Synthesizing Realistic Cracked Terrain for Virtual Arid Environment Generation

Wanwan Li
Department of Computer Science
University of Tulsa
Tulsa, Oklahoma, USA
wanwan-li@utulsa.edu

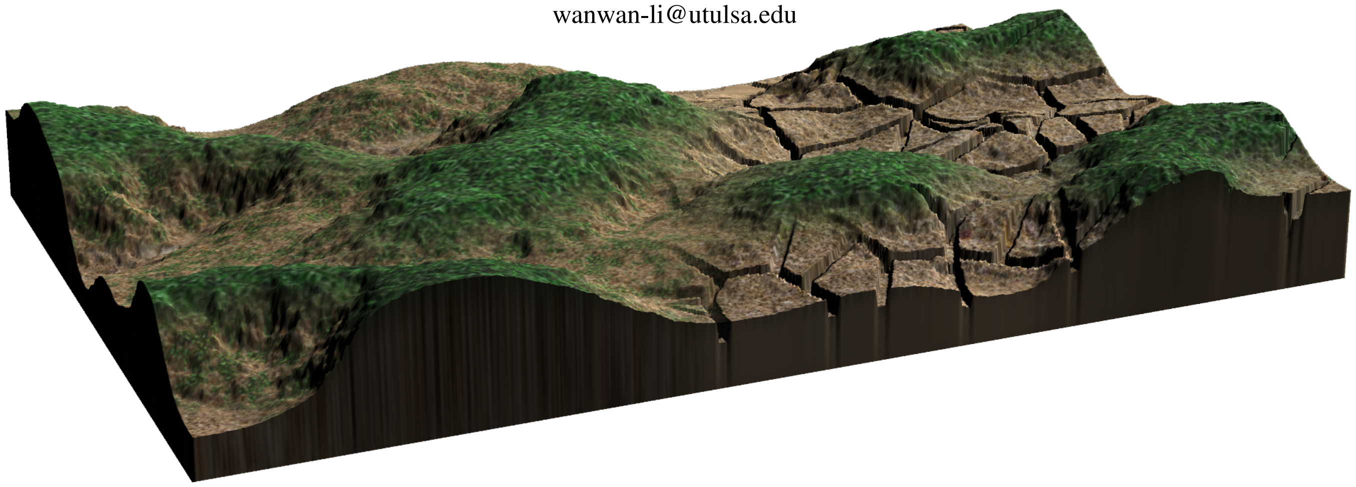


Fig. 1. Teaser. This teaser shows an example of synthesizing realistic cracked terrain for virtual arid environment generation. An original virtual terrain created with the existing procedural modeling approach (left half-figure) is automatically converted into a virtual cracked arid terrain (right half-figure) using our approach to enhance the authenticity of virtual arid environments.

***Abstract*—Virtual environments play a pivotal role in various applications such as video games, simulations, and virtual reality experiences. Creating realistic and immersive virtual arid environments necessitates the generation of terrain featuring cracked surfaces, mirroring the parched landscapes of arid regions. This paper presents a novel approach for synthesizing realistic cracked terrain to enhance the authenticity of virtual arid environments. We explore the underlying principles of terrain generation, the significance of realistic cracked features, and the computational techniques used to achieve this realism. Through a series of numerical experiments, we validate the correctness of our proposed technical approach for synthesizing realistic cracked terrain for virtual arid environment generation.**

***Keywords*—procedural modeling, terrain generation, Voronoi diagram, Dirichlet tessellation**

I. Introduction

Virtual environments are becoming increasingly indispensable for various applications, including gaming, education, training, and simulation. Terrain generation is a fundamental aspect of virtual environment creation. Realistic terrain models are vital for achieving immersive and visually appealing environments. Various methods, such as heightmap-based representations [1], procedural modeling techniques [2], data-driven approaches [3], environments synthesis for virtual reality applications [4]–[6], genetic algorithms-based landscape [7], [8], parametric-controllable terrain [9], [10], Delaunay triangulation-based landscape [11], diffusion equation-generated landscape [12], software agents-controlled terrain [13], hydrology features [14], volumetric terrain [15], large-scale terrain [16], GPU-based geometry clipmaps for terrain generations [17], [18], generative adversarial networks-based terrain generations [19]–[22], terrain point cloud rendering [23], gradient terrain authoring [24], have been used to create realistic virtual terrains and procedural landscapes.

However, there is no existing research work that has systematically explored the procedural modeling approaches to synthesize realistic cracked terrain for virtual arid environment generation. Therefore, given this observation, our focus is on incorporating realistic cracked features into given terrains through a novel procedural modeling approach. To create truly immersive experiences in virtual arid environments, the representation of terrain with realistic cracked surfaces is paramount. These cracks mimic the natural effects of arid climates and contribute to the overall authenticity of the environment. In this paper, we propose a novel method for synthesizing realistic cracked terrain to improve the visual quality and immersion of virtual arid environments. As shown in the example of Fig. 1, cracked terrain synthesized with our approach is characterized by irregular patterns of cracks, gaps, fissures, and fractures on the surface, resembling the natural effects of desiccation and weathering. Simulating cracked terrain adds an extra layer of realism to virtual arid environments.

