

VoroCrust Illustrated: Theory and Challenges

Ahmed Abdelkader¹, Chandrajit L. Bajaj², Mohamed S. Ebeida³,
Ahmed H. Mahmoud⁴, Scott A. Mitchell³, John D. Owens⁴, and
Ahmad A. Rushdi^{3,4}

1 Dept. of Computer Science, University of Maryland, College Park

2 Dept. of Computer Science, University of Texas, Austin

3 Center for Computing Research, Sandia National Laboratories

4 Dept. of Electrical and Computer Engineering, University of California, Davis

Abstract

Over the past decade, polyhedral meshing has been gaining popularity as a better alternative to tetrahedral meshing in certain applications. Within the class of polyhedral elements, Voronoi cells are particularly attractive thanks to their special geometric structure. What has been missing so far is a Voronoi mesher that is sufficiently robust to run automatically on complex models. In this video, we illustrate the main ideas behind the VoroCrust algorithm, highlighting both the theoretical guarantees and the practical challenges involved in its implementation.

1998 ACM Subject Classification I.3.5 Computational Geometry and Object Modeling

Keywords and phrases sampling conditions, surface reconstruction, polyhedral meshing, Voronoi

Digital Object Identifier 10.4230/LIPIcs.xxx.yyy.p

1 Introduction

Generating quality meshes is an important problem in computer graphics and geometric modeling. There has been a growing interest in polyhedral meshing as it promises significant advantages over tetrahedral or hex-dominant meshing in some contexts. For example, polyhedral meshing offers higher degrees of freedom per element and is more efficient in filling a space, because it produces fewer elements for the same number of vertices. Within the class of polyhedral mesh elements, Voronoi cells enjoy several geometric properties, e.g., planar facets and positive Jacobians, which make them particularly suitable for numerical simulations.

A conforming mesh exhibits two desirable properties *simultaneously*: 1) a decomposition of the enclosed volume, and 2) a reconstruction of the bounding surface. A common technique for producing boundary-conforming decomposition from Voronoi cells relies on *clipping*, i.e., intersecting and truncating, each cell by the bounding surface [9]. An alternative to clipping is to locally mirror the Voronoi generators on either side of the surface [5].

The desired mesher takes as input a description of the geometric model for which a conforming Voronoi mesh is to be generated, e.g., a set of surface samples or a complete CAD model. Having access to a CAD model allows the mesher to accurately sample new points as needed; a crucial requirement for high quality meshing. Deferring the sampling problem, we start in an idealized setting that allows us to establish strong theoretical guarantees on the correctness of the underlying approach. Then, we highlight the challenges involved in applying this approach to realistic inputs as needed for a robust implementation.

© Ahmed Abdelkader, Chandrajit L. Bajaj, Mohamed S. Ebeida, Ahmed H. Mahmoud, Scott A. John D. Owens and Ahmad A. Rushdi;



licensed under Creative Commons License CC-BY

Conference title on which this volume is based on.

Editors: Billy Editor and Bill Editors; pp. 1–4



Leibniz International Proceedings in Informatics

Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

2 The Basic

An abstract version of the proposed algorithm, geared towards volumes bounded by smooth surfaces, can be described as follows. Figure 1 illustrates the steps in 2D.

1. Take as input a σ -sparse ϵ -sampling \mathcal{P} on the bounding surface \mathcal{M} of the volume \mathcal{O} .
2. Define a ball B_i of radius $r_i = \delta \text{fls}(p_i)$ centered at each sample p_i . Let $U = \cup_i B_i$.
3. The corner points of ∂U yield the Voronoi generators $\mathcal{G} = \mathcal{G}_I \cup \mathcal{G}_O$ inside and outside \mathcal{O} .
4. Include in \mathcal{G} more generators, away from \mathcal{M} , inside \mathcal{O} to further decompose it.
5. Compute the Voronoi diagram $\text{Vor}(\mathcal{G})$. The mesh $\hat{\mathcal{O}} = \text{Vor}(\mathcal{G})$ approximates \mathcal{O} and the facets separating \mathcal{G}_I and \mathcal{G}_O form the surface reconstruction $\hat{\mathcal{M}}$ approximating \mathcal{M} .

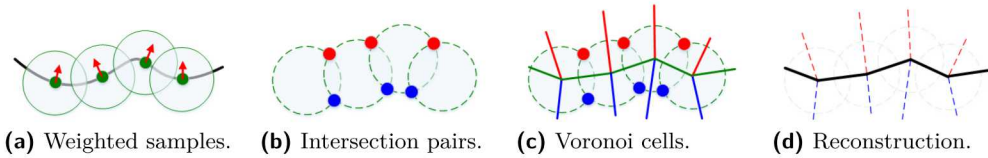


Figure 1 VoroCrust reconstruction, demonstrated on a planar curve. The weight of a point defines the radius of a ball around it. The reconstruction is the Voronoi facets separating the uncovered intersection pairs on opposite sides of the manifold.

For appropriate values of δ , the union of balls U is isotopic to the bounding surface \mathcal{M} [7]. In addition, the Voronoi facets in $\text{Vor}(\mathcal{G})$ separating \mathcal{G}_I and \mathcal{G}_O are in fact the medial axis of U [4], which is immediately isotopic to \mathcal{M} as well [6]. The sparsity condition helps in bounding the quality of mesh elements; the cells are fat. Finally, the enclosed volume can easily be refined without disrupting the surface reconstruction [10]. See [1] for the details and [2] for some earlier related work.

3 The Real

Unlike the idealized setting studied above, realistic inputs pose a number of challenges. Namely, sharp features do not fit well with the ϵ -sampling paradigm and require extra care to bound the quality of mesh elements in their neighborhoods. Even without sharp features, estimating the local feature size is impractical. In addition, for applications that require surface approximations with good normals, extra steps are needed to iron out any irregularities that the basic approach might yield. We address these issues in forthcoming publications.

