Generating realistic terrains using noise functions and erosion algorithms

Matthew Xuereb
Institute of Information & Communication Technology
Malta College or Arts, Science & Technology
Corradino Hill
Paola PLA 9032
matthew.xuereb.c10384@mcast.edu.mt

***Abstract*—The study set out to procedurally generate terrains with a realistic aesthetic. Noise functions such as Perlin and Cellular noise were used to construct the terrains, along with Hydraulic and Thermal erosion being simulated for terrain deformation to achieve the desired mention aesthetic. At the end of the terrain generation, texturing was carried with the help of a splat-map, used as a mask to indicate the 3 textures used the strength to be applied on the terrain at their given pixel coordinate. Lastly, vegetation was laid out by the use of a survival algorithm which simulates the natural competition of plant life to grow and survive. A survey was utilised to gather and analysis the opinions of the participants with the results leaving more to be desired. Along with the survey the timings of the terrain generation were logged across multiple devices shown a clear difference when more powerful hardware is at play, along with the affects of GPU bottle-necking.**

***Index Terms*—Procedural, Terrains, Simulations, Fluids, 3D Texturing**

I. INTRODUCTION

In the field of computer graphics virtual terrains could either be created manually by an artist or procedurally generated through code. Procedural terrain generation is often used as a substitute for an artist work, as manually creating the terrain takes time and money which the studio may not afford to dedicate.

However, procedural terrain generation alone does not create terrains that are visually appealing to the human eye as they are often repetitive and uninteresting. If the goal of the studio is to generate a terrain with a realistic aesthetic, simply using a noise function will not cut it as real life terrains are mainly shape through the long process of erosion, further explained in III.

Along with erosion, vegetation cover also played a role, it is how the terrain is covered with vegetation such as grasslands and forests, which is the process of an ecosystem growing in their evolved habitat and competing for survival.

The aim of the paper is the development of a prototype capable of generating procedural terrains with a realistic aesthetic. Multiple terrains will be generated and presented in a survey where the participants are asked about their opinions of the results, removing any bias that would have otherwise been present. The approach of the survey is a mixed of qualitative and quantitative, qualitative in a sense that the opinions of the participants will be collected and quantitative due to the participants guesses also being collected.

The gathered results will then be analysed in Section IV on how the participants performed in the survey and how they shaped the opinion of the prototype at the end.

Along with the survey the duration logs of the individual terrain generation will also be analysed in Section IV across multiple devices and their different timings will be studied.

II. REVIEW OF RESEARCH METHODOLOGY

In computer graphics noise is often used as a quick trick to generate an effect whether it is for a visual effect or a 2D image. Widely used in the process of terrain generation, especially in game development.

For this paper the FastNoiseLite library [] was used as it provides all the necessary functionality to generate the required noise maps needed to draw the terrain. While there are many type of noise functions for this paper only 2 noise functions were used, those being Perlin and Cellular noise.

Real world terrain is mainly shaped by the process of erosion. The two main processes are hydraulic erosion which is the interaction of water moving sediment around by breaking up the ground and depositing it else where, and thermal erosion which is the weathering of the terrain due to constant exposure to sun rays.

Before applying hydraulic erosion the water flow approach had to be decided. There were two types approaches that were considered, the first was to simulate water as a particle [T15] and the second was to simulate water as a field [J03]. The advantage of using the particle approach is that it replicates water flow in a realistic manner as it simulates the interaction of rain droplets on the terrain. However, the disadvantage of this method is that it has no data on exiting water sources like rivers and lakes. However, the field approach does solve this issue as it stores all its values in an array and calculates the water flow for each cell in the grid by the difference in the neighbouring fluid values.

By implementing the fluid flow method employed by [B11; MB11; SJ08] a suitably result is achieved at a low performance cost by sacrificing physical accuracy as the methods developed by Stam [J03] and [B11; MB11; SJ08] were not developed to be accurate, instead developed to be used in real time applications.

