SAND2019-0266C

VoroCrust: Voronoi Meshing Without Clipping

ANONYMOUS AUTHOR(S)

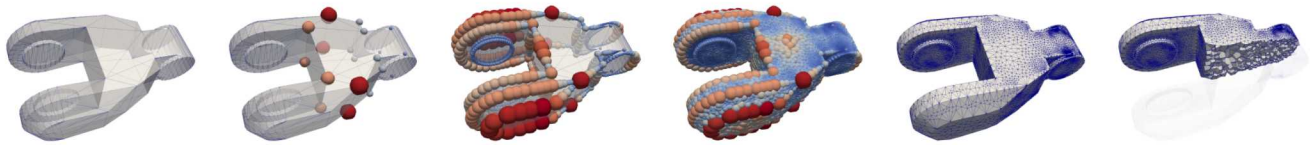


Fig. 1. VoroCrust main steps and results. From left to right: input CAD model with sharp features, corner spheres, edge spheres, surface spheres, surface mesh (unclipped Voronoi facets), and conforming Voronoi mesh.

Polyhedral meshes are increasingly becoming an attractive option with particular advantages over traditional meshes for certain applications. What has been missing is a robust polyhedral meshing algorithm that can handle broad classes of domains exhibiting both curved boundaries and sharp features. In addition, the power of primal-dual mesh pairs has been recognized as an important ingredient in multiple formulations. The VoroCrust algorithm is the first provably correct algorithm for conforming Voronoi meshing for non-convex and possibly non-manifold domains with guarantees on the quality of both surface and volume elements. A robust refinement process estimates a suitable sizing field that enables the placement of Voronoi seeds across the surface eliminating the need for clipping. The algorithm has the flexibility of filling the interior by either structured or random samples, while all sharp features are preserved in the output mesh. We demonstrate the capabilities of the algorithm on a variety of models and compare against state-of-the-art clipping-based methods establishing the clear advantage of VoroCrust output.

CCS Concepts: • **Computing methodologies** → **Computer Graphics**;

Additional Key Words and Phrases: Voronoi, Meshing, Refinement, Sharp Features, Union of Balls, Poisson-disk Sampling, Slivers

ACM Reference Format:
Anonymous Author(s). 2019. VoroCrust: Voronoi Meshing Without Clipping. *ACM Trans. Graph.* 1, 1 (January 2019), 10 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

The computational modeling of physical phenomena requires robust numerical algorithms and compatible high-quality domain discretizations. Finite element methods traditionally use simplicial or hexahedral meshes. Generalizations to arbitrary polyhedral elements have recently gained increasing attention in computer graphics [Martin et al. 2008], physically-based simulations [Bishop 2014], applied mathematics [Manzini et al. 2014], computational mechanics [Gain et al. 2014a] and computational physics [Lipnikov et al. 2014]. To ensure the fidelity of the discrete model, the fundamental properties of continuum equations have to be preserved [Desbrun et al. 2008]. A well-principled framework is enabled through the combined use of primal meshes and their orthogonal duals [Mullen et al. 2011]. The power of orthogonal duals, exemplified by Voronoi-Delaunay meshes, has recently been demonstrated on a range of applications in computer graphics [Goes et al. 2014] and computational physics [Engwirda 2018].

While the generation of tetrahedral meshes based on Delaunay refinement [Cheng et al. 2012] or variational optimization [Alliez et al. 2005] is well established, research on general polyhedral meshing is less mature. In this paper, we present VoroCrust: the first algorithm for meshing arbitrary non-convex, non-smooth, and possibly non-manifold domains by conforming Voronoi meshes. The crux of the algorithm is a robust refinement procedure that converges to a suitable sizing estimate yielding an isotopic volumetric mesh with guaranteed bounds on the quality and gradation of mesh elements while preserving all sharp features.

1.1 Background

Conventional mesh elements, as in tetrahedral and hexahedral meshes, often require excessive refinement when modeling complex geometries or domains undergoing large deformations, e.g., cutting, merging, fracturing, or adaptive refinement [Chen et al. 2014; Clausen et al. 2013; Wicke et al. 2010; Wojtan et al. 2009]. On the other hand, polyhedral elements can more easily adjust to deformation [Gain et al. 2014b; Martin et al. 2008] and topological changes [Wu et al. 2015], and are less biased to principal directions compared to regular tessellations [Talischi et al. 2013]. In addition, polyhedral elements typically have more neighbors, even at corners and boundaries, enabling better approximations of gradients and possibly more accurate solutions using the same number of conventional cells [CD-adapco 2014].

Unfortunately, robust algorithms for meshing general domains into polyhedra are still lacking. State-of-the-art approaches often rely on *clipping*, i.e., truncating cells of an initial mesh to fit the domain boundaries [Yan et al. 2010].

0730-0301/2019/1-ART $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

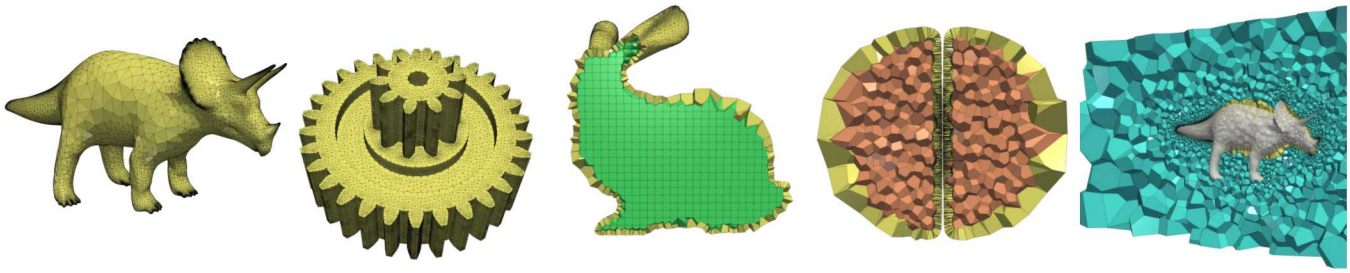


Fig. 2. VoroCrust handles different types of inputs and is capable of producing variable outputs depending on the user-application. This includes smooth (left), sharp-feature (second left) input surfaces, disconnected components (second right), and multiple material (right). VoroCrust allows different types of interior-fill Voronoi seeds e.g., random (right), seeds sampled from a grid (middle), or be subject of CVT optimization (second right).

Such an initial mesh can be obtained as a Voronoi mesh, e.g., with sites randomly generated inside the domain [Ebeida and Mitchell 2012] or optimized by centroidal Voronoi tessellations (CVT) [Yan et al. 2010]. Alternatively, an initial Voronoi mesh can be obtained by dualizing a conforming tetrahedral mesh [Garimella et al. 2014a]. Although no clipping is needed if the tetrahedralization is *well-centered*, generating such meshes is a very challenging problem [VanderZee et al. 2010]. A weaker *Gabriel property* ensures all tetrahedra have circumcenters inside the domain and can be guaranteed for polyhedral domains with bounded minimum angles [Si et al. 2010]; however, the dual Voronoi cells still need to be clipped.

While clipping is direct and efficient, it fails to produce true Voronoi cells, sacrificing key geometric properties [Ebeida and Mitchell 2012]. Specifically, clipping at sharp features may yield cells that are not convex, or even star shaped, which can be problematic for several applications [Beirão da Veiga et al. 2014; Wicke et al. 2007], e.g., for barycentric interpolation [Warren et al. 2007]. More importantly, a polyhedral Voronoi mesh has an orthogonal dual defined by the corresponding Delaunay mesh [Aurenhammer et al. 2013]. The combined use of primal meshes and their orthogonal duals has been recognized as an important framework for computational modeling [Desbrun et al. 2008; Mullen et al. 2011]

Orthogonal primal-dual mesh pairs are a class of unstructured staggered meshes [Harlow and Welch 1965] with desirable conservation properties [Perot 2000], enabling discretizations that closely mimic the continuum equations being modeled [Bochev and Hyman 2006; Desbrun et al. 2008]. The power of orthogonal duals [Mullen et al. 2011] was recognized in early works on structural design [Maxwell 1870; Rankine 1864] and numerical methods [Macneal 1953], and has recently been demonstrated on a range of applications in computer graphics [Goes et al. 2014], mesh parameterization [Mercat 2001], self-supporting structures [Akbarzadeh et al. 2015], and computational physics [Engwirda 2018]. In particular, Voronoi-Delaunay meshes are the default geometric realization to numerous formulations in numerical methods [Nicolaides and Wu 1997], fluid animation [Elcott et al. 2007], fracture modeling [Sukumar and Bolander 2009], and computational cell biology [Novak et al. 2007].

