EasyChair Preprint

№ 6762

Procedural Modeling of the Great Barrier Reef

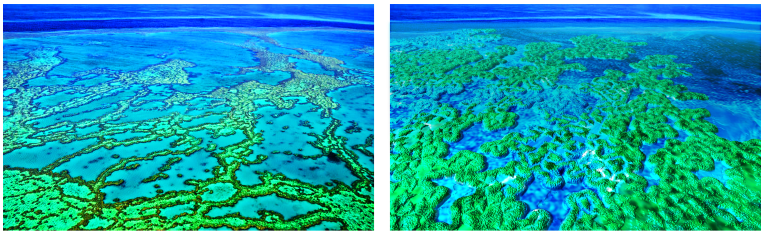
Wanwan Li

October 3, 2021

Procedural Modeling of the Great Barrier Reef

Wanwan Li

George Mason University



(a) Top-down view of real Great Barrier Reef. (b) Procedural Great Barrier Reef generation.

Fig. 1: Example of procedural Great Barrier Reef generation.

Abstract. Since terrain procedural modeling is widely adopted for the virtual natural scene generations in the game design, movie industry, and digital arts, lots of advanced techniques have been explored by researchers to procedurally synthesize a large variety of different types of terrain and landscapes. In this paper, we present a novel approach to generate a special type of landscape – the Great Barrier Reef – an amazing natural landscape that is currently being ignored. We propose a hypothesis that the Great Barrier Reef is grown with the diffusion-limited aggregation (DLA) model and simulate the DLA process to generate the Great Barrier Reef procedurally. As presented in the results, the procedural Great Barrier Reef generated with our approach looks natural when compared to the photos of the real ones.

Keywords: procedural modeling · landscape · terrain · Great Barrier Reef · diffusion-limited aggregation (DLA)

1 Introduction

As one of the most popular techniques applied in terrain authoring, procedural terrain modeling has been widely studied by researchers nowadays. Understanding the natural procedure of how the landscapes are formed is the key issue for researchers to get a deeper insight into designing efficient procedural modeling algorithms. Since the early ages, different representations has been proposed for different types of terrain modelings such as the elevation models [27](including discrete heightfields model [8] and layered mixture representations [19], etc) and the volumetric models (including voxels representation [13] and hybrid representations [23], etc). Given these observations, various types of procedural modeling algorithms have been applied to different types of terrain and landscapes synthesis. Fournier et al. [11] proposed a recursive midpoint displacement subdivision

algorithm to add fractal details on curves which can be further modified to apply to general procedural terrains. Image filters [21] can be introduced to add fractal properties on the terrain surface such as ridge filters, multi-fractal filters, and warped filters to raise crests, ridges, and valleys, etc [9]. De et al. [7] proposed a method for realistic procedural canyons generation. Featuring arches effect have been simulated on volumetric terrains by Becher et al. [5]. Terrains with thermal erosion effects have been synthesized by Musgrave et al. [19]. Spheroidal weathering effects have been implemented by Beardall et al. [4] to synthesize realistic procedural Goblins. Recently, other types of landscapes such as riverscapes synthesized by Paris et al. [22], desertscapes synthesized by Paris et al. [20], glacierscapes synthesized by Argudo et al. [2] are well generated by the applications of existing procedural modeling techniques.

However, according to the best of our knowledge, as one of the most attractive places for traveling in Australia, the Great Barrier Reef, which forms an extraordinary special landscape, has never been studied so far with respect to procedural modeling. Existing research works about the Great Barrier Reef modelings are mainly focusing on its hydrodynamic modeling [17], spectral discrimination modeling [16], water quality modeling [15], and biological population dynamics modeling [6], etc. Therefore, we open a new topic about generalizing the features observed from the Great Barrier Reef landscape and exploring an effective computer graphics algorithm to generate realistic procedural Great Barrier Reef landscapes. We conclude the contributions of our work as following:

- We initiate a discussion of an open problem about procedural Great Barrier Reef generation and propose a hypothesis that how the Great Barrier Reef is formed from a procedural modeling aspect.
- We propose a novel algorithm to procedurally synthesis the Great Barrier Reef and discuss how the parameters in our procedural model affect the landscape of the Great Barrier Reef.
- We conclude the limitations of our work and prospect the next steps to extend our work for the future study about procedural Great Barrier Reef generation and its potential applications.

