Generating Realistic River Patterns with Space Colonization

Haoran Feng
University of Auckland
School of Computer Science
Auckland, New Zealand
hfen962@aucklanduni.ac.nz

Burkhard C. Wünsche
University of Auckland
School of Computer Science
Auckland, New Zealand
burkhard@cs.auckland.ac.nz

Alex Shaw
University of Auckland
School of Computer Science
Auckland, New Zealand
l.shaw@auckland.ac.nz

ABSTRACT

River generation is an integral part of realistic terrain generation, since rivers shape terrains and changes in terrain, e.g., due to tectonic movements can change the path of rivers. Fast existing terrain generation methods often result in non-realistic river patterns, whereas physically-realistic techniques, e.g., building on erosion models, are usually slow. In this paper we investigate whether the space colonization algorithm can be modified to generate realistic river patterns. We present several extensions of the space colonization algorithm and show with a user study with $n = 55$ participants that some variants of the algorithm are capable of generating river patterns that are indistinguishable from real river patterns. Although our technique can not generate all types of natural river patterns, our results suggest that it can prove useful for developing plausible 2D maps and potentially can form the basis for new terrain generation techniques.

Keywords

Space Colonization, river pattern, procedural generation, realism

1 INTRODUCTION

Rivers are essential for realistic terrains since rivers effect the shape of terrains via erosion, and vice versa terrain changes can influence the flow of rivers. Most countries contain rivers within their boundaries [Way19], and for some countries, such as New Zealand, rivers cover a large part of the land surface [NIWA18].

There are many existing realistic terrain generation algorithms [Arc11, CBCB+16, DP10, BF02]. However, they generally focus on the generation of landforms, instead of the rivers within the landforms. They are suitable for the generation of terrain in mountainous areas, but bodies of water are missing from the generation, which lowers the degree of realism in the terrain. Although the area of river generation is underexplored, there also exist some specialized river generation algorithms [GGGP+13, PDGC+19]. However, all algorithms listed above generate rivers or terrain by simulating erosion using physical simulation engines, hence the generation process is computationally intensive, and large-scale terrain generation may be impractical for commercial uses. An alternative method of terrain generation that produces more realistic river patterns is one that first generates realistic river patterns, then generates the terrain surrounding the patterns accordingly.

An efficient method of pattern generation is the space colonization algorithm [RLP07], which has been used originally for the generation of tree structures. In this paper, we explore how to extend the algorithm for river pattern generation. We aim to answer the following research question:

Can the space colonization algorithm be modified to efficiently generate realistic river patterns?

2 BACKGROUND

2.1 Types of River Patterns

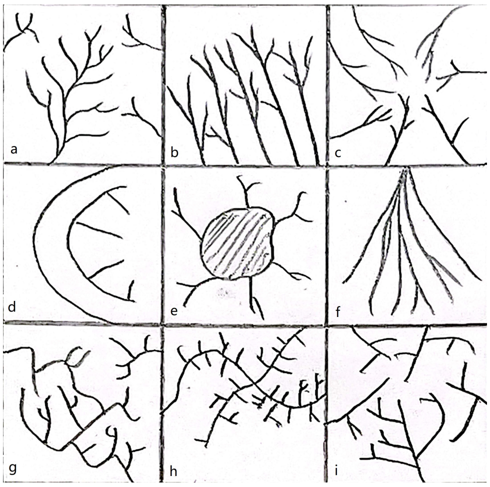


***Figure 2.1*: Different types of river patterns, adapted from Twidale's 2004 paper [Twi04].**

Realistic river generation is not a trivial task, as there are many factors influencing the formation of natural rivers, such as the slope of the terrain, the material of the terrain, and the rate of erosion. This can result in many types of river patterns [Twi04], as shown in *Figure 2.1*.

2.2 Space Colonization

Space colonization is a procedural content generation algorithm for generating realistic tree structures. It was created and published by Runions et al. in 2007. In their research paper, an overview of the algorithm is given, and the procedure of the algorithm is shown in detail. The paper also outlines the effects on the resultant tree structure if some parameters are changed. From the examples given in the paper, it is shown that the tree branches can be made denser or sparser, and also thicker or thinner, depending on the environment and the usage of the trees.

The procedure of the algorithm is outlined below, where black nodes are the nodes that are already in the tree structure, and blue nodes are the “attraction points" that control the generation and structure of the tree.

1. Start with a predetermined set of black nodes (i.e. start nodes) and a random set of blue nodes spread across the generation space.
2. Find the closest black node for each blue node under a distance and assign each of the blue nodes to the closest black node.
3. Determine for each black node the normalized directional vectors to its assigned blue nodes. If a black node is not assigned any blue nodes, then it skips step 4.
4. Sum all normalized directional vectors for each black node then normalize the resultant vector. Generate a new black node towards that direction a unit length away.
5. Determine if any blue nodes are within the vicinity of a black node, and remove the blue node if it is within a distance from a black node.
6. Return to step 2 until no new black nodes can be added.