Despite many attempts to design a robust Voronoi meshing algorithm, a general solution to the problem remained elusive. For example, the TOUGH2 simulator for mass and heat transport in porous media [Pruess 1991] computes gradients along nodal lines connecting neighboring cells, and hence requires that these dual edges are orthogonal to the common primal facet [Pruess 2004]. Several heuristic approaches to the generation of Voronoi meshes for TOUGH2 were developed [Bonduà et al. 2017; Freeman et al. 2014; Hu et al. 2016; Kim et al. 2015; Klemetsdal et al. 2017]. The situation is further complicated for multi-material domains, where the difficulty of generating conforming meshes necessitates dealing with mixed elements straddling the interface between multiple materials [Dawes 2017; Garimella and Lipnikov 2011; Kikinzon et al. 2017].

1.2 Contributions

The VoroCrust algorithm is the first algorithm for conforming Voronoi meshing that can handle a large class of domains with both curved boundaries and arbitrarily sharp features. The crux of the algorithm is a robust refinement procedure that converges to a suitable sizing function enabling the placement of Voronoi seeds to approximate the surface. The algorithm eliminates the need for clipping, which is the current state-of-the-art for Voronoi meshing, successfully avoiding its drawbacks. The algorithm has the flexibility of decomposing the interior using either structured or randomly generated seeds. A careful implementation is able to guarantee the quality of the output mesh both on the surface and in the interior. We demonstrate the application of the algorithm through a variety of challenging models and compare against state-of-the-art clipping-based methods establishing the advantage of the VoroCrust algorithm.

The rest of the paper is organized as follows. We start by surveying related work in Section 2. Then, we present the VoroCrust algorithm in Section 3. We demonstrate the capabilities of the algorithm and compare its output against

state-of-the-art in Section 4. We discuss the limitations of the algorithm in Section 5. Finally, we conclude highlighting potential directions for future work in Section 6.

2 RELATED WORK

We summarize the most relevant related work under two categories: Voronoi meshing and surface reconstruction. For background on meshing models with sharp features, see Dey and Levine [Dey and Levine 2009] and Cheng et al. [Cheng et al. 2010] and the references therein.

2.1 Voronoi Meshing

Polyhedral meshing is an increasingly important area, both in research and in commercial contexts [Beirão da Veiga et al. 2014; Bishop 2009; Bolander and Sukumar 2005; Oaks and Paoletti 2000; Rashid and Selimotic 2006]. It is used for a variety of dynamic simulations, including complex time-dependent multiphysics such as mechanical operation, stresses, multiphase materials, combustion and chemical reactions, electromagnetics, heat, and acoustics. Polyhedral dual meshes are currently useful for fracture simulations, material grain modeling [Bishop 2009], and topology optimization [Cameron et al. 2010].

Polyhedral meshes have advantages over other types (e.g., tetrahedral) in terms of filling space with fewer elements for the same number of vertices, computing numerical gradients and flux, tolerating stretched elements, axis-insensitivity, etc. [Chin et al. 2015]. Voronoi meshes are one type of polyhedral meshes where facets are strictly planar. Each Voronoi cell includes all the points in the domain closer to its seed than any other seed, which results in the natural convexity and positive Jacobians of Voronoi cells.

Many prior techniques create a Voronoi mesh of a volume [Ebeida and Mitchell 2011; Lévy and Bonneel 2013; Yan et al. 2010, 2013]. They involve creating or moving seeds, forming their Voronoi cells, then *clipping* the cells by the model's boundary. The surface mesh matches the geometry of the manifold, but is unconstrained by any prior sample locations, and the samples are not (the only) vertices of the surface mesh. That is, meshing techniques do not simultaneously solve the surface reconstruction problem. Interior seeds may by created by dualizing a well-centered tetrahedral mesh [de Goes et al. 2014; Garimella et al. 2014b; VanderZee et al. 2010], or by direct sampling [Ebeida and Mitchell 2011].

The Voronoi seeds may also be moved through an optimization to generate initial or improved seed locations [de Goes et al. 2014; Du et al. 2010, 2003; Du and Wang 2005; Sieger et al. 2010]. A Centroidal Voronoi Tessellation (CVT) can be used to orient seeds with a field while preserving sharp features. Using an L_p-distance function, it is possible to create an arrangement of seeds so that many primal (tetrahedral) elements can be combined into cubical (hex) cells [Lévy and Liu 2010]. CubeCover [Nieser et al. 2011] also creates a hex-dominant mesh using a frame field by first generating a 3D parameterization of the volume.

Although clipping methods are direct and easy to implement, their output cells are not truly Voronoi cells, sacrificing several key geometric properties [Ebeida and Mitchell 2011]. Boundary cells are ill-shaped if the surface is ill-shaped. For curved surfaces, facets may be non-planar. Moreover, sharp features may yield cells that are not convex, or even star shaped; an important precondition for several applications [Beirão da Veiga et al. 2014].

An abstract version of the VoroCrust algorithm for smooth surfaces was analyzed by the authors [Abdelkader et al. 2018], under the assumption that an epsilon-sampling is provided as input. Assuming access to the local feature size, we established strong theoretical guarantees on the quality of the resulting Voronoi mesh in that restricted setting. In an earlier phase of this work, we explored the related problem of generating a Voronoi mesh that conforms to restricted classes of piecewise-linear complexes, with more challenging inputs left for future work [Abdelkader et al. 2017]. The approach adopted in [Abdelkader et al. 2017] does not use a union of balls and relies instead on similar ideas to those proposed for conforming Delaunay meshing [Cohen-Steiner et al. 2002; Murphy et al. 2001; Rand and Walkington 2009]. In contrast, the present submission describes the VoroCrust algorithm, as implemented in the VoroCrust software, which is based on a robust and practical refinement procedure that estimates a suitable sizing function for domains with possibly non-manifold boundaries exhibiting arbitrarily sharp features and narrow regions, enabling the placement of Voronoi seeds to capture a faithful approximation of the bounding surface preserving all sharp features while having the flexibility of meshing the interior using either structured or random Voronoi cells.

2.2 Surface Reconstruction

Surface reconstruction is motivated by numerous computer graphics applications [Berger et al. 2016; Yu 1999], reverse engineering [Toll and Cheng 1999; Várady et al. 1997], and computer vision, e.g., visualization [Katz and Tal 2015] and reconstruction of faces from images [Piotraschke and Blanz 2016]. In a typical design workflow, a prototype model is scanned to obtain a mesh for numerical simulations. To satisfy boundary conditions, the generated mesh has to conform to the model boundary.

Given a set of sample points from the surface of a model, surface reconstruction [Dey 2011] aims to approximate the surface as part of the output mesh. This is a challenging problem due to a number of factors including inaccuracies due to sampling noise and nonuniformity [Estellers et al. 2015; Kolluri et al. 2004], topology constraints [Zou et al. 2015], variable sample density [Marton et al. 2009], and the dependence on an approximation of the medial axis to preserve complex features, which is often unstable. Scalability to large data sets with minimal post-processing is another requirement for surface reconstruction tools [Boltcheva and Lévy 2016].

