type) which gives the user a greater degree of control for directing the synthesis of the terrain. My un-optimized implementation takes approximately five minutes to synthesize a 1000×1000 terrain. However, the largest performance bottleneck is the lack of parallelization for the PatchMatch algorithm which can be solved using either a CPU or GPU option [11]. Overall, the core terrain-optimization algorithm produces satisfactory results for a wide range of input examples, however there are areas where the algorithm can be improved to synthesize more realistic terrain. The algorithm suffers from some flaws similar to those identified in Section 3.2:

Interpolation: Values in the output height-map are synthesized solving a least squares equation, which interpolates the data between the contributing patches. The larger the difference between contributing patches the more the output is blurred, thus making the terrain less realistic.

Limitations using the data: The patches selected from the source-map are not rotated, not flipped, and are centered on integer coordinates, which limits the use of the available data.

User expertise: There is no process that ensures that the structure of terrain is realistic, so the algorithm relies on the user to guide the synthesis to producing terrain that is structurally correct.

I attempt to address these flaws in the next Section 5.3, by experimenting with adaptations of the core terrain-optimization algorithm.

5.3 Terrain Optimization Experiments

In addition to the core terrain-optimization algorithm presented in Section 5.2 I have also experimented with adaptations of the algorithm. The core algorithm suffers from some of the flaws shared with other example-based methods, as discussed in Section 5.2.3, such as interpolation, limitations using the data, and user expertise. The following section details my experiments adapting the core terrain-optimization

algorithm to address these flaws and improve the results of the terrain synthesis. Each experiment details the adaptation, how it addresses the flaws, and discusses the results.

5.3.1 Expanding the PatchMatch Search Space

The core terrain-optimization algorithm uses integer coordinates and a single orientation for the patches in the source-map. This limits the use of the real-world data provided, which restricts the algorithm to being transform-limited, as discussed in Section 3.2.6 for the flaw of limitations using the data. This limitation can be overcome by extending the correspondence search to use the Generalized PatchMatch algorithm proposed by Barnes et. al [11]. By increasing the space of patches that can be used to synthesize a terrain, it is also possible to find more coherent overlapping patches, reducing the magnitude of the difference between patches and the flaw of interpolation. In this section I explore the use of the Generalized PatchMatch algorithm to extend the range of transformations that can be used to retrieve patches from the source-map and evaluate its effect on the resulting terrain synthesized.

The Generalized PatchMatch algorithm describes an addition to the base PatchMatch algorithm that allows the method to search for a range of rotations θ and scales s , extending the search space of the original PatchMatch algorithm from (x, y) to (x, y, θ, s) . While scaling is not appropriate for transferring terrain features, as discussed in Section 3.2.6, the same principles are used to extend the algorithm to search for reflection transformations. The ability to transform patches using height-offsets is also included. The PatchMatch algorithm is extended to calculate the best height-offset for a possible candidate, and is stored as part of the final transform. This is similar to the gain and bias adjustments used in the Image melding texture synthesis [44]. The final result is a PatchMatch algorithm that includes a search space (x, y, θ, r, h) , of translation x and y , rotation θ , reflection r , and height-offset h .

I compared the results of the synthesis when the PatchMatch algorithm was extended by independently adding one of each of

Figure 5.4: An example of expanding the PatchMatch search space for the core terrain-optimization algorithm with no constraints. (a) Default (no extensions). (b) Continuous translation. (c) Discrete rotation. (d) Continuous rotation. (e) Reflection. (f) Height-offset. (g) Continuous translation, continuous rotation, and reflection. (h) All search space extensions.

Figure 5.5: An example of expanding the PatchMatch search space for the core terrain-optimization algorithm with soft constraints. (a) Default (no extensions). (b) Continuous translation. (c) Discrete rotation. (d) Continuous rotation. (e) Reflection. (f) Height-offset. (g) Continuous translation, continuous rotation, and reflection. (h) All search space extensions.

Figure 5.6: Plot of the mean energy function of each reconstruction of the optimization process R_i^k , where k is the level in the Gaussian pyramid and i is the number of reconstructions performed at that level. Data shown for the synthesized examples in Figure 5.5.

- continuous translation,
- discrete rotation (increments of $\pi/4$),
- continuous rotation,
- horizontal and vertical reflection,
- and height-offset,

as well as a combination of continuous translation, rotation, and reflection, and a combination of all extensions. Two examples of the comparisons are presented in Figures 5.4 and 5.5 using source data that contains many parallel ridges-lines and a mixture of flat and mountainous areas. To compare the effect of interpolation for the synthesis I plotted the mean energy for the terrains in the latter figure (plot shown in Figure 5.6).

Continuous translation did not significantly alter the realism of the output, but did have a lower energy function overall. While the effects may not be visibly noticeable, the lower mean energy indicates that the extension reduces the disparity between patches during the optimization process. This reduction aids in addressing the issue of interpolation, as the underlying data is modified to a lesser extent and retains more of its realism.

Discrete rotation ($\frac{\pi}{4}$) and continuous rotation provide similar synthesized results, where the ridges of the hills are oriented in many directions. The most noticeable difference is that the discrete rotation extension has the same mean energy as the default PatchMatch algorithm, but continuous rotation extension has a lower mean energy, suggesting that continuous rotation can find slightly better matches. It is likely that because incremental adjustments can be made (of values less than $\frac{\pi}{4}$) the PatchMatch algorithm converges to an optimal result more quickly, and with greater accuracy. The rate of convergence for discrete rotation could be addressed by running additional iterations of the PatchMatch algorithm, but the accuracy would still not be as great as continuous rotation. Better matches are also evident when comparing the results of using the default PatchMatch algorithm and using the extension of rotation, as shown in Figure 5.5. In the default synthesis, there are

broken ridgelines along the top-right and bottom-left of the circle because there are no ridgelines in the example-map that align with the user constraints. However, with rotation these ride-lines are unbroken due to the ability to rotate patches to better match the soft constraints.

Reflection, alongside the extension that combined continuous translation, rotation, and reflection, had the highest mean energy of all the extensions. This is likely due to the fact that reflection increases the search space, but minor adjustments to reflection are not possible because it is a binary adjustment, unlike translation and rotation. PatchMatch iteratively makes smaller changes to nudge the best estimation in the right direction, but for reflection, the adjustment of flipping a patch can drastically change the candidate making it harder to converge on an optimal result. However, like discrete rotation, additional iterations may aid PatchMatch in finding a more optimal correspondence. Adding reflection to the search space is also similar to rotation, in that it does not respect the alignment of features in the example-map.

Height-offset, alongside the extension that combined of all of the extensions, have the lowest mean energy of all the extensions experimented with. This is due to the fact that the height-offset provides the algorithm with a means to reduce the disparity between overlapping patches. In this way, patches are being selected based on their difference in shape, rather than their difference in absolute elevation. However, as seen from the results, the output is significantly different than other extensions, because the algorithm is able to match the low-lying areas of the target-map with the low lying areas in the source-map, resulting in many low flat regions. However, as discussed in Section 3.2.6, while some features of terrain can be raised or lowered without losing their realism, care needs to be taken to ensure that features are not offset inappropriately. In addition, while the energy function may have been lowered indicating better matches between overlapping patches, the final result of terrain may not be what the artist intended, especially when contrasted with the other extensions used for synthesis. This means that a height-offset transformation should only be used when appropriate to achieve the most realistic result (such as when similar features in the source height-map are found at varying elevations) or if it is required to create the users desired terrain.

In summary, the search space PatchMatch algorithm for terrain-optimization can be extended to ensure that the algorithm is not transform-limited. These extensions include continuous translation, discrete and continuous rotation, reflection, and height-offsets. For general use, the continuous translation extension should be used as it slightly reduces the mean energy but produces a visually similar result. For data that has rotational and reflectional symmetry, the rotation and reflection can be used if the user deems it acceptable, but neither are strictly necessary for synthesizing realistic terrain. However, if rotation and reflection are included, any alignment of the features in the example-map may be lost in the synthesized result. The height-offset extension as described should not be used. Although it lowers the mean energy function, it cannot guarantee that the resulting synthesis is realistic with respect to the example-map.

5.3.2 Pit-Removal

Similar to the previous data-based methods that are based on example-based texture-synthesis methods, the terrain-optimization algorithm requires user guidance to ensure that a realistic structure is synthesized. As such, the core algorithm produces terrain that contains many endorheic basins, reducing the overall realism of the result. Following the discussion of using pit-removal algorithms for data-based synthesis methods in Chapter 4, I have applied three pit-removal algorithms to the core terrain-optimization algorithm to address the flaw of user expertise. These three algorithms are depression-filling, depression-breaching, and a hybrid method.

The core terrain-optimization algorithm is modified in the following way. During the optimization process, after the E-step when the synthesis is reconstructed, a pit-removal algorithm is used to ensure there are no endorheic basins. This process is repeated for every iteration of optimization except at the highest level of synthesis $k = 0$, to ensure that additional artifacts from the pit-removal are created after the last iteration of the optimization. This application of the pit-removal algorithm is similar to the Height-offset application described in Section 4.2.1.

Figure 5.7: A comparison between the core terrain-optimization algorithm and the results of pit-removal extension, using the three different pit-removal algorithms, on an unconstrained synthesis. (a) The core terrain-optimization algorithm. (b) Terrain-optimization using the fill-algorithm. (c) Terrain-optimization using the breach-algorithm. (d) Terrain-optimization using the hybrid-algorithm.

Figure 5.8: A comparison between the core terrain-optimization algorithm and the results of pit-removal extension, using the three different pit-removal algorithms, with soft constraints. (a) The core terrain-optimization algorithm. (b) Terrain-optimization using the fill-algorithm. (c) Terrain-optimization using the breach-algorithm. (d) Terrain-optimization using the hybrid-algorithm.

Examples of the pit-removal application are shown in Figure 5.7 and 5.8. Similar to what was found in Section 4.2.1, I personally assessed the depression-breaching to produce the most realistic terrain, as depression-filling and to a lesser extent the hybrid-method, leave behind artifacts that reduce the overall realism of the terrain. The application of pit-removal works well for terrain-optimization, removing the endorheic basins while allowing the algorithm to optimize the synthesis for a more realistic result. Further extensions could be investigated to, not only ensure that the endorheic basins are removed, but ensure the terrain surrounding the breach channels are shaped to reflect the underlying processes that would have led to the creation of the breach channels.

5.4 Conclusion

The terrain-optimization algorithm is a novel approach to terrain synthesis based on the texture-optimization approach by Kwatra et al. [85]. The results of terrain-optimization are comparable in realism to other state-of-the-art data-based terrain synthesis methods. Through my experiments, I found that the most consistently realistic results came from the continuous translation extension to the PatchMatch search space and using the depression-breaching as pit-removal in conjunction with the optimization. This method is tested against other algorithms in Chapter 6.

Chapter 6

Subjective Evaluation of Realism

Evaluating the realism of terrain is non-trivial and an area of research that has not been thoroughly explored, despite the long history of terrain synthesis techniques. As discussed in Section 2.1.2 the word ‘realism’ is overloaded with meaning. I have defined the term ‘realism’ to be an objective measure and ‘believability’ to be a subjective measure, in order to distinguish the difference between analytical evaluation and subjective evaluation. As a clarification, despite the use of the term ‘realism’, when previous terrain methods have evaluated terrain (through a subjective evaluation or otherwise) the measurement is of the believability of the terrain, not realism.

This chapter seeks to explore previous attempts at evaluating terrain realism through believability and discuss both a new method of evaluation I have created and the results of that method. Section 6.1 presents an overview of previous methods for evaluating the realism of terrain including two subjective evaluation studies. Section 6.2 discusses how a subjective evaluation for terrain realism should be designed. Section 6.3 describes the experiment I have designed and discusses the results of that experiment. The key-findings include that no method was consistently as believable as real terrain, although there is an example for each method where it is indistinguishable from real terrains, and that no method consistently outperformed any other method. I also found that those with a high level of expertise in the field of physical geography were able to identify real

terrain more accurately from synthesized terrain, in contrast to any other level in any other category.

6.1 Previous Subjective Evaluations of Realism

The overwhelming majority of terrain synthesis algorithms have not had their realism validated through experimental study. As realism is a major part of an ideal terrain synthesis algorithm (see Section 2.1.3), it is important that there is some evidence to support the claim that a terrain synthesis method produces realistic terrain.

While there is a lack of experimental evaluation of realism, there have been many studies that include experimental evaluation of quantitative data such as time-efficiency and space-efficiency, and qualitative data such as user-experience. Without an experimental evaluation of realism, the assessment of realism is performed visually by the researchers. For example, in the state-of-the-art model-based terrain synthesis paper by Cordonnier et al., they state:

“Validation is a challenge common to all but the simplest simulation methods. While I included real images for comparison, it is difficult to quantify how closely the results match corresponding effects in nature. Our evaluation is only visual.” (Cordonnier et al. [38, p. 11])

While this quote references simulations (model-based methods) in particular, it is exemplary of many studies that only evaluate the realism of their method by having the researchers visually assess the output. This is not sufficient evidence for justifying the realism of a terrain synthesis algorithm for a number of possible reasons:

- The researchers assess the terrain in an informal and unstructured manner. There is often no information on how the terrain is displayed when it is being visually assessed, despite the rendering influencing a person’s ability to assess realism accurately (discussed in Section 6.2.2). Without a structured

format and the parameters of assessment, it is hard to justify if the evaluation is accurate.

- The researchers are most familiar with their own method and are generally motivated to develop a method that is as realistic as possible to them. This familiarity and motivation present a clear potential bias when researchers evaluate the realism of their own method in comparison to previous works or real terrain.
- The researchers are often not well versed in the field of physical geography, even though they have presented a contribution to the field of terrain synthesis methods. This may be especially true for the authors of approximation methods that do not consider the natural processes that shape real terrain.
- The researchers are a small sample size. With subjective evaluations like believability, the smaller the sample size, the less accurate the results of the evaluation will be. It also exacerbates any biases that the researchers may possess.

While these issues are prevalent in the majority of studies, there are some exceptions where the researchers have attempted to address some of these issues. One example is the work by Cordonnier et al. [37]. The paper was co-authored by an expert in geology who helped validate the design of the algorithm. The researchers validate the realism of their results by:

- visually comparing the terrain profile between the method and a geological simulation,
- comparing of various numerical aspects of the terrain, such as fold direction and wavelengths, between the method and a geological simulation,
- visually comparing between the results of the method and various real-height-maps sourced from the USGS [144] using a similar shaded relief map view,
- collecting comments of the realism in a qualitative survey given to users after they used the method.

Figure 6.1: An example set of terrain images used in the study by Tasse et al. [137]. (left) is real terrain T_{real} , (middle) is terrain synthesized using a previous deformation method T_{def} [60], and (right) is terrain synthesized using the method by Tasse et al. T_{sys} . (Image by Tasse et al. [137, p. 10])

Cordonnier et al.’s validation is more credible than methods that rely solely on the visual assessment of terrain by the researchers, but it still falls short of a quantitative analysis of realism. In a previous unrelated model-based terrain synthesis paper, Cordonnier et al. [36, p. 10] noted that they only assessed the visual plausibility of the output and that proper validation should be supported by a subjective evaluation. To my knowledge there are only two studies that have conducted a subjective evaluation of realism on terrain synthesis methods - one by Tasse et al. [137] and the other by Gain et al. [61]. While both Tasse et al. and Gain et al. refer to these studies as an “experimental user study” and “user experiment” respectively, this terminology is misleading. The participants in the studies are not users, as they are not using a system or service as part of the experiment, so I instead refer to them as subjects for consistency, as they are providing a subjective evaluation.

6.1.1 Tasse et al. Subjective Evaluation

Tasse et al. [137] conducted a subjective evaluation to test the believability of synthesized terrain. They had subjects compare terrain created by their system T_{sys} , against a previous related terrain deformation method by Gain et al. [60] T_{def} , and real-world terrain T_{real} (see Figure 6.2). Their hypotheses were that T_{real} is more believable than both T_{sys} and T_{def} , and that T_{sys} is more believable than T_{def} . The study was a within-groups design conducted with 20 subjects. Each subject was given the task to evaluate the realism of three sets of terrain. Each set

contained three terrain types: one real-world terrain (of type T_{real}), one terrain synthesized by the deformation method by Gain et al. [60] (of type T_{def}), and one terrain synthesized by the method of Tasse et al. (T_{sys}) using the real terrain and deformation constraints, used for the other terrains, as input. The evaluation involved ranking the three terrains in each set, in ascending order of realism, from least realistic to most realistic. I assume that the ranking performed by the subjects gave a rank of realism to each terrain in each set - either a 1, 2, or 3, (in ascending order of realism).¹ Each terrain was presented to the subject as a 3D render with a brown tinted shaded relief map, as shown in Figure 6.2.

The aggregated data for the three sets gave a sample of 60 rankings for each of the three terrain types: T_{real} , T_{def} , and T_{sys} . The Friedman test [59] found a statistically significant difference between the three groups at the 95% confidence level. As post-hoc analysis, Tukey's honestly significant difference (HSD) test [141] found that T_{sys} was more believable than T_{def} at the 95% confidence level ($p < 0.001$), but they found no evidence that there was a difference in believability between T_{sys} and T_{real} at the 95% confidence level ($p < 0.981$). They concluded that terrain generated through their method is not dissimilar to real terrain.

This study has some flaws that need to be addressed:

- The statistical analysis was performed on an aggregation of data for the three sets instead of separately for each set, confounding the results. This is addressed in Section 6.2.1.
- No effort was made to synthesize the terrain T_{def} and T_{sys} to create a terrain similar in structure to the corresponding T_{real} (as shown in Figure 6.2), creating a 'feature-bias'. This is addressed in Section 6.2.3.

6.1.2 Gain et al. Subjective Evaluation

Gain et al. [61] also conducted a subjective evaluation on the realism of synthesized terrain. They had subjects compare real-world terrain against terrain synthesized

¹This assumption is made on the educated guess that this is the most likely scenario for having used the Friedman test for analyzing the differences between rankings.

Figure 6.2: An example of synthesized terrain images used in the study by Gain et al. [61]. (Images by Gain et al. [61, p. 116])

by their method. The hypothesis was that if the synthesized terrains are as realistic as real-world terrain, then the Bernoulli distribution of the results should be similar to a sequence of coin flips. The study was a within-groups design conducted with 25 subjects with a background in computer science but no expertise in physical geography at the tertiary level. Each subject performed a two-alternative forced choice (2AFC) task [56, 147] where they were presented with a randomized sequence of 60 terrain image-pairs, one real-world terrain (sourced from the USGS [144]) and one terrain synthesized using their new method. The synthetic terrains were created by a pair of expert users² from a database of 19 real-world exemplars, without using copy-pasting constraints. Each terrain was presented to the subject using two 3D viewpoints: overlooking and close-up, and rendered using a brown tinted model, with elevation contour lines and radiance scaling [146] to emphasize curves.

The aggregated data for the 60 terrain image-pairs sets gave a sample of 681 votes for the real-world terrain and 819 votes for the synthesized terrain. The binomial test indicated that the proportion of synthesized terrain being more believable than real-world terrain .546 was higher than the expected .5, $p = .0002$ (1-sided). From this result, the coin flip hypothesis was rejected showing that synthetic terrains were actually considered more believable than the real-world terrains. Through a post-experiment questionnaire, Gain et al. attributed the unusual result to subjects misidentifying prominent features in the real-world terrains, such as sharp ridges and river bends, as unrealistic. There was also anecdotal evidence that subject

²The expert users were James Gain, and Patrick Marias, the authors of the paper. This was confirmed in personal correspondence with James Gain, 9 May, 2019.

experts (those more familiar with physical geography) would be more likely to identify real-world from synthetic terrain.

This study has some flaws that need to be addressed:

- The statistical analysis was performed on an aggregation of votes from the 60 terrain image-pairs instead of separately for each pair, confounding the results. This is addressed in Section [6.2.1](#).
- An unorthodox method of rendering was used to display the terrains that may have influenced the subject's perception. This is addressed in Section [6.2.2](#).
- No effort was made to synthesize terrains that were similar in structure, or from a similar source to the real terrain used in the comparison pair,³ creating a 'feature-bias'. This is addressed in Section [6.2.3](#).

6.1.3 Summary

While the majority of previous research shows little effort to validate the realism of terrain synthesis methods, the work of Cordonnier et al. [37], Tasse et al. [137, p. 10], and Gain et al. [61] have been steps in the right direction. Cordonnier et al. employed the help of an expert in physical geography to guide the design of their method and confirm the visual appearance of the synthesized terrain to geological models. Tasse et al. ran a small experiment where subjects ranked their synthesis method, a previous synthesis method, and real terrain, in terms of realism. Gain et al. ran a similar experiment where subjects evaluated their method against real terrain, in a series of pair-wise comparisons. An experimental evaluation of terrain synthesis is currently the most viable way to validate realism in terrain, but there are many issues that must be addressed, including the design of the experiment and analysis, the presentation of terrain, and the presence of 'feature-bias'. The next Section ([6.2](#)) discusses each of these issues and proposes potential solutions.

³This was confirmed in personal correspondence with James Gain, 9 May, 2019.

6.2 Experimental Design

6.2.1 Testing Procedure and Analysis

Believability is an arbitrary subjective measurement and as such is unsuited to be assigned an *absolute* numerical value. It is impractical to make relevant measurements of believability to decide which of two terrains is preferable. Instead it is more suitable to be used as a *relative* measurement where one terrain would be judged to be more believable than another terrain, where the magnitude of the difference is not quantified. Here, three issues are discussed with the experimental procedure using human subjects to evaluate terrain realism using the *relative* measurement of believability: testing methodology, subject expertise, and result analysis.

