DOI: 10.1111/cgf.13133

EUROGRAPHICS 2017 / L. Barthe and B. Benes
(Guest Editors)

Volume 36 (2017), Number 2

Chamber Recognition in Cave Data Sets

Nico Schertler, Manfred Buchroithner, and Stefan Gumhold

TU Dresden, Germany
{nico.schertler, manfred.buchroithner, stefan.gumhold}@tu-dresden.de

Abstract

Quantitative analysis of cave systems represented as 3D models is becoming more and more important in the field of cave sciences. One open question is the rigorous identification of chambers in a data set, which has a deep impact on subsequent analysis steps such as size calculation. This affects the international recognition of a cave since especially record-holding caves bear significant tourist attraction potential. In the past, chambers have been identified manually, without any clear definition or guidance. While experts agree on core parts of chambers in general, their opinions may differ in more controversial areas. Since this process is heavily subjective, it is not suited for objective quantitative comparison of caves. Therefore, we present a novel fully-automatic curve skeleton-based chamber recognition algorithm that has been derived from requirements from field experts. We state the problem as a binary labeling problem on a curve skeleton and find a solution through energy minimization. A thorough evaluation of our results with the help of expert feedback showed that our algorithm matches real-world requirements very closely and is thus suited as the foundation for any quantitative cave analysis system.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Geometric algorithms, languages, and systems

1. Introduction

In the past few years, great efforts have been made in the field of speleology (science of caves) to acquire highly detailed 3D maps of underground cave systems, mostly with the help of terrestrial laser scanners [aE01,BG09,GAMNGL*09,MRB*15]. The resulting 3D models can then be used for analysis. In this paper, by *cave* we mean the entirety of a cave system, i.e. everything that is below the ground. Figure 1 shows an example cave.

A cave's size (e.g. measured by its volume) is of great interest to speleologists because extreme sizes may be caused by unique geological formations and also bear significant touristic attraction. Although computing this volume from a mesh is straight-forward, the total volume of a cave is of little significance in most cases. The entities whose volume is interesting to speleologists are cave chambers, i.e. enlargements of the cave's interior [Fie99] (possible chambers in Figure 1 are marked red). Therefore, volume calculation requires identification of chambers.

Volume and other quantitative measures that are derived from identified chambers are objective quantities that describe the chamber. This criterion is not met if experts identify the chambers because this approach introduces a subjective bias as seen in Figure 1. In this paper, we present an automatic chamber recognition algorithm, which we derived strictly from requirements stated by experts in the field. Because our algorithm is fully automatic, it has no subjective bias and can be used as a basis to generate objective quantitative measures.

We use the reconstructed surface from cave surveying expeditions, represented as a triangle mesh, as input for our algorithm. It is sufficient for the input mesh to have a low spatial resolution because details and features that are only present at higher resolutions do not have an impact on chamber recognition. Furthermore, we require the input mesh to be watertight and manifold in order to support skeleton extraction as we will explain in Section 3.4. We produce such meshes with Poisson Surface Reconstruction [KBH06] on the original point cloud.

The task of chamber identification is a segmentation problem, where segments represent chambers or the passages that connect them. In contrast to most other mesh segmentation algorithms, we do not segment the surface directly as this may cause inconsistencies between the sides of a chamber. Instead, we derive a curve skeleton [CSM07] from the shape and segment the vertices of the curve skeleton. The segmentation is then projected back onto the surface. This procedure ensures consistency of opposite surface vertices because they correspond to the same skeleton vertex. The segmentation is guided by a size measure defined on the skeleton vertices. We describe the presence of entrances (i.e. transitions between chambers and passages) in a probabilistic framework and derive the segmentation as the minimizer of a quadratic energy function. The resulting algorithm is invariant to rigid body transformations as well as to scaling.

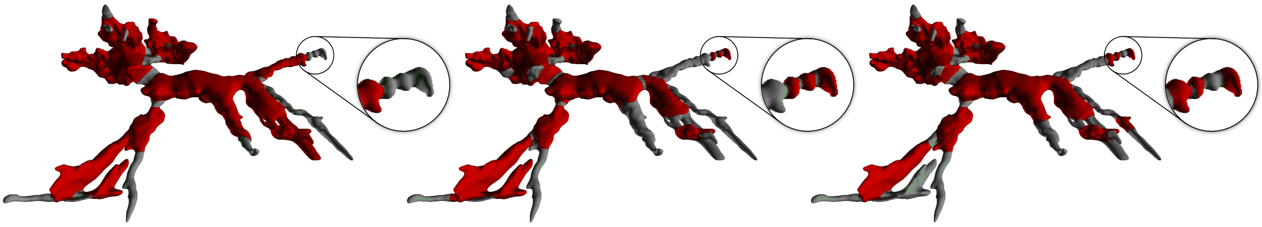


Figure 1: *A model of the Simud Puteh cave (Borneo) with chambers (red) identified by three different experts. The core parts of chambers are matched in all three segmentations. However, differences occur in controversial areas such as the far right branch.*

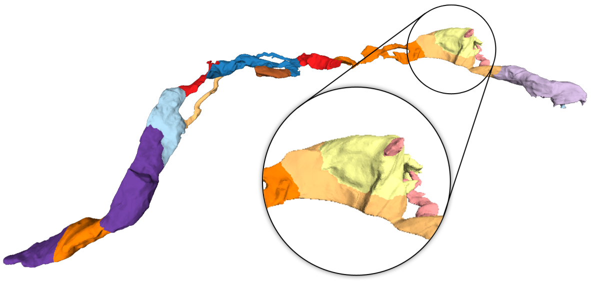


Figure 2: *SDF-based segmentation of the Eisriesenwelt cave (Austria). Strong narrowings as the entrance of the beige passage in the middle can be recognized very well. Chambers may consist of multiple segments as can be seen from the beige/yellow/pink segments on the right.*

Unfortunately, the field of speleology does not offer a rigorous definition of the term *chamber*, though experts have a common understanding of chamber characteristics. However, this understanding usually diverges in controversial areas of a cave as shown in Figure 1. In order to match our algorithm design with the common sense in speleology, we worked very closely together with experts in the field and evaluated the results quantitatively against feedback from experts.

2. Related Work

The problem of mesh segmentation is widely studied and a variety of approaches exist. For a detailed overview, we refer the reader to the surveys by Attene et al. [AKM*06] and Shamir [Sha08]. In the following, we review a selection of relevant work as well as more recent publications.

Although chamber recognition is a segmentation problem, mesh segmentation is a slightly different problem in the sense that most algorithms try to find a partitioning of a mesh that matches the human recognition system. The basis for this kind of segmentation has been laid by Hoffman and Richards [HR84], who stated that the human recognition system tends to partition a shape in such way that the boundaries coincide with negative minima of any of the principal curvatures (i.e. concave regions) and the according curvature lines (*minima rule*). As we will explain in Section 3.1, this rule does not always apply to chamber recognition, especially because concave creases are not necessarily entrances and curvature lines may fail to capture the entrance when the widening happens in a single direction. Furthermore, most approaches segment the shape into a number of individual segments without regarding semantics (i.e. whether or not the segment is a chamber). Therefore, it is usually not possible to easily extend existing approaches to chamber recognition, especially because chambers may be represented by multiple segments. Figure 2 shows the segmentation result of SDF-based segmentation [SSCO08] of the Eisriesenwelt caves. As can be seen, the segmentation captures narrowings very well. However, an interpretation of the segmentation with respect to chamber recognition is not directly possible. Even if a parameterization of the algorithm is found that captures all entrances, a more sophisticated approach would be necessary to perform the segmentation in a consistent way (i.e. one that labels all corresponding segments as a single chamber).

Graph-Based Approaches on Mesh Connectivity use the mesh graph and geometrical measures on the graph to run segmentation. SDF-based segmentation [SSCO08] as shown in Figure 2 defines the *Shape Diameter Function* on every surface facet, which serves as a local size measure. After a soft clustering of faces with respect to their SDF, a graph cut-based optimization is used to find the final segmentation. Similar clustering approaches are used with other attributes such as dihedral angle, curvature, convexity, etc. For details, please refer to the referenced surveys. Other clustering techniques based on geometrical features have also been proposed, such as Random Walks [LHMR08] and Mean Shift [ZLXH08]. Golovinskiy et al. [GF09] presented a method that does not only regard edges within a single mesh but also correspondences between a set of meshes to achieve consistent segmentations across the set. Au et al. [AZC*12] use geometric features to define scalar functions across the surface and derive borders of the segmentation from isolines of these functions. Wang et al. [WLAT14] define a concavity-aware Laplacian operator on the surface and derive a segmentation from its eigenvectors. Analyzing the behavior of the mesh under several operations can also be used for segmentation. De Goes et al. [DGGV08] consider a diffusion process, whereas Fang et al. [FSKR11] analyze heat flows. All these methods are not directly applicable to chamber recognition due to the reasons explained in the previous paragraph.