Algorithm 1: High-level VoroCrust algorithm

Input: PLC $\mathcal{T}$ approximating the domain $\mathcal{O}$,
and parameters θ, L and sz (Section 2.1)
$\mathcal{F} \leftarrow$ the set of sharp features w.r.t. θ (Section 2.2)
$\mathcal{B} \leftarrow$ a set of balls protecting all features in $\mathcal{F}$ (Section 2.3)
while $\mathcal{U}$ *does not cover* $\mathcal{T}$ **do**
Add balls to recover protection of $\mathcal{F}$ and cover $\mathcal{T}$
Shrink balls violating any ball conditions (Section 2.3)
or forming 4-way overlaps (Section 2.4)
end
$\mathcal{S}^{\updownarrow} \leftarrow$ pairs of seeds from triplets of overlapping balls in $\mathcal{B}$
$\mathcal{S}^{\downarrow\downarrow} \leftarrow$ seeds sampled from the interior of $\mathcal{O} \setminus \mathcal{U}$ (Section 2.5)
return $Vor(\mathcal{S}^{\updownarrow} \cup \mathcal{S}^{\downarrow\downarrow})$

3 THE VOROCRUST ALGORITHM

The crux of the algorithm is the generation of a set of weighted surface samples corresponding to a set of balls $\mathcal{B}$ whose union $\mathcal{U} = \cup\mathcal{B}$ approximates $\mathcal{M}$. Specifically, $\mathcal{U}$ covers $\mathcal{M}$ and has the same topology. In addition, $\mathcal{U}$ captures the sharp features of $\mathcal{M}$. To further guarantee the quality of surface approximation, the radii of surface balls vary smoothly and are sufficiently small w.r.t. the local curvature of $\mathcal{M}$. In other words, the radii of balls in $\mathcal{B}$ mimic a local feature size for $\mathcal{M}$. Finally, certain configurations of balls are perturbed to avoid having slivers in the resulting surface mesh. These requirements are used to design a refinement process that converges to a suitable union of balls. The conforming surface mesh is obtained by essentially dualizing $\mathcal{U}$ to obtain a set of Voronoi seeds $\mathcal{S}^{\updownarrow}$. Once $\mathcal{U}$ is obtained, the interior is easily meshed by sampling additional seeds $\mathcal{S}^{\downarrow\downarrow}$ outside $\mathcal{U}$. The output mesh is then computed as a subset of the Voronoi diagram of the seeds in $\mathcal{S}^{\updownarrow} \cup \mathcal{S}^{\downarrow\downarrow}$. In the remainder of this section, we elaborate on these steps per the high-level pseudocode above.

3.1 Input

The proposed algorithm can handle a domain $\mathcal{O}$ having as boundary a possibly non-manifold piecewise-smooth complex (PSC) $\mathcal{M}$. The algorithm takes as input a watertight piecewise-linear complex (PLC) $\mathcal{T}$ approximating the boundary $\mathcal{M}$. As in [Dey and Ray 2010], we assume that $\mathcal{T}$ approximates $\mathcal{M}$ in terms of both the Hausdorff error and the surface normals; this enables various predicates to be evaluated using the input PLC rather than the equations describing the underlying PSC [Cheng et al. 2010]. For the current implementation, we assume the input is a triangle surface mesh with no self-intersection. Well-established methods can be used to obtain such a mesh given a suitable representation of the domain $\mathcal{O}$ [Dey and Levine 2009; Hu et al. 2018; Tournois et al. 2009].

The boundary PSC $\mathcal{M}$ possibly contains *sharp features* where the normal to the surface does not vary smoothly. We make no assumption on how small the angles subtended by sharp features might be. The algorithm is provided with an angle threshold θ which is used to identify a set of sharp features in the PLC $\mathcal{T}$. The algorithm guarantees that all such sharp features are preserved; sharp corners appear exactly as vertices, while sharp edges are approximated by a set of edges in the output mesh. The algorithm is also provided with a smoothness parameter L that bounds the variation of radii in $\mathcal{B}$. The behavior of the refinement process depends crucially on both θ and L; see Section 2.3. Finally, the parameter sz, set to the diameter of $\mathcal{T}$ by default, is a sizing field that can be used to bound the size of elements in the output mesh.

3.2 Preprocessing

Before initiating the refinement process, the algorithm classifies and indexes the elements of the input PLC $\mathcal{T}$. Then, this information is used to construct a number of data structure for proximity queries against $\mathcal{T}$ and $\mathcal{U}$.

We define a *sharp edge* as an edge of $\mathcal{T}$ subtending a dihedral angle less than θ. All non-manifold edges incident to more than two facets are also treated as sharp edges. Similarly, a *sharp corner* is defined as a vertex of $\mathcal{T}$ incident to two facets whose normals differ by at least θ, or two sharp edges making an angle less than θ, or more than two sharp edges. A polyline arising from a chain of connected sharp edges is called a *crease*, and either forms a cycle or connects two sharp corners. The connected components of the boundary containing no sharp features, denoted $\mathcal{T}_S$, are called *surface patches*, and are possibly bounded by creases. The collection of sharp corners, creases and smooth patches are collectively referred to as the *features* of $\mathcal{T}$.

The algorithm uses θ to test each edge in $\mathcal{T}$, and collects all sharp edges in a set E. Then, each vertex is tested using E and θ, and the sharp corners are collected into the set $\mathcal{F}_C$. From E and $\mathcal{F}_C$, sharp edges are collected into the set of creases $\mathcal{F}_E$ by label propagation through common vertices except for sharp corners. As a byproduct, each crease is given an index and an orientation, applied consistently to all its sharp edges. Similarly, the facets of $\mathcal{T}$ are indexed, oriented and collected into the set of smooth patches $\mathcal{T}_S$ by label propagation across non-sharp edges. Finally, we set $\mathcal{F} = \mathcal{F}_C \cup \mathcal{F}_E$.

Upon generating a new sample point $p \in \mathcal{T}$, the algorithm needs to find the balls in $\mathcal{B}$ covering p, and estimate its distance to the elements of $\mathcal{T}$ satisfying certain conditions w.r.t. θ. Depending on whether p lies on an element in $\mathcal{F}_C$, $\mathcal{F}_E$ or $\mathcal{T}_S$, the queries need to be restricted to the respective set. To speed up such queries, the algorithm constructs three augmented kd-tree to index the elements in each set. The kd-trees for $\mathcal{F}_E$ and $\mathcal{T}_S$ are populated by supersampling the respective elements; see the supplementary materials for the details. Similarly, the balls in $\mathcal{B}$ are indexed into three kd-trees.

3.3 Ball Refinement

At a high level, the desired union of balls $\mathcal{U}$ has to (1) protect the sharp features of $\mathcal{T}$, and (2) cover $\mathcal{T}$ while matching its topology. The proposed algorithm achieves this through a set of *ball conditions* imposed on the balls in $\mathcal{B}$. Violations of these conditions drive a refinement process which converges

to a suitable union of balls. Before describing this process, we introduce a number of definitions and subroutines.

Smooth neighborhoods Fix a point $x \in \mathcal{T}$ and let σ be a face of $\mathcal{T}$ containing x. If σ is a sharp edge, define $v_{x,\sigma}$ as a unit vector parallel to σ. If σ is a smooth patch, define $v_{x,\sigma}$ as a unit vector normal to σ. $v_{x,\sigma}$ inherits the orientation of the crease or smooth patch containing σ. A path γ lying entirely in a unique crease or smooth patch Σ is called a *smooth path* iff for all $x, y \in \gamma$ we have that $\angle v_{x,\sigma}, v_{y,\tau} \leq \theta$, where σ and τ are the two top-dimensional faces of Σ containing x and y, respectively. Two points $x, y \in \mathcal{T}$ are called *co-smooth* iff they can be connected by a smooth path.

Ball conditions For a sample point $p \in \mathcal{T}$, let $b_p \in \mathcal{B}$ denote the ball centered at p and r_p denote its radius.

(C1) Coverage For any $b_p \in \mathcal{B}$ and all $x \in b_p \cap \mathcal{T}$, we require that p and x are co-smooth.

(C2) Overlap For any $b_p, b_q \in \mathcal{B}$ s.t. $b_p \cap b_q \neq \emptyset$, we require that $b_p \cup b_q$ contains a smooth path from p to q.

(C3) L-smoothness For any two balls $b_p, b_q \in \mathcal{B}$ such that either $p, q \in \mathcal{F}_C$ or both p and q lie on the same crease or smooth patch, we require that $r_p \leq r_q + L \cdot \|p - q\|$.

Sizing estimation A *sizing* assigns to each new sample p a radius r_p. We seek a sizing that satisfies all ball conditions. The algorithm computes such a sizing by dynamically evolving the assignments r_p for each ball $b_p \in \mathcal{B}$ in the course of the refinement process. To speed up convergence, a newly generated ball b_p is initialized with a conservative estimate which is more likely to satisfy all ball conditions. To help avoid coverage violations, we query the feature kd-trees using p to obtain a surrogate point q for the closest non-co-smooth point on $\mathcal{T}$. We set $r_p = \min(sz, 0.49 \cdot \|p - q\|)$.