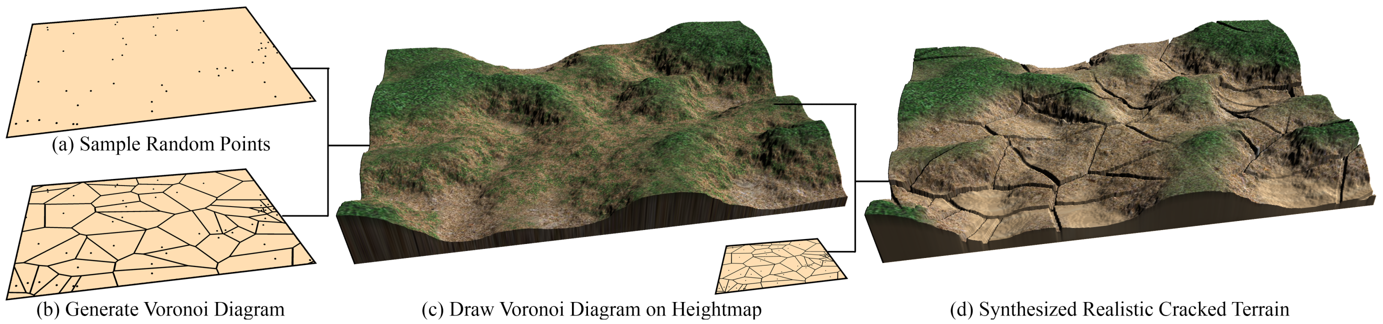

(a) Sample Random Points
(b) Generate Voronoi Diagram
(c) Draw Voronoi Diagram on Heightmap
(d) Synthesized Realistic Cracked Terrain

Fig. 2. Overview of our approach.

II. Overview

Fig. 2 shows the overview of our approach to generating realistic cracked terrains. The process begins with an arbitrary input terrain with a rectangular shape, where we initiate our approach by randomly sampling points within this rectangular area as depicted in (a). Subsequently, we employ the Bowyer-Watson algorithm [25], [26] to construct a graph data structure which serves as the basis for our proposed procedural modeling approach, called Delaunay Triangulation. After computing the dual graph of the Delaunay triangulation, the Voronoi diagram is created as visualized in (b). Then, we propose a novel procedural modeling approach to draw the Voronoi diagram on heightmap as shown in (c). We illustrate our method for rendering the Voronoi diagram on a heightmap, where each Voronoi cell's edges and vertices are given distinct elevations according to the gray color of the heightmap. This heightmap drawn with the Voronoi diagram forms the foundation for the final step in our process, where we synthesize a remarkably realistic cracked terrain that mirrors the intricacies of natural landscapes. In the end, a realistic cracked terrain is synthesized as shown in (d).

III. Technical Approach

Given a rectangle area of arbitrary input terrain, whose heightmap is denoted as $h(u,v) \in [0,1], (u,v) \in [0,1]^2$. First, randomly sample n points within this rectangle area as $V = \{\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_n,\}$, where $\mathbf{v}_i = (u_i, v_i) \in [0,1]^2$. Next, generate the Delaunay triangles data structure $G = (V, E)$ using the Bowyer-Watson algorithm, where $E = \{e_1, e_2, ..., e_m\}$ are m edges of Delaunay triangles. Then, let $G' = (V', E')$ be the dual graph of the Delaunay triangulation, then G' is called a Voronoi diagram. Mathematically, $G' = Vor(G) = (V', E')$, where $V' = \{\mathbf{v}'_1, \mathbf{v}'_2, ..., \mathbf{v}'_k, ...\}$, $\mathbf{v}'_k = (u'_k, v'_k) \in [0,1]^2$ and $E' = \{e'_1, e'_2, ..., e'_k, ...\}$, $e'_k = \{\mathbf{v}'_i, \mathbf{v}'_j\} = \{(u'_i, v'_i), (u'_j, v'_j)\}$. Finally, we draw the Voronoi diagram on the heightmap $h(u,v)$ to get a new heightmap $h'(u,v)$ to synthesize the cracked terrain. Mathematical, $h'(u,v)$ is calculated as:

$$h'(u,v) = (1 - \kappa\iota(h(u,v), \psi, \epsilon)\aleph(u,v))h(u,v), \quad (1)$$

where κ is crack depth and the lowpass function $\iota(x, \psi, \epsilon)$ is:

$$\iota(x,\psi,\epsilon) = \begin{cases} 1 & x \le \psi \\ \delta(|x-\psi|, \epsilon) & x > \psi \end{cases}, \quad (2)$$

where ψ is border line, ϵ is border error, and the errorpass function $\delta(d, \epsilon)$ comparing the distance d and error ϵ is:

$$\delta(d,\epsilon) = \begin{cases} 0 & d \ge \epsilon \\ 1 - d/\epsilon & d < \epsilon \end{cases} \quad (3)$$

In Eq. 1, crack map $\aleph(u,v) \in [0,1], (u,v) \in [0,1]^2$ is an alphamap to specify the distribution of the cracks on terrain, which is mathematically defined as:

$$\aleph(u,v) = \begin{cases} 1 & ||(u,v) - (u^*,v^*)|| \le \lambda/2 \\ 0 & ||(u,v) - (u^*,v^*)|| > \lambda/2 \end{cases}, \quad (4)$$

where λ is crack width and (u^*, v^*) is the point closest to one of the Voronoi diagram's edge. Mathematically, we have:

$$(u^*,v^*) = \underset{\{\mathbf{v}'_i, \mathbf{v}'_j\} \in E'}{\text{argmin}} \ \zeta\left((u^*,v^*), \{(u'_i,v'_i),(u'_j,v'_j)\}\right) \quad (5)$$