3 RELATED WORK

Rivers are usually generated as by product of terrain generation, i.e., a terrain is generated first and rivers are formed using flow simulations. Only a few authors generate explicitly, i.e., that the rivers are formed first and influence the terrain shape.

Génevaux et al. devised a terrain generation model based on hydrology [GGGP+13]. The authors take a boundary for the terrain and initial nodes on the said boundary as input, expand the river structure from the nodes, modify parts of the structure based on the Rosgen river classifications [Ros94], and generate the 3D model according to the resultant structure. The model generates realistic terrain and naturalistic river patterns with reasonable efficiency. However, the node expansion and structure generation processes are convoluted and difficult to understand and implement, so there is room for simplifications or the usage of more straightforward algorithms for potential accelerations in terrain generation.

A more recent model developed by Peytavie et al. [PDGC+19] follows a similar pipeline as the model of Génevaux et al., as it also constructs river networks following the Rosgen river classifications. Then the model derives the multiple aspects of the terrain from the constructed patterns (e.g. drainage, slope), and it follows an amplification-combination procedure to simulate water flow along the rivers. The model results in an efficient method of generating terrain with realistic fluid flow animations, but the paper does not focus on the patterns of the generated rivers, so the realism of the generated rivers, hence also the generated terrain, may be further improved.

Van den Hurk et al. present a sketch-based terrain modelling techniques where rivers are manually inserted into the terrain using sketching [vHYW11].

4 DESIGN

As discussed in Subsection 2.2, the space colonization algorithm succeeds in generating realistic tree structures. However, due to the difference between tree structures and river structures, the algorithm does not generate satisfactorily realistic river patterns. For the output patterns to more closely resemble the structure of natural rivers [Twi04], several modifications to the algorithm can be made, and they are outlined and discussed in this subsection.

4.1 River Pattern Requirements

***Figure 4.1*: Real river patterns: Godavari River (left), Mississippi River (middle), and Yangtze River (right).**[1]

Figure 4.1 shows three examples of river maps from famous rivers across the world. Comparing the maps with the patterns generated from space colonization, the following dissimilarities may be observed (they are referred to as Dissimilarity 1, 2, and 3 respectively).

1. Curvature: The real river patterns tend to be curvier than the generated patterns.

---

[1] All patterns are listed in *Table 6.8*

2. Parallelism: The river branches in real river patterns are generally not parallel as opposed to the generated patterns.
3. Lateral branches: Short and perpendicular branches, known as lateral branches, from generated patterns are uncommon to occur in natural rivers.

Accounting for these differences between real river patterns and generated patterns, we devise a range of modifications to reduce the dissimilarities, such that the generated patterns also contain similar features as real river patterns, hence are considered more realistic.

4.2 Space Colonization Momentum

To reduce Dissimilarity 1 and introduce more curvature into the structures, the simplest approach is to divert the course of node generation. Our proposed Momentum variant of the space colonization algorithm slightly alters the direction of node generation using a measure of momentum, inspired by the concept of momentum in the field of physics.

In physics, if an object contains momentum in a direction, then it tends to continue moving toward that direction until an external force is acted on the object. This concept is adapted to generate nodes, however in a slightly different manner. If momentum is applied directly, where the direction of the next node tends to stay in the same direction as the previous node, then the resultant patterns become even less curvy than those generated from the original space colonization. Since we want to introduce more curvature into the patterns, we make a modification that acts the opposite as described above. The procedure of the Momentum variant is shown in *Figure 4.2 (left)*, and this modification takes place at step 4, as described in Subsection 2.2.

In *Figure 4.2 (left)*, (M1) shows a structure after step 4 of the space colonization procedure outlined above, where A, B, and C are nodes in the structure, v_1 is the direction vector of the generation of B from A, and v_2 is the direction vector of the generation of C from B. (M2) shows the introduction of a momentum vector, v_1, shown in red. The momentum vector belongs to B, as it is the direction of its generation, and hence it is applied to the next generation to alter its direction. For the new node stemming from B, if original space colonization is used, then C is generated, however, for the Momentum variant, the direction vector for the new node is subtracted by the momentum vector, multiplied by a multiplier k, as shown in (M3). The multiplier k can be interpreted as the strength coefficient of momentum, or essentially the size of effect the momentum vector (i.e. the previous direction vector) has on the new direction vector. In (M4), the subtraction results in the new direction vector $u = v_2 - kv_1$, which is then normalized to $\hat{u}$ in (M5), where the tip lies the generated node under the momentum variant, C'. Comparing the generation of C and C' as shown in (M1) and (M5), it is evident that the three nodes have produced a larger angle in the Momentum variant than the original approach, thus successfully introducing more curvature on a local scale.

