Exploring Texture Synthesis Techniques in Terrain Generation

Literature Review

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

ABSTRACT

Terrain generation using texture synthesis techniques can create realistic, adaptable, and efficient terrains. However, there are many ways to do this, with different advantages and disadvantages. This review presents texture synthesis options and compares them qualitatively and quantitatively. Quantitative comparisons were chosen as the efficiency and scalability evaluation procedures, while user experiments were selected for realism and user adaptability.

Scott and Dodgson's [18] terrain optimisation technique had notable results for realism, especially when combined with their pit-removal algorithm [18]. Efros and Freeman's [7] image quilting algorithm and Xiao et al.'s [23] rock materials algorithm present promising avenues for future research and development.

Gain et al.'s [9] algorithm delivered itself as the premiere texture synthesis technique. Gain et al.'s [9] interface allows for impressive user adaptability. The algorithm's efficiency and scalability also promise development in texture synthesis methods for larger-scale terrain generation.

CCS CONCEPTS

• **Computing methodologies → Shape modeling**; *Image manipulation, Texturing, Image-based rendering*.

KEYWORDS

Computer Graphics, Terrain Generation, Texture Synthesis

1 INTRODUCTION

Terrain generation is utilised in various fields, including films, games, simulations, training, and even epidemiology research [15]. The challenge of creating realistic terrain that is achieved efficiently, scaled correctly, and adaptable to user requirements has been researched for decades. However, very little research and successful implementation of terrain generation has been completed on a global scale. This review will focus on bare-earth terrain, terrain without any additional features such as trees, bushes, and buildings. The synthesising of planets with bare-earth terrain is a difficult task due to efficiency concerns, lack of terrain data for an entire planet, and, most significantly, the scale of the task.

1.1 Honours Project

Significant efficiency and scalability issues exist when generating planets using texture synthesis techniques, especially when real-world terrain data is used. The honours project that this forms part of aims to synthesize earth-like planets using real-world terrain data.

1.2 Relevance of Review

This research and its review inform which texture synthesis techniques are the most suitable for the terrain generation of earth-like planets, as well as in general. Texture synthesis is a technique used to generate realistic terrain by taking input images and creating output with similar characteristics [10]. This review focuses on example-based methods as the project scope stipulates using real-world terrain data. The aim is to explore the different techniques for texture synthesis and evaluate them as techniques for terrain generation. This testing is conducted based on select evaluation procedures chosen.

1.3 Overview of Review

This review explores different techniques for generating terrain using example-based models for texture synthesis. Example-based models are only as effective as their inputs, and quality input retrieval must be prioritised.

This review begins with a brief overview of related work that has already been conducted. Subsequently, an assessment of potential evaluation procedures will be completed. The patch-based approaches, the pixel-based approaches, and the recent approaches to texture synthesis are then assessed. Finally, all texture synthesis methods are discussed, with comparisons based on the chosen evaluation procedures. The conclusion of this review will reiterate the chosen evaluation procedures and indicate which texture synthesis method is preferred to generate terrain, in particular on a planetary scale.

2 ALTERNATE APPROACHES FOR TERRAIN GENERATION

There are many alternative methods for terrain generation, each with advantages and disadvantages. This review will briefly discuss these alternate methods to ascertain the relative effectiveness of example-based methods.

2.1 Traditional Approaches

There are various traditional methods for terrain generation. Procedural generation and simulation approaches are the most popular.

2.1.1 Procedural Generation. Procedural generation is an approach to terrain generation whereby the effects of geological phenomena are reproduced [10]. This differs from the simulation and example-based approaches as procedural generation does not simulate the actual geological processes or use real-terrain data as an input; it simply reproduces the effects. Procedural generation techniques are usually used in either large-scale terrain generation or specific landform generation. This large-scale terrain generation initially seems promising for planetary creation; however, procedural methods can struggle to meet the efficiency and adaptability requirements that researchers and users seek. Therefore, large-scale terrain generation using procedural generation would not be viable [10]. Although specific land-form generation offers increased adaptability for the user, it is unreliable on a large scale due to the nature of its localised approach.