With the fluid flow simulation implemented, the erosion is next. In [SJ08] three methods were utilised to simulate erosion on the terrain, these being force, dissolution, and material slippage. Force and dissolution were handled by the Hydraulic erosion, while material slippage was handled by Thermal erosion [B11].

With the terrain generated and eroded, next comes the texturing to add detail to an otherwise blank looking terrain. For the purpose of this prototype only height, slope, and the level of water will be considered. The use of Splat-maps were utilised in both creation of the video game Battlefield 2 [J07] and in Mediterranean environments [NZ16]. A splat-map is an image that for each channel a texture is assigned to it, with each channel having either the height, slope, or water level dedicated to it.

When the splat-map is generated and is set to a plane, the texture can be displayed properly. But, when the splat-map is applied to a terrain which in our implementation is a plane with its vertices raised based on a height value, distortion is introduced and increases the higher a vertex is raised. This problem was tackled in [WR20] with the use of Triplanar mapping. Instead of mapping the texture in a single axis, triplanar mapping does what the name describes and maps the texture in all three axis. All three maps are then blended and the result is displayed on the terrain, removing the previous mentioned distortion.

In [NZ16] a survival algorithm was developed to simulate the growth of vegetation. The algorithm described by [B02] is what is needed for this task. It consists of a plant growing and competing for space and resources, the plant will die if it cannot achieve it minimum amount of resources or is colliding with a bigger plant.

III. Reflection on the chosen Methodology

The goal of this research paper was the procedural generation of terrains with a realistic appeal. To remove bias from the research a survey was utilised to ask participants which terrain they thaught was real from 2 images of one real terrain and the other generated. Along with their guesses their opinions was also collected.

Along with the survey, the timings of both the erosion and vegetation simulations were also obtained for both real and generated terrains. The process took for data gathering was to generate the terrain 5 times and logged, along with the logs the average, median and difference between the maximum and minimum were also gathered at the end of the process.

Any data collected from the survey was kept anonymous with the only data collected was the participant experience in procedural terrain generation, their guess of which terrain they think was real, and their opinion why.

The mentioned data collection was carried out on 2 devices, a desktop with an AMD r7 3700x and NVIDIA RTX 3060 Ti, as well as a laptop with an Intel i7 7700HQ and NVIDIA GTX 1050 Ti. Multiple devices were used to gather the timings from different hardware and analyse the effects of more powerful hardware in the terms of computer simulations.

Before generating the terrain, the real world terrain had to be extracted, extracted from the website USGS a US government agency national map down-loader [].

For this paper the terrains rather real or generated, were stored as a 2D 16-bit grayscale image with a terrain density of 1 meter. By default USGS downloads the DEM as a GEOTiff format so they had to converted to a 16-bit grayscale PNG. For the conversion the free and open source GDAL tool was utilised along with Anaconda CLI which was needed to install and utilise GDAL [;].

The terrain generation process consisted of 4 steps being noise generation, erosion, texturing, and lastly vegetation spawning.

The FastNoiseLite Library was used [], as it provided all the needed functionality for the prototype, saving time in development. The two noise functions used in the generation of the terrains were Perlin and Cellular. Perlin noise was used to generate the main features of the terrain as it results natural forming mountains and valleys. Cellular noise was combined with Perlin noise to make the end result less repetitive and more interesting.

The process of erosion was simulated with use of Hydraulics and Thermal. Hydraulic erosion simulated the process of water breaking apart the terrain and depositing elsewhere. The decision to either perform erosion or deposition was decided by the amount of suspended sediment found within the water. If the amount of suspended sediment exceeds a threshold it would have been deposited onto the terrain, otherwise the terrain was erode further adding to the suspended sediment. While the thermal erosion algorithm distributes the sediment based on its own height difference compared to its own neighbors 1.