2 Growth of the Great Barrier Reef: A Hypothesis

As we know, the Great Barrier Reef, as the world's largest coral reef system [18], most of its major components are individual reefs, combined with stretching islands as minor components. As coral reefs are the skeleton of coral worms, therefore, the landscape of the coral reefs' colonies looks more like living creatures rather than ordinary terrains. By assuming that coral reefs landscapes are sharing similar features with the distributions of leaving creatures that are aggregated into colonies naturally, we propose a hypothesis that the growth of the Great Barrier Reef flows the rules of diffusion-limited aggregation (DLA) process [26] which is a natural aggregation process caused by the individual particles' random walks due to their Brownian motion [25]. DLA algorithm [14] is a

computational procedure that simulates the process of how random particles are aggregated into colonies within the natural environment. To describe this phenomenon, in short, DLA simulations have three main steps: (1) setup a random aggregation kernel, (2) releasing random particles with Brownian motions, and (3) aggregate those particles whenever two individual particles are neighboring to each other. When evaluating whether such a DLA process fits well our problem definition of growing the Great Barrier Reef, we need to argue that whether these three steps in DLA are able to be mapped to the growth of the Great Barrier Reef. Here is a tentative inductive reasoning proof:

Proof. Step (1) in DLA is reasonable when assuming that there is the first coral reef appearing in the Great Barrier Reef. Step (2) in the DLA can be interpreted as there is the second coral reef given birth in a random place in the Great Barrier Reef, which can be reasonable as well. Step (3) is actually the discussion that whether the second coral reef is necessarily nearby the first coral reef. Fortunately, this can be proved by the fact that the coral reef is the skeleton of the coral worms and the coral worms are actually not moving very frequently. This means coral worms tend to give birth in place and hence every two colonies will not be too far away from each other if there is an inheritance relationship between these two.

Therefore, according to this proof of concept, the DLA process is reasonable for simulating the growth of the Great Barrier Reef and this theoretically defends our hypothesis from a philosophical view. Given this theoretical hypothesis and proof, in the following sections, we will present our proposed problem formulation, discuss the experimental results, and prospect the future works.

3 Problem Formulations

3.1 Terrain Representation

In our work, we are using the discrete heightfields, also called heightmap, to represent the elevations of terrain. The heightmap is an image storing the elevation height of the terrain on arbitrary pixels. Typically, heightmap is represented as a $M \times N$ normalized scalar matrix $H_{M\times N}$ consisted with real numbers between 0 and 1. Given any 2D vector $\mathbf{p} = (u, v) \in \mathbb{R}^2$ as a location on terrain, we can calculate the elevation of terrain surface at location $\mathbf{p}$, denoted as $h(\mathbf{p})$, using interpolation methods. Assume that (u_0, v_0) and (u_1, v_1) are the two corners on the diagonals of the rectangle region of the terrain respectively, then the heightmap grids interval $(\Delta u, \Delta v) = ((u_1-u_0)/(M-1), (v_1-v_0)/(N-1)$. The terrain's corresponding location of the pixel on heightmap's i^{th} row and j^{th} column can be calculated through a mapping function $\Phi(i, j) = (u_0, v_0)+(i, j)\cdot(\Delta u, \Delta v)$ where $(i, j) \in \mathbb{I}^2$. Noted that $h(\Phi(i, j)) = h_{\max}H_{i,j}$, where $h_{\max}$ is the maximum height of the terrain. As shown in Fig 2(a), we take a realistic terrain heightmap(gray) generated with the approach proposed by De et al. [7] with canyons effects as the base map $h_0(u, v)$ upon which our Great Barrier Reef grows.