Testing Methodology

To establish a relative measure of believability, subjects in the experiment must use comparative judgment, where terrains are evaluated using direct comparisons to establish the relative believability between terrains. For evaluating a single synthesis method, only direct comparisons against real-world terrain is needed per set, as demonstrated by Gain et al. [61] who performed multiple discrete pair-wise comparisons with terrain generated by their method against real-world terrain. With this method, a terrain synthesis algorithm can only be evaluated to synthesize terrain either less believable, as believable, or more believable than real terrain - not the magnitude that the synthesized terrain and real-world terrain differ in believability. For evaluating multiple synthesis methods, testing can be done with or without the inclusion of real-world terrain. If there are more than two terrains being evaluated, a comparison needs to be done between all of the methods (and the real-world terrain if it is included) within each set. This was demonstrated by Tasse et al. [137] who asked subjects to rank the given sets of terrain on the basis of realism.

An alternative to the ranking methodology of Tasse et al., is to use pair-wise comparison between each pair of methods to establish the relative believability between terrains. This holds many benefits:

- With pair-wise comparisons, the subject is making a decision based solely on the pair of images presented, without outside influences. The presence of additional terrain images when using a ranking method may affect the decisions of how a subject ranks a pair of images.
- The subject only needs to compare a single pair at a time allowing the greatest use of limited display area, maximizing the display size of each terrain image presented. It also reduces the mental effort of comparison for the subject because they only need to directly compare two images at once, as opposed to multiple images when doing a ranking.
- Ties can be allowed. Without ties (e.g. in a ranking, or a 2AFC test) the experiment relies on a close to equal number of responses for each choice to produce a result that both choices are statistically equal. With a small sample size, this can skew the results.

However, the ranking methodology has the advantage of reducing comparison fatigue, as fewer individual comparisons are required to provide a ranking for large sets of images. Overall, I believe using pair-wise comparisons is more beneficial for measuring the believability of terrain, because the flexibility allows it to be used for both single method, or multiple method comparisons, and allows ties that reduces the statistical noise introduced when subjects are forced to make to make a ranked choice.

Subjects and Expertise

Considering that human subjects are required to evaluate the realism of terrain, there are considerations that need to be made about the subject's ability to assess the terrain accurately and fairly, in order to achieve the best result. Influences can

include the subject's expertise in a certain field, their association to the research, and the clear communication of the study procedure.

The study by Gain et al. [61, p. 113] provides anecdotal evidence that subject experts (of physical geography and cartography) are more likely to have a higher accuracy when identifying real terrain from synthesized terrain. The idea is that subjects more familiar with terrain and its underlying processes have an understanding that helps them differentiate real and synthesized terrain. With education about the nature of terrain, a subject can rationalize and discern to a greater degree the accuracy or inaccuracy of a presented model. Without education, the subject is ignorant of the nature of terrain and unrealistic models may still appear plausible. This also extends to recognizable terrain where knowledge of common terrain shapes, or famous natural landmarks may influence a subject's perception of what is realistic terrain, resulting in a 'feature-bias' (discussed further in Section 6.2.3).

The most common user of terrain synthesis methods are visual artists, who regularly assess the realism of the synthesized terrain. While not all artists are familiar with modelling terrain, or using terrain synthesis methods, their discerning perception when it comes to assessing visual quality of an image may have an effect when assessing the realism of terrain. As Micheal McGuire noted in his photographic essay:

"Artists have always been keen observers of the natural world, ..." (Michael McGuire [96, p. 261])

It is possible to make a case that experts in the field of visual arts maybe as, if not more discerning, when it comes to the evaluation of realism for terrain. The same case could also be made for subjects that do not have experience in the visual arts, but have experience analyzing images, such as researchers in the field of computer graphics. However, a study would need to be conducted to confirm these predictions.

The subject's association with the research can also cause a potential researcher-bias when evaluating terrain. Those who are already familiar with the terrain generated by a certain method would evaluate the terrain knowing that it is synthesized, and

thus not real. In a pair-wise comparison experiment where the pairs contain one real and one synthesized terrain, the subject who is familiar with the synthesized terrain will be able to differentiate the two terrains without assessing for realism. Moreover, researchers directly involved in the research and who benefit from the believability of their method are potentially biased to ensure that terrain created by their method is more believable than other terrain in the study. This is an even larger issue when there is no formal experimental evaluation, as researchers can claim that their method is realistic, without doing any fair comparative evaluation. This situation is avoided by ensuring that the evaluation of realism is completed by impartial subjects who are unfamiliar with the synthesized terrains and unrelated to the research being conducted.

In previous studies the concept of realism is poorly defined, and for the purposes of an experimental evaluation it needs to be clearly communicated to the subjects of the experiment. In my experience, many subjects are naive to the concept of realism in terrain and it needs to be established what is expected when a subject is to judge the realism of terrain. Without this communication, subjects may select the most aesthetically pleasing image, or use some other unrelated metric for evaluating a terrain image. In my experience, this is especially important when evaluating the comparative realism of terrains that are unbelievable (are perceived to be unrealistic), as the evaluation still needs to be conducted to the definition of realism, as opposed to aesthetics.

In both previous subject evaluation studies, the subject demographic data, such as age and gender, is not collected. This additional information helps validate the results of an individual study because it reports the representation of the group of subjects.

Result Analysis

The analysis of the data collected is just as important as the testing methodology. From the data collected, by either pair-wise comparisons or ranking lists, a terrain can be established to be: less believable, as believable, or more believable than another terrain. The magnitude of the difference of believability between terrains is

not a measurement of how much more believable a terrain is, but rather the confidence with which that terrain is more believable.

With multiple sets of comparisons, the data should be analyzed by looking at the results of believability of each set. If a terrain from a source a is more believable than a terrain from source b in the majority of the sets, then the conclusion should be that a produces more believable terrain than source b . However, if the data for all sets are aggregated before the analysis, the results can be confounded.

For example, let N be the number of pairs of terrains produced from two sources a and b , and V_n^x be the number of times the terrain from the source $x \in a, b$ was voted more realistic than the other, for the pair $1 \leq n \leq N$. Let $\Omega(y)$ be the number of times that y is true, for $1 \leq n \leq N$. Assume the results of analyzing individual pairs show that $\Omega(V_n^a > V_n^b) > \Omega(V_n^a < V_n^b)$, leading to the conclusion that source a produces terrain that is more believable than source b , because it is voted to be more realistic in a greater number of the pairs. Now suppose that the results of the analyzing the aggregation of votes show that $\sum_n^N (V_n^a) < \sum_n^N (V_n^b)$, leading to the conclusion that source b produces terrain that is more believable than source a because it has a greater number of total votes for being more realistic. These conflicting results are possible in situations where the magnitude of difference between V_n^a and V_n^b are smaller on average when $V_n^a > V_n^b$, but larger on average when $V_n^a < V_n^b$.

When comparing terrains, the likely-hood that one terrain is more believable than another terrain for a given population is being measured. The magnitude of the difference in votes is not quantified as a measurement of how much more believable one terrain is over another, but the confidence that one terrain is more believable than another. This makes it inappropriate to aggregate the data before the analysis, as past terrain realism studies have done.

In the study by Tasse et al. [137], the statistical analysis was performed on aggregated data instead of individually on the three sets. This aggregation confounds the results, because T_{def} being more believable than T_{sys} in one set but less believable in the other two sets could possibly give a statistical similar result, depending on the magnitude of differences in the rankings for each set. The same

argument could be made about the conclusion that T_{sys} is similar in believability to T_{real} .

The study by Gain et al. [61] also aggregated the data, where the number of votes for real-world terrain and synthesized terrain were summed before the analysis. This aggregation confounds the results, because it is unclear if the synthesized terrain was consistently more believable than real terrain by a small margin for each pair, or if it was significantly more believable in less than half of the pairs, and slightly less believable in more than half. As Gain et al. did not perform an analysis on the individual pairs, and did not provide the data they collected, it is unclear how well the method performs when the data is not aggregated for analysis.

Without an analysis of the original data for both previous studies, the validity of their conclusions can only be questioned, not falsified. However, performing analysis on aggregated data, rather than the sub-populations, remains a flawed process for drawing conclusions about the believability of the sources that produce terrain and should be avoided in future studies.⁴

6.2.2 Terrain Visualization

Assessing the believability of terrain is a purely visual task, meaning the choice of how to visualize terrain is of great importance. My focus is on the static representation of a heightmap where no additional data is available (e.g. material composition, vegetation etc.). There are a number of options for displaying a height-map:

Two-Dimensional (2D) vs. Three-Dimensional (3D)

Terrain height-maps are typically viewed as either 2D or 3D representations, both of which have a variety of different visualization options.

A height-map can be seen as a function $T(p) = z$, where p is some position on the height-map and z is the elevation at that position. When visualizing the height-map

⁴This was validated by the statistical consultant at Victoria University of Wellington, 16 May, 2019.

Figure 6.3: A comparison between 2D and 3D visualizations, for shaded and unshaded height-maps. (a) A 2D view of a grey-scale map is a typical way to visualize a terrain. (b) A 2D view of a shaded relief map provides information about the surface of the terrain, but not the elevation. (c) A 3D view of a height-map mesh with a grey-scale map texture does not convey elevation information accurately, as the depth of the image is lost in the lack of detail in the texture. (d) A 3D view of a height-map with a shaded relief map texture, conveys elevation accurately because the shading gives the image a greater sense of depth.

in 2D, an image can be created where the pixel value at position p is some function of the height-map as a whole $I(T, p)$. This could be a colour based on the elevation z at that pixel $T(p)$, or some function of the elevation relative to the height-map as a whole (this is discussed later). The benefits of a 2D visualization include:

- maximizing the use of visual real-estate (as long as the height-map is rectangular),
- every sample of elevation in the height-map is visible and is viewed at the same resolution,
- distances between positions remains constant across the image,
- little variability in construction, making it consistent to produce similar image types for separate height-maps.

However, with a 2D visualization it is difficult to compare relative elevations or get a sense of scale from the top-down view, especially as this is not how viewers typically encounter terrain. These issues can be mitigated by adding symbols, as discussed later, but they are not as intuitive as a 3D view.

A 3D visualization is typically a rendered view of a height-map with elevations and distance between elevations scaled correctly. The benefits of a 3D visualization include:

- providing a more intuitive understanding of the relative elevations and a sense of scale for the entire terrain presented,
- large flexibility, allowing the visualization to show certain angles and features.

While the intuitive nature of a 3D visualization is beneficial for a subject, there are a large number of considerations to be made when constructing a 3D view. However, there are many cases where the vertical scaling of terrain has been exaggerated for visual effect.

The shading of the terrain must be considered as it is critical to giving the 3D visualization a sense of depth. Figure 6.3 highlights the need for shading in a 3D

view. The variability of how to visualize the height-map in 3D makes it more difficult to replicate equivalent images between studies. Variations include, the camera (position, rotation, and field-of-view), the construction of the height-map model (discrete mesh or implicit surface), the terrain skirt, the shading model used for rendering, and the image resolution. There is also the consideration that a 3D visualization may not show the entire height-map, either because some elevations are occluded from view, or because they are not in frame. These issues could be addressed with the use of an interactive model for viewing the terrain, however, the current study will focus on static representations due to the ease of application for online testing purposes.

The studies by Tasse et al. and Gain et al. both use 3D visualizations instead of 2D for showing terrains to their subjects. Tasse et al. used a height-map mesh with a terrain skirt, textured with a shaded relief map coloured with a brown hue. Gain et al. used two images for each terrain, one 3D view overlooking the entire terrain, and one 3D view closer to the terrain. Neither study provides the exact details of how the 3D views were set up, making it difficult to replicate equivalent images for new studies to set up the same experimental conditions. Their choice of colour and shading are discussed next.

Colour and Shading

There are a number of different ways to colourize a height-map for visualization, including grey-scale maps, hypsometric maps, shaded relief maps, and photo-realistic rendering.

Grey-scale maps represent the elevation as a gradient of shades, with black as the lowest elevation and white as the highest, as shown in Figure 6.4a. In computer graphics, height-maps are most often represented through grey-scale maps. A major downside to this style of visualization is the limited colour depth, with a typical range of 0 to 255, presenting a low fidelity representation of the elevation. It is also difficult to visually distinguish shape and curvature, especially in dark areas of the map.

Figure 6.4: Examples of different terrain colouring and shading. (a) A normalized grey-scale map. (b) A hypsometric map. (c) A shaded relief map. (d) A combination of hypsometric and shaded relief map created by multiplying them together.

Figure 6.5: An example of how realistic lighting and texturing can visually hide artifacts and skew the subject's perception of realism. Refer back to Figure 3.13 for a visualization of the artifacts present in this height-map. Image by Zhou et al. [160].

Hypsometric maps are coloured maps that use gradients where the colour can represent a range of elevations, as shown in Figure 6.4b. Neighbouring elevation ranges often have neighbouring colours in the spectrum, presenting a smooth transition between adjacent elevation levels similar to a gradient. The extra dimensions of hue and saturation greatly increase the range of elevations representable compared to a grey-scale map and thus present a higher fidelity representation of the height-map. While hypsometric tints are widely used and there are many popular gradients, they have not yet been fully standardized [100, p. 172]. The colourization of maps has a great deal of effect on how the map is perceived and interpreted. People not only respond emotionally to colour, but also create associations between the perceived colour and assumptions about the terrain. Common associations include, hues of green with vegetation, blue with water, red with high temperatures, and yellow with desert environments [100]. These associations may complicate the interpretation of the type of terrain displayed, such as when an elevation map of a desert is coloured green, or a mountainous forest is coloured yellow. Colour blindness is also a consideration when using colours to represent elevation data, as the information conveyed by the hue of the hypsometric tint may not even be as informative as a grey-scale map for some combinations of tints and subjects with certain types of colour blindness.

Shaded relief maps (also known as hill-shading), are created by computing a simple illumination of the height-map from a directional light source, as shown in Figure 6.4c. It provides the reader with a more accurate display of the relative shape and curvature of the terrain through shading and can be used by itself or in conjunction with a coloured map (like a hypsometric map as shown in Figure 6.4d). There are many variations to shading, including lighting models, adjustments to the azimuth and altitude of the light, and multi-directional lighting, as well as different ways of combining with other types of maps. The directional light source is typically located on the top-left of the map, (i.e. an azimuth of 315° , Northwest) to avoid the relief inversion effect, which gives the appearance of inverted elevations. The relief inversion effect is widely acknowledged in the geographic community and a recent study by Biland and Çöltekin [25] systematically confirmed the effect through a subjective evaluation. They also found that an

azimuth of 337.5° (North-northwest) created a higher accuracy than the current standard of 315° (Northwest) for landform identification. However, the effect is not currently well understood for 3D renderings, and it is uncertain whether the lighting azimuth has any effect on interpretation of elevation.

Photo-realistic rendering is an alternative method of visualizing a height-map, with many different approaches to make the final output look as real as possible. It is typically combined with a 3D view to make full use of rendering techniques such as shadow-mapping and atmospheric scattering. Commercial software, such as Terragen [118], can also add additional detail onto the terrain through the use of procedural texturing and tessellating the geometry with fractal-based detail. However, it is likely that this additional detail can distract subjects from evaluating the realism of the height-map. Cordonnier et al. [36, p. 10] mentioned that the main difficulty of comparing synthesized terrain against real-world terrain is that only the structure of the terrain, not the details, should be evaluated. Rendering software can also be used to smooth the terrain before rendering, which can eliminate the visual artifacts that may be present in the original height-map data. This makes realistic rendering, in general, a poor choice for evaluating the realism of a terrain height-map. For example, Figure 6.5 shows an example, from the paper by Zhou et al. [160], of how realistic rendering can hide artifacts from the synthesis method (see Figure 3.13 for a visualization of the artifacts).

Tasse et al. [137] use a form of shaded relief map with a brown hue for shading the 3D models they present in their images. They do not provide the parameters they used to generate their images and it is unclear if the lighting azimuth was consistent for each image in their experiments as the shading angle varies between images in their paper. Gain et al. [61] use the un-orthodox method of radiance scaling [146], with a brown hue base colour, to emphasize features with high curvature. This choice was made as part of the design for the creation interface to avoid colour interference with the creation interface while maintaining a proper sense of shape and scale. However, it is unclear how this rendering method affects the subject's perception of realism when evaluating terrain as the lighting is dissimilar to the real world and may be unfamiliar to the subject. In comparison, shaded relief maps

Figure 6.6: An example of the use of symbols on a height-map. (a) Contour lines on a hypsometric map with a contour interval of 100m. (b) Grid lines on a hypsometric map at 1000m intervals. (c) Two types of grid scales, line and box, on a hypsometric map.

and photo-realistic renderings are lit similarly to the real world and may be more familiar to the subject.

Having tried various methods and combinations I decided that a combination of a hypsometric map and shaded relief map gave the best static representation of the elevation to the subject. While there is the issue that subjects may assume additional information about the terrain due to the hypsometric tints (e.g. vegetation, temperature), the benefit of having additional information about the elevation outweighs the potential bias that it may introduce, especially for a 2D visualization.

Symbols

In geography, additional symbols aid in the interpretation of elevation and scale in a height-map. For the composition of static, images I considered contour lines, grid lines, and scale bars as the most obvious potential additions (see Figure 6.6).

Contour lines (in cartography) are lines that join points of equal elevation. The contour interval is the elevation difference between adjacent contour lines. The contour interval should be the same over a single map and the shapes of the lines can be used to discern additional information about the elevation. For instance,

sharp-pointed v-shapes are usually found in stream valleys, with the drainage channel passing through the tip of the v-shape with the v-shape pointing upstream (as a consequence of erosion). Closed loops are normally downhill on the outside and uphill on the inside, with the innermost loop being the highest area. The spacing of the contours also provide a sign of the steepness of an area, with close contours indicating a steep slope and distant contours indicating a shallow slope. When two or more contours merge, it indicates a cliff with a height greater than the contour interval.

Grid lines are part of a grid reference (in cartography) and used to define locations in maps using Cartesian coordinates. They also help readers judge relative distances between terrain features by providing a frame of reference. The most common form is a square grid, usually with sizes tied to a specific measurement (e.g. latitude-longitude, kilometers). This frame of reference is especially important in cases where large areas of terrain are being displayed using non-cylindrical projections.

Scale bars (also known as bar scales or linear scales) are marked boxes or lines, with labels, indicating the scale of the map. They are often used in conjunction with a representative fraction when the map is 'drawn to scale' but can still be used on their own when that is not the case. The intention of a scale bar is that the user of the map can use a divider compass (or even two fingers) to measure a distance between two points by comparing it to the labeled marks on scale bar. The addition of a scale bar provides a frame of reference, similar to grid lines, but comes with some downsides. If the scale bar is placed outside the map the entire image will require more space (with limited space this will require the map to be displayed smaller) and if it is placed inside the map it will occlude part of the map (especially opaque box scale bars).

Gain et al. [61] uses both contour and grid lines in a very subtle manner. From the images provided in their paper the contour lines are a slightly darker shade than the terrain and the grids are not visible. From my tests, thin contour lines that are dark shades can be misinterpreted as ridges when combined with other maps such as the shaded relief map. From my informal experimentation, I decided that contour lines provide a large benefit to the observer as long as they are appropriately coloured

and their thickness does not obstruct or distract from the rest of the map. I did not find that grid lines or scale bars provided any additional benefit for judging the realism of terrain (as long as the size of the terrain is known beforehand) and that their addition may just complicate the image.

Multiple Diagrams

Using multiple diagrams for testing provides many benefits, primarily communicating more information to the subject without overcrowding a single image with detail and allowing the subject to see the terrain using images that could not be combined otherwise (such as both a 2D and 3D view). However, there are a few downsides to using multiple diagrams including the increase of total image area, variability of testing, and conflicting information.

Assuming that the visual real-estate available for the terrain images is limited, increasing the number of images decreases the maximum resolution that all images can be viewed at once. For instance, two non-overlapping equal sized images presented together can only be displayed, at most, half the resolution of a single image. With a lower resolution, terrain details are harder to identify which decreases the subject's ability to evaluate elevation data. A digital medium can solve this by allowing the subject to expand and shrink images, switch between images, or scroll through a list of images. However, there is still a possibility that by not having all of the terrain images in view may affect the subject's ability judge the realism of terrain, as they can not see and compare all of the available information at once.

Additional images present increased variability in the testing method and make it harder to compare results across experiments that have similar testing methodologies. This variability increases with the number of images used for a single terrain and overall makes it harder for the analysis to conclude whether certain displays make a difference in evaluation of realism for the terrain.

Presenting multiple images may introduce a conflict in the way a subject evaluates the realism of terrain if the believability of the individual images differs significantly. Multiple images give the subject an opportunity to view the terrain

from different perspectives, highlighting features that may not be visible in a single format. However, in cases where the believability of the images differ, it may interfere with the experimental evaluation, especially when doing pair-wise comparisons. For example, consider two terrains T_1 and T_2 and their respective terrain images, X_1 and X_2 in one format and Y_1 and Y_2 in another format. If the believability of images differ by type, $b(X_1) > b(X_2)$ and $b(Y_1) < b(Y_2)$, the subject may find it difficult to evaluate which terrain, X_1 or X_2 , is more realistic. This introduces the variability that the subject bases their judgment on their preferred image type (either X_i or Y_i) rather than the terrain that the images represent.

Gain et al. [61] used two 3D images for each terrain in their study: a complete overview of the terrain, and a zoomed in view of the terrain. Both 3D views were rendered using the same method, minimizing the differences in believability between the two images.

6.2.3 Terrain Feature-Bias

As I defined in Section 2.1.2, believability is not a measure of realism, it is the perception of realism evaluated by an observer. For a test to evaluate the relative realism through believability, the test must be constructed in a way that for any pair of terrains being compared, T_i and T_j , that if one is as real or more realistic than the other, $r(T_i) \geq r(T_j)$, then the same should hold true for believability, $b(T_i) \geq b(T_j)$. The validity of a test is compromised if this is not true when the test relies on assumption that believability is indicative of realism.