Fig. 1. $current - neighbor$

The vegetation simulation was the process of the individual trees competing with their environment for the chance to further grow with the end goal to spawn new saplings before dying. The simulation naturally favors trees of the same specie to gather around each other due to the trees spawning their sapling nearby. To ensure trees do not grow were they should not, limits were introduced in the simulation. The first that a tree cannot grow on very steep hills, while the second was that a tree could not survive under a certain amount under water.

A splat-map was utilised as a mask to tell each texture the strength to be applied on the terrain at any point and was split in three channels based on the slope of the terrain and water level. Triplanar texturing was used to texture the terrains without introducing any texture distortion. Three textures were used in the process of texturing with each being assigned a colour channel in the splat-map, with each texture being applied separately based on their value in the splat-map and later blended together.

For the procedural terrain generation all the 4 mentioned steps are executed in order, but when generating the terrain from a DEM the noise generation process was skipped, as the height values were read from an imported height-map. A when erosion was also not needed instead water was placed on the terrain and the water flow simulation was carried out without any erosion present.

IV. Results, analysis and discussion

Before any of the participants could begin the survey they had to agree to a consent form, agreeing that their participation will be used in a research paper and that their responses would be kept anonymous.

As of 23th May 2023, a total of 9 people participated in the research survey, were the participants were asked to compare real world terrains with procedural generated terrains. With a total participation numbered at 9 people, with only 4 people completing the entire survey, the survey would probably be considered as a small survey.

The first question of the survey asked the participant what experience they had in procedural terrain generation, with the experience of participates leaning towards having some experience in procedural terrain generation, with 6 people having past experience and the reaming 2 having no past experience, with 1 that never heard what procedural terrain generation was.

For the remaining questions the participant was show 2 terrains and they were asked to pick which terrain they think was real, along with giving their opinion why. This question was asked 5 times throughout the survey, with each question different terrains were shown. In total the terrain comparison question was answered 27 times, with 9 guesses being correct.

Looking at the responses, some common patterns that was noticed were aspects that were not related to the actual terrains themselves but on how they were presented. One of the noticed patterns was texture repetition. With texturing being applied as a grid of textures and no anti-repetition techniques being applied, repetition was to be expected regardless if the terrain was real or generated. Another pattern was that areas that were far away from the camera would not display shadows were tree shades would normally be visible, this was due to a renderer setting to not render shadows beyond a certain distance to preserve performance.

Most of the answers came to the conclusion based on characters of the terrains mostly through how the hills appeared or how the rivers presented themselves. Some mentioned that one terrain had a better looking hill or that the river look more natural or had islets.

With a correct guess of 33 percent it showed that the participants had a hard time guessing which terrain was real or generated. Although, unintended aspects such as graphics also played a role in participants choosing the wrong answer.

The duration statistics of the terrain generation of both real and procedural were logged across several devices in order to get the comparison between different hardware generations. The devices used were a desktop with an AMD r7 3700x with a Nvidia RTX 3060 Ti and a laptop with an Intel i7 7700HQ with a Nvidia GTX 1050 Ti. The terrains were generated 5 times with the average, median, and difference between the maximum and minimum timings being logged.

Although for both the real and generated there were 3 terrains each, the timings were all very close to each other as the same amount of work should have been done to all the terrains equally.

Starting from the desktop device the timings from the erosion processes from all generations took around 35 to 40 seconds to complete, with the longest process taking 40.9 seconds and the shortest being 35.46 seconds, giving the difference of 5.44 seconds. With an average of 36.58 seconds and median of 35.65 seconds with the simulations leaned more towards the minimum. Further investigating of the erosion simulations graph leads to the conclusion of the one outlier of the highest duration of 40.9 seconds, getting the difference between the highest and the second highest value of 38.82 returns 3.36 seconds a significant difference between the first and highest durations.