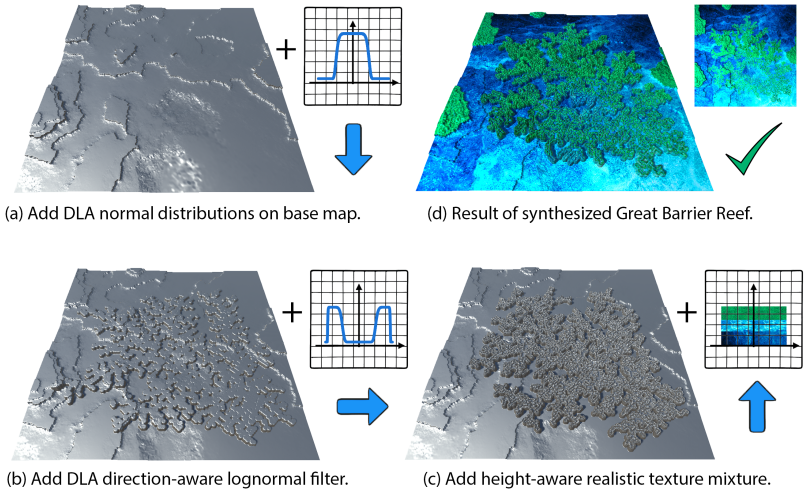


Fig. 2: Overview of our technical approach.

3.2 DLA Normal Distributions

DLA normal distribution is a compound term that combines the term of DLA (short for diffusion-limited aggregation) and the term of normal distributions. As this term suggests, DLA-normal distribution is the scale mixtures of normal distributions [1] where the centers of the individual normal distributions are following a spatial distribution pattern of diffusion-limited aggregation. Mathematically, we form a DLA distribution on a 2D terrain coordinate space. First, we create a boolean matrix DLA grids mask $B_{M\times N} \in \{0,1\}_{M\times N}$. Then, we turn an arbitrary bit of the DLA grid mask on and take it as the diffusion kernel. Finally, we repeatedly sample pixel (i,j) on the terrain with random walk step $(\Delta i, \Delta j)$, turn on the pixel (i,j)'s mask $B_{i,j} = 1$ where exists any one of its eight adjacent neighbor pixels $(i',j') \in \{(i \pm k_1\Delta i, j \pm k_2\Delta j)|k_1,k_2 \in \{0,1\}\}$ that has mask also been turned on, namely, when $B_{i',j'} = 1$. Noted that the random walk step $(\Delta i, \Delta j)$ is also interpreted as DLA grids cell size, we will discuss more about how the grids cell size settings influence the results in the following section. After the DLA calculation step, we accumulate the convolutions between terrain's base map $h_0(\mathbf{p})$ and Gaussian function with pixel center (i, j) as a new heightmap $h_1(u,v)$ through Eq. 1:

$$h_1(\mathbf{p}) = w_0 h_0(\mathbf{p}) + w_1 \sum_{\mathbf{q}\in\{\Phi(i,j)|B_{i,j}=1\}} h_0(\mathbf{p}) \circ e^{-||\mathbf{p}-\mathbf{q}||^k} \tag{1}$$

where we empirically set $k = 10$ for a realistic result. As shown in Fig. 2(b), after this convolution process, those that belong to the DLA areas on the base map are lifted up to a certain level specified by w_1, the edges are smoothed by the Gaussian filter. This procedural is used to simulate the aggregation of the

corals under the sea. As the amount of the coral reef increases, they will float above sea level and form into islands.

3.3 Directional DLA Log-Normal Filter

Fig. 3: Edge Uplift Effects.