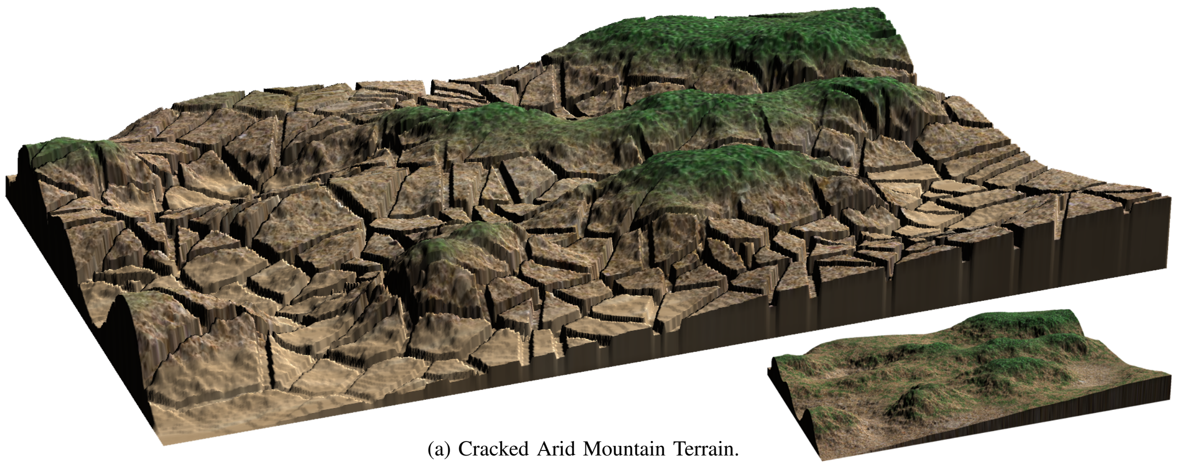
where $\zeta(\mathbf{p}, \{\mathbf{q}, \mathbf{r}\})$ is the distance between point $\mathbf{p}$ and line segment $\{\mathbf{q}, \mathbf{r}\}$, Mathematically, $\zeta(\mathbf{p}, \{\mathbf{q}, \mathbf{r}\})$ is calculated as:

$$\zeta(\mathbf{p}, \{\mathbf{q},\mathbf{r}\}) = \left|\left|\mathbf{p} - \mathbf{q} - \frac{\mathbf{r}-\mathbf{q}}{||\mathbf{r}-\mathbf{q}||}\left((\mathbf{p}-\mathbf{q}) \cdot \frac{\mathbf{r}-\mathbf{q}}{||\mathbf{r}-\mathbf{q}||}\right)\right|\right|$$

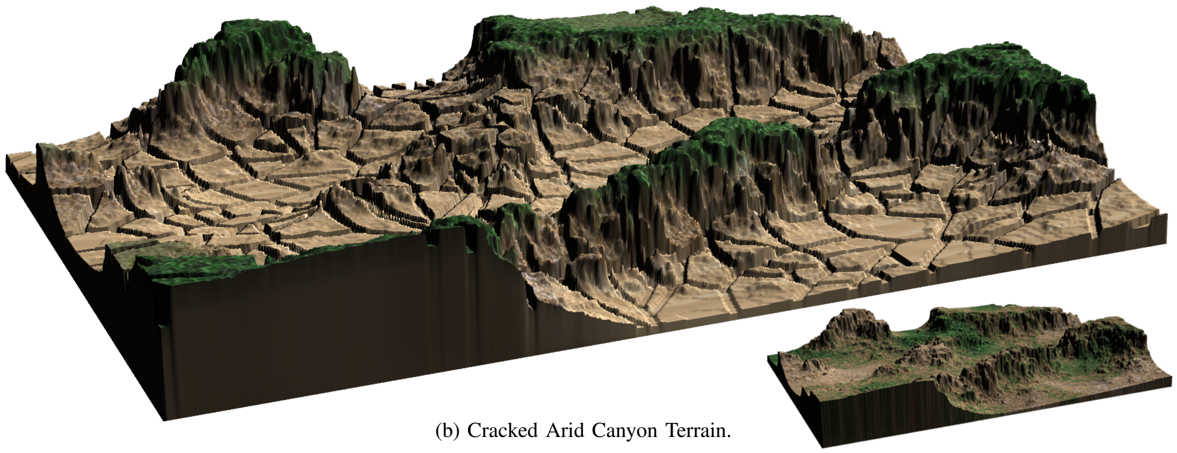
IV. Experiment Results

In order to validate the efficacy of our proposed technical approach, a group of numerical experiments are conducted on synthesizing cracked terrain for virtual arid environment generation with different parameter settings. We implemented our proposed approach using Unity 3D with the 2019 version and generated these experiment results with hardware configurations containing Intel Core i5 CPU, 32GB DDR4 RAM, and NVIDIA GeForce GTX 1650 4GB GDDR6 Graphics Card. These numerical experiments are conducted under the default parameter settings including: random sample points count $n = 200$, crack depth $\kappa = 0.3$, crack width $\lambda = 0.1$, border line $\psi = 0.4$, and border error $\epsilon = 0.2$.