Feature-bias is an effect where the terrain features of a height-map have an effect on the believability of the terrain independent of the realism. For example, 'typical terrain' features such as volcanic cones and/or dendritic river networks make the terrain more believable than terrain that is absent of those features, even if the terrains are equally realistic. That is, if $r(T_i) = r(T_j)$ and T_i contains 'typical terrain' but T_j does not, then feature-bias would create an effect such that $b(T_i) > b(T_j)$. This feature-bias therefore invalidates a test where the assumption is made that believability is indicative of realism.

In this section I attempt to, through an experimental study, show the presence of feature-bias by pair-wise comparisons of a set of real terrains.

Feature-Bias Study

The study was a repeated-measures (within-groups) design where the subjects performed a forced choice multiple paired comparison task through an online survey (which can be found in Appendix A). The task involved being presented with a random sequence of 21 pairs of terrain images and selecting the terrain in each pair that was the most realistic. The 21 pairs of images consisted of all combinations of size two, from a set of seven terrain images (as shown in Figure 6.7): three real height-maps contain ‘typical terrain’ features, three real height-maps that do not contain ‘typical terrain’ features, and one height-map synthesized using a fractal-based method (approximation method). The six real terrain height-maps were selected as a diverse range of terrains created by various natural processes. There were 36 subjects between 19–62 years of age (M 37, SD 10.9). 26 of the subjects were male, 11 were female, and 1 was genderqueer, all of whom stated they were free from any serious visual impairments. Subjects took part in the study willingly through an online survey and were not given any compensation for their participation.

I hypothesized that the height-maps that contained more ‘typical terrain’ features (recognizable features such as volcanic cones, or dendritic river networks) would be more believable than height-maps that contained less or no ‘typical terrain’ features. I also hypothesized that the synthesized height-map would be more believable than terrain that contained no ‘typical terrain’ features.

The results and analysis of the pair-wise comparisons is presented in Table 6.1. The table shows the number of times a terrain was evaluated to be more realistic than the other terrains, the total sum of votes, the Bradley-Terry score π_i for each terrain, and the magnitude of the significant differences between each pair of terrains. The sum of the votes gives a good indication of the ordering of believability for the terrains, where the more believable a terrain is, the larger its sum of votes. The main issue

Figure 6.7: Terrain used in the realism pilot study. (a) Mount Sundoro and Mount Sumbing, Central Java, Indonesia. (b) Serranía del Aguaragüe, Tarija, Bolivia. (c) West Făgăraș Mountains, Romania. (d) Sand dunes in the Southern El Djouf region of the Sahara Desert, Mali. (e) Northwest of Gobernador Gregores, Santa Cruz Province, Argentina. (f) West of Mount Woodroffe (Ngarutjaranya), Australia. (g) Fractional Brownian motion created with Perlin noise.

T_i	More realistic than							Total	π_i	Significance						
	c	a	f	b	e	g	d			c	a	f	b	e	g	d
c		21	29	30	34	36	38	188	1.0000			***	***	***	***	***
a	17		29	27	31	34	38	176	0.5007			***	***	***	***	***
f	9	9		18	31	34	37	138	0.3186	***	***			***	***	***
b	8	11	20		24	31	37	131	0.2746	***	***			**	***	***
e	4	7	7	14		29	37	98	0.1342	***	***	***	**		***	***
g	2	4	4	7	9		26	52	0.0421	***	***	***	***	***		***
d	0	0	1	1	1	12		15	0.0115	***	***	***	***	***	***	

Table 6.1: Results of the feature-bias study and analysis using the Bradley-Terry model with Holm–Bonferroni correction. Significance values are interpreted as the confidence that the terrains are different at a given level: ‘.’ $p < 0.1$, ‘*’ $p < 0.05$, ‘**’ $p < 0.01$, ‘***’ $p < 0.001$.

is where the statistically significant differences are between the given methods. For this I used the Bradley-Terry model [29].

The Bradley-Terry model is a widely used method for deriving a latent preference scale from paired comparison data, as shown in the extensive bibliography by Davidson and Farquhar [45]. The Bradley-Terry model assumes that there are values $\pi_i \geq 0$ for each terrain T_i , such that for any pair of terrains, T_i and T_j , the probability that terrain T_i is voted more realistic than terrain T_j is modelled as

$$P(b(T_i) > b(T_j)) = \frac{\pi_i}{\pi_i + \pi_j}. \quad (6.1)$$

The Bradley-Terry model finds the π_i values that best fit the data collected though the pair-wise comparisons and gives p-values for differences between every pair. The scale factor for the series is unimportant, so for convenience I scaled the values so the largest $\pi_i = 1$. Additionally, I used the Holm–Bonferroni method [76] to control the family-wise error rate by correcting the p-values. This accounts for Type I errors when doing multiple comparisons with the same data.

The Bradley-Terry model with Holm–Bonferroni correction shows that there is a significant difference between five sets of terrains at the 95% confidence level. The

sets in order of most believable to least believable are $\{T_a, T_c\}$, $\{T_b, T_f\}$, $\{T_e\}$, $\{T_g\}$, and $\{T_d\}$.

The results from this analysis provides evidence of a feature-bias that influences subject's judgment of realism when evaluating terrain. The terrains T_a and T_c , with the most typical terrain features, were the most believable terrains. T_b and T_f , contain less typical terrain features but still have recognizable structures, and were the next most believable. T_e , with an unusual structure was the second least believable real-world terrain. And T_d , containing little to no identifiable features, ranked the least believable (even below the synthesized terrain T_g). Despite every terrain (excluding T_g) being equally realistic, due to them all being existing real-world terrain (see Section 2.1.2), there is a clear ordering of realism where terrains with more 'typical terrain' features were evaluated to be more realistic than terrains without those features.

How does this affect the evaluation of synthesized terrain? Let there be two height-maps of real terrain, R_a and R_b , such that the believability of R_a is greater than R_b , ($b(R_a) > b(R_b)$). Suppose there is a data-based method that synthesizes terrain as realistic as the example terrain provided to it and it is used to synthesize two height-maps, S_a and S_b , from the example data R_a and R_b respectively. It then follows that $b(S_a) = b(R_a)$ and $b(S_b) = b(R_b)$. However, if the realism of the synthesis method were to be evaluated by comparing dissimilar terrain, the conclusion may be that the synthesis method is more believable than real terrain ($b(S_a) > b(R_b)$) or less believable ($b(S_b) < b(R_a)$) depending on the comparison made in the evaluation of believability. This contradicts the actual believability of the synthesis method. Therefore, to avoid this, only like terrains should be compared when evaluating the relative realism of terrain.

Feature-Bias Summary

Feature-bias is present when comparing dissimilar terrains based on their realism. Both Tasse et al. [137] and Gain et al. [61] do not control for feature-bias in their experimental studies. In the study by Tasse et al., it appears that no effort was made to synthesize T_{def} and T_{sys} to create a terrain similar in structure to the

corresponding T_{real} . In the test set they provide in the paper (shown in Figure 6.2), the example of T_{def} is an atypical structure surrounded by a flat region and the example of T_{sys} contains a similar structure surround by more typical terrain. The study by Gain et al. was similar, where the synthetic terrains were not modeled to match layout or structure of the real-world terrains that they were compared against.⁵

One possible way to reduce the effect of feature-bias when evaluating synthesized terrains, through pair-wise comparisons, is to compare terrains that are similar in structure and contain similar terrain features. To compare a synthesized and a real terrain, you would control the artistic direction of the synthesis method towards replicating the real terrain. For comparing two synthesis methods, you would control the artistic direction of both methods to produce a similar result, regardless of whether it imitates a real (existing) terrain or not.

In addition, for data-based methods, you should not use the terrain you are trying to replicate as the source for the generation because features could be copied exactly, for example, with the correct selection of patches (Tasse), or with copy-paste constraints (Gain, Scott). One solution is to have two height-maps, one target terrain for the terrain synthesis methods to replicate, and one source used as the example data for input, where the target and source height-maps are similar in terrain features. This could be achieved using real height-maps that are located nearby geographically.

6.3 Realism Subjective Evaluation

This study seeks to examine the believability of several data-based terrain synthesis methods by asking subjects, including a substantial proportion whom have expertise in geomorphology, to evaluate the realism of the terrain methods and real terrain using pair-wise comparisons. This study includes four methods of data-based terrain synthesis:

- Tasse (algorithm and implementation by Tasse et al. [137])

⁵This was confirmed in personal correspondence with James Gain, 9 May, 2019.

- Gain (algorithm by Gain et al. [61], implementation by me)
- G-Pit (algorithm presented in Section 4.2.1 with depression-breaching)
- Scott (algorithm presented in Section 5.3.2 with depression-breaching)

I omitted including the algorithm by Zhou et al. [160] in the test for four reasons. First, there is no available implementation of the algorithm and the creators could not be reached to obtain a working implementation. Second, there are algorithm details in the paper that are ambiguous (see Section 3.1.1), meaning assumptions must be made to implement the algorithm which may not be faithful to the original intention of the authors. Third, after making assumptions in my implementation I was not able to reproduce results comparable to the output provided by Zhou et al. Finally, the work of Tasse et al. [137] extends that of Zhou et al. and I am using an implementation of Tasse et al.'s algorithm as one of my methods to compare.

Based on personal evaluation I hypothesized that the methods G-Pit and Scott would be considered more believable than the methods Gain and Tasse, and just as believable as real terrain. I also hypothesize that subjects in the experiment with a greater expertise of physical geography, cartography, image analysis, or visual arts, will be able to more accurately identify real terrain from synthesized terrain, than non-experts.

6.3.1 Design

The study used a repeated-measures (within-groups) design around the pairwise comparisons of seven sets of terrain images. Each set used a different set of constraints and source terrain as input for each of the terrain synthesis methods (with the exception of Set 6 and 7 which used different constraints but the same source terrain). Sets 1–5 contained five images per set (10 pairwise comparisons): four images created from the synthesized terrains, and one image created from real terrain. Sets 6–7 contained four images per set (6 pairwise comparisons): all created from synthesized terrains. The independent variable was a pair of terrain images to be evaluated (with 62 pairs). The dependent variable was the response of which of

the terrains in the pair of terrain images was more real, with three options: “left is more realistic”, “both are equally realistic”, and “right is more realistic”. Artistic direction of the algorithms was controlled by using similar constraints for each synthesis method within each set. There were several extraneous variables that may have affected this study: the subject’s familiarity with source or target terrains, subject’s visual condition, and the window size the survey was conducted with.

6.3.2 Subjects

There were 245 subjects who completed the online survey, which was advertised through mailing lists to the geography and visual effects community and through lectures at Victoria University of Wellington. Subjects were asked to contribute to the research by completing the survey and no compensation was given. None of the researchers associated with the experiment, or those who had viewed the stimuli beforehand, participated in the survey. Of the completed responses there were 149 males between 18–69 years of age (M 34, SD 11.3) and 96 females between 18–68 years of age (M 33, SD 12). 6 subjects indicated that they had some form of red-green colour blindness but were included in the final results as it was not considered a serious enough visual impairment to be excluded.

6.3.3 Apparatus and Materials

The source and target terrains for Sets 1–5 were selected to be similar in elevation range, structure arrangement, and terrain features (see Figure 6.8). The scale and resolution of the source, target, and synthesized height-map were a resolution of 1000×1000 pixels with a sample spacing of 1 arc-second (approximately 30 meters). Source terrain for Sets 6–7 was a resolution of 2000×2000 pixels with a sample spacing of 1 arc-second, but the synthesized height-map remained at a resolution of 1000×1000 pixels. See table 6.2 for coordinates of the height-maps used for the source and target terrains. Set 1 terrain contains an incised plateau with dry riverbeds and a high drainage density. Set 2 terrain contains an alpine post-glacial landscape with glacial cirque valleys and rivers that are only just

i	Region	Source		Target	
1	East-central Yemen	16.167°N	48.139°E	16.167°N	48.417°E
2	Rocky Mountains, CO, USA	39.139°N	106.583° W	39.417°N	106.583°W
3	Făgăraș Mountains, Romania	45.695°N	24.583°E	45.695°N	24.861°E
4	Central Java, Indonesia	7.389°S	110.083°E	7.500°S	110.445°E
5	Ngarutjaranya, Australia	26.278°S	131.500°E	26.278°S	131.778°E
6,7	Marlborough, New Zealand	41.722°S	173.722°E	n/a	

Table 6.2: Approximate region and latitude-longitude coordinate (in decimal degrees) for center of the real-data used for the source and target terrain for each terrain Set i . Set 6 and 7 share the same source terrain and have no real-world terrain target.

Figure 6.8: An example of selecting a source terrain and target terrain with similar elevation range, structure arrangement, and terrain features. In this example, I used elevation maps from Central Java, Indonesia. The source height-map for Set 4 (blue/left) contains Mount Sundoro and Mount Sumbing. The target height-map for Set 4 (red/right) contains Mount Merbabu and Mount Merapi. Both maps contain two neighbouring stratovolcanoes with a similar layout and pattern of erosion.

starting to incise. Set 3 terrain contains a mix of erosional and depositional landscapes, where the hills are part of an uplifting front fault range that transitions into coalescing alluvial fans along a mountain front. Set 4 terrain contains young neighbouring stratovolcanoes with a radial drainage pattern. Set 5 terrain contains a degraded ancient landscape of inselbergs, featuring relict river valleys and recent aeolian erosion. Set 6 and 7 terrain contains a tectonic modified threshold landscape, with fault bounded mountain ranges and strong parallel structural control.

The artistic direction for synthesis methods within each set was controlled by using roughly equivalent constraints for each method. Sets 1–5 used constraints to reproduce the target terrain in the set, Set 6 used no constraints, and Set 7 used constraints to reproduce an artificial structure. For the constraints of Tasse, I used the target terrain as input directly, as the algorithm automatically extracts the ridgelines and valley-lines it required for the target feature-map through the use of PPA. Providing the target terrain directly also provided Tasse with the correct elevations and noise variance required as part of its feature-matching process. I also ran the algorithm numerous times with different patch sizes (feature and non-feature) and PPA feature type (ridge, valley, or both) before selecting the best result to use in the study. The other parameters were left to their default values, set by the original authors of the code. For the constraints of Gain, I created a program to aid in the specification of the constraints. I was able to trace the ridgelines and prominent features of the target terrain, and the program created curve constraints using the traced path, extracting the correct elevation and gradient values. I then manually corrected the area-of-effect for the constraints and other small aspects of the constraints, and the final result was a set of curve constraints that matched the target terrain. The same constraints that were created for Gain were also used for G-Pit. For the constraints of Scott, I down-sampled the target terrain four times to a resolution of 62×62 pixels, then up-sampled it to 1000×1000 pixels, to use as the input for soft-constraints. This emulated the rough input an artist would use to guide the algorithm to produce the correct shape of terrain, without specifying specific details.

A terrain texture was created for each terrain height-map consisting of a combination of a hypsometric map and a shaded relief map. A colour ramp by MOSSMAN [101] was used to create the hypsometric map, with discrete colours at equally spaced intervals of 100m, from 100m to 4800m. A shaded relief map was generated with QGIS (v3.6.0-Noosa) [120] using an azimuth of 315° and an altitude of 45° , and was multiplied (at 66.7% opacity) with the hypsometric map. The 2D image, was created by overlaying thin blue contours (0.5px width) generated at every 100m on the terrain texture. The 3D image was rendered as an unlit scale model with skirt that extended to sea-level and textured with the terrain texture. The camera, for the 3D view, was positioned at an altitude of 50° , a distance from the model center of 30,000m, and with a field of view of $\pi/3$. The final image for each terrain was created by stacking the 2D image above the 3D image. All terrain images can be found in Appendix B.

Qualtrics [121] was used to create an online survey which consisted of four sections:

- Participant information sheet, containing the information required by the Victoria University of Wellington Ethics Committee, informing the subject about the study and their rights when participating before asking for the subject's consent to participate in the study.
- Terrain information, containing information about the how the study will be conducted, the definition of realism, and an example and explanation of the terrain images that will be presented.
- Pair-wise comparisons, a section of 62 questions broken up into eight separate pages of six to eight questions per page. The order of and left-right orientation of images in the 62 questions were randomized. Each question presented the pair of terrain images above a horizontal radio button with three options: 'left is more realistic', 'both are equally realistic', and 'right is more realistic'.
- Participant questionnaire, a section of questions asking for the subject's name, age, any visual impairments, and expertise in physical geography, cartography, image analysis, and visual arts. There were also two optional questions asking for feedback: what their strategy was to determine the most

realistic terrain and what image they found most useful in the multiple images provided for each terrain.

Subjects required a computer with access to the internet to complete the survey, and were encouraged to complete it on a device with a large screen as opposed to a small mobile device. A copy of the survey can be found in Appendix B.

6.3.4 Procedure

The survey was emailed to potential subjects through various mailing lists, asking if the recipient would like to participate. The survey was received by Victoria University of Wellington students and research groups, local visual effects companies, and an international mailing list of professional geographers. Data from the survey was collected over a period of 21 days, from when the online survey was first released to when no more responses were accepted. Subjects with unfinished surveys, or who indicated they had a visual condition that would greatly affect their results, were rejected from the final analysis.

6.3.5 Results

Tables 6.3, 6.4, and 6.5, contain sub-tables showing the analysis of comparing the seven sets individually as well as the combination of Sets 1–5 and Sets 1–7 (nine sub-tables total), using the Bradley-Terry model. The each sub-table shows the number of times a terrain produced by a method was voted more realistic than a terrain produced by another method, the sum of votes and the Bradley-Terry score, and the magnitude of the significant differences between the terrain produced by each method. As in Section 6.2.3 I used the Bradley-Terry model with Holm–Bonferroni correction to analyze the pair-wise comparison data. Because I included a third option that specified a draw, I interpreted each draw vote as 0.5 votes for each terrain in the pair. This analysis with the Bradley-Terry model provides an ordering of believability for the terrain synthesis methods in each set, and their statistical similarities and differences to other methods.

Terrain Set 1

Method	Voted more realistic than					Total	π_i	Significance				
	Gain	Real	G-Pit	Tasse	Scott			Gain	Real	G-Pit	Tasse	Scott
Gain		113.0	117.5	211.5	194.5	636.5	1.0			*	***	***
Real	132.0		124.0	115.0	220.5	591.5	0.8512				***	***
G-Pit	126.5	120.0		121.5	211.5	579.5	0.8086	*			***	***
Tasse	31.5	128.0	122.5		208.5	490.5	0.5876	***	***	***		***
Scott	49.5	21.5	32.5	34.5		138.0	0.1316	***	***	***	***	

Terrain Set 2

Method	Voted more realistic than					Total	π_i	Significance				
	Real	Scott	Gain	G-Pit	Tasse			Real	Scott	Gain	G-Pit	Tasse
Real		172.5	169.5	180.5	186.0	708.5	1.0		***	***	***	***
Scott	69.5		123.5	167.0	172.5	532.5	0.5243	***		.	***	***
Gain	73.5	119.5		120.0	174.5	487.5	0.4463	***	.		.	***
G-Pit	63.5	75.0	123		173.5	435.0	0.3704	***	***	.		***
Tasse	57.0	69.5	67.5	69.5		263.5	0.1965	***	***	***	***	

Terrain Set 3

Method	Voted more realistic than					Total	π_i	Significance				
	Real	Scott	Tasse	Gain	G-Pit			Real	Scott	Tasse	Gain	G-Pit
Real		120.5	138.0	225.0	227.0	710.5	1.0			***	***	***
Scott	123.5		138.5	213.0	217.0	692.0	0.9251			***	***	***
Tasse	106.0	104.5		186.5	190.0	587.0	0.5997	***	***		***	***
Gain	18.0	31.0	56.5		123.0	228.5	0.1337	***	***	***		
G-Pit	17.0	26.0	53.0	119.0		215.0	0.1251	***	***	***		

Table 6.3: Results of the subjective evaluation for terrain Sets 1, 2, and 3, ordered by the Bradley-Terry score π_i . Significance values are interpreted as the confidence that the terrains are different at a given level: ‘.’ $p < 0.1$, ‘*’ $p < 0.05$, ‘***’ $p < 0.01$, ‘****’ $p < 0.001$.

Terrain Set 4

Method	Voted more realistic than					Total	π_i	Significance				
	Tasse	Real	Gain	G-Pit	Scott			Tasse	Real	Gain	G-Pit	Scott
Tasse		122.5	145.0	150.5	171.5	589.5	1.0			***	***	***
Real	120.5		148.5	153.0	164.0	586.0	0.9885			***	***	***
Gain	99.0	95.5		128.0	144.0	466.5	0.6603	***	***			***
G-Pit	93.5	90.0	115.0		147.5	446.0	0.6168	***	***			***
Scott	71.5	80.0	99.0	95.5		346.0	0.4371	***	***	***	***	

Terrain Set 5

Method	Voted more realistic than					Total	π_i	Significance				
	Real	Scott	Tasse	Gain	G-Pit			Real	Scott	Tasse	Gain	G-Pit
Real		181.5	164.5	194.5	201.5	742.0	1.0		***	***	***	***
Scott	61.5		127.5	158.0	164.5	511.5	0.426	***		**	***	***
Tasse	79.5	114.5		115.0	122.0	431.0	0.3215	***	**			**
Gain	48.5	85.0	127.0		125.5	386.0	0.2755	***	***			
G-Pit	42.5	78.5	122.0	115.5		358.5	0.2487	***	***	**		

Aggregation of Terrain Sets 1–5

Method	Voted more realistic than					Total	π_i	Significance				
	Real	Tasse	Scott	Gain	G-Pit			Real	Tasse	Scott	Gain	G-Pit
Real		724.0	859.0	869.5	886.0	3338.5	1.0		***	***	***	***
Tasse	493.0		668.5	545.5	654.5	2361.5	0.5091	***		*	*	***
Scott	356.0	544.5		643.0	676.5	2220.0	0.4629	***	*			**
Gain	348.5	668.5	574.0		614.0	2205.0	0.4582	***	*			**
G-Pit	333.0	563.5	538.5	599.0		2034.0	0.4075	***	***	**	**	

Table 6.4: Results of the subjective evaluation for terrain Sets 4, 5, and the aggregation of Sets 1–5, ordered by the Bradley-Terry score π_i . Significance values are interpreted as the confidence that the terrains are different at a given level: ‘.’ $p < 0.1$, ‘*’ $p < 0.05$, ‘**’ $p < 0.01$, ‘***’ $p < 0.001$.

