Topological Effects of Grid Size and Volume Fraction Threshold in Sculpt
Research Note

Brian Shawcroft* Kendrick Shepherd† Scott A. Mitchell‡

Abstract

Subdividing Sculpt's background grid often improves the topological and geometric fidelity of the output mesh to the input geometry. However, in some simple 2D examples, topology does not converge as the background grid is refined. The inclusion of subcells in the output mesh does not behave monotonically under refinement. However, for a fixed grid size, the volume fraction threshold for inclusion is a valid persistence parameter for defining a monotonic filtration. Thus we can use persistent homology to efficiently predict or control mesh topology for the volume fraction parameter (unlike grid size). We demonstrate this on geographical data through a two-dimensional version of Sculpt.

1 Introduction.

Generating a mesh suitable for analysis often requires the analyst to modify the input geometry. Common issues are gaps and overlaps that prevent meshing; unneeded features smaller than the desired mesh size that increase element count; and small angles and thin regions that produce poor-quality elements.

However, the mesh is a discrete approximation to the geometry. Why require higher fidelity in the input than we aspire to in the output? Hence, the community is now creating tools to mesh ugly geometry directly.

Sculpt [10, 11, 12] is one such tool, achieving a hexahedral mesh of reasonable quality, but reconstructing an approximation of the input geometry and topology. Inexact reconstruction is a *benefit* in the case of gaps, overlaps, and small features. The Sculpt algorithm starts with a background grid overlaying the input geometry. The fraction of each grid cell that lies inside the geometry of an input material is its *volume fraction.* Cells with volume fractions above a threshold (e.g., one-half) are retained; the rest are discarded. Heuristics remove undesirable global topology such as pinch points and connected components consisting of only a few cells. Retained cells are then snapped to the geometry, and mesh quality is achieved through pillowing, smoothing, etc. Alternatively, TetWild [5] can be used to create tet-meshes of ugly geometry. Both Sculpt and TetWild produce meshes with valid topology, but there is no a priori knowledge of how the mesh's homology (i.e., the number of components, loops, and voids) will compare to the input, or to the analyst's desires.

In this work, we explore how to achieve a mesh of the desired element size and topology without changing the input. To do so, we leverage the notion of volume fraction used in Sculpt, but generalized to also operate on invalid geometries. We then use persistent homology [3, 9] to enable the analyst to measure and select the desired mesh topology.

For research and to demonstrate that this technique is effective, we develop a planar two-dimensional prototype in Rhinoceros 3D. It mimics the initial steps of Sculpt, using a background grid, volume fractions, and generalized winding numbers [1, 6] to define the "interior" of both valid and invalid geometries. Subsequently, persistent homology is used to explore the topological structure of potential meshes. Based on this data, topologically-appropriate quadrilateral meshes from both clean and messy datasets are generated. These meshes can then serve as input for subsequent steps in Sculpt to improve geometric fidelity, including snapping, pillowing, and smoothing.

2 Background Material

The topology of a mesh should contain the significant features of a domain for its intended computational analysis. For simplicity, we shall study mesh topology using a simplicial complex: nodes are zero-cells, edges are one-cells, and triangles are two-cells. Homology [4] is a mathematical tool that distinguishes simplicial complexes using certain algebraic groups, and the *Betti numbers* B_i count their ranks. Specifically, B_0 equals the number of connected components, B_1 is the number of holes, and B_2 is the number of cavities or voids. For planar domains B_2 will always be zero.

Persistent homology [3, 9] describes homology changes as objects are added and connections are made. A *filtration* has a "persistence parameter" which defines when a simplex enters the complex. A filtration is monotonic so no simplex may ever leave. However, the homol-

*Brigham Young Univiersity
†Brigham Young Univiersity
‡Sandia National Laboratories

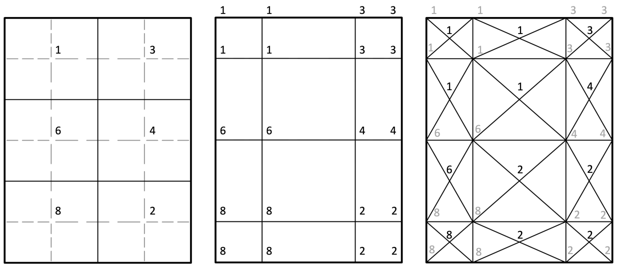


Figure 1: An example of converting cell volume fractions into a filtration of a simplicial complex. Left, the ordering of grid cells. Center, the dual grid with vertex values transferred from grid cells. Right, each 2-cell is subdivided into four triangles. The central vertex gets the smallest value of its neighbors.

