MATTopo: Topology-preserving Medial Axis Transform with Restricted Power Diagram

NINGNA WANG, University of Texas at Dallas, USA
HUI HUANG, Shenzhen University, China
SHIBO SONG, Independent Researcher, China
BIN WANG, Tsinghua University, China
WENPING WANG, Texas A&M University, USA
XIAOHU GUO[*], University of Texas at Dallas, USA

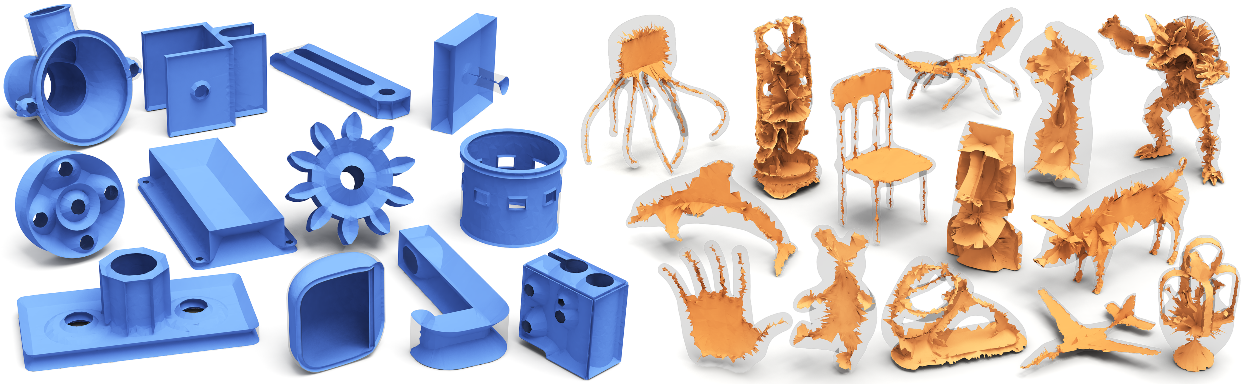


Fig. 1. A gallery of our topology-preserving 3D medial axis results. The input shapes are shown in transparency, while the computed medial axis are shown in blue (left) and brown (right) colors. The left side displays the medial axis computed on CAD models, and the right side shows those on organic models.

We present a novel topology-preserving 3D medial axis computation framework based on volumetric restricted power diagram (RPD), while preserving the medial features and geometric convergence simultaneously, for both 3D CAD and organic shapes. The volumetric RPD discretizes the input 3D volume into sub-regions given a set of medial spheres. With this intermediate structure, we convert the homotopy equivalency between the generated medial mesh and the input 3D shape into a localized contractibility checking for each restricted element (power cell, power face, power edge), by checking their connected components and Euler characteristics. We further propose a fractional Euler characteristic algorithm for efficient GPU-based computation of Euler characteristic for each restricted element on the fly while computing the volumetric RPD. Compared with existing voxel-based or point-cloud-based methods, our approach is the first to adaptively and directly revise the medial mesh without globally modifying the dependent structure, such as voxel size or sampling density, while preserving its topology and medial features. In comparison with the feature preservation method MATFP [Wang et al. 2022], our method provides geometrically comparable results with fewer spheres and more robustly captures the topology of the input 3D shape.

CCS Concepts: • **Computing methodologies** → **Shape analysis**.

Additional Key Words and Phrases: Medial Axis Transform, Topology Preservation, Feature Preservation, Restricted Power Diagram

ACM Reference Format:
Ningna Wang, Hui Huang, Shibo Song, Bin Wang, Wenping Wang, and Xiaohu Guo. 2024. MATTopo: Topology-preserving Medial Axis Transform with Restricted Power Diagram. *ACM Trans. Graph.* 43, 6, Article 218 (December 2024), 16 pages. https://doi.org/10.1145/3687763

1 Introduction

As a fundamental geometric structure1, the medial axis [Blum et al. 1967] captures the topological equivalence and geometric protrusions of the input shape. The medial axis $\mathcal{M}$ of a shape $\mathcal{S}$ is defined as the set of vertices with two or more nearest neighbors on the shape boundary $\partial\mathbf{S}$. The *medial axis transform* (MAT) is a combination of the medial axis and its radius function. The topological and geometric properties of medial axis allow it to become the foundation for other skeletal shape descriptors [Tagliasacchi et al. 2016]

[*] Corresponding author

Authors' Contact Information: Ningna Wang, University of Texas at Dallas, Texas, USA, ningna.wang@utdallas.edu; Hui Huang, Shenzhen University, Shenzhen, China, hhzhiyan@gmail.com; Shibo Song, Independent Researcher, Shanghai, China, longmaythess@outlook.com; Bin Wang, Tsinghua University, Beijing, China, wangbins@tsinghua.edu.cn; Wenping Wang, Texas A&M University, Texas, USA, wenping@tamu.edu; Xiaohu Guo, University of Texas at Dallas, Texas, USA, xguo@utdallas.edu.

ACM 1557-7368/2024/12-ART218
https://doi.org/10.1145/3687763

and has been used in approximating [Ge et al. 2023; Hu et al. 2022, 2023; Lan et al. 2020; Petrov et al. 2024; Yang et al. 2018, 2020], simplifying [Dou et al. 2022; Li et al. 2015; Yan et al. 2016], and analyzing shapes [Dou et al. 2020; Fu et al. 2022; Hu et al. 2019; Lin et al. 2021; Liu et al. 2024; Noma et al. 2024; Xu et al. 2023, 2024, 2022]. Some literature [Jalba et al. 2013; Kustra et al. 2016; Tagliasacchi et al. 2016] refers to the term 'medial axis' as 2D skeletons and uses 'medial surface' for 3D structures. For clarification, we consider 'medial axis' as a broader definition that includes both 2D and 3D. However, in this paper, we focus exclusively on the 3D medial axis.

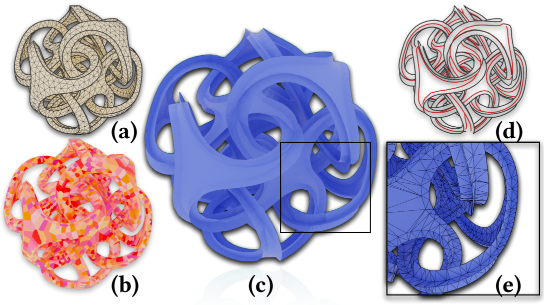


Fig. 2. We propose a novel volumetric RPD-based framework for computing the medial axis while preserving topology, medial features, and geometry. (a) Input tetrahedral mesh with pre-detected surface sharp features; (b) the RPD; (c) the generated medial mesh; (d) the generated external (in black) and internal (in red) features; (e) a zoomed-in view of the generated medial mesh.

Since an exact 3D medial axis is notoriously difficult to compute, most existing methods resort to computing an approximated MAT. This approximation aims to retain as many properties of the medial axis as possible, both topologically and geometrically, while being capable of handling various inputs, including smooth shapes (*i.e.,* organic models) and non-smooth shapes (*i.e.,* CAD models). Existing MAT approximation methods often consider these properties as a trade-off. For instance, the Voxel Cores (VC) method [Yan et al. 2018] provides a strong theoretical guarantee, allowing it to approximate the medial axes of C^2 smooth shapes with homotopy equivalence. However, it performs poorly in terms of geometric convergence when approximating CAD models with non-smooth external features (*i.e.,* convex sharp edges and corners). Recent progress for computing MAT has demonstrated advantages in preserving *medial features* (see Fig. 2 (d) and Sec. 3.1) as well as geometric convergence for CAD models using the surface restricted power diagram (RPD) based framework [Wang et al. 2022]. However, it cannot guarantee the topological preservation for its generated medial mesh w.r.t. to the input model, as their experiments show inconsistency of the Euler characteristics. The topology preservation property (also called 'homotopy equivalence') of the medial axis with respect to the input shape refers to the concept that these two spaces can be continuously deformed into one another without tearing or gluing, ensuring they share the same fundamental shape or structure. Without this property, the computed medial mesh may encounter challenges in various downstream applications. For example, Fig. 3 shows two medial meshes generated using MATFP [Wang et al. 2022] and our method for the Ant model, along with two skeletonization results from the Q-MAT algorithm [Li et al. 2015], using either method as the initial medial mesh for MAT simplification to identify significant and stable parts of the medial axis. Due to the lack of homotopy equivalence in MATFP [Wang et al. 2022], both the initial and simplified medial meshes for the Ant model exhibit 'broken' legs, whereas our method successfully preserves this property.

In this paper, we present a novel framework for computing a topology-preserving MAT that is homotopy-equivalent to the input 3D shape. Our framework can also preserve *medial features* (Sec. 3.1) and ensure the geometric approximation accuracy. We have found that the volumetric restricted power diagram (RPD, see the definition in Sec. 3.2) serves as a simple but effective intermediate structure between the input volumetric shape and the generated medial mesh. The medial mesh is generated as the dual structure of the volumetric RPD, followed by a thinning process (Sec. 5.3). This structure allows us to localize the topological inconsistencies between the restricted power elements (cells, faces, edges, vertices) and its dual mesh simplices (vertices, edges, triangles, tetrahedrons), based on the Nerve Theorem (Sec. 3.3). Owing to recent progress in GPU-based 3D power diagram computations [Basselin et al. 2021; Liu et al. 2020; Ray et al. 2018], we propose a volumetric RPD-based strategy for topological checking and fixing that amends the medial mesh in a local manner. Additionally, we introduce a novel *fractional Euler characteristic* strategy for efficient GPU-based computation of the Euler characteristic of each restricted element (such as cells, faces, edges) on the fly while computing the volumetric RPD.

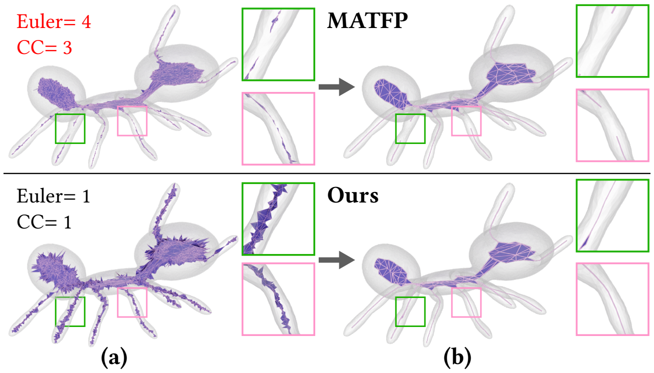


Fig. 3. Illustration of the importance of the medial axis's homotopy equivalence property. (a) Two medial meshes generated by MATFP [Wang et al. 2022] and our method. Both initial medial meshes contain approximately 5k medial spheres. The Euler characteristic of MATFP's mesh is 4, while our result is 1 (the ground truth is 1). The connected component (CC) results are 3 and 1, respectively (the ground truth is 1). (b) Two simplified medial meshes generated using the simplification algorithm Q-MAT [Li et al. 2015] with the target number of medial spheres set to 170. Both the initial and simplified medial meshes from MATFP [2022] exhibit 'broken' legs, whereas our method preserves the structure.