More recently, Tripkovic [21], developed a tool for generating planets using procedural generation. The planets were generated by creating a mesh sphere, scaling it, and finally deforming the surface using three-dimensional procedural noise [21]. This solution also involved the placement of trees and bushes, which is outside the scope of this review. However, using procedural methods to generate planets helps compare procedural methods with example-based approaches like texture synthesis. Although the speed of the generation was satisfactory, the realism was not. The planets generated were very clearly computer-generated, and there was very little room for adaptation once they were created. Therefore, a procedural approach to creating planets would not be suitable in the context of this review.

2.1.2 Simulation Techniques. Simulation techniques model the causes and effects of a simulation process [10]. Simulation techniques are primarily developed based on erosion phenomena. Erosion is the movement of surface material from one location to another. Prominent simulation techniques model terrain using thermal, hydraulic, and tectonic erosion. Meanwhile, erosion, such as glacial, coastal, lightning, and aeolian erosion, is less used by simulation methods. Erosion techniques for simulation are computationally expensive and take a significant amount of time to simulate. While erosion simulations offer more geological accuracy due to the underlying physics, they provide little user control. The user often must change initial conditions and start over again if unsatisfied with the output [10]. Finally, erosion simulation does not scale well and would be too time-consuming to apply to generating planets. Therefore, simulation techniques would not be suited to the context of large-scale terrain generation.

2.2 Machine Learning

Machine learning techniques for generating terrain are a relatively new avenue of study. Machine learning requires extensive input data to train a model that can generate terrains. By training a conditional generative adversarial network (CGAN), a type of neural network, on Digital Elevation Model (DEM) exemplars, Guérin et al. [11] were able to train two competing neural networks — one for the creation of terrain and another for the assessment of the created terrain against real terrain. Digital Elevation Models or DEMs are collections of altitudes on a 2D grid [10]. The adaptability and authoring capabilities provided to the user using the machine learning technique are excellent [10]. This method allows users to annotate specific terrains that fit their preferences. The potential downside to machine learning is its reliance on high-quality input data and sufficient pre-processing time to train the models. With only one Earth-like planet, obtaining enough data to train a model to generate Earth-like planets may be difficult.

3 EVALUATION PROCEDURES

To effectively explore and compare different texture synthesis methods, a suite of constant evaluation procedures must be researched. These evaluation procedures will cover quantitative and qualitative evaluation and allow for a comprehensive assessment of the different texture synthesis methods and their four essential characteristics: scalability, realism, user adaptability, and efficiency.

3.1 Quantitative Evaluation

Firstly, the quantitative element of the suite of evaluation procedures to be explored includes scale testing, efficiency testing, geomorphological testing, and perceived terrain realism.

3.1.1 Scalability Testing. Synthesising planets requires texture synthesis methods that work well on large-scale terrain generation. The extent and precision of potential terrain generation methods are essential metrics to be accessed. The extent can be defined as the length of one side of a terrain in km and the precision as the distance (m or km) between grid samples in that dimension [10]. There is often a trade-off between extent and precision as increased precision drives up computational costs and limits the extent to which the method can be used in a reasonable amount of time. Example-based approaches to terrain generation allow for a range of extents with a fixed resolution by changing the sampling precision of the exemplars [10]. For a planetary generation, the scale needs to be extensive (over 5,000 km), but the precision does not need to be as high due to it being so large (over 2.5 km/pixel) [4, 9].

3.1.2 Efficiency Testing. Time and space efficiency are valuable tests to include in the suite because they give a comparable metric of the performance of the terrain generation technique. Time efficiency is the computation cost of the texture synthesis method. However, comparing can be difficult as most techniques use different hardware [10]. Texture synthesis techniques will likely result in time efficiency of longer than one minute [10], especially in contexts of increased scale. Space efficiency refers to the amount of storage space a given technique uses during terrain generation and in subsequent use. Example-based methods have higher memory overheads than procedural and simulation during generation because they are data-driven and not algorithm-driven [10]. However, in the subsequent use of the example-based methods, memory requirements should remain low at 1 − 4Mb [10].