It is noteworthy that the addition of momentum may affect different nodes differently. This difference becomes evident when there exists a large difference of angles between nodes, as shown in *Figure 4.2 (middle)*. (M6) shows a moderate angle between the three nodes, hence the shift in angles, θ_1, is moderate, whereas (M7) and (M8) show the effects of the shift in angles when the angle between the three nodes is smaller and larger, θ_2 and θ_3 respectively, when using the same strength coefficient k. From the comparisons, we learn that with a constant k, the size of the angle made by the three nodes is directly proportional to the shift in angles from the application of momentum.

Figure 4.3 (middle) shows an example pattern after applying momentum to space colonization. The difference between before and after the modification is evident, as the structure includes more curvature. The curvatures of some branches have a higher resemblance to the curvatures of natural rivers, but some extremely sharp turns and zig-zag patterns are also introduced in some branches as byproducts, which are rare in natural river patterns, hence lowering the overall degree of realism for an artificial river pattern.

4.3 Space Colonization Curl

To alleviate the problem of zig-zag patterns in the structure, we propose an alternative modification to the space colonization algorithm coined Curl. The Curl variant can be considered as a generalized version of Momentum with a randomization element, as the principles are similar, such as the modification occurring at the same step in space colonization, and it also alters the angle of the next node generated. The difference between Curl and Momentum is that Curl does not rely on the direction vector of the previous node to determine the shift in the next node, but it simply uses a random variable.

Essentially, before the node is generated, it is rotated at a random small angle either clockwise or anticlockwise, in which the direction is predetermined depending on the node. The direction of the rotation for the first node is preset, then the direction is changed after each set of c new nodes, where c is the length of each section of the pattern before bending toward the other direction. Then the angle of rotation is randomized to a value between 0 and θ_c radians.

The parameter θ_c can be interpreted as the degree of shift in angles allowed. The higher is θ_c, the greater shifts are allowed, and more curvature is introduced,

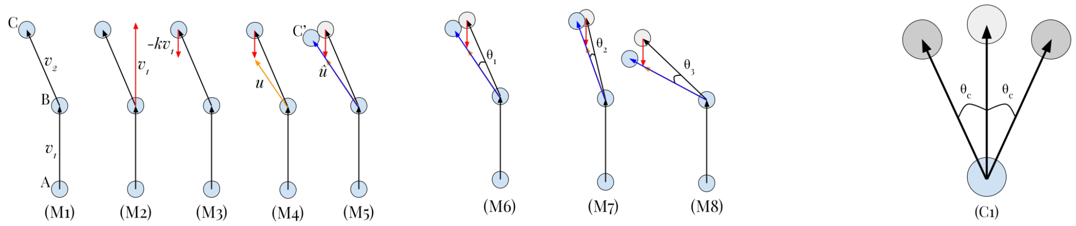


***Figure 4.2*: The effects of the proposed variants on the generated patterns. Left: The process of determining the position of C', i.e. the node generated instead of C using Momentum. Middle: The different effects of Momentum on generating nodes at different angles. Right: The possible range of nodes to be generated, where the light grey node signifies the next node if the original space colonization is applied, and the dark grey nodes signify the left and right extremes of the possible new nodes.**

and vice versa. Although choosing too high of a value for θ_c may result in unpredictable and chaotic patterns. The effects of Curl on the node generation are illustrated in *Figure 4.2 (right)*.

As shown in *Figure 4.3 (right)*, the patterns have become more unpredictable, and more curvature and a higher complexity have been introduced to the structure, the number of extremely sharp turns and zig-zag patterns has also been reduced when compared with the Momentum variant. Dissimilarity 2 has also been reduced, as the generated branches are not parallel with one another. However, the resultant patterns are still significantly unrealistic for river patterns, as there is too much detail at the ends of the structure, and Dissimilarity 3 still occurs in the patterns, as there exist short, perpendicular branches across long branches, which is unlikely to occur naturally.

4.4 Trimmed Space Colonization

To reduce Dissimilarity 3 and remove details at the ends of structures, a postprocessing step, namely trimming, is devised to remove these detailed endings and short branches, hence to further improve the degree of realism. Before the procedure of trimming is outlined, definitions of *children*, *parents*, and *depth* of nodes need to be introduced. For the purpose of trimming, we say that node Y is a child of node X, and X is the parent of Y, if Y is generated from the basis of X, and the *depth* of a node is defined recursively as the following.

- If a node has no children (i.e. is a leaf), then it has a depth of 0.
- If a node has one child of depth d, then it has a depth of $d+1$.
- If a node has n children of depths $\{d_0,...,d_{n-1}\}$, then it has a depth of $max(\{d_0,...,d_{n-1}\})+1$.