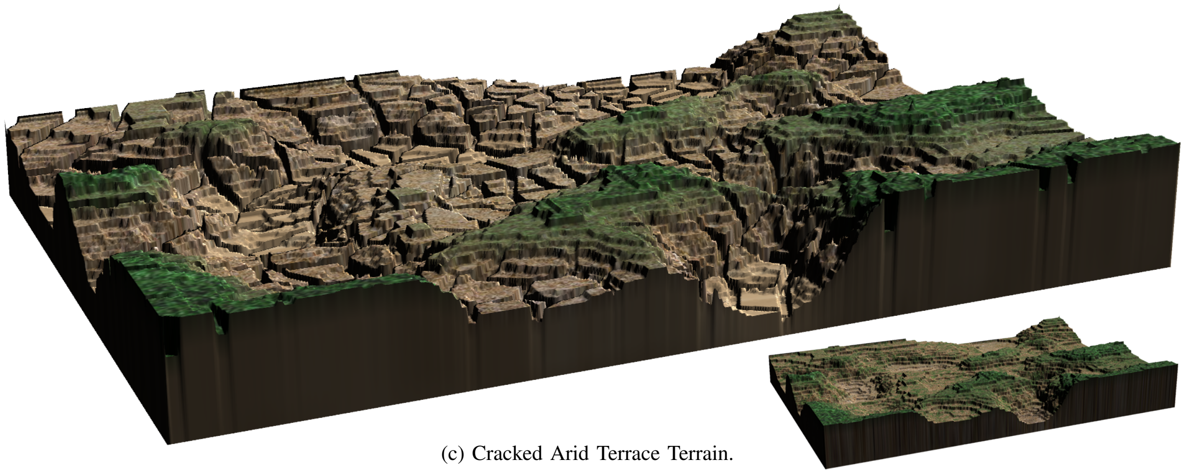
Fig. 3 shows the results of synthesizing cracked arid terrains on different types of input terrains revealing distinct and fascinating characteristics. As shown in (a), the cracked arid terrain synthesized on mountain terrain exhibits rugged, undulating features, with deep crevices and elevated plateaus, reminiscent of the natural topography of arid mountain regions. As shown



(a) Cracked Arid Mountain Terrain.



(b) Cracked Arid Canyon Terrain.

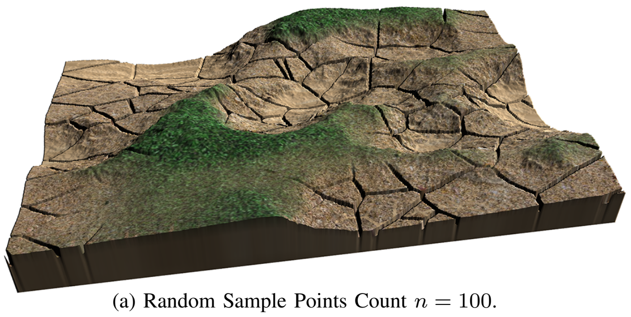


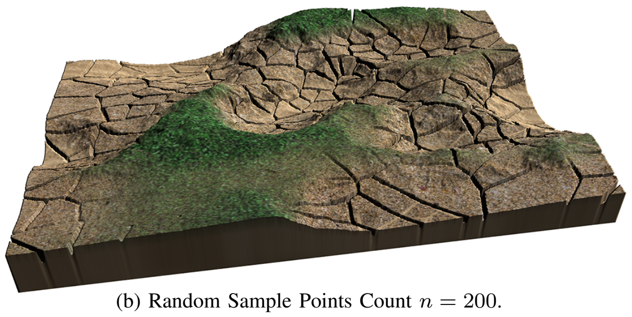
(c) Cracked Arid Terrace Terrain.

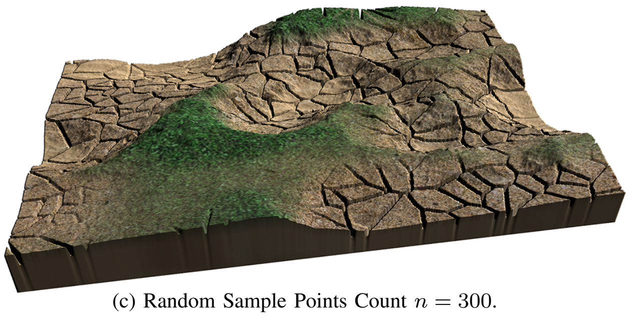
Fig. 3. Different Terrains. These results are cracked arid terrains synthesized on different types of input terrains.

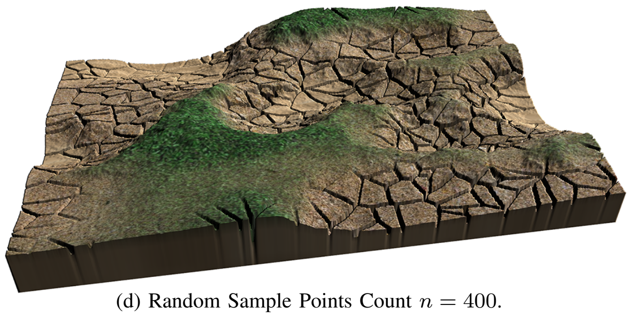
in (b), when the same cracked arid terrain synthesis process is applied to canyon terrain, it results in intricate, winding canyons surrounded by cracked, parched earth, creating an intricate blend of erosion patterns. as shown in (c), in contrast, synthesizing cracked arid terrain on terrace terrain showcases a unique blend of geometric patterns, with terraced steps resembling ancient agricultural practices. These results underscore the versatility of the synthesis process and its ability to mimic diverse landscapes while highlighting the interplay between terrain types and the arid conditions they represent.

