

# Synthesizing Geologically Coherent Cave Networks

A. Paris<sup>1</sup>, E. Guérin<sup>1</sup>, A. Peytavie<sup>1</sup>, P. Collon<sup>2</sup>, and E. Galin<sup>1</sup>

<sup>1</sup>CNRS, Université de Lyon, LIRIS, France

<sup>2</sup>Université de Lorraine, CNRS, GeoRessources, F-54000 Nancy, France

## Abstract

We present a geologically-based method to generate complex karstic networks. Karsts are a type of landscape formed by the dissolution of highly soluble rocks (generally limestones). In particular, they are characterized by complex underground networks made of varieties of tunnels and breakout chambers with stalagmites and stalactites. Our method computes skeletons of karstic networks by using a gridless anisotropic shortest path algorithm according to field data of the underground system (such as inlets and outlets), geomorphological features and parameters such as faults, inception horizons, fractures, and permeability contrasts. From this skeleton, we define the geometry of the conduits as a signed distance function construction tree combining primitives with blending and warping operators. Our framework provides multiple levels of control, allowing us to author both the structure of the karstic network and the geometric cross-section shapes and details of the generated conduits.

**Keywords:** Procedural modeling, implicit surfaces, karstic networks, caves

## 1. Introduction

Three-dimensional landforms such as arches, overhangs, and caves are distinctive visual elements of virtual terrains. Despite the wide application of artificial terrains in the entertainment industry as well as in geological studies, modeling complex realistic landforms remains challenging. Most existing techniques focus on elevation terrains and do not allow the creation of volumetric features. Even though recent techniques address the modeling of volumetric landforms [PGMG09, BKRE19, PGP\*19], they do not lend themselves for the creation of karstic networks consistent with geological information such as inception features or permeability. Karstic systems are underground networks composed of conduits and caves that have grown by the dissolution of the host bedrock, generally limestone. Although karstic landscapes cover around 12% of the planet's continental surface [HGW\*14], modeling cave networks has not received much attention from the computer graphics community. Their volumetric nature combined with the winding paths and complex geometry of the underground conduits makes it a challenging problem.

The difficulties stem from the fact that karstic networks are formed by multiple, interconnected geological processes (rock fracturing, percolation) operating at different time scales (from a few years to hundreds of thousands of years) and resulting from various geological settings (fracture distributions, inception features) and hydrogeological time-varying conditions.

Karstic networks can be interpreted at multiple spatial scales: large scale refers to the global topology of the network, whereas local scale (or mesoscale) refers to the shape of the conduits.

Simulating these processes is therefore immense, largely under-constrained from a geological point-of-view, and would not allow for interactive control.



**Figure 1:** Given an input relief, geological characteristics, and user-defined key points, our method generates complex karstic networks and the corresponding detailed geometric model.

In this paper, we propose a geologically-based framework for modeling karstic networks. We focus on user-control and the generation of realistic and detailed conduit shapes. Given an input relief, we automatically compute a three-dimensional geometric graph connecting control points corresponding to sinks (inlets), springs (outlets), and known passages inside the bedrock. The skeleton of the karstic network is constructed by applying a gridless anisotropic shortest path taking into account geological parameters such as the permeability contrasts of the bedrock, fracture orientations, and inception horizons (geological surfaces particularly prone to karstification [FJT09]). Around this skeleton, a detailed volumetric geometry of the conduits is defined as a signed distance function construction tree combining volumetric primitives with Boolean, warp-

ing, and blending operators. Our framework provides multiple levels of control. The user may define inlets, outlets, and waypoints as passages to constrain the construction of the karstic network, prescribe the different volumetric geological parameters, and adjust the paths of the tunnels inside the bedrock.

The main contributions are as follows: 1) A geological framework for modeling karstic systems, taking into account known inlets, outlets, and underground passages, as well as parameters such as inception horizons locations, fracture orientations, and bedrock permeability contrasts, 2) An implicit modeling approach with new primitives and operators organized in a construction tree for creating the geometry and details of the caves around the simulated skeletons and 3) An interactive authoring framework providing multiple direct controls to the user for tuning the network as well as the final geometry of the cave. We compared the results with real data and show that generated karstic systems accord with geological observations.

## 2. Related work

Standard and widespread digital elevation models are limited to terrains without overhangs, arches, or caves. Here we review data structures and terrain generation algorithms that can deal with volumetric features. For more complete coverage on terrain modeling, the reader is referred to [GGP\*19] and to [NLP\*13] for subsurface geology specifically.

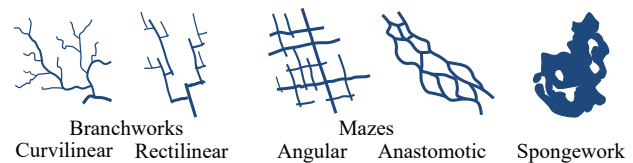
**Volumetric terrain modeling** Volumetric representations can be classified into three categories: voxel-based models, function-based approaches, and hybrid representations. Voxels offer a way to describe volumetric terrains but are memory demanding because of the underlying explicit spatial enumeration. They were employed for modeling landforms such as hoodoos and goblins [BFO\*07, JFBB10] or arches and overhangs [BKRE19]. Hybrid models, inspired by layered-material representations, exploit vertical run-length encoding of the different layer stacks to compress layers into intervals of identical material [PGMG09]. The terrain is defined as a continuous implicit surface whose field function is computed using a convolution on the discrete layer stacks. Function representations define the terrain as a compact implicit surface model whose field function can be constructed by combining primitives organized into a construction tree [PGP\*19]. Elevation models can be augmented with landforms such as arches, overhangs and caves, which may be either sculpted interactively or procedurally generated using invasion percolation. However, this approach lacks user control and underground tunnels obtained with noise-based sphere primitives do not exhibit realistic karstic shapes.

A few methods focus on the creation of caves or their components, such as procedurally placed stalagmites and stalactites using an octree [CCZ11], however, the global karstic network as well as the shape of the conduits are not produced. Pytel *et al.* proposed a two-stage simulation pipeline for the modeling of karstic networks [PM15]. While they account for geological parameters such as rock porosity, the resulting networks are modeled by a set of cubes and no solution is proposed for synthesizing the detailed geometry of the tunnel. L-systems have been used to create karstic networks

that are later augmented with objects such as stalactites and high-frequency details with noise [MBMT15]. This approach generates caves with insufficient precision due to the limited underlying voxel grid resolution, only provides indirect control, and does not account for the geological characteristics of the terrain. To the best of our knowledge, there exists no method for generating and authoring volumetric karstic networks interactively and efficiently modeling detailed conduit shapes, which is the focus of this paper.

**Geomorphology** Karsts are an ongoing subject of research in geomorphology and hydrology. Concerning numerical modeling, motivated by a better assessment of the associated underground flow estimation uncertainties, stochastic simulation of unexplored karstic networks has focused on various research for a decade. We can classify the developed methods in two families: the ones generating network skeletons (large scale), and the ones focusing on the conduit shape modeling (local scale).

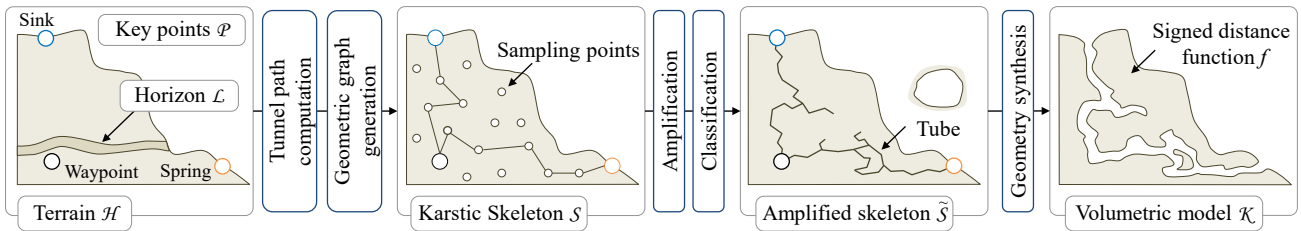
At a large scale, karstic networks feature a variety of different patterns. Palmer *et al.* proposed a classification of common patterns found in solutional caves [Pal03] such as curvilinear and rectilinear branchworks, anastomotic and angular mazes, spongework, or ramiform caves (Figure 2). The two last types are more specific to hypogenic caves (aggressive fluid coming from depth) and are not skeleton networks descriptors as their main characteristic is large dissolutional chambers.