Looking next at the vegetation spawning, most of the simulations took around 10 to 13 seconds to complete, with the longest taking 13.33 seconds and the shortest being 10.43 seconds, giving the difference of 2.9 seconds across all simulations. With an average of 11.28 seconds and median of 11.37 seconds the simulations did not appear to demonstrate any outliers.

Although the actual erosion process was not carried out when generating the real terrain, the water flow simulation was still carried out. Taking account for all the simulations the minimum duration was 40.69 seconds and a maximum being 41.23 seconds, with a difference of 0.54 seconds. Calculating the average gives us a timing of 40.86 seconds, along with a median of 40.85 seconds.

The vegetation spawning took around 11 to 12 seconds to complete. With the minimum taking 10.93 seconds and the maximum 11.43 seconds, the timings were all fairly close to each other as the difference between the minimum and maximum are 0.5 seconds. Along with the difference, the average and median were calculated at 11.2 and 11.21 seconds.

Moving to the laptop device the timings from the erosion processes from all generations took around 128 to 133 seconds to complete, with the longest process taking 133.61 seconds and the shortest being 128.64 seconds, giving the difference of 4.97 seconds. With an average of 131.83 seconds and median of 132.04 seconds.

Vegetation spawning next, most of the simulations took around 14 to 16 seconds to complete, with the longest taking 18.69 seconds and the shortest being 14.04 seconds, giving the difference of 4.65 seconds across all simulations, with an average of 14.98 seconds and median of 14.81 seconds.

As mentioned previously only the water flow simulation was carried out for the real terrain. Taking account for all the simulations the minimum duration was 52.86 seconds and a maximum being 54.12 seconds, with a difference of 1.26

TABLE I
DESKTOP - PROCEDURAL TERRAIN FINDINGS

Erosion	**1**	**2**	**3**	**4**	**5**
#1	**38.74**	**38.71**	**38.82**	**40.9**	**35.91**
#2	**35.52**	**35.47**	**35.52**	**35.66**	**35.46**
#1	**35.52**	**35.65**	**35.49**	**35.47**	**35.89**
	Average	**Median**	**Min**	**Max**	**Difference**
	36.58	**35.65**	**35.46**	**40.9**	**5.44**
Vegetation	**1**	**2**	**3**	**4**	**5**
#1	**11.83**	**11.71**	**11.78**	**13.33**	**11.52**
#2	**10.43**	**10.59**	**10.51**	**10.75**	**10.43**
#1	**11.04**	**11.16**	**11.37**	**11.37**	**11.45**
	Average	**Median**	**Min**	**Max**	**Difference**
	11.28	**11.37**	**10.43**	**13.33**	**2.9**
Total	**1**	**2**	**3**	**4**	**5**
#1	**50.57**	**50.42**	**50.6**	**54.23**	**47.43**
#2	**45.95**	**46.06**	**46.03**	**46.41**	**45.89**
#1	**46.56**	**46.81**	**46.86**	**46.84**	**47.34**
	Average	**Median**	**Min**	**Max**	**Difference**
	47.86	**46.84**	**45.89**	**54.23**	**8.34**

TABLE II
DESKTOP - REAL TERRAIN FINDINGS

Fluid	**1**	**2**	**3**	**4**	**5**
#1	**41.14**	**40.89**	**41.06**	**40.87**	**40.84**
#2	**41.23**	**40.79**	**40.85**	**40.74**	**40.85**
#1	**40.89**	**40.71**	**40.69**	**40.69**	**40.75**
	Average	**Median**	**Min**	**Max**	**Difference**
	40.86	**40.85**	**40.69**	**41.23**	**0.54**
Vegetation	**1**	**2**	**3**	**4**	**5**
#1	**10.93**	**11.1**	**11.43**	**11.34**	**11.14**
#2	**11.06**	**11.21**	**11.21**	**11.1**	**11.28**
#1	**11.34**	**11.2**	**11.24**	**11.3**	**11.25**
	Average	**Median**	**Min**	**Max**	**Difference**
	11.20	**11.21**	**10.93**	**11.43**	**0.5**
Total	**1**	**2**	**3**	**4**	**5**
#1	**52.07**	**51.99**	**52.49**	**52.21**	**51.98**
#2	**52.29**	**52**	**52.06**	**51.84**	**52.13**
#1	**52.23**	**51.91**	**51.93**	**51.99**	**52**
	Average	**Median**	**Min**	**Max**	**Difference**
	52.07	**52**	**51.84**	**52.49**	**0.65**