Sampling basics The refinement process uses Maximal Poisson-Disk Sampling (MPS) [Ebeida et al. 2011; Gamito and Maddock 2009] to generate the sample points needed to protect the creases and cover the smooth patches. The MPS algorithm maintains an *active pool* of faces, initialized by all faces on the feature to be meshed. Each face in the active pool can be chosen for sampling with a probability proportional to its measure, defined as the length for edges or area for facets. To generate a new sample p, the chosen face is sampled uniformly at random. If p is not covered by the balls in $\mathcal{B}$, it is assigned a radius r_p and the ball b_p is added into $\mathcal{B}$. Otherwise, the algorithm increments a *miss counter* and discards p. When the miss counter reaches a preset maximum value, taken as 100 in our implementation, all faces in the active pool are *subdivided* into *subfaces* and the counter is reset; edges are split in half and facets are star-decomposed. The algorithm discards any subface that is completely covered and the remaining subfaces become the new active pool.

Deep coverage For any point $x \in \mathcal{T}$, we require a stronger form of coverage by the balls in $\mathcal{B}$. We say that $x \in \mathcal{T}$ is *deeply covered* by a ball $b_p \in \mathcal{B}$, for $\alpha < 1$, if $\|p - x\| \leq \alpha \cdot r_p$; we set $\alpha = 0.5$ in our implementation. The reason for that is two fold. First, any point x in the proximity of a crease Σ must be closer to the weighted samples on Σ than the samples on any other feature of $\mathcal{T}$. Second, to ensure a sufficient distance between the pairs of seeds generated by triplets of overlapping balls, we require that adjacent balls intersect *deeply*. The refinement process achieves this by modifying the coverage test for MPS as follows. First, a new sample is only accepted if it is *not* deeply covered. Second, upon subdividing a face in the active pool, a subface is discarded only if it is completely deeply covered by a single ball. To apply this coverage test to a point $x \in \mathcal{T}$, the algorithm finds the nearest sample of each type using the respective ball kd-tree. Then, the algorithm queries the trees for neighboring balls and checks whether deep coverage condition is satisfied for any of these balls.

Detecting violations Before MPS discards a subface, the algorithm checks for violations of C1 and shrinks encroaching balls as necessary. A subface is safely discarded only if of its points are deeply covered by a single ball with a co-smooth center. The algorithm checks for potential violations of C3 by checking the neighboring balls of any ball that gets shrunk; shrinking possibly cascades through a relatively large fraction of the balls in $\mathcal{B}$ until C3 is satisfied. The algorithm only checks for violations of C2 before terminating the refinement phase. The reason is that C2 requires checking the existence of a smooth path, which is an expensive operation involving the computation of geodesics on $\mathcal{T}$. Hence, checking this condition is deferred as long as there is a chance that balls may be shrunk to satisfy the other conditions.

Protection and coverage The refinement process is realized as a recursive MPS (RMPS) algorithm that goes through three phases, ordered by the dimension of the underlying feature, starting with the protection of sharp corners to the protection of creases and finally the coverage of smooth patches. At each phase, if refinement shrinks any of the balls belonging to a previous phase, the algorithm recurses by rerunning RMPS on the affected lower dimensional feature before proceeding. The process starts by initializing the set of balls with one *corner ball* centered at each sharp corner. As the base case of RMPS, the algorithm enforces C3 among corner balls by brute-force, i.e., each ball is checked against the rest and is shrunk as needed. Then, each crease Σ is protected by a set of *edge balls* by running RMPS on Σ. If any corner ball has to be shrunk, RMPS recurses to adjust corner balls. After successfully protecting all creases, the algorithm proceeds to cover each smooth patch Σ by running RMPS on Σ. Similarly, if any corner or edge ball has to be shrunk, RMPS recurses to the respective phase.

We point out that extra care is needed to avoid the well-known clustering phenomenon resulting from the greedy generation of samples. This can be mitigated by biasing the sampling to avoid introducing new sample points at the boundaries of existing balls. In addition, non-manifold boundaries require a slight adjustment of the smoothness condition. We defer these details to the supplementary material.

3.4 Sliver Elimination

Before terminating the refinement process, the algorithm restarts RMPS to search for problematic configurations of

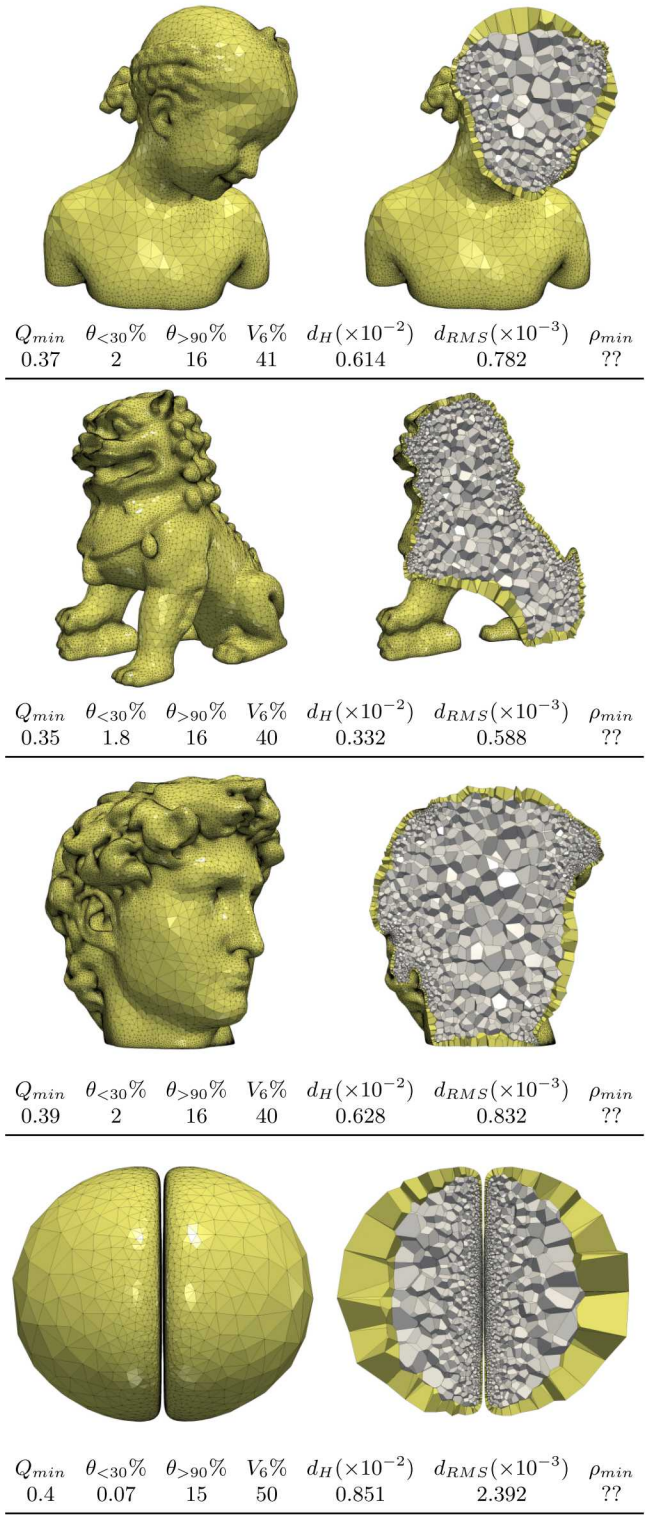
balls. Observe that if coverage is completed, with all ball conditions satisfied, no additional samples would be added. The algorithm proceeds with facet subdivision and keeps track of subfaces that lie completely in a single ball. A subface that lies completely in the intersection of 3 balls is called a *witness* to the 3-way overlap of those balls; any subface overlapping less than 3 balls is discarded. For every triplet of overlapping balls, the algorithm computes the potential Voronoi seeds arising on the intersection of their bounding spheres. Then, the algorithm checks for *half-covered seeds*, i.e., pairs of seeds where exactly one seed is contained in a *fourth* ball of $\mathcal{B}$. The presence of half-covered seeds results in extra undesirable Steiner vertices one the surface of the output mesh [Abdelkader et al. 2018]. The facets incident to such Steiner vertices may not be aligned with the input surface, which decreases the quality of the resulting surface mesh. To eliminate these defects, the algorithm resolves half-covered seeds by shrinking some of the balls in such problematic configurations; see the supplementary materials for more details. After shrinking, the algorithm restarts the refinement process to recover the protection and coverage properties of $\mathcal{B}$ while satisfying all ball conditions. Finally, to speed up convergence, the resolution of half-covered seeds is interleaved with the refinement process; refer to the supplementary material for an evaluation of this strategy.

