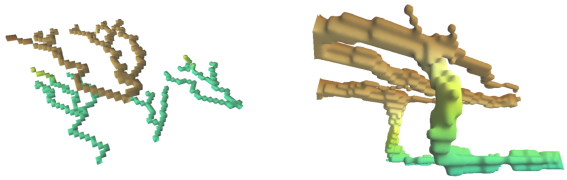
Procedural Modeling of Cave-like Channels

Alex Pytel Stephen Mann
University of Waterloo



(a) (b)

Figure 1. Procedurally modeled channel network: (a) protochannels; (b) channels with breakthrough from lower to higher level.

Abstract

Hydraulic erosion that takes place underground leads to the formation of complex channel networks whose morphology emerges from the dynamic behavior of each channel, based on the presence of other channels nearby. Our approach to the problem of modeling such channel networks for computer graphics application involves a self-organized model of channel development and a two-stage simulation for constructing the geometry of the channels. By emphasizing self-organization of flow and pressure, our simulation is able to reproduce several types of channel behavior known from hydrogeomorphology, such as tributary capture.

1. Introduction

Dissolution of carbonate rock, such as limestone, by subterranean water flow leads to the development of complex channel networks and gives rise to a type of landscape called *karst* and a class of caves called karstic caves. The morphology of such networks emerges from the dynamic behavior of their constituent channels, which has an analogue in streams produced on the surface of terrains: flow becomes concentrated into channels that may compete for water and capture neighbors as tributaries. However, the behavior of 3D underground channels is more complex due to the possibility that pressure may cause water to flow upward. In particular, a configuration of channels called a *phreatic loop* may arise when one channel breaks through to another that is located above it.

 ISSN 2331-7418

As channels grow to be the size of caves (i.e., large enough to be enterable by a human), they can also develop complicated cross-sections, as well as several types of features that exist on a finer scale than the spatial relationships between the components of the channel network. Such features include dissolution features on the walls of conduits and *speleothems*, or structures formed by deposition, such as stalagmites. Additionally, the channels may also undergo changes in flow due to collapse and seasonal input fluctuations.

These considerations greatly complicate modeling of cave-like 3D channel networks for computer graphics applications. However, the more abstract problem of modeling the development of a channel network according to the basic principles of channel behavior, such as tributary capture, and properties of the environment, such as the presence of fractures in the surrounding rock, is well-suited to be solved with a procedural approach based on simulation of self-organized behavior.

The main contribution of this paper is both a self-organized model of flow and pressure in 3D channel networks and a two-stage procedural modeling method for constructing the geometry of cave-like channels. The first stage simulates the development of protochannels, which grow and link up during the second simulation stage (Figure 1). The procedure is able to reproduce formation of phreatic loops, which is essential for simulating realistic channel development in 3D and which so far has not been attempted in procedural modeling. Additionally, the model of flow and pressure that lies at the foundation of our simulation provides for self-organization of pressure in a novel fashion by using semantics similar to avalanching of sand.

2. Background and Previous Work

2.1. Modeling of Caves

A common method used to analyze channel networks on the surface of terrains is based on treating each terrain site as a node in a graph and assigning a directed edge between neighboring nodes that encodes the "contributes water to" relationship. One of the modeling approaches for caves proposed by Boggus and Crawfis [2009] approximates the branching appearance of 3D channels with a 2D simulation of river-like flow according to the 2D channel network graph. As the scheme constrains any constructed channels to lie in a plane, the heightmap representing the floor of the channels is mirrored to create their ceiling.

Complexity of geometric data structures is a major obstacle to simulating more general channels that can form arbitrary loops in 3D and multiple levels of passages that interconnect. Another modeling framework by Boggus and Crawfis [2010] is based on a specialized data structure called a prismfield, which is a type of generalized heightmap capable of modeling overhangs. However, the framework is intended for sculpting caves manually without the benefit of procedural modeling. The man-

ual modeling problem is perhaps better solved as part of a complete terrain modeling package, as exemplified by the Arches framework developed by Peytavie et al. [2009]. Arches can construct terrains with caves via implicit surface modeling and also provides tools for sweeping out passages with complicated cross-sections and placing collapsed rock.

The modeling framework of Cui et al. [2011] retains procedural modeling capability, but dispenses with simulation of water flow. The framework represents both cave chambers and speleothems with voxels that are later polygonized into a mesh representation. The voxel-based approach proves to be suitable for creating a plausible cave environment consisting of a tubular passage with a cross-section derived from a noise function, stalactites, and stalagmites. However, the level of detail is too great to use the method to construct a network of passages, and the method lacks a model for simulating the development of such a network in the first place. The framework of Cui et al. also does not model the growth of speleothems. The growth of stalactites and stalagmites has been simulated by Tortelli and Walter [2009] using a simple approach that works directly on the polygonal representation of the speleothems.

In summary, existing methods put a greater emphasis on modeling relatively small-scale features, such as speleothems, and do not attempt to automate construction of extensive 3D channel networks using simulation. In contrast, the goal of this paper is to simulate the formation and growth of channel networks with complex topology based on a new model that follows principles of channel development known from hydrogeomorphology (Section 2.2). By focusing only on channel behavior, our model can be implemented using a uniform voxel discretization, and there is no need to manage level of detail issues. As a separate problem, construction of speleothems can be deferred to one of the existing models.

2.2. Formation of Subterranean Channels by Dissolution

The scientific description of underground channels according to hydrogeomorphology [Ford and Williams 2007] accounts for many variations in the phenomenology of channel development that have been observed in karstic landforms associated with the presence of channels. However, by relying on the expressive power of self-organization the modeling approach of this paper can make many simplifications and still reproduce the factors that result in topologically complex networks. We disregard the chemistry of dissolution, assume that there is no variation in flow input, and emphasize formation of channels below the water table.