**Figure 2:** Top-view classification of archetypes of cave patterns as proposed by [Pal03].

Recently, [JVA\*17] adapted this classification to updated speleogenetical observations including the dimensions and shape of caves. When only considering skeletons of the karstic networks, we can keep the following criteria: 1) real systems range between elongated hierarchical branchwork systems and more anarchic anastomotic mazes; 2) depending on the rock fracturing degree, these systems show rectilinear (more or less aligned with the fracture orientations) or curvilinear conduits; 3) in the vadose zone (above the phreatic level), conduits have a vertical-dominant development. Several approaches have then been proposed to simulate these different types of cave networks based on geostatistical methods [PIDC\*12, VJF\*14], anisotropic shortest paths [BRJ12, CHP12] or more recently percolation clusters [HR16]. Most of them construct the network skeletons using a pre-defined grid, which considerably slows down the simulation process when dealing with large systems and generates stair-step conduits due to aliasing when not aligned with the grid edges.

At a smaller scale, conduits often feature shape variations such as abrupt narrowing or enlargements, which are known to play a key role in fluid flows [FP00]. With the generic Object-Distance Simulation Method, [HCC10] proposed to model conduit envelopes by



**Figure 3:** Given an input terrain  $\mathcal{H}$ , geological parameters such as inception horizons  $\mathcal{L}$  and control key points  $\mathcal{P}$ , we synthesize the skeleton of the karstic system  $\tilde{\mathcal{S}}$  using an anisotropic shortest path operating on a 3D graph combined with a  $\gamma$ -skeleton geometric graph construction. We then convert the skeleton in a karstic volumetric terrain model  $\mathcal{K}$  defined as a procedural signed distance function obtained by combining implicit primitives with blending operators.

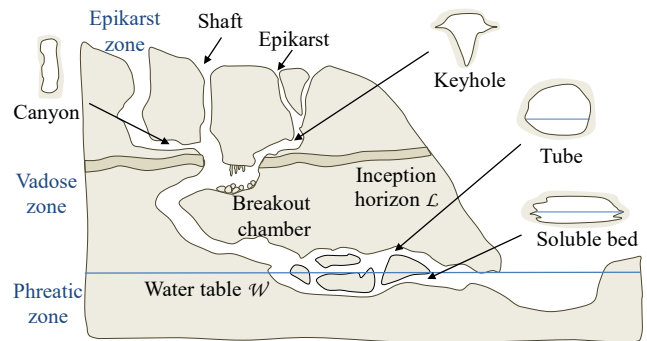
combining an Euclidean distance field around a skeleton with a random threshold. The resulting cross-section is irregular at small-scale but globally cylindrical at the mesoscale. [RCDF14] described various shapes of karstic conduits and builds on the previous method to account for perturbations along weakness planes such as fractures, inception horizons, or faults, allowing one to generate a larger diversity of conduits. Few works focus on the tunnel shapes because their reconstruction is currently limited by the acquisition process: the captured data by speleologists is often sparse, incomplete, or incorrect, which makes reconstruction difficult.

Collon *et al.* statistically studied several metrics such as conduit tortuosity, conduit orientations, or average vertex degree and average shortest path lengths of the network graph over a variety of real karstic networks [CBVR17]. They identified several important parameters for the description and comparison of actual natural systems. We take inspiration from this work and use geomorphological parameters as key user-controls in our authoring system, and rely on these parameters to analyze simulated networks.

### 3. Overview

Karstic systems are underground networks made of conduits and caves that have grown by the dissolution of the host rocks, generally limestone. In epigenic systems, the most common and documented ones, meteoric water penetrates the ground through diffuse infiltration or point sources of recharge. During this process, the water progressively carves different tunnel shapes such as canyon, keyhole, or tube passages based on different geological and physical conditions such as permeability, pressure, and water velocity.

The water-table  $\mathcal{W}$  position delimits the upper unsaturated zone (also called vadose), and the lower saturated zone (also called phreatic). The phreatic zone is generally directly connected to a spring (or the base-level), which constitutes the output of the karstic network. Note that some systems have several connected outputs. In the vadose zone, the conduits develop preferentially vertically along fractures, with canyon passages linking them along inception horizons (Figure 4). When approaching the phreatic zone, the development becomes progressively more horizontal-dominant. The conduits develop preferentially at or below the water-table in case of long-time steady-state base-level [JVA\*17]. If the water-table fluctuates highly with seasons, the networks develop more in maze



**Figure 4:** Idealized cross-section through a complex underground cave network.

patterns. As the base-level can change over geological times, today networks can show several levels of horizontal-dominant drains that witness the past water-table positions.

Simulating these complex natural phenomena over time remains challenging, computationally intensive, and would not allow for effective user-control. Instead, we use a procedural approach with an anisotropic shortest path algorithm taking into account a variety of geological constraints to create complex karstic networks.

The geological constraints and key points  $\mathcal{P}$  corresponding to sinks (inlets), springs (outlets), and known passages inside the bedrock (waypoints) are at the heart of our procedural approach (Figure 3). We focus on the generation of a geometric graph connecting a set of key points  $\mathcal{P}$  using a non-Euclidean metric.

The user first provides an initial elevation terrain  $\mathcal{H}$  and specifies the geological parameters such as the active water-table  $\mathcal{W}$ , inception horizons  $\mathcal{L}$ , permeability  $\pi$ , and preferred conduit orientations which are linked to fracturation. All these parameters may be interactively edited or chosen among a variety of template presets. Then, the user may prescribe key points  $\mathcal{P}$  and label geometric nodes: points may be sinks (inlets), springs (outlets), or known passages inside the bedrock, and influence the karstic structure generation. The user may also force points to be linked together, thus creating a path that will guide the karstic structure.

From these inputs, the network generation proceeds in three

steps (Figure 3). Starting from key points and geological constraints, we first generate a 3D geometric graph connecting those key points using an anisotropic shortest path algorithm (Section 4). The cost function takes into account multiple geological parameters such as conduit orientations, horizons and bedrock permeability. The second step aims at generating the large-scale skeleton  $\mathcal{S}$  of the karstic network (Section 5). To do so, the geometric graph is simplified by using a 3D  $\gamma$ -skeleton approach. Tunnel paths are then labeled into different categories according to their geometrical and geomorphological parameters, and the network is further refined with small tunnels or mesoscale dendritic structures to obtain an augmented skeleton  $\tilde{\mathcal{S}}$  representing the final network. The third step consists of synthesizing the volumetric model  $\mathcal{K}$  (Section 6). The mesoscale geometry  $\mathcal{K}$  of the karstic system is generated from  $\tilde{\mathcal{S}}$  by constructing a hierarchical primitive-based signed distance function  $f: \mathbb{R}^3 \rightarrow \mathbb{R}$  representing a signed distance bound to the surface. Our approach takes inspiration from the construction tree presented in [PGP\*19], with the difference that we rely on a signed distance function instead of compactly supported primitives [WGG99]. Implicit surfaces provide a memory efficient framework for modeling complex volumetric features. We use different types of sweep primitives to synthesize the variety of conduit shapes, and create micro scale details and irregularities of the surface using noise. The final carved terrain model is directly obtained as the difference between the terrain  $\mathcal{H}$  and the volumetric model of the karstic system  $\mathcal{K}$ .