3.5 Volume Meshing

Once the refinement process terminates, the set of balls $\mathcal{B}$ is fixed and a conforming surface mesh can be generated. To further decompose the interior into a set of graded Voronoi cells, additional weighted samples are generated in the interior of the domain. Similar to surface balls, the balls corresponding to interior samples are required to satisfy the L-smoothness condition. Standard MPS may be used for sampling the interior. However, to reduce the memory footprint of this step, the spoke-darts algorithm is used instead [Mitchell et al. 2018]; see the supplementary materials for details. Alternatively, the interior samples may be chosen as the vertices of a structured lattice; this can be used to output a hex-dominant mesh conforming to the surface. The quality of the resulting volume mesh can be further improved by applying CVT before computing the Voronoi diagram.

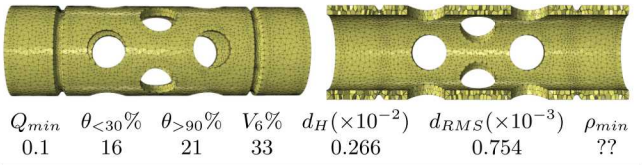
4 RESULTS

We demonstrate the capabilities of the VoroCrust algorithm on a variety of models ranging from smooth models as in Figure 3 to models with sharp features as in Figure 4. In both cases, we report the quality of surface approximation in terms of the Hausdorff error (as a percentage of the diagonal of the bounding box), the root mean squared distance, the minimum triangle quality, as well as the percentage of surface triangles with angles less than 30 or more than 90 degrees. To assess the quality of volumetric cells, we report the worst aspect ratio over the entire mesh.

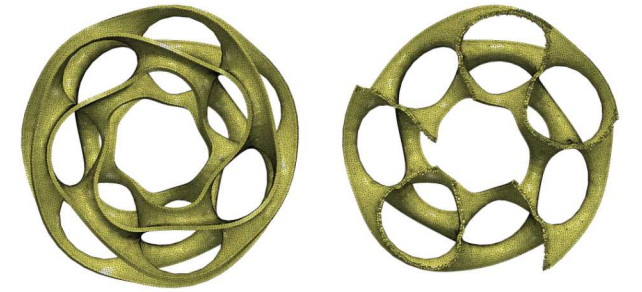


Q_{min}	$\theta_{<30}\%$	$\theta_{>90}\%$	$V_6\%$	$d_H(\times10^{-2})$	$d_{RMS}(\times10^{-3})$	ρ_{min}
0.37	2	16	41	0.614	0.782	??
0.35	1.8	16	40	0.332	0.588	??
0.39	2	16	40	0.628	0.832	??
0.4	0.07	15	50	0.851	2.392	??

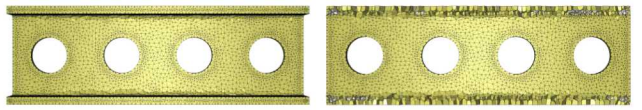
Fig. 3. Q_{min} is the minimum triangle quality. $\theta_{<30}\%$ and $\theta_{>90}\%$ are the percentage triangles with angle $< 30°$ and $> 90°$, respectively. $V_6\%$ is the percentage of surface mesh vertices with valence 6. d_{RMS} is the root mean square distance and d_H is Hausdorff distance normalized by the diameter of the bounding box. ρ_{min} is the minimum Voronoi cell aspect ratio.



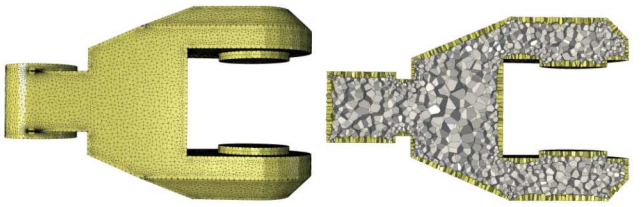
Q_{min}	$\theta_{<30}\%$	$\theta_{>90}\%$	$V_6\%$	$d_H(\times10^{-2})$	$d_{RMS}(\times10^{-3})$	ρ_{min}
0.1	16	21	33	0.266	0.754	??



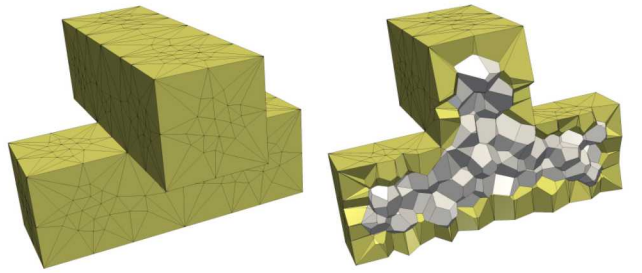
Q_{min}	$\theta_{<30}\%$	$\theta_{>90}\%$	$V_6\%$	$d_H(\times10^{-2})$	$d_{RMS}(\times10^{-3})$	ρ_{min}
0.273	11	19	35	0.087	0.124	??



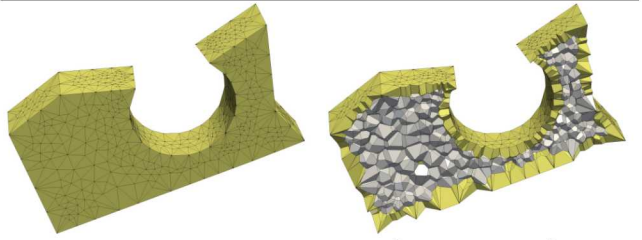
Q_{min}	$\theta_{<30}\%$	$\theta_{>90}\%$	$V_6\%$	$d_H(\times10^{-2})$	$d_{RMS}(\times10^{-3})$	ρ_{min}
0.221	14	22	33	0.202	0.205	??



Q_{min}	$\theta_{<30}\%$	$\theta_{>90}\%$	$V_6\%$	$d_H(\times10^{-2})$	$d_{RMS}(\times10^{-3})$	ρ_{min}
0.0303	14	21	36	0.098	0.398	??



Q_{min}	$\theta_{<30}\%$	$\theta_{>90}\%$	$V_6\%$	$d_H(\times10^{-2})$	$d_{RMS}(\times10^{-3})$	ρ_{min}
0.236	35	30	27	0	0	??



Q_{min}	$\theta_{<30}\%$	$\theta_{>90}\%$	$V_6\%$	$d_H(\times10^{-2})$	$d_{RMS}(\times10^{-3})$	ρ_{min}
0.2	29	30	29	0.025	0.0032	??

Fig. 4. .

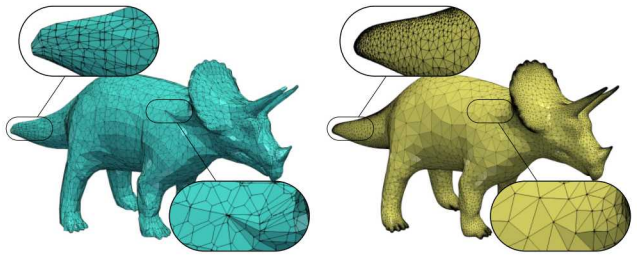


Fig. 5. Clipping Voronoi cells [Yan et al. 2010] is sensitive to the input surface tessellation (left). VoroCrust (right) is capable of producing high quality surface mesh alongside the 3D Voronoi mesh.

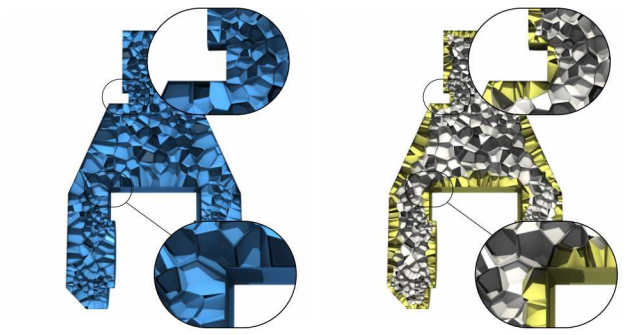


Fig. 6. Clipping Voronoi cells near the surface boundaries may destroy the Voronoi cells convexity (left). VoroCrust (right) guarantees to produce convex Voronoi cells everywhere irregardless to the surface boundaries.