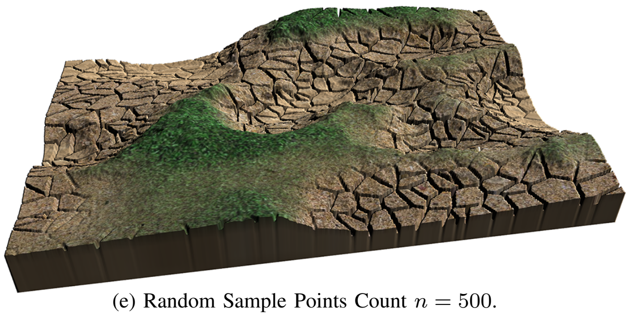
Fig. 4 shows the results of synthesizing cracked arid terrains with varying numbers of random sample points (from 100 to 800) to demonstrate the impact of sample density on the generated landscapes. (a) With $n = 100$ random sample points, the terrain exhibits a sparser distribution of cracks and features, resulting in a more simplistic, less-detailed arid landscape. (b) Doubling the random sample points to $n = 200$ in (b) adds more intricacy to the terrain, with finer cracks and patterns emerging. (c) At $n = 300$ in (c), the terrain becomes more complex, with a greater level of detail and

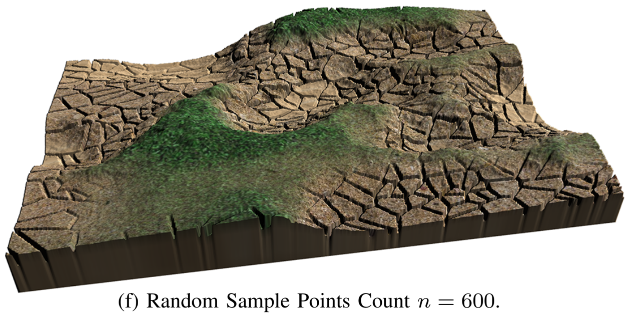

(a) Random Sample Points Count $n = 100$.

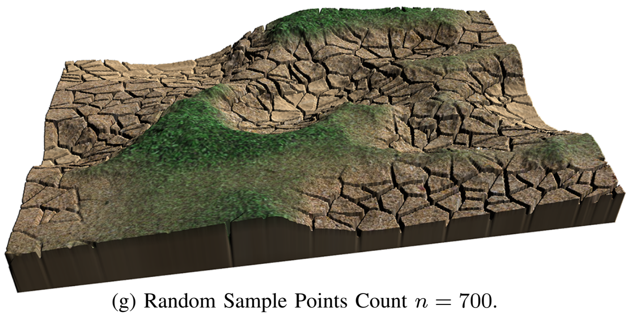

(b) Random Sample Points Count $n = 200$.


(c) Random Sample Points Count $n = 300$.


(d) Random Sample Points Count $n = 400$.


(e) Random Sample Points Count $n = 500$.


(f) Random Sample Points Count $n = 600$.


(g) Random Sample Points Count $n = 700$.

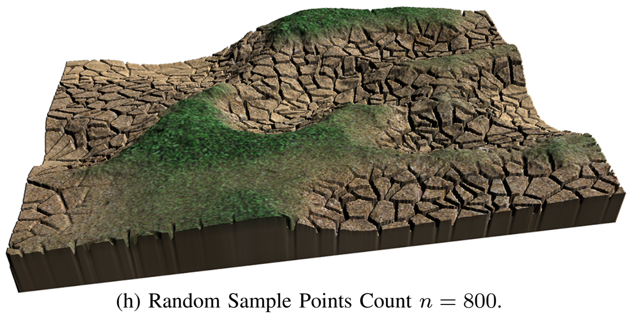
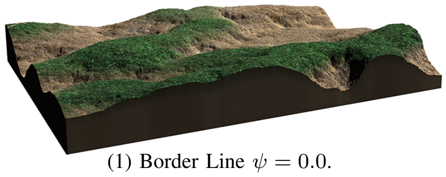

(h) Random Sample Points Count $n = 800$.

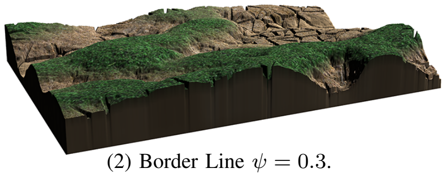
Fig. 4. Different Sample Count. These results are cracked arid terrains synthesized with different amounts of sample points.

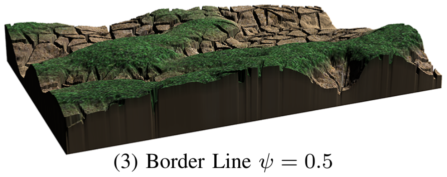
texture, mimicking the subtle intricacies of arid environments. (d) Further increasing the random sample points to $n = 400$ in (d) enhances the overall realism, portraying a dynamic arid landscape with diverse features. The trend continues, with the landscapes becoming increasingly sophisticated, showing a richer portrayal of cracked arid terrains as sample points count increases. These results underline the importance of sample density in achieving high-fidelity synthesis of arid terrains.