#### 4. Tunnel path computation

In this section we address the construction of a path  $\rho$  connecting two key points  $\mathbf{a}$  and  $\mathbf{b}$  in the bedrock. The path should minimize the line integral over the path of a cost weighting function  $c(\mathbf{p}, \dot{\mathbf{p}})$  representing the characteristics of the bedrock at a given position  $\mathbf{p}$  and in a given direction  $\dot{\mathbf{p}}$ . Let  $\mathcal{C}$  denote the set of all continuous paths inside the bedrock  $\mathcal{H} \subset \mathbb{R}^3$  from  $\mathbf{a}$  to  $\mathbf{b}$  that are piecewise continuously differentiable, *i.e.*, the set of functions  $\rho: [0, T] \rightarrow \mathcal{H}$ , for which  $\rho(0) = \mathbf{a}$  and  $\rho(T) = \mathbf{b}$ . Let  $\chi: \mathcal{P} \rightarrow [0, \infty[$  denote the function characterizing the cost of a path  $\rho \in \mathcal{C}$ :

$$\chi(\rho) = \int_0^T c(\mathbf{p}(t), \dot{\mathbf{p}}(t)) dt$$

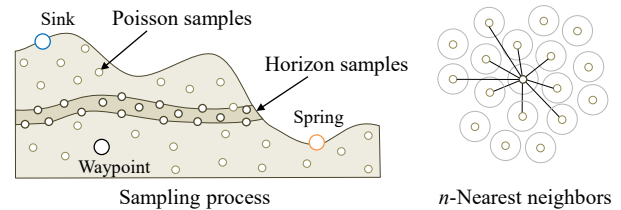
The continuous anisotropic shortest path problem consists in finding a path  $\rho^*$  that minimizes the functional  $\chi(\rho)$ :

$$\rho^* = \underset{\rho \in \mathcal{C}}{\operatorname{argmin}} \chi(\rho)$$

We approximate the solution by adaptively sampling the bedrock  $\mathcal{H}$  with a set of points  $\mathcal{Q}$ . We define the path as a set of segments connecting points, which converts the continuous shortest-path problem into a discrete shortest-path problem on a finite geometric graph  $\mathcal{G}$  whose edges store the approximation of the line integral of the cost function  $c$ . We proceed in two steps. First, we adaptively sample the bedrock according to the geological characteristics and connect samples using a relative neighborhood approach. Then, we evaluate the cost function over the edges of the graph. The discrete anisotropic shortest path is finally directly computed using an A\* algorithm.

#### 4.1. Sampling

Sampling is a crucial step for approximating the continuous anisotropic shortest path problem. Uniform grid sampling sustains two important limitations. First, it often produces regular axis-aligned patterns and yields unrealistic large-scale networks. Moreover, it does not conform to the geological characteristics of the bedrock unless utilizing a dense sampling, which in turn dramatically increases computations. Improving the neighborhood distance between samples to introduce more directions and enhance the angle accuracy between path segments [GPMG10] only partially alleviates the problem.



**Figure 5:** The adaptive sampling of  $\mathcal{H}$  allows the generated network to precisely follow features such as inception horizons; empty remaining space is filled with a Poisson sphere distribution.

Instead, we adaptively sample the bedrock  $\mathcal{H}$  according to its geological characteristics (Figure 5). The set of points  $\mathcal{Q}$  is first initialized with key points  $\mathcal{P}$  representing sinks (inlets), springs (outlets) and interior points in the terrain. We then distribute procedurally-generated samples for every important geological feature, such as horizons, current water table  $\mathcal{W}$  and high permeability volumes (Section 4.2), using a Poisson sphere distribution process [LD06] with a small radius to capture the geometry of the geological features. At a large scale, horizons and the water table can be modeled as a function of altitude, which means that geological samples are distributed at these elevations. As for permeability volumes, we adaptively sample space and distribute samples where  $\pi(\mathbf{p})$  is high. This step ensures that the generated network accords to the geological features as observed in real karstic systems. We fill the rest of the domain with samples by utilizing a Poisson sphere distribution with a larger radius. The graph  $\mathcal{G}$  is constructed as the nearest neighbor graph connecting  $n$  samples (in our experiments we used  $n = 40$ ) to obtain sufficient angle accuracy between the directions of the edges of the graph.

#### 4.2. Geology-based cost functions

At the heart of the path generation process is the cost function  $c$  defined as a weighted sum of different terms, each representing the influence of a geological factor. The relative weights defined by the user allow control over the impact of the geological characteristics which include the inception horizons  $\mathcal{L}$ , fracturing orientations  $\mathcal{F}$ , and permeability of the bedrock  $\pi$ , whose corresponding cost functions are denoted as  $c_{\mathcal{L}}$ ,  $c_{\mathcal{F}}$ , and  $c_{\pi}$  respectively. In addition, the user may prescribe specific paths to influence the generation by assigning a small cost to some of the edges of the graph.

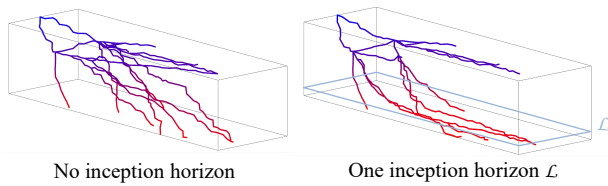
*Horizons* refer to the bedding surfaces that mark a particular



change in the lithology of rocks. They generally influence the karstification process as karstic systems tend to develop along particular inception horizons [FJT09]. In monoclinical contexts where the underlying geology is organized into multiple, almost parallel strata, they can be modeled as a function of altitude. The corresponding cost function is defined as a function of the vertical distance to the nearest horizon  $d(\mathbf{p}, \mathcal{L}) = |\mathbf{p}_z - \mathcal{L}_z|$ :

$$c_{\mathcal{L}}(\mathbf{p}) = g(d(\mathbf{p}, \mathcal{L})/r)$$

The smooth step function  $g(x) = 3x^2 - 2x^3$ , for  $x \in [0, 1]$ , limits the influence of the horizon beyond a distance  $r$ . Figure 6 shows the impact of an inception horizon over the generation process.

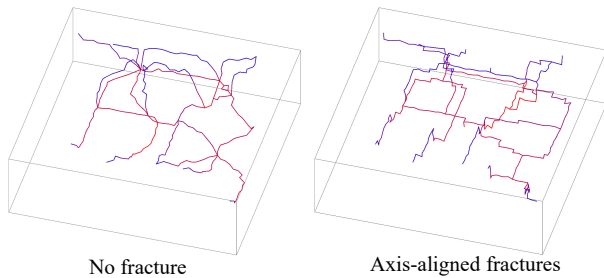


**Figure 6:** Influence of an inception horizon on the karstic network: conduits tend to follow the inception horizon elevation. The blue plane indicates the elevation of the horizon.

Fractures play an important part in the development of karstic systems. A fracture is a break of continuity in the bedrock resulting from tectonics. They have different orientations, such as axis-aligned or diagonal distributions [Pal01]. At the mesoscale, karstic conduits tend to follow local fracture plane orientations. At large-scale (a few kilometers), fracture directions remain almost constant, therefore we represent every fracture as a normalized direction  $\mathbf{n}_k$  representing the orthogonal vector to the fracture plane, and a weight  $w_k$ . Let  $\mathcal{F}$  denote the set of fractures in the bedrock, we define the corresponding cost function  $c_{\mathcal{F}}$  as:

$$c_{\mathcal{F}}(\mathbf{p}, \hat{\mathbf{p}}) = \sum_{k=0}^n w_k (1 - (\mathbf{n}_k \cdot \hat{\mathbf{p}})^2)$$

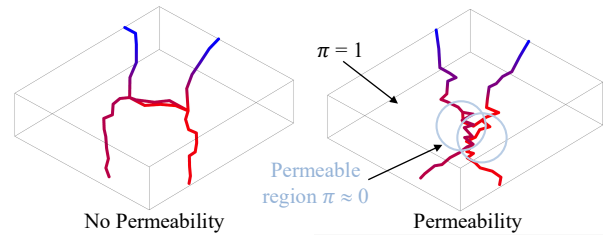
The term  $\hat{\mathbf{p}} = \mathbf{p}/\|\mathbf{p}\|$  represents the unit direction of the path at vertex  $\mathbf{p}$ . Edges with a direction parallel to a fracture plane have a smaller cost value than orthogonal ones and are therefore preferred as illustrated in Figure 7.