Terrain Set 6

Method	Voted more realistic than				Total	π_i	Significance			
	G-Pit	Gain	Tasse	Scott			G-Pit	Gain	Tasse	Scott
G-Pit		139.0	165.0	193.5	497.5	1.0		**	***	***
Gain	104.0		159.5	180.5	444.0	0.7802	**		***	***
Tasse	78.0	84.5		135.5	298.0	0.4118	***	***		***
Scott	49.5	62.5	107.5		219.5	0.2885	***	***	***	

Terrain Set 7

Method	Voted more realistic than				Total	π_i	Significance			
	Tasse	Scott	G-Pit	Gain			Tasse	Scott	G-Pit	Gain
Tasse		140.5	173.0	194.0	507.5	1.0		.	***	***
Scott	103.5		177.0	187.5	468.0	0.8287	.		***	***
G-Pit	70.0	66.0		154.5	290.5	0.3721	***	***		***
Gain	49.0	55.5	88.5		193.0	0.234	***	***	***	

Aggregation of Terrain Sets 1–7

Method	Voted more realistic than				Total	π_i	Significance			
	Tasse	Scott	Gain	G-Pit			Tasse	Scott	Gain	G-Pit
Tasse		944.5	824.0	905.5	2674.0	1.0			*	**
Scott	755.5		893.0	903.0	2551.5	0.9308				
Gain	877.0	810.0		806.5	2493.5	0.8998	*			
G-Pit	798.5	798.0	892.5		2489.0	0.8973	**			

Table 6.5: Results of the subjective evaluation for terrain Sets 6, 7, and the aggregation of Sets 1–7, ordered by the Bradley-Terry score π_i . Significance values are interpreted as the confidence that the terrains are different at a given level: ‘.’ $p < 0.1$, ‘*’ $p < 0.05$, ‘**’ $p < 0.01$, ‘***’ $p < 0.001$.

From this analysis, for individual Sets 1 to 5, reality is always the most believable or it is statistically indistinguishable from the most believable method. This is a positive result that shows that no method has produced terrain that is more believable than reality. Interestingly, a result of every method is statistically indistinguishable (at the 99.9% level) from reality for at least one terrain set: Tasse in C_4 , Gain and G-Pit in C_1 , and Scott in C_3 .

Gain and Gain-pit were statistically indistinguishable at the 95% confidence level in four of the seven sets. The most notable difference is in the two artificial terrain sets, 6 and 7, where Gain-pit is more believable. Grouping Gain and G-Pit together, the results show that all pairs of methods have three wins, three losses and one tie. There is no one method that has generated the most believable, or least believable, terrain in all cases, rejecting my hypothesis that G-Pit and Scott would be more believable than the Tasse and Gain in all cases.

From the analysis on the aggregation of Sets 1–5 and 1–7 (where the votes are combined before the analysis), the methods are less distinguishable from one another in terms of believability. For Sets 1–5, Real is the most believable, followed by Tasse, then by Scott and Gain which are equally believable (at the 95% confidence level), and G-Pit as the least believable. For Sets 1–7 all four methods are almost indistinguishable from one another statistically, with the exception being Tasse is more believable than both Gain and G-Pit (at the 95% confidence level). This shows that the aggregation of data across the comparative terrain confounds results from the individual terrain sets.

I compared the accuracy of each level of expertise in the categories of physical geography G_i , cartography C_i , image analysis I_i and visual arts A_i (distributions shown in Figure 6.9 and a description each expertise level is shown in Table 6.6). Accuracy was calculated as the fractional value of the number of times a subject indicated a real-world terrain was more realistic than a synthesized terrain, divided by the number of pairs that contained a real terrain (M 0.6963, SD 0.1635). A single factor ANOVA was computed separately for each category of expertise. ANOVA for the physical geographic expertise ($F(3, 241) = 13.55, p < 0.001$) shows that there are statistically significant differences at the $p < .05$ level. ANOVA for the cartography expertise ($F(3, 241) = 3.83, p = 0.0104$) shows that there are

User Expertise	
Level	Description
Physical geography (knowledge of underlying processes)	
G_1	At most, a high-school education.
G_2	Currently in a bachelor's degree or equivalent.
G_3	Completed a bachelor's degree or equivalent.
G_4	Completed a postgraduate degree or working in the field.
Cartography (ability to interpret maps)	
C_1	Cannot read a map.
C_2	Some previous experience reading a map.
C_3	Can read a map well.
C_4	An expert at reading maps.
Image analysis (experience identifying artifacts in images)	
I_1	No experience.
I_2	Some previous experience.
I_3	Regularly has to identify artifacts in images.
Visual arts (experience participating in the visual arts)	
A_1	No experience.
A_2	Some previous experience.
A_3	Regularly participates in the visual arts.

Table 6.6: A description of the levels for each category of expertise.

Figure 6.9: A box plot of the distribution of accuracy for the independent variables of physical geography G_i , cartography C_i , image analysis I_i , and visual arts A_i . The mean accuracy for the tested population is marked with a grey line.

statistically significant differences at the $p < .05$ level. ANOVA for the image analysis expertise ($F(2, 242) = 3.80, p = 0.0237$) shows that there are statistically significant differences at the $p < .05$ level. ANOVA for the visual arts expertise ($F(2, 242) = 0.22, p = 0.8025$) shows that there are no statistically significant differences at the $p < .05$ level.

A post-hoc analysis using Tukey's HSD test [141] revealed the following statistically significant differences at a 95% confidence level (see Figure 6.10). First, for physical geography expertise, G_4 is statistically significantly different from any of the other three levels, but there is no statistically significant difference between any pair of the other three levels. Next, for cartography expertise, C_4 is statistically significantly different from C_2 , but that there is no statistically significant difference between any other pair of levels. The reason that C_4 and C_1 cannot be distinguished is likely owing to the small number of subjects in C_1 ($N_{C1} = 10$), which makes the confidence intervals substantially wider for comparisons against C_1 than for comparisons that do not include C_1 . Finally, for image analysis expertise, I_3 is statistically significantly

Figure 6.10: Ranges for the Tukey HSD test for the different categories of expertise. A test where the range includes zero is not statistically significant at the 95% confidence level. A test where the range excludes zero is statistically significant at the 95% confidence level. Source data is in Table C.5.

different from I_1 , but there is no statistically significant difference between either other pair of levels. For visual arts expertise, the ANOVA showed no significant differences between any pair of levels, so it is unnecessary to run the Tukey's HSD test for this category.

Out of the 245 subjects, 180 provided feedback on their strategy for determining the most realistic terrain (which can be found in Appendix D), and all 245 provided their preferred terrain image types. 30 preferred the 2D view, 108 preferred the 3D view, and 107 preferred having both the 2D and 3D view of the terrain, showing that either a 3D view, or a combination of 2D and 3D, was the most preferred visualization for the purpose of determining realism.

6.3.6 Discussion

From the results, it is evident that no method could synthesize terrain that was consistently as believable as real terrain or more believable than another method's synthesized terrain. This mixed result provides evidence that making a comparison between terrain synthesis methods is a difficult task, even when many variables are accounted for. The two contributing factors in this study that lead to this mixed result are likely to be, the terrain type (the source and target for the terrain synthesis) and the user constraints (the control over the synthesis).

In order to get a fair representation of the capabilities of the synthesis for each method I selected a range of different terrains that were varied in geographical features. I did not include terrains that contained large bodies of water such as lakes and seas, as this would expand the level of control each method would need to make to get a fair result. Gain, G-Pit, and Scott, all have type constraints that would be used to ensure that bodies of water would be placed correctly, whereas Tasse does not. If this was the case I would have to make the decision between allowing type constraints, which would bias against Tasse because the algorithm would treat the body of water like a flat plane of terrain, or to disallow type constraints, which would be biased against Gain, G-Pit and Scott for not using features of the algorithms when they would most likely be used.

The variability and complexity of each terrain most likely contributed to the difference in results between the methods. For instance, in Set 1, the terrain contained a lot of ridgelines. Scott performed poorly, with the overall terrain looking blurred compared to the other methods, but Gain performed as well as real-world terrain. In comparison, in Set 3, the terrain contained a combination of hills and plains. Scott terrain performed as well as real-world terrain, but Gain performed poorly, with blurred hills that had little to no definition.

To control both the feature-bias and the artistic direction of the study, I created constraints for each method to synthesize the same structure presented in the real-world target terrain for each set. This was an important aspect of the study, as I established in Section 6.2.3 that feature-bias can severely affect the results of believability for a subjective evaluation. However, creating equal constraints for each method was a challenging task as the inputs for Tasse, Gain/G-pit, and Scott all differ. There are two major issues I identified while creating the constraints for each method, over-constraining the synthesis and user expertise. Over-constraining the synthesis is an issue that applies to both Gain/G-pit and Scott, where the number, and the strictness, of the constraints reduces the algorithm's ability to be assessed on realism. As discussed in Section 2.1.3, the ideal terrain synthesis algorithm is one that *"produces terrain that is as real as possible, while meeting user specified constraints"*. To assess the realism of an algorithm while accounting for feature-bias, an algorithm should have as few and as lenient constraints as possible, needed to replicate the structure of the target terrain. User expertise is also an issue, where the expertise of the user creating the terrains can impact the highest level of realism that can be assessed for each algorithm. As discussed in Section 3.2.1, the parameterization of Tasse is a problem, where a large amount of the user's time is spent tuning parameters to produce the best result. I decided that I would only control the patch sizes and feature type, and leave the rest of the parameters as defaults. I also did not have the same 3D modelling interface for Gain/G-pit that the original authors specified, instead opting for an approach that gave me an approximated set of constraints which I had to manually edit. In both cases, if I were the original author of the code and more experienced with the algorithm, the potential for synthesizing more realistic

results for each algorithm would be increased. As such, both the issues of over-constraining the synthesis and user expertise had effect on how the terrains were synthesized in each set.

Another important observation is the difference between the analysis of the data when performed separately for each set, and when performed on the aggregation of the sets. From the analysis for each separate Set 1–5, it is clear that no algorithm was consistently more believable than any other algorithm. But in the aggregated Sets 1–5, the analysis shows that Tasse is more believable than Scott and Gain, which are more believable than G-pit, which contradicts the conclusion that I drew from the individual sets. Moreover, conflicting conclusions are also drawn when comparing the two aggregated sets. For Sets 1–5, Scott and Gain are both more believable than G-pit, but when two more sets (Sets 6–7) are aggregated for Sets 1–7, there is no difference in believability between the three methods, owing to G-pit performing well on the two artificial sets. This gives clear evidence that there is a difference between aggregating the analysis of sets and analyzing the aggregation of sets, supporting the argument made in Section 6.2.1 that the aggregation of this data does not give a clear indication as to which method produces more realistic terrain.

The results also indicated a significant difference in the accuracy (the success rate of identifying real terrains from synthesized terrain) between subjects based on their levels of expertise in certain categories. In line with the anecdotal evidence provided by Gain et al. [61], subjects who are experts in the field of physical geography had a higher accuracy, than non-experts. Surprisingly though, this is only true for subjects who have a post-graduate degree or who are currently working in the field of physical geography (G_4). Subjects who are currently enrolled in (G_2), or have completed (G_3), a bachelor's degree or equivalent were indistinguishable from non-experts (G_1) in accuracy. Likewise, subjects that are well versed in map-reading (C_4) had a higher rate of accuracy than subjects with little experience reading maps (C_2). However, as stated in the Results, the reason that C_4 and C_1 cannot distinguished is likely due to the small sample of subjects tested that cannot read a map (C_1). This can also be visually confirmed by the size of the 95% confidence interval for each pair of values that included C_1 in Figure

6.10. Furthermore, there was a difference in accuracy for image analysis expertise, where subjects who regularly have to identify artifacts (I_3) had a higher accuracy than subjects that have no experience identifying artifacts (I_3), but surprisingly there was no difference between subjects with different levels of expertise in the visual arts (A_1 , A_2 , and A_3).

From these results I conclude that the most qualified subjects to evaluate terrain are experts at the highest level in the fields of physical geography, cartography and image analysis. In fact, the mean accuracy of subjects with high expertise in physical geography was the highest of any level in any group, indicating that the most qualified individuals overall to assess the realism of terrain are experts in the field of physical geography. As a result, future studies should take into account this expertise of their subjects to ensure that the results of their subsection evaluation of realism is accurately portrayed.

Because experience in the visual arts is not an indicator that a subject is able to accurately distinguish between real and synthesized terrains, it brings up a point about importance of user expertise when employing terrain synthesis methods. As discussed in Section 3.2.7, one issue concerning user expertise is the ability to accurately assess the realism of synthesized terrain, where users who are unable to accurately assess the realism of terrain are more likely to synthesize terrain that is less realistic than possible with a given method. As I have found, visual arts expertise is not an indicator of accuracy when identifying real from synthesized terrains, indicating that methods that rely on the user's assessment of realism are not suitable for an industry of artists with little to no expertise in physical geography. Instead, terrain synthesis methods should be developed to ensure that the user's accuracy is irrelevant to synthesizing the most realistic terrains possible.

The written feedback regarding the subjects' strategies was informative and insightful regarding what they were looking for when evaluating the realism of terrain. The complete set of feedback is in Appendix D. I have identified three main points from this feedback:

Consistency Terrain features that were unrelated, or were repeated, were considered unrealistic.

Detail Terrain features that were too smooth or too sharp were considered unnatural.

Drainage The presence of closed (endorheic) basins and the disconnection of drainage networks, were considered unrealistic.

Many of the subjects' strategies (both experts and non-experts) evaluating terrain involved looking at the drainage network and for the presence of closed basins, which supports my argument in Chapter 4 about how the presence of endorheic basins in synthesized terrains reduces the overall realism. This argument is also supported by the results for the terrain Sets 6 and 7, where pit-removal is needed more in order to remove the endorheic basins and synthesize more realistic terrain. This is supported by the result that G-pit was more believable than Gain in both sets, where the only difference between the methods is the integration of a pit-removal algorithm.

6.3.7 Future Work

While this study had the largest number of participants to date, and collected a large amount of data, there are many ways that it could be improved for future use. To reduce the variability of the study I controlled many aspects of the synthesis, including how the terrain was displayed, the terrain data format, the types of real-world terrain used, the constraints used for the synthesis, and the number of terrain sets tested. These variables were controlled to limit the scope of the study, avoiding complications in drawing conclusions from the collected data and to keep the quantity of required comparisons reasonable for the subjects. However, future studies could investigate terrain realism further by manipulating these variables.

To display the terrain to the subject I used a combination of a 2D view with contour lines and a 3D view, both with a shaded relief map blended with a hypsometric map. Feedback from the subjects show some had difficulty discerning the type of environment the terrain was based in, which may have stemmed from the influence of the hypsometric tint I chose to represent elevation. Because of this, they were not able to fully apply their knowledge of physical geography to analyze

the terrain presented. Alternative methods of blending shaded relief and hypsometric maps are worth investigating to improve the subjects' interpretation of the elevation data. For example, using cross blended hypsometric tints that do not fall prey to confusion between vegetation and lowland areas, such as the method by Patterson and Jenny [112]. When there is no extra information available shading by vegetation or humidity is impossible, but if the information from the source data is collected, this technique could possibly be used for future terrain synthesis studies.

In addition, I only used static 2D and 3D views of the terrain, whereas during my own research and analysis of terrain synthesis I have the ability to orient and interpret the data anyway I see fit. This could be an area of improvement where future studies might not just present multiple views of the terrain but provide the subject with an interactive application that allows them to navigate the terrain as they see fit. Careful consideration would need to be made to ensure that each subject can intuitively navigate the scene equally, but this freedom would ensure that the evaluation of the terrain uses the full expertise of the subjects.

For the terrain data format, I used terrain resolution of 1000×1000 pixels with a sample spacing of 1 arc-second (approximately 30m). A 1000×1000 pixel resolution was selected, because it is similar to the standard size for most examples of terrain synthesis, large enough to prevent pixellation and to contain a variety of terrain features, but small enough that the terrain easy to analyze and fast to synthesize. I used 1 arc-second because of its availability through surveyors like the USGS [144]. Future studies could test methods using alternative resolutions and sample spacing, especially when there is a wide variety of both the availability of real-world data and applications of the synthesized terrain in the industry.

The terrain targets and sources for Sets 1-5 and the source for Sets 6 and 7 involved a variety of different terrain types, but none of them contained large bodies of water, such as lakes, seas, or the ocean. The decision was made to limit the scope of the study in consideration of the wide range of methods that I was testing. Future studies should investigate a terrain synthesis algorithm's ability to synthesis coastlines, which is a significant feature of terrain that has not yet been thoroughly researched as a target of terrain synthesis.

The study was designed to have each method synthesize similar terrains using similar constraints, controlling the variability of artistic expression and the presence of feature-bias. However, the types of constraints vary significantly between the methods and the expertise of the user and the decision of how constrained the terrain synthesis should be, play a huge role in the overall outcome. For instance, an expert user might be able to provide more suitable constraints for Gain than I have been able to, as I had to implement the method myself and have used it only for generating a few examples. To control for this familiarity with the algorithm, future studies may have to ensure that the constraints set for each method are provided by an impartial user who is equally experienced with the methods being tested. This could involve employing novices or experienced artists or even a mixture of both.

Finally, the length of the survey was carefully considered. A pilot study for the survey revealed that each pair of terrains took an average of 20 seconds to evaluate, meaning each additional set of terrains with a real-world example (10 pairs) would add an additional 3 minutes and 20 seconds to the average survey time. Even with 62 pairs (an estimated completion time of 20 minutes and 40 seconds plus time to read the participant information sheet, the terrain information sheet, and fill out the participant information questions), I had just as many unfinished survey responses as finished responses. Many of the subjects who participated complained about the length of the survey, both in personal correspondence and in the feedback, with particular concern about the survey being too long and repetitive. Evaluation fatigue is common in multiple paired comparison testing, especially with more samples. This needs to be taken into consideration for future studies to ensure that the subjects are engaged long enough to evaluate the terrain to the best of their ability rather than out of boredom. One option is to increase the variety of terrain and reduce the number of methods that are tested for each set.

6.4 Conclusion

Evaluating the realism of a terrain synthesis method is a difficult task. The majority of studies do not perform an experimental evaluation of realism, and those that do

have some significant flaws. I have explored and discussed different methods of evaluating the realism of terrain synthesis methods through the measure of believability, as well as addressing the problems of existing studies.

I developed and ran an experimental evaluation of four data-based terrain synthesis methods, Tasse, Gain, G-Pit and Scott. I found that no method was consistently as believable as real terrain, and that no method consistently outperformed any other method. The experiment also showed that those with an expertise in the field of physical geography were able to identify real terrain more accurately from fake terrain, providing a more accurate view of the difference between methods. There was no indication that expertise in visual arts provided a higher accuracy.

The study provided an in-depth look at realism in terrain synthesis methods and can be used as the foundation for future research into the experimental evaluation of realism in terrain synthesis methods.

Chapter 7

Conclusion

In this dissertation I have reviewed and classified the previous methods of terrain synthesis, discussed the flaws of previous state-of-the-art data-based methods, experimented with improvements for a previous data-based method, presented an original terrain synthesis algorithm based on terrain-optimization, and conducted the largest terrain realism study to date.

7.1 Main Contributions

I have presented five chapters of contributions to the field of terrain synthesis.

Chapter 2 defined the ideal terrain synthesis algorithm, defined the term “realism” and how it differs from “believability”, reviewed a wide range of terrain synthesis papers, and presented background on various methods of texture synthesis. The ideal terrain synthesis algorithm was discussed to be a combination of realism, user control, and speed. The term “realism” was clarified to be the objective measure of the likelihood a terrain could exist and is separate from the subjective measure of “believability”, which is often mistaken for realism. An extensive background of previous terrain methods as categorized by methodology (approximation, model-based, or data-based) and reviewed in relation to the ideal terrain synthesis algorithm. Finally, a background in patch-based, pixel-based, and texture-optimization texture synthesis algorithms was presented as the precursor

to the discussion of the data-based methods in Chapter 3, and my work in developing new methods in Chapters 4 and 5.

Chapter 3 presented a discussion of previous data-based methods and an in-depth exploration of the flaws I identified with the algorithms. The data-based methods discussed were all based on texture-synthesis methods: Zhou et al. [160], Tasse et al. [137], and Gain et al. [60]. The method of each algorithm was described in detail and flaws with each algorithm were summarized. This was followed by a discussion of the flaws, some of which are shared between the algorithms: parameterization, patch size, feature identification, thin-plate splines, interpolation, limitations using the data, and user expertise. Each flaw was explored in detail with how it causes issues with the relevant algorithms and how it affects terrain synthesis in general.

Chapter 4 addressed the flaw of relying on user expertise (described previously in Chapter 3) by experimenting with using pit-removal algorithms in conjunction with an existing data-based method. Endorheic basins were discussed as an issue to synthesizing more realistic terrains using data-based methods. Three fundamental pit-removal algorithms from GIS were discussed in their relation to applications to terrain synthesis: depression-filling, depression-breaching, and a hybrid approach. Three experiments were conducted to apply pit-removal algorithms to the data-based method by Gain et al. [61]: height-offset modification, constraint addition, and flow-accumulation appearance space. All three pit-removal algorithms were tested as part of the height-offset modification application, where the pit-removal was performed in alternation with the correction phase of the Gain algorithm. Depression-breaching was used as part of the constraint addition application, where breach paths calculated using the depression-breaching were used as constraints for the Gain algorithm during each iteration of the synthesis. Flow-accumulation, another algorithm used in GIS, was used as part of the flow-accumulation appearance space addition, where the accumulated flow over the example-map was used as an additional component of the appearance space for the Gain algorithm. I concluded that the most successful application of pit-removal algorithms was the height-offset modification using the depression-breaching algorithm.