When water flows through a porous matrix it initially encounters a substantial resistance and creates *protochannels*, which consequently may take a long time to grow. The flow inside of the protochannels is increasingly impeded, as the pores surrounding each protochannel become filled with water and pressurized. In the isotropic case (Figure 2a), the protochannels can be expected to grow radially to produce a

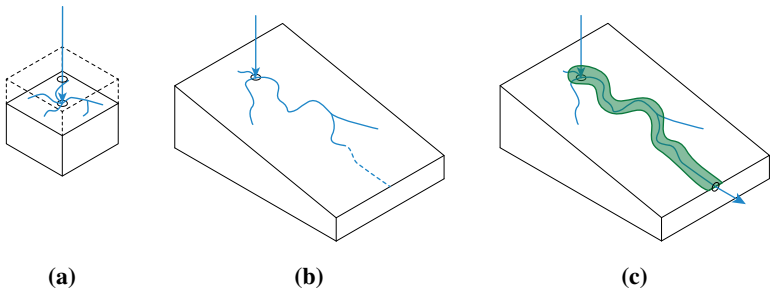


Figure 2. Protochannel development: (a) isotropic case; (b) growth; (c) breakthrough and eventual channel.

wormiform network similar to experiments performed by McDuff et al. [2010], who injected acid into carbonate rock and used CT scanning to image the dissolution patterns. However, it is more typical that protochannels develop in a preferred direction due to pressure and presence of fracture planes in the rock (Figure 2b).

Eventually, a protochannel may reach a fracture or cavity in the rock where resistance to flow is lower. Following this event, called *breakthrough*, flow inside the protochannel rapidly increases as the protochannel and the surrounding pores drain into the low resistance area. The concomitant erosion enlarges the protochannel, now referred to as a channel (Figure 2c), and increases its carrying capacity, which causes more erosion and can even remove much of the rock containing the original protochannels.

Figure 3 illustrates the development of protochannels before and just after breakthrough in more detail. When the rock matrix impedes flow, it creates a differential of pressure between the pressurized region surrounding the source and the low pressure zone around the sink. The figure shows this using idealized pressure isolines and a cross-section of the idealized pressure surface. As a protochannel breaks through, the

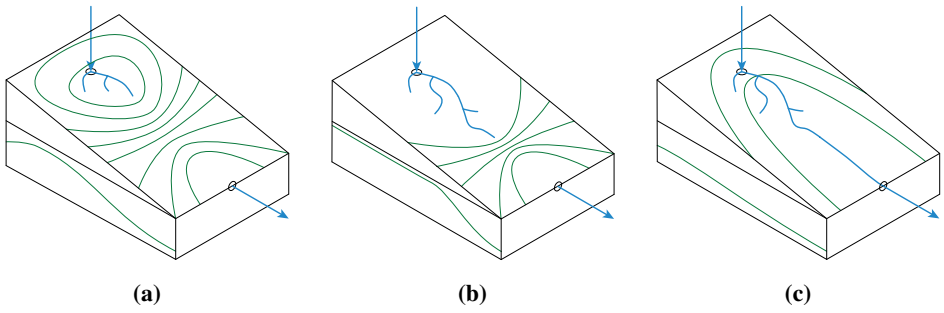


Figure 3. Protochannel behavior during breakthrough: (a) pressure grows around source; (b) protochannel nears low pressure zone around sink; (c) pressure drops after breakthrough.

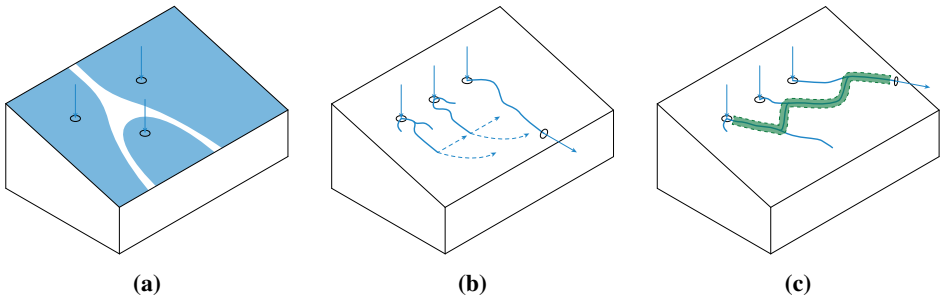


Figure 4. Interaction between channels: (a) pressure zones; (b) tributary capture; (c) phreatic loops.

pores in its vicinity drain to the sink and the pressure within the surrounding rock drops. However, as long as the channel is saturated, the pressure around the source will continue to be at least slightly larger than around the sink due to drag along the walls of the channel.

The effects of pressure also help to explain the behavior of systems with multiple inputs, as shown in Figure 4. Prior to breakthrough, pressurization about each source creates pressure zones that act as obstacles to protochannel development. After breakthrough, the breaking-through channel creates a pressure gradient around itself that may allow it to capture neighboring channels as tributaries after they redirect themselves according to the gradient. As the effect of decrease in pressure propagates, additional channels may become linked in a sequence. However, it is also possible that the sink's area of low pressure is large enough for several nearby channels to begin growing towards it without any of the channels increasing the extent of the low pressure zone first. This case typically occurs when the permeability of the rock matrix around the sink is high due to the presence of fractures, so that it remains at a pressure close to that of the sink, as the low resistance to flow prevents formation of large pressure differentials. Figure 4b shows the two possible modes of development. The pressure gradient can cause one channel to break through to another even if this requires water to flow up. When this happens in the zone of rock below the water table, called the phreatic zone, the resulting configuration of a channel and its tributary is called a *phreatic loop*.

2.3. More Mechanisms Producing Cave-like Channels and Features

For a more balanced presentation, we extend our discussion of channel formation with the following cave-related processes that are of interest from the point of view of procedural modeling, but fall outside of the scope of our approach for simulating channel networks formed by solution. The processes can be grouped into three types: first, processes related to complex water flow above the water table; sec-

ond, dissolution related to rising flow; third, cave-forming processes not involving solution.

The first category of processes may form two types of features that are especially complex and visually appealing. Meandering canyons can be swept out by either a retreating waterfall or an entire channel segment that cuts lengthwise into the rock above or below it. Vertical shafts range between two extremes: shafts cut out by powerful waterfalls that tend to be somewhat irregular due to interaction between the waterfall and the surrounding rock and domepits, in which flow adheres to the walls, disperses, and creates flutings [Ford and Williams 2007].