**Figure 7:** Influence of an axis-aligned fracture orientation distribution on the karstic network.

Permeability is an important geological parameter as it expresses

the capacity of a rock to let the fluids circulate through its pores. Permeable bedrock lets more water flow, and is thus more prone to chemical dissolution, *i.e.*, karstification. We do not model directly this parameter, but a qualitative parameter expressing a normalized relative resistance to flow passage (the inverse of permeability) using a function  $\pi : \mathbb{R}^3 \rightarrow [0, 1]$  defined at each point  $\mathbf{p}$ . The least permeable rocks have thus the highest value 1, and the most permeable rocks, the lowest value 0. The permeability cost is then defined as  $c_{\pi}(\mathbf{p}) = \pi(\mathbf{p})$ . The function  $\pi$  is defined as a construction tree similar to the one described in [PGP\*19]. The leaves of the construction tree are skeletal primitives that define the local permeability; internal nodes combine primitives to define the permeability function over the entire domain. Depending on the requirements, different primitives and operators such as strata or faults can be used. Our system implements sphere and strata primitives that provide control to the user. Figure 8 illustrates the influence of permeability: the karstic network expands in the most permeable region determined by two user-defined sphere primitives with a high permeability coefficient.



**Figure 8:** Influence of permeability on the karstic network: conduits are generated in the most permeable ( $\pi \approx 0$ ) regions.

## 5. Network generation

We generate the skeleton of the karstic system in two steps. First, we build a large-scale skeleton  $\mathcal{S}$  by computing the 3D  $\gamma$ -skeleton of the key points  $\mathcal{P}$ . Then, we amplify the network by applying a stochastic subdivision step and generating ramifications to obtain a mesoscale skeleton  $\hat{\mathcal{S}}$ .

### 5.1. Large-scale network

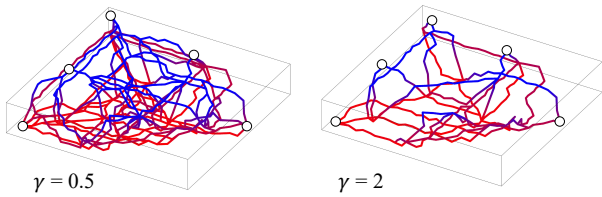
We address the generation of a geometric skeleton connecting the set of key-points  $\mathcal{P}$  using a non-Euclidean metric. The distance metric is directly drawn from the computation of the cost between two points, and defined as  $d(\mathbf{a}, \mathbf{b}) = \chi(\rho^*(\mathbf{a}, \mathbf{b}))$ . Our method takes its inspiration from [GPG11], with the difference that we operate on a volumetric domain and incorporate several different types of key points to constrain the graph generation.

We first compute the complete graph formed by the set of paths connecting  $\mathcal{P}$  by applying multiple anisotropic shortest path algorithms between all pairs of key points as described in Section 4. We then prune the edges using an empty region criterion to create a proximity graph based on the metric  $d$ . Proximity graphs, also referred to as neighborhood graphs, are defined on a finite set of vertices such that there exists an edge between any two vertices

if they satisfy proximity conditions in the context of other connections in the graph. To account for geological features and thus obtain a 3D karstic network, we use a gamma skeleton formalism using this non-Euclidean metric  $d$ . Let  $\mathbf{a}$  and  $\mathbf{b}$  two points in  $\mathcal{P}$ , and  $\gamma \in ]0, +\infty[$ , the parameterized neighborhood region of  $(\mathbf{a}, \mathbf{b})$  is defined as:

$$\Omega_\gamma(\mathbf{a}, \mathbf{b}) = \{ \mathbf{p} \in \Omega, d(\mathbf{a}, \mathbf{b})^\gamma < d(\mathbf{a}, \mathbf{p})^\gamma + d(\mathbf{p}, \mathbf{b})^\gamma \}$$

The neighborhood graph of  $\mathcal{P}$  is created from this definition of the region  $\Omega_\gamma(\mathbf{a}, \mathbf{b})$  which is associated to candidate edges:  $(\mathbf{a}, \mathbf{b})$  forms an edge in the graph if and only if  $\Omega_\gamma(\mathbf{a}, \mathbf{b}) \cap \mathcal{P} = \emptyset$ . More precisely, an arc connecting two key points  $\mathbf{a}$  and  $\mathbf{b}$  of distance  $d(\mathbf{a}, \mathbf{b})$  is kept if and only if there is no path connecting  $\mathbf{a}$  to  $\mathbf{b}$  passing through another key point  $\mathbf{k}$  having  $d(\mathbf{a}, \mathbf{k})^\gamma + d(\mathbf{k}, \mathbf{b})^\gamma < d(\mathbf{a}, \mathbf{b})^\gamma$ .

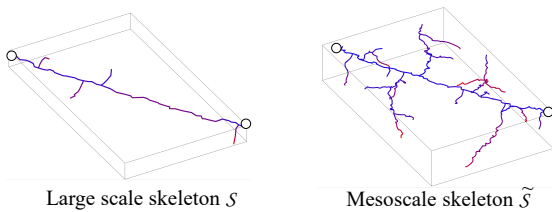


**Figure 9:** Influence of the parameter  $\gamma$ : lower values produce denser networks (key-points are represented with a white circle).

When the value of  $\gamma$  decreases, the neighborhood region shrinks and fewer edges are pruned by the process, which leads to denser skeletons. The parameter  $\gamma$  creates a variety of skeletons connecting the points in  $\mathcal{P}$  and provides a simple and global control to the user over the density of the karstic network (Figure 9). Particularly, note that as  $\gamma \rightarrow \infty$ , the generated graph is equivalent to the relative neighbor graph, and using  $\gamma = 2$  creates the same graph as the Gabriel graph created with a non-Euclidean metric.

## 5.2. Network amplification

The amplification step generates a refined mesoscale skeleton featuring dendritic conduits and more tortuous paths (Figure 10). We improve the large-scale skeleton into a mesoscale skeleton by adding secondary branches and tributaries. We then refine the trajectories of the paths by adding stochastic displacement.



**Figure 10:** Procedural amplification of a large-scale skeleton  $S$  with new tunnels into a refined structure  $\tilde{S}$ .

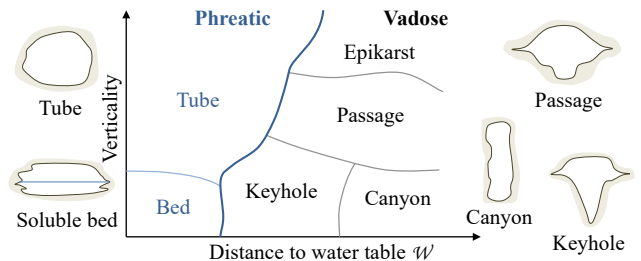
The ramification process starts by distributing new key points randomly distributed in the bounding box of  $S$ , inside the bedrock

$\mathcal{H}$ , and connecting them to  $S$  using the procedure described in Section 4. Points may be of two different types: interior or dead-end nodes. Interior nodes increase the overall complexity of the network possibly leading to mazes, whereas dead-end points produce branchwork structures (Figure 10). The number of new points is controlled by the user.

As real karstic systems exhibit tortuous trajectories, we refine edges using a stochastic midpoint displacement parameterized by a tortuosity factor  $\theta$ . In geomorphology, tortuosity (also referred to as sinuosity) has been proposed to characterize karsts [CBVR17], even if it is heavily dependent on the data acquisition strategy. Consequently, we model tortuosity as a qualitative parameter expressing the maximum displacement factor for the tunnels, ranging from a few centimeters ( $\theta = 0$ ) to 4 meters ( $\theta = 1$ ).