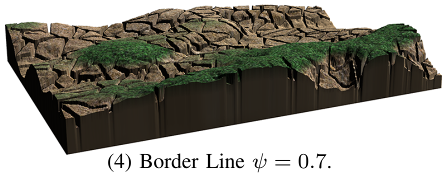
Fig. 5 shows the results of synthesizing cracked arid terrains across three distinct terrain subgroups - (a) Mountain Terrain, (b) Ridged Terrain, and (c) Cross Hill Terrain - with varying border line settings from 0.0 to 1.0 reveals a wide spectrum of outcomes. As shown in the results, lower border line settings such as 0.0 and 0.3 produce smoother terrains, more vegetation, with subtle surface variations, gradually transitioning to increasingly rugged and cracked landscapes as the border line setting progresses towards 1.0.

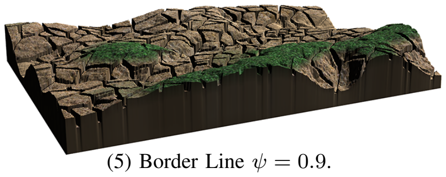
Fig. 6 shows the details of the synthesized cracked arid terrain from five different perspectives, each rendered from strategically placed cameras within the virtual landscape synthesized through our innovative approach. From various angles, it provides an in-depth exploration of the virtual terrain, showcasing the varying scales of cracked patterns, subtle elevation changes, and the interplay of light and shadow.

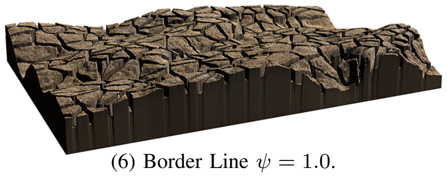

(1) Border Line $\psi = 0.0$.


(2) Border Line $\psi = 0.3$.

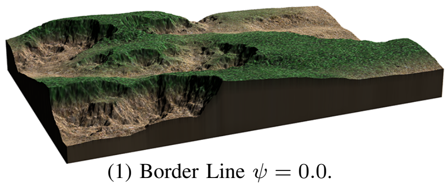

(3) Border Line $\psi = 0.5$

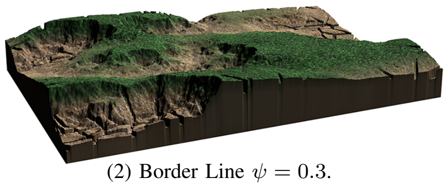

(4) Border Line $\psi = 0.7$.

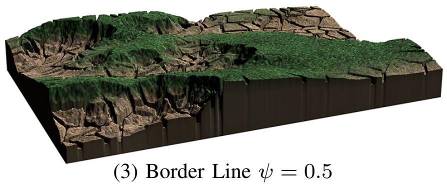

(5) Border Line $\psi = 0.9$.

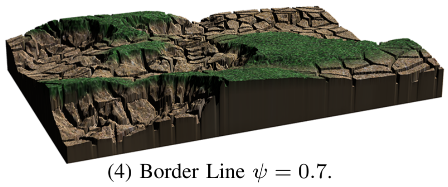

(6) Border Line $\psi = 1.0$.

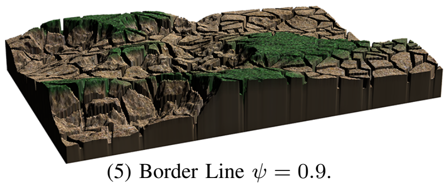
(a) Cracked Arid Mountain Terrain.

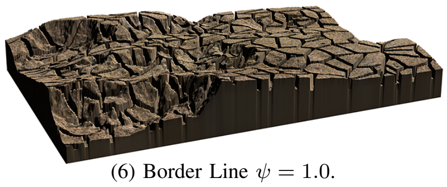

(1) Border Line $\psi = 0.0$.


(2) Border Line $\psi = 0.3$.

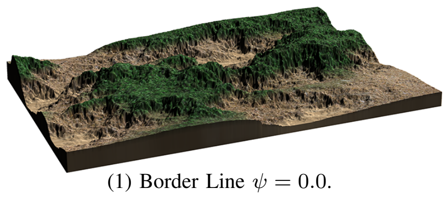

(3) Border Line $\psi = 0.5$

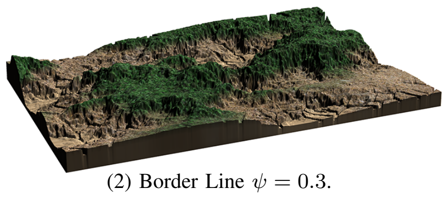

(4) Border Line $\psi = 0.7$.

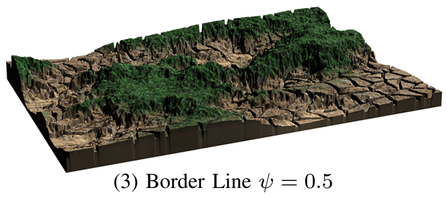

(5) Border Line $\psi = 0.9$.

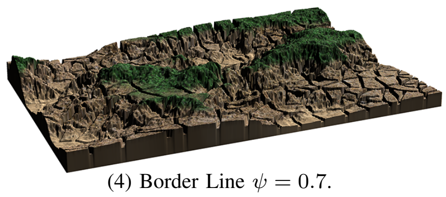

(6) Border Line $\psi = 1.0$.