Part-based approaches consider the volume enclosed by the surface instead of segmenting the surface directly. Fitting-based approaches [WY11, YWLY12] try to fit quadrics or spline surfaces to the geometry and use the fitting error for segmentation. Solomon et al. [SBCBG11] analyze the intrinsic symmetry of shapes with approximate Killing vector fields to find a segmentation. These approaches are inappropriate for chamber recognition due to the irregularity of caves. Agathos et al. [APPS10] use feature points to classify the shape into a body part and protrusible parts. The assumption of a single body part, however, contradicts chamber recognition since a cave usually contains multiple chambers (which would be represented by body parts). A different approach is shown by Feng et al. [FJT15], who use the medial axis in a volumetric data set to explore the space of possible cuts and derive the final segmentation from statistical analysis of the cut space. This approach is also inapplicable because it lacks semantic information of segments. Kaplansky et al. [KT09] show how to improve a coarse segmentation based on optimization of level set functions.

Tubular Part Extraction [MPS*04a, MPS*04b, GDB06, MST10] as a special case of part-based extraction is the family closest to chamber recognition. Similarly to our approach, Dey et al. [DS06] also use a curve skeleton to segment a shape, although with a simple thresholding procedure. Even this family solves a different problem because chambers can have tubular shapes as well. Additionally, passages are not necessarily generalized tubes as they can be very irregular [Pal07].

Data-Driven Approaches use Machine Learning methods to generate a segmentation without explicitly modeling the requirements. Kalogerakis et al. [KHS10] and Lv et al. [LCHB12] use a set of manually labeled training shapes in a conditional random field framework to learn the segmentation objective based on geometric features. Instead of learning the segmentation, Benhabiles et al. [BLVD11] learn the presence of a boundary.

3. Chamber Recognition

In this section, we explain the core concepts of our chamber recognition algorithm. We start with a review of cave characteristics that lay the foundation of our algorithm. We continue with a formal problem statement, then give an overview of the algorithm, and finally elaborate on the details.

3.1. Geometry of Caves

In general, caves exhibit a very irregular surface without symmetry [Bec80, SRFVP13]. Due to erosion, there may be several creases in the surface, which do not necessarily coincide with entrances (i.e. the minima rule does not apply). There may be large differences in size, both between different caves and even within the same cave, which requires the algorithm to be scale-independent.

In the field of speleology, there is no definition of chambers. In order to establish a guide line, we conducted several interviews with speleology experts and found the following characteristics to be the common sense. Speleologists identify chambers based on the cave's perceptible size for an observer that is located in the cave, which is a subjective measure of the cave's local extents. For a cylindrical cave and an observer that is located inside the cylinder, the perceptible size correlates with the cylinder's radius. In fact, the absolute size is not as important as the size change. An observer that moves through the cave is assumed to pass an entrance if he recognizes a sudden widening of the cave's size, i.e. the presence of chambers is mainly dictated by the presence of entrances. This widening may happen in a single or in both principal directions orthogonal to the observer's path.

3.2. Problem Statement

Given a mesh $M = (V_M, F)$ with vertices V_M and faces $F \subseteq \binom{V_M}{3}$, we want to find a labeling for the vertices $L \in \{\mathcal{C}, \mathcal{P}\}^{|V_M|}$, such that all vertices that belong to a chamber are labeled with $\mathcal{C}$ and all vertices that belong to a passage are labeled with $\mathcal{P}$. Once this labeling is calculated, separate chambers can be extracted by connected component analysis.

3.3. Overview

Figure 3 shows the four steps of our algorithm. The algorithm starts by extracting a curve skeleton from the input mesh, which will serve as the segmentation domain. The curve skeleton is a smooth 1D structure that is centered in the shape. As such, it is a natural path for an observer through the cave, which is needed to evaluate widening and narrowing of the perceptible cave size (cf. Section 3.1). By design, the utilization of a curve skeleton ensures that surface vertices that correspond to the same skeleton vertex receive the same label, which guarantees segmentation consistency along the cross-section of the cave in regions where the skeleton is manifold. Furthermore, the curve skeleton comprises significantly fewer vertices with sparser connectivity than the surface mesh, which allows for very efficient computation of the final solution. We explain the process of skeleton extraction in Section 3.4.

In a second step, the perceptible size is calculated for every skeleton vertex, resulting in a scalar field defined over the curve skeleton. This calculation is based on a combination of ray shooting and extraction of significant enclosing lines on the surface, which results in a measure that matches the subjective local cave size as perceived by an observer located at the according skeleton vertex. We point out the details of this measure in Section 3.5.

The perceptible size field is then used to segment the curve skeleton. We model the segmentation problem as an energy minimization within a probabilistic framework. The key idea is that the presence of entrances (i.e. sudden widenings of the cave) is closely related to the presence of maxima in the second derivative of the perceptible size field with respect to the skeleton, while the first derivative describes the direction of the entrance. This model results in a quadratic energy function, which we minimize using a graph cut-based solver. We explain the structure of the energy function and the process of minimizing it in Section 3.7.

Finally, the segmentation is projected back onto the surface using the skeleton-shape correspondence that is calculated by the skeleton extraction process.

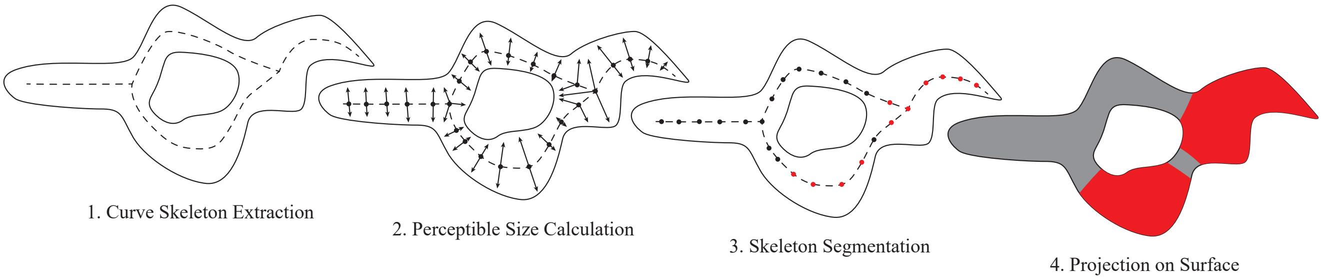


Figure 3: *Overview of the chamber recognition pipeline*

3.4. Skeleton Extraction

From the variety of skeleton extraction algorithms, we found that *Mean Curvature Skeletons* [TAOZ12] perform best on the cave data sets and they allow to choose the degree of centeredness and smoothness. As a requirement, the input mesh must be watertight and manifold. As explained in Section 1, the generation of such meshes is straight-forward.

Since the algorithm relies on the medial axis, we apply HC Laplacian smoothing [VMM99] before extracting the skeleton. Due to the reduction in noise, the medial axis is much cleaner and leads to more expressive skeletons without changing the original mesh too much.

The result of skeleton extraction is a skeleton $S = (V_S, E)$ with vertices V_S and directed edges $E \subseteq V_S^2$, which are oriented arbitrarily for computational reasons (cf. Section 4.2). We subdivide edges that are longer than a given target size to achieve near uniform sampling of the skeleton.

Furthermore, the extraction process produces a correspondence map $C : V_M \rightarrow V_S$. In general, the corresponding mesh vertices for a given skeleton vertex interleave those of its neighbor vertices. Therefore, we clean the correspondence map by assigning mesh vertices to neighbors of their original corresponding skeleton vertex whenever the neighbor is closer to the mesh vertex than the original correspondence. This local optimization influences only the back-projection step and cleans the borders between segments. Figure 4 shows an example of the extracted curve skeleton.

3.5. Perceptible Size

As explained in Section 3.1, the perceptible size inside the cave is a crucial measure for the recognition of chambers. Therefore, we define it as an attribute on the skeleton vertices $size : V_S \rightarrow \mathbb{R}$. In the following section, we present the characteristics of this attribute, which we have derived from discussions with speleologists. The according implementation details are explained in Section 4.1.