3.1.3 Geomorphological Testing for Realism. Geomorphological testing for realism assumes that if an algorithm simulates the correct geological phenomena, it must be realistic [10]. However, this test is limited in its assessment of realism. It also requires complicated tests to be built to automate the evaluation procedure [10]. For these reasons, this review will instead use a qualitative measure of realism.

3.1.4 Perceived Terrain Realism. Perceived Terrain Realism Metrics (PTRM) is a set of perceptual metrics developed by Rajasekaran et al. [17]. These metrics are derived from an input set of normalised Geomorphons for a DEM terrain. Geomorphons are combinations of characteristics that describe geomorphic land-form features [17]. PTRM were found to have a positive relationship with the perceived realism of terrain. However, Geomorphons describe terrain based on ten land-form characteristics which are localised to small areas. Therefore, applying PTRM on a larger scale would be ineffective [17].

3.2 Qualitative Evaluation

Finally, the qualitative evaluation of texture synthesis techniques will allow for the assessment of elements such as realism and adaptability through user experiments.

3.2.1 User Experiments. User experiments provide promising evaluation potential for realism and adaptability requirements. For realism, this can be accessed via static renderings where the user can perform a side-by-side comparison of generated and real terrain [10]. For adaptability, the user is asked to attempt to author a terrain and subsequently questioned about the ease of adapting the terrain that the technique allows. Although these methods are simple to execute, they have some disadvantages. The user may not be accustomed to authoring or accessing bare-earth terrain, which may distort the results [10]. However, user experiments allow for informative feedback on adaptability, and realism can be identified. Therefore, user experiments remain a valuable form of qualitative evaluation that could be applied to any scale.

4 PATCH-BASED APPROACHES

The patch-based approach is the first class of texture synthesis techniques this review assesses. Patch-based approaches combine patches of terrain taken from a set of real terrain data [5]. First, it assesses an approach by Zhou et al. [25] that uses exemplar DEMs, and then, second, it assesses an adapted approach of the first by Tasse et al. [20].

The first was an early expedition into the use of texture synthesis in terrain generation. Zhou et al. [25] developed a patch-based approach that takes sections of the example data, which are DEMs, and attempts to match them to a user's sketch. This matching process is achieved through geomorphological comparison between the example height field (DEMs) and the user's sketch. The DEMs were taken from the U.S. Geological Survey which has a substantial amount of height field data. Once the matching has successfully occurred the final patches are built by overlapping matched patches and cleanly connecting them together using graph cuts and Poisson editing [25]. Graph cuts and Poisson editing are algorithms used to remove similar sets of edges [13, 16]. A tree-ordered patch placement algorithm is then used to determine the order in which to place the patches. A bread-first traversal was found to perform better than other placement orders [25]. Zhou et al.'s [25] method allows for initial user interaction with user designs to be clearly visible in the result. However, once generation begins, the user cannot adapt the terrain or control the elevation [10].

A wide variety of terrain can be synthesised using this method, and the quality of the terrain can be impressive as seen in Figure 1. The time and space efficiency are particularly notable. Terrain synthesis takes approximately 5 to 6 minutes on a processor with 2GB of memory [25]. However, there were some significant disadvantages to this thriving yet early foray into texture synthesis's use in terrain generation. The final terrain is entirely dependent on the available terrain data. The method cannot produce a specific terrain feature if it is not in the input data. In addition, the user