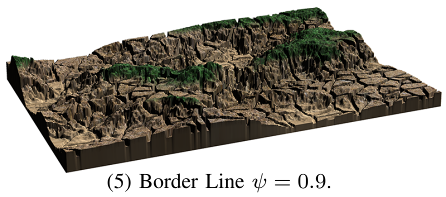
(b) Cracked Arid Ridged Terrain.

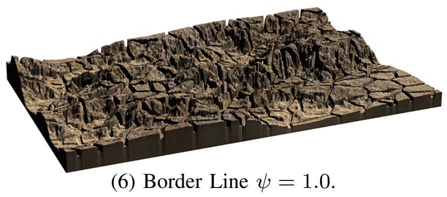

(1) Border Line $\psi = 0.0$.


(2) Border Line $\psi = 0.3$.


(3) Border Line $\psi = 0.5$


(4) Border Line $\psi = 0.7$.


(5) Border Line $\psi = 0.9$.


(6) Border Line $\psi = 1.0$.

(c) Cracked Arid Cross Hill Terrain.

Fig. 5. Different Border Line. These results are cracked arid terrains synthesized with different border line settings.

V. Conclusion

The creation of realistic virtual arid environments holds immense potential for various applications, including video games, simulations, environmental modeling, and geospatial analysis. In this paper, we introduce an innovative approach for synthesizing realistic cracked terrain for virtual arid terrain generation. We discuss the methodology employed, which involves terrain synthesis, crack pattern generation, and the incorporation of various environmental factors to enhance realism. Our approach allows for the generation of diverse arid terrains, from mountainous deserts to ridge-filled landscapes and more. Our approach offers a versatile tool for a wide range of applications. We present results demonstrating the versatility and fidelity of our approach, including variations in terrain types, sample point densities, and border line settings. By providing different parameter settings, our method has the potential to empower game designers to craft convincing virtual arid landscapes for gaming, simulations, and more. The potential impact of this technology on industries such as video games, and environmental modeling is significant, and it opens up exciting possibilities for the development of highly realistic and immersive virtual arid environments.

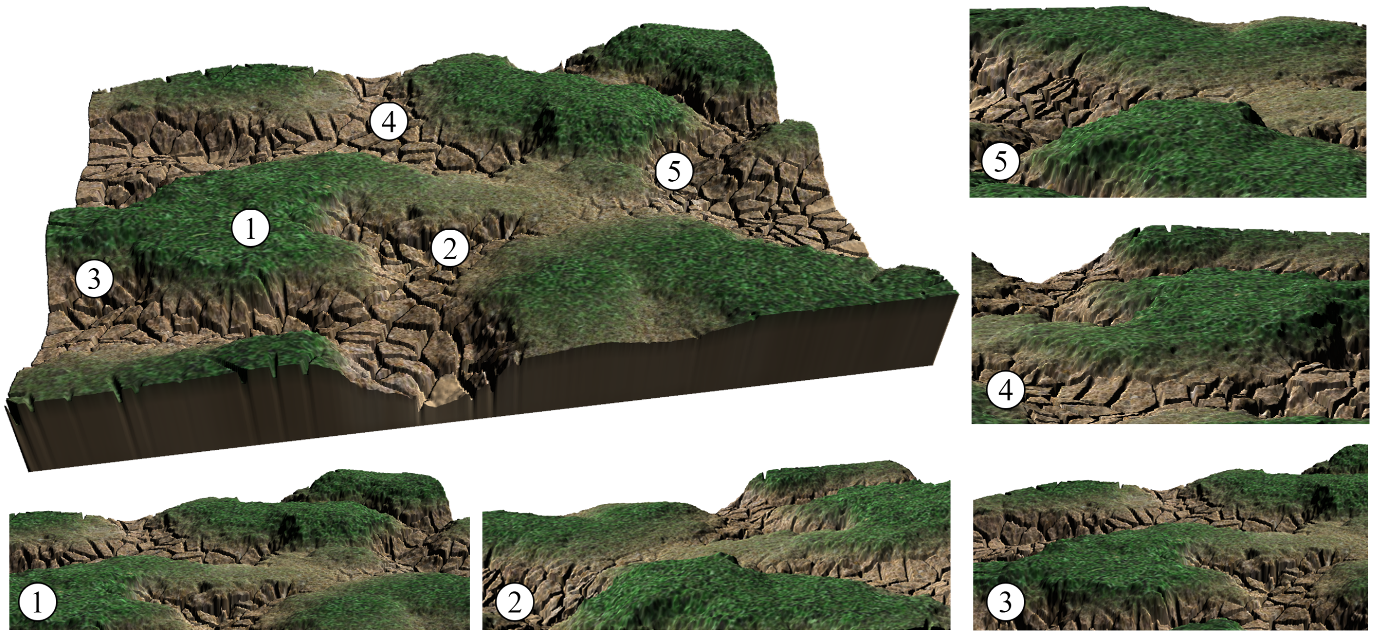


Fig. 6. Terrain Details. This figure shows the details of synthesized cracked arid terrain. Five different views are rendered from five cameras placed in the virtual cracked arid terrain that is synthesized with our approach.

References

[1] W. Li, "Patch-based monte carlo terrain upsampling via gaussian laplacian pyramids," in *Proceedings of the 2023 5th International Conference on Image, Video and Signal Processing*, 2023, pp. 172–177.

[2] ——, "Procedural modeling of the great barrier reef," in *Advances in Visual Computing: 16th International Symposium, ISVC 2021, Virtual Event, October 4-6, 2021, Proceedings, Part I*. Springer, 2021, pp. 381–391.