4 The Illustration

We follow the tradition of illustrating the outputs of our algorithm applied to the infamous model known as the Stanford Bunny [12]. We used *ParaView* [3] to render our 3D models and the *Processing* programming language to generate our animations. We synthesized speech using the *Bing Speech API* [8] and composed the video using *OpenShot* [11].

References

- 1 Ahmed Abdelkader, Chandrajit L. Bajaj, Mohamed S. Ebeida, Ahmed H. Mahmoud, Scott A. Mitchell, John D. Owens, and Ahmed A. Rushdi. Sampling Conditions for Conforming Voronoi Meshing by the VoroCrust Algorithm (accepted for publication). In *34th International Symposium on Computational Geometry (SoCG 2018)*, Leibniz International Proceedings in Informatics (LIPIcs), 2018.
 - 2 Ahmed Abdelkader, Chandrajit L. Bajaj, Mohamed S. Ebeida, and Scott A. Mitchell. A Seed Placement Strategy for Conforming Voronoi Meshing. In *Canadian Conference on Computational Geometry*, 2017.
 - 3 James Ahrens, Berk Geveci, and Charles Law. Paraview: An end-user tool for large-data visualization. In Charles D. Hansen and Chris R. Johnson, editors, *Visualization Handbook*, pages 717 – 731. Butterworth-Heinemann, Burlington, 2005.
 - 4 Nina Amenta and Ravi Krishna Kolluri. The medial axis of a union of balls. *Computational Geometry*, 20(1):25 – 37, 2001. Selected papers from the 12th Annual Canadian Conference.
 - 5 Joseph E. Bishop. Simulating the pervasive fracture of materials and structures using randomly close packed Voronoi tessellations. *Computational Mechanics*, 44(4):455–471, September 2009.
 - 6 Frédéric Chazal and David Cohen-Steiner. A condition for isotopic approximation. *Graphical Models*, 67(5):390 – 404, 2005. Solid Modeling and Applications.
 - 7 Frédéric Chazal and André Lieutier. Smooth manifold reconstruction from noisy and non-uniform approximation with guarantees. *Computational Geometry*, 40(2):156 – 170, 2008.
 - 8 Microsoft Corporation. Bing Speech API.
 - 9 Mohamed S. Ebeida and Scott A. Mitchell. Uniform random Voronoi meshes. In *International Meshing Roundtable (IMR)*, pages 258–275, 2011.
 - 10 Scott A. Mitchell and Stephen A. Vavasis. Quality mesh generation in three dimensions. In *Proceedings of the Eighth Annual Symposium on Computational Geometry*, SCG '92, pages 212–221, New York, NY, USA, 1992. ACM.
 - 11 OpenShot. OpenShot Video Editor.
 - 12 Greg Turk and Marc Levoy. Zippered polygon meshes from range images. In *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '94, pages 311–318, New York, NY, USA, 1994. ACM.
- 

A The Narrative

This is an illustration of the VoroCrust algorithm for Voronoi meshing, highlighting the core theoretical concepts and further technical challenges. We study the problem of generating a Voronoi mesh that naturally conforms to a given closed surface M : a subset of the cells in the generated mesh decomposes the volume \mathcal{O} enclosed by the surface, while the facets on the boundary of the union of those cells yields a surface approximation of M . Our work is motivated by recent interest in polyhedral meshes as a more efficient alternative to tetrahedral meshes in certain applications.

We introduce the core principles underlying the proposed solution in an ideal setting easy to describe in 2D. We start with an epsilon-sample on a smooth surface M . Assuming the local feature size can be evaluated, we place a ball of radius δ times the local feature size at each sample. For appropriate values of δ , the union of such balls yields a good approximation of the surface M . In particular, the two surfaces defined by the inside and outside boundaries of the union of balls are isotopic to M .

Our next step is to place a set of Voronoi seeds at the corner points of the union of balls, where each seed is labeled as either inside or outside the enclosed volume \mathcal{O}_h . The Voronoi facets common to one interior and one exterior seeds constitute the medial axis of the union of balls. As the medial axis is itself a surface isotopic to both boundaries of the union of balls, it is immediately isotopic to M as well. With the surface protected, the interior can be meshed easily.

We demonstrate output meshes obtained by a preliminary implementation, which is considerably different than the Sandia VoroCrust software. In order to guarantee that each input sample appears as a vertex in the resulting surface approximation, we require a sparsity condition on the input epsilon-sample. This also provides a lower-bound on the quality of surface elements, modulo certain artifacts that can be mitigated in a clean-up phase. To further decompose the enclosed volume \mathcal{O} , without disrupting the surface approximation, we generate more samples within \mathcal{O} away from the surface \mathcal{M} . The sparsity condition together with grading allow us to establish a lower-bound on the fatness of all cells.

The desired mesher takes as input a CAD model describing the object to be meshed. Having access to a CAD model allows the mesher to accurately sample new points as needed; a crucial requirement for quality meshing.

For realistic inputs, however, the situation is considerably more complicated than the ideal setting we used to establish our core theoretical guarantees. In particular, sharp features do not fit well with the epsilon-sampling paradigm and require extra care to bound the quality of mesh elements in their neighborhoods. Even without sharp features, estimating the local feature size is impractical. In addition, for applications that require surface approximations with good normals, extra steps are needed to iron out any irregularities that the basic approach might yield. We address these issues in a forthcoming publication and illustrate a number of sample meshes produced by our implementation.

For more information, please reach out to the point of contact at Sandia National Labs.