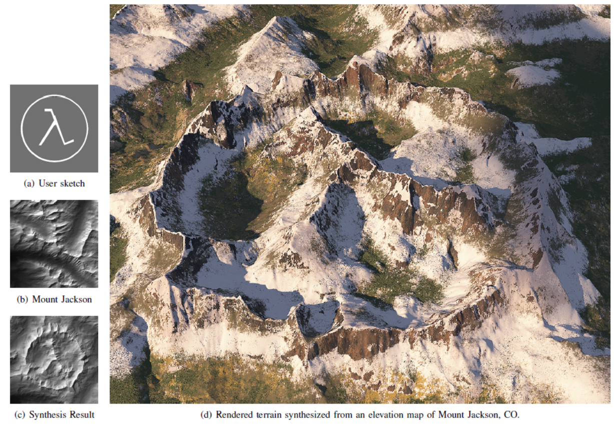

Figure 1: Output using Zhou et al.'s [25]'s synthesis algorithm with an inputted user sketch and Mount Jackson

must specify the patch size, affecting resolution and scale [25]. This, therefore, requires fine-tuning to achieve the optimal characteristics for a given terrain. Zhou et al.'s [25] algorithm has limitations regarding efficiency and control, and sometimes it generates clear patch seams.

An adapted process developed by Tasse et al. [20] was created to try to mitigate the limitations of the approach by Zhou et al. [25]. The adapted process aims to increase efficiency, improve control, and minimise noticeable patch seams [10]. Firstly, efficiency is increased by running the patch selection process in parallel. However, this requires graphics hardware to execute the complex ordering processes in parallel [20]. This parallelism speeds up the patch matching by a multiple of 6 to 225 times depending on specific patches being compared [20]. Secondly, improved control is attained by using a post-synthesis process to constrain the elevation of terrains to match user sketches [20].

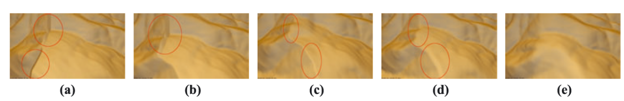

Figure 2: Comparison of patch merging techniques. (a) No patch merging. (b) Graphcut algorithm [13]. (c) Shepard Interpolation [19]. (d) Graphcut+Poisson seam removal [16] proposed by Zhou et al. [25]. (e) Tasse et al.'s Method [20]: Graphcut+Shepard Interpolation of the gradient+Poisson equation.

Lastly, Tasse et al. [20] use a Shepard gradient interpolation process [19] for patch merging in addition to the Poisson equation and the Graphcut algorithm used by Zhou et al. [25]. The different methods can be seen in Figure 2. Patch merging is the process of removing seams (artefacts) caused by the overlapping of patches [20]. These seams look unrealistic to the user and are an inherent issue of the patch-based approaches. User experiments by Tasse et al. [20] showed a decrease in recognised artefacts in their patch merging process compared to other processes. However, instances of bad matching where artefacts were evident were also found.

5 PIXEL-BASED APPROACHES

Pixel-based approaches generate terrain by assigning output pixels one at a time based on neighbouring pixels around the input exemplar [22]. An early exploration of pixel-based texture synthesis for terrain generation was proposed by Dachsbacher et al. [6]. More recently, Gain et al. [9] developed a parallel method for texture synthesis, significantly improving the efficiency and user adaptability from earlier methods.

Firstly, the matching algorithm proposed by Dachsbacher et al. [6] used a pixel-based non-parametric sampling method [8]. This method is accomplished by comparing the neighbourhoods in the input exemplar to those in an output area. It then matches similar neighbourhoods in the input and applies them around a single pixel at the output [10]. The matching algorithm was initially used for images, and therefore, the colour and space metrics needed to be replaced by terrain features such as elevation and slope [6]. Although this method represented a crucial early venture into texture synthesis, it still had limitations. Primarily, the issues revolved around precision and realism. Dachsbacher et al.'s [6] method used a blurring technique to remove notable artefacts, and this removed detail in the terrain [10]. The method did not produce realistic terrains and lacked efficiency.

Secondly, the more recent texture synthesis algorithm developed by Gain et al. [9] improves efficiency and user control by conducting the synthesis in parallel and enabling user authoring. This algorithm uses a database of DEM exemplars from the U.S. Geographical Survey [9].

