A Procedural Technique for 3D Cave Generation

João Vítor Farias Silva[1], **Rafael Piccin Torchelsen**[1,2]

[1]Centro de Desenvolvimento Tecnológico - Universidade Federal de Pelotas (UFPel)

[2]AKCIT Lab - Instituto de Informática - UFRGS

{jvfsilva,rafael.torchelsen}@inf.ufpel.edu.br

Abstract. Introduction: *Procedural Content Generation (PCG) has become an important approach for creating large-scale virtual environments in digital games. However, generating three-dimensional cave systems remains challenging due to their irregular structure and the need to balance automation with artistic control.* ***Objective:*** *This work aims to propose a procedural method capable of generating 3D cave environments that provide both structural control for designers and visual plausibility for real-time applications.* ***Methodology:*** *The proposed approach is based on a modular pipeline that uses spline-defined paths to guide the cave structure, mesh sweeping techniques to generate geometry, and noise-based deformation to introduce organic surface variation. Additionally, environmental elements such as rocks and stalactites are procedurally placed using spatial sampling strategies derived from the generated spline.* ***Results:*** *The results demonstrate that the method is capable of generating visually coherent and diverse cave environments suitable for integration into game engines. By producing static meshes, the technique ensures compatibility with standard rendering pipelines while allowing efficient iteration and designer-driven control.*

Keywords *Procedural Content Generation, Caves, Computer Graphics.*

1. Introduction

Modern computer graphics play a central role in the creation of virtual environments for digital games, simulations, and immersive applications. As digital games continue to grow in complexity and scale, the demand for increasingly detailed and diverse environments has intensified, increasing the need for efficient content creation techniques. Creating such environments manually is often time-consuming and requires significant human effort, especially for complex structures such as caves, which involve irregular geometries, interconnected tunnels, and detailed surface variation.

Procedural Content Generation (PCG) has emerged as an effective approach to automate the creation of complex virtual environments, reducing manual workload while enabling scalable content production [Fletcher et al. 2010]. PCG techniques have been successfully applied to terrain generation, urban environments, and vegetation [Zhou et al. 2007, Chen et al. 2008, Lluch et al. 2003]. However, the creation of more complex structures, such as three-dimensional caves, remains a challenging problem due to their irregular structure and non-linear spatial organization [Antoniuk 2023].

Supplementary video demonstrating the proposed method is available here (click to view).

This work proposes a procedural method for generating three-dimensional caves that balances automation and artistic control. The approach is based on the use of spline-defined paths to guide the main structure of the cave, combined with mesh generation and noise-based deformation to produce natural-looking surfaces. Additionally, the method allows designers to influence the overall shape and structure of the generated environment, enabling a collaborative workflow while benefiting from algorithmic generation.

The main contributions of this work are: (i) a spline-based procedural pipeline for 3D cave generation, (ii) a method for combining structural control with geometric variation, and (iii) a practical approach that supports designer-driven content creation while maintaining procedural efficiency.

2. Related Work

Procedural cave generation has been explored through a variety of approaches, each emphasizing different aspects such as structural generation, realism, efficiency, and artistic control. One class of methods focuses on generating the main cave structure through rule-based or emergent systems. For instance, L-systems have been used to generate branching cave layouts by recursively expanding symbolic rules, producing complex tunnel networks from simple definitions [Mark et al. 2015]. Similarly, cellular automata (CA) are applied to generate organic cave-like structures through local interactions between grid cells [Johnson et al. 2010, Antoniuk 2023]. While these approaches are effective in producing complex patterns, they are typically limited to two-dimensional representations or offer limited direct control over the final structure.

Another line of work is based on physically inspired or simulation-driven methods. These approaches model geological processes such as erosion and water flow to generate cave systems with high geological fidelity [Boggus e Crawfis 2006, Franke e Müller 2021]. Although they produce plausible formations, they often provide limited artistic control, as the final result depends on indirect parameter tuning rather than explicit design input.

Noise-based methods and voxel representations are also commonly used to generate volumetric cave structures. These techniques treat the environment as a continuous scalar field and extract geometry using thresholding and surface reconstruction algorithms [Cui et al. 2011]. While they are effective in generating coherent 3D volumes, they typically lack explicit control over the shape and layout of interconnected tunnels, making it difficult for designers to guide the overall structure.