According to the definition of log-normal distribution in statistical science [24], it is used to describe the distributions whose logarithm is normally distributed. From the shape of the log-normal distribution function, it looks more like a biased normal distribution. As one interesting phenomenon shown in Fig. 3 which is downloaded from the website [3], the Great Barrier Reef is captured from a top-down view. As we can see, the edges of the coral reef are frequently uplifted due to the aggregation of corals near the seashore. Inside the edges, the corals reefs look more plat. This observation inspire us to use the log-normal distribution function as a filter to uplift the edges of the procedural landscape to generate such effect. In this scenario, we propose a directional DLA log-normal filter to convolute the edges of the DLA grids using log-normal distributions function. After adding the normal distributions in the previous step, we accumulate the convolutions between terrain's DLA normal distributions map $h_1(\mathbf{p})$ and the Log-Gaussian function with pixel centers (i, j) in the DLA grids as a final heightmap $h_2(u, v)$ through Eq. 2:

$$h_2(\mathbf{p}) = h_1(\mathbf{p}) + w_2 \sum_{\mathbf{q}\in\{\Phi(i,j)|B_{i,j}=1\}} h_0(\mathbf{p}) \circ w_{\mathrm{d}}(\mathbf{p},\mathbf{q}) e^{-\ln k_1 (1-||\mathbf{p}-\mathbf{q}||)^{k_2}} \quad (2)$$

where we empirically set $k_1 = 5$ and $k_2 = 10$ for a realistic rendering result. Furthermore, we add random values on k_1 to simulate the noises on the edges.

The directional weight sampler $w_{\mathrm{d}}(\mathbf{p},\mathbf{q})$ returns a value between 0 and 1. $w_{\mathrm{d}}(\mathbf{p},\mathbf{q})$ is calculated through the following steps: Step (1) Calculate the angle between the y-axis and $\mathbf{q}-\mathbf{p}$. The calculated angle is between 0 and 360 degrees. Step (2) we divide the 360-degree angles into eight directions and check out the calculated angle belongs to which direction. Step (3), we search the current pixel index (i, j) of the location $\mathbf{q}$ and its adjacent neighbor pixels at that calculated direction, if that neighbor belongs to the DLA grids, then we set the weight of that direction into 0, otherwise, we set $w_{\mathrm{d}}(\mathbf{p},\mathbf{q}) = 1$. As shown in Fig. 2(c), after applying the directional DLA log-normal filter, those lifted regions' edges are furtherly lift up to a certain level specified by w_2. The edges are noised. The inner area is not lifted as much as the edges due to the directional weight multiplier. This procedural is used to realistically simulate the effects shown in Fig. 3. After applying these two major steps of the convolutions on the terrain map, another two optional minor steps are applied on the terrain heightmap including a blurring step to remove high-frequency artefacts and a normalization step to refine the heightmap values ranging between 0 and 1.

3.4 Height-Aware Terrain Texture Synthesis

Texture synthesis is an important step to generate a realistic terrain appearance. There are lots of different types of texture synthesis methods such as pixel-based non-parametric sampling methods proposed by Efros et al. [10] and its applications on terrain texture [12]. In our work, we consider the terrain characteristics such as the absolute elevation function $h_2(\mathbf{p})$ calcualted from the above steps. According to the observation, the deep sea areas has texture with dark blues, shallow sea ares are colored with light blues, and the coral reef island has a green blue texture. Terefore, in orer to blend the textures smoothly, we propose two functions as the alphamap values for texture mixture: Lowpass Function P_{low} and Highpass Function P_{high} which are defined in Eq. 3 and Eq. 4 respectively:

$$P_{\text{low}}(h(\mathbf{p}), h_{\text{low}}) = \begin{cases} 1 & h(\mathbf{p}) \leq h_{\text{low}} \\ P_{\text{error}}(|h(\mathbf{p}) - h_{\text{low}}|, \epsilon_{\text{low}}) & h(\mathbf{p}) > h_{\text{low}} \end{cases} \tag{3}$$

$$P_{\text{high}}(h(\mathbf{p}), h_{\text{high}}) = \begin{cases} 1 & h(\mathbf{p}) \geq h_{\text{high}} \\ P_{\text{error}}(|h(\mathbf{p}) - h_{\text{high}}|, \epsilon_{\text{high}}) & h(\mathbf{p}) < h_{\text{high}} \end{cases} \tag{4}$$

where Errorpass Function $P_{\text{error}}(d, \epsilon)$ compare the distance d and the error ϵ as shown in Eq. 5:

$$P_{\text{error}}(d, \epsilon) = \begin{cases} 0 & d \geq \epsilon \\ 1 - d/\epsilon & d < \epsilon \end{cases} \tag{5}$$