In the second category of cave-forming processes, rising flow causes dissolution via a convective or hydrothermal mechanism or one based on sulphuric acid chemistry. The resulting channel networks tend to be ramified or maze-like, especially in the presence of high-fissure frequency, or if the growing channels intrude on a pre-existing inert network.

Cave-forming processes of the third type involve the action of volcanic lava and meltwater in ice. Lava-tube caves form as a result of a complex interplay of cooling and melting as lava flows underground and, additionally, due to crusting over the surface of lava rivers [Halliday 2004]. Glacier caves form in a manner similar to solution caves, but with melting instead of solution. However, there are some major differences: there is less matrix flow of water in ice and the resulting channels tend to be short-lived, especially since ice creeps under sustained load [Smart 2004].

3. Model and Simulation

Our solution to the problem of modeling 3D channel networks consists of a procedural model of protochannel and channel development that we use in a simulation that tracks the geometry of a channel network as it evolves. Our model and simulation are connected at a fundamental level, because we rely on principles of self-organization to express the behavior of channel-network components. Our most significant contribution is the self-organized model of pressure and flow that we use to simulate growth of channels, including formation of phreatic loops.

We discretize the simulation domain as a grid of voxels. Each voxel element represents a unit of rock that can contain an amount of water corresponding to its porosity. Additionally, the voxel grid can use porosity to represent bedding planes and fractures in the rock matrix. When erosion increases porosity of the voxels inside channels, the faces of appropriate voxels become visualized via polygonization. The protochannel development stage (Section 3.1) creates initial channels beginning at voxels that contain sources of flow. The channel-growth and linkage stage (Section 3.2) simulates flow from source voxels to sink voxels via a self-organized formulation that details how pressure responds to erosion, how flow responds to pressure, and how flow causes

erosion. Tributary capture can occur in an emergent manner when a channel is able to drain the rock matrix in its vicinity and create a low pressure zone that redirects other channels.

3.1. Protochannel Stage

The goal of the protochannel stage is to create physically-motivated initial conditions for the channel-growth stage in the form of vestigial channels that grow out of each source, elongate according to rock permeability and pressure differentials, and have not yet broken through to a sink. As protochannels grow, they compete for space by pressurizing the pores of the rock in their vicinity, slowing down the development of affected protochannels (Figure 4a). The first part of our model for protochannels accounts for this competition by letting each protochannel's source claim an area according to its pressure.

The second part of our model details the behavior of the protochannels as they grow. Since we are assuming that the breakthrough has not occurred yet, the sinks are inactive and flow through the rock matrix is small. Additionally, the flow is further constrained, because each source has created a pressure zone of water-filled pores, which have to be displaced for flow to occur. We assume that displacement of water in the pores happens extremely slowly due to drag along the surface of the pores, so that pressure of interacting pressure zones can become nearly equalized. Therefore, only local changes in porosity and a small local gradient of flow affects the growth of each channel. We simulate the presence of an idealized small-flow gradient using a set of protosinks on the boundaries of each source's pressure zone. In the isotropic case, this causes the protochannels to branch out in all directions. Clustering the protosinks results in protochannels that grow in a preferred direction.

We implement the first part of our protochannel model by using fast marching (Section 5) to approximate the extent of each pressure zone according to the distance to the closest source weighted by pressure of the source. Using distance this way

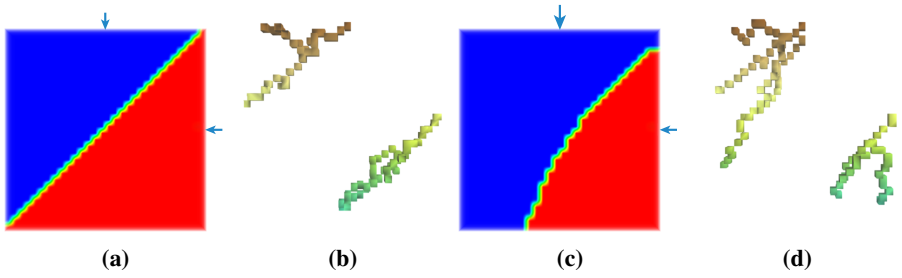
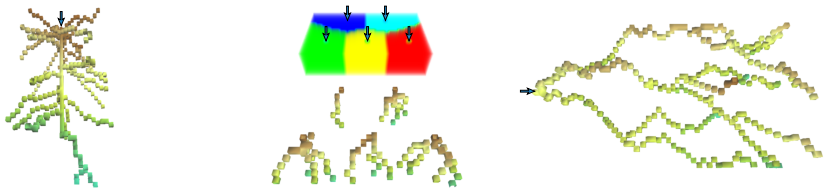


Figure 5. Pressure zones in protochannel development: (a)–(b) two sources with equal pressure; (c)–(d) pressure of one source is higher.

simulates the effect of a larger pressure capturing more area before equalization, as demonstrated in Figure 5. The increase in area allows the protochannel system of the source with higher pressure to also become larger.

We implement the second part of our protochannel model using the Ford–Fulkerson flow network algorithm, which computes maximum flow in a network subject to capacity constraints [Cormen et al. 2001]. We have chosen this algorithm, because according to our model the flow in protochannels is primarily affected by porosity, which acts like a capacity constraint. The neighbor-adjacency relationship in the voxel discretization extends the grid of voxels into a graph, so that the flow algorithm can be applied. Following the path of maximal flow from a source creates a single protochannel. We have found that to create a number of branching protochannels, it is better to instead follow paths of maximal flow from the sinks. Otherwise, it is difficult to judge when branches should separate and the protochannels clump into a mass.



(a) (b) (c)

Figure 6. Protochannel simulation results. Sources are marked with arrows.

The protochannel stage of our simulation can construct several types of protochannel systems exemplified in Figure 6. Figure 6a shows protochannels that branch from a central "core", similar to experiments of McDuff et al. [2010]. The development of protochannels in Figure 6b encounters obstacles due to pressure zones (shown in different colors) of nearby inputs, according to a configuration with multiple ranks of inputs. Figure 6c contains a longer system of protochannels.