Chapter 5 introduced a new approach to terrain synthesis based on the texture-optimization approach of Kwatra et al. [85] and explored potential improvements that address the flaws of data-based texture synthesis discussed in Chapter 3. Kwatra et al.'s algorithm and additions for texture synthesis are discussed in detail. The core algorithm for the new terrain-optimization approach is described and discussed in relation to how Kwatra et al.'s approach was adapted to terrain synthesis. Experiments were made to improve the realism of the terrain-optimization algorithm by addressing some of the flaws from Chapter 3, including: patch-based initialization to address interpolation, expanding the PatchMatch search space to address limitations using the data, and pit-removal to address the user-expertise. Patch-based initialization altered the target map by placing patches extracted from the example map, before the synthesis began optimizing. Expanding the search space altered the PatchMatch algorithm by including non-discrete translations, rotations, flipping, and height-offsets, in order to increase the utilization of the data available from the example-map. Pit-removal used the algorithms discussed in Chapter 4 to remove endorheic basins formed during the synthesis, to reduce the reliance on user expertise to synthesize realistic terrain. I concluded that the core terrain-optimization algorithm can synthesize terrain at least as realistic as previous data-based methods, and that the pit-removal was the only experiment that improved the realism of the synthesis.

Chapter 6 explores how to evaluate the realism of synthesized terrain, presents a new experimental evaluation method, and discusses the results this experiment. Previous methods of evaluating terrain were discussed and flaws in their methodology were identified. The various aspects of evaluating the realism of terrain were discussed and the flaws in the previous experiments were addressed in each discussion. These aspects were the testing and analysis methodology, terrain visualization, and feature-bias: a bias that causes some terrains to be viewed as more realistic than others. I proposed a method of evaluating terrain through pair-wise comparisons, based on the previous discussion. With this new method I conducted the largest terrain realism survey to date, with a total of 245 participants with a wide range of expertise in physical geography, cartography, image analysis, and visual arts. The results of the experiment showed that every method was

indistinguishable from reality in certain circumstance and that these circumstances differ for each method tested, and that real-world terrain was always considered as or more realistic than terrain produced by every method. I found that those with a high level of expertise in physical geography are most qualified for identifying real terrain from synthesized terrain, which was consistent with previous theories.

7.2 Future Work

While I have progressed the field of terrain-synthesis there is further research that could be conducted.

Participant feedback from the experimental evaluation I conducted in Chapter 6 revealed that a significant portion of participants with expertise in physical geography evaluated the realism of the terrain based on the drainage patterns and global structure of the terrain rather than local features. However, the results of the experiment show that there was no significant difference in believability (in most cases) between the Gain algorithm, which does not enforce drainage, and my adaptation of the Gain algorithm incorporating pit-removal, which does. Future research could be conducted to explore the possibilities of using other techniques from physical geography to improve the result of data-based methods. This could be done by investigating combinations of data-based and model-based approaches.

In Chapter 6 I discussed many possible ways to conduct an experimental evaluation of terrain realism. My decisions on how to construct my experiment were informed by previous experimental methodology and investigations of alternative solutions. While the experiment had the most participants of any terrain realism experiment to date, it is not a viable solution for future researchers to evaluate the realism of new methods for terrain synthesis because it is impractical to consistently recruit more than 100 experts in geomorphology to evaluate every new terrain synthesis algorithm. Future work in this area could involve created a reliable methodology that can be consistently replicated and compared between researchers. A standardized test with the requirement of only a few experts in physical geography would allow researchers to not only evaluate

the realism of their own method but also allow them to estimate the differences between their method and other terrain synthesis methods that have been evaluated with the same test. A standardization could involve: requiring a certain number of participants with a certain level of expertise in physical geography, a common set of real-world terrain to compare the terrain synthesis method against, a standardized testing methodology (either 2AFC or 3 choices as in my experiment), and guidelines for how the terrain should be displayed to the participant.

Finally, while the focus of this thesis has been on the realism and the evaluation of realism in data-based terrain synthesis, realism is only one of the three major components of the ideal terrain synthesis algorithm outlined in Section 2.1.3, the other two being control and speed. The flaws outlined in Chapter 3 address the problem of realism in existing algorithms, when there are many aspects of control that could be addressed for the state-of-the-art data-based methods. Similarly, as identified in Chapter 2, model-based methods of terrain synthesis also generally lack control, which could be improved to bring these methods closer to being an ideal terrain synthesis algorithm. Speed is less of an issue, as there are already methods that can synthesize terrain in real-time. Once a method has been developed that has appropriate realism and control, the issue of speed can be addressed through optimization.

7.3 Concluding Remarks

This thesis presents an investigation of terrain synthesis methods with particular emphasis on data-based methods that are based on example based-texture optimization techniques, and the concept, synthesis, and evaluation of realism in terrain synthesis. With the work presented in this thesis I have furthered the knowledge of data-based methods and laid the ground-work for future research in the field of terrain synthesis and terrain realism.

Appendices

Appendix A

Feature-bias Survey

This appendix contains screenshots of the survey and terrain images used for the subjective evaluation in Section [6.2.3](#).

Participant Information

You are invited to take part in a study investigating *image quality*. This study involves one session, during which you will view images and answer questions about them.

Data collected during this study will be stored in an anonymous form, in which no personally identifying information about you will be kept. You may withdraw from this study at any time.

This Participant Information will help you decide if you'd like to take part. It sets out why I am doing the study, what your participation would involve, what the benefits and risks to you might be, and what would happen after the study ends. Feel free to ask the researcher any questions before you consent to participation.

If you agree to take part in this study, you will be asked to accept the consent form below.

Please make sure you have read and understood both this Participant Information and the Consent Form before consenting.

This research has VUW Human Ethics Committee (HEC) approval.

What are the conditions for my participation in this study?
In order to participate in this study, you must:

- Be over 18 years of age
- Able to view images on a screen
- Not have any serious visual abnormalities (amblyopia, spherical aberrations, etc.)

If you have a visual condition (e.g. colour blindness) and are not sure if it will be an issue, please mention this at the end of the survey. Note that requiring prescription focal correction (i.e. long or short sightedness) is not an issue.

What is the purpose of this study?
To determine how well our terrain synthesis algorithm works to produce realistic landscapes in comparison with other algorithms and real data.

What will participating in this study involve?
You will be invited to view a series of images and answer questions for each one.

What are the possible benefits and/or risks of this study?
There are no risks with this study. The results of this study will benefit the researchers by providing a way of assessing the researcher's algorithms.

What happens once this study ends?
The researcher will remove all identifying information about you from the results, and only store aggregated data: you may request from me these results, but they will not be tailored to you or mention you in any specific manner.

These aggregate results will be stored indefinitely.

The aggregates results of this study will used in my thesis, and in a proposed research paper to be published at a later date.

Figure A.1: Screenshot of the participant information sheet and consent form.
Continued in Figure A.2.

If you have any questions, who can you contact?
 Student researcher: Joshua James Scott
 Email [REDACTED]

Supervisor: Neil Dodgson
 Phone [REDACTED] or email [REDACTED]

Human Ethics Committee Information
 If you have any concerns about the ethical conduct of the research you may contact the Victoria University HEC Convener: Associate Professor Susan Corbett. Phone [REDACTED] or email [REDACTED]

Thank you very much for your time and help in making this study possible.

Consent Form

Please read the following notices.
 By selecting "I consent": I, the participant, understand and agree that:

- My participation in this research is voluntary and I am aware that I am able to withdraw at any point.
- I confirm that I have been provided, read and understand the Participant Information above.
- I have had the opportunity to ask any questions about this research and had them answered.
- I understand that all personal information will remain confidential and that all efforts will be made to ensure I cannot be identified.
- I agree that data gathered in this study may be stored anonymously and securely, and may be used for future research.
- I agree to take part in this study.
- If I have any further concerns and/or questions, I am aware I can use the following contacts:
 - Student researcher: Joshua James Scott
Email: [REDACTED]
 - Supervisor: Neil Dodgson
School of Engineering and Computer Science
Phone: [REDACTED] or email: [REDACTED]
 - Human Ethics Committee Convener: Associate Professor Susan Corbett
Phone [REDACTED] or email [REDACTED]

☐ I consent, begin the study
☐ I do not consent, I do not wish to participate

0% 100%
 Survey Completion

Powered by Qualtrics

Figure A.2: Continuation of the participant information sheet and consent form.

In the following sections, you will be presented with pairs of terrain images and asked to select the terrain that is more realistic. Some of the landscapes presented are generated to form artificial structures. Please assess these in the same way as the others, selecting the terrain that is more realistic.

A terrain is more realistic if it is more likely to be formed by natural processes on Earth.

Here is an example of a terrain:

The upper image is a shaded relief-map with blue contour lines spaced at every 100m, and the lower image is a 3d view of the shaded relief-map. The color ramp represents the elevation above sea-level.

All terrains are approximately 30 kilometers squared. None of the following terrains contains large bodies of water (eg: lakes, sea, ocean).

0% 100% Survey Completion

Powered by Qualtrics

Figure A.3: Screenshot of the terrain information.

Figure A.4: Screenshot of an example for a pair-wise comparison.

TE WHARE WĀNANGA O TE ŌPŌKO O TE IKA A MĀUI
VICTORIA
UNIVERSITY OF WELLINGTON

Age (in years)

Gender

☐ Male

☐ Female

☐ Other (please specify)

Do you have a visual condition (e.g. colour blindness) that may have affected your ability to compare the realism of the terrains in the previous sections?

☐ Yes (please specify)

☐ No

0% Survey Completion 100%

← →

Powered by Qualtrics

Figure A.5: Screenshot of the participant questionnaire.

Figure A.6: Terrain images used in the pair-wise comparisons. Continued in Figure [A.7](#).

Figure A.7: Continuation of the terrain images used in the pair-wise comparisons.

Appendix B

Subjective Evaluation Survey

This appendix contains screenshots of the survey and terrain images used for the subjective evaluation in Section [6.3](#).

Participant Information

You are invited to take part in a study investigating *image quality*. This study involves one session, during which you will view images and answer questions about them.

Data collected during this study will be stored in an anonymous form, in which no personally identifying information about you will be kept. You may withdraw from this study at any time.

This Participant Information will help you decide if you'd like to take part. It sets out why I am doing the study, what your participation would involve, what the benefits and risks to you might be, and what would happen after the study ends. Feel free to ask the researcher any questions before you consent to participation.

If you agree to take part in this study, you will be asked to accept the consent form below.

Please make sure you have read and understood both this Participant Information and the Consent Form before consenting.

This research has VUW Human Ethics Committee (HEC) approval.

What are the conditions for my participation in this study?
In order to participate in this study, you must:

- Be over 18 years of age
- Able to view images on a screen
- Not have any serious visual abnormalities (amblyopia, spherical aberrations, etc.)

If you have a visual condition (e.g. colour blindness) and are not sure if it will be an issue, please mention this at the end of the survey. Note that requiring prescription focal correction (i.e. long or short sightedness) is not an issue.

What is the purpose of this study?
To determine how well our terrain synthesis algorithm works to produce realistic landscapes in comparison with other algorithms and real data.

What will participating in this study involve?
You will be invited to view a series of images and answer questions for each one.

What are the possible benefits and/or risks of this study?
There are no risks with this study. The results of this study will benefit the researchers by providing a way of assessing the researcher's algorithms.

What happens once this study ends?
The researcher will remove all identifying information about you from the results, and only store aggregated data: you may request from me these results, but they will not be tailored to you or mention you in any specific manner.

These aggregate results will be stored indefinitely.

The aggregates results of this study will used in my thesis, and in a proposed research paper to be published at a later date.

Figure B.1: Screenshot of the participant information sheet and consent form.
Continued in Figure B.2.

If you have any questions, who can you contact?
 Student researcher: Joshua James Scott
 Email [REDACTED]

Supervisor: Neil Dodgson
 Phone [REDACTED] or email [REDACTED]

Human Ethics Committee Information
 If you have any concerns about the ethical conduct of the research you may contact the Victoria University HEC Convener: Associate Professor Susan Corbett. Phone [REDACTED] or email [REDACTED]

Thank you very much for your time and help in making this study possible.

Consent Form

Please read the following notices.
 By selecting "I consent": I, the participant, understand and agree that:

- My participation in this research is voluntary and I am aware that I am able to withdraw at any point.
- I confirm that I have been provided, read and understand the Participant Information above.
- I have had the opportunity to ask any questions about this research and had them answered.
- I understand that all personal information will remain confidential and that all efforts will be made to ensure I cannot be identified.
- I agree that data gathered in this study may be stored anonymously and securely, and may be used for future research.
- I agree to take part in this study.
- If I have any further concerns and/or questions, I am aware I can use the following contacts:
 - Student researcher: Joshua James Scott
Email: [REDACTED]
 - Supervisor: Neil Dodgson
School of Engineering and Computer Science
Phone: [REDACTED] or email: [REDACTED]
 - Human Ethics Committee Convener: Associate Professor Susan Corbett
Phone [REDACTED] or email [REDACTED]

☐ I consent, begin the study
☐ I do not consent, I do not wish to participate

0% 100%
 Survey Completion

Powered by Qualtrics

Figure B.2: Continuation of the participant information sheet and consent form.

In the following sections, you will be presented with pairs of terrain images and asked to select the terrain that is more realistic. Some of the landscapes presented are generated to form artificial structures. Please assess these in the same way as the others, selecting the terrain that is more realistic.

A terrain is more realistic if it is more likely to be formed by natural processes on Earth.

Here is an example of a terrain:

The upper image is a shaded relief-map with blue contour lines spaced at every 100m, and the lower image is a 3d view of the shaded relief-map. The color ramp represents the elevation above sea-level.

All terrains are approximately 30 x 30km. None of the following terrains contain large bodies of water (eg: lakes, sea, ocean).

0% 100% Survey Completion

← →

Powered by Qualtrics

Figure B.3: Screenshot of the terrain information.

Left is more realistic ☐

Both are equally realistic ☐

Right is more realistic ☐

● ● ●

Left is more realistic ☐

Both are equally realistic ☐

Right is more realistic ☐

0% 100%

Survey Completion

Figure B.4: Modified screenshot of the pair-wise comparisons. Each page contained 6–8 pairs.

TE WHARE WĒHANGA O TE ŌPŌKO O TE IKA-A-MĀUI
VICTORIA
UNIVERSITY OF WELLINGTON

Age (in years)

Gender

☐ Male

☐ Female

☐ Other (please specify)

Do you have a visual condition (e.g. colour blindness) that may have affected your ability to compare the realism of the terrains in the previous sections?

☐ Yes (please specify)

☐ No

How much experience do you have studying or researching in the field of Physical Geography (including Geomorphology, Geology, etc.)?

☐ I have little or no experience with Physical Geography (at most a high-school education in Geography)

☐ I have some experience with Physical Geography (currently enrolled in a bachelors degree or equivalent)

☐ I have experience with Physical Geography (completed a bachelors degree or equivalent)

☐ I have a lot experience with Physical Geography (postgraduate degree or currently working in the field)

How much experience do you have of map-reading for recreation (e.g. hiking)?

☐ I cannot read a map, or do not use a maps in recreational activities

☐ I have some experience of reading maps

☐ I can read a map well

☐ I am expert at reading maps

Figure B.5: Screenshot of the participant questionnaire. Continued in Figure B.6.

How much experience do you have of identifying visual artifacts (or inconsistencies) in images?

☐ I have never had to identify visual artifacts in images

☐ I have occasionally had to identify visual artifacts in images (e.g. as part of a survey)

☐ I regularly have to identify visual artifacts in images (e.g. as part of a hobby or as work)

How much experience do you have in the visual arts (e.g. drawing, painting, photography, film)?

☐ I have no experience in the visual arts

☐ I have occasionally participated in the visual arts (e.g. in art classes or as recreation)

☐ I regularly participate in the visual arts (e.g. as part of a hobby or as work)

While comparing the realism of the terrains in the previous sections, what was your strategy to determine the most realistic terrain? (optional)

While comparing the realism of the terrains in the previous sections, what images did you find most useful for assessing the realism of the terrain? (optional)

☐ Flat relief map with contours (top image)

☐ 3D rendered view (bottom image)

☐ Both were equally useful

0% Survey Completion 100%

← →

Powered by Qualtrics

Figure B.6: Continuation of the of the participant questionnaire.

Figure B.7: Set 1 terrain images used in the pair-wise comparisons (except the source).

Figure B.8: Set 2 terrain images used in the pair-wise comparisons (except the source).

Figure B.9: Set 3 terrain images used in the pair-wise comparisons (except the source).

Figure B.10: Set 4 terrain images used in the pair-wise comparisons (except the source).

Figure B.11: Set 5 terrain images used in the pair-wise comparisons (except the source).

Figure B.12: Set 6 terrain images used in the pair-wise comparisons (except the target and the source).

Figure B.13: Set 7 terrain images used in the pair-wise comparisons (except the source).

Appendix C

Expertise Analysis

This appendix contains supplementary tables for the analysis of the accuracy for each category of expertise.

Summary					
Exp	Count	Sum	Mean	Variance	
G_1	55	35.50	0.6455	0.0219	
G_2	45	28.12	0.6248	0.0173	
G_3	21	13.10	0.6238	0.0242	
G_4	124	93.89	0.7572	0.0257	

ANOVA					
Source of Variation	SS	df	MS	F	P-value
Between Groups	0.9419	3	0.3140	13.55	3.34×10^{-8}
Within Groups	5.5824	241	0.0232		
Total	6.5244	244			

Table C.1: ANOVA for the accuracy of the physical geography expertise G_i . $F(3, 241) = 13.55, p < 0.001$, which shows that there are statistically significant differences at the $p < .05$ level.

Summary					
Exp	Count	Sum	Mean	Variance	
C_1	10	6.35	0.6350	0.0161	
C_2	65	42.47	0.6533	0.0176	
C_3	78	54.15	0.6942	0.0267	
C_4	92	67.64	0.7352	0.0319	

ANOVA					
Source of Variation	SS	df	MS	F	P-value
Between Groups	0.2972	3	0.0991	3.83	0.0104
Within Groups	6.2272	241	0.0258		
Total	6.5244	244			

Table C.2: ANOVA for the accuracy of the cartography expertise C_i . $F(3, 241) = 3.83, p = 0.0104$, which shows that there are statistically significant differences at the $p < .05$ level.

Summary					
Exp	Count	Sum	Mean	Variance	
I_1	69	45.07	0.6531	0.0210	
I_2	99	69.70	0.7040	0.0271	
I_3	77	55.84	0.7252	0.0295	

ANOVA						
Source of Variation	SS	df	MS	F	P-value	
Between Groups	0.1987	2	0.0993	3.80	0.0237	
Within Groups	6.3257	242	0.0261			
Total	6.5244	244				

Table C.3: ANOVA for the accuracy of the image analysis expertise I_i . $F(2, 242) = 3.80, p = 0.0237$, which shows that there are statistically significant differences at the $p < .05$ level.

Summary					
Exp	Count	Sum	Mean	Variance	
A_1	60	41.05	0.6842	0.0244	
A_2	124	86.79	0.6999	0.0279	
A_3	61	42.77	0.7011	0.0274	

ANOVA						
Source of Variation	SS	df	MS	F	P-value	
Between Groups	0.0119	2	0.0059	0.22	0.8025	
Within Groups	6.5125	242	0.0269			
Total	6.5244	244				

Table C.4: ANOVA for the accuracy of the visual arts expertise A_i . $F(2, 242) = 0.22, p = 0.8025$, which shows that there are no statistically significant differences at the $p < .05$ level.

Tukey's HSD Test				
	diff	lower	upper	p adj.
$G_2 - G_1$	-0.0206	-0.0998	0.0585	0.9066
$G_3 - G_1$	-0.0216	-0.1226	0.0793	0.9458
$G_4 - G_1$	0.1117	0.0479	0.1755	< 0.0001
$G_3 - G_2$	-0.0010	-0.1051	0.1030	1.0000
$G_4 - G_2$	0.1323	0.0638	0.2009	< 0.0001
$G_4 - G_3$	0.1333	0.0404	0.2263	0.0014
$C_2 - C_1$	0.0183	-0.1229	0.1596	0.9869
$C_3 - C_1$	0.0592	-0.0804	0.1989	0.6917
$C_4 - C_1$	0.1002	-0.0383	0.2387	0.2428
$C_3 - C_2$	0.0409	-0.0289	0.1107	0.4301
$C_4 - C_2$	0.0819	0.0145	0.1493	0.0101
$C_4 - C_3$	0.0410	-0.0230	0.1050	0.3494
$I_2 - I_1$	0.0509	-0.0089	0.1107	0.1126
$I_3 - I_1$	0.0720	0.0088	0.1352	0.0209
$I_3 - I_2$	0.0211	-0.0368	0.0791	0.6657

Table C.5: Results of post-hoc analysis with Tukey's HSD test. For expertise in physical geography, there are statistically significant differences between G_4 and each of the other three categories but not between any pair of G_1 , G_2 , and G_3 . For expertise in cartography, there are statistically significant differences between C_4 and C_2 , but not between any other pair of categories. For expertise in image analysis, I_1 and I_3 are statistically significantly different.

Appendix D

Subject Evaluation Strategies

This appendix contains responses to the question “While comparing the realism of the terrains in the previous sections, what was your strategy to determine the most realistic terrain? (optional)”, sorted by participant expertise. Responses that contain personally identifying information have been redacted or excluded.