Then, we have the final alphamap values for deep sea texture α_{d}, shallow sea texture α_{s}, and coral reef island texture α_{c} are calculated through Eq. 6:

$$\begin{bmatrix} \alpha_{\text{d}}(\mathbf{p}) \\ \alpha_{\text{s}}(\mathbf{p}) \\ \alpha_{\text{c}}(\mathbf{p}) \end{bmatrix} = \begin{bmatrix} P_{\text{low}}(h(\mathbf{p}), h_{\text{low}}) \\ (1 - P_{\text{low}}(h(\mathbf{p}))(1 - P_{\text{high}}(h(\mathbf{p}))) \\ (1 - P_{\text{low}}(h(\mathbf{p}))P_{\text{high}}(h(\mathbf{p})) \end{bmatrix} \tag{6}$$

Empirically, we have set height texture borderlines $h_{\text{low}} = 0.3$ and $h_{\text{high}} = 0.5$ Borderlines errors are set as $\epsilon_{\text{low}} = 0.1$ and $\epsilon_{\text{high}} = 0.05$ respectively. As shown in Fig. 2(d), the final terrain mesh generated with our approach has been colored with this proposed height-aware texture synthesis method mentioned above. As we can see, those areas that belong to the DLA areas on the heightmap are colored as dark green-blue texture. This mimics the green island floating above the sea. In the middle area, they are shallow water with light bright blue. As the deeper blue highlight the areas where belong to the deep sea. This rendering's visual effects look realistic as the texture synthesis methods obey the natural rules observed from real photography as we explained above.

4 Experimental Results

We have implemented the proposed approach to synthesize the Great Barrier Reef using Unity 3D with the 2019 version. The hardware configurations contain Intel Core i5 CPU, 32GB DDR4 RAM, and NVIDIA GeForce GTX 1650 4GB GDDR6 Graphics Card. Our proposed algorithm is implemented on the CPU. We have tested our synthesized Great Barrier Reef landscapes with different settings. In this section, we will demonstrate the robustness of our proposed approach by changing the input terrains, changing the DLA aggregation centers, changing the DLA grids sizes, and changing the textures.

4.1 Changing Input Terrains

As shown in Fig 4, we have synthesized Great Barrier Reef landscapes with three different terrain inputs. All of the terrains are randomly generated with the same parameter settings but with different random numbers generator seeds inputs. All terrains are generated with a heightmap of 4096x4096 pixels. The terrain's real width is 10km and its real height is 500m. Among these three results, the DLA grid size is set to 150 (dimension of the DLA grids), the grids radius is 40 (convolution kernel size) and the aggregation center is (120, 120). The DLA particle shooting number is 3000. The maximum Brown motion random walk steps are 30000. The heightmap blending weights for base map h_0, DLA normal distribution map h_1 and the DLA log-normal filter map h_2 are $w_0 = 0.5$, $w_1 = 0.2$, and $w_2 = 0.1$ respectively. The execution times for these three terrain synthesis results are 21 sec, 26 sec, and 24 sec respectively. As we can see, our approach is robust to different kinds of terrain inputs.