The starting point for the development of the measure definition is a skeleton vertex v that is located at the center of a perfectly cylindric cave. In this case, the size should be the cylinder's radius. This can be easily extended to cylinders with irregular cross-sections, in which case the size should be the average radius:

$$size(v) = \frac{1}{2\pi} \int_0^{2\pi} radius(\phi)\, \mathrm{d}\phi, \tag{1}$$

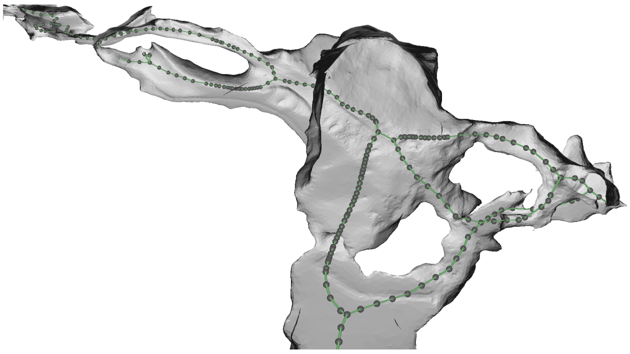


Figure 4: *Visualization of the extracted curve skeleton with edges shown in green and vertices represented as gray spheres. The front faces of the cave surface have been removed in the rendering.*

where $radius(\phi)$ is the radius of the cross-section in the direction of ϕ. Upon closer inspection, the resulting size is the radius of a circle whose perimeter is equal to that of the cross-section. The calculation can be modified to give the radius of a circle whose area is equal to the area of the cross-section by incorporating an exponent $e = 2$:

$$size(v) = \left(\frac{1}{2\pi} \int_0^{2\pi} radius(\phi)^e\, \mathrm{d}\phi \right)^{1/e}, \tag{2}$$

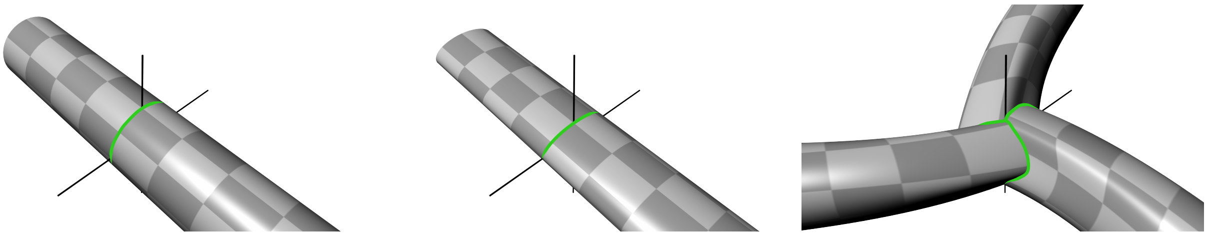
In fact, any exponent can be used in order to match the size measure to the actual perceptible size, e.g. the exponent 1.75 as proposed by Flannery [Fla71] may be used to use the circle's perceptual size. We evaluate the choice of this exponent in Section 6.

We generalize this principle of averaging the radius over a path network across the surface for arbitrarily shaped caves surrounding v. Assume that $radius_v(d)$ is now a spherical field of distances from v to the cave's surface in the direction d. For an arbitrary (not necessarily manifold) path network P on the surface of the unit sphere, the average radius is then:

$$avgRadius(v, P) = \left(\frac{1}{length(P)} \oint_P radius_v(d)^e\, \mathrm{d}d \right)^{1/e}, \tag{3}$$

where both length and the differential are measured using the geodesic distance on the unit sphere:

$$dist(d_1, d_2) = \arccos\langle d_1, d_2 \rangle \tag{4}$$



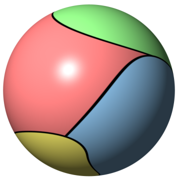
(a) *Cylindrical cave*

(b) *Elliptical cylinder*

(c) *General cave shape with three path segments.*

Figure 5: *Visualization of the averaging path network used for perceptible size calculation for different cave shapes. The path network is visualized in green. The according skeleton vertex is located at the intersection of the two black lines.*

For the cylindric case, the path network is a circle on the unit sphere. Figures 5a and 5b show the averaging network for the two cases discussed so far. We express P as a set of open path segments, where each segment is represented by a discrete sampling. The crucial question is how to define the network on general cave shapes.



As can be seen from the cylindrical cases, the chosen path network is closed and always encompasses the according skeleton vertex completely. More formally, this means the following: The path network partitions the surface of the unit sphere into distinct patches (in the case of the cylindrical examples in exactly two hemispheres). The nearby figure shows these patches for a more general network. The solid angle of each of these patches must be smaller than a threshold $\overline{\theta}_\Omega$. Intuitively, this restricts the size of generated cells of the network, which ensures that the cave part surrounding v is examined in its entirety instead of a single non-representative area. Similarly, a lower bound on the solid angle $\underline{\theta}_\Omega$ is enforced to ensure a minimum distance of path segments and to avoid redundancy as a consequence of near-collapsing cells. In our computation, we define these thresholds not directly but in a 2D plane as simple angles. We will explain the details of this constraint in Section 4.1.

Furthermore, the chosen network is the most compact one, i.e. it is a minimizer of (3). Together with the aforementioned constraints, this completes the definition of the perceptible size:

$$
\begin{aligned}
size(v) := \min_P & \; avgRadius(v, P) \\
\text{s.t. } & P \text{ is closed} \\
\text{s.t. } & \underline{\theta}_\Omega \leq solidAngle(p) \leq \overline{\theta}_\Omega \forall p \in patches(P),
\end{aligned}
\tag{5}
$$

where $patches(P)$ is the set of patches on the surface of the unit sphere induced by the path network P. Figure 5c shows this network for a general cave shape.

We evaluate this equation by finding an optimal path network on the surface and averaging the radius field over it. The starting point for this optimization is a heuristic initialization based on the local minima of the radius field, which is then refined through gradient descent-based optimization. For details, refer to Section 4.1.

3.6. Derived Measures

The perceptible size is the key measure that controls the segmentation. In order to define the segmentation energy function, we derive a few more measures from the perceptible size, most notably the first two derivatives. In the following section, we explain the procedure used to extract these measures.

We define both derivatives of the perceptible size on the edges instead of skeleton vertices. Therefore, the first derivative can be calculated easily using finite differences of the perceptible sizes of the incident vertices. The edge direction serves as a natural direction for the derivative value. Obviously, the derivative in the reversed direction is the negated value, which we will use in a later procedure. Similarly, the second derivative is calculated with central differences from the first derivative. To compute the central difference on an edge, where at least one vertex is a branching, we average the first derivatives over all incoming (incident to the edge's first vertex) and all outgoing (incident to the edge's second vertex) edges and use these two values for the finite difference. For implementation details, please refer to Section 4.2. In order to remove noise and to make the derivative calculation more robust, we apply Gaussian smoothing to both the perceptible size and its first derivative before calculating the next derivative.

From the three measures presented so far, only the first derivative is independent of scale because any uniform scaling cancels out in the differential quotient $\frac{\mathrm{d}size(t)}{\mathrm{d}t}$, where t is the position of a vertex on the skeleton branch, measured in length units. This fact prohibits the direct use of the other measures for an energy function of a scale-invariant chamber recognition algorithm. Therefore, we derive a local scale measure that allows the definition of relative, scale-independent measures. We use this cave scale in the following ways: We express the kernel sizes used for smoothing the perceptible size and its first derivative as a multiple of the scale. And we multiply the second derivative by the cave scale to derive a normalized, scale-invariant second derivative $\widehat{size''}$. In the following, we will refer to this measure as *Normalized Curvature*. The reason for that is the according differential quotient having a single scale measure in the numerator and two in its denominator. By multiplying with another scale measure, all scale dependency cancels out, leaving a scale-invariant value. The advantage of using a local cave scale instead of a global one is its adaptivity to the actual proportions in an observed portion of the cave. As discussed in Section 3.1, caves can exhibit large differences in size, which makes the use of a global scale inappropriate.