To further demonstrate the advantage of the VoroCrust algorithm against state-of-the-art methods based on clipping, we compare against the restricted Voronoi diagram (RVD) of [Yan et al. 2009]. As shown in Figures 5, VoroCrust exhibits superior quality in terms of the surface mesh, where the RVD exhibits an imprint of the input mesh with many small facets. For the non-convex domain in 6, the RVD results in non-convex cells in the decomposition while VoroCrust is able to conform to the boundary with true Voronoi cells while maintaining the quality of the surface elements.

5 LIMITATIONS

The main limitation of the presented algorithm is the apparent restriction on the placement of surface seeds. In particular, this limits the quality of Voronoi cells in the vicinity of the surface. While mesh improvement techniques can be applied to perturb bad elements, this risks compromising the surface approximation. The most troubling aspect is the presence of short Voronoi edges in the output mesh, which can be a limiting factor in many applications.

Another limitation is the isotropic nature of the proposed sampling routine. This potentially leads to the generation of an unnecessarily large number of cells in narrow regions.

This can be mitigated by utilizing so-called boundary layers of elongated cells that better match the boundary.

Finally, the algorithm the input triangulation is a faithful approximation of the underlying domain. Hence, the algorithm may not be able to handle noisy inputs or unclean input meshes.

6 CONCLUSION

We presented VoroCrust, the first algorithm for conforming Voronoi meshing that can robustly handle a large class of domains containing having both curved boundaries and arbitrarily sharp features. The core of the algorithm is a robust refinement procedure to estimate a suitable sizing function enabling the placement of Voronoi seeds to capture the boundary while having the flexibility of decomposing the interior using either structured or randomly generated samples. We demonstrated the capabilities of the algorithm using a variety of models and compared against state-of-the-art clipping-based methods establishing the advantage of the proposed VoroCrust algorithm.

For future work, we consider the generation of boundary layers, which is very important in many applications. In the same spirit, extending the algorithm to align the cells according to a given anisotropy would be the natural next step. We believe that the VoroCrust refinement can be extended to accommodate additional requirements catering to the quality of the cells while preserving the surface approximation. In particular, eliminating short Voronoi edges is a critical requirement as it factors into the time step in numerical simulations.

ACKNOWLEDGEMENT

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 multi-mission laboratory managed and operated by National Technology and 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

Ahmed Abdelkader, Chandrajit L. Bajaj, Mohamed S. Ebeida, Ahmed H. Mahmoud, Scott A. Mitchell, John D. Owens, and Ahmad A. Rushdi. 2018. Sampling Conditions for Conforming Voronoi Meshing by the VoroCrust Algorithm. In *34th International Symposium on Computational Geometry (SoCG 2018) (Leibniz International Proceedings in Informatics (LIPIcs))*, Vol. 99. 1:1–1:16.

Ahmed Abdelkader, Chandrajit L. Bajaj, Mohamed S. Ebeida, and Scott A. Mitchell. 2017. Shattering an Embedded Triangulation by Voronoi Faces. In *Canadian Conference on Computational Geometry*. in submission.

Masoud Akbarzadeh, Tom Van Mele, and Philippe Block. 2015. On the equilibrium of funicular polyhedral frames and convex polyhedral force diagrams. *Computer-Aided Design* 63 (2015), 118 – 128.

Pierre Alliez, David Cohen-Steiner, Mariette Yvinec, and Mathieu Desbrun. 2005. Variational Tetrahedral Meshing. *ACM Trans. Graph.* 24, 3 (July 2005), 617–625.

Franz Aurenhammer, Rolf Klein, and Der-Tsai Lee. 2013. *Voronoi diagrams and Delaunay triangulations*. Vol. 8. World Scientific.

L. Beirão da Veiga, F. Brezzi, L. D. Marini, and A. Russo. 2014. The Hitchhiker's Guide to the Virtual Element Method. *Mathematical Models and Methods in Applied Sciences* 24, 08 (2014), 1541–1573.

Matthew Berger, Andrea Tagliasacchi, Lee M. Seversky, Pierre Alliez, Gaël Guennebaud, Joshua A. Levine, Andrei Sharf, and Claudio T. Silva. 2016. A Survey of Surface Reconstruction from Point Clouds. *Computer Graphics Forum* (2016). https://doi.org/10.1111/cgf.12802

J.E. Bishop. 2014. A displacement-based finite element formulation for general polyhedra using harmonic shape functions. *Internat. J. Numer. Methods Engrg.* 97, 1 (2014), 1–31.

Joseph E. Bishop. 2009. Simulating the pervasive fracture of materials and structures using randomly close packed Voronoi tessellations. *Computational Mechanics* 44, 4 (Sept. 2009), 455–471. https://doi.org/10.1007/s00466-009-0383-6

Pavel B. Bochev and James M. Hyman. 2006. Principles of Mimetic Discretizations of Differential Operators. In *Compatible Spatial Discretizations*. 89–119.

J. E. Bolander and N. Sukumar. 2005. Irregular lattice model for quasistatic crack propagation. *Phys. Rev. B* 71 (Mar 2005), 094106. Issue 9. https://doi.org/10.1103/PhysRevB.71.094106

Dobrina Boltcheva and Bruno Lévy. 2016. *Simple and Scalable Surface Reconstruction*. Technical Report hal-01349023v2. LORIA-Université de Lorraine; INRIA Nancy.

S. Bonduà, V. Bortolotti, P. Macini, E. Mesini, and E. M. Vasini. 2017. 3D Voronoi Pre- and Post- Processing Tools for Using the Tough2 Family of Numerical Simulator for Hydrocarbon Gas Migration. In *Offshore Mediterranean Conference*.

Talischi Cameron, Paulino Glaucio H., Pereira Anderson, and Menezes Ivan F. M. 2010. Polygonal finite elements for topology optimization: A unifying paradigm. *Internat. J. Numer. Methods Engrg.* 82, 6 (2010), 671–698.

CD-adapco. 2014. Polyhedral Mesh Generation. Available online, click `link`. (2014). Date accessed 8 May 2014.

Zhili Chen, Miaojun Yao, Renguo Feng, and Huamin Wang. 2014. Physics-inspired Adaptive Fracture Refinement. *ACM Trans. Graph.* 33, 4, Article 113 (July 2014), 113:1–113:7 pages.

Siu-Wing Cheng, Tamal K. Dey, and Edgar A. Ramos. 2010. Delaunay Refinement for Piecewise Smooth Complexes. *Discrete & Computational Geometry* 43, 1 (01 Jan 2010), 121–166.

Siu-Wing Cheng, Tamal K Dey, and Jonathan Shewchuk. 2012. *Delaunay Mesh Generation*. CRC Press.

Eric B. Chin, Jean B. Lasserre, and N. Sukumar. 2015. Numerical integration of homogeneous functions on convex and nonconvex polygons and polyhedra. *Computational Mechanics* 56, 6 (01 Dec 2015), 967–981.

Pascal Clausen, Martin Wicke, Jonathan R. Shewchuk, and James F. O'Brien. 2013. Simulating Liquids and Solid-liquid Interactions with Lagrangian Meshes. *ACM Trans. Graph.* 32, 2, Article 17 (April 2013), 17:1–17:15 pages.

D. Cohen-Steiner, É.-C. de Verdière, and M. Yvinec. 2002. Conforming Delaunay Triangulations in 3D. In *Proceedings of the Eighteenth Annual Symposium on Computational Geometry (SCG '02)*. 199–208.

A.S. Dawes. 2017. Three-dimensional multi-material polyhedral method for diffusion. *Computers & Fluids* 156 (2017), 485 – 495.

Fernando de Goes, Pooran Memari, Patrick Mullen, and Mathieu Desbrun. 2014. Weighted Triangulations for Geometry Processing. *ACM Transactions on Graphics (TOG)* 33, 3 (May 2014), 28:1–28:13.