Now the procedure of trimming may be introduced. Trimming takes place after the generation of the entire tree structure, firstly the depths of all nodes in the structure are computed, then the nodes X that satisfy both *Inequality 1* and *Inequality 2* are removed from the structure, where *Inequality 1* and *Inequality 2* are outlined below, and t is a preset variable, named the trimming threshold.

$$depth(X) < t \tag{1}$$

$$depth(X) < depth(parent(X)) - 1 \tag{2}$$

We require both inequalities to hold for the removal of the node. The reasoning for the first inequality is intuitive, as all nodes that are “close" (defined by t) to endings are removed. The second inequality is not necessary if only one structure is generated, however, if multiple river patterns are generated simultaneously on the same surface, then the exclusion of the second inequality would cause gaps between the structures. To prevent this, we include the second inequality, such that only small, perpendicular branches and ending branches with excessive detail are removed from the structure, whereas regular endings are kept in place.

Since trimming is a postprocessing step, hence the modification can be applied in parallel with each of the Momentum and Curl modifications. This results in a total of six variants of space colonization, i.e., original, momentum, curl, trimmed original, trimmed momentum, and trimmed curl. Some patterns from the trimmed variants are shown in *Figure 4.4*. Comparing the patterns, it is evident that the amounts of detail in the structures have decreased from the “untrimmed" variants, as there exist fewer shorter branches in the patterns.

Referring back to the desired qualities outlined for generated patterns, the degree of curvature has increased in the patterns, and some parameters (k, c, θ_c) have been introduced to control the degree of curvature; as a result of the enhanced curvature, the branches in generated patterns are no longer parallel, and the patterns

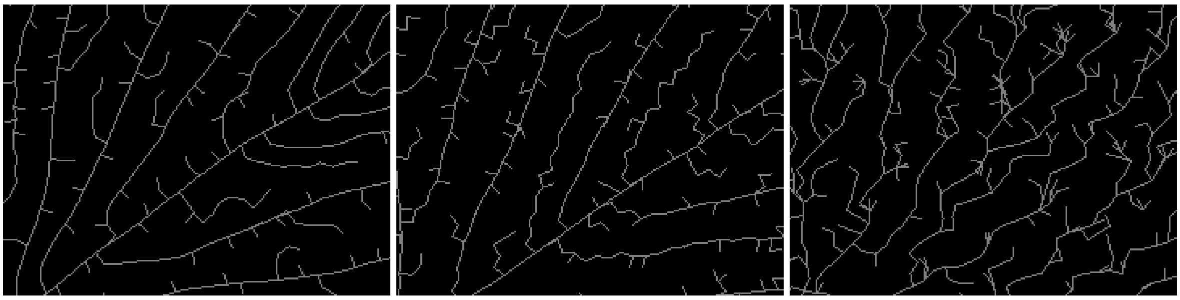


***Figure 4.3*: Example patterns generated from different variants of space colonization: Original (left), Momentum (middle), Curl (bottom). As shown in the patterns, Momentum introduces more curvature into the patterns but also introduces unrealistic zig-zag patterns, and Curl also introduces more curvature and adds more complexity to the patterns.**

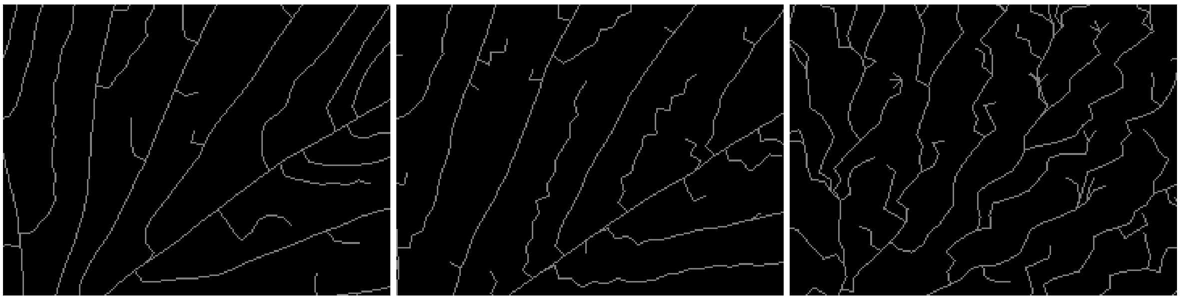


***Figure 4.4*: Example patterns generated from different variants of trimmed space colonization: Trimmed Original (left), Trimmed Momentum (middle), Trimmed Curl (right). Compared to the patterns shown in *Figure 4.3*, short branches at the ends of patterns are pruned after trimming, and the amount of detail in the patterns reduces.**

are also less predictable; the option to remove lateral branches has also been introduced using trimming. In summary, the dissimilarities outlined in Subsection 4.1 have been reduced. To further examine the degree of realism each space colonization variant provides when generating river patterns, a user study is conducted, and further details are discussed in Subsection 6.1.

5 IMPLEMENTATION

To produce results from the generation algorithms outlined above, a system incorporating the algorithms was developed using C++14 and OpenGL3.3. The IDE (Integrated Development Environment) used for system development is Microsoft Visual Studio Community 2022. GitHub was also used for version control of the written code. The system was built using a 64-bit Windows 10 Home operating system, on a Dell Inspiron 15-3568 with 8GB RAM, and an Intel(R) Core(TM) i5-7200U CPU.