In addition to structural generation, an important aspect of procedural cave modeling is the level of creative control provided to the user. Existing approaches range from fully automatic systems, where the designer has minimal influence over the outcome [Mark et al. 2015, Cui et al. 2011], to more interactive methods that allow user-defined sketches or layouts to guide the generation process [Antoniuk 2023]. Sketch-based approaches enable a higher degree of artistic control by allowing designers to define the main structure explicitly, while the algorithm does the heavy lifting.

Another relevant aspect is the representation of cave data. Volumetric approaches based on voxels are widely used due to their flexibility [Mark et al. 2015, Cui et al. 2011, Franke e Müller 2021], but they often come with high memory costs. Alternative representations, such as point clouds, can efficiently describe structural layouts but require additional processing to generate renderable geometry [Antoniuk 2023]. Hybrid approaches combine volumetric modeling with surface extraction techniques to balance performance and visual quality [Franke e Müller 2021].

From the perspective of Procedural Content Generation (PCG) taxonomy, the proposed approach belongs to the constructive category. Constructive PCG methods generate content directly through predefined rules and algorithms without requiring iterative search or optimization. In our approach, the cave geometry is deterministically produced from a spline-guided pipeline. This enables efficient generation while providing direct artistic control over the resulting cave layout.

Despite the diversity of existing methods, a gap remains between fully automated approaches and designer-driven workflows. Many techniques either prioritize realism at the cost of control or provide flexibility without generating coherent three-dimensional structures. Furthermore, the integration of environmental elements is often treated as a secondary step or not addressed at all [Cui et al. 2011, Antoniuk 2023]. Table 1 summarizes the main characteristics of the reviewed approaches.

Table 1. Comparison of related works on procedural cave generation

Work	**Structure Generation**	**Data Representation**	**Artistic Control**	**Elements**	**Main Focus**
Mark et al. (2015)	L-System	Voxels + Marching Cubes	Low	3D noise for stalactites and stalagmites	General cave structure
Johnson et al. (2010)	Cellular Automata (CA)	2D Map	Medium	Not addressed	Real-time generation
Cui et al. (2011)	3D Perlin Noise	Octree (Voxels)	Low	Not addressed	Memory cost reduction
Boggus and Crawfis (2006)	Geological simulation	3D volume from 2D map	Low	Not addressed	Geological realism
Antoniuk (2023)	Schematic maps + CA	3D Point Cloud	High	Not addressed	Multi-level creative control
Franke and Müller (2021)	Natural simulation + 3D noise	Voxels + Marching Cubes	Medium	Detailed speleothem modeling	Visual and geological realism
Proposed Work	Spline (Hermite)	Polygonal Mesh	High	Intersection rays on surface and ceiling	General cave structure and artistic freedom

3. Methodology

The proposed pipeline was designed to satisfy three objectives: (1) provide designers with direct artistic control over the generated cave layout, (2) produce visually plausible cave environments suitable for digital games, and (3) maintain compatibility with real-time rendering while supporting the procedural placement of environmental elements.

These objectives guided the design of each stage of the proposed pipeline. The pipeline was implemented using Unreal Engine 5 and its Geometry Scripting framework, which provides the procedural mesh processing operations.

The pipeline consists of five main stages: (i) spline-based trajectory definition, (ii) mesh generation via sweep, (iii) noise-based surface deformation, (iv) procedural placement of environmental elements, and (v) exposure of user-controlled parameters. An overview of the pipeline is shown in Figure 1.

Figure 1. Overview of the proposed procedural pipeline

3.1. Spline-Based Structure Definition

The structural backbone of the cave is defined using a single three-dimensional spline. The proposed method employs cubic Hermite splines, which allow explicit control over positions and tangent vectors at control points. Hermite splines were chosen because they provide direct control over both control-point positions and tangent vectors, enabling intuitive editing of the cave trajectory. In contrast, Bézier and B-spline representations offer smoother continuity but less direct local control, making them less suitable for the proposed designer-driven workflow. Hermite splines allow explicit control over positions and tangent vectors at control points. The position along each spline segment is evaluated using the standard cubic Hermite interpolation:

$$\mathbf{P}(t) = h_{00}(t)\mathbf{P}_0 + h_{10}(t)\mathbf{T}_0 + h_{01}(t)\mathbf{P}_1 + h_{11}(t)\mathbf{T}_1, \tag{1}$$

where $t \in [0, 1]$, $\mathbf{P}_0$ and $\mathbf{P}_1$ denote the segment endpoints, $\mathbf{T}_0$ and $\mathbf{T}_1$ are their associated tangent vectors, and the cubic Hermite basis functions are given by

$$\begin{aligned} h_{00}(t) &= 2t^3 - 3t^2 + 1, \\ h_{10}(t) &= t^3 - 2t^2 + t, \\ h_{01}(t) &= -2t^3 + 3t^2, \\ h_{11}(t) &= t^3 - t^2. \end{aligned} \tag{2}$$