ogy has both additions and removals, because adding a simplex could e.g., create a new connected component, or combine two components into one. The parameter value at which a new homological group is created is called its "birth," while the value it disappears is called its "death." Persistent homology not only counts Betti numbers, but tracks individual components and holes.

This work studies the persistent homology of a quadrilateral background grid using volume fractions as the persistence parameter. (In other work the signed distance to a domain boundary was the parameter [8].)

Volume fractions for watertight domains may be calculated by sampling points and counting the fraction of them inside the geometry. (For simplicity we will only discuss domains with a single material.) However, for poorly defined domains, what is "inside" may be poorly defined. Despite this, the generalized winding number [1, 6] gives reasonable and intuitive answers for most domains. The winding number is a continuous value, where 0 indicates outside and 1 is inside. The winding number gives the same answer as ray shooting for watertight domains with properly oriented boundaries. For geometry with gaps, the winding number near a gap is between 0 and 1. (In extreme cases, invalid geometries may give values beyond these bounds.)

3 Methodology

In this work, we restrict our discussion to two-dimensional domains. A background grid of cells is created, e.g., by subdividing an axis-aligned bounding box into an anisotropic regular grid. Each cell's volume fraction is computed as the average of the winding numbers of its sample points. We study how the topology changes as we lower the volume fraction threshold for including a cell in the output mesh.

The volume fraction is the desired persistence parameter, but we must first transform our grid of squares into a topologically-equivalent filtration of a simplicial complex. They key challenge is to ensure that before adding a simplex, all of its subsimplices have already been added. In fig. 1 we convert volume fractions into the order cells are included. Each ordinal corresponds to some volume fraction, and cell 1 has the highest volume fraction. We then form the dual of the background grid. Grid cells dualize into vertices with the same value. Then each 2-cell (dual to a grid node) is divided into four triangles with a central vertex. The central vertex's value is the minimal value of any adjacent node (dual cell). At dual-persistence value 1, all vertices with value 1 are added, then all edges between already-added vertices, then all triangles formed by already-added edges. At value 2, all vertices with value 2 are added, then all edges between already-added vertices, etc.