3.2. Channel Growth and Linkage Stage

Our model for the channel-growth stage describes the behavior of channels after breakthrough, as well as how they self-organize to form a channel network. Figures 3 and 4 suggest that channel development is guided by pressure differentials that each channel creates in its vicinity. We propose a new model for the effect of pressure, in which change in pressure behaves as a front that propagates similar to an avalanche in sand. To complete the model and make it suitable for procedural modeling, we also incorporate feedback behavior between erosion, pressure, and flow.

Figure 7 illustrates how pressure responds to erosion in our model. In Figure 7a, two channels develop side by side at a similar level of pressure, until one of them

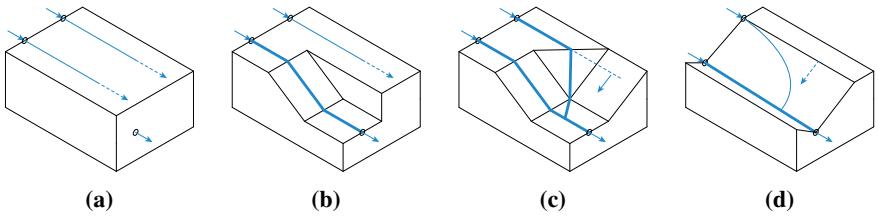


Figure 7. Model for interaction between channels: (a)–(c) pressure drops around one channel causing redirection of another channel; (d) more realistic pressure drop off around the main channel.

undergoes substantial erosion (e.g., breakthrough). Erosion causes a reduction in pressure as the rock matrix drains of water (Figure 7b). If the sharp jump in pressure between the lowered pressure and its original level in the second channel persists, then the rock between the two channels must have an infinite resistance to flow and the channels will never interact.

It is more realistic to consider the "wall" of rock between the channels to be penetrable, similarly to the rest of the rock carrying the channels. In this case, we assume that the resistance of the rock is such that pressure rises linearly with its thickness, leading to an adjustment in pressure between the two channels shown in Figure 7c. The adjustment in pressure can be interpreted in terms of a maximal rate of growth in pressure that the rock can support, so that excess pressure forces water out of the pores until the pressure is reduced. This behavior is similar to avalanching of sand grains on a slope that is steeper than the slope of repose.

In our model, we use two angles to differentiate the rate of growth in pressure inside channels and between them, as shown in Figure 7d. Reduction in pressure such that the pressure gradient points away from a channel is the main factor leading to linkage of channels in our model. We implement pressure updates by using a version of the fast marching algorithm (Section 5), which is ideally suited to simulating propagation of fronts.

To have flow respond to pressure in our model, we use a simulation based on a standard 2D water-column algorithm [Olsen 2004] extended to three dimensions. In our version of the algorithm, each voxel distributes water to its neighbors based on a weight that takes into account pressure difference and remaining capacity. Section 5 provides more details.

Finally, we formulate the relationship between flow and erosion, similar to a model of hydraulic erosion on terrains by Perron et al. [2008], but with several simplifications. The original model of Perron et al. relates the amount of erosion at each site to the local slope of the terrain and the amount of water flow at the site, which is determined based on the size of the area that contributes water to it. In our formulation, we also calculate the amount of erosion based on the local amount of flow f,

but we instead use our version of the water-column algorithm to compute the flow. Since the water-transport scheme directs flow to neighbors with a greater difference in pressure, it is redundant to incorporate terms based on pressure gradient into the equation for erosion. The following simplified equation for erosion, where R stands for the porosity of a voxel, is the result:

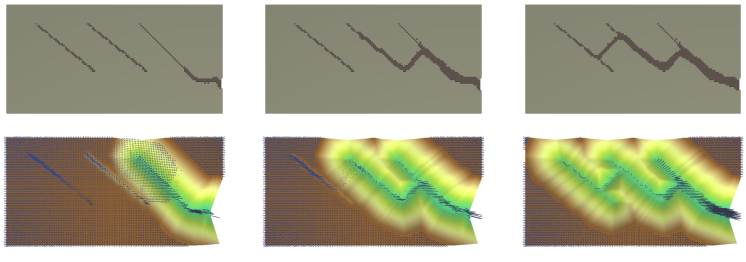
$$\frac{\partial R}{\partial t} = \min(w_0 f^m, 1 - R). \tag{1}$$

The simulation space constrains R and the amount of material in a voxel, defined as $1 - R$, to be in the range $[0, 1]$. The exponent m is typically in the range $(0, 1]$ with smaller values increasing the amount of erosion inside smaller channels. The factor w_0 controls the amount of erosion. To calculate f, the amount of flow, we use the following equation:

$$f = (w_a f_a + f_c). \tag{2}$$

The water-column algorithm provides the value f_c as the local magnitude of flow. Computing the amount of erosion based on this value alone will have a tendency to produce channels that are as narrow as the discretization density allows. For the channels to have plausible width, channels with large flows must collapse their walls and capture nearby flows. To achieve this effect, our simulation blends a fraction of f_c to nearby uphill neighbors, producing the value f_a at each site. The factor w_a is an additional parameter that makes nearby flows contribute more to erosion.

To illustrate the behavior of our simulation of the channel-growth stage, we first provide examples using a 2D version of the simulation, which makes it easier to visualize pressure, flow, and the forming channels simultaneously. The top row of Figure 8 contains a sequence of snapshots of developing channels, while the bottom row contains the corresponding flow (blue bars) and pressure surface (visualized as a terrain with green color mapping to the lowest value). The initial conditions for this simulation are three protochannels constructed synthetically. There are three sources,



(a) **(b)** **(c)**

Figure 8. Channel-growth simulation in 2D. Top row: forming channels; bottom row: pressure and flow.

(a) (b) (c)

Figure 9. Alternative mode of channel growth (in 2D) without formation of phreatic loops.

which are located at the tops of the protochannels, and one sink, which is located along the right boundary of the domain. Note that channel growth causes them to link up in a sequence by forming phreatic loop patterns. This behavior is possible due to self-organization of pressure and flow that causes redirection of flow from one channel towards another.