This ensures that the generated curve interpolates user-defined points while maintaining intuitive manipulation. In practice, the spline is constructed by defining a sequence of control points each associated with a tangent vector where each curve segment between the points is evaluated using the cubic Hermite interpolation formulation. The tangents can be manually specified or derived from neighboring points to ensure smooth transitions along the curve.

The spline defines the central path of the cave and is sampled at regular intervals to extract position, orientation, and scale. These samples guide all subsequent stages. Sampling density directly affects geometric smoothness and computational cost, allowing a trade-off between performance and visual quality.

Additionally, the spline representation acts as a compact and flexible descriptor of the cave layout, enabling rapid iteration over different structural configurations without modifying the underlying generation pipeline. By adjusting control points and tangents, designers can produce a wide range of tunnel profiles, from smooth, elongated passages to more abrupt directional changes. This parametric control also supports the creation of visually plausible cave patterns, approximating natural formations while avoiding the need for complex geological simulation. As a result, the spline serves not only as a geometric guide but also as a key element for balancing artistic control and procedural variability.

3.2. Mesh Generation via Sweep

A 3D mesh is generated by sweeping a circular cross-section along the spline. The circular profile was chosen due to its geometric stability and ability to avoid self-intersection artifacts observed in alternative shapes. The spline is first transformed into a sequence of oriented sample points, each defined by a position, a rotation, and a local scale. These oriented points effectively encode the spatial configuration of the curve and serve as guides for the sweep operation. At each sample, the predefined two-dimensional circular profile is positioned, rotated, and scaled according to these parameters.

The final mesh is constructed by connecting consecutive profiles, forming a continuous tubular structure aligned with the spline. Local scaling variations allow for the generation of tunnels with different widths. Although the cave is defined by a single three-dimensional spline, bifurcation-like structures can be produced by applying boolean subtraction operations at regions where different portions of the tunnel intersect. An internal tunnel mesh is used as the subtraction volume to carve openings between intersecting tunnel segments, creating connected passages while preserving the underlying single-spline representation. This sweep-based approach provides a consistent mapping between the abstract spline representation and the resulting geometry, ensuring that structural changes in the curve are directly reflected in the mesh. By leveraging parametric sampling, the method maintains coherence even in regions with high curvature, reducing visual artifacts and preserving surface continuity.

The use of mesh-based representation also enables compatibility with standard real-time rendering pipelines, supporting operations such as UV mapping and efficient shading. The generated mesh also includes automatically generated UV coordinates for texture mapping. Since UV generation is intended primarily for visualization and is not a core component of the proposed pipeline, the resulting parameterization may introduce stretching in highly curved regions and discontinuities near boolean-generated intersections. These artifacts are discussed as limitations of the current implementation. Furthermore, the combination of controlled scaling and boolean operations allows for the generation of more complex cave layouts, including intersecting tunnels, while maintaining geometric consistency across the structure. Figure 2 illustrates the sweep process, in yellow the cubic hermite spline and in green the generated mesh.

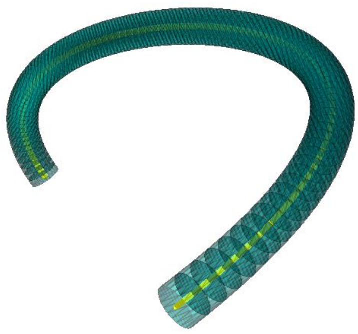


Figure 2. Sweep-based mesh generation along a spline

3.3. Noise-Based Surface Deformation

To reduce geometric uniformity, surface deformation is applied using noise functions. The Perlin noise is used due to its ability to generate smooth and spatially coherent variations.