The initial medial mesh is generated using a small number of spheres (*e.g.,* 50 spheres), which is then revised iteratively by adding new spheres in local regions and updating the corresponding partial RPD. Each revision only happens locally, involving the checking of each individual restricted power cell (Sec. 4.1), their relation with neighboring power cells (Sec. 4.2), and their approximation to the input surface boundary (Sec. 4.3). Compared with existing point-cloud-based methods [Amenta et al. 2001; Miklos et al. 2010], voxel-based methods [Yan et al. 2018], and surface-RPD based method [Wang et al. 2022], our adaptive strategy can approximate the medial axis of both 3D CAD shapes and organic models with the preservation of homotopy equivalence, medial features, and geometric convergence, requiring a lower number of medial spheres.

The main contributions of this paper include:

- We present a complete volumetric RPD-based framework for computing 3D medial mesh of both CAD models and organic models, while preserving the homotopy equivalence, medial features, and geometric convergence.
- For input volumetric shapes with no cavities, we propose an adaptive partial-RPD updating strategy for topological checking and amendment. This approach involves localized assessment of the connected components and Euler characteristics of each restricted element.
- We propose a novel fractional Euler characteristic algorithm for the efficient GPU-based computation of Euler characteristic for each restricted cell, face, and edge on the fly while computing the volumetric RPD in parallel.
- We propose a GPU-based local geometric metric that enables us to adaptively refine the 3D medial mesh until its enveloping volume (see 12) achieves a desired geometric accuracy relative to the input shape.

2 Related Works

In this section, we review several representative works for approximating the 3D medial axis. Additionally, since our work relies on the computation of power diagrams in restricted domains, we also review algorithms for computing restricted Voronoi and power diagrams in volumes.

2.1 Medial Axis Computation

The approximation of the 3D medial axis mainly falls into three categories. Additionally, there are many regularization methods for pruning noisy branches of the medial axis [Dou et al. 2022; Li et al. 2015; Wang et al. 2024; Yan et al. 2016], which require either an initial approximation of the 3D medial axis or initial candidate inner balls as input. Please refer to survey materials [Siddiqi and Pizer 2008; Tagliasacchi et al. 2016] for further discussion on this topic.

Algebraic methods. These methods tend to create exact and analytic representation of MAT from a given boundary representation while tracing the features of medial axes from the shape boundary inward (*e.g.,* seam and junction) [Culver et al. 2004; Milenkovic 1993; Sherbrooke et al. 1996]. However, these methods are algorithmically challenging and computationally expensive due to the need for solving nonlinear systems, which limits their applicability to a simple class of shapes (*e.g.,* polyhedron meshes with at most hundreds of faces).

Voxel-based methods. These methods attempt to approximate the 3D medial axis with a piecewise-constant interpolation based on a uniform sampling (*i.e.,* voxels) [Saha et al. 2016; Sobiecki et al. 2014; Yan et al. 2018]. Many voxel-based methods are guided by a Euclidean distance field [Hesselink and Roerdink 2008; Rumpf and Telea 2002], a derived gradient field [Siddiqi et al. 2002], or more global shape information [Jalba et al. 2015].

Point-cloud-based methods. These methods place sample points around the boundary of the shape and consider either an interior subset of Voronoi diagram of those surface samples or some derivative structures. *Sphere-shrinking-based* (SS) methods [Jalba et al. 2013; Kustra et al. 2016] generate maximal inscribed balls (so-called *medial spheres*) from given point clouds with normals using the *sphere-shrinking* algorithm [Ma et al. 2012]. *Angle-based filtering methods* [Amenta et al. 2001; Brandt and Algazi 1992; Dey and Zhao 2002, 2004] are approaches to filter the Voronoi diagram and select the subset of Voronoi diagram of the boundary samples that meets an angle criterion. *λ-medial axis methods* [Chazal and Lieutier 2005, 2008] discard a medial sphere if its radius is smaller than a given filtering threshold λ. The scale axis transform (SAT) [Miklos et al. 2010] exploits union-of-balls (UoB) and removes spikes while retaining small features by scaling medial spheres. MATFP [Wang et al. 2022] uses the inner Voronoi vertices as initial sphere centers, then updates those spheres' centers and radii as close as possible to the ground truth (GT), through an energy optimization framework.

Table 1. Summary of properties for five selected 3D medial axis approximation methods: PC [Amenta et al. 2001], SAT [Miklos et al. 2010], λMA [Chazal and Lieutier 2005, 2008], SS [Jalba et al. 2013; Kustra et al. 2016], VC [Yan et al. 2018], MATFP [Wang et al. 2022], and our method. The ✓means the property is fulfilled and ◎ represents the property is conditionally satisfied.

Property	PC	λMA	SAT	SS	VC	MATFP	Ours
Homotopy	◎	◎	◎		◎		✓
Medial Features	◎	◎	◎		◎	✓	✓
Thinness		✓			✓	✓	✓
Centeredness	◎	◎	◎	✓	◎	✓	✓
Reconstructibility	◎	◎	◎	◎	◎	✓	✓
Scalability	✓	✓	✓	✓			

We select several representative methods mentioned above, and discuss several important properties of 3D medial axis that are addressed by these methods. The summary is given in Table 1.

Homotopy Equivalence. While most existing methods maintain the homotopy equivalence between the extracted 3D medial axis with the input shape, such equivalence is often conditioned on the sampling rate of points on the shape boundary. For example, both the PC [Amenta et al. 2001] method and SAT [Miklos et al. 2010] method rely on the *r-sample* condition measured by the *local feature size* (LFS) as the 'sufficiently dense' input sample density. The λ-medial axis [Chazal and Lieutier 2005] is homotopy equivalent to the medial axis when λ is less than the weak feature size. Similarly, VC [Yan et al. 2018] shows that for any voxel sizes smaller than $(2\sqrt{3}/3)r$, the union of voxels is homeomorphic to the original shape. Our method, on the contrary, has no assumption on the sampling density of the input shape in order to preserve the topology of the generated medial mesh. Both the SS method [Jalba et al. 2013; Kustra et al. 2016] and the MATFP method [Wang et al. 2022] do not provide any guarantee on the topology of their generated medial axis approximation.

Medial Feature Preservation. Please refer to Sec. 3.1 for a detailed discussion of the medial sphere classification and medial features (*external features* and *internal features*), which are also intensively discussed in seminal works [Kustra et al. 2016; Tagliasacchi et al. 2016; Wang et al. 2022]. To the best of our knowledge, MATFP [Wang et al. 2022] is the first method that attempts to approximate medial axes with non-smooth regions, such as convex sharp edges and corners (so called external features). Our method uses the same strategy for external feature preservation, by placing zero-radius spheres on non-smooth regions, as described in Sec. 4.2. All other methods mentioned in Table. 1 perform poorly in terms of external features preservation. Internal features are hidden structure of the shape, which consists of spheres with more than two tangent points on the surface (*i.e.,* on seams or junctions). The SS methods [Jalba et al. 2013; Kustra et al. 2016] are not able to preserve internal features. This limitation arises because the sphere-shrinking algorithm only provides medial spheres of type T_2 (on sheet), as each sphere is only tangent to two surface points. For medial spheres with more than two tangent points on the surface, such as T_3 (on seam) or T_4 (on junction), these methods cannot generate spheres that lie on the internal features of the 3D medial axis. The MATFP [Wang et al. 2022] and our method are able to detect the ill-posed regions where the internal feature spheres (*e.g.,* $T3$ or $T4$) are lacking, and use *sphere-optimization* algorithm to insert new spheres. All other sampling-based methods (PC, λMA, SAT, VC) in Table. 1 require dense sampling rate (as discussed above) in order to generate approximated internal feature spheres.

Thinness. By definition, the 3D medial axis is a thin structure, so the approximated medial mesh should contain no 3-dimensional cells. This property does not hold for PC [Amenta et al. 2001] and SAT [Miklos et al. 2010], as their results contain large amounts of 'closed pockets'. The per-manifold connectivity strategy proposed by Kustra et al. [2016] does not consider the thinness property of the 3D medial axis. A subset of Voronoi diagram from a sufficiently close and dense noisy sampling, filtered as the λMA [Chazal and Lieutier 2005, 2008] methods, retains the thinness property as no element of this subset is dual to a 0-dimensional vertex. Similarly, the VC [Yan et al. 2018] method achieves this thinness because the *voxel core* is a direct consequence of its duality with Delaunay triangulation. Our method follows the same thinning process as MATFP [Wang et al. 2022] (described in Sec. 5.3) to preserve this property.

Centeredness. The PC [Amenta et al. 2001], λMA [Chazal and Lieutier 2005, 2008], SAT [Miklos et al. 2010], and VC [Yan et al. 2018] methods all require a certain sampling density to ensure that the centeredness of the generated medial spheres is preserved. This requirement arises because these methods use inner Voronoi balls as initial candidate spheres and then select a subset of them. However, Voronoi balls often protrude from the surface as they are naturally circumscribed over the sampling points. The MATFP method [Wang et al. 2022] addresses this by updating and pushing these protruding inner Voronoi balls to be tangential to the surface. Our method inherits this advantage from MATFP and preserves the centeredness property by directly creating medial spheres tangent to at least two points on the surface using the *sphere-shrinking* and *sphere-optimization* algorithms described in Sec. 5.1. The SS methods [Jalba et al. 2013; Kustra et al. 2016] can also maintain the centeredness property as they use the *sphere-shrinking* algorithm [Ma et al. 2012] to generate spheres tangent to exactly two points on the surface.

Reconstructibility. Since only the MATFP method [Wang et al. 2022] and our method take external features (also see the discussion of *Medial Features* above) into account, all other methods shown in Table 1 would reconstruct rounded shapes for 3D CAD models with convex sharp edges and corners. Our method also proposes a local geometric metric that can adaptively refine the 3D medial mesh until its enveloping volume, as the main component of the reconstruction, reaches a desired accuracy.

Scalability. The VC method [Yan et al. 2018] requires fine vwoxel resolution, hence incurring higher computational cost to achieve a comparable geometric accuracy as point-cloud-based methods such as PC [Amenta et al. 2001], λMA [Chazal and Lieutier 2005, 2008], and SAT [Miklos et al. 2010]. Both RPD-based methods in Table 1 (MATFP [Wang et al. 2022] and Ours) requires multiple runs of surface or volumetric RPD computation through the clipping process, which is inevitably more time-consuming than others. Please see Sec. 7 for more discussions.