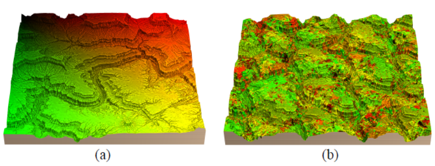

Figure 3: A patch view of synthesis: (a) single exemplar with coordinates mapped to red (x-axis) and green (y-axis), (b) resulting synthesis coloured according to exemplar coordinates. Figure by Gain et al. [9]

The key features of this algorithm are user control and realistic results. To improve user control and interactivity, the interface contains constraints, a copy-paste feature, and sketching and painting tools [9]. The realism created by this method is made possible mainly due to its novel synthesis framework. The texture synthesis framework is a parallel hierarchical structure that generates textures at various resolutions. Figure 3 illustrates an example of a patch undergoing the synthesis process. Effectively, the exemplar data ranges from low resolution (coarse) to high resolution (fine) and is designed into a pyramid. The pyramid is a reference pattern for the parallel synthesis [9]. To increase time efficiency, they adapted the texture synthesis method of Lefebvre et al. [14], a stacked multi-resolution technique [10]. This created a process that can synthesise different terrain parts simultaneously [9]. Upsampling and corrections are completed after execution to improve the realism of the terrain [9].

The synthesis averaged 63ms and 151ms for resolutions of 512^2 and 1024^2 [9]. However, curve constraints introduced additional overhead [9]. Regarding storage efficiency, 42.67 bytes per pixel were used in the synthesis pyramid and 84 bytes per pixel for the synthesis [9]. Overall, this method has a high performance and user acceptance. The limitations include a lack of geomorphological consideration and difficulty modelling fine details in flat plains [9].

6 RECENT TEXTURE SYNTHESIS APPROACHES

An analysis of recent explorations into texture synthesis is essential to assess the current algorithms available. A recent pit-removal algorithm has removed pits that lead to unrealistic terrains [18]. An image quilting process used as a simple, efficient texture synthesis algorithm [7]. Finally, a novel method for generating diverse rock materials [23].

Scott and Dodgson [18] have developed a method to improve global terrain realism and created an example-based multi-resolution terrain synthesis algorithm. The improvement to global terrain realism is accomplished by three pit-removal algorithms that reduce the presence of large-scale endorheic basins and small-scale pits [18]. Endorheic basins are depressions in terrain caused by tectonic plates, and evaporation [24]. Pits are smaller and have no water at their lowest point [3]. Depression breaching is the most suitable pit-removal method for terrain generation. These algorithms create a downward slope to a central sink by decreasing elevations around the pit [18]. This enables a more realistic terrain.

Scott and Dodgson's [18] example-based multi-resolution terrain synthesis algorithm, known as terrain optimisation, is based on an approach by Kwatra et al. [12]. Although it is like previous work in terms of implementation, the algorithm does incorporate Generalised PatchMatch and the pit removal algorithms [18]. PatchMatch is a fast algorithm designed to evaluate dense approximate nearest neighbour correspondences between patches in two different regions [1, 2]. The terrain optimisation algorithm took 5 minutes to synthesise a 1000^2-pixel terrain [18]. This compared to two minutes for Gain et al.'s [9] implementation on the same machine. The addition of the pit-removal algorithm adds no significant cost to run time [18]. In terms of realism, Scott and Dodgson's [18] algorithm performs similarly to other methods [18]. Although it offers a valuable alternative, the terrain optimisation would not be favoured over the Gain et al. [9] implementation. However, the pit-removal algorithm is a helpful tool that could be added to any terrain generation algorithm to improve realism.

Many texture synthesis methods for terrain generation start out being used for images. Efros and Freeman's [7] image quilting process stitches together small patches of existing images to synthesise a new image [7]. This image quilting process is a simple and quick texture synthesis technique. In addition, they extend the process to texture transfer [7]. Texture transfer is achievable because the image quilting algorithm selects output patches based on local image information [7]. This could be utilised in terrain generation, where different terrains are transferred onto different bare-earth surfaces. This is a potential, future research and development avenue.