During development, several headers in the C++ Standard Library were used, which are chrono, cmath, iostream, random, and vector. No external libraries were used for C++. Some commonly used external libraries for OpenGL were also used, which are Glad (OpenGL Loading Library), GLFW (Graphics Library Framework), GLM (OpenGL Mathematics), and KHR (Khronos).

Some algorithms were implemented and tested with Python before being implemented with C++, for ease of development and debugging. The version of Python used is Python 3.7.2. Some modules and libraries were used during the process, which are Matplotlib, NumPy, OpenCV, random, and time.

In the evaluation process (further discussed in Section 5), some image processing was conducted. The image processing program for raw images was written with Python 3.7.2 using the IDE Visual Studio Code, with the assistance of the libraries Matplotlib, NumPy, OpenCV, and Skimage. Software Microsoft Paint and Microsoft Paint 3D were also used for the viewing and cropping of the images.

Lastly, the processing of raw data received from responses to the user study was carried out using Microsoft Excel.

6 EVALUATION

To answer our research question whether the space colonization algorithm can be used to generate realistic river patterns, we conducted a user study comparing the realism of real river patterns with river patterns generated using the algorithms outlined in Section 4.

6.1 Methodology

To evaluate the realism of river patterns generated from algorithms outlined in Subsection 4.1, a user study was carried out in the form of an online questionnaire, and it was propagated through private messaging among friends and classmates. The study was anonymous and participation was voluntary.

The questionnaire begins with a list of background questions to determine the demographic groups of the participant. The information on the gender, age, and visual conditions (if any) of the participants is collected, as well as their level of experience in the fields of geography, map-reading, and visual arts.

The demographic information of the participants is collected, in confidence, for future reference, such that any potential hidden relationships between demographic variables and observed variables may be discovered if relevant.

The questionnaire then asks the participant to rate the degree of realism of 30 river patterns on a 7-point Likert scale. Each river pattern is presented with an image, where the areas within a river are labeled white, and the areas outside are labeled black, some examples are shown in *Figure 6.1*.

***Figure 6.1*: Examples of patterns presented in the user study questionnaire.**

There exists a mix of real and generated river patterns within the 30 river patterns shown to users, 15 of which are collected from real, natural rivers, and the remainder 15 are patterns generated from variants of space colonization. Out of the 15 generated patterns, 3 are generated from the original space colonization algorithm, 1 from Momentum, 1 from Curl, 4 from Trimmed original, 3 from Trimmed Momentum, and 3 from Trimmed Curl. All patterns used in the user study are shown in *Table 6.8*.

The information of whether a pattern is from a real river or generated is withheld from the participant, hence blinding is used to reduce bias and ensure that participants do not rate patterns from different origins differently.

To ensure that only the images presented only contain information about the patterns of the real/generated rivers, some preprocessing is necessary. Online images of river maps often contain information irrelevant to patterns, such as labels and colors, and it needs to be removed before being presented to the participants, hence a 4-step preprocessing procedure is applied to all river maps retrieved through online sources. The procedure is outlined below.

1. Color extraction: Choose a range of RGB colors that include all pixels relevant to the river patterns.
2. Manual label removal and resizing: Use image editing software to erase any remaining text, and resize the image to an appropriate scale for cropping.
3. Skeletonization: Apply a skeletonization algorithm, such as the Zhang-Suen Thinning algorithm [ZS84], such that the information on the widths of rivers is removed.
4. Snippet extraction: Crop the image to a preset, fixed size. (The images used in this user study are 100 pixels high and 100 pixels wide.)

To ensure that the generated patterns are representative, a range of parameters are used across generations of the same variant, and the snippets of patterns are cropped using a randomly generated set of image coordinates. Furthermore, to keep the colors and scales in the images consistent between real and generated patterns, steps 3 and 4 of the preprocessing procedure are also applied to the generated patterns.

One potential problem with this questionnaire is that each participant has a different view on the “realism" of river patterns. Some participants may generally rate patterns higher on the Likert scale than others, and vice versa, this may skew the overall distribution of the ratings. But since we are comparing the distributions of two subgroups of data, each subgroup of data is likely skewed to a similar degree, hence this does not pose a problem in the evaluation process.

6.2 Results

Out of the 52 participants in this user study, there are 34 males, 16 females, and 2 preferred to not specify their gender. The majority of participants (49 participants) are between 18 and 25 years of age, with a mean of 19.94 years and a standard deviation of 1.30 years, and there are 3 participants outside of this range, who are 45, 52, and 52 years of age. 7 participants reported having myopia (short-sightedness). The results of the other background questions are shown in *Table 6.2*.