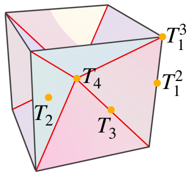
2.2 Restricted Voronoi and Power Diagram in Volumes

Robust and accurate computation of 3D volumetric RPD is not a trivial task. The classical clipping algorithm [Yan et al. 2010] and industrial-quality libraries, such as CGAL [Fabri and Pion 2009] and Geogram [Lévy and Filbois 2015], are still overly time-consuming to be used in tasks that require frequent iterative RPD computations. Fortunately, recent GPU-based approaches has shown strong ability to compute Voronoi and power diagrams on highly parallel architectures, where the geometry of each Voronoi or power cell can be evaluated independently [Basselin et al. 2021; Liu et al. 2020; Ray et al. 2018]. Basselin et al. [2021] proposed a method to directly evaluate the integrals over every restricted power cell without computing the combinatorial structure of power diagram explicitly. Since our work requires explicit structure of each power cell, we build upon the 'Tet-Cell' strategy proposed by Liu et al. [2020]. After discretizing the volume into a tetrahedral mesh [Hu et al. 2020], the intersection of a tetrahedron (*tet* for short) with a cell can be calculated in a parallel manner. This parallel GPU-based volumetric RPD implementation reduces the computational cost significantly, *e.g.,* from 8s to 0.6s for a model with 10k tets and 10k cells.

3 Preliminaries

3.1 Medial Axis and Medial Mesh

Medial Aixs. The medial axis $\mathcal{M}$ of a closed, oriented, and bounded shape $\mathcal{S} \in \mathbb{R}^2 \backslash \mathbb{R}^3$, as shown in Fig. 4 (a), is defined as the locus of centers of spheres that are tangent to at least two boundary points of $\mathcal{S}$, without containing any other boundary points in its interior. The *medial axis transform* (MAT) is formed by the medial axis $\mathcal{M}$ and its radius function. It has been shown that any bounded open set $\mathcal{S}$ is homotopy equivalent to its medial axis $\mathcal{M}$ [Lieutier 2004; Lieutier and Wintraecken 2023; Miklos et al. 2010].



Medial Features. We follow the same notation for classifying medial spheres as MATFP [Wang et al. 2022]. Taking the medial axis of a cube model as an example. The medial spheres of type T_2 are tangent to the boundary of the shape $\partial\mathcal{S}$ at exactly two distinct tangential points, which is the most ordinary case as they lie on 2-manifold **sheets** of $\mathcal{M}$. The intersection of three or more sheets forms a **seam** of $\mathcal{M}$, consisting of a set of T_3 medial spheres tangent to $\partial\mathcal{S}$ at three distinct points. Multiple seams could intersect at a **junction** sphere of $T_{\geq 4}$. The **internal features** of medial axis include those internal spheres located on seams and junctions. For a non-smooth model (*i.e.,* CAD model) that contains sharp edges and corners, either convex or concave, we use a dihedral angle less than $\pi - \phi$ and greater than $\pi + \phi$ to define the *convex sharp edge* and *concave sharp edge* respectively [Abdelkader et al. 2020; Wang et al. 2022]. Here ϕ is a user-defined variable and the user can also mark sharp edges manually. For CAD models, the **external features** of medial axis consist of those convex sharp edges and their associated corners. The medial axis should pass through those surface points that are locally convex and non-smooth. As zero-radius spheres are placed on external features, these medial spheres can be either T_1^2 (on a convex edge) or $T_1^{\geq 3}$ (on a convex corner). Please refer to Wang et al. [2022] for a more complete categorization.

Medial Mesh. Following the convention in Q-MAT [Li et al. 2015], we approximate the MAT of a 3D shape using a non-manifold simplicial complex, called *medial mesh* $\mathcal{M}_s$, consisting of three types of medial primitives: vertices, edges and triangles. Each vertex of $\mathcal{M}_s$ represents a *medial sphere* $\mathbf{m} = (\boldsymbol{\theta}, r)$, with center $\boldsymbol{\theta} \in \mathbb{R}^3$ and radius $r \in \mathbb{R}$. The union of enveloping volume of all the medial primitives can be used to reconstruct the surface. The enveloping volume of an edge of $\mathcal{M}_s$ is called a *medial cone*. As shown in Fig. 4 (b), medial cone $\mathbf{e}_{ij}$ is a linear interpolation of two spheres $\mathbf{e}_{ij} = t\mathbf{m}_i + (1-t)\mathbf{m}_j$, $t \in [0, 1]$. Similarly, the enveloping volume of a triangle $\mathbf{f}_{ijk}$ of $\mathcal{M}_s$ is called *medial slab*, shown in Fig. 4 (c).

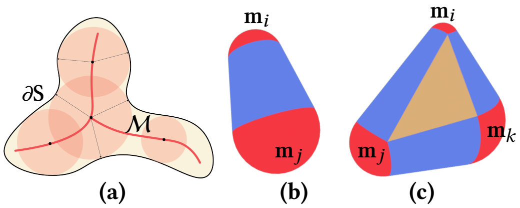


Fig. 4. (a) The medial axis $\mathcal{M}$ of a shape $\mathcal{S}$ in $\mathbb{R}^2$. (b) The medial cone as a linear interpolation of two medial spheres $\mathbf{m}_i$ and $\mathbf{m}_j$. (c) The medial slab as a linear interpolation of three spheres $\mathbf{m}_i$, $\mathbf{m}_j$, and $\mathbf{m}_k$.

3.2 Volumetric Restricted Power Diagram and its Dual

We follow a similar RPD-based strategy as MATFP [Wang et al. 2022] to construct the medial mesh $\mathcal{M}_s$ as the approximation of the 3D MAT $\mathcal{M}$. Here, $\mathcal{M}_s$ is generated from the *restricted regular triangulation* (RRT), which is the dual of *restricted power diagram* (RPD). The main difference is that MATFP computes the medial mesh $\mathcal{M}_s$ by selecting a subset of simplices in *regular triangulation* (RT) whose dual elements in *power diagram* (PD) has non-empty intersections with the input shape $\mathcal{S}$, as MATFP relies on surface-RPD only. In this paper, we construct $\mathcal{M}_s$ directly from the dual of volumetric RPD (see Sec. 5.2 for technical details). In this section, we discuss the formal definitions of RPD and its dual.

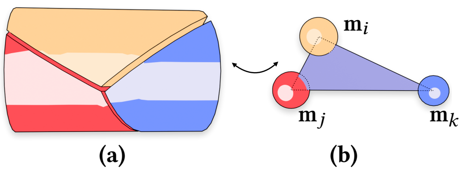


Fig. 5. The duality between the volumetric RPD (a) of three medial spheres and the generated medial mesh (b)

As a generalization of the Voronoi diagram, the power diagram (PD) [Aurenhammer 1987] is generated by a set of weighted points, and coincides with the Voronoi diagram in the case of equal weights. The power diagram of a set of medial spheres $\Gamma = \{\mathbf{m}_i\}_{i=1}^n$ is a partition of the domain $\Omega \subset \mathbb{R}^d$ into a set of *power cells*. Each power cell Ω_i^{pow} consists of the points $\mathbf{x} \in \Omega$ closest to a particular sphere $\mathbf{m}_i$ as:

$$\Omega_i^{pow} : \{\mathbf{x} \in \Omega | d_{pow}(\mathbf{x}, \mathbf{m}_i) \leq d_{pow}(\mathbf{x}, \mathbf{m}_j), j \neq i\}, \tag{1}$$

where $d_{pow}(\mathbf{x}, \mathbf{m}_i) = ||\mathbf{x} - \boldsymbol{\theta}_i||^2 - r_i^2$ is the *power distance* between the point $\mathbf{x}$ and the medial sphere $\mathbf{m}_i = (\boldsymbol{\theta}_i, r_i)$.

A power diagram restricted within a bounded shape $\mathcal{S}$ is called the *restricted power diagram* (RPD) as:

$$\mathcal{Q}(\Gamma) = \bigcup_{\mathbf{m}_i \in \Gamma} \omega_{\mathbf{c}}(\mathbf{m}_i), \tag{2}$$

where each sub-domain $\omega_{\mathbf{c}}(\mathbf{m}_i)$ is the restriction of the power cell Ω_i^{pow} of the medial sphere $\mathbf{m}_i$ within $\mathcal{S}$:

$$\omega_{\mathbf{c}}(\mathbf{m}_i) = \Omega_i^{pow} \cap \mathcal{S}. \tag{3}$$

Fig. 5 (a) shows the RPD of three spheres, where the input volume $\mathcal{S}$ is divided into three sub-domains clipped by three radical hyperplanes. We show its duality in Fig. 5 (b). The duality between the volumetric RPD and the medial mesh $\mathcal{M}_s$ can be summarized as follows:

- Each sub-domain $\omega_{\mathbf{c}}(\mathbf{m}_i)$ is called a *restricted power cell* (RPC) of medial sphere $\mathbf{m}_i$, which is dual to a vertex on $\mathcal{M}_s$.
- The face shared by two adjacent RPCs is called a *restricted power face* (RPF), $\omega_{\mathbf{f}}(\mathbf{m}_i, \mathbf{m}_j) = \omega_{\mathbf{c}}(\mathbf{m}_i) \cap \omega_{\mathbf{c}}(\mathbf{m}_j)$, which is dual to an edge $\mathbf{e}_{ij}$ on $\mathcal{M}_s$.
- The edge shared by three RPCs is called a *restricted power edge* (RPE), $\omega_{\mathbf{e}}(\mathbf{m}_i, \mathbf{m}_j, \mathbf{m}_k) = \omega_{\mathbf{c}}(\mathbf{m}_i) \cap \omega_{\mathbf{c}}(\mathbf{m}_j) \cap \omega_{\mathbf{c}}(\mathbf{m}_k)$, which is dual to a triangle face $\mathbf{f}_{ijk}$ on $\mathcal{M}_s$.
- A vertex shared by four RPCs $\omega_{\mathbf{v}}(\mathbf{m}_i, \mathbf{m}_j, \mathbf{m}_k, \mathbf{m}_s)$ is called a *restricted powper vertex* (RPV), which is dual to a tetrahedron on $\mathcal{M}_s$, if it exists. Note that all tetrahedra in the medial mesh will be pruned using a geometry-guided thinning algorithm proposed by MATFP [Wang et al. 2022] (see 5.3 for details).

3.3 Nerve Theorem and Homotopy Equivalence

In this section, we present our assumptions on the input model, as well as the theoretical foundation of our RPD-based topology-preserving strategy, grounded on the *Nerve Theorem*.

Assumption. *The input for our method is a manifold tetrahedral mesh with a single connected component, no self-intersection, and* ***no cavity****. It should be noted that this "no cavity" assumption holds for all CAD models and organic models that we found online.*

Definition 1 (Nerve). Let X be a topological space, and $\mathcal{U} = \{U_i\}_{i\in I}$ be any covering of X where I is a set of indices, so $X = \bigcup_{i\in I} U_i$. The **nerve** of $\mathcal{U}$, denoted $\mathcal{N}(\mathcal{U})$, is defined as a set of finite subsets of the index set I. It contains all finite subsets $J \subseteq I$ such that the intersection of the U_i whose sub-indices are in J is non-empty [Carlsson and Vejdemo-Johansson 2021]:

$$\mathcal{N}(\mathcal{U}) = \left\{ J \subseteq I : \bigcap_{j\in J} U_j \neq \emptyset,\ J \text{ finite set} \right\}. \tag{4}$$

Theorem 1 (Nerve Theorem). If for each finite subset $J \subseteq I$, the set $\bigcap_{j\in J} U_j$ is either empty or contractible, then $\mathcal{N}(\mathcal{U})$ is homotopy-equivalent to $X = \bigcup \mathcal{U}$, that is, $\mathcal{U}$ is a good cover [Leray 1950].