## 5.3. Classification strategy and parameter computation

In order to construct the detailed geometry from the mesoscale skeleton  $\tilde{S}$  (section 6), we compute the type of tunnels for all the graph edges and their corresponding geometric parameters such as tunnel diameters. Edges are labeled as *keyhole*, *passage*, *epikarst*, *tube*, *canyon*, *bed*, or *chamber*, depending on the local geological characteristics. Current knowledge in geomorphology does not allow the identification of precise criteria for the placement of tunnel types. Therefore, we propose a simple labeling strategy (Figure 11) correlating the type to the distance to the current water table  $\mathcal{W}$  and the verticality of the trajectory. A notable advantage of this approach is that it can be adapted to account for a more accurate classification as knowledge in geomorphology evolves.



**Figure 11:** Classification of tunnel types according to their verticality and the distance to the water table.

Vertical tunnels, such as shafts or pits, expose roughly circular shapes (tubular vertical shafts), whereas horizontal tunnels reveal a variety of types such as canyon-shaped tunnels, keyhole-shaped (in vadose zones), or soluble bed conduits (in phreatic zones). Epikarst tunnels are narrow vertical conduits that start from the surface to reach the vadose zone before morphing into a different configuration. Breakout chambers are large caves in the karst. They are created when a tunnel traverses a ghost rock zone – a region where permeability is particularly high ( $c_\pi(\mathbf{p}) \approx 0$ ) and where large collapses are more likely to happen.

Concerning the equivalent radius of the karstic conduits, speleological observations do not allow to identify any specific rule of size distribution. A recent statistical study performed on real networks was unable to show any hierarchy in size distribution, or



Figure 12: Interior of a vadose cave with canyon, bed and tube tunnels, and a phreatic tunnel partially overflowed with water.

relation with the relative position of the springs [FCRV21]. With a variographic analysis, [FCRV21] identified a spatial correlation along the networks, with empirical semi-variogram ranges ranging between 50 to 200 m. This means that two points along a conduit separated by a distance superior to that range have no correlation.

In our model, Epikarst tunnels have a fixed radius of 50 cm as they start from the surface and are usually small. For other types, the radius is computed randomly in a 0.5 – 4.0 m range. These values can be adjusted by the user, or computed by heuristics as knowledge in geomorphology evolves.

### 6. Implicit cave modeling

The generation of the mesoscale geometry addresses two complementary challenges: creating the walls of conduits, chambers, and pits that accord to the cross-sections observed in geology, and generating a sufficiently compact volumetric model allowing to represent karstic networks of a few kilometers with a high level of detail.

Our method relies on a hierarchical construction tree that procedurally defines a signed distance function  $f : \mathbb{R}^3 \rightarrow \mathbb{R}$ , taking positive values outside of the bedrock and negative values inside. We model a continuous, 1-Lipschitz [Har96] signed distance function that defines an upper distance bound to the surface. The Lipschitz property is crucial for establishing surface exclusion criteria, with applications in sphere tracing and polygonization. The leaves are skeletal primitives combined together using Boolean and blending operators located at the internal nodes of the construction tree (Figure 13).

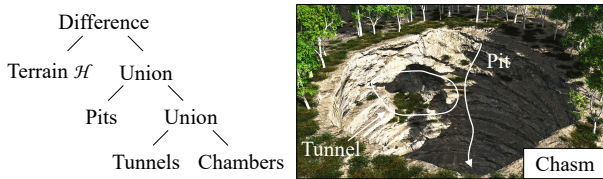


Figure 13: Synthetic representation of the construction tree of a chasm carved into an elevation terrain.

The construction tree is created in two steps: we first convert the input elevation model of the terrain  $\mathcal{H}$  into a signed distance function, then we generate the geometry of the karstic conduits and breakout chambers, and finally add small-scale geometric details such as fractures and rock irregularities.

### 6.1. Terrain

We first convert the input elevation model of the terrain  $\mathcal{H}$ , characterized by an elevation function  $h : \mathbb{R}^2 \rightarrow \mathbb{R}$ , into a signed distance function  $f_{\mathcal{H}}$ . Let  $\lambda$  denote the Lipschitz constant of  $h$ , i.e., the upper bound of the gradient  $\nabla h$ . The signed distance function associated to the terrain is defined as:

$$f_{\mathcal{H}}(\mathbf{p}) = \frac{\mathbf{p}_z - h(\mathbf{p}_{xy})}{\sqrt{1 + \lambda^2}}$$

This equation guarantees that  $f$  is 1-Lipschitz, i.e., represents a signed distance bound to the surface of the terrain (see Appendix A).

### 6.2. Mesoscale geometry of tunnels

Depending on their tortuosity, tunnels are either modeled as sweep primitives, or as unions of spheres. This scheme guarantees that the resulting signed distance function is continuous and 1-Lipschitz. Sweep primitives are compact in memory but are restricted to straight paths, whereas unions of spheres are adapted to tortuous paths but more memory consuming.

Cross-sections identified and referenced in geology may have irregular and asymmetric silhouettes with folds, thus there is a need for a general sweep primitive. Accelerated primitives introduced in [CBS96] are limited to star-shaped cross-sections. The swept volumes method described in [SAJ21] could generate conduits, however, in our case, the tunnels may result from the interpolation of different cross-sections. Additionally, our approach creates a procedurally defined signed distance field, whereas [SAJ21] uses a discrete representation using a sparse voxel grid.

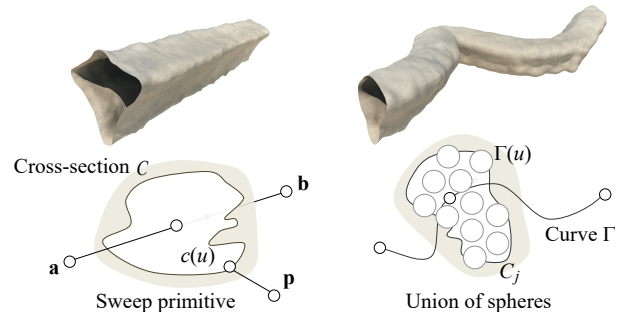
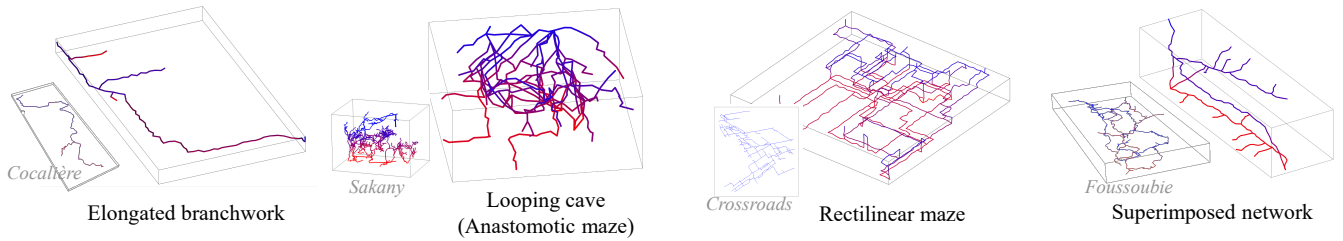


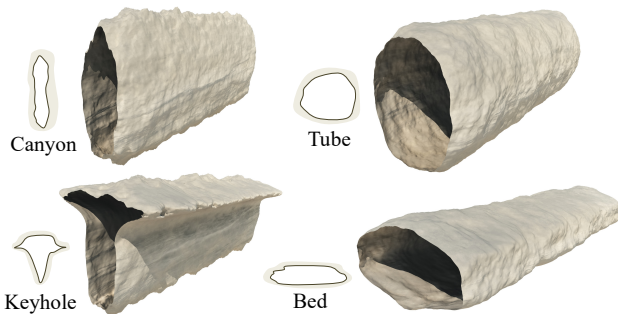
Figure 14: Signed distance computation for sweep primitives.