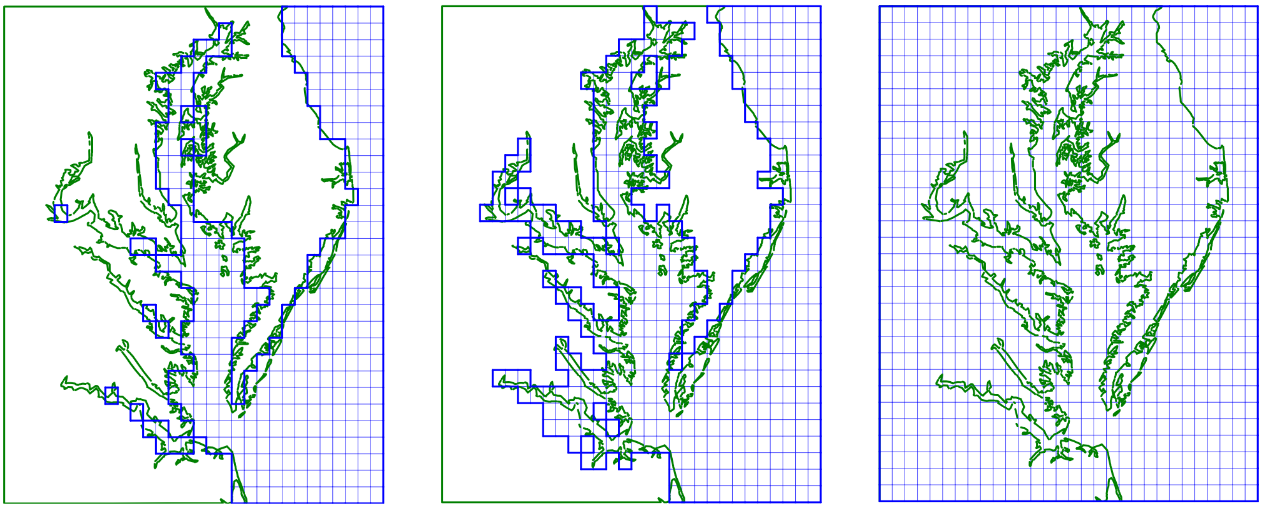
4 Results

4.1 Computational Results The proposed framework was developed and evaluated in 2D using a custom plugin to Rhinoceros 3D and Grasshopper. Winding numbers were computed using libigl [7]. Persistent homology was computed using Aleph [13], which is based on PHAT [2].

The framework is tested on an oriented planar representation of the Chesapeake Bay.[1] Snapshots of computed volume fractions, with their associated Betti numbers, are shown in fig. 2. Model errors include overlapping edges, repeated/offset edges, and numerous gaps. Despite the "interior" of the bay being ill-defined, the proposed method still captures the intended geographic domain with respect to both the continent and to islands. A complete view of the homological structure based on varying the volume fraction is shown in the persistence diagram of fig. 3. Results demonstrate that a mesh with the desired homological structure could be extracted from the background grid by selecting the right threshold. Finally, we note that these figures are primarily for illustrative purposes—in practice, a finer grid may be needed to better capture local behavior.

4.2 Topological Effects of Grid Refinement We conjectured that persistent homology can measure the necessary grid size to achieve a desired topology. When features are isolated or globally the same scale, grid refinement has intuitive and predictable topological effects. However, we discovered that this is *not* true for general inputs. Counterexamples show non-monotonic filtration behavior by grid size. Discretization by grid

[1]Model derived from `https://vecta.io/symbols/281/ecosystems-maps/93/usa-md-va-chesapeake-bay-line-map`



(a) Vol. Frac.: 50%; $B_0 = 4; B_1 = 2$ (b) Vol. Frac.: 15%; $B_0 = 1; B_1 = 7$ (c) Vol. Frac.: None; $B_0 = 1; B_1 = 0$

Figure 2: Chesapeake Bay meshes and their Betti numbers change as the volume fraction threshold is lowered. Left, cells with volume fraction ≥ 0.5 in the bay. Center, threshold ≥ 0.15. Right, all cells. The model contains deliberate errors: gaps, overlaps, and offsets.

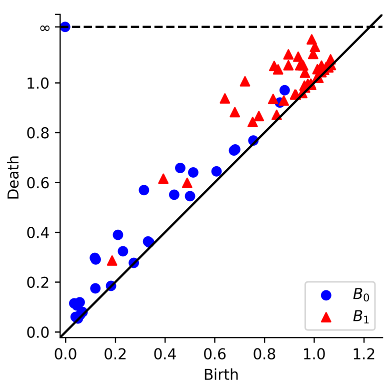


Figure 3: The persistence diagram of the Chesapeake Bay grid, with parameter 1 minus the volume fraction.

cells and their alignment with input features strongly effects topological behavior. Thus Sculpt algorithm parameters of when to refine the background grid may have unpredictable effects on mesh topology.

Convergence For some inputs, as the grid is refined, topological features of the input are resolved and the output mesh topology becomes stable. For these inputs there may be some way to define a filtration, with simplices only appearing, never disappearing. However, for some other inputs, the topology never converges and no filtration is possible.

A parallel axis-aligned gap is closed or open depending on its size and position relative to the grid; see fig. 4. A gap smaller than half the grid size is always closed. A gap larger than the grid size is always open. Between these, shifting the grid to the left will cause the mesh to alternate between closed and open.

In figs. 5 and 6 a feature is inconsistently resolved due to aliasing effects of unaligned grids. For the constant-width gap in fig. 5, refining or coarsening the grid makes the gap resolved consistently as open or closed. However, for the variable-sized gap in fig. 6, global uniform refinement merely moves where the problem occurs. The example is a wedge of material bounded by two lines meeting at a small angle α at an apex. In locations where the grid size is about the same as the local width, whether a cell is included or excluded can change every few grid cells, leading to many separate connected grid components. For any small grid size, there will be some portion of the wedge where the lines are about that size apart, specifically in the range $[\frac{1}{2}, 1] \cot \alpha$ squares away from the apex. The grid topology may be constant over refinement, but is undesirable.