In the context of this paper, the input shape $\mathcal{S}$ defines the topological space X, and RPD $\mathcal{Q}(\Gamma)$ is a covering $\mathcal{U}$ of X given the set of n medial spheres $\Gamma = \{\mathbf{m}_i\}_{i=1}^n$. That is, the RPD $\mathcal{Q}(\Gamma)$ is a covering of the shape $\mathcal{S}$. The medial mesh $\mathcal{M}_s$, which is computed from the dual of the RPD $\mathcal{Q}(\Gamma)$, can be considered as the nerve of the RPD $\mathcal{Q}(\Gamma)$. This is because every simplex of the mesh $\mathcal{M}_s$, *i.e.,* vertex, edge, triangle, tetrahedron, is exactly dual to each RPC, intersecting face (RPF) of two adjacent RPCs, intersecting edge (RPE) of three adjacent RPCs, and intersecting vertex (RPV) of four adjacent RPCs, respectively.

According to the Nerve Theorem, if every restricted elements (*i.e.,* RPCs, RPFs, RPEs, RPVs) are contractible, then the medial mesh $\mathcal{M}_s$ is homotopy-equivalent to the input shape $\mathcal{S}$.

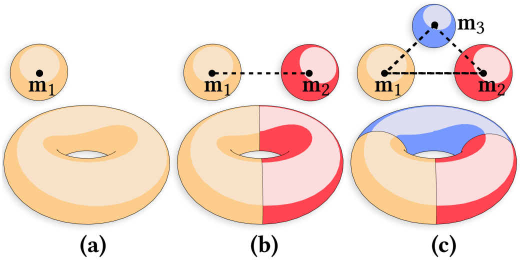


Fig. 6. The duality between the medial mesh (top) and volumetric RPD (bottom) for a 3D torus shape, where the homotopy equivalence does not hold for case (a) and case (b). An additional medial sphere $\mathbf{m}_3$ needs to be inserted in order to maintain the homotopy equivalence, as shown in (c). The dual edges are depicted as black dotted lines.

If we use ω to represent any type of restricted element, including RPC $\omega_{\mathbf{c}}$, RPF $\omega_{\mathbf{f}}$, RPE $\omega_{\mathbf{e}}$, or RPV $\omega_{\mathbf{v}}$, the Euler-Poincaré formula shows:

$$\mathcal{X}(\omega) = \beta_0(\omega) - \beta_1(\omega) + \beta_2(\omega), \tag{5}$$

where $\mathcal{X}$ is the *Euler characteristic* and β_k is the kth Betti number representing the number of k-dimentional holes in ω. Since our input shape $\mathcal{S}$ contains no cavity, *i.e.,* $\beta_2(\mathcal{S}) = 0$, all restricted elements also contains no cavity $\beta_2(\omega) = 0$. The formula can be rewritten as:

$$\mathcal{X}(\omega) = \beta_0(\omega) - \beta_1(\omega). \tag{6}$$

To test whether ω is contractible, we need to check whether ω is homotopy-equivalent to a one-point space. That is to say, the contractibility of ω could be tested by two topological indicators: (1) whether its number of *connected components* $\beta_0(\omega) = 1$; and if so, (2) whether its *Euler characteristic* $\mathcal{X}(\omega) = 1$, which implies that its number of 'circular' holes $\beta_1(\omega) = 0$.

Let us take the shape torus in Fig 6 as an example. When we use a single medial sphere $\mathbf{m}_1$ to represent the shape, its RPC $\omega_{\mathbf{c}}(\mathbf{m}_1)$ covers the whole torus but does not have the same topology as the medial sphere, as $\beta_0 = 1$ but $\mathcal{X} = 0$. The RPC $\omega_{\mathbf{c}}(\mathbf{m}_1)$ in Fig. 6 (b) has $\beta_0 = 1$ and $\mathcal{X} = 1$ as expected, but its associated RPF $\omega_{\mathbf{f}}(\mathbf{m}_1, \mathbf{m}_2)$ has $\beta_0 = 2$. In Fig. 6 (c) all restricted elements have $\beta_0 = 1$ and $\mathcal{X} = 1$ for all three medial spheres $\mathbf{m}_1$, $\mathbf{m}_2$ and $\mathbf{m}_3$. Note that only RPCs and RPFs exist in this example (c) without any RPEs, hence only vertices and edges exist in the dual medial mesh. This contractibility testing process inspires our topology preservation algorithm proposed in Sec. 4.1.

4 Method

Given a manifold tetrahedral mesh (satisfying the Assumption in Sec. 3.3) with its surface features (sharp edges and corners) pre-detected, our pipeline starts with an initial medial mesh of a small number (*e.g.,* 50) of randomly placed medial spheres computed with the *sphere-shrinking* (Sec 5.1) algorithm [Ma et al. 2012]. The medial mesh is then refined iteratively through the following three steps. First, we preserve the homotopy equivalence of the generated medial mesh w.r.t. the input 3D shape by examining two topological indicators (β_0 and $\mathcal{X}$) of all restricted elements for each medial sphere (Sec 4.1), as shown in Fig 7 (b). Then, the *medial features* are preserved by assessing if the RPCs of two adjacent medial spheres covers the same surface regions (Sec. 4.2), as shown in Fig 7 (c). Finally, we ensure the geometric convergence by checking if the distance error between the shape boundary ∂S and the enveloping volume of medial mesh $\mathcal{M}_s$ is smaller than a user-defined threshold δ^ϵ (Sec. 4.3), as shown in Fig 7 (d). Each examination step is performed locally and new medial spheres are inserted for each preservation using two sphere generation strategies described in Sec. 5.1. For each iteration, we only update the RPD partially for cells that are related to the newly added medial spheres, rather than re-computing the whole RPD for all medial spheres. Please refer to both Sec. 5 and Supplementary Document for more implementation details.

4.1 Topology Preservation

As discussed in Sec. 3.3, the homotopy equivalence between the input shape $\mathcal{S}$ and the generated medial mesh $\mathcal{M}_s$ is enforced by examining the following two topological indicators for each individual RPC and its associated RPFs and RPEs:

- The number of *connected components* (CC number) β_0 tells us the maximal subset of a topological space;

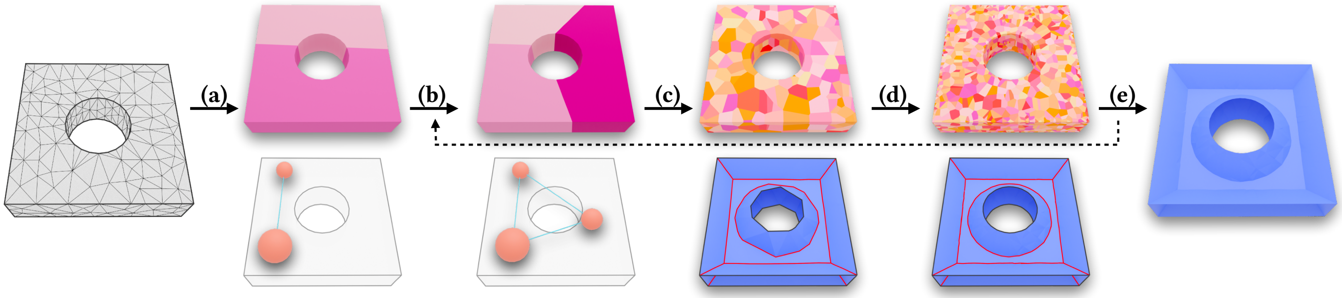


Fig. 7. The overview of our computational pipeline. Given a tetrahedral mesh with surface features (sharp edges and corners) pre-detected as input, our method starts with an initial medial mesh of a small number of spheres (*i.e.,* 2 spheres) using the *sphere-shrinking* algorithm [Ma et al. 2012], shown in (a). Then the homotopy equivalence of the generated medial mesh w.r.t. the input shape is preserved by examining the topological equivalence of individual RPCs (Sec 4.1), shown in (b). We preserve the *medial features* using the same method as MATFP [Wang et al. 2022] (Sec 4.2), shown in (c). Finally we preserve the geometric convergence based on a user-defined error threshold δ^{ϵ} (Sec 4.3), shown in (d). We repeat this process until both topological preservation and geometric convergence are satisfied, and output the final result of generated medial mesh, shown in (e).

- The *Euler characteristic* $\mathcal{X}$ describes a topological space's structure regardless of the way it is bent. It can be calculated using following formula: $Euler = V - E + F - C$, where V is the number of vertices, E is the number of edges, F is the number of faces, and C is the number of volumetric cells.

To ensure homotopy equivalence, we are expecting each restricted element (*i.e.,* RPC, RPF, RPE) to have $CC\ \beta_0 = 1$ and $Euler\ \mathcal{X} = 1$. For each medial sphere $\mathbf{m}_i$, we perform a localized topological check for its corresponding restricted elements, and apply a straightforward refinement by adding new medial spheres if either their $CC \neq 1$ or $Euler \neq 1$. The details of our preservation strategy for those two topological indicators are given in Sec. 4.1.1 and Sec. 4.1.2.

4.1.1 CC Number β_0. For a medial sphere $\mathbf{m}_i$ and its corresponding restricted elements $\omega_{\mathbf{c}}(\mathbf{m}_i)$, $\omega_{\mathbf{f}}(\mathbf{m}_i, \mathbf{m}_j)$ and $\omega_{\mathbf{e}}(\mathbf{m}_i, \mathbf{m}_j, \mathbf{m}_k)$, where $\mathbf{m}_j$ and $\mathbf{m}_k$ are the neighboring medial spheres of $\mathbf{m}_i$, we can trace their CC number β_0 using a simple traversal algorithm effortlessly. If $\beta_0 > 1$, we add a new medial sphere to the connected component other than the one of the current medial sphere $\mathbf{m}_i$. In Fig. 6 (b), the RPF $\omega_{\mathbf{f}}(\mathbf{m}_1, \mathbf{m}_2)$ has $\beta_0(\omega_{\mathbf{f}}) = 2$, while in Fig. 8 (a), the RPC $\omega_{\mathbf{c}}(\mathbf{m}_2)$ has $\beta_0(\omega_{\mathbf{c}}) = 2$. To fix these issues, we randomly choose a surface point on one of the CC of the RPF, or choose a surface point on the other CC of the RPC (other than where the medial sphere resides), and use it as the *pin point* for the *sphere-shrinking* algorithm (see Sec. 5.1) to add a new medial sphere $\mathbf{m}_3$, as shown in Fig. 6 (c) and Fig 8 (b).