**Figure 15:** A variety of karstic networks generated with our method compared to real networks in inset.

In our model, primitives are defined by sweeping a closed piecewise quadratic curve  $\mathcal{C}$  along a line segment  $\mathbf{ab}$ . This representation is compact in memory, but the evaluation of the signed distance function is computationally intensive. The signed distance in the cross-section plane is defined as  $f_{\mathcal{C}}(\mathbf{p}) = d(\mathbf{p}, \mathcal{C}) \delta(\mathbf{p}, \mathcal{C})$  where  $d(\mathbf{p}, \mathcal{C})$  denotes the Euclidean distance to  $\mathcal{C}$ , and  $\delta(\mathbf{p}, \mathcal{C})$  the sign function for a point  $\mathbf{p}$ , equal to 1 if  $\mathbf{p}$  is inside, and equal to  $-1$  otherwise (Figure 14). The 3D distance is derived from  $f_{\mathcal{C}}(\mathbf{p})$  according to the segment  $\mathbf{ab}$  (see Appendix B).

Defining a continuous signed distance function for a sweep object with cross-sections swept along a curve  $\Gamma$  is not possible in the general case, in particular when the curve has a high curvature because of discontinuities in the distance function. Thus, we generate the volumetric model of conduits with varying cross-sections along tortuous paths as unions of spheres sampled along the trajectory. This general approach allows us to define a coherent, 1-Lipschitz, signed distance function and preserve the distance bound property.



**Figure 16:** Tunnels generated with sweep primitives.

First, we compute a Poisson sphere distribution  $\mathcal{D} = \{\mathcal{D}_i\}$  inside the bounding box of the curve  $\Gamma$  extended by a radius  $R$ . We then generate  $n$  interpolating cross-sections  $\mathcal{C}_j$  along the curve at regular intervals taking into account the local frame of the curve. Then, we select the Poisson spheres whose minimum distance to the cross sections is lower than  $R/2$  (Figure 14):

$$\tilde{\mathcal{D}} = \{\mathcal{D}_i \mid \min_{j=0}^n d(\mathcal{D}_i, \mathcal{C}_j) < R/2\}$$

Using a small radius  $R \approx 20\text{cm}$  provides a good approximation of the extruded volume, at the expense of memory. Typically, modeling a 10 meters long straight tunnel using a single sweep primitive

takes less than 1kb of memory, whereas the same tunnel modeled with thousands of spheres is  $\approx 100$  times more expensive.

## 7. Results

We implemented the proposed method in C++ and all the karstic systems were generated on a desktop computer equipped with Intel® Core i7, clocked at 4GHz with 16GB of RAM, and an Nvidia GTX 1080ti graphics card. The implicit surface representing the final terrain was polygonized [WMW86] and the resulting mesh directly streamed and procedurally textured into Vue Xstream® to render the final images (Figure 1, 12, 13, 16). The code for synthesizing the karstic systems is available at <https://github.com/aparis69/Karst-Synthesis>.

Our method is the first capable of generating a variety of karstic networks as well as the detailed geometry of the tunnels. Figure 15 shows complex networks generated according to different geological parameters such as inception horizons, orientations and permeability of the bedrock. Figures 12 and 16 show the capabilities of the implicit modeling approach for synthesizing the detailed geometry of the tunnels, capturing different shapes such as keyhole, canyon and bed tunnels.

### 7.1. Performance

In our implementation, the computation of the anisotropic shortest path combined with the 3D gamma skeleton geometric graph construction completes in less than a second, which allows for interactive authoring even for relatively large networks featuring thousands of nodes and dozens of key-points (Table 1). The synthesis of the nearest neighbor graph between all sample points  $\mathcal{Q}$  is the most computationally intensive step and takes up to 3 seconds in the most complex geological configurations.

One significant advantage of the signed distance function construction tree is the reduced memory footprint compared to hybrid [PGMG09] and voxel-based models [BKRE19]. The tortuosity factor  $\theta$  directly relates to the construction tree generation time: trajectories with a high coefficient  $\theta$  require instantiating a high number of spheres. Still, the generation of the construction tree is completed in a few seconds in the worst case scenario (Table 1).

A typical editing session takes up to a few minutes (see accompanying video): the user inserts points in the scene, sets the weights of the cost function, adjusts geological parameters (such as fracture



Karstic system	Extent		Topography		Skeleton			Geometry					
	Size	Length	# $\mathcal{Q}$	# $\mathcal{P}$	# $\mathcal{S}$	Time	$\theta$	$\gamma$	# $\mathcal{N}$	Sweeps	Spheres	$\mathcal{M}$	Time
Branchwork	$0.4 \times 0.6 \times 0.1$	2.3	25265	17	154	0.2	0.1	2.0	31256	59	15569	2.9	2.9
Looping cave	$0.5 \times 0.7 \times 0.2$	22.1	59246	47	668	0.5	0.5	0.8	67236	809	32809	7.4	7.5
Rectilinear	$0.9 \times 1.1 \times 0.1$	19.6	65887	26	830	0.5	0.4	0.8	29719	1076	13783	3.3	3.0
Superimposed	$0.2 \times 0.7 \times 0.2$	2.9	92007	17	209	0.9	0.1	2.0	43016	189	21319	4.7	5.2

**Table 1:** Karstic system extent [km<sup>3</sup>], skeleton length [km], number of point samples # $\mathcal{Q}$ , and number of key points # $\mathcal{P}$  placed by the user; karstic skeleton node count # $\mathcal{S}$ , skeleton generation time [s], tortuosity factor  $\theta$  given by the user, neighborhood parameter  $\gamma$  given by the user; number of node in the construction tree # $\mathcal{N}$ , number of sweep and sphere primitives, memory footprint  $\mathcal{M}$  of the construction tree [Mb] and construction tree generation time [s].

orientations and horizon elevations), and triggers and the computation of the network. During user interaction, visualization uses a symbolic representation of the caves and tunnels. Those steps are repeated until the user is satisfied with the generated network. The detailed geometry of the karstic network is finally computed for final visualization and rendering.

## 7.2. Control

Our approach combines the placement of key points, the definition of geological features (inception horizons, orientation distributions), the tuning of the cost function as well as setting the  $\gamma$  coefficient. This parameter provides user control and allows the reproduction of identified patterns found in real karstic systems. Figure 15 shows synthesized networks compared to real ones displayed in insets. Typically, rectilinear and anastomotic mazes are created by setting  $\gamma = 0.8$  and tuning orientation distributions (three axis-aligned directions for rectilinear, and 8 directions on the sphere for the looping cave). In contrast, an elongated branchwork results from the placement of a single sink and a spring. A superimposed network results from two inception horizons and several amplification steps (3 in the presented models) adding dendritic tunnels to the network.

Our method allows the interactive authoring of complex karstic networks spanning over several kilometers (see Figure 15 and accompanying video). Labeled key points  $\mathcal{P}$  not only control the location of sinks, springs, or breakout chambers inside the bedrock but also allow the user to prescribe tunnel sub-paths that are directly taken into account in the network generation and locally influence the structure of the skeleton. The cost function may also be tuned with interactive visual feedback (see accompanying video). The geological parameters and control waypoints provide user-control over the density and structure of the resulting network. At the mesoscale, sweep primitives allow for a detailed and varied reconstruction of the conduits (Figure 16). A high tortuosity parameter  $\theta$  significantly impacts the complexity of the construction tree by producing tortuous trajectories as observed in real karstic systems. The looping cave example (Figure 15) was generated with the highest tortuosity ( $\theta = 0.5$ ) with a memory consuming construction tree featuring 32809 sphere primitives (see Section 6.2). In contrast, the elongated branchwork or superimposed networks with  $\theta = 0.1$  have a reduced memory footprint.

## 7.3. Comparison with real karstic networks

Collon *et al.* [CBVR17] presented and discussed a set of metrics to describe, compare and quantify karstic networks. Among the 21 tested metrics, they recommend computing 6 of them to identify the geometry and the topology of a network. The open-source Karstnet code (<https://github.com/PaulineCollon/karstnet>) implements them in Python and was used to compute the metrics for the karstic networks presented in Figure 15 and to compare them with a dataset of 34 real caves [CBVR17].