Figure 9 demonstrates that our model of channel growth captures the alternative mode of channel development that does not result in the formation of phreatic loops. The alternative behavior occurs when the resistance of the rock matrix is low and the protochannels are able to reach the sink without first going through a neighbor. In our model, resistance to flow corresponds to increase in pressure with rock thickness, which is different for rock with and without a present channel. Therefore, to specify low resistance in our model, it is necessary to set the pressure growth rate for channel-free rock to be low. So the configuration of channels in Figure 9 emerges when the rates are set to $0.0025 : 0.001$ (for rock without and with channels, respectively), as compared to $0.05 : 0.001$ for Figure 8. However, the condition that the resistance of the rock matrix without channels is low also implies that development towards a sink will be preferable over linking up with a neighboring channel. Our model supports this interpretation and the result of the simulation remains the same as long as the ratio of the rates is not changed.

4. Results and Evaluation

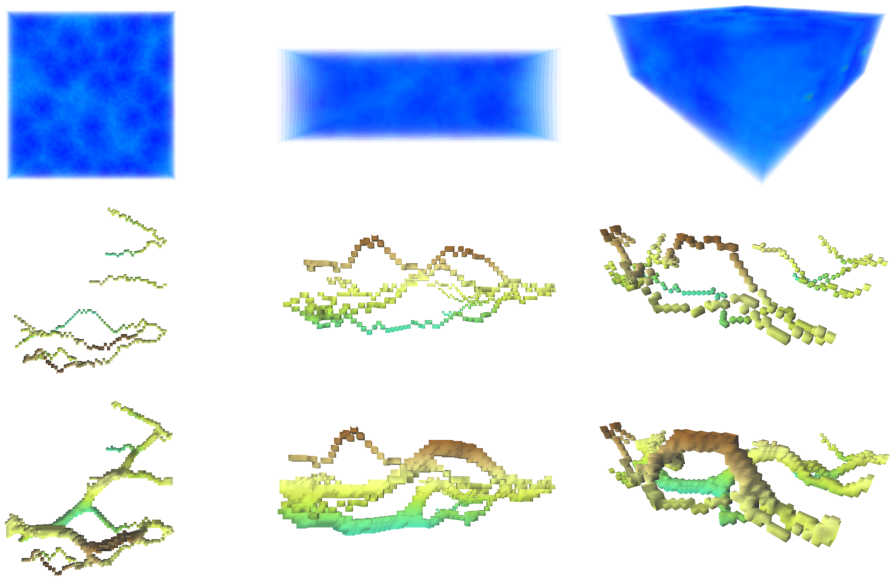
To model realistic cave-like channel networks, we begin by specifying initial conditions in the form of a voxel volume that represents rock with fractures and contains voxels marked as sources. Afterwards we apply the protochannel simulation stage, mark some cells as sinks, and apply the channel simulation stage. In Figures 10–12 the result of the first simulation stage is shown with additional erosion, so that protochannels are visible. In the second stage, the porosity of the formed protochannels is reset to their original values, so that new channels become visible and grow due to the action of the channel erosion simulation only.

To evaluate the cave-like channel networks constructed with our method, we compute *morphometric indices*, or aggregate measures of shape, that have been used to compare different sets of natural caves. The indices can show not only whether pro-

cedurally generated caves possess some properties of real caves, but also whether the properties taken in combination fit a particular type of cave.

4.1. Caves Created Using Two-Stage Simulation

The cave shown in Figure 10 contains three sources that, in the first simulation stage, produce protochannels that branch away from the source locations. The second stage replaces the protochannels with larger channels that converge to a single sink area, which is composed of multiple cells. The channels corresponding to sources that are farther from the sink area are able to reach it due to being captured as tributaries. The initial porosity matrix contains spherical regions in which porosity decreases towards the center. These obstacles cause both types of channels to meander, producing several noticeable arches (visible in the side and perspective views).

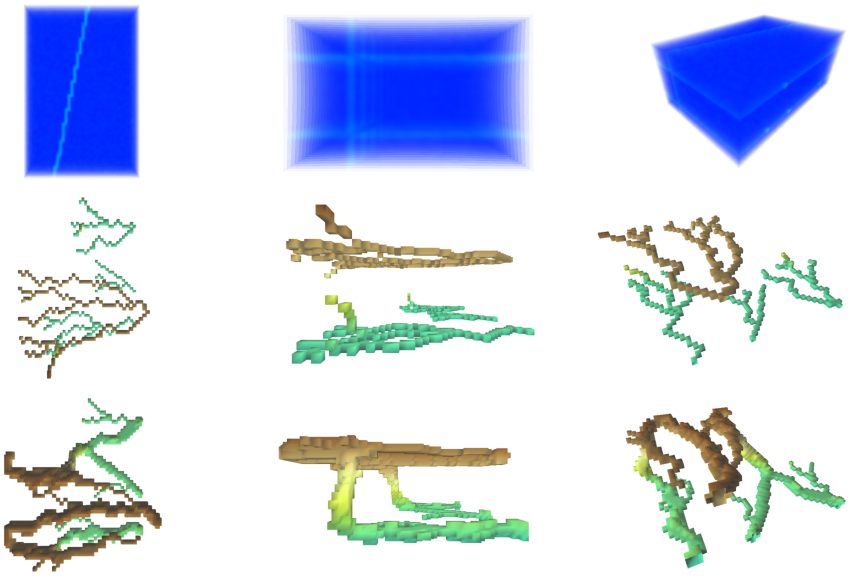


(a) top **(b)** side **(c)** angled

Figure 10. Cave with three sources that form channels towards a single sink area.