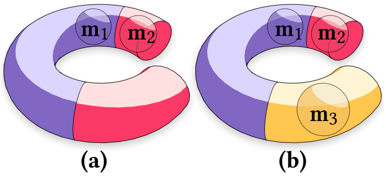


Fig. 8. Illustration of solving the CC number inequivalence described in Sec 4.1.1. The RPC of medial sphere $\mathbf{m}_2$ in (a) contains two connected components, which is not contractible. Hence we add a new sphere $\mathbf{m}_3$ to the other connected component of the RPC of $\mathbf{m}_2$ for maintaining the contractibility of each RPC.

4.1.2 Euler Characteristic $\mathcal{X}$. One can compute the Euler characteristic $\mathcal{X}$ of each restricted element (RPC, RPF and RPE) after obtaining the explicit RPD representation through the clipping process (Sec. 5.2). However, current GPU-based implementations either directly evaluate integrals over every cells without computing the combinatorial data structure of power diagram [Basselin et al. 2021], or store the dual form of cells using a simple triangle mesh [Liu et al. 2020; Ray et al. 2018], but requires costly post-processing steps to access the exact combinatorial data structure of the cells. This is due to the fact that these methods are well-designed for the high parallelism of GPU, and are optimized for applications that only requires integrals over the cells, *e.g.,* fluid dynamics simulations. Therefore, in this paper we propose a *Fractional Euler Characteristic* strategy to collect Euler characteristics on-the-fly, by taking full advantage of the existing GPU-based volumetric RPD computation pipeline.

Similar to the *Tet-Cell* strategy proposed by Liu et al. [2020] (see Sec. 5.2), our pipeline takes the tetrahedral mesh as input, where the Euler characteristic is inherited during the clipping process. For each tetrahedron of the related medial sphere $\mathbf{m}_i$, we clip the RPC $\omega_{\mathbf{c}}(\mathbf{m}_i)$ as the intersection of half-spaces bounded by the bisectors of $\mathbf{m}_i$ and its power neighbors [Ray et al. 2018]. Implementation details regarding finding tet-sphere relations and sphere neighbors are provided in the Supplementary Document.

Fractional Euler Characteristics. Fig. 9 illustrates our proposed *fractional Euler characteristics* in a 2D setting, demonstrating how these fractional numbers are inherited throughout the clipping process during RPD computation. In this example, the target Euler characteristic is 1 for the 2D shape, *i.e.,* $\mathcal{X} = 4$ (vertices) - 5 (edges) + 2 (faces) - 0 (cells) = 1. It contains 2 triangles **A** and **B** in Fig. 9 (i), and the fractional Euler characteristic of each element is shown in Fig. 9 (ii). Note that both vertex **b** and **d** are shared between **A** and **B**, so their fractional Euler characteristic within each triangle is only $\frac{1}{2}$. The same rationale applies to the fractional Euler characteristic $\frac{1}{2}$ for the edge $(\mathbf{b}, \mathbf{d})$ within each triangle. As an initialization step for our GPU-based tet-cell clipping, whenever a tetrahedral mesh is inputted to the GPU, these fractional Euler characteristics

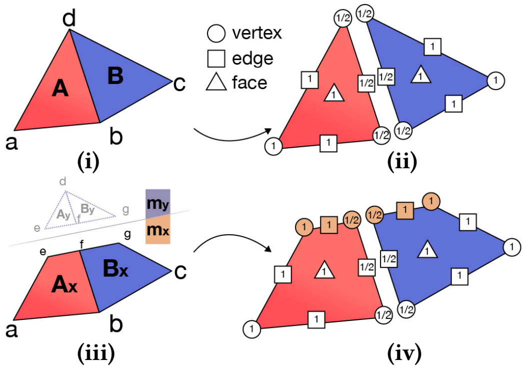


Fig. 9. 2D illustration of our GPU-based Euler characteristic computation on-the-fly during the clipping process of RPD. The number shown in each vertex, edge, and face represents its *Fractional Euler Characteristic*, as described in Sec 4.1.2.

are included alongside the mesh, derived from the combinatorial structure of the tetrahedral mesh.

Inheritance of Fractional Euler Characteristics during RPD Clipping. During the parallel execution of clipping process [Ray et al. 2018], the half-space $\Pi^+(\mathbf{m}_x, \mathbf{m}_y)$ of two medial spheres $\mathbf{m}_x$ and $\mathbf{m}_y$ cuts each triangle into two parts, thus new vertices and edges emerge. The fractional Euler characteristics of new vertices are inherited from their pre-clipped edges, and those of the new edges are inherited from its pre-clipped faces. For the example of Fig. 9, the new vertex $\mathbf{f}$ inherits the fractional Euler characteristic $\frac{1}{2}$ from its pre-clipped edge $(\mathbf{b}, \mathbf{d})$, and the new edge $(\mathbf{e}, \mathbf{f})$ inherits the fractional Euler characteristic 1 from its pre-clipped face $\mathbf{A}$. Note that during the clipping process, although the new generated vertices, edges, and faces could be shared between different cells, we no longer divide their fractional Euler characteristics. For example, in Fig. 9 (iii), although the new edge $(\mathbf{e}, \mathbf{f})$ is shared between two cells $\mathbf{A}_x$ and $\mathbf{A}_y$, we only set its fractional Euler characteristic as 1 for cell $\mathbf{A}_x$, and 1 for cell $\mathbf{A}_y$. This is because the use of fractional Euler characteristics is to facilitate the easy counting of Euler characteristic of each restricted power element associated with a particular sphere. For the example of Fig. 9 (iv), we are only interested in the Euler characteristics for the restricted elements related to sphere $\mathbf{m}_\mathbf{x}$, thus the fractional Euler characteristics of those newly generated vertices and edges will not be shared with the other spheres.

Through such inheritance, the fractional Euler characteristic of all restricted elements (RPCs, RPFs, RPEs) for sphere $\mathbf{m}_x$ can be obtained on-the-fly during the parallel clipping process. In Fig. 9 (iv), the final RPFs of sphere $\mathbf{m}_x$ consists of two convex hulls $\mathbf{A}_x$ and $\mathbf{B}_x$, and its Euler characteristic can be computed by $\mathcal{X}(\mathbf{A}_x) + \mathcal{X}(\mathbf{B}_x) = 1$, where:

$$\begin{aligned} \mathcal{X}(\mathbf{A}_x) &= (1 + 1 + \frac{1}{2} + \frac{1}{2}) - (1 + 1 + 1 + \frac{1}{2}) + (1) = \frac{1}{2}, \\ \mathcal{X}(\mathbf{B}_x) &= (1 + 1 + \frac{1}{2} + \frac{1}{2}) - (1 + 1 + 1 + \frac{1}{2}) + (1) = \frac{1}{2}. \end{aligned} \tag{7}$$

For each medial sphere, we collect the fractional Euler characteristics for all of its restricted elements (RPCs, RPFs, RPEs) during the runtime of clipping process, then check if the target Euler characteristic for each element is expected or not, *e.g.*, in Fig. 6 (a), the sphere $\mathbf{m}_1$ has its RPC $\omega_\mathbf{c}(\mathbf{m}_1)$ with $\mathcal{X}(\omega_\mathbf{c}(\mathbf{m}_1)) = 0$. To fix this issue, we search all surface triangles among the RPC of sphere $\mathbf{m}_1$ and target the furthest one as the *pin point* and uses the *sphere-shrinking* algorithm (see Sec. 3) to add a new sphere $\mathbf{m}_2$. After that, the RPF between $\mathbf{m}_1$ and $\mathbf{m}_2$ has CC number $\beta_0 = 2$, which will trigger the insertion of a new sphere $\mathbf{m}_3$, with the same process described in Sec. 4.1.1.

4.2 Medial Feature Preservation

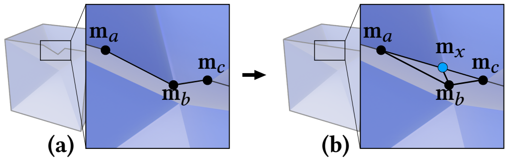


Fig. 10. Illustration of external feature preservation. (a) A non-feature medial sphere $\mathbf{m}_b$ destroys the connectivity of two zero-radius feature spheres $\mathbf{m}_a\mathbf{m}_c$ in the medial mesh. (b) Adding a new zero-radius medial sphere $\mathbf{m}_x$ could preserve the connectivity of external feature.

External features, such as convex sharp edges and corners where more than two convex sharp edges coincide, are commonly seen in CAD models and are pre-detected as input for our algorithm. We adopt the similar feature preservation strategies as MATFP [Wang et al. 2022], as shown in Fig 10. Since the generated medial mesh should pass through points where the surface is locally convex and non-smooth, we place zero-radius T_1^2 medial spheres on convex edges and a single zero-radius T_1^u $(u \geq 3)$ medial sphere at each corner. Unlike MATFP which starts with a dense number of initial spheres, our method begins with a smaller set of non-feature medial spheres. Hence, instead of removing redundant non-feature spheres that 'invade' the RPCs of two neighboring feature spheres on a sharp edge, we examine the individual RPC and add new zero-radius feature spheres iteratively if any sharp edge belongs to the cell of a non-feature sphere. As a result, all convex sharp edges reside in cells corresponding to some zero-radius feature spheres, which preserves the convex sharp edges in the generated medial mesh as a dual of volumetric RPD. For *corners*, we adopt the same corner preservation strategy as MATFP [2022], where we approximate the medial mesh structure within a small, selected region around the corner. This scheme operates by recursively tracing the medial axis sheets, beginning from those adjacent to convex sharp edges, until their intersecting seams are identified. On a *concave edge*, the medial spheres tangential to this feature are not zero-radius. To achieve a smooth transition on the medial axis, similar to MATFP [2022], we densely sample medial spheres using the *sphere-shrinking* algorithm [2012] (see Sec. 5.1), with tangent surface points selected on the concave edge acting as the 'pin'. For more details on the operations used to preserve *corners* and *concave features*, please refer to MATFP [Wang et al. 2022].

Internal features are preserved by inserting new internal feature spheres after detecting the deficiency, similar to MATFP [Wang et al. 2022]. We show an illustration in Fig. 11. Here we maintain a queue

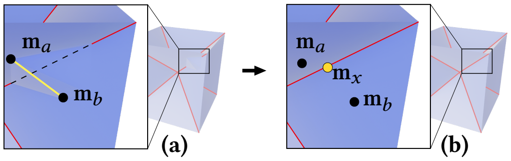


Fig. 11. Illustration of internal feature preservation. (a) An ill-posed connection (yellow line) of two T_2 spheres $\mathbf{m}_a\mathbf{m}_b$ on two different medial sheets. (b) The internal medial feature (red line) is preserved after inserting a T_3 sphere $\mathbf{m}_x$.

of all medial edges in the medial mesh, and check whether two connected medial spheres of each edge belong to the same medial sheets as they touch the same surface regions using the surface part of their RPCs.