Karstic system	$\bar{l}$	$H_O$	$\bar{k}$	$r_k$	$\overline{SPL}$	$CPD$
Branchwork	96	0.9	2.0	-0.4	4.1	0.6
Looping cave	79	0.9	3.2	0.04	5.0	0.2
Rectilinear maze	121	0.6	3.0	0.0	5.6	0.1
Superimposed net	101	0.8	2.1	-0.3	4.9	0.4
Observed range	8	0.8	1.8	-0.6	2.3	0.0
	331	1.0	2.6	-0.2	55.7	0.6

**Table 2:** Average branch length  $\bar{l}$  [m], entropy of orientation  $H_O$ , average vertex degree  $\bar{k}$ , correlation of vertex degrees  $r_k$ , average shortest path length  $\overline{SPL}$ , central point dominance  $CPD$ .

Table 2 reports statistics and compares to a range of values computed on a dataset [CBVR17] of 34 real karstic networks. The average branch length  $\bar{l}$  and the entropy of orientation  $H_O$  describe the geometry of the network. The simulated networks show similar values to the karsts of the database, except for the entropy of orientation  $H_O$  of the *rectilinear maze* which is slightly lower than those of studied networks. This is, however, not surprising as we voluntarily restricted the influence of fracture orientation to generate this *stereotyped* network. In the available database, there was no 3D network with such marked orientation. The data of the *crossroads* karst, presented in Figure 15, was only available as a 2D map projection [Pal91]. Computing this metric on *crossroads* data yields  $H_O = 0.46$ , demonstrating that the value  $H_O = 0.62$  obtained on the *rectilinear maze* is acceptable for this kind of network.

The computed values also remain consistent for the topological metrics. Average shortest path length  $\overline{SPL}$  and central point dominance  $CPD$  fall both into the observed range of values for all four

simulated networks, as well as average vertex degree  $\bar{k}$  and correlation of vertex degrees  $r_k$  for the *elongated branchwork* and *superimposed* networks. The maze-like networks *looping cave* and *rectilinear maze* have  $\bar{k}$  values slightly superior to what was observed. It is admittedly rare that real karstic networks have such a high amount of crossing points where more than 3 conduits meet, and breakout chambers with more than 4 conduits (and consequently node degree superior or equal to 4) are scarce. This is again a consequence of our intention to generate stereotyped networks, which also explains the corresponding  $r_k \simeq 0$  while natural systems tend to be slightly disassortative.

#### 7.4. Comparison with other techniques

Few methods for modeling complex karstic networks exist. The Arches model [PGMG09] requires manual editing for authoring terrains and does not automatically generate tunnels or caves. The invasion-percolation simulation introduced in [PGP\*19] does not consider relevant geological parameters such as tunnel orientations or inception horizons. While voxel-based approaches [PM15] analyse parameters such as pressure and fractures, those simulations are computationally intensive and do not lend themselves to synthesizing large karstic networks. Our approach extends the grid-based method presented in [GPGB11] and proposes a grid-less technique that can adaptatively sample three-dimensional domains. In particular, our algorithm solves geometric aliasing issues resulting in unrealistic tunnel paths, which occur when using grids. To the best of our knowledge, this is the first controllable procedural method that captures the complex structure of karstic networks both at large scale and mesoscale.

Compared to data-oriented or hybrid models, the construction tree is compact in memory and allows for modeling large karstic networks. Tunnels modeled with computationally intensive noise-based primitives [PGP\*19] showed uniform circular cross-sections, whereas sweep primitives capture the different configurations reported in geology (see Figure 16).

#### 7.5. Limitations

We rely on a simple strategy for the classification of tunnel types and parameters computation, as knowledge in geomorphology is sparse on this topic. However, the proposed framework provides effective user-control and processing parts could be easily replaced with more accurate computations in the future.

Although implicit surfaces allows modeling of the highly detailed geometry of karstic conduits, visualization remains expensive as the signed distance function  $f$  is defined by a hierarchical combination of thousands of nodes. In our implementation, polygonizing a  $1 \times 1 \times 0.4 \text{ km}^3$  volume at 0.5m precision takes up to two minutes for a dense complex karstic model featuring 30 000 nodes. The construction tree may be amenable to simplification; primitives and operators could be further optimized and field function queries performed in parallel using compute shaders on graphics hardware.

#### 8. Conclusion

We have introduced a geologically-based framework for modeling karstic networks. Given an input terrain, our method com-

putes a three-dimensional geometric graph connecting key points corresponding to sinks (inlets), springs (outlets), and known passages inside the bedrock. The paths connecting control points are constructed by using a gridless anisotropic shortest path taking into account geological parameters such as the permeability of the bedrock, fracture orientations, and inception features. The geometry of the conduits is finally constructed by a signed distance function defined as a construction tree combining volumetric primitives. The synthesized karstic networks accord with real data obtained from geological observations.

Our method allows for modeling large scale karstic networks as well as their detailed mesoscale geometry. Small scale details such as stalactites, stalagmites, or wall irregularities could be added at the expense of a more complex construction tree. Tortuous trajectories currently require many spherical primitives organized into memory demanding sub-trees, and computationally intensive signed distance evaluations. More conforming sampling schemes, as well as a simplification of the hierarchical construction tree, would probably speed up computations.

#### Acknowledgments

This work was part of the project AMPLI ANR-20-CE23-0001, supported by Agence Nationale de la Recherche Française. We thank Stéphane Jaillet, Jean-Pierre Cassou, and Arthur Palmer for providing the data of karstic networks. Finally, we thank Lucie Fournier for implementing the code for modeling the geometry of the shafts and some parts of the karstic systems.

#### References

- [BFO\*07] BEARDALL M., FARLEY M., OUDERKIRK D., REIMSCHUESSEL C., SMITH J., JONES M., EGBERT P.: Goblins by spheroidal weathering. In *Proceedings of Third Eurographics Conference on Natural Phenomena* (2007), pp. 7–14. 2
- [BKRE19] BECHER M., KRONE M., REINA G., ERTL T.: Feature-based volumetric terrain generation and decoration. *IEEE Transactions on Visualization and Computer Graphics* 25, 2 (2019), 1283–1296. 1, 2, 8
- [BRJ12] BORGHI A., RENARD P., JENNI S.: A pseudo-genetic stochastic model to generate karstic networks. *Journal of Hydrology* 414-415 (2012), 516–529. 2
- [CBS96] CRESPIN B., BLANC C., SCHLICK C.: Implicit sweep objects. *Computer Graphics Forum* 15, 3 (1996), 165–174. 7
- [CBVR17] COLLON P., BERNASCONI D., VUILLEUMIER C., RENARD P.: Statistical metrics for the characterization of karst network geometry and topology. *Geomorphology* 283 (2017), 122 – 142. 3, 6, 9
- [CCZ11] CUI J., CHOW Y.-W., ZHANG M.: Procedural generation of 3D cave models with stalactites and stalagmites. *International Journal of Computer Science and Network Security* 11 (2011), 94–101. 2
- [CHP12] COLLON P., HENRION V., PELLERIN J.: An algorithm for 3D simulation of branchwork karst networks using horton parameters and A\*: Application to a synthetic case. *Geological Society of London – Special Publications* 370 (1) (11 2012). 2
- [FCRV21] FRANTZ Y., COLLON P., RENARD P., VISEUR S.: Analysis and stochastic simulation of geometrical properties of conduits in karstic networks. *Geomorphology* 377 (2021), 107480. 7
- [FJT09] FILIPPONI M., JEANNIN P.-Y., TACHER L.: Evidence of inception horizons in karst conduit networks. *Geomorphology* 106, 1 (2009), 86–99. 1, 5