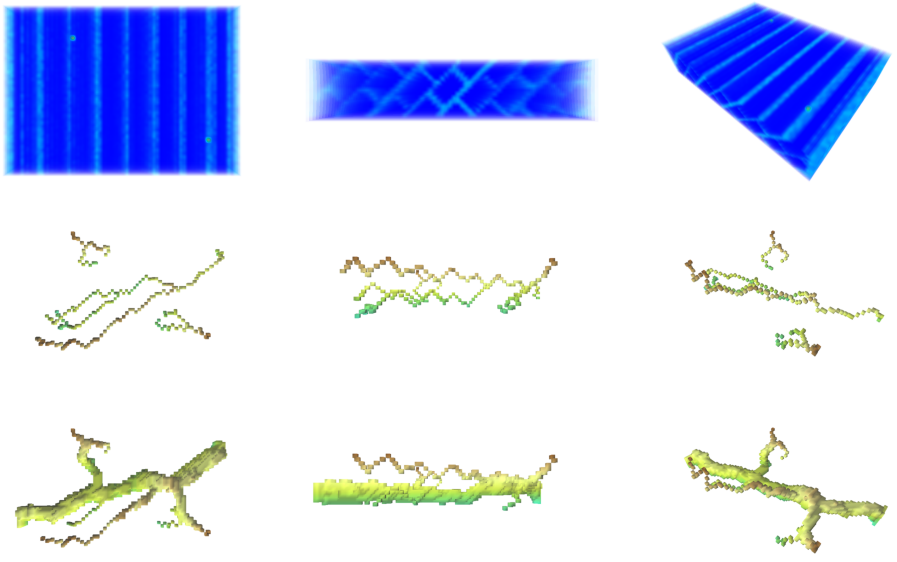
The cave shown in Figure 11 contains four sources and two sink areas. It develops according to the presence of two horizontal bedding planes and a perpendicular fracture plane, which cause both protochannels and cave passages to stay relatively planar as they follow the pattern of fractures. In particular, there are two nearly vertical passages that coincide with the location of the vertical fracture plane.

The initial conditions for the cave shown in Figure 12 represent a configuration of bedding planes and fractures similar to the rock matrix in Figure 11, but with a denser pattern of fractures and the bedding planes inclined. The cave of Figure 12 forms primarily due to the action of a single source area (i.e., several neighboring



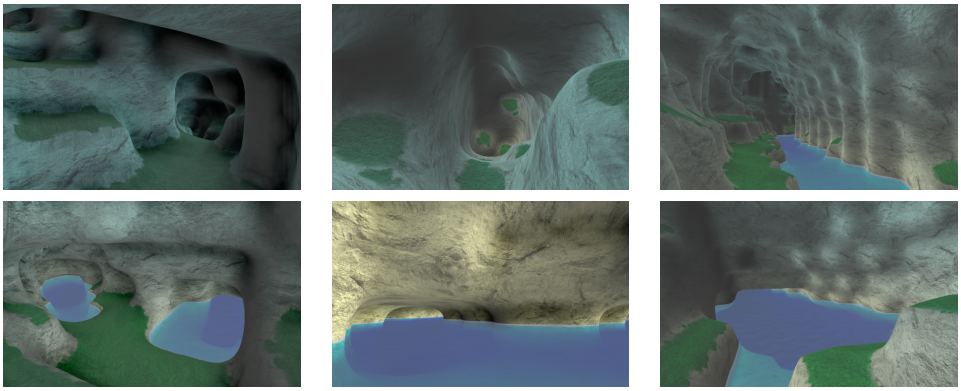
(a) top **(b)** side **(c)** angled

Figure 11. Cave with four sources, two sink areas, and a pattern of perpendicular fractures.



(a) top **(b)** side **(c)** angled

Figure 12. Cave with one source area, one sink area, two weaker sources, and a pattern of fractures and bedding planes.



(a) Figure 10 **(b)** Figure 11 **(c)** Figure 12

Figure 13. Caves rendered with textures and screen-space water.

voxels introducing water into the system), which is paired with a sink area in the opposite wall of the simulated volume. The result is a cave passage that is wider than those in the previous figures. Additionally, there are two weak sources at the top of the simulated volume that produce tributaries of the main channel. In nature, the large underground source could be originating at a nearby lake, while the two weak sources could result from water collecting at the ground level and entering through the inclined bedding planes, which are exposed at the surface.

Figure 13 shows a different type of visualization of the interior of the caves with adaptive subdivision and texturing. The textures are applied procedurally based on slope and composited based on masks that are generated using Perlin noise. In addition, some of the images feature a water surface, which is rendered in screen space. The transparency and color of the water takes depth into account.

Table 1 lists the simulation parameters for caves in Figures 10–12. The ratio of the rates of pressure growth outside and inside of channels ($P_{\text{rate}} : P_{\text{rate}}^{\text{channel}}$) is higher in this set of simulations than in the simulations of Figures 8 and 9, because the resolution of the voxel simulations is lower, while the channels should drain roughly proportional regions. Section 5 provides more information about the grid sizes of the cave simulations. The parameters w_0, w_a, and m are used in Equations (1) and (2).

	P_{rate}	$P_{\text{rate}}^{\text{channel}}$	w_0	w_a	m
Fig. 10	0.50	0.001	0.15	0.25	1.0
Fig. 11	0.50	0.001	0.15	0.25	1.0
Fig. 12	0.85	0.001	0.10	0.40	1.0

Table 1. Simulation parameters for modeled caves.

4.2. Evaluation of Results

Morphometric indices are meaningful parameters of cave shape, because their value ranges can be used to distinguish some common cave types, as shown by Klimchouk [2006] and Frumkin and Fischhendler [2005]. The indices are also suitable for analysis of procedurally-generated caves. We compute the following quantities: A_{field}, which is the area of the cave field, or the polygon surrounding the cave plan; V_{block}, which is the volume of the cave block, estimated as the product of A_{field} and the vertical range of the cave; *network density*, or the ratio of the cave's length to A_{field}; *areal coverage*, or the ratio of the cave's area to A_{field}; *porosity*, or the ratio of the cave's volume to V_{block}; W, which is the average width of the cave's passages, estimated as the ratio of the cave's area to its total length.

	unconfined		confined	
	range	average	range	average
network density $\left(\frac{m}{m^2}\right)$	$0.012 - 0.023$	0.017	$0.049 - 0.406$	0.167
areal coverage $(\%)$	$3.77 - 8.55$	6.17	$13.56 - 58.51$	29.74
porosity $(\%)$	$0.15 - 0.77$	0.40	$1.1 - 12$	5.0
$W\ (m)$	$2.52 - 4.56$	3.77	$0.65 - 4.53$	2.04

Table 2. Morphometric analysis of Klimchouk [2006] with optimized cave fields.