Expertise				Response
G_i	C_i	I_i	A_i	
1	1	2	3	Clarity
1	1	2	3	super smooth hills looked more like golf courses than real terrain.
1	1	3	1	Try to not think too much, quick glance reaction, gut feeling.
1	1	3	2	Relative consistency (no sharp change between smooth and rough terrain). And when there are sharp changes, do they occur consistently.
1	2	1	1	“Clarity Less flat areas with no contours”

Expertise				Response
G_i	C_i	I_i	A_i	
1	2	1	1	<p>“Use of colour- bright greens/blues were not realistic compare to real vegetation/ water</p> <p>Using both images to make a final decision- sometimes the flat relief map looked more realistic but then used smaller 3D image to make final decision</p> <p>I had little engagement in the process- it was meant to represent physical geography so it didn’t matter to me how realistic it was,— it was close enough”</p>
1	2	1	2	Put my first glance impression as my answer.
1	2	1	2	Sharpness or blur of the lines. How many contours there were.
1	2	1	2	To determine what image is more realistic I based it off prior knowledge as well as when picking one, I would also base it off my previous answers to have a more consistent answer throughout the process
1	2	1	3	I just went by what appeared to have the most detail in terms of shadowing and colour variations
1	2	2	1	by picking which one could actually be a terrain
1	2	2	1	<p>“Realistic lines of mountains was a big one. If I thought it was islands in a body of water then more gentle contours. Sometimes I just picked one that seemed more believable as a gut instinct.</p> <p>I feel that mountains would never form a Half Life Lambda, so picked the fuzzier ones.”</p>
1	2	2	1	Sharpness and space between high points
1	2	2	2	Geological extremities or too smooth slopes
1	2	2	2	gestalt feeling
1	2	2	2	The sharpness of the lines and the space between the topo lines
1	2	2	3	familiar types of features, erosion, water channels etc

Expertise				Response
G_i	C_i	I_i	A_i	
1	2	2	3	I considered how the water would run off the terrain. Fully enclosed valleys were considered unnatural as any water within the valley would have no exit. Ridges in terrain I've observed in real life tends to appear like tree graphs. I considered terrain with a more structured appearance to be more realistic. Some parts of the terrain seemed to be as if it had been copy and pasted several times, which reduced realism.
1	2	2	3	<p>"Looking out at the scale of the details (some impressions in the terrain appeared too deep, scale of hills were too small)</p> <p>Looking for less uniformity to decide realism (many repeated features look less realistic)</p> <p>Looking for consistency in level of details (terrain with very smoothed contours next to defined ones look less realistic)"</p>
1	2	2	3	not too much and not too little in variation, also looking for non-regular shapes and patterns to imply more realism
1	2	2	3	Nothing too overly clean or uniform, sharp edges were less realistic than a sharper edge with like a buffer of land
1	2	3	3	<p>"Checking Noise / Ridges too consistent</p> <p>Flatness too consistent</p> <p>Are features believable? Rivers / mountains / patterns in sensible locations"</p>
1	3	1	1	factures looks too detailed in some cases
1	3	1	1	Figuring out what might appear in nature.
1	3	1	1	First impression of natural appearance.
1	3	1	2	I mostly went with my first impression
1	3	1	2	If something was too uniformed then I struck it off as being unrealistic. Also if there was too much blank spaces this too made it look unrealistic

Expertise				Response
G_i	C_i	I_i	A_i	
1	3	1	2	What looked more life like, not too perfect, colours and sharp lines made it look less real life.
1	3	2	1	which looked more like NZ maps I'm used to? did straight-line features look likely to be fault traces? did a ring of mountains enclose an area with nowhere for water to go?
1	3	2	2	Natural structure (avoid artificial looking)
1	3	2	2	Real terrain should have a fractal structure, i.e. you should expect to count a comparable amount of "same size" objects for any size region selected on map.
1	3	2	2	Some looked too smooth while others looked too sharp - Many would have been places that would be too steep to make sense with the land surrounding them or too smooth to seem like they had formed naturally.
1	3	3	2	avoiding obvious repeating patterns, looking for good mix of high frequency and low frequency patterns
1	3	3	3	I generally tried to track natural features like long flowing deltas to see if they were consistent, also was looking out for large tectonic features that were consistent between surroundings.
1	3	3	3	Scale, noise, realistic contours, topographical consistency
1	4	3	2	Imagine flying over the terrain. Look for unnatural shapes.
2	1	2	1	i looked at colour, slope, ridge lines
2	2	1	1	contouring, colour sharpness, contrast
2	2	1	1	Detail richness and colour pattern to represent relief
2	2	1	2	Analysing the steepness of a drop and its height above sea level. Analysing the climate apparent in the image and how water might or might not appear in lakes or rivers etc. Also looking at mountain and hill ranges to see their relation to real ones that I have seen. A lot of it was estimation, as I was not aware fully of the climate.

Expertise				Response
G_i	C_i	I_i	A_i	
2	2	1	2	Consider possible reasons for formation of structures, better reasons = more realistic
2	2	1	2	It was tricky as I felt out of the 2 terrains to choose from, one could have been (for example) a sand dune, or one could have been rock in which case both terrains looked realistic. In the end I just went for the terrain that was the most familiar.
2	2	1	2	Look at the shapes, lines and colours
2	2	1	2	looking at the mountain ranges i looked to see if evidence of fluvial processes was present and more prominent relief
2	2	1	2	The shadows, jaggedness of the ridges, whether it was blurred.
2	2	1	3	how steep the river gorges were and whether stream flow carried through
2	2	1	3	I made my choice on what option was more realistic by looking at how refined each one was.
2	2	1	3	Two-fold: one, look intuitively at the map to decide (not thinking about it much), and two, to consider whether certain erosion patterns were likely (although I wouldn't really know).
2	2	1	3	Which one looks more believable; which one looks like a map I've seen
2	2	2	1	Fewer straight lines
2	2	2	2	Looking for unlikely features
2	2	2	2	Terrains that looked more eroded generally seemed to be more realistic
2	2	2	2	What has the most detail
2	2	2	2	Whether or not formation of the land like that is likely and how weathering processes may alter the landscape.

Expertise				Response
G_i	C_i	I_i	A_i	
2	2	2	3	Shading, more prominent land forms. There's a slight consistency, though not too uniform, i.e. cone volcano not completely round and uniform- there were some random structures interacting with it, which to me looks more realistic.
2	2	2	3	To determine the most realistic terrain I looked at which one had more detail represented and how if the steepness of the slopes appeared too abrupt relative to the horizontal plane. If it was too abrupt I didn't find it realistic.
2	2	3	3	I work with World Machine/Terrain tools frequently for VFX environment work
2	3	1	1	I tended to select the image that had sharper shapes and was more defined
2	3	1	2	felt real, or was more rough/bumpy/cut deeper
2	3	1	2	What seemed to me to be the most real type of land form and how it would look on a topographic map.
2	3	2	1	if the topography looked realistic for the setting
2	3	2	2	Find the one that has a similar geographical equivalent or looks like it is a natural product of geographical processes.
2	3	2	2	Terrains with topo lines closer together always looked more realistic because the 3D model used more exact data and had more detail. I generally looked at the 3D models because those look more like what you see in nature and more easy to compare to a "realistic terrain."
2	3	2	2	The images that seemed to give more detail in the terrain's structure. That tended to be the ones with more topo lines and sharper rather than rounded 3D images

Expertise				Response
G_i	C_i	I_i	A_i	
2	3	3	3	<p>“look at the way ridge lines are formed and how water run off might flow if there are clear places that should have been eroded it draws the eye and prompts disbelief</p> <p>in addition look at how slopes and inclines form eg. rounded edge, or arbitrary straight lines in addition to unnaturally consistent elevations”</p>
2	4	2	1	Seeing what contour lines look the most realistic
2	4	2	2	features formed from fluvial erosion such as ridges and v-shaped valleys were the best indication to me that a terrain was more realistic. most environments on earth have water drainage features which always go down slope and are continuous, not usually stopping abruptly or forming depressions. the sharpness of hills also indicated more realistic terrains to me.
2	4	2	3	Finding a balance between too much relief and too flat. Some of the too flat scenes looked unlike terrains I have seen which seem to ease from terrain to a flat place rather than a quick change from mountains to a flat surface.
3	2	2	1	Look for unrealistic features; i.e. a stream cutting through a topographic high in an unusual orientation. Also, look for unreasonable “smoothness” of the landscape.
3	2	2	3	Nature is not perfect and so images that seemed very straight, sharp or otherwise ‘perfect’ in design were distinguished as not natural.
3	3	1	1	The countour lines spacing and how “softly” the trancision was among the lines. Also the spacing. Additionally, the prersence of “abnormal” geomorphological features, such as sections of isolated drainages
3	3	1	3	Relate it back to maps i’ve seen in the past, and decide which of the two is most similar

Expertise				Response
G_i	C_i	I_i	A_i	
3	3	2	1	"hydrographic network repetition of some features"
3	3	2	1	Imagining erosion and how shaded relief would best represent those erosional effects.
3	3	2	3	thinking if these landscapes could have been formed by physical processes and looking out for visual artifacts
3	3	3	2	looking at features that repeat, erosion patterns, over smoothing, "fake" looking
3	3	3	3	<p>"If there were any flat areas it seemed unrealistic. More detail in the landscape the more realistic.</p> <p>As for the uniform landscapes (like the circle symbol) i chose the ones with more ruggedness because the occurrence of every peak coming to a single elevation is unlikely.</p> <p>As for some of the flatter images, the ones with rounded peaks seemed odd because of the ways mountains erode in general unless it is in an arid climate.</p> <p>Valley bottoms that were rounded only seemed like it could have been in an out wash plain in a desert region so I always said they were equally realistic.</p> <p>To be honest, the contour lines didn't really help, it just made me think you stretched the TIN out differently and therefore I counted more as equally realistic instead of judging them purely on appearance because the more contour lines/ more variety in elevation the more detailed, if they weren't as detailed it doesn't mean they aren't realistic."</p>
3	3	3	3	Often both were equally realistic but portrayed different conditions. I marked those as "same" in terms of realistic but as different looking places. Not sure if that was what was intended though.

Expertise				Response
G_i	C_i	I_i	A_i	
3	4	1	3	“check artefacts, direction of valleys, accumulation of sediments, check too simple forms, check direction of potential faults it lacks the answer ““none of them seem realistic for some couple of images”””
3	4	2	2	How water would move over the landscape and comparative heights for the type of landscape
3	4	2	2	“The presence of believable drainage basins, how extreme gradients are. In the presence of Black Mesa, how believable the mountain chain would be (there are crazy formations on Earth).”
3	4	3	2	Looking for repetitive features, especially ones located right next to each other. Looking for large areas of completely flat elevation, looking for areas that should have had clean cut features that were rounded off.
4	1	2	1	Sharp edged and color contrast
4	2	1	1	looking for realistic drainage directions (ie, forming of basins by river erosion
4	2	1	1	Try and see if I could identify realistic drainage directions.
4	2	3	2	Comparing it to the many terrain models I have seen in my work, looking at what looks “too smooth”, or has odd artifacts such as a flat surface with small rectangles that look too regular to be natural.
4	3	1	1	“1) drainage patterns- river incision downhill, and no/few closed basin 2) some had too much diffusion vs incision, looked unnatural”
4	3	1	1	“looking for discontinuous drainage networks - ie rivers that go up. Overly “smooth” or “rough” surfaces - thinking that weathering should be generally consistent within the area. I got a little rushed towards the end and was spending a bit less time per image pair than at the beginning.”

Expertise				Response
G_i	C_i	I_i	A_i	
4	3	1	1	Smoothness vs. roughness, irregularities vs. regularities (e.g., too regularly spaced valleys), shape of the valleys and planes.
4	3	1	2	A lot was based on intuition, but some features were clearly disconcerting: strange lumpiness across the whole landscape overall, not enough features rendered giving a sense of fuzziness (one of the strike-slip fault example), or patterns being repeated too regularly (e.g., valley spacing).
4	3	1	2	Look at the continuity of river network, think about features expected in different geological contexts
4	3	1	2	Looking for more asymmetry, a wide range of reliefs, and believable evidence of river incision.
4	3	1	2	Visual gut/ clarity and colour
4	3	1	3	how natural the level of weathering (smoothing) or active erosion (sharper features) looked
4	3	2	1	Did the contour lines in the relief maps look plausible? (ie. based on the surrounding terrain, how likely was it that the base of the map was *actually* flat?).
4	3	2	1	I tried to recognize the terrain the “felt” more realistic
4	3	2	2	Checking for realistic drainage network organization and topographic gradients.
4	3	2	2	Comparison to landscapes I have seen or topography maps I’ve looked at
4	3	2	2	“Incision. Then boredom as too many examples to closely care about.”

Expertise				Response
G_i	C_i	I_i	A_i	
4	3	2	2	<p>“Look for directionality:</p> <ul style="list-style-type: none"> - river catchments and networks, uninterrupted flowpaths, non aberrant flowpaths (bi-directional river in some of the Lambda landscapes), mountain chains - Look for heterogeneity in lowlands: 0-slope with no hummocks is very rare unless in massive floodplains - Look for transitions: abrupt transitions exist in nature, but must be consistent with a change in landscape morphology (e.g. rugged vs. smooth) that suggests a change in rock type, or faults.”
4	3	2	2	Not too smoothed. No artifacts. No regularly spaced patterns.
4	3	2	2	Quick visual inspection of fluvial connectivity.
4	3	2	2	<p>“Realistic drainages vs closed basins</p> <p>Increased irregularity at high elevations</p> <p>Sensible structural control of topography”</p>
4	3	2	2	Smoothness of the terrain, lack of straight lines, lack of very regularly-repeated features.
4	3	2	2	<p>“Weird artifacts indicated not realistic, as did too much symmetry</p> <p>looked for realistic/consistent drainage patterns”</p>
4	3	2	3	I was simply trying to assess what most closely resembled similar environments and erosive patterns that I have been to in real life, out of the two images I was looking at.
4	3	2	3	Intuition

Expertise				Response
G_i	C_i	I_i	A_i	
4	3	2	3	<p>“unrealistic maps:</p> <p>too many closed catchments (while they exist in, for instance, limestone geology or cirques, they still are probably less common/to be expected than ‘normal’ drainage patterns),</p> <p>repetition of almost identical shapes,</p> <p>shapes that strongly drew from artificial human abstractions (e.g., lambda-like images) - while they can exist naturally I would think they are very rare (e.g., the face on Mars),</p> <p>some pairs were equally unrealistic</p> <p>realistic maps:</p> <p>lack of aforementioned points</p> <p>have i seen similar landscapes before - my experience with reading real maps & my knowledge of processes that can shape the topography”</p>
4	3	3	1	To see if there were obvious erosional processes - e.g. river valleys
4	3	3	2	<p>“1) Make an attempt to classify this into an analog environment e.g. Glacially modified, River dominated, Volcanic & Impact structure)</p> <p>2) Identify possibly inconsistency (artefacts) with analog environment based on my experience.”</p>
4	3	3	2	closures in contours. Artifacts
4	3	3	2	I looked for closed depressions and the consistency of the river networks.

Expertise				Response
G_i	C_i	I_i	A_i	
4	3	3	2	<p>"I was looking for suspiciously linear features or repeated features that would be less likely in a natural landscape. Some images also appeared highly smoothed. Those tended to look less real to me as well.</p> <p>Drainage networks should continue through the entire image as well. Some of the images appeared to have drainage networks that smoothed to nothing for no apparent reason."</p>
4	3	3	3	<p>If one had something I knew to be uncommon (or impossible) geomorphically then I chose the other one. If they both seemed to have equal amounts of "uncommon geomorphic" features I chose "equal." The one problem is that almost everything presented is 'possible.' That's why I tried to choose based on commonality.</p>
4	4	1	1	<p>"1) I tried to imagine the hydroclimate and the bedrock geology</p> <p>2) I look at drainage networks and preferred those that did not have basins, but rather drained.</p> <p>3) I looked for ""streams"" that had some sinuosity."</p>
4	4	1	2	<p>Assessing realistic scale of erosion domains (e.g., diffusive vs. advective), connectedness of flow paths.</p>
4	4	1	2	<p>"blurriness = unrealism for me</p> <p>I couldn't decide if the ultra flat terrains were realistic or not (perhaps they were supposed to be water bodies?)"</p>
4	4	1	2	<p>I looked at terrain roughness, the alignment of drainage networks, and connectivity of ridges.</p>
4	4	1	2	<p>look at both map view and oblique view images to assess them against my experience of natural terrain I have worked with/in. Particularly using the topographic contours.</p>
4	4	2	1	<p>Whether drainage networks appeared connected. While closed basins exist, they are rare. I also considered if the topography made sense from geomorphic process, glaciers, hillslope diffusion, etc.</p>

Expertise				Response
G_i	C_i	I_i	A_i	
4	4	2	2	<p>“1) do the river drainage patterns make sense (first, logically flowing downhill, then looking for consistency)</p> <p>2) is there a logical tectonic history that could create this pattern?”</p>
4	4	2	2	1st impression, erosion / drainage patterns, gridding artefacts
4	4	2	2	<p>“An obvious sign of non-realism was abundant internal drainage (closed contours) in synthetics.</p> <p>Synthetics often had small, anomalous peaks and valleys scattered throughout the landscape that were unusual.</p> <p>The synthetics also seemed to exaggerate drainage density at high elevations. Variability in drainage density is usually harder to spot.”</p>
4	4	2	2	Do the images make sense from a structural geology perspective (e.g., faults, folds) and possible bedrock types (i.e., mud rocks =more rounded topography).
4	4	2	2	I asked myself if it made ‘sense’, and I often tried to think of a real example - e.g., front range into a plain, or drowned coastlines.
4	4	2	2	I checked for the consistency of drainages systems, and where equally consistent I picked the landscapes that showed both long (1–10 km) and short (<1 km) wavelength morphologies
4	4	2	2	I focused on drainage development. Images that had no drainage network and a lot of closed depressions were unrealistic.
4	4	2	2	I looked at drainage systems, and lineaments.
4	4	2	2	I looked for realistic drainage patterns, especially in the flatter terrain away from the primary structures, and also terrain that wasn’t overly smoothed or roughened. The topography that suggested underlying geologic structures appeared more realistic to me versus randomly roughened terrain.
4	4	2	2	instinct - which one immediately looked more realistic.

Expertise				Response
G_i	C_i	I_i	A_i	
4	4	2	2	Landscape features that could have been formed by normal geologic processes (river erosion, glacial erosion, volcanism, tectonics). Connected, downward sloping drainage basins. Nothing too flat or too uniform. Consistent “age” of the landscape (all heavily eroded? all recently uplifted?).
4	4	2	2	Look for familiar patterns I know from my own work such as stream incision and related formation of river terraces and at the same time look for “strange” features that seem to not have been formed by natural processes like odd irregular spikes or smooth oddly shaped ridges. Whenever I found something looking strange I tried to think of natural processes forming it such as stratigraphic layers of different hardness or tectonic processes. I’m sure I was wrong several times though :)
4	4	2	2	looking for erosional features and shapes, comparing elevation change to patterns in erosional networks
4	4	2	2	“realistic smoothness and roughness (and guessing a reasonable scale and resolution) connectivity of flow isotropy of flow”
4	4	2	2	realistic types of topography and roughness
4	4	2	2	The main things I looked for that made me say things were unrealistic were strange geomorphology (e.g., rivers flowing weird directions), too much randomness (e.g., surface just looking too rough/pitted), random artifacts

Expertise				Response
G_i	C_i	I_i	A_i	
4	4	2	2	<p>“These are hard and fascinating, I hope at some point that you email</p> <p><i>redacted</i></p> <p>again upon completion of the study and tell us what the differences were and the ““answers”” as far as realism goes. I’m much less sure of my answers than I thought I’d be.</p> <p>How watersheds were integrated (first to 2nd to 3rd order streams in a logical pattern) was probably the single most important factor, as well as some sense of the degree of regularity of valleys and ridges (ie not too regular, but not too irregular). At first I thought about the images and tried to compare contour lines, but after a few I just did it based on a quick judgement call and moved on.”</p>
4	4	2	2	Understanding geologic structure and the likelihood of this influencing drainage effects from non-fluvial systems like karst, volcanism, and glacial geomorphology. I also used my memory of having visited landscapes and studied maps.
4	4	2	2	Where applicable, I looked for drainage patterns that seemed like a reasonable explanation of the overall terrain.

Expertise				Response
G_i	C_i	I_i	A_i	
4	4	2	3	<p>“by the end, I was mostly looking at areas of internal drainage and how ‘fuzzy’ features appeared to be.</p> <p>In typical mountain ranges, it’s uncommon to have lots of large, internally-drained areas, so those were red flags. Similarly, I was looking to see if rivers went anywhere or if they just terminated in the bottom of a valley with high curvature, bc this would also be suspicious. Many images had repeated artifacts (patterns) that unlikely to be produced in nature.</p> <p>Many of the images also looked sort of ‘fuzzy’ for lack of a better word. Something just seemed off about the curvature of the positive topographic relief.</p> <p>Also, lots of linear features jutting up in all different directions in otherwise flat ground seemed unrealistic.”</p>
4	4	2	3	<p>“I was looking for streams and where they might flow. Streams flowing into a closed basin on a plateau were used as indicator of less realistic landscapes. Similarly, terrains with closed basins in the middle of a valley were interpreted as less realistic.</p> <p>The presence of strait elongated and narrow islands was attributed to artefacts because of the creation of a synthetic landscape.”</p>
4	4	2	3	<p>“River patterns were often important.</p> <p>Some terrains were too ‘crinkly’.”</p>
4	4	3	1	<p>“Softness” of terrain (e.g., from diffusion) and ‘noise’ (short wavelength roughness).</p>
4	4	3	1	<p>artifact presence</p>
4	4	3	1	<p>experience with DEMs</p>

Expertise				Response
G_i	C_i	I_i	A_i	
4	4	3	1	I evaluated whether physical processes could reasonably produce the terrain, evaluated consistency of roughness across the terrain, looked for unusual geometries not familiar to me, looked for appropriately sloping drainage paths. I wonder if I was subconsciously selecting higher roughness/detail images as more realistic, even if that was not supported by other lines of evidence.
4	4	3	1	“Roughness of landscapes Connectivity of drainage”
4	4	3	1	Watching for regional to local variations that may indicate things were not as they were supposed to be.
4	4	3	2	“Connected drainage networks. Repeating topographic artefacts. Realistic features not easily reproduced by models (i.e., terraces, cirques).”
4	4	3	2	drainage pattern
4	4	3	2	First, it was overall “feeling” - hard to explain but it was “does it look right?”. I then looked at things like sections of internal drainage, and whether or not that was likely to occur in real life. Some of the surfaces looks like they were too rough or too smooth, like someone had artificially changed the surface from a true initial condition. Finally, I looked at things like channel development - did their depth and spacing appear real to me.
4	4	3	2	I looked for closed basins, particularly at ridge tops and in the middle of valleys (that otherwise appeared fluvial), looked for clear steps across contour lines (this was particularly acute in some of the images of two peaks that look like volcanoes), things that were “too smooth”, which I can’t even really describe in any sort of quantitative way, it just didn’t sit right, and an abundance of straight lines.