TABLE III
LAPTOP - PROCEDURAL TERRAIN FINDINGS

Erosion	**1**	**2**	**3**	**4**	**5**
#1	**131.38**	**128.64**	**131.33**	**132.5**	**131.01**
#2	**132.56**	**132.18**	**133.28**	**132.04**	**133.61**
#1	**130.85**	**130.81**	**131.95**	**133.23**	**132.2**
	Average	**Median**	**Min**	**Max**	**Difference**
	131.83	**132.04**	**128.64**	**133.61**	**4.97**
Vegetation	**1**	**2**	**3**	**4**	**5**
#1	**18.69**	**15.22**	**15.37**	**15.36**	**15.39**
#2	**14.25**	**14.24**	**14.04**	**14.1**	**14.09**
#1	**14.67**	**14.74**	**14.82**	**14.81**	**14.91**
	Average	**Median**	**Min**	**Max**	**Difference**
	14.98	**14.81**	**14.04**	**18.69**	**4.65**
Total	**1**	**2**	**3**	**4**	**5**
#1	**150.07**	**143.86**	**146.7**	**147.86**	**146.4**
#2	**146.81**	**146.42**	**147.32**	**146.14**	**147.7**
#1	**145.52**	**145.55**	**146.77**	**148.04**	**147.11**
	Average	**Median**	**Min**	**Max**	**Difference**
	146.81	**146.77**	**143.86**	**150.07**	**6.21**

TABLE IV
LAPTOP - REAL TERRAIN FINDINGS

Fluid	**1**	**2**	**3**	**4**	**5**
#1	**53.32**	**52.86**	**53.37**	**53.4**	**53.51**
#2	**53.51**	**53.58**	**53.11**	**53.51**	**53.5**
#1	**53.49**	**53.48**	**53.7**	**52.89**	**54.12**
	Average	**Median**	**Min**	**Max**	**Difference**
	53.42	**53.49**	**52.86**	**54.12**	**1.26**
Vegetation	**1**	**2**	**3**	**4**	**5**
#1	**13.88**	**13.88**	**14.05**	**13.93**	**13.89**
#2	**13.74**	**13.89**	**14.65**	**13.85**	**13.85**
#1	**13.99**	**14.03**	**13.89**	**14.07**	**13.76**
	Average	**Median**	**Min**	**Max**	**Difference**
	13.95	**13.89**	**13.74**	**14.65**	**0.91**
Total	**1**	**2**	**3**	**4**	**5**
#1	**67.2**	**66.74**	**67.42**	**67.33**	**67.4**
#2	**67.25**	**67.47**	**67.76**	**67.36**	**67.35**
#1	**67.48**	**67.51**	**67.59**	**66.96**	**67.88**
	Average	**Median**	**Min**	**Max**	**Difference**
	67.38	**67.4**	**66.74**	**67.88**	**1.14**

seconds. Calculating the average gives us a timing of 53.42 seconds, along with a median of 53.49 seconds.

With the vegetation spawning taking around 13.5 to 14.5 seconds to complete. With the minimum taking 13.74 seconds and the maximum 14.65 seconds, the timings were all fairly close to each other as the difference between the minimum and maximum are 0.91 seconds. Along with the difference, the average and median were calculated at 13.95 and 13.89 seconds.