Finally, Xiao et al. [23] developed a method for generating diverse rock materials that aids in understanding and implementing its fracturing process [23]. It can produce parallel images of rock materials that can be utilised in feature replication [23]. The transfer of this technique to terrain generation could improve geomorphological realism by learning from fracturing processes found in differing rock materials.

7 DISCUSSION

The critical analysis of the evaluation procedures is vital to ensure a comprehensive and accurate comparison of the texture synthesis methods. The evaluation procedures will then be chosen, and the texture synthesis methods will be assessed. This will allow for significant and reliable analysis of the texture synthesis methods.

7.1 Analysis and Choices of Evaluation Procedures

Quantitative and qualitative measures were discussed, and each method assessed at least one of the following characteristics of terrain generation: scale, realism, user adaptability, and efficiency.

7.1.1 Scalability. The quantitative scale evaluation is an important evaluation procedure for this review as it assesses the viability of using a particular texture synthesis technique on the planetary scale. Scalability testing by quantitatively comparing various scales that the texture synthesis approaches use is the simplest and most effective way to compare scales across approaches.

7.1.2 Realism. There are a few evaluation procedures to assess realism, both quantitative and qualitative. Firstly, geomorphological testing is helpful in situations where adherence to geological natural phenomena is essential [10]. However, it also requires complex testing methods and does not always directly correlate to how people assess realism [10]. Perceived Terrain Realism Metrics (PTRM) is another option for evaluating realism. Although PTRM is incredibly accurate at determining the realism of terrain, the Geomorphons it consists of are localised to small areas [17]. Therefore, PTRM would not be helpful on a larger scale. Finally, user experiments could assess the realism of terrains via static renderings [10]. Users could compare the two terrains and decide which one is more realistic. However, the user is not always knowledgeable enough to make assessments of terrain and may make a mistake. To combat this, expert users can be called into the experiments [10]. User experiments allow for informative feedback from users on specific points of realism and areas of concern. Therefore, user experiments will be the approach used in this article to assess realism. It should be noted that this is a review, and the user experiments must be present in the publishing of texture synthesis methods to use this method.

7.1.3 User Adaptability. User adaptability will also be assessed through user experiments. This will be achieved through users using the texture synthesis approaches and reporting their thoughts on ease of adaptation and responsiveness to sketches. Once again, there is the issue of inexperienced users [10]. However, users who are experienced enough to use texture synthesis software should be able to access it. Therefore, user experiments remain an evaluation procedure for user adaptability.

7.1.4 Efficiency. Efficiency testing will cover time and space efficiency through quantitative comparison between texture synthesis approaches. This is a simple and fast evaluation procedure and will be effective.

To conclude, realism and adaptability will be evaluated by user experiments conducted by the various texture synthesis methods. For efficiency and scale evaluation, quantitative comparisons between the results of the methods will be assessed.

7.2 Analysis of Texture Synthesis Approaches

The various texture synthesis approaches will be analysed based on quantitative evaluations of scale and efficiency and user experiments for realism and user adaptability.

7.2.1 Scalability. The availability and quality of terrain input data, especially at a global scale, can significantly affect the possible scale given any texture synthesis approach. Applying the evaluation procedure of scalability testing mentioned in the previous subsection makes it possible to compare the different synthesis approaches. Gain et al.'s [9] algorithm was able to produce large-scale terrains of approximately 1000km across [9]. Zhou et al.'s [25] algorithm requires that the patch size be specified, which allows for the scale to be adjusted, but this comes at the loss of efficiency and resolution. The scale ranged from 1200m^2 to 4097m^2 [25]. Tasse et al.'s [20] adapted algorithm does not improve on the previous scalability of Zhou et al.'s [25] algorithm. The limited amount of memory on a GPU restricts the size of its terrains [20]. Scott and Dodgson's [18] terrain optimisation algorithm was tested on 30 x 30 km^2 terrains. This is not large enough for a planetary scale, but it is large enough to show that there is an opportunity to increase the scale.