We found that the maximum perceptible size in the neighborhood surrounding a specific skeleton vertex v is well suited as the scale measure for v. This neighborhood constitutes vertices that have a geodesic distance smaller than a multiple μ_{scale} of the perceptible size from v. Expressing the distance threshold as a multiple of the size leads to a scale-invariant process. Specifically, with $dist(v_1, v_2)$ being the geodesic distance of two vertices along the skeleton, the cave scale is defined as:

$$scale(v) \leftarrow \max\{size(i) | dist(v,i) \leq \mu_{scale} \cdot size(v)\} \tag{6}$$

This essentially pulls the maximum cave size from all neighbors within a given radius.

3.7. Segmentation

With the definition of derived measures, all necessary input data for the segmentation are specified. In the following, we present the utilized energy function and the according minimization method.

We start by quickly recalling the segmentation objective: Given the curve skeleton S, we want to find a labeling $L_S \in \{\mathcal{C}, \mathcal{P}\}^{|V_S|}$ that specifies for each skeleton vertex if it belongs to a chamber or to a passage. Since the presence of chambers is dictated by the presence of entrances, the energy function has the following general form of a Markov Random Field:

$$\underset{L}{\arg\min} \sum_{(v_i,v_j)\in E} \phi_{i,j}(L_i, L_j), \tag{7}$$

where $\phi_{i,j} : \{\mathcal{C}, \mathcal{P}\}^2 \to \mathbb{R}$ is a per-edge potential that describes in a probabilistic framework how likely a given labeling of the two incident vertices is. Instead of modeling the potential directly, we model the according probabilities and derive the potential from their negative-log space:

$$\phi_{i,j}(L_i, L_j) = -\log \pi_{i,j}(L_i, L_j) \tag{8}$$

There are two ingredients for the definition of the probabilities. The function $\pi_{entrance}$ models the probability of an edge being an entrance. The second function $\pi_{direction}$ models the probability of a possible entrance to be directed along the edge (i.e. the source vertex belongs to a passage and the target vertex to a chamber). These two probabilities can be used to define the probability function for the edge. The probability of the edge not being an entrance is therefore the complement of the entrance probability:

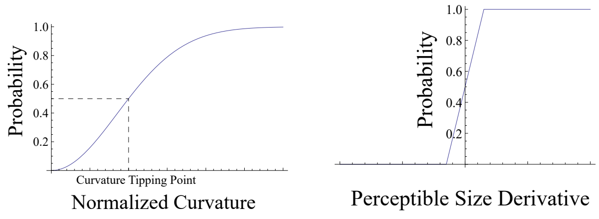
$$\begin{aligned} \pi_{i,j}(\mathcal{C},\mathcal{C}) &= 1 - \pi_{entrance}(\widehat{size''}(i,j)) \\ \pi_{i,j}(\mathcal{P},\mathcal{P}) &= 1 - \pi_{entrance}(\widehat{size''}(i,j)) \end{aligned} \tag{9}$$

The remaining two cases must take the direction into account:

$$\begin{aligned} \pi_{i,j}(\mathcal{P},\mathcal{C}) &= \pi_{entrance}(\widehat{size''}(i,j)) \cdot \pi_{direction}(size'(i,j)) \\ \pi_{i,j}(\mathcal{C},\mathcal{P}) &= \pi_{entrance}(\widehat{size''}(i,j)) \cdot (1 - \pi_{direction}(size'(i,j)) \end{aligned} \tag{10}$$

In the following, we present how we model the partial probabilities.

Like most other mesh segmentation algorithms, which use a curvature measure, we use the second derivative of the perceptible size as the basis for the entrance probability. More specifically, high values of the second derivative correspond to a high probability of the presence of an entrance. This matches the requirement for an entrance to be located at a sudden widening of the cave (cf. Section 3.1). Formally, this probability is defined as follows:



(a) *Entrance probability* **(b)** *Direction probability*

Figure 6: *Visualization of the utilized probability functions.*

$$\pi_{entrance}(\widehat{size''}) = \begin{cases} 0 & \widehat{size''} < 0 \\ 1 - \exp\left(-\frac{\widehat{size''}^2}{2\sigma^2}\right) & \text{otherwise} \end{cases}, \tag{11}$$

where we define the parameter σ based on the curvature tipping point θ_{tip}, i.e. the value of the normalized curvature that results in an entrance probability of 0.5, which is the border line between deciding for a passage or a chamber:

$$\begin{aligned} &\pi_{entrance}(\theta_{tip}) = 0.5 \\ \iff &\sigma = \frac{\theta_{tip}}{\sqrt{2\ln 2}} \end{aligned} \tag{12}$$

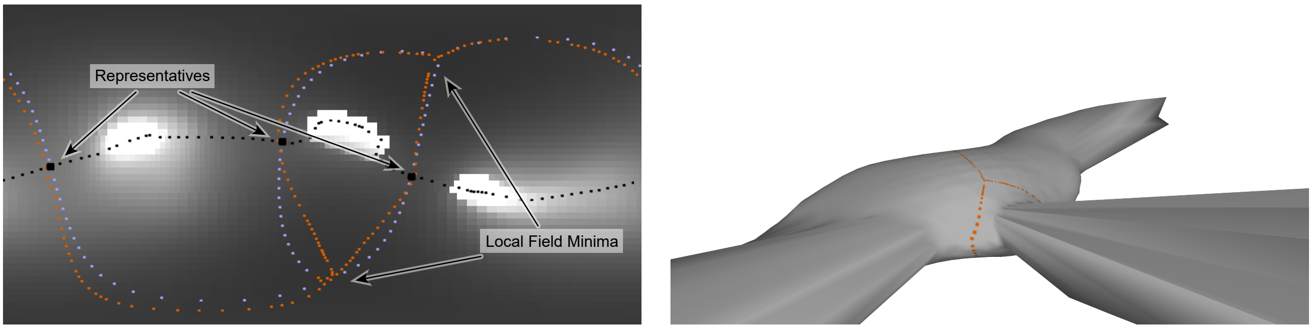
Figure 6a visualizes this function. Note that this function never reaches the probability of 1. This is necessary in order to allow the algorithm to mark edges with high second derivatives as non-entrances to ensure global consistency.

The direction probability specifies the direction of the entrance, by which we mean the direction, in which the label changes from passage to chamber. Naturally, this should be the direction, in which the cave size increases, i.e. the direction, in which the first derivative is positive. However, near local minima of the cave size, the first derivative may vanish. Therefore, we allow a small tolerance θ_{dir} for the direction. The probability of a possible entrance being directed in the same direction as the underlying edge is therefore:

$$\pi_{direction}(size') = \begin{cases} 0 & size' < -0.5\,\theta_{dir} \\ 1 & size' > 0.5\,\theta_{dir} \\ 0.5 + \frac{size'}{\theta_{dir}} & \text{otherwise} \end{cases} \tag{13}$$

Figure 6b visualizes this function, where the slope in the middle is defined by the direction tolerance.

Minimizing the resulting energy is in general NP-hard. The proof for that can be found in the appendix. However, there are only few edges that prevent the problem from being solvable in polynomial time (so called non-submodular edges; for details see the appendix). QPBO (quadratic pseudo-boolean optimization) [RKLS07] is a solver that is well-suited for this kind of energy function. It is a graph-cut based solver that produces a pseudo-



(a) *Intensity-coded parameter space. The purple dots near the orange path network are the initial circular arcs. The ridge line is represented as black dots, where representatives are enlarged.*

(b) *3D visualization of the portion of the cave that is visible from the evaluated vertex (inside the bulge). The path network encompasses the skeleton vertex completely and passes nicely between the three incident passages.*

Figure 7: *Visualization of the sampled radius field for a representative skeleton vertex and* $e = 1$ *as well as the final path network represented as orange dots.*

boolean labeling in the sense that it may leave some vertices unlabeled. However, due to the small number of submodular edges and the sparse connectivity, QPBO is able to label all vertices in all our test data sets. If this was not the case, it would be possible to run a combinatorial solver on the remaining vertices because all labels that are computed by QPBO are part of a globally optimal solution. Since this is not the case for the tested cave data sets, a solution can be computed very quickly (in average in less than ten milliseconds), which makes the solver feasible for interactive exploration of the parameter space.

After calculating the solution on the curve skeleton, a connected component analysis is performed to assign individual indices to every chamber. After that, the solution is projected back onto the surface with the correspondence map.

4. Implementation Details

In the following section, we explain the details of the calculation of the perceptible size and derived measures.

4.1. Computation of Perceptible Size

The basis of the perceptible size is the radius field, which we sample by ray-shooting from the skeleton vertex in several directions. The directions are sampled by a regular sampling scheme based on spherical coordinates that produces nearly uniform solid angles of influence per ray (i.e. there are fewer samples along latitudes close to the poles). In order to keep neighbor queries simple and efficient, we restrict the number of samples along latitudes to powers of two.