***Table 6.2*: Distribution of experience levels on different skills of participants in the user study.**

Skill	None	Little	Some	Much
Geography	34	8	10	0
Map-reading	7	22	20	0
Visual Arts	22	26	3	1

Regarding the questions for rating the realism of the river patterns, the average realism scores[2] across multiple meaningful groups of patterns are calculated for each participant, such as the average score for all real patterns given by a participant. This produces a list of average scores for each participant. Then the mean and median scores are calculated across all participants, and the results are shown in *Table 6.3-6.5*.

***Table 6.3*: The mean and median realism score between real patterns and generated patterns across all participants.**

Pattern Type	Mean	Median
Real	4.41	4.47
Generated	3.94	4.03

***Table 6.4*: The mean and median realism score between untrimmed generated patterns and trimmed generated patterns across all participants.**

Pattern Type	Mean	Median
Untrimmed	3.53	3.60
Trimmed	4.15	4.20

***Table 6.5*: The mean and median realism score between all variants of space colonization across all participants.**

Variant	Mean	Median
Original	3.31	3.00
Momentum	4.67	5.00
Curl	3.02	3.00
Trimmed Original	3.85	3.75
Trimmed Momentum	4.19	4.00
Trimmed Curl	4.53	4.67

From the results shown in *Table 6.3-6.5*, we observe that for the participants in the user study, on average, real patterns achieved a higher realism score than generated patterns, trimmed generated patterns achieved a higher realism score than untrimmed patterns, and out of all space colonization variants, momentum achieved the highest realism score, with trimmed curl achieving the second highest, and both achieved a higher realism score than real patterns.

6.3 Significance Testing

The research question asks whether space colonization and its variants can generate realistic river patterns, we can reframe the question to whether space colonization and its variants can generate patterns that are indistinguishable from real river patterns, and that question can be answered from the data collected from the user study, by comparing the realism scores between groups of data. This can be achieved by using a series of significance tests.

There are several options of significance tests for Likert scale data (i.e. ordinal categorical data), some commonly used ones are the Mann-Whitney U test, Wilcoxon Signed-Rank test, and paired t-test. Different tests differ in requirements within the data and their strictness. Given the nature of our dataset, the Mann-Whitney U test is unsuitable, since two groups of compared data are collected from the same set of participants, hence are dependent [Gle22]. Thus Wilcoxon Signed-Rank test and paired t-test are used for significance testing. Although some commonly used requirements for the paired t-test are that each group of data needs to be normally distributed, and the data needs to be numerical, which does not comply with the collected Likert scale data. However, there has been research showing that the paired t-test is robust even on skewed data and ordinal categorical data [Nor10]. Hence despite the requirements stated above, the paired t-test is used on the collected data.

For significance testing, each variant or each collection of variants is treated as a group, and all groups, except for the Real group, are compared to the Real group for any significant difference using both the two-sided Wilcoxon Signed-Rank test and the two-sided paired t-test, with 95% confidence. The null hypothesis (H_0) and alternative hypothesis (H_1) are listed below, and the results of the tests are shown in *Table 6.6-6.7*.

H_0: There does not exist a significant difference between the groups, i.e. the patterns in the two groups are indistinguishable and the generated patterns are considered realistic.

H_1: There exists a significant difference between the groups, i.e. the patterns in the two groups are distinguishable and the generated patterns are considered unrealistic.

***Table 6.6*: The p-values of the two-sided Wilcoxon Signed-Rank tests (95% confidence) comparing the Real group with other groups.**

Group	p-value
Generated	0.000
Untrimmed	0.000
Trimmed	0.003
Original	0.000
Momentum	**0.087**
Curl	0.000
Trimmed Original	0.000
Trimmed Momentum	**0.073**
Trimmed Curl	**0.238**

---

[2] A "realism score" is measured on a 7-point Likert scale, where 1 indicates highly unrealistic, and 7 indicates highly realistic.

***Table 6.7*: The p-values of the two-sided paired t-tests (95% confidence) comparing the Real group with other groups.**

Group	p-value
Generated	0.000
Untrimmed	0.000
Trimmed	0.002
Original	0.000
Momentum	**0.186**
Curl	0.000
Trimmed Original	0.000
Trimmed Momentum	0.044
Trimmed Curl	**0.316**

The common usage of significance tests is to reject the null hypothesis, and the notion of accepting the null hypothesis is discouraged. However, there has been research showing that it is adequate to accept the null hypothesis for certain scenarios [Fri95]. We aim to investigate whether a group of patterns are realistic, hence the goal of the significance tests is to accept the null hypothesis. If the p-value of a significance test is greater than 0.05, then the significance test is considered successful, and vice versa.

6.4 Discussion

From the results of the significance tests, we may conclude that the Trimmed Curl variant produces patterns that are indistinguishable from real river patterns, as the null hypothesis is accepted in both scenarios, also notably with relatively high p-values.

The variants Original, Curl, and Trimmed Original are deemed unsuitable for river pattern generation, as the null hypothesis is rejected with minuscule p-values.