7.2.2 Realism. Comparing user responses to the realism of the terrain produced by a specific synthesis method will allow for a complete analysis of texture synthesis approaches' ability to produce realistic terrains. User experiments conducted by Tasse et al. [20] showed that terrain generation was as realistic as real landscapes, however, there were instances of bad matching where users found clear artefacts [20]. Dachsbacher et al.'s [6] early algorithm struggled in terms of realism, partly due to the blurring technique used to remove notable artefacts that also removed much of the detail [6]. Scott and Dodgson's [18] pit-removal algorithm is useful for improving realism. Their terrain optimisation algorithm also underwent a user experiment with varying results [18]. In two of the terrains in which it was tested, the algorithm performed the best in terms of realism, and in another three, it performed the worst [18]. Efros and Freeman's [7] image quilting process could provide realistic terrain in the future. Xiao et al.'s [23] rock algorithm could improve geological realism when added to other texture synthesis terrain generation techniques. Finally, a user experiment conducted by Gain et al. [9] resulted in more users (54.6'%') believing their synthetic terrains were real than the actual real terrains [9]. This impressive result showcases the superior realism of this method. Upsampling also improves the realism in this method [9].

7.2.3 User Adaptability. Not every texture synthesis approach that has been discussed was developed with user adaptability in mind, and so, not all methods included user experiments for this. However, there are some impressive adaptability features in some of them that improve the user's ability to create personalised terrains. In Tasse et al.'s [20] adapted method, they improved control by using a post-synthesis process to constrain the elevation of terrains to match user sketches [20]. There is no ability to edit a terrain or control elevation after generation using the technique developed by Zhou et al. However, user sketching is used as the initial input [25]. There is limited user adaptability in the terrain optimisation algorithm by Scott and Dodgson [18]. Gain et al.'s [9] algorithm has a state-of-the-art interface that contains constraints, a copy-paste feature, and sketching and painting tools [9]. These all help in improving the user's ability to control the terrain being generated. A usability test was conducted, and users were able to effectively use all the interface tools [9].

7.2.4 Efficiency. Lastly, each texture synthesis method's time and space efficiency should be assessed. Zhou et al.'s [25] terrain synthesis takes approximately five to six minutes on a processor with 2GB of memory [25]. The adapted version by Tasse et al. [20] averaged around eighty seconds across the different terrain types. Due to differing hardware, the speedup is difficult to fully attribute to the adapted implementation. This was performed on a CPU with 3GB memory [20]. Scott and Dodgson's [18] terrain optimisation algorithm running on a CPU with 8 GB of memory took five minutes to synthesise a 1000 x 1000-pixel terrain [18]. Gain et al.'s [9] algorithm completed it in two minutes [18]. Running on 6GB of RAM and an NVIDIA GTX 680 with 2GB, Gain et al.'s [9] algorithm averaged 63ms and 151ms for resolutions 512^2 and 1024^2, respectively [9]. 42.67 bytes per pixel were used in the synthesis pyramid, and 84 bytes per pixel for the synthesis [9].

8 CONCLUSIONS

In summary, texture synthesis approaches, including patch-based, pixel-based, and recent approaches, can effectively complete terrain generation. These methods have varying scalability, realism, user adaptability, and efficiency levels.

A suite of evaluation procedures was selected to assess and compare the various texture synthesis methods. Quantitative comparisons were chosen for efficiency and scalability testing, while user experiments were chosen for realism and user adaptability.

The assessment of all the texture synthesis techniques discussed in this review was completed using the chosen and above-mentioned evaluation procedures. Scott and Dodgson's [18] terrain optimisation technique had notable results for realism, especially when combined with their pit-removal algorithm [18]. Efros and Freeman's [7] image quilting algorithm and Xiao et al.'s [23] rock algorithm present promising future research and development avenues.

Gain et al.'s [9] algorithm delivered the premiere texture synthesis technique in all the assessment aspects. Gain et al. [9] developed a state-of-the-art interface that allowed for highly successful user adaption. The efficiency of the algorithm was unmatched by any others in this review. Importantly to this review, the scalability of this algorithm provides promise for development in texture synthesis methods for planetary-scale terrain generation.