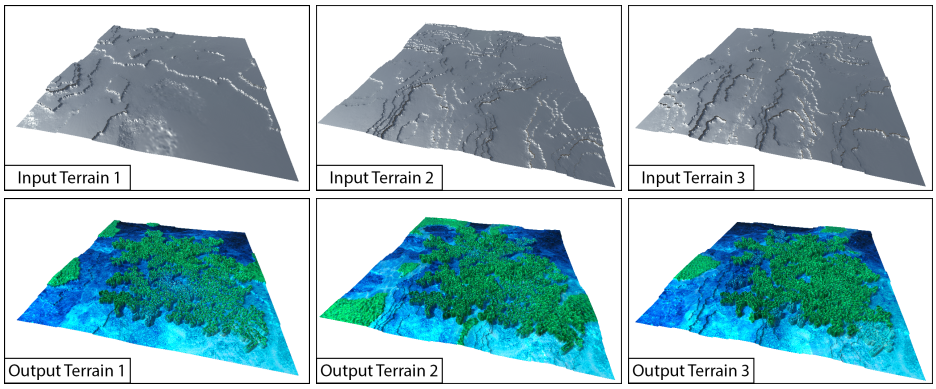


Fig. 4: Experiments results of changing input terrains.

4.2 Changing Aggregation Centers

Another interesting phenomenon to show is that our approach is robust to aggregation centers settings as well. As shown in Fig. 5, we have three different aggregation center settings respectively. The three centers are labeled through three red pins. In this test, we have the DLA grids settings as same as above where the DLA grid size is set to 150 and the grids radius is 40. However, we have three different aggregation centers settings are left bottom (20, 130), middle center (80, 80), and right top (130, 20) respectively. The aggregation center is the DLA kernel where the first particle is put in DAL girds. Then other particles will be aggregated around this kernel. As we can see, different DLA kernels will affect the overall distributions of the coral reefs in our synthesized Great Barrier Reef. The shape of the landscape is smoothly moving from the left bottom (left subfigure) towards the middle center (middle subfigure) and finally move to the right top (right subfigure).

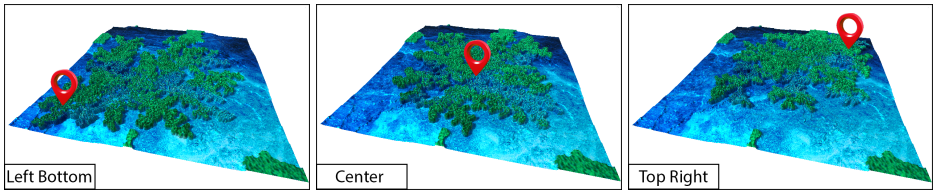


Fig. 5: Experiments results of changing aggregation centers.

4.3 Changing DLA Grid Sizes

The DLA grid size settings are essential for our algorithm as it adjusts the resolution of our synthesized Great Barrier Reef. As shown in Fig. 6, we have three different grid Sizes settings respectively.

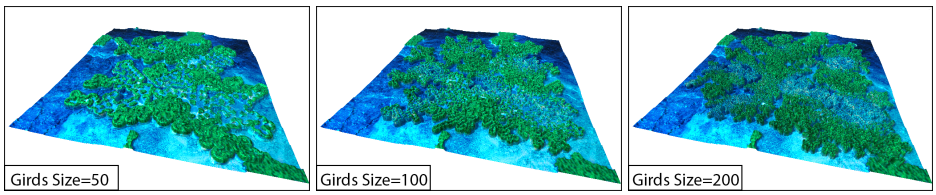


Fig. 6: Experiments Results of changing DLA grid sizes.

The left subfigure shows the result synthesized with the grid size of 50, grids radius of 120, and the aggregation center is (40, 40), the particle shooting number is 500, and its running time is 19 secs. The middle subfigure shows the result synthesized with the grid size of 100, grids radius of 60, and the aggregation center is (80, 80), the particle shooting number is 2000, and its running time is 22 secs. The right subfigure shows the result synthesized with the grid size of 200, grids radius of 30, the aggregation center is (180, 180), the particle shooting number is 8000, and its running time is 28 secs. As we can see, as the grid size decreases, more particles are needed to aggregate into the same large-scale terrain and in turn, it takes more time for running, and it results in more details of the synthesized terrain.