Mathieu Desbrun, Eva Kanso, and Yiying Tong. 2008. *Discrete Differential Forms for Computational Modeling*. Oberwolfach Seminars, Vol. 38. Birkhäuser Basel, 287–324.

Tamal K. Dey. 2011. *Curve and Surface Reconstruction: Algorithms with Mathematical Analysis*. Cambridge University Press.

Tamal K. Dey and Joshua A. Levine. 2009. Delaunay Meshing of Piecewise Smooth Complexes without Expensive Predicates. *Algorithms* 2, 4 (2009), 1327–1349.

Tamal K. Dey and Tathagata Ray. 2010. Polygonal surface remeshing with Delaunay refinement. *Engineering with Computers* 26, 3 (2010), 289–301.

Qiang Du, Max Gunzburger, and Lili Ju. 2010. Advances in studies and applications of centroidal Voronoi tessellations. *Numerical Mathematics: Theory, Methods and Applications* 3, 2 (2010), 119–142.

Qiang Du, Max D. Gunzburger, and Lili Ju. 2003. Constrained Centroidal Voronoi Tessellations for Surfaces. *SIAM Journal on Scientific Computing* 24, 5 (2003), 1488–1506. https://doi.org/10.1137/S1064827501391576

Qiang Du and Desheng Wang. 2005. The optimal centroidal Voronoi tessellations and the Gersho's conjecture in the three-dimensional space. *Computers & Mathematics with Applications* 49, 9 (May 2005), 1355–1373.

Mohamed S. Ebeida and Scott A. Mitchell. 2011. Uniform Random Voronoi Meshes. In *International Meshing Roundtable (IMR)*. 258–275.

Mohamed S. Ebeida and Scott A. Mitchell. 2012. Uniform Random Voronoi Meshes. In *Proceedings of the 20th International Meshing Roundtable*. 273–290.

Mohamed S. Ebeida, Anjul Patney, Scott A. Mitchell, Andrew Davidson, Patrick M. Knupp, and John D. Owens. 2011. Efficient Maximal Poisson-Disk Sampling. *ACM Transactions on Graphics (TOG)* 30, 4 (2011), 49:1–49:12. https://doi.org/10.1145/1964921.1964944

Sharif Elcott, Yiying Tong, Eva Kanso, Peter Schröder, and Mathieu Desbrun. 2007. Stable, Circulation-preserving, Simplicial Fluids. *ACM Trans. Graph.* 26, 1, Article 4 (Jan. 2007).

Darren Engwirda. 2018. Generalised primal-dual grids for unstructured co-volume schemes. *J. Comput. Phys.* 375 (2018), 155 – 176.

Virginia Estellers, Michael Scott, Kevin Tew, and Stefano Soatto. 2015. Robust Poisson surface reconstruction. In *International Conference on Scale Space and Variational Methods in Computer Vision*. Springer, 525–537.

C.M. Freeman, K.L. Boyle, M. Reagan, J. Johnson, C. Rycroft, and G.J. Moridis. 2014. MeshVoro: A three-dimensional Voronoi mesh building tool for the TOUGH family of codes. *Computers & Geosciences* 70 (2014), 26 – 34.

Arun L. Gain, Cameron Talischi, and Glaucio H. Paulino. 2014a. On the Virtual Element Method for three-dimensional linear elasticity problems on arbitrary polyhedral meshes. *Computer Methods in Applied Mechanics and Engineering* 282 (2014), 132 – 160.

Arun L. Gain, Cameron Talischi, and Glaucio H. Paulino. 2014b. On the Virtual Element Method for three-dimensional linear elasticity problems on arbitrary polyhedral meshes. *Computer Methods in Applied Mechanics and Engineering* 282 (2014), 132 – 160.

Manuel N. Gamito and Steve C. Maddock. 2009. Accurate Multidimensional Poisson-Disk Sampling. *ACM Transactions on Graphics (TOG)* 29, 1 (Dec. 2009), 8:1–8:19.

Rao V. Garimella, Jibum Kim, and Markus Berndt. 2014a. Polyhedral Mesh Generation and Optimization for Non-manifold Domains. In *Proceedings of the 22nd International Meshing Roundtable*. 313–330.

Rao V. Garimella, Jibum Kim, and Markus Berndt. 2014b. Polyhedral Mesh Generation and Optimization for Non-manifold Domains. In *International Meshing Roundtable (IMR)*. Springer International Publishing, 313–330. https://doi.org/10.1007/978-3-319-02335-9_18

R. V. Garimella and K. Lipnikov. 2011. Solution of the diffusion equation in multi-material domains by sub-division of elements along reconstructed interfaces. *International Journal for Numerical Methods in Fluids* 65, 1112 (2011), 1423–1437.

Fernando de Goes, Pooran Memari, Patrick Mullen, and Mathieu Desbrun. 2014. Weighted Triangulations for Geometry Processing. *ACM Trans. Graph.* 33, 3, Article 28 (June 2014), 28:1–28:13 pages.

Francis H. Harlow and J. Eddie Welch. 1965. Numerical Calculation of TimeDependent Viscous Incompressible Flow of Fluid with Free Surface. *The Physics of Fluids* 8, 12 (1965), 2182–2189.

Litang Hu, Keni Zhang, Xiaoyuan Cao, Yi Li, and Chaobin Guo. 2016. IGMESH: A convenient irregular-grid-based pre- and post-processing tool for TOUGH2 simulator. *Computers & Geosciences* 95 (2016), 11 – 17.

Yixin Hu, Qingnan Zhou, Xifeng Gao, Alec Jacobson, Denis Zorin, and Daniele Panozzo. 2018. TetWild - Tetrahedral Meshing in the Wild. *ACM Transactions on Graphics (TOG)* 999, 999 (Aug. 2018), 99:1–99:99.

Sagi Katz and Ayellet Tal. 2015. On the Visibility of Point Clouds. In *Proceedings of the IEEE International Conference on Computer Vision*. IEEE, 1350–1358.

Evgeny Kikinzon, Yuri Kuznetsov, Konstatin Lipnikov, and Mikhail Shashkov. 2017. Approximate static condensation algorithm for solving multi-material diffusion problems on meshes non-aligned with material interfaces. *J. Comput. Phys.* 347 (2017), 416 – 436.

Seong-Kyun Kim, Gwang-Ok Bae, and Kang-Kun Lee. 2015. Improving accuracy and flexibility of numerical simulation of geothermal heat pump systems using Voronoi grid refinement approach. *Geosciences Journal* 19, 3 (01 Sep 2015), 527–535.

Ø S. Klemetsdal, R. L. Berge, K.-A. Lie, H. M. Nilsen, and O. Møyner. 2017. *SPE-182666-MS*. Chapter Unstructured Gridding and Consistent Discretizations for Reservoirs with Faults and Complex Wells.

Ravikrishna Kolluri, Jonathan Richard Shewchuk, and James F. O'Brien. 2004. Spectral Surface Reconstruction From Noisy Point Clouds. In *Eurographics Symposium on Geometry Processing (SGP04)*. 11–22.

Bruno Lévy and Nicolas Bonneel. 2013. Variational anisotropic surface meshing with Voronoi parallel linear enumeration. In *International Meshing Roundtable (IMR)*. Springer, 349–366.

Bruno Lévy and Yang Liu. 2010. L_p Centroidal Voronoi Tessellation and Its Applications. *ACM Transactions on Graphics (TOG)* 29, 4, Article 119 (July 2010), 11 pages. https://doi.org/10.1145/1778765.1778856

Konstantin Lipnikov, Gianmarco Manzini, and Mikhail Shashkov. 2014. Mimetic finite difference method. *J. Comput. Phys.* 257 (2014), 1163 – 1227. Physics-compatible numerical methods.

R. H. Macneal. 1953. AN ASYMMETRICAL FINITE DIFFERENCE NETWORK. *Quart. Appl. Math.* 11, 3 (1953), 295–310.

Gianmarco Manzini, Alessandro Russo, and N. Sukumar. 2014. New perspectives on polygonal and polyhedral finite element methods. *Mathematical Models and Methods in Applied Sciences* 24, 08 (2014), 1665–1699.

Sebastian Martin, Peter Kaufmann, Mario Botsch, Martin Wicke, and Markus Gross. 2008. Polyhedral Finite Elements Using Harmonic Basis Functions. *Computer Graphics Forum* 27, 5 (2008), 1521–1529.