The central objective of perceptible size calculation is the generation of the minimal path network. Depending on the exponent chosen in (3), we raise the radius field to the power of e to derive a generic height field. In the following, we explain how we optimize the path network over this generic height field. Once the optimal average height h^* is found, the corresponding average radius is $\sqrt[e]{h^*}$. Since the space of possible path networks on the unit sphere is infinite and their average height is non-convex, the exact computation of the perceptible size is usually not feasible. In the following, we present an algorithm that approximates the optimal average height with satisfying accuracy.

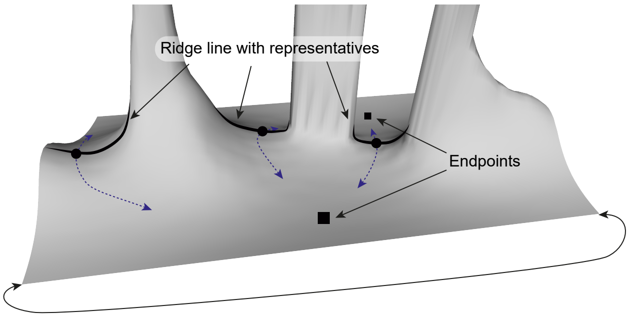


Figure 8: *Height field of Figure 7 unrolled to a planar domain*

As explained in Section 3.5, we express the path network as a set of manifold, open segments that are represented by an ordered point sampling. As a simplification, we assume that all segments share their two endpoints with each other. The steps to calculate the path network are then as follows (the result as well as intermediate steps can be found in Figure 7): First, we find the shared endpoints on the sphere surface, which are usually local minima of the heightfield. Once these endpoints are known, all path segments should run through passes (usually saddle points of the height field) between the endpoints. We found that this approach serves as a good heuristic for minimizing the segment's average height (cf. Figure 8). Therefore, we find the ridge line that separates the two endpoints (black in Figure 7) and select representative points from it, where we would like the path segments to pass through (enlarged dots). For each representative, we then construct a path segment and initialize it with a circular arc between the endpoints through the representative (purple in Figure 7). Finally, we refine the initial guess by letting the segments flow towards valleys in the height field (orange in Figure 7).

Endpoints of the path segments. Since the height field is smooth in general (except from discontinuities that are caused by occlusions), a path network that minimizes (3) is likely to pass through the field's local minima. Therefore, in the case of a height field that contains exactly two local minima, we fix the endpoints to those. Similarly, if only one local minimum is present in the 2D height field, we choose the global minimum of its opposite longitude as the second endpoint.

If more than two minima exist, we pick the two minima with the least height but favor pairs with a large distance with a soft constraint in order to not degenerate the solution. More specifically, we find the pair of minima $m_1, m_2 \in \mathbb{R}^3$ (unit vectors representing the direction) that solve:

$$\underset{m_1,m_2\in M}{\arg\min}\ \lambda\left(1-\frac{\arccos(\langle m_1,m_2\rangle)}{\pi}\right)+\frac{height(m_1)+height(m_2)}{2\cdot\max_{m\in M}(height(m))}, \tag{14}$$

where M is the set of all local minima and $\lambda \in [0,\infty)$ is a trade-off parameter that balances between the proximity criterion (first term) and the height criterion (second term). In our implementation, we achieved good results with $\lambda = 2$. The denominators of the two terms are used to normalize the according values. Due to the small size of the optimization domain, the optimal pair can be found quickly with an exhaustive search.

Ridge Line.

Ridge Line. Due to the overall smoothness of the height field, the ridge line usually lies on or near the bisecting plane between the two endpoints. Therefore, we initialize the ridge line (represented as a point sampling) with the circle that intersects the sphere and the bisector. All samples are then moved iteratively towards the closest ridge or peak by gradient ascent, such that the line converges to the actual ridge line. During this procedure, the line is re-sampled where necessary.

Number of Segments and Initial Guess. To initialize the path segments, we find representative points on the ridge line. We assume that the average height of a path segment correlates closely with the height of the corresponding representative (as this is usually the highest point of that path segment). Therefore, one objective of representative selection is to find representatives with least average height. Furthermore, the representatives have to be chosen in a way that the resulting path network fulfills the solid angle constraints (5). To simplify computation, we project these constraints on the bisecting plane, which results in the new constraints:

$$\begin{aligned}&O := \text{center of sphere}\\ &\underline{\theta}_\phi \le \measuredangle(proj(R_i), O, proj(R_{i+1 \bmod |R|})) \le \overline{\theta}_\phi \quad \forall \quad 0 \le i < |R|,\end{aligned} \tag{15}$$

where $proj(p)$ is the perpendicular projection of a point onto the bisector plane. In this formulation, the bounds previously expressed as solid angles have become simple angle bounds $\underline{\theta}_\phi$ and $\overline{\theta}_\phi$. Empirically, we found that $\underline{\theta}_\phi = \frac{1}{3}\pi$ and $\overline{\theta}_\phi = \frac{4}{3}\pi$ result in good partitionings of the unit sphere. The representatives are then calculated as the minimizer of the resulting average height:

$$\begin{aligned}&\underset{R}{\arg\min}\frac{1}{|R|}\sum_{r\in R} height(r)\\ &s.t. (15)\end{aligned} \tag{16}$$

This can be computed efficiently with a dynamic program (for details, refer to Appendix B). To speed up computation even further, we run this program only on local minima of the ridge line. If this does not produce a valid output, we re-run it on the entire ridge line.

Path Refinement After the sequence of representatives have been found and the path network has been initialized with the according circular arcs, all path segments are then refined through gradient descent, which lets the initial line flow towards the closest valley. The result of this procedure is a path network that encompasses the entire sphere (due to the angle bounds) and that has approximately minimum average height. Finally, (3) is evaluated by calculating the average height over the path and taking the e-th root.

4.2. Calculation of Derived Measures

The second derivative of the perceptible size is calculated on the skeleton edges using central differences of the first derivatives of incident edges. However, since the curve skeleton is non-manifold, we average the derivative values over all incident edges of the source and target vertex of the evaluated edge, respectively. In the case that an incident edge is oriented in the opposite direction as the evaluated edge (i.e. both edges have the same target or source vertex), we use its negated first derivative, which results in the derivative in the direction of the evaluated edge. The resulting second derivative is direction independent, i.e. the derivative in the reverse direction of an edge is the same as in the forward direction.

Smoothing is done similarly. We employ a modified version of Dijkstra's algorithm to find the relevant neighbors for the processed vertex or edge. The smoothing weights are then calculated based on the geodesic distance along the curve skeleton. If a direction dependent measure such as the first derivative is smoothed, we negate the measure's sign where necessary.

4.3. Utilized Libraries

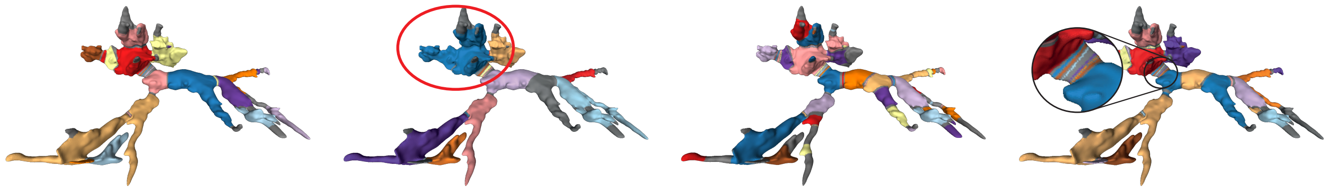
For ray-shooting in the calculation of the perceptible size, we use data structures from CGAL [ATW16]. For minimization of the Markov Random Field, we rely on the QPBO implementation provided by OpenGM [ABK12].

5. Results

In the following section, we present some results of our algorithm and discuss limitations. A thorough evaluation with respect to the common sense in speleology is given in Section 6.

The algorithm presented so far has six parameters: The exponent e defines what measure is used to derive the perceptible size. The size of the smoothing kernel used to derive the cave scale is represented as a multiple of the local cave size $\sigma_{scale}(v) = \mu_{scale} \cdot size(v)$. The kernel widths for the cave size σ_{size} and the first derivative $\sigma_{size'}$ are then expressed as multiples of the cave scale $\sigma_{size}(v) = \mu_{size} \cdot scale(v)$, $\sigma_{size'}(v) = \mu_{size'} \cdot scale(v)$. Finally, the energy definition requires the curvature tipping point θ_{tip} and the direction tolerance θ_{dir}.