The deformation is applied directly to mesh vertices by displacing them along their respective surface normals. For each vertex v, a noise value is sampled based on its position in three-dimensional space, and this value is used to offset the vertex in the normal direction. Since the normal direction is fixed per vertex, the displacement preserves the overall structure of the mesh while introducing localized surface variation. The noise amplitude controls the maximum displacement magnitude, determining how far vertices are moved from their original positions, while the frequency controls the scale of the features generated across the surface.

This step significantly improves visual realism by introducing irregularities commonly observed in natural cave formations. Figure 3 presents an example of the deformation process.

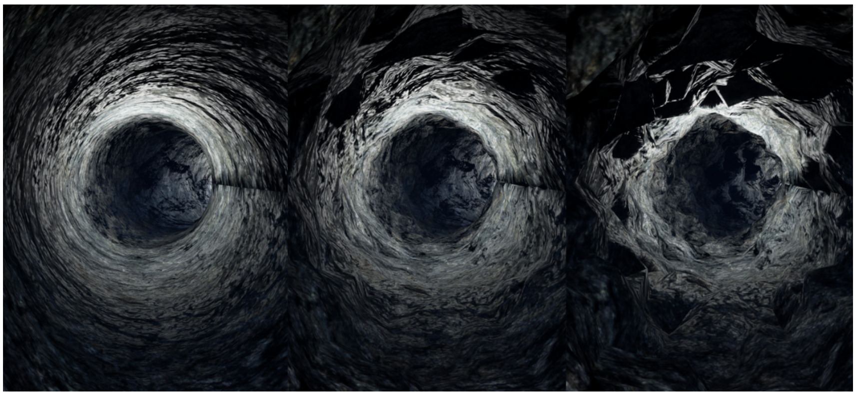


Figure 3. Surface deformation using noise

By operating in three-dimensional space, the noise function ensures continuity across the surface, avoiding discontinuities that could arise from planar or texture-based approaches. This contributes to a more consistent appearance, especially in regions with complex curvature.

Additionally, the separation between structural definition (spline and sweep) and surface detailing (noise) allows independent control over large-scale shape and fine-grained features. This modularity makes the method more flexible, enabling different visual styles without altering the underlying geometry. As a result, the technique achieves a balance between computational efficiency and visual richness, which is particularly relevant for real-time applications.

3.4. Procedural Placement of Environmental Elements

Environmental elements such as rocks and stalactites are procedurally placed to enhance visual richness. Placement is guided by sampling points along the spline. From each sample, additional candidate positions are generated with small horizontal offsets, bounded by the radius of the circular cross-section, ensuring that spawning occurs within the cave interior rather than strictly along the central path.

From these positions, rays are cast both upward and downward. When a ray intersects the cave mesh, the hit position and corresponding surface normal are retrieved. These values are then used to place and orient environmental elements, aligning them with the local geometry. Upward ray intersections (ceiling) are typically used to spawn suspended elements, while downward intersections (floor) are used for ground-based elements. However, the system is not restricted to specific asset types, allowing different categories of objects to be assigned to each surface depending on the desired visual style. Random variations in scale, rotation, and positional offset are applied to avoid repetitive patterns, while parameters such as density and spacing provide additional control over distribution. Figure 4 illustrates the rays cast along the path defined by the spline. Rays shown in red indicate successful intersections with the tunnel surface, where procedural environmental elements are placed.

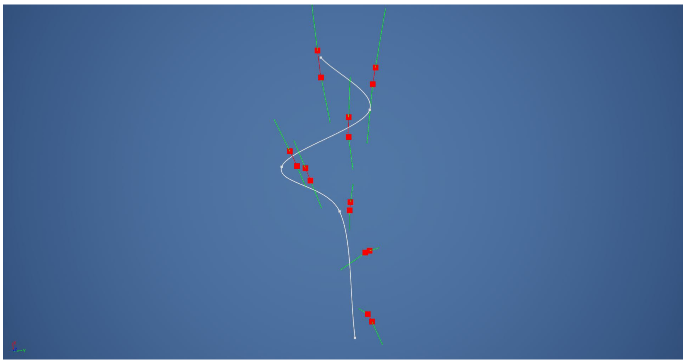


Figure 4. Example of rays casted along the spline

By relying on ray-based intersection tests, the method adapts naturally to local geometric variations introduced during the deformation stage, ensuring that elements remain properly aligned with irregular surfaces. This avoids common artifacts such as floating or intersecting objects, which can break visual immersion. Furthermore, the decoupling between placement logic and asset definition allows the system to support different types of environmental elements without modifying the core algorithm. This makes the approach extensible and suitable for a variety of cave styles and thematic settings. As a result, the technique enhances scene complexity while preserving structural coherence and real-time performance.