Expertise				Response
G_i	C_i	I_i	A_i	
4	4	3	2	I looked for landscapes that seemed to behave with respect to basic landscape evolution and transport laws (e.g., stream power). But I also looked for ones that had the weirdness of real landscapes that reflects things like transient forcings, or variable substrates that might create roughness or cliffs in one place and broad alluvial planes (with some incision) in others.
4	4	3	2	I looked for realistic erosion patterns, including continuity of drainages and likely variations in erosion processes with elevation.
4	4	3	2	<p>"I looked for two things primarily where I asked myself:</p> <p>1) Does the image as a whole look believable considering all the varied terrain on Earth (karst, arid/humid climates, weathering vs transport limited areas, etc.).</p> <p>2) Are there technical issues that just seems wrong (e.g., missing alluvial fans at the base of a slope OR dike like objects that seem to be random, rather than aligned with a stress direction OR internally drained areas that didn't look otherwise like karst in the highlands)"</p>
4	4	3	2	<p>"I primarily looked for two things:</p> <p>- logical landscape shapes: isolated or truncated drainages, abrupt illogical changes in drainage pattern, etc.</p> <p>- whether the topography seemed to be random or to follow rules: in other words, are there patterns to the irregularity, or is it seemingly random?"</p>
4	4	3	2	I thought about whether or not what I saw made sense in terms of processes that could shape landscapes. I had difficulty though in deciding whether landscapes that look more fluvially shaped (i.e., well connected) appeared "more realistic" than those that look to be shaped by groundwater and karst development (less connected); my experience is more in the former, which might have affected my choices.

Expertise				Response
G_i	C_i	I_i	A_i	
4	4	3	2	Identify landscapes which are eroded by stream runoff (water flows downhill) as being more likely to be real. The terrains with numerous depressions could easily have been karst, where dissolution and subsurface water passages form, but I was biased toward surface runoff eroded landscapes. The roughness/smoothness aspect could be related to the relative importance of diffusive and advective processes. If the surface was too rounded, I tended to think it was unrealistic.
4	4	3	2	<p>“Identifying drainage patterns was useful. The degree of dissection was helpful, since heavily dissected landscapes with broad hilltops seems less realistic to me than sharp hilltops in regions that are heavily dissected (Peclet number). I also avoided landscapes that had obvious internally-drained basins.</p> <p>I similarly avoided the landscapes that appeared to have significant visual artifacts.</p> <p>Wasn’t too sure how to respond to the landscapes with the lambda imposed on topography...”</p>
4	4	3	2	If I saw anything obvious, like straight lines or repeating patterns, that one was automatically less realistic in my head. I would also look for strange drainage patterns. Some were guesses because they looked equally unrealistic.
4	4	3	2	Logical geomorphic patterns
4	4	3	2	Looking for drainage patterns, linear features/artifacts that could not be easily explained by geology, inconsistent stream incision (e.g., getting shallow after being deeply incised upstream)
4	4	3	2	Natural topography is complementary to simple geomorphological principles. For example, if a mountaintop is jagged (high erosion) the valley will often be accompanied by gradually sloping walls indicating high deposition from the eroded materials above.

Expertise				Response
G_i	C_i	I_i	A_i	
4	4	3	2	<p>“Not too rounded if well channelised.</p> <p>Few closed basins in steep terrain.</p> <p>Isolated linear features are weird, unless clearly very deliberate.</p> <p>All else equal, more jagged terrain should be steeper.”</p>
4	4	3	2	Realistic drainage networks.
4	4	3	2	Seeking natural land forms, particularly realistic drainage networks.
4	4	3	2	Unrealistic slopes, unusually parallel contour lines, awkward drainage patterns, etc.
4	4	3	2	visually preferred sharp contrasts, was weary of excess smoothing
4	4	3	3	<p>“1) Does it look like something I have seen before?</p> <p>2) Are there obvious features that are physically impossible or unlikely (e.g. multiple valleys without outlets)?</p> <p>3) Does the pattern simply appear “disturbing” to me?</p> <p>4) Is the topography excessively smooth in areas of high relief?</p> <p>5) Do both map view and 3D view give me the same impression about how realistic the terrain is?</p> <p>6) If it takes me a long time to decide, then the images are equally realistic to me (which can mean either they are both realistic, or they are both unrealistic).”</p>
4	4	3	3	<p>“1) Landscape connectivity, e.g . how physically connected different parts of the terrain are</p> <p>2) Smoothness and/or Roughness”</p>
4	4	3	3	An important criterion was through going stream networks. While I understand that karst terrain is common, some of the images did not have karst that made sense to me. I also looked for natural looking bedrock control that could be explained by fractures or stratification.

Expertise				Response
G_i	C_i	I_i	A_i	
4	4	3	3	<p>“Any terrain can be realistic depending on climate and geology. I found the absence of a scale bar confusing. Therefore, for a lot of answers I selected they are equally realistic. Scale is very important. The only terrains I thought were unrealistic were those having flat surfaces, although some lidar maps have those features when water is not present.”</p>
4	4	3	3	<p>“Drainage connectivity</p> <p>What single process or series of processes could have produced a landscape.</p> <p>What inherent geologic conditions could exist to produce a landscape.</p> <p>How would sediment move through the landscape.</p> <p>What climatic conditions could exist to produce it.</p> <p>Is this a landscape I could see developing on Earth or Mars.</p> <p>Is there extreme topographic development with no obvious source or sink for sediment.</p> <p>Is this something I could see as a 100m DEM or a 10m DEM.”</p>
4	4	3	3	Drainage network and presence/absence of closed basins
4	4	3	3	<p>“Fractal patterns in erosional gullies always look more realistic, especially in flat topography. Rounded surfaces usually looked less realistic.</p> <p>Gullying and roughness on volcanic features where slopes are steep is realistic.”</p>
4	4	3	3	I avoided topographies with unusual lineations, nearly-identical repeated sections, recognizable symbols, and higher relief in lower elevations.
4	4	3	3	I used gut instinct. I benefit from a PhD in geomorphology and work in numerical modeling of landscape evolution.

Expertise				
G_i	C_i	I_i	A_i	Response
4	4	3	3	Judging the likely geomorphic processes of the landscapes and looking for evidence of them. For example, images with aeolian or alluvial processes tend to be broad and flat in the lowlands, and often have well defined channels / canyons as outlets onto the plains. Or ridge and valley topography with high soil production or low uplift / erosion often has broad ridge crests with small side valleys eating into the ridges.
4	4	3	3	Look for closed basins, or large rivers draining into blind valleys. Look for odd distributions of raised "bumps", especially when their alignment was inconsistent throughout the image. Look for mismatch between the sharpness of small features and the smoothness of the overall terrain.
4	4	3	3	Recurring features
4	4	3	3	"topographical irregularity or homogeneity, organic flow. Please note some were both very unrealistic and i couldn't mark them as realistic, they were equally unrealistic"

Bibliography

- [1] ADAMS, D., EGBERT, P., AND BRUNNER, S. Feature-based interactively sketched terrain. In *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games* (New York, NY, USA, 2012), I3D '12, ACM, pp. 208–208.
- [2] ALEXEJ PERSSON, M. Terrain generation, part 1, (9 March 2011). <https://notch.tumblr.com/post/3746989361/terrain-generation-part-1> Accessed: 28-06-2019.
- [3] ANH, N. H., SOURIN, A., AND ASWANI, P. Physically based hydraulic erosion simulation on graphics processing unit. In *Proceedings of the 5th International Conference on Computer Graphics and Interactive Techniques in Australia and Southeast Asia* (New York, NY, USA, 2007), GRAPHITE '07, ACM, pp. 257–264.
- [4] ARGUDO, O., ANDUJAR, C., CHICA, A., GUÉRIN, E., DIGNE, J., PEYTAVIE, A., AND GALIN, E. Coherent multi-layer landscape synthesis. *Vis. Comput.* 33, 6-8 (June 2017), 1005–1015.
- [5] ARIYAN, M., AND MOULD, D. Terrain synthesis using curve networks. In *Proceedings of the 41st Graphics Interface Conference* (CAN, 2015), GI '15, Canadian Information Processing Society, p. 9–16.
- [6] ASHIKHMIN, M. Synthesizing natural textures. In *Proceedings of the 2001 Symposium on Interactive 3D Graphics* (New York, NY, USA, 2001), I3D '01, ACM, pp. 217–226.

- [7] ASTER. Aster global digital elevation map announcement. <https://asterweb.jpl.nasa.gov/gdem.asp> Accessed: 28-06-2019.
- [8] BANGAY, S., DE BRUYN, D., AND GLASS, K. Minimum spanning trees for valley and ridge characterization in digital elevation maps. In *Proceedings of the 7th International Conference on Computer Graphics, Virtual Reality, Visualisation and Interaction in Africa* (2010), ACM, pp. 73–82.
- [9] BARKAN, K. What's wrong with the visual effects industry?), (28 February 2014). <https://www.siggraph.org/whats-wrong-with-the-visual-effects-industry/> Accessed: 28-06-2019.
- [10] BARNES, C., SHECHTMAN, E., FINKELSTEIN, A., AND GOLDMAN, D. B. Patchmatch: A randomized correspondence algorithm for structural image editing. In *ACM SIGGRAPH 2009 Papers* (New York, NY, USA, 2009), SIGGRAPH '09, ACM, pp. 24:1–24:11.
- [11] BARNES, C., SHECHTMAN, E., GOLDMAN, D. B., AND FINKELSTEIN, A. The generalized patchmatch correspondence algorithm. In *Proceedings of the 11th European Conference on Computer Vision Conference on Computer Vision: Part III* (Berlin, Heidelberg, 2010), ECCV'10, Springer-Verlag, pp. 29–43.
- [12] BARNES, R., LEHMAN, C., AND MULLA, D. An efficient assignment of drainage direction over flat surfaces in raster digital elevation models. *Comput. Geosci.* 62 (Jan. 2014), 128–135.
- [13] BARNES, R., LEHMAN, C., AND MULLA, D. Priority-flood: An optimal depression-filling and watershed-labeling algorithm for digital elevation models. *Computers & Geosciences* 62 (2014), 117–127.
- [14] BECHER, M., KRONE, M., REINA, G., AND ERTL, T. Feature-based volumetric terrain generation. In *Proceedings of the 21st ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games* (New York, NY, USA, 2017), I3D '17, ACM, pp. 10:1–10:9.
- [15] BELHADJ, F. Terrain modeling: A constrained fractal model. In *Proceedings of the 5th International Conference on Computer Graphics, Virtual Reality,*

- Visualisation and Interaction in Africa* (New York, NY, USA, 2007), AFRIGRAPH '07, ACM, pp. 197–204.
- [16] BELHADJ, F., AND AUDIBERT, P. Modeling landscapes with ridges and rivers: Bottom up approach. In *Proceedings of the 3rd International Conference on Computer Graphics and Interactive Techniques in Australasia and South East Asia* (New York, NY, USA, 2005), GRAPHITE '05, ACM, pp. 447–450.
- [17] BENEŠ, B. Physically-based hydraulic erosion. In *Proceedings of the 22Nd Spring Conference on Computer Graphics* (New York, NY, USA, 2006), SCCG '06, ACM, pp. 17–22.
- [18] BENEŠ, B., AND FORSBACH, R. Visual simulation of hydraulic erosion. *Journal of WSCG* 10, 2 (2002), 79–86.
- [19] BENEŠ, B., TĚŠÍNSKÝ, V., HORNYŠ, J., AND BHATIA, S. K. Hydraulic erosion. *Computer Animation and Virtual Worlds* 17, 2 (2006), 99–108.
- [20] BENEŠ, B. Real-time erosion using shallow water simulation. In *4th Workshop in Virtual Reality Interactions and Physical Simulation* (2007), VRIPHYS'07, pp. 43–50.
- [21] BENEŠ, B., AND ARRIAGA, X. Table mountains by virtual erosion. In *Proceedings of the First Eurographics Conference on Natural Phenomena* (Aire-la-Ville, Switzerland, Switzerland, 2005), NPH'05, Eurographics Association, pp. 33–40.
- [22] BENEŠ, B., AND FORSBACH, R. Layered data representation for visual simulation of terrain erosion. In *Proceedings of the 17th Spring Conference on Computer Graphics* (Washington, DC, USA, 2001), SCCG '01, IEEE Computer Society, pp. 80–86.
- [23] BENEŠ, B., AND FORSBACH, R. Parallel implementation of terrain erosion applied to the surface of mars. In *Proceedings of the 1st International Conference on Computer Graphics, Virtual Reality and Visualisation* (New York, NY, USA, 2001), AFRIGRAPH '01, ACM, pp. 53–57.

- [24] BERNHARDT, A., MAXIMO, A., VELHO, L., HNAIDI, H., AND CANI, M.-P. Real-time terrain modeling using cpu-gpu coupled computation. In *Proceedings of the 2011 24th SIBGRAPI Conference on Graphics, Patterns and Images* (Washington, DC, USA, 2011), SIBGRAPI '11, IEEE Computer Society, pp. 64–71.
- [25] BILAND, J., AND ÇÖLTEKIN, A. An empirical assessment of the impact of the light direction on the relief inversion effect in shaded relief maps: Nnw is better than nw. *Cartography and Geographic Information Science* 44, 4 (2017), 358–372.
- [26] BLINN, J. Consider the lowly 2×2 matrix. *IEEE Computer Graphics and Applications* 16, 2 (1996), 82–88.
- [27] BLOOMENTHAL, J., AND WYVILL, B., Eds. *Introduction to Implicit Surfaces*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1997.
- [28] BOOKSTEIN, F. L. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Trans. Pattern Anal. Mach. Intell.* 11, 6 (June 1989), 567–585.
- [29] BRADLEY, R. A., AND TERRY, M. E. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika* 39, 3/4 (1952), 324–345.
- [30] BROSZ, J., SAMAVATI, F. F., AND SOUSA, M. C. Terrain synthesis by-example. In *Advances in Computer Graphics and Computer Vision: International Conferences VISAPP and GRAPP 2006*, J. Braz, A. Ranchordas, H. Araújo, and J. Jorge, Eds. Springer, Berlin, Heidelberg, 2007, ch. Terrain Synthesis By-Example, pp. 58–77.
- [31] CANDÈS, E. J., ROMBERG, J. K., AND TAO, T. Stable signal recovery from incomplete and inaccurate measurements. *Communications on Pure and Applied Mathematics* 59, 8 (2006), 1207–1223.
- [32] CANNY, J. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-8*, 6 (Nov 1986), 679–698.

- [33] CHANG, Y.-C., SONG, G.-S., AND HSU, S.-K. Automatic extraction of ridge and valley axes using the profile recognition and polygon-breaking algorithm. *Computers & Geosciences* 24, 1 (1998), 83–93.
- [34] CHIBA, N., MURAOKA, K., AND FUJITA, K. An erosion model based on velocity fields for the visual simulation of mountain scenery. *The Journal of Visualization and Computer Animation* 9, 4 (1998), 185–194.
- [35] CHIBA, N., MURAOKA, K., YAEGASHI, K., AND MIURA, M. Terrain simulation based on the recursive refinement of ridge-lines. In *Proceedings of CAD/Graphics* (1991), vol. 91, pp. 19–24.
- [36] CORDONNIER, G., BRAUN, J., CANI, M.-P., BENEŠ, B., GALIN, E., PEYTAVIE, A., AND GUÉRIN, E. Large scale terrain generation from tectonic uplift and fluvial erosion. In *Computer Graphics Forum* (2016), vol. 35, Wiley Online Library, pp. 165–175.
- [37] CORDONNIER, G., CANI, M.-P., BENEŠ, B., BRAUN, J., AND GALIN, E. Sculpting mountains: Interactive terrain modeling based on subsurface geology. *IEEE transactions on visualization and computer graphics* 24, 5 (May 2018), 1756–1769.
- [38] CORDONNIER, G., GALIN, E., GAIN, J., BENEŠ, B., GUÉRIN, E., PEYTAVIE, A., AND CANI, M.-P. Authoring landscapes by combining ecosystem and terrain erosion simulation. *ACM Trans. Graph.* 36, 4 (July 2017), 134:1–134:12.
- [39] CRIMINISI, A., PEREZ, P., AND TOYAMA, K. Region filling and object removal by exemplar-based image inpainting. *Trans. Img. Proc.* 13, 9 (Sept. 2004), 1200–1212.
- [40] CRUZ, L., AND VELHO, L. *High-Level Techniques for Landscape Creation*. PhD thesis, The Instituto Nacional de Matemática Pura e Aplicada, Rio de Janeiro, Brazil, 3 2015.
- [41] CRUZ, L., VELHO, L., GALIN, E., PEYTAVIE, A., AND GUÉRIN, E. Patch-based terrain synthesis. In *International Conference on Computer Graphics Theory*

- and Applications* (Berlin, France, 2015), Proceedings of the 10th International Conference on Computer Graphics Theory and Applications,, GRAPP.
- [42] CUI, G., WILLIAMS, B., AND KUCZERA, G. A stochastic tokunaga model for stream networks. *Water resources research* 35, 10 (1999), 3139–3147.
- [43] DACHSBACHER, C., MEYER, M., AND STAMMINGER, M. Height-field synthesis by non-parametric sampling. *Vision, Modeling and Visualization* 2005 (2005), 297–302.
- [44] DARABI, S., SHECHTMAN, E., BARNES, C., GOLDMAN, D. B., AND SEN, P. Image melding: Combining inconsistent images using patch-based synthesis. *ACM Trans. Graph.* 31, 4 (2012), 82–1.
- [45] DAVIDSON, R. R., FARQUHAR, P. H., ET AL. A bibliography on the method of paired comparisons. *Biometrics* 32, 2 (1976), 241–252.
- [46] DE CARPENTIER, G. J. P., AND BIDARRA, R. Interactive gpu-based procedural heightfield brushes. In *Proceedings of the 4th International Conference on Foundations of Digital Games* (New York, NY, USA, 2009), FDG '09, ACM, pp. 55–62.
- [47] DERZAPF, E., GANSTER, B., GUTHE, M., AND KLEIN, R. River networks for instant procedural planets. In *Computer Graphics Forum* (2011), vol. 30, Wiley Online Library, pp. 2031–2040.
- [48] DIJKSTRA, E. W. A note on two problems in connexion with graphs. *Numerische mathematik* 1, 1 (1959), 269–271.
- [49] DODGSON, N., PATTERSON, J., AND WILLIS, P. What’s up prof? current issues in the visual effects & post-production industry. *Leonardo* 43, 1 (2010), 92–93.
- [50] DODGSON, N. A. Going to the movies: Lessons from the film industry for 3d libraries. In *3D Research Challenges in Cultural Heritage: A Roadmap in Digital Heritage Preservation*, M. Ioannides and E. Quak, Eds. Springer Berlin Heidelberg, Berlin, Heidelberg, 2014, pp. 93–103.

- [51] DORAN, J., AND PARBERRY, I. Controlled procedural terrain generation using software agents. *IEEE Transactions on Computational Intelligence and AI in Games* 2, 2 (2010), 111–119.
- [52] DOS PASSOS, V. A., AND IGARASHI, T. Landsketch: A first person point-of-view example-based terrain modeling approach. In *Proceedings of the International Symposium on Sketch-Based Interfaces and Modeling* (New York, NY, USA, 2013), SBIM '13, ACM, pp. 61–68.
- [53] DUCHON, J. Splines minimizing rotation-invariant semi-norms in sobolev spaces. In *Constructive theory of functions of several variables*. Springer, 1977, pp. 85–100.
- [54] EFROS, A. A., AND FREEMAN, W. T. Image quilting for texture synthesis and transfer. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 2001), SIGGRAPH '01, ACM, pp. 341–346.
- [55] EFROS, A. A., AND LEUNG, T. K. Texture synthesis by non-parametric sampling. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on* (Washington, DC, USA, 1999), vol. 2 of ICCV '99, IEEE Computer Society, pp. 1033–1038.
- [56] FECHNER, G. T. *Elemente der Psychophysik: Zweiter Teil*. Breitkopf & Härtel, 1889.
- [57] FORD, L., AND FULKERSON, D. R. Flows in networks. 1962. *Princeton U. Press, Princeton, NJ* (1962).
- [58] FOURNIER, A., FUSSELL, D., AND CARPENTER, L. Computer rendering of stochastic models. *Commun. ACM* 25, 6 (June 1982), 371–384.
- [59] FRIEDMAN, M. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the American Statistical Association* 32, 200 (1937), 675–701.