4.3 Geometric Preservation

The medial mesh generated through the above steps is topologically correct and captures medial features. However, its reconstruction may not geometrically converge to the input shape, since we adaptively insert spheres from a low number. We propose an error-bounding strategy so that the geometric error from the input shape to the enveloping volume of medial mesh is bounded by a user-controlled threshold δ^{ϵ}.

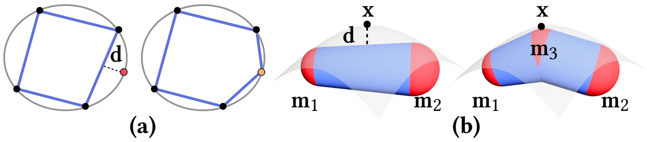


Fig. 12. Illustration of geometric preservation. (a) We compute the distance d from the sampled points (in red) on external features to the nearest feature enveloping cone and add zero-radius medial sphere (in yellow) if the distance is too large. For other cases (b), we randomly sample surface points and compute their distance to the nearest enveloping primitive (*e.g.,* medial cone) of the medial mesh, and insert a new T_2 medial sphere (*e.g.,* $\mathbf{m}_3$) if the distance is too large.

For preserving the geometry of external features, we sample points directly on those pre-detected features, such as Fig 12 (a) in red, and compute their distance d to the nearest enveloping cone. When the distance is larger than our threshold, which often happens for curved feature lines, we insert a new zero-radius feature sphere (Fig 12 (a) in yellow).

For other cases, we use the distance from any surface sample $\mathbf{x}$ to its closest enveloping element $\mathbf{s}_t$ as the metric. Each $\mathbf{s}_t$ can be a medial sphere, a medial cone, or a medial slab (see Sec. 3.2). For each surface sample, we traverse all possible enveloping elements defined by the medial mesh and compute its distance to the closest enveloping element on GPU. Fig 12 (b) shows an example of the distance from the surface sample $\mathbf{x}$ to a medial cone defined by two medial spheres $\mathbf{m}_1$ and $\mathbf{m}_2$. If the ratio of the distance d over the diagonal of the bounding box is larger than a user-defined threshold δ^{ϵ}, we insert a new non-feature sphere with the surface sample as the pin point using the sphere-shrinking algorithm [Ma et al. 2012] (see Sec. 5.1), shown in Fig. 12 (b) as $\mathbf{m}_3$. As more non-feature spheres are inserted, the enveloping volume of the medial mesh is converging to the input shape geometrically. We show an ablation study on the impact of the value of δ^{ϵ} in Sec 6.4.

5 Technical Details

5.1 Generation of Medial Spheres

We use two strategies for computing medial spheres that are tangent to at least two surface points. These two strategies enable us to maintain the centeredness property of our generated medial mesh.

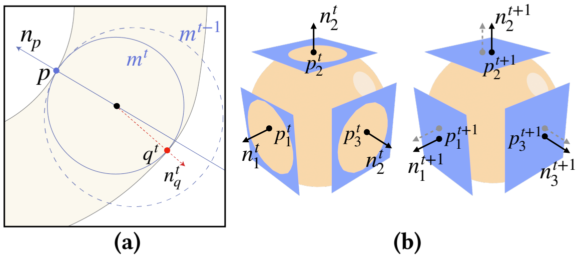


Fig. 13. Illustration of two medial sphere generation algorithms. (a) Given 'pin' point $\mathbf{p}$ on the model boundary $\partial\mathrm{S}$, the sphere-shrinking algorithm [Ma et al. 2012] decreases the sphere radius iteratively until the sphere $\mathbf{m}^t$ is a maximal empty ball (that is the interior of $\mathbf{m}^t$ contains no point of $\partial\mathrm{S}$), while the another tangent point touches $\mathbf{q}^t$. (b) The sphere-optimization algorithm [Wang et al. 2022] iteratively updates medial spheres in two alternating steps. The first step locks the aggregated tangent pairs $\{(\mathbf{p}_k, \mathbf{n}_k)\}_{k=1}^{N}$ and updates the sphere center and radius. The second step updates each tangent pair by fixing the previously updated medial sphere.

- *Sphere-shrinking* algorithm [Ma et al. 2012] for computing T_2 medial spheres that are on **sheets** of the medial structure, and tangent to two different places on the surface, as shown in Fig. 13 (a);
- *Sphere-optimization* algorithm [Wang et al. 2022] for computing T_N ($N \geq 3$) medial spheres that are on **seams** or **junctions** of the medial structure, and tangent to at least three different places on the surface, as shown in Fig. 13 (b).

5.2 Computation of Restricted Power Cells

Similar to a Voronoi cell, a power cell is a convex polyhedron that is the intersection of half-spaces. The only difference is that these half-spaces are not generated by bisectors, but by radical hyperplanes.

We use the data structure introduced by Ray et al. [2018] to represent the convex polyhedrons, which is highly compact and well-suited for GPU implementation. Each half-space of the polyhedron is represented by a float4 storing 4 coefficients (a, b, c, d) of the plane equation $ax + by + cz + d > 0$. Each vertex of the polyhedron is the intersection of three half-spaces, therefore, is represented by a triplet of integers in a clockwise order, storing indices of three adjacent half-spaces.

To restrict the power cells in an input domain, we extend the 'Tet-Cell' strategy proposed by Liu et al. [2020]. Given an input shape represented as a tetrahedral mesh [Hu et al. 2020], we compute the intersection between the tetrahedral mesh and power cells. An illustration of 'Tet-Cell' strategy in 2D is shown in Fig. 14. Each step discussed in Sec. 4 will adaptively insert new medial spheres

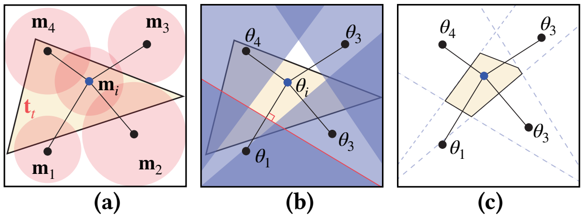


Fig. 14. Illustration of 'Tet-Cell' strategy in 2D using triangles. Here we show the medial sphere as $\mathbf{m}_i = (\boldsymbol{\theta}_i, r_i)$, where $\boldsymbol{\theta}_i$ is the center of $\mathbf{m}_i$ and r_i is its radius. (a) For each triangle-sphere pair $(\mathbf{t}_t, \mathbf{m}_i)$, we traverse all neighboring spheres of $\mathbf{m}_i$. The searching of sphere neighbors and tet-sphere relations are further discussed in Supplementary Document. (b) Each pair of two spheres $(\mathbf{m}_i, \mathbf{m}_k)$ defines a half-space that may potentially 'clip' the triangle $\mathbf{t}_t$. (c) The part of the power cell inside the triangle (*i.e.,* tet in 3D) $\mathbf{t}_t$ for sphere $\mathbf{m}_i$ can be generated after the clipping process.

when topological or geometrical deficiency is detected. We thus only update the RPD partially with cells relating to the newly added medial spheres, instead of recomputing the whole RPD. Please refer to our implementation details in the Supplementary Document for further discussions.

5.3 Thinning of Medial Mesh

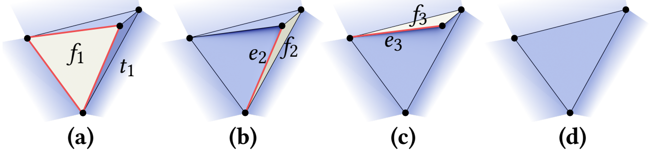


Fig. 15. The thinning process for one flat tetrahedron t_1 in a 3D medial mesh. The simple tet-face pair (t_1, f_1) in (a) is selected to prune with the least importance. Then we continue pruning those face-edge simple pairs, *i.e.,* (f_2, e_2) in (b) and (f_3, e_3) in (c). The pruned result is shown in (d).

The medial mesh $\mathcal{M}_s$ constructed as the dual of the RPD inevitably contains some fat but solid tetrahedrons. We apply the *geometry-guided thinning algorithm* [Wang et al. 2022] that prunes simple pairs of simplices in the medial mesh. A *simple pair* (x, y) [Liu et al. 2010] is a pair of simplices such that y is on the boundary of x, and there is no other cell in the complex with y on its boundary. Ju et al. [2007] have shown that the removal of simple pairs will not impact the topology of the complex. We first rank all possible tet-face pairs by the importance factor α_{ijk} of a given medial triangle f_{ijk} in $\mathcal{M}_s$ in ascending order. The α_{ijk} is defined as the ratio of the length of *restricted power edge* (RPE) $\omega_{\mathbf{e}}(\mathbf{m}_i, \mathbf{m}_j, \mathbf{m}_k)$ over the average diameter of three medial spheres $\mathbf{m}_i$, $\mathbf{m}_j$ and $\mathbf{m}_k$. Note that the RPE $\omega_{\mathbf{e}}$ is dual to the medial triangle f_{ijk} on $\mathcal{M}_s$, discussed in Sec. 3.2. Then we remove tet-face pairs iteratively with the least importance until all tetrahedra are pruned. We continue pruning face-edge simple pairs that belong to the original tetrahedra until a target importance factor σ is reached as a stop sign. To avoid over-prunning for models whose medial mesh boundaries are not external features, we set $\sigma = 0.3$ for CAD models and $\sigma = 0.1$ for organic models in our experiments. Fig 15 shows an illustration of the thinning process with one tetrahedron.

6 Experiments

In this section, we show quantitative and qualitative evaluations of the proposed method. We implemented our algorithm in C++ and CUDA, using Geogram [Lévy and Filbois 2015] for linear algebra routines. We ran our experiments on a computer with a 3.60GHz Intel(R) Core(TM) i7-9700K CPU, NVIDIA GeForce RTX 2080 Ti GPU, and 32 GB memory. We ran our method on the first 100 models in the ABC dataset [Koch et al. 2019] under the *10k/test* folder using 2048 as the number of mesh vertices, same as MATFP [Wang et al. 2022]. We also test our method on 14 organic models with various topology. All model sizes are normalized to the $[0, 1000]^3$ range. We use fTetwild [Hu et al. 2020] for computing the initial tetrahedral mesh from triangle mesh with parameters $l = 0.5$. Our code ↗ is available at our project website ↗ , and the generated medial mesh can be viewed using the tool blender-mat-addon ↗ [Song and Wang 2023].

Evaluation Metrics. We use the *Euler characteristics* $\mathcal{X}$ as the topology measures for the generated medial mesh, and show the ground truth Euler characteristic of the input shape as 'GT $\mathcal{X}$'. We compute the Euler characteristic using $\mathcal{X} = V - E + F - C$, where V is the number of vertices, E is the number of edges, F is the number of faces, and C is the number of volumetric cells. In our experiment, we consider only tetrahedrons as the 3D-complex cells for C. We use the two-sided Hausdorff distance error ϵ to measure the the surface reconstruction accuracy using the generated medial meshes. ϵ^1 is the one-sided Hausdorff distance from the original surface to the surface reconstructed from MAT, and ϵ^2 is the distance in reverse side. All Hausdorff distances are evaluated as percentages of the distance over the diagonal lengths of the models' bounding box. The ϵ^{max} is the maximum of ϵ^1 and ϵ^2. We show #*s* as the number of medial spheres for the medial meshes generated from each method, and use #*t* as the number of tets in the input tetrahedral mesh.