3.5. Parameter Exposure

To support designer-driven workflows, the system exposes a set of parameters that control all stages of the pipeline. These parameters are organized according to their influence on structure, geometric detail, and environmental composition. Table 2 presents the parameters with their effects and default values.

Table 2. Summary of the main parameters exposed by the proposed pipeline.

Parameter	Effect	Default
Circular Resolution	Defines the number of vertices used to construct the circular cross-section. Higher values produce a smoother tunnel profile.	32
Spline Subdivisions	Determines the number of samples generated between consecutive spline control points, controlling mesh resolution along the cave.	16
Outer Radius	Defines the outer radius of the tunnel profile. Can be adjusted globally or locally along the spline.	300
Inner Radius	Defines the inner radius of the circular profile.	250
Noise Amplitude	Controls the magnitude of the surface deformation. Higher values produce more pronounced irregularities.	1.0
Noise Scale	Controls the spatial frequency of the noise. Lower values generate broader features, while higher values create finer surface details.	0
Noise Seed	Defines the seed of the noise function, allowing reproducible cave geometries.	0
Apply Intersections	Performs boolean subtraction to remove overlapping tunnel regions. Recommended only after the cave geometry has been finalized due to its computational cost.	False
Object Spacing	Defines the spacing between procedurally placed environmental elements. Lower values increase object density.	200
Ground Scale (Min–Max)	Specifies the minimum and maximum scale of objects placed on the cave floor.	0.8-1.5
Ceiling Scale (Min–Max)	Specifies the minimum and maximum scale of objects placed on the cave ceiling.	0.8-1.5

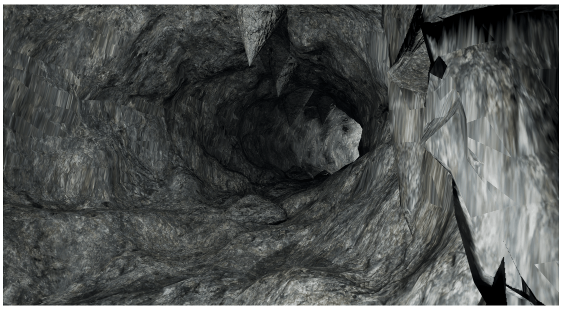
4. Results

This section presents the results of the proposed procedural cave generation technique, focusing on its applicability in digital games. Rather than geological accuracy, the evaluation emphasizes visual quality, structural coherence, and integration into real-time rendering pipelines, showing how the method efficiently generates caves aligned with the standards of modern games.

Although the technique does not strictly replicate geological processes, the generated structures exhibit visual characteristics common in contemporary game environments. The combination of spline-guided structure, mesh sweeping, and noise-based deformation produces tunnels with organic variation, smooth curvature, and controlled irregularities.

The visual style was inspired by modern games such as Elden Ring [FromSoftware 2022] and The Witcher 3 [CD Projekt Red 2015], in which caves present smooth yet irregular paths, diameter variation along the trajectory, branching passages, and surface detail that avoids artificial repetition. The technique reproduces these patterns by letting designers control the main trajectory through splines while delegating surface complexity to procedural deformation, yielding environments that feel designed rather than purely random while retaining procedural richness.

Moreover, tunnel intersections enable more complex layouts, similar to the multi-path cave systems used in level design to encourage exploration and player choice. The examples below, produced with different parameter configurations, demonstrate the full integration of all stages — spline definition, mesh generation, noise deformation, and environmental element placement — and the system's flexibility.



(a) Inner part of the cave generated by the technique proposed in this work

(b) A section of the cave with an intersection just ahead offering two possible paths

Figure 5. Examples generated using the proposed pipeline.

(a) A section illustrating three optional paths within the cave

(b) A passage with a straight path and an intersection to the right

Figure 6. Additional cave passages with branching paths.

Table 3. Qualitative evaluation of the proposed method.