Figure 9 shows the influence of select parameters. Increasing the



(a) *Base parameter set, $e = 1.5$, $\mu_{scale} = 10$, $\mu_{size} = 0.2$, $\mu_{size'} = 0.2$, $\theta_{tip} = 0.3$, $\theta_{dir} = 0.2$*

(b) *Increased smoothing leads to larger segments, $\mu_{size} = 0.4$*

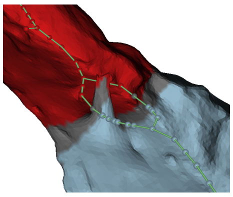
(c) *Smaller tipping curvature leads to more segments, $\theta_{tip} = 0.1$*

(d) *Higher direction tolerance leads to segment oscillations, $\theta_{dir} = 0.4$*

Figure 9: *Impact of select parameters on the final result*

exponent e results in steeper profiles of the perceptible size, which increases its derivatives. As a consequence, higher exponents result in a larger number of segments. Small values of μ_{scale} cause a more localized evaluation of the scale, whereas large values result in less global variation. In general, higher values lead to a larger number of segments because the scale increases, especially in small-sized areas. Increasing either of the smoothing kernels μ_{size} or $\mu_{size'}$ results in a smoother profile of the perceptible size and thus in fewer recognized entrances. The curvature tipping point θ_{tip} relates directly to the number of recognized entrances. Higher values produce fewer segments. Finally, the direction tolerance θ_{dir} affects only areas with a near-vanishing first derivative of the perceptible size. Higher values lead to more short segments that are only one or two skeleton vertices long. This happens because the weak guidance of the first derivative cannot be used to decide for an entrance direction if the entrance probability is higher than 0.5 and multiple entrances in alternating directions are created.

A limitation of our algorithm is posed by big rocks or stalagnates inside the cave that reduce the perceptible size significantly (see nearby figure). In this case, the skeleton is split by the stalagnate, which results in a sudden drop in perceptible size. Therefore, entrances are created at this location. A possible solution for this limitation is a preprocessing step that recognizes and eliminates such problems in the mesh, which we will explore in our future work.



6. Evaluation

In the following section, we explain our approach of evaluating the presented algorithm against real-world requirements from experts. We express the quality of our segmentation with a plausibility measure $\in [0,1]$ that compares the results of our algorithm with expert feedback. This measure represents the percentage of the cave surface that our algorithm segmented in conformance with the experts. The plausibility value is then evaluated on a sampling of the parameter space in order to derive the best set of parameters. In the following, we give a brief overview of the evaluation method and results. The details on the plausibility measure follow in Section 6.2.

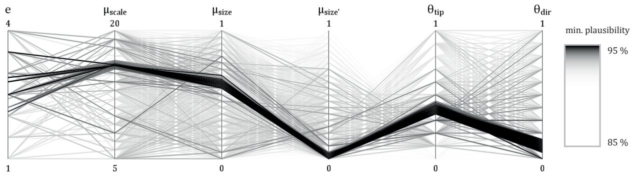


Figure 10: *Parallel Coordinates plot of the parameter set samples that result in the highest minimal plausibility ($\geq 85\%$)*

6.1. Overview

We use the two cave data sets that have already been presented in the paper. Both data sets were acquired using highly detailed terrestrial laser scanners using marker-based registration. The resulting point clouds have been used as input for Poisson surface reconstruction to get a manifold watertight mesh of desired resolution.

In a first step, we gathered expert opinion with the help of an interactive tool. The experts were presented a 3D view of the cave and could mark areas as either passages or chambers by painting on the surface. We instructed them to mark only areas where they were certain of their decision and leave doubting areas blank. To minimize the time required for the user study, we did not require them to paint everything but only to place a few strokes. A full painting would not give much more information because it is very likely that our algorithm matches the entire part that a stroke represents if it matches the stroke itself. We have gathered a total of six feedback data sets. Figure 11a shows one example.

Gathering this feedback allows to calculate the plausibility measure for each cave data set, given a set of segmentation parameters. We used the minimum plausibility over both cave data sets as the overall plausibility of a parameter set, which ultimately measures the quality of a parameter set over all available test data sets. By sampling the parameter space uniformly, we find parameter sets that result in high plausibilities, which are visualized in Figure 10. The diagram shows a clear band of high-quality parameters, which we see as an indicator that our algorithm is stable to small changes of the parameters and that our approach offers enough degrees of freedom to generalize to other data sets. Since a correlation of parameters is not directly apparent from this figure, we believe that all parameters are necessary and do not bear any redundancy.

This sampling also allows to find the best parameter set, which is visualized in Figure 11b and achieves an overall plausibility of

95.5% (i.e. our segmentation conforms to expert feedback on at least 95.5% of the cave surface over both data sets). The majority of chambers match the feedback from Figure 11a closely. Most areas that do not match exhibit ambiguous expert feedback (e.g. some experts classified the middle light-blue branch as a chamber).

6.2. Plausibility Measure

We start by explaining the characteristics of the plausibility measure in the case of a single feedback data set and then expand this to multiple feedback data sets. For every vertex, we can compare the segmentation result and the expert opinion. A vertex for which both match has a plausible segmentation. Additionally, vertices that have been left blank by the expert are also considered plausible because there is no evidence that suggests otherwise. The overall plausibility of a segmentation can then be calculated as the ratio of the number of plausible vertices and total number of vertices. The resulting plausibility value is a number in the range $[0,1]$ and is bounded from below by the percentage of blank vertices in the feedback. This can be formulated equivalently as the complement of unplausibility, where unplausibility is the percentage of unplausible vertices. We will use this formulation for our generalization to more feedback data sets.

If more than one feedback exists, the following constellations for a specific vertex may occur: 1. All feedbacks agree on a segmentation or left the vertex blank. In this case, the vertex can be uniquely classified and plausbility can be calculated as above. 2. Feedbacks do not agree. In that case, a unique classification of the vertex is not possible, even if a majority of feedbacks vote for a certain segmentation. To capture these situations, we use a soft classification of the vertices, denoted by the chamber probability. We calculate the chamber probability as the average from all feedbacks (i.e. 0 for a *passage* segmentation and 1 for a *chamber* segmentation, cf. Figure 12), excluding blank segmentations. For unique classifications, this probability assumes the value of either 0 or 1 whereas contradicting feedback leads to an intermediate value. To evaluate a vertex' unplausibility, we first check if the algorithm result matches the majority vote (e.g. if the algorithm decided for a chamber and the chamber probability is larger than 0.5). In these cases, we define the unplausibility to be zero because the algorithm reached the same decision as the majority of experts. If that is not the case, we use $2|p-0.5|$ as the unplausibility of the vertex, where p is the chamber probability. This results in an unplausibility of 0 for undecided vertices (i.e. chamber probability of 0.5) and of 1 for uniquely classifiable vertices (i.e. chamber probability of 0 or 1). The overall segmentation plausibility is then the complement of the average unplausibility. As before, this measure is bounded from below by the percentage of vertices that are blank in all feedbacks.

7. Conclusions and Future Work

In this paper, we presented an algorithm for fully-automatic cave chamber recognition that we derived strictly from real-world requirements. The key of this segmentation approach is a curve skeleton, which guarantees segmentation consistency along the cross section of the cave and reduces the computation time. As shown in our evaluation, the results of this algorithm match very closely to the opinions of experts.

However, some scenarios cannot be captured by our algorithm, such as stalagnates or big rocks in the cave. Therefore, we will continue to explore further improvements such as cleaning preprocessing steps and additional terms in the energy function.

Due to limited availability of cave data sets, we were only able to perform our quantitative evaluation on two data sets. As more data sets become available, we would like to perform the evaluation on a significantly larger data base, which could actually result in a parameter set that generalizes well to unseen caves. Furthermore, a larger number of experts and higher-quality feedback may improve evaluation results.

Finally, a refinement of the skeleton-based segmentation similar to [KT09] may be desirable, especially if the segmentation is to be transferred onto the high-resolution mesh.

8. Acknowledgments

We would like to thank our speleology experts, most notably Donald McFarlane and Guy van Rentergem, who gave valuable information on the characteristics of caves, actively participated in discussions, and provided feedback for our evaluation.