The Trimmed Momentum variant passes one of the two significance tests, which implies that it is on the borderline to being suitable for river pattern generation, but since Trimmed Curl outperforms Trimmed Momentum significantly, the former is preferred over the latter.

Surprisingly, the null hypothesis for the Momentum variant is accepted, as the variant is not expected to perform well. However, since there is only one pattern out of the 30 that is from the Momentum variant, we cannot conclude that it produces convincing river patterns in general, as the one pattern used does not nearly cover the large range of possible patterns across different parameters. Whereas there are three patterns used for Trimmed Curl across ranges of parameters, covering a large range of possible patterns. At this stage, we cannot conclude whether the Momentum variant produces realistic river patterns, and another user study may need to be conducted to reach a more conclusive outcome.

For larger groups (Generated, Untrimmed, Trimmed), the null hypotheses are all rejected. This is to be expected since if at least one algorithm in the collection of algorithms produces unrealistic patterns, then the collections of patterns cannot be realistic altogether.

From the results of the user study, we can conclude that the trimmed curl variant of space colonization generates local patterns that are indistinguishable from real river patterns. This means that it can be used to reliably generate realistic local river patterns. Secondary choices are the momentum and trimmed momentum variants. However, since no advantages have been discovered using the secondary variants, hence the trimmed curl variant is the optimal approach for generating realistic river patterns.

6.5 Limitations

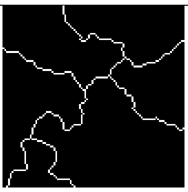
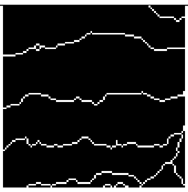
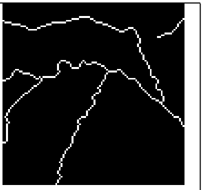
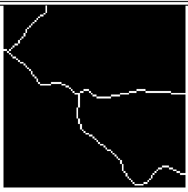
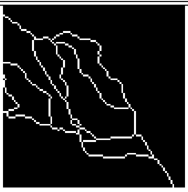
There are several limitations to the evaluation of this research, some with potentially significant impact on the conclusion are outlined and discussed in this subsection.

There is one major limitation regarding the user study. The patterns shown to the participants only capture local snippets of the structures, whereas, for larger areas, or global snippets with a wider view of the structures, the degree of realism that algorithms deliver may differ. However, for the purpose of this research project, local snippets are sufficient to capture the essence of the patterns and provide meaningful results.

Another limitation of the user study, as discussed briefly above, is the limited number of patterns from each variant. Considering that six variants were elevated simultaneously, the number of patterns for each variant needed to be controlled, as participants may lose interest, hence reducing the quality of responses, if a multitude of patterns is shown. To resolve this, we assigned more patterns to variants that are expected to perform well, and fewer to those not, so that there is more supporting evidence for a well-performing variant if it receives positive results. However, this results in the lack of evidence for the variants that are not expected to perform well, such as the Momentum variant as seen in the results, and another user study may need to be conducted, incorporating more patterns from that variant, to more conclusively evaluate its performance.

The evaluation of the river pattern generation component of the research has proven to be quite successful. However, the generated river patterns are limited to be local, i.e. on a small scale, whereas for more general uses of terrain generation, realistic global patterns may be more beneficial. A possible extension of the research is to conduct another user study comparing real patterns and generated patterns on a larger scale, such that the

***Table 6.8*: 30 patterns shown (in order) in the user study and their origins.**

					
Pattern ID	1	2	3	4	5
Group(s)	G0	G0	G1, G3, G9	G1, G3, G8	G0
Origin	Ohio River	Orinoco River	Trimmed Curl	Trimmed Momentum	Indus River
					
Pattern ID	6	7	8	9	10
Group(s)	G1, G3, G9	G1, G3, G7	G1, G3, G8	G0	G1, G3, G7
Origin	Trimmed Curl	Trimmed Original	Trimmed Momentum	Congo River	Trimmed Original
					
Pattern ID	11	12	13	14	15
Group(s)	G1, G2, G6	G1, G3, G8	G1, G2, G5	G0	G0
Origin	Curl	Trimmed Momentum	Momentum	Yangtze River	Volga River
					
Pattern ID	16	17	18	19	20
Group(s)	G1, G3, G7	G0	G1, G3, G9	G0	G0
Origin	Trimmed Original	Amur River	Trimmed Curl	Irrawaddy River	Amazon River
					
Pattern ID	21	22	23	24	25
Group(s)	G0	G0	G0	G0	G1, G2, G4
Origin	Thames River	Murray River	Mississippi River	Tigris River	Original
					
Pattern ID	26	27	28	29	30
Group(s)	G0	G1, G2, G4	G1, G3, G7	G1, G2, G4	G0
Origin	Zambezi River	Original	Trimmed Original	Original	Godavari River

performance of the devised algorithms for global patterns may also be determined. Another possible extension is to extend the proposed methods, such that they are also capable of generating a wider range of river patterns, as described in Subsection 2.1.