6.1 Comparison with MATFP Method

We compare our method with MATFP [Wang et al. 2022] regarding the topology preservation. To our best knowledge, MATFP is the state-of-the-art method for computing medial axis of CAD models, which preserves both external and internal features. However, it also has a clear drawback with no guarantee of topology preservation for the generated medial mesh w.r.t. the input model. The key advantage of our method is its ability in preserving the topology, while maintaining the ability to capture external and internal features. We show the Euler characteristic $\mathcal{X}$ in Table 2 for those models that the output of MATFP deviates from the GT $\mathcal{X}$. Our method, on the contrary, can preserve the topology while keeping a competitive reconstruction quality.

Beside preserving topology, our method also generates almost 10 times less number of medial spheres #*s* than MATFP in the final medial mesh, shown in Table 2 and Fig 16. MATFP favors preserving external or internal features by adding as many feature spheres as possible during each step. This will inevitable result in a large number of medial spheres, mostly redundant, in the generated medial mesh. Owing to the proposed adaptive refinement strategy, our method can add only few (even single) number of new medial spheres at each iteration and update the RPD partially using GPU.

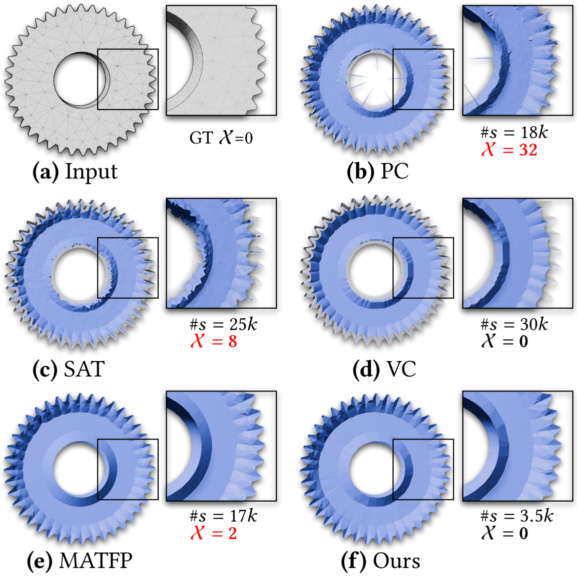


Fig. 16. Qualitative comparison of topology preservation between our method and MATFP [Wang et al. 2022], PC [Amenta et al. 2001], SAT [Miklos et al. 2010], and VC [Yan et al. 2018]. The ground truth *Euler characteristic* $\mathcal{X}$ is shown in (a), and #s represents the number of medial spheres generated.

Each updated partial RPD will change the medial mesh connectivity within the corresponding local regions, as dual of volumetric RPD. These newly added medial spheres may already satisfy the criteria for local structures, thus avoiding other similar spheres (*i.e.,* spheres with similar centers and radii) to be inserted.

Since we use the same feature preservation strategy as MATFP, our methods shows comparable if not better reconstruction quality, as shown in Table 2 and Fig. 18. More visualizations are shown in Fig. 22. Specifically, we use $\delta^{\epsilon} = 0.6$ for models shown in Fig 18. For more detailed statistics, please refer to the Supplementary Document.

Table 2. Quantitative comparison on topology preservation with MATFP [Wang et al. 2022]. #s is the number of generated medial spheres. ϵ^{max} is the two-sided Hausdorff Distance between the original surface and reconstruction, as maximum of ϵ^1 and ϵ^2 described in Sec. 6. We show the Euler characteristic as $\mathcal{X}$ and ground truth as 'GT $\mathcal{X}$'. Comparing to MATFP which has no guarantee of homotopy equivalence, our method gives correct Euler characteristic with lower number of generated medial spheres and competitive reconstruction quality.

Model ID	MATFP			Ours		
(GT $\mathcal{X}$)	#s	ϵ^{max}	$\mathcal{X}$	#s	ϵ^{max}	$\mathcal{X}$
549 (-6)	21k	1.282	-5	$7.2k^{\star}$	**1.077**	**-6**
4123 (-3)	17k	3.799	-1	$5.1k^{\star}$	**1.095**	**-3**
5227 (-5)	12k	**0.442**	5	$3.2k^{\star}$	1.419	**-5**
8315 (-4)	11k	3.351	-2	$1.5k^{\star}$	**1.481**	**-4**
8964 (-72)	25k	**0.251**	-37	$8.7k^{\star}$	0.753	**-72**
10836 (-1)	3k	**1.067**	1	$2.7k^{\star}$	1.496	**-1**
11299 (-24)	19k	2.324	-13	$5.8k^{\star}$	**0.961**	**-24**
11790 (0)	17k	1.79	2	$5.3k^{\star}$	**0.819**	**0**
11835 (0)	21k	2.252	-16	$5.4k^{\star}$	**1.015**	**0**
13026 (-4)	26k	2.203	-2	$10k^{\star}$	**1.241**	**-4**
13607 (-13)	19k	**0.956**	-19	$7.3k^{\star}$	1.486	**-13**
14621 (-3)	34k	2.812	-9	$12k^{\star}$	**0.982**	**-3**
15094 (-8)	35k	**1.045**	-4	$7.2k^{\star}$	1.462	**-8**

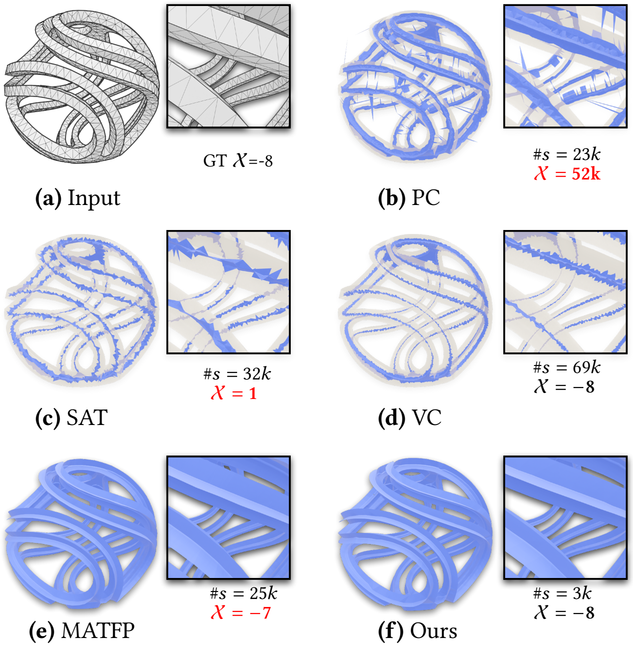


Fig. 17. Comparison of the medial feature and topology preservation ability between our method and PC [Amenta et al. 2001], SAT [Miklos et al. 2010], VC [Yan et al. 2018], and MATFP [Wang et al. 2022]. Our method can not only preserve medial features, but also output the correct topology of the medial mesh.

6.2 Comparison with PC, SAT, and VC Methods

We compare our method with three classical methods for approximating medial axis, including two point-cloud-based methods – PC (Power Crust [Amenta et al. 2001]) and SAT (Scaled Axis Transform [Miklos et al. 2010]), and one voxel-based method VC (Voxel Cores [Yan et al. 2018]), regarding feature and topology preservation quality (Fig 17), and surface reconstruction quality from the generated medial mesh (Fig 18).

We test the PC [Amenta et al. 2001] method using two different sampling densities, and experiment SAT [Miklos et al. 2010] method using two values of sampling parameter: $\delta = 0.04$ and $\delta = 0.03$ in Fig 18, and set the scale parameter as default 1.0 in all experiments. Similar to all other point-cloud-based methods, the quality of medial mesh generated would improve when the surface sampling density increases. However, these two methods generate medial meshes that are not thin with large number of flat tetrahedrons, and their Euler characteristic $\mathcal{X}$ varies as shown in Fig 16, *i.e.,* $\mathcal{X} = 32$ and $\mathcal{X} = 8$ respectively. In addition, they cannot preserve any medial features, both externally and internally, as shown in Fig 17.

We also compare with VC [Yan et al. 2018] method regarding the reconstruction quality and the feature preservation quality of generated medial mesh. We use two voxel sizes 2^8 and 2^9 with pruning parameters $\lambda = 0.03$, shown in Fig 18. Even though VC can output the topologically-correct medial mesh, it requires large number of medial spheres for outputting a smooth structure around the internal features, almost 10 times more than our method (*i.e.,* 30k in Fig 16). This is due to the fact that VC controls the sampling globally based on the size of the voxels. Smaller voxel size would generates denser medial spheres. It cannot, however, directly control the sampling rate of medial spheres as needed. We also found that the medial mesh generated from VC shrinks more as the value of pruning parameter increases, which will result in a more rounded reconstruction result around the external features.

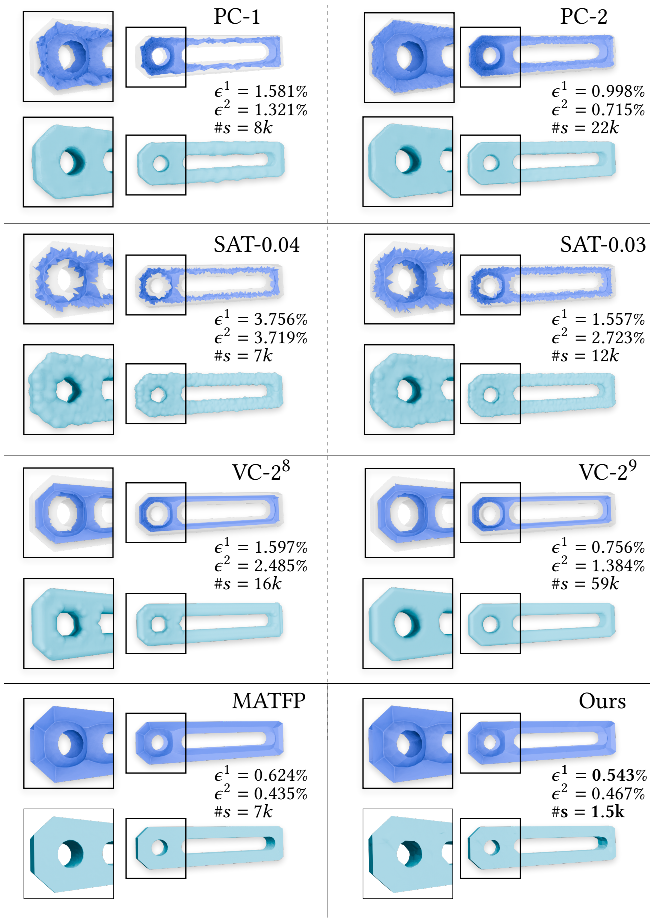


Fig. 18. Qualitative comparison of the medial mesh and the reconstructed mesh among ours and PC [Amenta et al. 2001], SAT [Miklos et al. 2010], VC [Yan et al. 2018], and MATFP [Wang et al. 2022]. Here ϵ^1 and ϵ^2 are the Hausdorff distance errors described in Sec 6.