This work has been partially funded by the German Ministry of Education and Research (project number 03ZZ0516A).

References

[ABK12] ANDRES B., BEIER T., KAPPES J. H.: OpenGM: A C++ library for discrete graphical models. *ArXiv e-prints* (2012). arXiv:1206.0111. 8

[aE01] AM ENDE B. A.: 3d mapping of underwater caves. *Computer Graphics and Applications 21*, 2 (2001), 14–20. 1

[AKM*06] ATTENE M., KATZ S., MORTARA M., PATANÉ G., SPAGNUOLO M., TAL A.: Mesh segmentation-a comparative study. In *SMI* (2006), IEEE. 2

[APPS10] AGATHOS A., PRATIKAKIS I., PERANTONIS S., SAPIDIS N. S.: Protrusion-oriented 3d mesh segmentation. *The Visual Computer 26*, 1 (2010), 63–81. 3

[ATW16] ALLIEZ P., TAYEB S., WORMSER C.: 3D fast intersection and distance computation. In *CGAL User and Reference Manual*, 4.9 ed. CGAL Editorial Board, 2016. 8

[AZC*12] AU O. K.-C., ZHENG Y., CHEN M., XU P., TAI C.-L.: Mesh segmentation with concavity-aware fields. *TVCG 18*, 7 (2012). 2

[Bec80] BECK B. F.: *An introduction to caves and cave exploring in Georgia*. American Geosciences Institute, 1980. 3

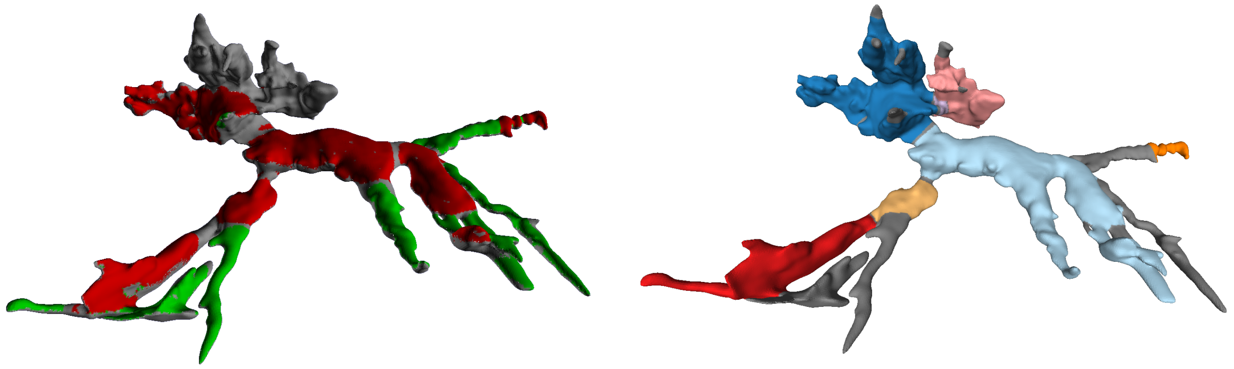
[BG09] BUCHROITHNER M. F., GAISECKER T.: Terrestrial laser scanning for the visualization of a complex dome in an extreme alpine cave system. *Photogrammetrie-Fernerkundung-Geoinformation 2009*, 4 (2009), 329–339. 1

[BLVD11] BENHABILES H., LAVOUÉ G., VANDEBORRE J.-P., DAOUDI M.: Learning boundary edges for 3d-mesh segmentation. *CGF 30*, 8 (2011), 2170–2182. 3

[CSM07] CORNEA N. D., SILVER D., MIN P.: Curve-skeleton properties, applications, and algorithms. *TVCG 13*, 3 (2007), 0530–548. 1

[DGGV08] DE GOES F., GOLDENSTEIN S., VELHO L.: A hierarchical segmentation of articulated bodies. In *CGF* (2008), vol. 27, Wiley Online Library, pp. 1349–1356. 2

[DS06] DEY T. K., SUN J.: Defining and computing curve-skeletons with medial geodesic function. In *Symposium on geometry processing* (2006), vol. 6, pp. 143–152. 3



(a) *Sample feedback. Red areas represent chambers, green areas represent passages, unlabeled areas are grey.*

(b) *Segmentation.* $e = 2.9, \mu_{scale} = 15.9, \mu_{size} = 0.57, \mu_{size'} = 0.0, \theta_{tip} = 0.38, \theta_{dir} = 0.06$

Figure 11: *Sample expert feedback and segmentation of the Simud Puteh data set with the globally best parameter set.*

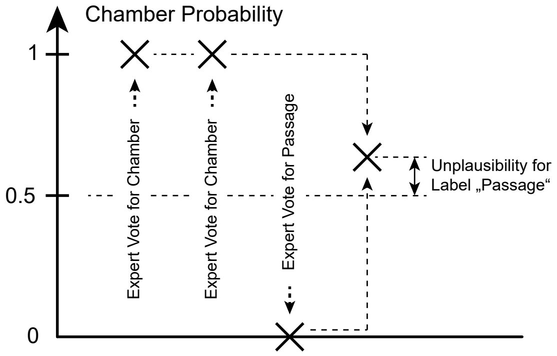


Figure 12: *Calculation of the unplausibility of a vertex. The three expert votes result in a chamber probability of* $\frac{2}{3}$. *Thus, the resulting unplausibility for the label* $\mathcal{C}$ *is* 0 *and* $\frac{1}{3}$ *for* $\mathcal{P}$

[Fie99] FIELD M. S.: *A lexicon of cave and karst terminology with special reference to environmental karst hydrology*. National Center for Environmental Assessment, US EPA, 1999. 1

[FJT15] FENG C., JALBA A. C., TELEA A. C.: Part-based segmentation by skeleton cut space analysis. In *ISMM* (2015), Springer. 3

[Fla71] FLANNERY J. J.: The relative effectiveness of some common graduated point symbols in the presentation of quantitative data. *Cartographica 8*, 2 (1971), 96–109. 4

[FSKR11] FANG Y., SUN M., KIM M., RAMANI K.: Heat-mapping: A robust approach toward perceptually consistent mesh segmentation. In *CVPR* (2011), IEEE, pp. 2145–2152. 2

[GAMNGL*09] GONZÁLEZ-AGUILERA D., MUÑOZ-NIETO A., GÓMEZ-LAHOZ J., HERRERO-PASCUAL J., GUTIERREZ-ALONSO G.: 3d digital surveying and modelling of cave geometry: Application to paleolithic rock art. *Sensors 9*, 2 (2009), 1108–1127. 1

[GDB06] GOSWAMI S., DEY T. K., BAJAJ C. L.: Identifying flat and tubular regions of a shape by unstable manifolds. In *SPM* (2006), ACM, pp. 27–37. 3

[GF09] GOLOVINSKIY A., FUNKHOUSER T.: Consistent segmentation of 3d models. *Computers & Graphics 33*, 3 (2009), 262–269. 2

[HR84] HOFFMAN D., RICHARDS W.: Parts of recognition. *Cognition 18*, 1-3 (1984), 65 – 96. 2

[KBH06] KAZHDAN M., BOLITHO M., HOPPE H.: Poisson surface reconstruction. In *SGP* (2006), vol. 7. 1

[KHS10] KALOGERAKIS E., HERTZMANN A., SINGH K.: Learning 3d mesh segmentation and labeling. *ACM TOG 29*, 4 (2010), 102. 3

[KT09] KAPLANSKY L., TAL A.: Mesh segmentation refinement. In *CGF* (2009), vol. 28, Wiley Online Library, pp. 1995–2003. 3, 10

[KZ04] KOLMOGOROV V., ZABIN R.: What energy functions can be minimized via graph cuts? *IEEE Transactions on Pattern Analysis and Machine Intelligence 26*, 2 (Feb 2004), 147–159. 12

[LCHB12] LV J., CHEN X., HUANG J., BAO H.: Semi-supervised mesh segmentation and labeling. *CGF 31*, 7 (2012), 2241–2248. 3

[LHMR08] LAI Y.-K., HU S.-M., MARTIN R. R., ROSIN P. L.: Fast mesh segmentation using random walks. In *SPM* (2008), ACM. 2

[LRB07] LEMPITSKY V., ROTHER C., BLAKE A.: Logcut - efficient graph cut optimization for markov random fields. In *ICCV* (Oct 2007), pp. 1–8. 12