- [60] GAIN, J., MARAIS, P., AND STRASSER, W. Terrain sketching. In *Proceedings of the 2009 Symposium on Interactive 3D Graphics and Games* (New York, NY, USA, 2009), I3D '09, ACM, pp. 31–38.
- [61] GAIN, J., MERRY, B., AND MARAIS, P. Parallel, realistic and controllable terrain synthesis. *Computer Graphics Forum* 34, 2 (2015), 105–116.
- [62] GAME INFORMER. A behind-the-scenes tour of no man's sky's technology. [Video file], (5 December 2014). <https://youtu.be/h-kifCYToAU> Accessed: 28-06-2019.
- [63] GAMITO, M. N., AND MUSGRAVE, F. K. Procedural landscapes with overhangs. In *10th Portuguese Computer Graphics Meeting* (2001), vol. 2, p. 3.
- [64] GEISS, R. Generating complex procedural terrains using the gpu. In *Gpu Gems 3*, H. Nguyen, Ed., first ed., vol. 3. Addison-Wesley Professional, 2007, ch. 1, pp. 7–37.
- [65] GÉNEVAUX, J.-D., GALIN, E., GUÉRIN, E., PEYTAVIE, A., AND BENEŠ, B. Terrain generation using procedural models based on hydrology. *ACM Trans. Graph.* 32, 4 (July 2013), 143:1–143:13.
- [66] GÉNEVAUX, J.-D., GALIN, E., PEYTAVIE, A., GUÉRIN, E., BRIQUET, C., GROSBELLET, F., AND BENEŠ, B. Terrain modelling from feature primitives. *Computer Graphics Forum* 34, 6 (2015), 198–210.
- [67] GESCH, D., OIMOEN, M., GREENLEE, S., NELSON, C., STEUCK, M., AND TYLER, D. The national elevation dataset. *Photogrammetric engineering and remote sensing* 68, 1 (2002), 5–11.
- [68] GOLUBEV, K., ZAGARSKIKH, A., AND KARSAKOV, A. Dijkstra-based terrain generation using advanced weight functions. *Procedia Computer Science* 101 (2016), 152 – 160. 5th International Young Scientist Conference on Computational Science, YSC 2016, 26-28 October 2016, Krakow, Poland.
- [69] GUÉRIN, E., DIGNE, J., GALIN, E., AND PEYTAVIE, A. Sparse representation of terrains for procedural modeling. In *Computer Graphics Forum* (2016), vol. 35, Wiley Online Library.

- [70] GUÉRIN, E., DIGNE, J., GALIN, E., PEYTAVIE, A., WOLF, C., BENEŠ, B., AND MARTINEZ, B. Interactive example-based terrain authoring with conditional generative adversarial networks. *ACM Trans. Graph.* 36, 6 (Nov. 2017), 228:1–228:13.
- [71] HACOHEEN, Y., SHECHTMAN, E., GOLDMAN, D. B., AND LISCHINSKI, D. Non-rigid dense correspondence with applications for image enhancement. *ACM Trans. Graph.* 30, 4 (July 2011), 70:1–70:10.
- [72] HAN, C., RISSE, E., RAMAMOORTHY, R., AND GRINSFELD, E. Multiscale texture synthesis. In *ACM SIGGRAPH 2008 Papers* (New York, NY, USA, 2008), SIGGRAPH '08, ACM, pp. 51:1–51:8.
- [73] HAN, J., ZHOU, K., WEI, L.-Y., GONG, M., BAO, H., ZHANG, X., AND GUO, B. Fast example-based surface texture synthesis via discrete optimization. *The Visual Computer* 22, 9-11 (2006), 918–925.
- [74] HERTZMANN, A., JACOBS, C. E., OLIVER, N., CURLESS, B., AND SALESIN, D. H. Image analogies. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques* (2001), ACM, pp. 327–340.
- [75] HNAIDI, H., GUÉRIN, E., AKKOUCH, S., PEYTAVIE, A., AND GALIN, E. Feature based terrain generation using diffusion equation. In *Computer Graphics Forum* (2010), vol. 29, Wiley Online Library, pp. 2179–2186.
- [76] HOLM, S. A simple sequentially rejective multiple test procedure. *Scandinavian journal of statistics* (1979), 65–70.
- [77] ITO, T., FUJIMOTO, T., MURAOKA, K., AND CHIBA, N. Modeling rocky scenery taking into account joints. In *Computer Graphics International, 2003. Proceedings* (July 2003), IEEE, pp. 244–247.
- [78] JÁKÓ, B., AND TÓTH, B. Fast hydraulic and thermal erosion on gpu. In *Proc. of Eurographics* (2011), Citeseer, pp. 57–60.
- [79] KAMAL, K. R., AND UDDIN, Y. S. Parametrically controlled terrain generation. In *Proceedings of the 5th International Conference on Computer*

- Graphics and Interactive Techniques in Australia and Southeast Asia* (New York, NY, USA, 2007), GRAPHITE '07, ACM, pp. 17–23.
- [80] KASPAR, A., NEUBERT, B., LISCHINSKI, D., PAULY, M., AND KOPF, J. Self tuning texture optimization. In *Computer Graphics Forum* (2015), vol. 34, Wiley Online Library, pp. 349–359.
- [81] KASS, M., AND MILLER, G. Rapid, stable fluid dynamics for computer graphics. In *ACM Siggraph Computer Graphics* (1990), vol. 24, ACM, pp. 49–57.
- [82] KELLEY, A. D., MALIN, M. C., AND NIELSON, G. M. Terrain simulation using a model of stream erosion. In *Proceedings of the 15th Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 1988), SIGGRAPH '88, ACM, pp. 263–268.
- [83] KOLÁŘ, M., DEBATTISTA, K., AND CHALMERS, A. A subjective evaluation of texture synthesis methods. In *Computer Graphics Forum* (2017), vol. 36, Wiley Online Library, pp. 189–198.
- [84] KRTEN, R. Generating realistic terrain. *Dr Dobb's Journal-Software Tools for the Professional Programmer* 19, 7 (1994), 26–31.
- [85] KWATRA, V., ESSA, I., BOBICK, A., AND KWATRA, N. Texture optimization for example-based synthesis. In *ACM SIGGRAPH 2005 Papers* (New York, NY, USA, 2005), SIGGRAPH '05, ACM, pp. 795–802.
- [86] KWATRA, V., SCHÖDL, A., ESSA, I., TURK, G., AND BOBICK, A. Graphcut textures: image and video synthesis using graph cuts. In *ACM Transactions on Graphics (ToG)* (2003), vol. 22, ACM, pp. 277–286.
- [87] LEFEBVRE, S., AND HOPPE, H. Parallel controllable texture synthesis. In *ACM SIGGRAPH 2005 Papers* (New York, NY, USA, 2005), SIGGRAPH '05, ACM, pp. 777–786.
- [88] LEFEBVRE, S., AND HOPPE, H. Appearance-space texture synthesis. In *ACM SIGGRAPH 2006 Papers* (New York, NY, USA, 2006), SIGGRAPH '06, ACM, pp. 541–548.

- [89] LEWIS, J. P. Generalized stochastic subdivision. *ACM Trans. Graph.* 6, 3 (July 1987), 167–190.
- [90] LINDSAY, J. B. Efficient hybrid breaching-filling sink removal methods for flow path enforcement in digital elevation models. *Hydrological Processes* 30, 6 (2016), 846–857.
- [91] LINDSAY, J. B., AND DHUN, K. Modelling surface drainage patterns in altered landscapes using lidar. *International Journal of Geographical Information Science* 29, 3 (2015), 397–411.
- [92] MANDELBROT, B. B. Stochastic models for the earth's relief, the shape and the fractal dimension of the coastlines, and the number-area rule for islands. *Proceedings of the National Academy of Sciences* 72, 10 (1975), 3825–3828.
- [93] MANDELBROT, B. B. *The Fractal Geometry of Nature*. W. H. Freeman and Company, 1977.
- [94] MANDELBROT, B. B. Fractal landscapes without creases and with rivers. In *The Science of Fractal Images*, H.-O. Peitgen and D. Saupe, Eds. Springer-Verlag New York, Inc., New York, NY, USA, 1988, ch. Fractal Landscapes Without Creases and with Rivers, pp. 243–260.
- [95] MAPMART, H. Harris-intrasearch geoeye-1 1m dsm and 5m dtm. <http://cms.mapmart.com/Products/DigitalElevationModel/HIDEMGeoEye1.aspx> Accessed: 30-06-2016.
- [96] MCGUIRE, M. *An eye for fractals: a graphic/photographic essay*. Addison-Wesley Longman Publishing Co., Inc., 1991.
- [97] MEI, X., DECAUDIN, P., AND HU, B.-G. Fast hydraulic erosion simulation and visualization on gpu. In *PG '07 - 15th Pacific Conference on Computer Graphics and Applications* (Maui, United States, Oct. 2007), M. Alexa, S. J. Gortler, and T. Ju, Eds., Pacific Graphics 2007, IEEE, pp. 47–56.
- [98] MICHEL, E., EMILIEN, A., AND CANI, M.-P. Generation of folded terrains from simple vector maps. In *Eurographics 2015 short paper proceedings* (Zurich,

- Switzerland, May 2015), B. Bickel and T. Ritschel, Eds., The Eurographics Association, p. 4.
- [99] MILLER, G. S. P. The definition and rendering of terrain maps. In *Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 1986), SIGGRAPH '86, ACM, pp. 39–48.
- [100] MONMONIER, M. *Color: Attraction and Distraction*. University of Chicago Press, 1996, ch. 11, pp. 163–173.
- [101] MOSSMAN, J. New color system enhances relief maps. <https://www.esri.com/news/arcuser/0101/shademax.html> Accessed: 28-06-2019.
- [102] MUSGRAVE, F. K., KOLB, C. E., AND MACE, R. S. The synthesis and rendering of eroded fractal terrains. In *Proceedings of the 16th Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 1989), SIGGRAPH '89, ACM, pp. 41–50.
- [103] NAGASHIMA, K. Computer generation of eroded valley and mountain terrains. *The Visual Computer* 13, 9 (1998), 456–464.
- [104] NATALI, M., LIDAL, E. M., PARULEK, J., VIOLA, I., AND PATEL, D. Modeling terrains and subsurface geology. In *EuroGraphics 2013 State of the Art Reports (STARs), 2013* (2013), Eurographics, Eurographics 2013 - State of the Art Reports, pp. 155–173.
- [105] NEEDELS, C. J. Parameterization of terrain in army combat analysis. Master's thesis, NAVAL POSTGRADUATE SCHOOL MONTEREY CALIFORNIA, 1976.
- [106] NEIDHOLD, B., WACKER, M., AND DEUSSEN, O. Interactive physically based fluid and erosion simulation. In *Proceedings of the First Eurographics Conference on Natural Phenomena* (Aire-la-Ville, Switzerland, Switzerland, 2005), NPH'05, Eurographics Association, pp. 25–33.
- [107] NEW YORKER. No man's sky exclusive demo: Part two — the new yorker festival. [Video file], (5 December 2014). <https://youtu.be/CqUd3qj102Y> Accessed: 28-06-2019.

- [108] O'BRIEN, J. F., AND HODGINS, J. K. Dynamic simulation of splashing fluids. In *Proceedings Computer Animation'95* (1995), IEEE, pp. 198–205.
- [109] OLSEN, J. Realtime procedural terrain generation-realtime synthesis of eroded fractal terrain for use in computer games, 2004.
- [110] PARBERRY, I. Designer worlds: Procedural generation of infinite terrain from real-world elevation data. *Journal of Computer Graphics Techniques (JCGT)* 3, 1 (March 2014), 74–85.
- [111] PARRY, S. H. The generation and use of parameterized terrain in land combat simulation. In *Proceedings of the 9th Conference on Winter Simulation - Volume 1* (1977), WSC '77, Winter Simulation Conference, pp. 422–431.
- [112] PATTERSON, T., AND JENNY, B. The development and rationale of cross-blended hypsometric tints. *Cartographic Perspectives*, 69 (2011), 31–46.
- [113] PÉREZ, P., GANGNET, M., AND BLAKE, A. Poisson image editing. In *ACM Transactions on Graphics (TOG)* (2003), vol. 22, ACM, pp. 313–318.
- [114] PERLIN, K. An image synthesizer. *SIGGRAPH Comput. Graph.* 19, 3 (July 1985), 287–296.
- [115] PETRILLO, F., PIMENTA, M., TRINDADE, F., AND DIETRICH, C. Houston, we have a problem...: A survey of actual problems in computer games development. In *Proceedings of the 2008 ACM Symposium on Applied Computing* (New York, NY, USA, 2008), SAC '08, ACM, pp. 707–711.
- [116] PEYTAVIE, A., GALIN, E., GROSJEAN, J., AND MERILLOU, S. Arches: a framework for modeling complex terrains. In *Computer Graphics Forum* (2009), vol. 28, Wiley Online Library, pp. 457–467.
- [117] PICKOVER, C. A. Generating extraterrestrial terrain. *IEEE Computer Graphics and Applications* 15, 2 (Mar. 1995), 18–21.
- [118] PLANETSIDES SOFTWARE. Terragen. [Computer software]. <https://planetside.co.uk/> Accessed: 28-06-2019.

- [119] PRUSINKIEWICZ, P., AND HAMMEL, M. A fractal model of mountains and rivers. In *Proceedings of Graphics Interface '93* (1993), vol. 93, CANADIAN INFORMATION PROCESSING SOCIETY, pp. 174–180.
- [120] QGIS DEVELOPMENT TEAM. *QGIS Geographic Information System*. Open Source Geospatial Foundation, 2009.
- [121] QUALTRICS LABS, INC. Qualtrics, 2009. <https://www.qualtrics.com> Accessed: 28-06-2019.
- [122] RED BLOB GAMES. Polygonal map generation for games, (4 September 2010). <http://www-cs-students.stanford.edu/~amitp/game-programming/polygon-map-generation/> Accessed: 28-06-2018.
- [123] ROSGEN, D. L. A classification of natural rivers. *Catena* 22, 3 (1994), 169–199.
- [124] ROUDIER, P., PEROCHE, B., AND PERRIN, M. Landscapes synthesis achieved through erosion and deposition process simulation. *Computer Graphics Forum* 12, 3 (1993), 375–383.
- [125] RUSNELL, B., MOULD, D., AND ERAMIAN, M. Feature-rich distance-based terrain synthesis. *The Visual Computer* 25, 5-7 (Apr. 2009), 573–579.
- [126] SAUNDERS, R. L. Realistic terrain synthesis using genetic algorithms. Master's thesis, Texas A&M University, 2006.
- [127] SAUPE, D. Point evaluation of multi-variable random fractals. In *Visualisierung in Mathematik und Naturwissenschaften*. Springer, 1989, pp. 114–126.
- [128] SCHNEIDER, J., BOLDTE, T., AND WESTERMANN, R. Real-time editing, synthesis, and rendering of infinite landscapes on gpus. In *Proc. of Vision, modeling, and visualization* (2006), p. 145.
- [129] SHEPARD, D. A two-dimensional interpolation function for irregularly-spaced data. In *Proceedings of the 1968 23rd ACM national conference* (1968), ACM, pp. 517–524.

- [130] SIMAKOV, D., CASPI, Y., SHECHTMAN, E., AND IRANI, M. Summarizing visual data using bidirectional similarity. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on* (2008), IEEE, pp. 1–8.
- [131] SOILLE, P. Morphological carving. *Pattern Recognition Letters* 25, 5 (2004), 543–550.
- [132] STACHNIAK, S., AND STUERZLINGER, W. An algorithm for automated fractal terrain deformation. In *In Proceedings of Computer Graphics and Artificial Intelligence* (2005), vol. 1, pp. 64–76.
- [133] STATISTA. Video games are getting as expensive as movies. <https://www.statista.com/chart/2713/video-games-are-getting-as-expensive-as-movies/> Accessed: 28-06-2019.
- [134] STATISTA. The world’s most expensive film productions as of april 2018 (in million u.s. dollars). <https://www.statista.com/statistics/273416/most-expensive-film-productions-worldwide/> Accessed: 28-06-2019.
- [135] ŠT’AVA, O., BENEŠ, B., BRISBIN, M., AND KŘIVÁNEK, J. Interactive terrain modeling using hydraulic erosion. In *Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Aire-la-Ville, Switzerland, Switzerland, 2008), SCA ’08, Eurographics Association, pp. 201–210.
- [136] TALGORN, F.-X., AND BELHADJ, F. Real-time sketch-based terrain generation. In *Proceedings of Computer Graphics International 2018* (New York, NY, USA, 2018), CGI 2018, ACM, pp. 13–18.
- [137] TASSE, F. P., GAIN, J., AND MARAIS, P. Enhanced texture-based terrain synthesis on graphics hardware. *Computer Graphics Forum* 31, 6 (2012), 1959–1972.
- [138] TEOH, S. T. River and coastal action in automatic terrain generation. In *Proceedings of the 2008 International Conference on Computer Graphics & Virtual Reality* (2008), Citeseer, pp. 3–9.

- [139] THE NUMBERS. Movie budget and financial performance records. <https://www.the-numbers.com/movie/budgets/> Accessed: 28-06-2019.
- [140] TONG, X., ZHANG, J., LIU, L., WANG, X., GUO, B., AND SHUM, H.-Y. Synthesis of bidirectional texture functions on arbitrary surfaces. In *Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 2002), SIGGRAPH '02, ACM, pp. 665–672.
- [141] TUKEY, J. W., ET AL. Comparing individual means in the analysis of variance. *Biometrics* 5, 2 (1949), 99–114.
- [142] UBISOFT. Ubisoft reports first-half 2013-14 sales and earnings figures. [Press release], (12 November 2013). <https://www.ubisoft.com/en-US/company/press/detail.aspx?id=119686> Accessed: 28-06-2018.
- [143] UNITED NATIONS ENVIRONMENT PROGRAMME. Endorheic lakes: Waterbodies that don't flow to the sea. http://www.unep.or.jp/ietc/publications/short_series/lakereservoirs-2/10.asp Accessed: 28-06-2019.
- [144] USGS. Shuttle radar topography mission, 2006. College Park, Maryland, February 2000.
- [145] VANEK, J., BENEŠ, B., HEROUT, A., AND STAVA, O. Large-scale physics-based terrain editing using adaptive tiles on the gpu. *IEEE Computer Graphics and Applications* 31, 6 (2011), 35–44.
- [146] VERGNE, R., PACANOWSKI, R., BARLA, P., GRANIER, X., AND SCHLICK, C. Radiance scaling for versatile surface enhancement. In *Proceedings of the 2010 ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games* (New York, NY, USA, 2010), I3D '10, pp. 143–150.
- [147] WALLRAVEN, C., AND CUNNINGHAM, D. W. *Experimental design: From user studies to psychophysics*. AK Peters/CRC Press, 2011.
- [148] WEI, L.-Y., LEFEBVRE, S., KWATRA, V., AND TURK, G. State of the art in example-based texture synthesis. In *Eurographics 2009, State of the Art Report, EG-STAR* (2009), Eurographics Association, pp. 93–117.

- [149] WEI, L.-Y., AND LEVOY, M. Fast texture synthesis using tree-structured vector quantization. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 2000), SIGGRAPH '00, ACM Press/Addison-Wesley Publishing Co., pp. 479–488.
- [150] WEI, L.-Y., AND LEVOY, M. Order-independent texture synthesis, 2003. <https://graphics.stanford.edu/papers/texture-synthesis-sig03/> Accessed: 28-06-2019 (Earlier version is Stanford University Computer Science TR-2002-01).
- [151] WEXLER, Y., SHECHTMAN, E., AND IRANI, M. Space-time completion of video. *IEEE Transactions on pattern analysis and machine intelligence* 29, 3 (2007), 463–476.
- [152] WHAT CULTURE. Fallout 4's map is bigger than skyrim, but still less than the witcher 3, (11 November 2015). <http://whatculture.com/gaming/fallout-4s-map-is-bigger-than-skyrim-but-still-less-than-the-witcher-3> Accessed: 28-06-2019.
- [153] WHIPPLE, K. X., AND TUCKER, G. E. Dynamics of the stream-power river incision model: Implications for height limits of mountain ranges, landscape response timescales, and research needs. *Journal of Geophysical Research: Solid Earth* 104, B8 (1999), 17661–17674.
- [154] WU, Q., AND YU, Y. Feature matching and deformation for texture synthesis. In *ACM SIGGRAPH 2004 Papers* (New York, NY, USA, 2004), SIGGRAPH '04, ACM, pp. 364–367.
- [155] WYVILL, B., GUY, A., AND GALIN, E. Extending the csg tree. warping, blending and boolean operations in an implicit surface modeling system. *Computer Graphics Forum* 18, 2 (1999), 149–158.
- [156] ZELINKA, S., AND GARLAND, M. Towards real-time texture synthesis with the jump map. In *Proceedings of the 13th Eurographics Workshop on Rendering* (Aire-la-Ville, Switzerland, Switzerland, 2002), EGRW '02, Eurographics Association, pp. 99–104.

- [157] ZELINKA, S., AND GARLAND, M. Jump map-based interactive texture synthesis. *ACM Trans. Graph.* 23, 4 (Oct. 2004), 930–962.
- [158] ZHANG, H., MA, Z., LIU, Y., HE, X., AND MA, Y. A new skeleton feature extraction method for terrain model using profile recognition and morphological simplification. *Mathematical Problems in Engineering* 2013 (2013).
- [159] ZHANG, H., QU, D., HOU, Y., GAO, F., AND HUANG, F. Synthetic modeling method for large scale terrain based on hydrology. *IEEE Access* 4 (2016), 6238–6249.
- [160] ZHOU, H., SUN, J., TURK, G., AND REHG, J. M. Terrain synthesis from digital elevation models. *IEEE Transactions on Visualization and Computer Graphics* 13, 4 (July 2007), 834–848.
- [161] ZHOU, Y., SHI, H., LISCHINSKI, D., GONG, M., KOPF, J., AND HUANG, H. Analysis and controlled synthesis of inhomogeneous textures. In *Computer Graphics Forum* (2017), vol. 36, Wiley Online Library, pp. 199–212.