4.4 Changing Textures

For showing some special visual effects, we have generalized the idea of "Great Barrier Reefs". Hereby, we have used three different texture styles to synthesize the Great Barrier Reefs. As shown in Fig. 7, the left subfigure shows the Great Barrier Reefs rendered with a stony mountain style. In this case, the lower part is filled with stones and grass, the middle part is the stony hills, and the upper parts are filled with green vegetation. The middle subfigure shows the Great Barrier Reefs rendered with a desert hill style. In this case, the lower part is filled with yellow sands, the middle part is filled with some sparse vegetation, while the upper parts are filled with sandy hills. The subfigure on the right shows the Great Barrier Reefs rendered with a volcano style. In this case, the lower part is filled with burned stones, the middle part is filled with red lava, while the upper parts are filled with volcano outbreaks. As we can see, the results shown here look realistic and is ready to be considered to be used in the movie for rendering some special visual effects.

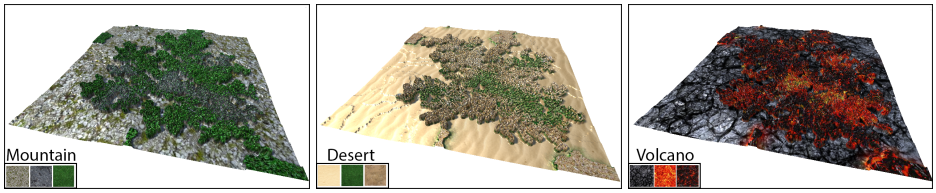


Fig. 7: Experiments results of changing textures.

5 Conclusion

In this paper, we have proposed an interesting question about how to generate the Great Barrier Reef using procedural modeling techniques. By proposing a

hypothesis that the Great Barrier Reef is formed through the diffusion-limited aggregation (DLA) process, we have simulated the DLA process and combining it onto a realistic terrain through two types of convolutional kernels: DLA normal distribution kernel and the DLA log-normal edge enhancement kernel. After applying directionally weighted randomized DLA log-normal filters, we can synthesize the terrains with realistic visual effects. As shown in the experimental results, We have visually proved the robustness of our proposed approach by changing the parameters such as changing the input terrains, changing the DLA aggregation centers, changing the DLA grids sizes, and changing the textures.

In the future work, we will further explore our approach by adding more details such as including the corals and marine environment on the generated Great Barrier Reef. This will prepare our work ready to be used in the movie industry and game industry for the scenarios that are near the seashore with a special landscape that looks like the Great Barrier Reef. We believe our work will inspire more researchers to follow up and generate more eye-catching results about realistic procedural modelings of the Great Barrier Reef.

References

1. Andrews, D.F., Mallows, C.L.: Scale mixtures of normal distributions. Journal of the Royal Statistical Society: Series B (Methodological) **36**(1), 99–102 (1974)
2. Argudo, O., Galin, E., Peytavie, A., Paris, A., Guérin, E.: Simulation, modeling and authoring of glaciers. ACM Transactions on Graphics (TOG) **39**(6), 1–14 (2020)
3. Arikoglu, L.: Scientists discover secret reef...behind the great barrier reef. https://www.cntraveler.com/story/scientists-discover-secret-reef-below-the-great-barrier-reef (2016)
4. Beardall, M., Farley, M., Ouderkirk, D., Smith, J., Jones, M., Egbert, P.K.: Goblins by spheroidalweathering. In: NPH. pp. 7–14 (2007)
5. Becher, M., Krone, M., Reina, G., Ertl, T.: Feature-based volumetric terrain generation. In: Proceedings of the 21st ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games. pp. 1–9 (2017)
6. Chaloupka, M.: Stochastic simulation modelling of southern great barrier reef green turtle population dynamics. Ecological modelling **148**(1), 79–109 (2002)
7. De Carli, D.M., Pozzer, C.T., Bevilacqua, F., Schetinger, V.: Procedural generation of 3d canyons. In: 2014 27th SIBGRAPI Conference on Graphics, Patterns and Images. pp. 103–110. IEEE (2014)
8. De Carpentier, G.J., Bidarra, R.: Interactive gpu-based procedural heightfield brushes. In: Proceedings of the 4th International Conference on Foundations of Digital Games. pp. 55–62 (2009)
9. Ebert, D.S., Musgrave, F.K., Peachey, D., Perlin, K., Worley, S.: Texturing & modeling: a procedural approach. Morgan Kaufmann (2003)
10. Efros, A.A., Leung, T.K.: Texture synthesis by non-parametric sampling. In: Proceedings of the seventh IEEE international conference on computer vision. vol. 2, pp. 1033–1038. IEEE (1999)
11. Fournier, A., Fussell, D., Carpenter, L.: Computer rendering of stochastic models. Communications of the ACM **25**(6), 371–384 (1982)
12. Gain, J., Merry, B., Marais, P.: Parallel, realistic and controllable terrain synthesis. In: Computer Graphics Forum. vol. 34, pp. 105–116. Wiley Online Library (2015)