7 CONCLUSION AND FUTURE WORK

We explored the usefulness of the space colonization algorithm [RLP07] for river pattern generation, and we devised several variants of the algorithm to achieve higher measures of realism for this purpose. We showed that one of the variants, Trimmed Curl, can generate patterns that are indistinguishable from real patterns extracted from natural rivers. This has proven to be novel, as the original space colonization algorithm did not return promising results from the user study, similarly for several other innovated variants.

The invention of the Trimmed Curl variant broadened the usage of space colonization to be not limited to tree generation, but also river generation. It also serves as a more computationally inexpensive alternative to existing river generation algorithms, such that rivers may be artificially generated on a larger scale, at a faster rate.

In future work we would like to explore how synthesised river patterns can be used for map and terrain generation. Since terrains in game engines are usually represented as greyscale maps, a possible solution might be an exemplar-based texture synthesis technique finding for river sections patches of matching terrain information and then completing the greyscale texture using a texture infilling method [NWDL14].

8 REFERENCES

[Arc11] Archer, T. (2011). Procedurally generating terrain. In *44th annual midwest instruction and computing symposium, Duluth (pp. 378–393).*

[BF02] Benes, B., & Forsbach, R. (2002). Visual Simulation of Hydraulic Erosion. Journal of WSCG, 10(1), 79–86.

[CBCB+16] Cordonnier, G., Braun, J., Cani, M.-P., Benes, B., Galin, E., Peytavie, A., & GuErin, E. (2016). Large Scale Terrain Generation from Tectonic Uplift and Fluvial Erosion. *Computer Graphics Forum, 35(2), 165–175.* doi:10.1111/cgf.12820

[DP10] Doran, J., & Parberry, I. (2010). Controlled Procedural Terrain Generation Using Software Agents. *IEEE Transactions on Computational Intelligence and AI in Games, 2(2), 111-119.* doi:10.1109/tciaig.2010.2049020

[Fri95] Frick, R. W. (1995). Accepting the null hypothesis. Memory & Cognition, 23(1), 132–138. doi:10.3758/bf03210562

[GGGP+13] Génevaux, J., D., Galin, É., Guérin, E., Peytavie, A., & Beneŝ, B. (2013). Terrain generation using procedural models based on hydrology. *ACM Transactions on Graphics, 32(4), 1.* doi:10.1145/2461912.2461996

[Gle22] Stephanie Glen. (2022). *Mann Whitney U Test: Definition, How to Run in SPSS.* From StatisticsHowTo.com: Elementary Statistics for the rest of us!. Accessed 9 Nov 2022. https://www.statisticshowto.com/mann-whitney-u-test

[NIWA18] National Institute of Water and Atmospheric Research. (2018). *Map of New Zealand Rivers | NIWA.* Accessed 2 October 2022. <https://niwa.co.nz/freshwater/nzffd/NIWA-fish-atlas/map-of-NZ-rivers>.

[Nor10] Norman, G. (2010). Likert scales, levels of measurement and the “laws” of statistics. Advances in Health Sciences Education, 15(5), 625–632. doi:10.1007/s10459-010-9222-y

[NWDL14] Nguyen, H. M., Wünsche, B. C., Delmas, P., Lutteroth, C. (2014). Poisson Blended Exemplar-based Texture Completion. Proceedings of the Thirty-Seventh Australasian Computer Science Conference. 2014, 147, 99–104.

[PDGC+19] Peytavie, A., Dupont, T., GuÃ©rin, E., Cortial, Y., Benes, B., Gain, J., & Galin, E. (2019). Procedural Riverscapes. *Computer Graphics Forum, 38(7), 35–46.* doi:10.1111/cgf.13814

[RLP07] Runions, Adam & Lane, Brendan & Prusinkiewicz, Przemyslaw. (2007). Modeling Trees with a Space Colonization Algorithm. *Natural Phenomena. 63–70.*

[Ros94] Rosgen, D. L. (1994). A classification of natural rivers. CATENA, 22(3), 169–199. doi:10.1016/0341-8162(94)90001-9

[Twi04] C.R. Twidale (2004). River patterns and their meaning. *Earth-Science Reviews, Volume 67, Issues 3-4, Pages 159–218.* ISSN 0012-8252.

[vHYW11] van den Hurk, S., Yuen, W., & Wünsche, B. (2011) Real-time Terrain Rendering with Incremental Loading for Interactive Terrain Modelling. Proc. of GRAPP, 181–186.

[Way19] Wayback Machine/Central Intelligence Agency. (2019). *Field Listing: Waterways | The World Factbook - CIA.* https://web.archive.org/web/20190111040953 /https://www.cia.gov/library/publications/the-world-factbook/fields/386.html.

[ZS84] Zhang, T. Y., & Suen, C. Y. (1984). A fast parallel algorithm for thinning digital patterns. Communications of the ACM, 27(3), 236–239. doi:10.1145/357994.358023