Tables 2 and 3 present value ranges of the indices that were determined by Klimchouk and Frumkin and Fischhendler, respectively, for the sets of caves that they analyzed. The first set demonstrates the difference between unconfined and confined caves and also analyzes caves that are much longer than those of the second set, which compares chamber and confined caves. The first set also uses a procedure that makes the cave field fit around the cave plan tighter than a rectangle, resulting in an optimized value of A_{field}.

	chamber	confined
network density $\left(\frac{m}{m^2}\right)$	$0.01 - 0.25$	$0.07 - 0.35$
areal coverage $(\%)$	$20 - 80$	$10 - 40$
$W\ (m)$	$2 - 100$	$0.9 - 8.0$

Table 3. Morphometric analysis of Frumkin and Fischhendler [2005] with rectangular cave fields.

Table 4 contains morphometric indices of the channels generated with our method. The main conclusion is that the channels fit well into the confined cave category. The confined character is not surprising since, on the one hand, we put more sources than sinks in our simulations to force channels to link up and produce more interesting caves, and, on the other hand, our simulation causes growing cave passages to realistically exploit weaknesses in the rock matrix, which has the effect of preventing unconfined behavior.

	Figure 10	Figure 11	Figure 12
network density $\left(\frac{m}{m^2}\right)$	0.317 (0.122)	0.462 (0.179)	0.277 (0.105)
areal coverage (%)	46.51 (17.94)	77.54 (30.05)	63.66 (24.05)
porosity (%)	5.59 (2.16)	8.74 (3.39)	14.06 (5.31)
W (m)	2.28	1.95	3.33

Table 4. Morphometric indices of modeled caves. Values in parentheses use rectangular cave field.

However, the generated caves fit the second data set much better than the first, since the second and third caves exceed the ranges of the first data set for two of the indices. This occurs for two different reasons, since the morphology of the third cave is dominated by a single channel unlike the more elaborate networks of the other two caves. The second cave has network density and areal coverage that are too high as compared to the first data set, because its length and area are slightly too high for its value of A_{field}, which means that the cave has relatively long passages that are clustered together. The comparatively compact shape of the third cave lowers its network density, but increases its areal coverage and porosity to be slightly outside the range of the first data set. Optimization of A_{field} in this case brings both area coverage and porosity closer to 100%, especially if the passages are not spread in the vertical direction.

Since the first data set consists of significantly longer caves than the second one, it is likely that comparison of generated caves with the first data set can be improved if longer caves are generated. Note that slight deviation from the first data set does not affect the categorization of caves based on the indices, since the differences between index values of different categories of caves can be as large as an order of magnitude.

5. Implementation and Performance

In the implementation of our simulation we use several variants of the fast marching algorithm described by Sethian [1995]. The algorithm simulates front propagation on a discrete grid by computing values of $T(x)$, or time of arrival of the front at x, in a narrow band around known values. The updates take place in an "upwind" order, from smaller values of T to larger ones, which is similar to Dijkstra's algorithm.

For the protochannel stage of our simulation we use the standard version of the fast marching algorithm. For the channel-growth stage of our simulation, we use fast marching to determine when updates to pressure should occur, so that they happen in the optimal "upwind" order.

Another essential subroutine in our simulation is our variant of the water-column algorithm, extended to work on a 3D grid with each site having 26 neighbors. Olsen [2004] describes several 2D variants of the algorithm. Our variant is outlined

```
1  for each neighbor N of S {
2      if (N.capacity > N.contents && N.pressure < S.pressure)
3          N participates in distribution
4  }
5
6  distributed = 0
7
8  for each neighbor N of S {
9      if (N participates) {
10         update = weight(pressure difference) * (N.capacity - N.
               contents)
11         update -= max(distributed + update - S.contents, 0)
12         distributed += update
13         N.contents += update
14     }
15 }
16 S.contents -= distributed
```

Listing 1. Voxel-based water transport algorithm.

in Listing 1 using pseudocode. In the 2D versions of the algorithm, water flow occurs according to two variables: the height of the terrain and the amount of water at each site. In our version of the algorithm, water flow depends on the remaining capacities of the sites and pressure differences between them, as well as on the amount of water they contain.

Table 5 summarizes the performance of both stages of our simulation. We chose a uniform grid for simplicity, but, as Figure 8 indicates, there are sites whose values are never updated, because no channels develop in their vicinity. Therefore, performance can be likely improved with an adaptive scheme. The code was written in C++, compiled as a single-threaded 32-bit application, and executed on AMD Athlon 4400+ 2.5 GHz.

	grid size	stage 1	stage 2	
		runtime	iterations	runtime
Figure 10	60×60×20	1m 43s	240	22m 30s
Figure 11	40×60×25	1m 3s	180	9m 39s
Figure 12	70×50×15	4m 6s	180	12m 41s

Table 5. Performance of cave simulations.

6. Conclusion

We have presented an expressive procedural modeling technique that is able to construct complex 3D networks of cave-like channels based on a small number of simu-

lation parameters, locations of sources and sinks, and a rock-porosity matrix. Emergence of complex shapes, such as tributary-capture patterns, in the produced channel network is the result of channel self-organization according to our model of channel development. Additionally, our simulation provides for competition for space between protochannels and enlargement of channels based on a water-flow simulation.

References

BOGGUS, M., AND CRAWFIS, R. 2009. Procedural Creation of 3D Solution Cave Models. Tech. rep., Ohio State University. TR19. URL: ftp://ftp.cse.ohio-state.edu/pub/tech-report/2009/TR19.pdf. 11

BOGGUS, M., AND CRAWFIS, R. 2010. Prismfields: a Framework for Interactive Modeling of Three Dimensional Caves. In *ISVC*, 213–221. URL: http://dx.doi.org/10.1007/978-3-642-17274-8%5F21, doi:10.1007/978-3-642-17274-8_21. 11

CORMEN, T. H., LEISERSON, C. E., RIVEST, R. L., AND STEIN, C. 2001. *Introduction to Algorithms*, 2nd ed. The MIT Press, Cambridge, MA, 643–664. 17

CUI, J., CHOW, Y.-W., AND ZHANG, M. 2011. Procedural Generation of 3D Cave Models with Stalactites and Stalagmites. *International Journal of Computer Science and Network Security 11*, 8, 94–101. URL: http://www.uow.edu.au/~minjie/pub-ps.dir/IJCSNS-2.pdf. 12