V. CONCLUSION

Results from the survey left more to be desired as with the small number of participants and answers unrelated to the actual terrains themselves, very little data was collected that was deemed as useful regarding which terrain was real or generated. A lot of the responses were solely answered based on the graphics of the presented images, which was irrelevant as both terrains use the same Unity renderer. Some of the unrelated graphics questions were that of texture repetition and lack of shadows at a distance. Future survey attempts should take graphics more seriously and aim to achieve photo-realism to remove any bias towards bad graphics.

Unlike the survey, the logs of the timings obtained during the terrain generation of both real and procedural terrains were very useful as they demonstrated how the same processes across multiple devices affect the total duration of the processes. They also highlighted how GPU bottle-necking has a significant role in increasing the duration of the erosion process and that GPUs that do not suffer bottle-necking fair far more better in terms of duration of erosion simulation.

Although for the purpose of this paper the prototype was seen as good enough, it does present several limitations, those being limiting poor noise blending options, texture repetitions, and lackluster vegetation spawning.

In the prototype blending consisted of multiply 2 noise-maps into a single result. While multiplying still has is use cases, other blend modes could prove to more optimal in certain situations. Some of the alternative blend modes could include simple equations such as addition, subtraction, and averaging, while other more complex blends could be picking either the highest or lowest value out of the 2 noise-maps.

Another improvement to blending could be to not be limited to just 2 inputs at a time, in some situations 3 or more noise-maps might be better to be blending all together at once instead of blending 2 at a time, as it could result in more interesting and unique results.

While Triplanar texturing addressed the problem of texture distortion along the z-axis, it did not solve the issue of textures repeating themselves in a grid pattern. In future additions anti texture repetitions should be implemented to increase the visual atheistic of the terrains.

After the process of vegetation spawning it was noticed that trees would spawn in areas under a certain amount of water were the trees should not have been allowed to live there, it was also noticed that will trees of the same specie did tend to group nearby each other, in some instances they were too close to one another nearly touching each other.

VI. Github

Link to github repository:

https://github.com/MatthewXuereb/Matthew_Xuereb_MSD_6.3A_Diss.git

References

[B02] Benes B. "A Stable Modeling of Large Plant Ecosystems". In: *Mexico, Ciudad de Mexico* (2002).

[J03] Stam J. "Real-Time Fluid Dynamics for Games". In: *Canada, Ontario Alias wavefront* (2003).

[J07] Andersson J. "Terrain Rendering in Frostbite using Procedural shader splatting". In: *SIGGRAPH* (2007), pp. 46–54.

[SJ08] Brisbin M Stava O Bene B and Kriv J. "Interactive Terrain Modeling Using Hydraulic Erosion". In: *Eurographics / ACM SIGGRAPH Symposium on Computer Animation* (2008).

[B11] Jako B. "Fast Hydraulic and Thermal Erosion on the GPU". In: *Hungary, Budapest University of Technology and Economics* (2011).

[MB11] Decaudin P Mei X and Hu B. "Fast Hydraulic Erosion Simulation and Visualization on GPU". In: *United States, Pacific Conference on Computer Graphics and Applications* (2011).

[T15] Beyer H T. "Implementation of a method for hydraulic erosion". In: *Germany, University of Munich* (2015).

[NZ16] Mikulicic N and Mihajlovic Z. "Procedural Generation of Mediterranean Environments". In: *Crotia, International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)* (2016).

[WR20] Bayer F Weiss S and Westermann R. "Triplanar Displacement Mapping for Terrain Rendering". In: *Germany, University of Munich* (2020).

[] *Anaconda*. https://www.anaconda.com/. Accessed: 05-05-2023.

[] *FastNoise*. https : / / github . com / Auburn / FastNoiseLite/. Accessed: 15-05-2023.

[] *GDAL*. https://gdal.org/. Accessed: 05-05-2023.

[] *USGS TNM Download (v2.0)*. https : / / apps . nationalmap . gov / downloader/. Accessed: 05-05-2023.