Zoltan Csaba Marton, Radu Bogdan Rusu, and Michael Beetz. 2009. On fast surface reconstruction methods for large and noisy point clouds. In *IEEE International Conference on Robotics and Automation (ICRA '09)*. IEEE, 3218–3223.

James Clerk Maxwell. 1870. On Reciprocal Figures, Frames, and Diagrams of Forces. *Transactions of the Royal Society of Edinburgh* 26, 1 (1870), 140.

Christian Mercat. 2001. Discrete Riemann Surfaces and the Ising Model. *Communications in Mathematical Physics* 218, 1 (01 Apr 2001), 177–216.

Scott A. Mitchell, Mohamed S. Ebeida, Muhammad A. Awad, Chonhyon Park, Anjul Patney, Ahmad A. Rushdi, Laura P. Swiler, Dinesh Manocha, and Li-Yi Wei. 2018. Spoke-Darts for High-Dimensional Blue-Noise Sampling. *ACM Trans. Graph.* 37, 2, Article 22 (May 2018), 20 pages. https://doi.org/10.1145/3194657

Patrick Mullen, Pooran Memari, Fernando de Goes, and Mathieu Desbrun. 2011. HOT: Hodge-optimized Triangulations. *ACM Trans. Graph.* 30, 4, Article 103 (July 2011), 103:1–103:12 pages.

M. Murphy, D. Mount, and C. Gable. 2001. A POINT-PLACEMENT STRATEGY FOR CONFORMING DELAUNAY TETRAHEDRALIZATION. *International Journal of Computational Geometry & Applications* 11, 06 (2001), 669–682.

Roy A. Nicolaides and Xiaonan Wu. 1997. Covolume Solutions of Three-Dimensional Div-Curl Equations. *SIAM J. Numer. Anal.* 34, 6 (1997), 2195–2203.

M. Nieser, U. Reitebuch, and K. Polthier. 2011. CubeCover – Parameterization of 3D Volumes. *Computer Graphics Forum* 30, 5 (Aug. 2011), 1397–1406. https://doi.org/10.1111/j.1467-8659.2011.02014.x

Igor L. Novak, Fei Gao, Yung-Sze Choi, Diana Resasco, James C. Schaff, and Boris M. Slepchenko. 2007. Diffusion on a curved surface coupled to diffusion in the volume: Application to cell biology. *J. Comput. Phys.* 226, 2 (2007), 1271 – 1290.

Wayne Oaks and Stefano Paoletti. 2000. Polyhedral Mesh Generation. In *International Meshing Roundtable (IMR)*. 57–67.

Blair Perot. 2000. Conservation Properties of Unstructured Staggered Mesh Schemes. *J. Comput. Phys.* 159, 1 (2000), 58 – 89.

Marcel Piotraschke and Volker Blanz. 2016. Automated 3d face reconstruction from multiple images using quality measures. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 3418–3427.

K. Pruess. 1991. TOUGH2: A general-purpose numerical simulator for multiphase fluid and heat flow. *NASA STI/Recon Technical Report N* 92 (May 1991).

K. Pruess. 2004. The TOUGH CodesA Family of Simulation Tools for Multiphase Flow and Transport Processes in Permeable Media. *Vadose Zone Journal* 3 (2004).

Alexander Rand and Noel Walkington. 2009. Collars and Intestines: Practical Conforming Delaunay Refinement. In *Proceedings of the 18th International Meshing Roundtable*.

William John Macquorn Rankine. 1864. Principle of the equilibrium of polyhedral frames. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 27, 180 (1864), 92–92.

M. M. Rashid and M. Selimotic. 2006. A three-dimensional finite element method with arbitrary polyhedral elements. *Internat. J. Numer. Methods Engrg.* 67, 2 (July 2006), 226–252. https://doi.org/10.1002/nme.1625

H. Si, K. Gärtner, and J. Fuhrmann. 2010. Boundary Conforming Delaunay Mesh Generation. *Computational Mathematics and Mathematical Physics* 50, 1 (2010), 38–53.

Daniel Sieger, Pierre Alliez, and Mario Botsch. 2010. Optimizing Voronoi diagrams for polygonal finite element computations. In *International Meshing Roundtable (IMR)*. Springer, 335–350.

N Sukumar and JE Bolander. 2009. *Voronoi-based Interpolants for Fracture Modelling*. Springer-Verlag.

Cameron Talischi, Glaucio H. Paulino, Anderson Pereira, and Ivan F. M. Menezes. 2013. Polygonal finite elements for topology optimization: A unifying paradigm. *Internat. J. Numer. Methods Engrg.* 82, 6 (2013), 671–698.

Bill Toll and Fuhua Cheng. 1999. Surface reconstruction from point clouds. In *Machining Impossible Shapes*. Springer, 173–178.

Jane Tournois, Camille Wormser, Pierre Alliez, and Mathieu Desbrun. 2009. Interleaving Delaunay Refinement and Optimization for Practical Isotropic Tetrahedron Mesh Generation. *ACM Trans. Graph.* 28, 3, Article 75 (July 2009), 75:1–75:9 pages.

Evan VanderZee, Anil N. Hirani, Damrong Guoy, and Edgar A. Ramos. 2010. Well-Centered Triangulation. *SIAM Journal on Scientific Computing* 31, 6 (2010), 4497–4523. https://doi.org/10.1137/090748214

Tamás Várady, Ralph R Martin, and Jordan Cox. 1997. Reverse engineering of geometric models—an introduction. *Computer-Aided Design* 29, 4 (1997), 255–268. https://doi.org/10.1016/S0010-4485(96)00054-1

Joe Warren, Scott Schaefer, Anil N. Hirani, and Mathieu Desbrun. 2007. Barycentric coordinates for convex sets. *Advances in Computational Mathematics* 27, 3 (01 Oct 2007), 319–338.

Martin Wicke, Mario Botsch, and Markus Gross. 2007. A Finite Element Method on Convex Polyhedra. *Computer Graphics Forum* 26, 3 (2007), 355–364.

Martin Wicke, Daniel Ritchie, Bryan M. Klingner, Sebastian Burke, Jonathan R. Shewchuk, and James F. O'Brien. 2010. Dynamic Local Remeshing for Elastoplastic Simulation. *ACM Trans. Graph.* 29, 4, Article 49 (July 2010), 49:1–49:11 pages.

Chris Wojtan, Nils Thürey, Markus Gross, and Greg Turk. 2009. Deforming Meshes That Split and Merge. *ACM Trans. Graph.* 28, 3, Article 76 (July 2009), 76:1–76:10 pages.

Jun Wu, Rüdiger Westermann, and Christian Dick. 2015. A Survey of Physically Based Simulation of Cuts in Deformable Bodies. *Computer Graphics Forum* 34, 6 (2015), 161–187.

Dong-Ming Yan, Bruno Lévy, Yang Liu, Feng Sun, and Wenping Wang. 2009. Isotropic Remeshing with Fast and Exact Computation of Restricted Voronoi Diagram. *Computer Graphics Forum* 28, 5 (July 2009), 1445–1454. https://doi.org/10.1111/j.1467-8659.2009.01521.x

Dong-Ming Yan, Wenping Wang, Bruno Lévy, and Yang Liu. 2010. Efficient Computation of 3D Clipped Voronoi Diagram. In *Advances in Geometric Modeling and Processing*. Lecture Notes in Computer Science, Vol. 6130. Springer, 269–282. https://doi.org/10.1007/978-3-642-13411-1_18

Dong-Ming Yan, Wenping Wang, Bruno Lévy, and Yang Liu. 2013. Efficient Computation of Clipped Voronoi Diagram for Mesh Generation. *Computer-Aided Design* 45, 4 (April 2013), 843–852. https://doi.org/10.1016/j.cad.2011.09.004

Yizhou Yu. 1999. Surface reconstruction from unorganized points using self-organizing neural networks. In *IEEE Visualization*. IEEE, 61–64.

Ming Zou, Michelle Holloway, Nathan Carr, and Tao Ju. 2015. Topology-constrained surface reconstruction from cross-sections. *ACM Transactions on Graphics* 34, 4, Article 128 (July 2015), 10 pages. https://doi.org/10.1145/2766976