[3] ——, "Terrain synthesis for treadmill exergaming in virtual reality," in *2023 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*. IEEE, 2023, pp. 263–269.

[4] ——, "Procedural marine landscape synthesis for swimming exergame in virtual reality," in *2022 IEEE Games, Entertainment, Media Conference (GEM)*. IEEE, 2022, pp. 1–8.

[5] ——, "Elliptical4vr: An interactive exergame authoring tool for personalized elliptical workout experience in vr," in *Proceedings of the 2023 5th International Conference on Image, Video and Signal Processing*, 2023, pp. 111–116.

[6] ——, "Synthesizing procedural landscape for volcanic eruption simulation in virtual reality," in *2023 IEEE International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT2023)*. IEEE, 2023, pp. 1–8.

[7] T. J. Ong, R. Saunders, J. Keyser, and J. J. Leggett, "Terrain generation using genetic algorithms," in *Proceedings of the 7th annual conference on Genetic and evolutionary computation*, 2005, pp. 1463–1470.

[8] P. Walsh and P. Gade, "Terrain generation using an interactive genetic algorithm," in *IEEE Congress on Evolutionary Computation*. IEEE, 2010, pp. 1–7.

[9] K. R. Kamal and Y. S. Uddin, "Parametrically controlled terrain generation," in *Proceedings of the 5th international conference on Computer graphics and interactive techniques in Australia and Southeast Asia*, 2007, pp. 17–23.

[10] J. Gain, B. Merry, and P. Marais, "Parallel, realistic and controllable terrain synthesis," in *Computer Graphics Forum*, vol. 34, no. 2. Wiley Online Library, 2015, pp. 105–116.

[11] F. H. Razafindrazaka, "Delaunay triangulation algorithm and application to terrain generation," *International Institute for Software Technology, United Nations University, Macao*, 2009.

[13] J. Doran and I. Parberry, "Controlled procedural terrain generation using software agents," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 2, no. 2, pp. 111–119, 2010.

[12] H. Hnaidi, E. Guérin, S. Akkouche, A. Peytavie, and E. Galin, "Feature based terrain generation using diffusion equation," in *Computer Graphics Forum*, vol. 29, no. 7. Wiley Online Library, 2010, pp. 2179–2186.

[14] J.-D. Génevaux, É. Galin, E. Guérin, A. Peytavie, and B. Benes, "Terrain generation using procedural models based on hydrology," *ACM Transactions on Graphics (TOG)*, vol. 32, no. 4, pp. 1–13, 2013.

[15] A. Santamaría-Ibirika, X. Cantero, M. Salazar, J. Devesa, I. Santos, S. Huerta, and P. G. Bringas, "Procedural approach to volumetric terrain generation," *The Visual Computer*, vol. 30, no. 9, pp. 997–1007, 2014.

[16] G. Cordonnier, J. Braun, M.-P. Cani, B. Benes, E. Galin, A. Peytavie, and E. Guérin, "Large scale terrain generation from tectonic uplift and fluvial erosion," in *Computer Graphics Forum*, vol. 35, no. 2. Wiley Online Library, 2016, pp. 165–175.

[17] A. Asirvatham and H. Hoppe, "Terrain rendering using gpu-based geometry clipmaps," *GPU gems*, vol. 2, no. 2, pp. 27–46, 2005.

[18] A. Bernhardt, A. Maximo, L. Velho, H. Hnaidi, and M.-P. Cani, "Real-time terrain modeling using cpu-gpu coupled computation," in *2011 24th SIBGRAPI Conference on Graphics, Patterns and Images*. IEEE, 2011, pp. 64–71.

[19] É. Guérin, J. Digne, E. Galin, A. Peytavie, C. Wolf, B. Benes, and B. Martinez, "Interactive example-based terrain authoring with conditional generative adversarial networks." *ACM Trans. Graph.*, vol. 36, no. 6, pp. 228–1, 2017.

[20] r. r. spick and j. walker, "Realistic and textured terrain generation using gans," in *Proceedings of the 16th ACM SIGGRAPH European Conference on Visual Media Production*, 2019, pp. 1–10.

[21] V. Gorbatsevich, M. Melnichenko, and O. Vygolov, "Enhancing detail of 3d terrain models using gan," in *Modeling Aspects in Optical Metrology VII*, vol. 11057. SPIE, 2019, pp. 296–302.

[22] G. Voulgaris, I. Mademlis, and I. Pitas, "Procedural terrain generation using generative adversarial networks," in *2021 29th European Signal Processing Conference (EUSIPCO)*. IEEE, 2021, pp. 686–690.

[23] J. Kordež, M. Marolt, and C. Bohak, "Real-time interpolated rendering of terrain point cloud data," *Sensors*, vol. 23, no. 1, p. 72, 2022.

[24] E. Guérin, A. Peytavie, S. Masnou, J. Digne, B. Sauvage, J. Gain, and E. Galin, "Gradient terrain authoring," in *Computer Graphics Forum*, vol. 41, no. 2. Wiley Online Library, 2022, pp. 85–95.

[25] A. Bowyer, "Computing dirichlet tessellations," *The computer journal*, vol. 24, no. 2, pp. 162–166, 1981.

[26] D. F. Watson, "Computing the n-dimensional delaunay tessellation with application to voronoi polytopes," *The computer journal*, vol. 24, no. 2, pp. 167–172, 1981.