Non-convergence In the example in fig. 7 the output mesh topology does not converge under refinement. That is, there is no grid size below which the output mesh topology does not change. In fig. 7 the background grid is uniform, but we only draw some of the relevant cells at each level of refinement. Blue (closure) cells are mostly material and thus included in the output mesh. Red (gap) cells are unfilled and excluded. Under refinement, the mesh alternates between

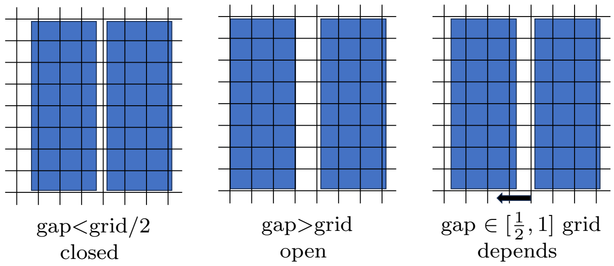
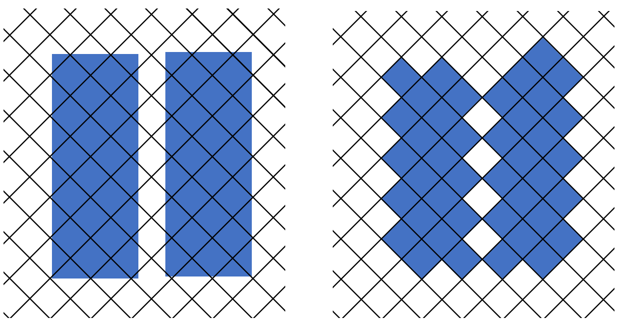


Figure 4: Small grid-aligned gaps are closed, large are open, and for intermediate it depends on their offset.



(a) raw geometry (b) filled cells

Figure 5: This unaligned gap is resolved inconsistently.

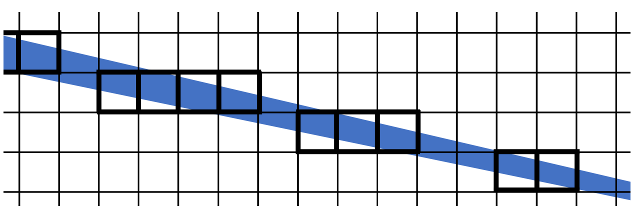


Figure 6: Aliasing may cause several connected components near where two lines meet at a sharp angle. Bold-outlined cells are filled, thin are open.

one and two connected components ad infinitum. The grid squares containing the corner alternate between filled and open, because of the corner's relative position inside its square. The description of the geometry is finite, just two triangular blocks meeting at a point. The geometry is simple and without sharp angles or unusual features, and plausible to occur in practice.

There may be two of these features, but with alternate sizes of when they are open and closed. Hence the two sides are always connected by exactly one of the features, giving the homology of a disk. The homology does not change under refinement, but the local connectivity does. Hence, even if we were to use persistent homology to measure and predict the

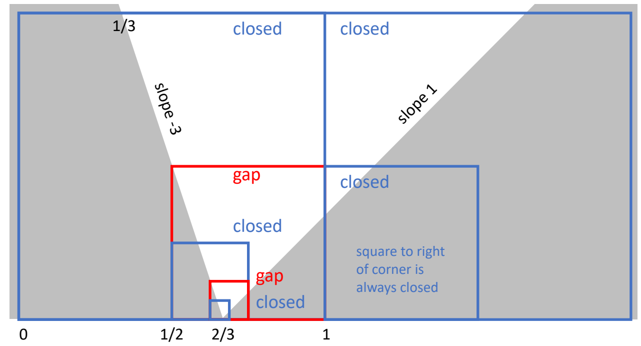


Figure 7: In this counterexample to convergence, the grid topology alternates between one and two connected components ad infinitum under refinement.

topology, it would not distinguish between this case and a smooth solid block of material.