- [FP00] FIELD M., PINSKY P.: A two-region nonequilibrium model for solute transport in solution conduits in karstic aquifers. *Journal of Contaminant Hydrology* 44 (2000), 329–351. 2
- [GGP\*19] GALIN E., GUÉRIN E., PEYTAIVIE A., CORDONNIER G., CANI M.-P., BENES B., GAIN J.: A review of digital terrain modeling. *Computer Graphics Forum (Proceedings of Eurographics 2019)* 38, 2 (2019), 553–577. 2
- [GPGB11] GALIN E., PEYTAIVIE A., GUÉRIN E., BENEŠ B.: Authoring hierarchical road networks. *Computer Graphics Forum* 30, 7 (2011), 2021–2030. 5, 10
- [GPMG10] GALIN E., PEYTAIVIE A., MARÉCHAL N., GUÉRIN E.: Procedural Generation of Roads. *Computer Graphics Forum* 29 (2010), 429–438. 4
- [Har96] HART J. C.: Sphere tracing: A geometric method for the antialiased ray tracing of implicit surfaces. *The Visual Computer* 12, 10 (1996), 527–545. 7
- [HCC10] HENRION V., CAUMON G., CHERPEAU N.: ODSIM: An object-distance simulation method for conditioning complex natural structures. *Mathematical Geosciences* 42, 8 (2010), 911–924. 2
- [HGW\*14] HARTMANN A., GOLDSCHIEDER N., WAGENER T., LANGE J., WEILER M.: Karst water resources in a changing world: Review of hydrological modeling approaches. *Reviews of Geophysics* 52, 3 (2014), 218–242. 1
- [HR16] HENDRICK M., RENARD P.: Subnetworks of percolation backbones to model karst systems around tulum, mexico. *Frontiers in Physics* 4 (2016), 43. 2
- [JFBB10] JONES M., FARLAY M., BUTLER M., BEARDALL M.: Directable weathering of concave rock using curvature estimation. *IEEE Transactions on Visualization and Computer Graphic* 16, 1 (2010), 81–97. 2
- [JVA\*17] JOUVES J., VISEUR S., ARFIB B., BAUDEMONT C., CAMUS H., COLLON P., GUGLIELMI Y.: Speleogenesis, geometry and topology of caves: a quantitative study of 3D karst conduits. *Geomorphology* 298 (2017), 86–106. 2, 3
- [LD06] LAGAE A., DUTRÉ P.: Poisson sphere distributions. In *Vision, Modeling, and Visualization* (2006), pp. 373–379. 4
- [MBMT15] MARK B., BERECHE T., MAHLMANN T., TOGELIUS J.: Procedural generation of 3D caves for games on the GPU. *Foundations of Digital Games* (2015). 2
- [NLP\*13] NATALI M., LIDAL E. M., PARULEK J., VIOLA I., PATEL D.: Modeling Terrains and Subsurface Geology. *Computer Graphics Forum (Proceedings of Eurographics 2013)* (2013), 155–173. 2
- [Pal91] PALMER A.: Origin and morphology of limestone caves. *Geol. Soc. Am. Bulletin* 103 (1991), 1–21. 9
- [Pal01] PALMSTROM A.: *Measurement and characterization of rock mass jointing*. Rotterdam, 2001, pp. 49–97. 5
- [Pal03] PALMER A.: Speleogenesis in carbonate rocks. *Speleogenesis and Evolution of Karst Aquifers* (01 2003), 11. 2
- [PGMG09] PEYTAIVIE A., GALIN É., MÉRILLOU S., GROSJEAN J.: Arches: A framework for modeling complex terrains. *Computer Graphics Forum* 28, 2 (2009), 457–467. 1, 2, 8, 10
- [PGP\*19] PARIS A., GALIN E., PEYTAIVIE A., GUÉRIN E., GAIN J.: Terrain Amplification with Implicit 3D Features. *ACM Transactions on Graphics* 38, 5 (2019). 1, 2, 4, 5, 10
- [PIDC\*12] PARDO-IGUZQUIZA E., DOWD P. A., CHAOSHUI X., DURAN-VALSERO J. J., PARDO-IGUZQUIZA E., XU C., DURAN-VALSERO J. J.: Stochastic simulation of karst conduit networks. *Advances in Water Resources* 35 (2012), 141–150. 2
- [PM15] PYTEL A., MANN S.: Procedural modeling of cave-like channels. *Journal of Computer Graphics Techniques* 4, 2 (2015), 10–29. 2, 10
- [RCDF14] RONGIER G., COLLON-DROUAILLET P., FILIPPONI M.: Simulation of 3D karst conduits with an object-distance based method integrating geological knowledge. *Geomorphology* 217 (2014), 152–164. 3
- [SAJ21] SELLÁN S., AIGERMAN N., JACOBSON A.: Swept volumes via spacetime numerical continuation. *ACM Trans. Graph.* 40, 4 (2021), 55:1–11. 7
- [VJF\*14] VISEUR S., JOUVES J., FOURNILLON A., ARFIB B., GUGLIELMI Y.: 3D stochastic simulation of caves : application to Saint-Sébastien case study (SE, France). *Karstologia* 64 (2014), 17–24. 2
- [WGG99] WYVILL B., GUY A., GALIN E.: Extending the CSG tree – warping, blending and boolean operations in an implicit surface modeling system. *Computer Graphics Forum* 18, 2 (1999), 149–158. 4
- [WMW86] WYVILL G., MCPHEETERS C., WYVILL B.: Data structure for soft objects. *The Visual Computer* 2 (1986), 227–234. 8

### Appendix A: Distance function of a heightfield

Let  $h: \mathbb{R}^2 \rightarrow \mathbb{R}$  denote the elevation function of the terrain  $\mathcal{H}$ . Let  $\lambda$  denote the Lipschitz constant of  $h$ , which represents the maximum slope of the terrain. Then the squared gradient of  $f(\mathbf{p}) = \mathbf{p}_z - h(\mathbf{p}_{xy})$  is  $\|\nabla f\|^2 = 1 + (\partial h/\partial x)^2 + (\partial h/\partial y)^2 = 1 + \|\nabla h\|^2$ . Therefore,  $f(\mathbf{p}) = (\mathbf{p}_z - h(\mathbf{p}_{xy}))/\sqrt{1 + \lambda^2}$  is 1-Lipschitz.

### Appendix B: Distance function for the segment sweeping

Let  $\mathbf{u} = (\mathbf{b} - \mathbf{a})/\|\mathbf{b} - \mathbf{a}\|$  the unit vector of the segment  $\mathbf{ab}$ . Let  $l = (\mathbf{p} - \mathbf{a}) \cdot \mathbf{u}$  denote the distance of the projection of  $\mathbf{p}$  on the segment to  $\mathbf{a}$ . Recall that  $d(\mathbf{p}, \mathcal{C})$  denotes the distance to the contour  $\mathcal{C}$  in its supporting plane. The signed distance to the sweep primitive is defined as:

$$d(\mathbf{p}) = \sqrt{s(\mathbf{p})^2 + d(\mathbf{p}, \mathcal{C})^2} \quad s(\mathbf{p}) = \begin{cases} -l & \text{if } l < 0 \\ l - \|\mathbf{b} - \mathbf{a}\| & \text{if } l > \|\mathbf{b} - \mathbf{a}\| \\ 0 & \text{otherwise.} \end{cases}$$

### Appendix C: Signed distance to a contour

Let  $d(\mathbf{p}, \mathcal{C})$  denote the (positive) Euclidean distance to the contour  $\mathcal{C}$ , i.e., the minimum Euclidean distance between  $\mathbf{p}$  and the points  $\mathbf{c} \in \mathcal{C}$ . The signed distance to  $\mathcal{C}$  is obtained by determining whether  $\mathbf{p}$  is inside or outside of the contour, by computing the number of intersections between a line  $\Delta$  and  $\mathcal{C}$ , and changing the sign accordingly.

$$f(\mathbf{p}, \mathcal{C}) = d(\mathbf{p}, \mathcal{C}) \delta(\mathbf{p}, \mathcal{C}) \quad \delta(\mathbf{p}, \mathcal{C}) = \begin{cases} -1 & \text{if } \#\Delta \cap \mathcal{C} \text{ odd} \\ +1 & \text{otherwise.} \end{cases}$$

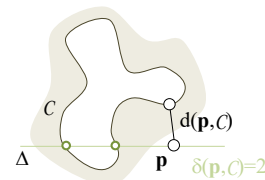


Figure 17: Signed distance to a contour  $d(\mathbf{p}, \mathcal{C})$ .