[MPS*04a] MORTARA M., PATANÉ G., SPAGNUOLO M., FALCIDIENO B., ROSSIGNAC J.: Blowing bubbles for multi-scale analysis and decomposition of triangle meshes. *Algorithmica 38*, 1 (2004). 3

[MPS*04b] MORTARA M., PATANÉ G., SPAGNUOLO M., FALCIDIENO B., ROSSIGNAC J.: Plumber: a method for a multi-scale decomposition of 3d shapes into tubular primitives and bodies. In *SM* (2004). 3

[MRB*15] MCFARLANE D. A., ROBERTS W., BUCHROITHNER M., VAN RENTERGEM G., LUNDBERG J., HAUTZ S.: Terrestrial lidar-based automated counting of swiftlet nests in the caves of gomantong, sabah, borneo. *International Journal of Speleology 44*, 2 (2015), 191. 1

[MST10] MOHAN V., SUNDARAMOORTHI G., TANNENBAUM A.: Tubular surface segmentation for extracting anatomical structures from medical imagery. *IEEE T-MI 29*, 12 (2010). 3

[Pal07] PALMER A. N.: *Cave geology*. Cave books Dayton, 2007. 3

[RKLS07] ROTHER C., KOLMOGOROV V., LEMPITSKY V., SZUMMER M.: Optimizing binary mrfs via extended roof duality. In *CVPR* (June 2007), pp. 1–8. 6

[SBCBG11] SOLOMON J., BEN-CHEN M., BUTSCHER A., GUIBAS L.: Discovery of intrinsic primitives on triangle meshes. In *Computer Graphics Forum* (2011), vol. 30, Wiley Online Library, pp. 365–374. 3

[Sha08] SHAMIR A.: A survey on mesh segmentation techniques. In *CGF* (2008), vol. 27, Wiley Online Library, pp. 1539–1556. 2

[SRFVP13] SILVESTRE I., RODRIGUES J. I., FIGUEIREDO M., VEIGA-PIRES C.: Cave chamber data modeling and 3d web visualization. In *8th International Conference on Geometric Modeling & Imaging* (2013). 3

[SSCO08] Shapira L., Shamir A., Cohen-Or D.: Consistent mesh partitioning and skeletonisation using the shape diameter function. *The Visual Computer 24*, 4 (2008), 249–259. 2

[TAOZ12] Tagliasacchi A., Alhashim I., Olson M., Zhang H.: Mean curvature skeletons. *CGF 31*, 5 (2012), 1735–1744. 4

[VMM99] Vollmer J., Mencl R., Müller H.: Improved laplacian smoothing of noisy surface meshes. *CGF 18*, 3 (1999), 131–138. 4

[WLAT14] Wang H., Lu T., Au O. K.-C., Tai C.-L.: Spectral 3d mesh segmentation with a novel single segmentation field. *Graphical Models 76*, 5 (2014), 440–456. 2

[WY11] Wang J., Yu Z.: Surface feature based mesh segmentation. *Computers & Graphics 35*, 3 (2011), 661–667. 3

[YWLY12] Yan D.-M., Wang W., Liu Y., Yang Z.: Variational mesh segmentation via quadric surface fitting. *Computer-Aided Design 44*, 11 (2012), 1072–1082. 3

[ZLXH08] Zhang X., Li G., Xiong Y., He F.: 3d mesh segmentation using mean-shifted curvature. In *International Conference on Geometric Modeling and Processing* (2008), Springer, pp. 465–474. 2

A. Proof of NP-hardness of the segmentation problem

To evaluate the computational complexity of minimizing a Markov Random Field, we analyze the energy's submodularity, which is defined as follows [KZ04, LRB07]:

$$\begin{aligned} &\textit{submodular} \iff \\ &\phi_{i,j}(\mathcal{C},\mathcal{C}) + \phi_{i,j}(\mathcal{P},\mathcal{P}) \le \phi_{i,j}(\mathcal{C},\mathcal{P}) + \phi_{i,j}(\mathcal{P},\mathcal{C}) \\ &\qquad \forall (i,j) \in E \end{aligned} \tag{17}$$

In the following, we will show that in general the energy defined in (7) is non-submodular, rendering the minimization problem NP-hard [KZ04]. However, since the parameter space for which an edge is non-submodular is very small and always in regions of high $\pi_{i,j;entrance}$, there are usually only a few (less than 1 %) non-submodular edges.

For reasons of brevity, we define

$$\begin{aligned} e &:= \pi_{entrance}(\widehat{size''(i,j)}) \\ d &:= \pi_{direction}(size'(i,j)) \end{aligned} \tag{18}$$

$$\begin{aligned} &\textit{submodular} \iff \\ &\phi_{i,j}(\mathcal{C},\mathcal{C}) + \phi_{i,j}(\mathcal{P},\mathcal{P}) \le \phi_{i,j}(\mathcal{C},\mathcal{P}) + \phi_{i,j}(\mathcal{P},\mathcal{C}) \\ &-2\log(1-e) \le -\log(e \cdot d) - \log(e \cdot (1-d)) \\ &\log\left((1-e)^2\right) \ge \log\left(e^2 \cdot d \cdot (1-d)\right) \\ &(1-e)^2 \ge e^2 \cdot d \cdot (1-d) \\ &1 - 2e + e^2(1-d+d^2) \ge 0 \\ &e \le \frac{1-\sqrt{d-d^2}}{1-d+d^2}. \end{aligned} \tag{19}$$

The resulting space of submodularity is visualized in Figure 13.

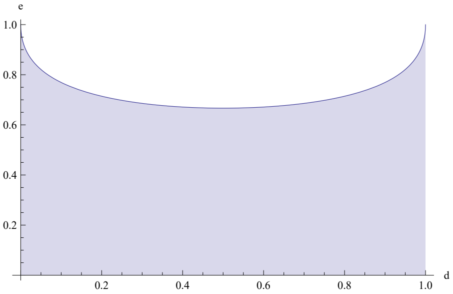


Figure 13: *Visualization of the parameter space of entrance probability e and direction probability d, for which the resulting potential is submodular (shaded).*

B. Dynamic Program for (16)

In the following, we present the dynamic program that solves

$$\begin{aligned} \arg\min_R \frac{1}{|R|} \sum_{r \in R} radius(r) \\ s.t. (15), \end{aligned} \tag{20}$$

where the output R is a subset of the input samples $S \ni (\alpha_i, h_i)$ consisting of the angular location on the circle and the according height value. The input samples are sorted by α_i.

We represent R by a list of indices into S. To find these indices, we propagate the function $\Sigma(f,l,k)$ over the relevant solution space in a DP manner, where f and $l > f$ correspond to the first and last entry in R, respectively, and k refers to the cardinality of R. The function value is a partial optimum, i.e. the maximum sum of height values of the set R that fulfills the constraints f,l,k. Once this function is known, the global optimum is

$$\min_{f \le \alpha_1 + \overline{\theta}_\phi, \alpha_f + 2\pi - \overline{\theta}_\phi \le l \le \alpha_f + 2\pi - \underline{\theta}_\phi, k} \left(\frac{1}{k} \Sigma(f,l,k) \right) \tag{21}$$

and the inducing set R can be found by backtracking.

We initialize the dynamic program with $\Sigma(f,f,1) = h_f$ for all relevant f and $\Sigma(\cdot) = \infty$ for all remaining entries. To propagate the function over the solution space, we start at a known entry (f,l,k) and explore all valid successors updating the according entries:

$$\begin{aligned} &\forall (f,m,k+1), s.t. \alpha_l + \underline{\theta}_\phi \le \alpha_m \le \alpha_l + \overline{\theta}_\phi : \\ &\quad \Sigma(f,m,k+1) \leftarrow \min\left(\Sigma(f,m,k+1), \Sigma(f,l,k) + h_m\right) \end{aligned} \tag{22}$$

At the same time, we store the predecessor with the new entry. If an entry falls in the acceptable range defined by (21), we immediately update the global optimum.

After this propagation, the minimum defined by (21) is directly available and we find the minimizer by following the predecessor links set up during propagation. Under the assumption that the samples are distributed approximately uniformly, the time complexity of this algorithm is $\mathcal{O}(\overline{\theta}_\phi \cdot n^2 \cdot (\overline{\theta}_\phi - \underline{\theta}_\phi))$, where the first $\overline{\theta}_\phi$ stands for the number of samples that need to be considered for f, n is the total number of input samples, and the term $\overline{\theta}_\phi - \underline{\theta}_\phi$ represents the number of explored samples during propagation.