The analytic description of the geometry in fig. 7 is two blocks of material with slopes -3 and 1 meeting at a corner with coordinate $(\frac{2}{3}, 0)$. If we start with a unit grid, then under refinement the grid square containing the corner alternates between having the corner 2/3 of the way along the bottom edge (blue), and 1/3 of the way (red). A blue square containing the gap is exactly half filled with material from the left triangle, and more than half filled when including the material from the right triangle. This construction is not tight: the slopes may be different. The corner may lie at some x-coordinate other than 1/3, but will never be at a grid corner as long as its x-coordinate is not $k/2^m$ for some $\{k, m\} \in \mathbb{Z}$. Thus more complicated sequences may be constructed.

5 Conclusion

Sculpt, and perhaps other volume-fraction based algorithms, behave quite differently than boundary-fitted algorithms, such as Delaunay Refinement, when the local mesh size is decreased. For boundary-fitted algorithms, reducing the mesh size to the local feature size or less allows the mesh to have good quality and recover the input topology exactly. For Sculpt, the mesh quality is good regardless of the local feature size, but in some cases the topology does not converge as the mesh size decreases by subdividing the background grid. However, for a fixed grid, we may predict the Sculpt topology, measure how that changes as we vary the volume-fraction threshold using persistent homology, and select an appropriate topology for the mesh's intended purpose. Future work will focus on mesh optimization to better match input geometry and on deploying these techniques for volumetric data.

Acknowledgements

We thank Steve Owen for providing Sculpt and explaining its subtleties to us. We thank Daniele Panozzo for answering our questions about libigl. We thank Alec Jacobson and the other authors of libigl, especially the winding number component, for making it freely available to the community. We thank Darren Engwirda for discussing coastal climate modeling with us.

This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research (ASCR), Applied Mathematics Program. Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

References

[1] G. Barill, N. G. Dickson, R. Schmidt, D. I. Levin, and A. Jacobson, *Fast winding numbers for soups and clouds*, ACM Transactions on Graphics (TOG), 37 (2018), pp. 1–12.

[2] U. Bauer, M. Kerber, J. Reininghaus, and H. Wagner, *PHAT – persistent homology algorithms toolbox*, Journal of Symbolic Computation, 78 (2017), pp. 76–90. Algorithms and Software for Computational Topology.

[3] H. Edelsbrunner, D. Letscher, and A. Zomorodian, *Topological persistence and simplification*, Discrete Computational Geometry, 28 (2002), pp. 511–533.

[4] A. Hatcher, *Algebraic Topology*, Cambridge University Press, 2001.

[5] Y. Hu, Q. Zhou, X. Gao, A. Jacobson, D. Zorin, and D. Panozzo, *Tetrahedral meshing in the wild*, ACM Trans. Graph., 37 (2018).

[6] A. Jacobson, L. Kavan, and O. Sorkine-Hornung, *Robust inside-outside segmentation using generalized winding numbers*, ACM Trans. Graph., 32 (2013), p. 33.

[7] A. Jacobson, D. Panozzo, et al., *libigl: A simple C++ geometry processing library*, 2018. https://libigl.github.io/.

[8] C. Moon, S. A. Mitchell, J. E. Heath, and M. Andrew, *Statistical inference over persistent homology predicts fluid flow in porous media*, Water Resources Research, 55 (2019).

[9] N. Otter, M. A. Porter, U. Tillmann, P. Grindrod, and H. A. Harrington, *A roadmap for the computation of persistent homology*, EPJ Data Science, 6 (2017), p. 17.

[10] S. J. Owen, *Parallel smoothing for grid-based methods*, International Meshing Roundtable, Research Note, (2012), pp. 161–178.

[11] S. J. Owen and T. R. Shelton, *Evaluation of grid-based hex meshes for solid mechanics*, Engineering with Computers, 31 (2015), pp. 529–543.

[12] S. J. Owen, M. L. Staten, and M. C. Sorensen, *Parallel hex meshing from volume fractions*, in International Meshing Roundtable, W. R. Quadros, ed., Berlin, Heidelberg, 2012, Springer Berlin Heidelberg, pp. 161–178.

[13] B. Rieck et al., *Aleph — a library for exploring persistent homology*, 2016. https://github.com/Pseudomanifold/Aleph.