13. Geiss, R.: Generating complex procedural terrains using the gpu. GPU gems **3**, 7–37 (2007)
14. Hurd, A.J., Schaefer, D.W.: Diffusion-limited aggregation in two dimensions. Physical review letters **54**(10), 1043 (1985)
15. Khan, U., Cook, F.J., Laugesen, R., Hasan, M.M., Plastow, K., Amirthanathan, G.E., Bari, M.A., Tuteja, N.K.: Development of catchment water quality models within a realtime status and forecast system for the great barrier reef. Environmental Modelling & Software **132**, 104790 (2020)
16. Kutser, T., Dekker, A.G., Skirving, W.: Modeling spectral discrimination of great barrier reef benthic communities by remote sensing instruments. Limnology and Oceanography **48**(1part2), 497–510 (2003)
17. Legrand, S., Deleersnijder, E., Hanert, E., Legat, V., Wolanski, E.: High-resolution, unstructured meshes for hydrodynamic models of the great barrier reef, australia. Estuarine, coastal and shelf science **68**(1-2), 36–46 (2006)
18. Lucas, P., Webb, T., Valentine, P., Marsh, H.: The outstanding universal value of the great barrier reef world heritage area (1997)
19. Musgrave, F.K., Kolb, C.E., Mace, R.S.: The synthesis and rendering of eroded fractal terrains. ACM Siggraph Computer Graphics **23**(3), 41–50 (1989)
20. Paris, A., Peytavie, A., Guérin, E., Argudo, O., Galin, E.: Desertscape simulation. In: Computer Graphics Forum. vol. 38, pp. 47–55. Wiley Online Library (2019)
21. Perlin, K.: An image synthesizer. ACM Siggraph Computer Graphics **19**(3), 287–296 (1985)
22. Peytavie, A., Dupont, T., Guérin, E., Cortial, Y., Benes, B., Gain, J., Galin, E.: Procedural riverscapes. In: Computer Graphics Forum. vol. 38, pp. 35–46. Wiley Online Library (2019)
23. Peytavie, A., Galin, E., Grosjean, J., Mérillou, S.: Arches: a framework for modeling complex terrains. In: Computer Graphics Forum. vol. 28, pp. 457–467. Wiley Online Library (2009)
24. Siano, D.B.: The log-normal distribution function. Journal of Chemical Education **49**(11), 755 (1972)
25. Uhlenbeck, G.E., Ornstein, L.S.: On the theory of the brownian motion. Physical review **36**(5), 823 (1930)
26. Witten Jr, T.A., Sander, L.M.: Diffusion-limited aggregation, a kinetic critical phenomenon. Physical review letters **47**(19), 1400 (1981)
27. Zhou, H., Sun, J., Turk, G., Rehg, J.M.: Terrain synthesis from digital elevation models. IEEE transactions on visualization and computer graphics **13**(4), 834–848 (2007)