Design Goal	Evidence
Artistic control	Spline editing, local/global radius control, noise parameters, and procedural placement provide direct designer control.
Visual plausibility	Sweep generation and noise deformation create organic tunnels with varying diameters and natural-looking surface irregularities.
Real-time suitability	Polygonal meshes generated in Unreal Engine integrate with real-time rendering and procedural environmental placement.

4.1. Limitations

The surface deformation may introduce mesh artifacts under high noise values near tunnel intersections where boolean subtraction creates complex topology.

UV coordinates are generated by cylinder projection, intended for visualization rather than as a contribution of this work; although effective for most tunnel segments, it may cause texture stretching in highly curved regions and discontinuities near boolean-generated intersections.

Boolean subtraction may also generate small disconnected fragments, removed by filtering connected components below a predefined size threshold, though complex intersections may still exhibit minor geometric artifacts. Finally, the method represents the cave using a single spline: boolean operations create the appearance of branching passages, but true topological branching through multiple connected splines remains future work.

5. Conclusions

This work presented a procedural technique for generating three-dimensional cave systems with artistic control and structural consistency. The results demonstrate that it is possible to produce visually plausible tunnel networks suitable for digital games and interactive simulations while allowing designers to guide the generation process. The proposed approach proved capable of generating diverse environments while maintaining coherence and stability. Although the method does not aim for strict geological accuracy, it achieves convincing visual results that approximate natural cave formations. Additionally, the modular design of the technique enables flexibility and facilitates its integration into different applications and pipelines.

Future work includes improving UV parameterization, supporting true branching cave topologies through multiple connected splines, reducing artifacts introduced by boolean operations, and expanding the evaluation through user studies and additional comparisons with existing procedural cave generation techniques.

6. Acknowledgments

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001, and partially by CNPq–Brazil (grant 421962/2023-2), and by the Advanced Knowledge Center in Immersive Technologies (AKCIT), with financial resources from the PPI IoT/Manufatura 4.0 / PPI HardwareBR of the MCTI grant number 057/2023, signed with EMBRAPII.

References

Antoniuk, I. (2023). Generating layout for complex cave-like levels with schematic maps and cellular automata. *Machine Graphics and Vision*, 32:45–65.

Boggus, M. e Crawfis, R. (2006). Procedural creation of 3D solution cave models. In *Proceedings of the IASTED International Conference on Modelling, Simulation and Optimization*, Banff. IASTED.

CD Projekt Red (2015). The witcher 3: Wild hunt. CD Projekt.

Chen, G., Esch, G., Wonka, P., Mueller, P., e Zhang, E. (2008). Interactive procedural street modeling. *ACM Transactions on Graphics*, 27(3):1–10.

Cui, J., Chow, Y.-W., e Zhang, M. (2011). Procedural generation of 3D cave models with stalactites and stalagmites. *International Journal of Computer Science and Network Security*, 11.

Fletcher, D., Yue, Y., e Kader, M. A. (2010). Challenges and perspectives of procedural modelling and effects. In *Proceedings of the International Conference on Information Visualisation*, pages 543–550, Washington. IEEE Computer Society.

Franke, K. e Müller, H. (2021). Procedural generation of 3D karst caves with speleothems. *Computers and Graphics*, 102.

FromSoftware (2022). Elden ring. Bandai Namco Entertainment.

Johnson, L., Yannakakis, G. N., e Togelius, J. (2010). Cellular automata for real-time generation of infinite cave levels. In *Proceedings of the 2010 Workshop on Procedural Content Generation in Games*, New York, NY, USA. ACM.

Lluch, J., Camahort, E., e Vivó, R. (2003). Procedural multiresolution for plant and tree rendering. In *Proceedings of the 2nd International Conference on Computer Graphics, Virtual Reality, Visualisation and Interaction in Africa*, AFRIGRAPH '03, page 31–38, New York, NY, USA. Association for Computing Machinery.

Mark, B., Berechet, T., Mahlmann, T., e Togelius, J. (2015). Procedural generation of 3D caves for games on the GPU. In *Proceedings of the 10th International Conference on the Foundations of Digital Games (FDG 2015)*, Pacific Grove. Society for the Advancement of the Science of Digital Games.

Zhou, H., Sun, J., Turk, G., e Rehg, J. M. (2007). Terrain synthesis from digital elevation models. *IEEE Transactions on Visualization and Computer Graphics*, 13(4):834–848.