FORD, D., AND WILLIAMS, P. 2007. *Karst Hydrogeology and Geomorphology*. John Wiley & Sons, Ltd, New York. 12, 15

FRUMKIN, A., AND FISCHHENDLER, I. 2005. Morphometry and Distribution of Isolated Caves as a Guide for Phreatic and Confined Paleohydrological Conditions. *Geomorphology 67*, 457–471. URL: http://geography.huji.ac.il/personal/Frumkin/pdf/Geom%20Isolated%20caves2005.pdf, doi:10.1016/j.geomorph.2004.11.009. 24

HALLIDAY, W. R. 2004. Volcanic Caves. In *Encyclopedia of Caves and Karst Science*, J. Gunn, Ed. Taylor & Francis Books, New York. 15

KLIMCHOUK, A. 2006. Unconfined Versus Confined Speleogenetic Settings: Variations of Solution Porosity. *International Journal of Speleology 35*, 1, 19–24. URL: http://scholarcommons.usf.edu/cgi/viewcontent.cgi?article=1200&context=ijs, doi:10.5038/1827-806X.35.1.3. 24

MCDUFF, D., JACKSON, S., SHUCHART, C., AND POSTL, D. 2010. Understanding Wormholes in Carbonates: Unprecedented Experimental Scale and 3D Visualization. *Journal of Petroleum Technology 62*, 10, 78–81. URL: http://dx.doi.org/10.2118/129329%2DJPT, doi:10.2118/129329-JPT. 13, 17

OLSEN, J. 2004. Realtime Procedural Terrain Generation. Tech. rep., University of Southern Denmark. 18, 25

PERRON, J. T., DIETRICH, W. E., AND KIRCHNER, J. W. 2008. Controls on the Spacing of First-Order Valleys. *Journal of Geophysical Research 113*. F04016. URL: http://web.mit.edu/perron/www/files/2007JF000977.pdf, doi:10.1029/2007JF000977. 18

PEYTAVIE, A., GALIN, E., GROSJEAN, J., AND MERILLOU, S. 2009. Arches: a Framework for Modeling Complex Terrains. *Computer Graphics Forum 28*, 2, 457–467. URL: http://arches.liris.cnrs.fr/publications/articles/EG2009%5FArchesAFrameworkModelingComplexTerrains.pdf, doi:10.1111/j.1467-8659.2009.01385.x. 12

SETHIAN, J. A. 1995. Theory, Algorithms, and Applications of Level Set Methods for Propagating Interfaces. Tech. rep., University of California, Berkeley. URL: https://math.berkeley.edu/~sethian/2006/Papers/sethian.actanumerica.1995.pdf. 25

SMART, C. 2004. Glacier Caves and Glacier Pseudokarst. In *Encyclopedia of Caves and Karst Science*, J. Gunn, Ed. Taylor & Francis Books, New York. 15

TORTELLI, D. M., AND WALTER, M. 2009. Modeling and Rendering the Growth of Speleothems in Real-time. In *GRAPP*, 27–35. URL: http://www.cin.ufpe.br/~dmt/paper52.pdf. 12

Index of Supplemental Materials

For clarification of implementation details, the paper is supplemented with C++ source code of the full channel network simulation. The source code is commented and includes examples of creating various kinds of channels. However, the provided code does not include a visualization component and will not function as a stand-alone application. No guarantees are made regarding the functionality of the provided code, as it has been reorganized and simplified from the original implementation.

The code adheres to the following naming convention: declarations related to the channel simulation are prefixed with `vxc` for "voxelized channel simulation," while some other data structure names are prefixed with `vx` for "voxels." Additionally, traversal procedures that use fast marching have the prefix `FM`, while subroutines related to the flow network solution have the prefix `FNET`. The code consists of the following source files.

`vxc_common.h`: Auxiliary code and declaration of simulation settings. The simulation requires several common data structures, such as 3D array (`vx_array`) and heap (`vxc_heap`).

`vxc_fwk.h` and `.cpp`: The core of the simulation.

`vxc_sim.h` and `.cpp`: Code that configures specific usage scenarios of the simulation and creates initial conditions.

The simulation operates on a 3D array whose cells contain simulation data, such as pressure and porosity values. The porosity values represent a domain that contains a porous matrix and any developing channels. These and similar simulation data can be used for visualization

directly. However, the framework also uses a 3D array of `float` to hold temporary values for visualization, because a host application is likely to have a facility for displaying such an array.

Three types of simulations are possible: those that use only the first (protochannel) stage, those that begin directly with the second stage (most likely with synthetically created protochannels), and those that use both stages in sequence. In the latter case, an interim configuration step is required to prepare the domain for the second stage. The first stage (`vxc_fwk::stage1_water_sim()`) operates according to the following steps.

1. Divide the domain into zones (`vxc_fwk::FM_pressure_division()`) according to the placement of sources and their pressure, as in Figures 5 and 6b.
2. Initialize (`vxc_fwk::FNET_init()`) and solve (`vxc_fwk::FNET_main()`) the flow network inside each zone independently. Note that the initialization step is logically a part of creating initial conditions for a specific construction.
3. After solving each flow network create protochannels (`vxc_fwk::protochannels()`) inside the appropriate zone.

The second (channel) stage is captured in the function `vxc_fwk::stage2_water_sim()`. The code iterates the water distribution algorithm from Listing 1 and applies Equation (1) to increase the porosity of the cells according to their flows.

Author Contact Information

Alex Pytel
University of Waterloo
200 University Av. W
Waterloo, ON N2L 3G1
apytel@uwaterloo.ca

Stephen Mann
University of Waterloo
200 University Av. W
Waterloo, ON N2L 3G1
smann@uwaterloo.ca

A. Pytel and S. Mann, Procedural Modeling of Cave-like Channels, *Journal of Computer Graphics Techniques (JCGT)*, vol. 4, no. 2, 10–29, 2015
http://jcgt.org/published/0004/02/02/

Received:	2014-11-27		
Recommended:	2015-04-07	Corresponding Editor:	Cindy Grimm
Published:	2015-05-29	Editor-in-Chief:	Morgan McGuire