REFERENCES

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

[2] Connelly Barnes, Eli Shechtman, Dan B Goldman, and Adam Finkelstein. 2010. The generalized patchmatch correspondence algorithm. In *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Crete, Greece, September 5-11, 2010, Proceedings, Part III 11*. Springer, 29–43.

[3] Richard Barnes, Clarence Lehman, and David Mulla. 2014. Priority-flood: An optimal depression-filling and watershed-labeling algorithm for digital elevation models. *Computers & Geosciences* 62 (2014), 117–127. https://doi.org/10.1016/j.cageo.2013.04.024

[4] Yann Cortial, Adrien Peytavie, Éric Galin, and Éric Guérin. 2020. Real-time hyper-amplification of planets. *The Visual Computer* 36, 10 (2020), 2273–2284.

[5] Leandro Cruz, Luiz Velho, Eric Galin, Adrien Peytavie, and Eric Guérin. 2015. Patch-based Terrain Synthesis. In *International Conference on Computer Graphics Theory and Applications (Proceedings of the 10th International Conference on Computer Graphics Theory and Applications,)*. Berlin, France, 6 pages. https://doi.org/10.5220/0005360201890194

[6] Carsten Dachsbacher, Martin Meyer, and Marc Stamminger. 2005. Height-field synthesis by non-parametric sampling. *Vision, Modeling and Visualization 2005* (2005), 297–302.

[7] Alexei A. Efros and William T. Freeman. 2001. Image quilting for texture synthesis and transfer. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '01)*. Association for Computing Machinery, New York, NY, USA, 341–346. https://doi.org/10.1145/383259.383296

[8] Alexei A Efros and Thomas K Leung. 1999. Texture synthesis by non-parametric sampling. In *Proceedings of the seventh IEEE international conference on computer vision*, Vol. 2. IEEE, 1033–1038.

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

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

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

[12] Vivek Kwatra, Irfan Essa, Aaron Bobick, and Nipun Kwatra. 2005. Texture optimization for example-based synthesis. In *ACM SIGGRAPH 2005 Papers* (Los Angeles, California) *(SIGGRAPH '05)*. Association for Computing Machinery, New York, NY, USA, 795–802. https://doi.org/10.1145/1186822.1073263

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

[14] Sylvain Lefebvre and Hugues Hoppe. 2005. Parallel controllable texture synthesis. In *ACM SIGGRAPH 2005 Papers*. 777–786.

[15] Lawrence L O'Boyle. 2018. Agent Based Terrain Generator: Cruthú. Master's Thesis. Grand Valley State University, United States of America.

[16] Patrick Pérez, Michel Gangnet, and Andrew Blake. 2003. Poisson image editing. *ACM Trans. Graph.* 22, 3 (2003), 313–318. https://doi.org/10.1145/882262.882269

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

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

[19] Donald Shepard. 1968. A two-dimensional interpolation function for irregularly-spaced data. In *Proceedings of the 1968 23rd ACM national conference*. 517–524.

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

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

[22] Li-Yi Wei, Sylvain Lefebvre, Vivek Kwatra, and Greg Turk. 2009. State of the Art in Example-based Texture Synthesis. In *Eurographics 2009, State of the Art Report, EG-STAR*. Eurographics Association, Munich, Germany, 93–117. https://inria.hal.science/inria-00606853

[23] Huaiguang Xiao, Lei He, Xing Li, Qianbing Zhang, and Wengui Li. 2021. Texture synthesis: A novel method for generating digital models with heterogeneous diversity of rock materials and its CGM verification. *Computers and Geotechnics* 130 (2021), 103895. https://doi.org/10.1016/j.compgeo.2020.103895

[24] Vadim Yapiyev, Zhanay Sagintayev, Vassilis J. Inglezakis, Kanat Samarkhanov, and Anne Verhoef. 2017. Essentials of Endorheic Basins and Lakes: A Review in the Context of Current and Future Water Resource Management and Mitigation Activities in Central Asia. *Water* 9, 10 (2017). https://doi.org/10.3390/w9100798

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