6.3 Comparisons on Organic Models

We also test our method on 12 organic models, and compare it with PC [Amenta et al. 2001], SAT [Miklos et al. 2010], VC [Yan et al. 2018] and MATFP [Wang et al. 2022]. We show the visual and quantitative comparison in Fig. 21. More statistics can be found in Table. 3 of the Supplementary Material.

We experiment SAT using the sampling parameter $\delta = 0.03$ and set the scale parameter as default value 1.0. We compare with the VC using the voxel size 2^8 and the pruning parameter $\lambda = 0.03$. For our method, we set $\delta^\epsilon = 1.5$ for all organic models tested.

We have found that our method normally generates similar reconstruction accuracy with much fewer medial spheres (*i.e.,* PC 14k, SAT 17k, VC 36k, MATFP 11k, and ours 4k for the 'Fertility' model). Our method can also maintain the topology and thinness property of the 3D medial axis w.r.t. the input shape, same as VC (*i.e.,* PC 35k, SAT 7, VC 1, MATFP 4, and ours 1 for the Euler characteristic $\mathcal{X}$ of 'Rozy' model).

6.4 Ablation Study

Geometric Error Bound δ^ϵ. One important parameter used in our method is the user-defined geometric error bound δ^ϵ, which controls the one-sided reconstruction accuracy (from input surface to the reconstructed mesh). Here the error is measured by the distance from surface samples to the enveloping volume of the medial mesh. Similar to Hausdorff distance described in Sec 6, this error is also scaled based on the diagonal length of the shape's bounding box. We show the effect of two different values of the parameter δ^ϵ in Fig. 19. A smaller value of δ^ϵ would generate a smoother medial mesh around non-feature regions, as more non-feature spheres are sampled. It can also reduce the non-smooth connections on medial features, either external or internal ones (see the black and red curves respectively in Fig. 19). As a result, a smaller error bound will inevitability output a reconstructed mesh with better quality.

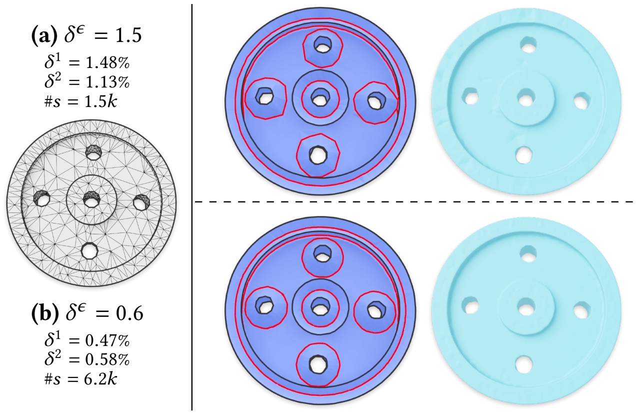


Fig. 19. Ablation study on geometric error bound δ^ϵ. We use two different error bounds: (a) δ^ϵ = 1.5 and (b) δ^ϵ = 0.6. Smaller error bound can generate a medial mesh with smoother features which results in a better reconstruction quality.

Initialization. There are two parameters that may impact our running time. One is #t as the number of tets in the input tetrahedral mesh; the other is #s_{init} as the number of initial spheres. We use the 'Happy Buddha' model [Dey et al. 2007] (genus=8) for testing with the error bound δ^ϵ = 1.5 and show the results in Fig. 20. Our method can generate the medial mesh with the correct topology ($\mathcal{X}$=-7) and

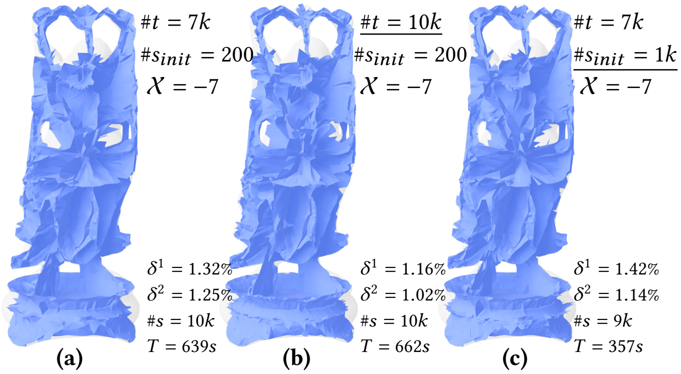


Fig. 20. Ablation study on two initialization parameters: (1) $\#t$ as the number of tets in the input tetrahedral mesh; (2) $\#s_{init}$ as the number of initial spheres. The base case (a) uses $\#s_{init} = 200$ and $\#t = 7k$. While keeping the same and correct topology (GT genus=8, GT $\mathcal{X} = -7$) and similar reconstruction ability, a higher number of $\#t$ would inevitably increase the running time (b), and a higher number of $\#s_{init}$ may reduce the processing time (c).

Table 3. Statistic of our running time in seconds. $\#t$ is the number of tets in the given tetrahedral mesh. $\#s$ is the number of generated medial spheres. #RPD is the number of volumetric RPD calculated. S_{topo} is the running time of topology preservation step (Sec. 4.1). S_{extf} and S_{intf} is the running time for preserving external features and internal features (Sec 4.2). The model's ID# correspond to those shown in Table 2.

Model ID	$\#t$	$\#s$	#RPD	S_{topo}	S_{extf}	S_{intf}	S_{geo}	Total (s)
549	8k	7.2k	34	74	115	376	99	664
4123	1.2k	5.1k	35	15	42	109	32	198
5227	1k	3.2k	31	10	22	23	17	72
8315	4k	1.5k	37	64	73	444	112	693
8964	3.6k	8.7k	28	36	113	394	16	559
10836	3.8k	2.7k	41	30	29	126	63	248

similar reconstruction quality using different value of $\#t$ and $\#s_{init}$. Increasing $\#t$ (*i.e.*, smaller value of 'l' parameter of fTetwild [Hu et al. 2020]) would inevitably make our RPD calculation slower as more 'Tet-Cell' clipping steps need to be performed. A larger number of $\#s_{init}$ (*i.e.,* use $1k$ than 200 in Fig. 20) would reduce the processing time.

7 Limitations and future work

Our method can produce an approximated MAT that is visually and quantitatively similar to those of MATFP [Wang et al. 2022] geometry-wise, with less spheres and the additional benefit of topology preservation. Although both our method and MATFP rely on RPD, the former employs volumetric RPD, which necessitates cutting tets by half-spaces inside the volume, while the latter utilizes surface RPD, which only considers the intersection between triangles and half-spaces. Consequently, our method requires more computational time. As the runtime statistics shown in Table 3, our method takes minutes for each model even though each step's calculation of volumetric RPD on GPU only takes about 1-3 seconds. In the future, we will consider parallelising the feature preserving (S_{extf} and S_{intf}) stage as much as possible to reduce the runtime.

Moreover, our current GPU-based implementation may fail to compute RPD if the given tetrahedral mesh has a very large number of tets. For example, the model #12280 in Supplementary Document contains over 5.5 million tets even using the fTetwild [Hu et al. 2020] with the largest length parameter value $l = 1$. We leave this computational issue for future exploration.

Acknowledgments

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions. Ningna Wang and Xiaohu Guo were partially supported by the National Science Foundation (OAC-2007661). Hui Huang was partially supported by the National Natural Science Foundation of China (U21B2023, U2001206) and Guangdong Basic and Applied Basic Research Foundation (2023B1515120026).

References

Ahmed Abdelkader, Chandrajit L Bajaj, Mohamed S Ebeida, Ahmed H Mahmoud, Scott A Mitchell, John D Owens, and Ahmad A Rushdi. 2020. VoroCrust: Voronoi meshing without clipping. *ACM Transactions on Graphics (TOG)* 39, 3 (2020), 1–16.

Nina Amenta, Sunghee Choi, and Ravi Krishna Kolluri. 2001. The power crust. In *Proceedings of the sixth ACM symposium on Solid modeling and applications*. 249–266.

Franz Aurenhammer. 1987. Power diagrams: properties, algorithms and applications. *SIAM J. Comput.* 16, 1 (1987), 78–96.

Justine Basselin, Laurent Alonso, Nicolas Ray, Dmitry Sokolov, Sylvain Lefebvre, and Bruno Lévy. 2021. Restricted power diagrams on the GPU. In *Computer Graphics Forum*, Vol. 40. Wiley Online Library, 1–12.

Harry Blum et al. 1967. *A transformation for extracting new descriptors of shape*. Vol. 43. MIT press Cambridge, MA.

Jonathan W Brandt and V Ralph Algazi. 1992. Continuous skeleton computation by Voronoi diagram. *CVGIP: Image understanding* 55, 3 (1992), 329–338.

Gunnar Carlsson and Mikael Vejdemo-Johansson. 2021. *Topological data analysis with applications*. Cambridge University Press.

Frédéric Chazal and André Lieutier. 2005. The "λ-medial axis". *Graphical Models* 67, 4 (2005), 304–331.

Frédéric Chazal and André Lieutier. 2008. Smooth manifold reconstruction from noisy and non-uniform approximation with guarantees. *Computational Geometry* 40, 2 (2008), 156–170.

Tim Culver, John Keyser, and Dinesh Manocha. 2004. Exact computation of the medial axis of a polyhedron. *Computer Aided Geometric Design* 21, 1 (2004), 65–98.

Tamal K Dey, Kuiyu Li, and Jian Sun. 2007. On computing handle and tunnel loops. In *2007 International Conference on Cyberworlds (CW'07)*. IEEE, 357–366.

Tamal K Dey and Wulue Zhao. 2002. Approximate medial axis as a voronoi subcomplex. In *Proceedings of the seventh ACM symposium on Solid modeling and applications*. 356–366.

Tamal K Dey and Wulue Zhao. 2004. Approximating the medial axis from the Voronoi diagram with a convergence guarantee. *Algorithmica* 38, 1 (2004), 179–200.

Zhiyang Dou, Cheng Lin, Rui Xu, Lei Yang, Shiqing Xin, Taku Komura, and Wenping Wang. 2022. Coverage Axis: Inner Point Selection for 3D Shape Skeletonization. In *Computer Graphics Forum*, Vol. 41. Wiley Online Library, 419–432.

Zhiyang Dou, Shiqing Xin, Rui Xu, Jian Xu, Yuanfeng Zhou, Shuangmin Chen, Wenping Wang, Xiuyang Zhao, and Changhe Tu. 2020. Top-down shape abstraction based on greedy pole selection. *IEEE Transactions on Visualization and Computer Graphics* 27, 10 (2020), 3982–3993.

Andreas Fabri and Sylvain Pion. 2009. CGAL: The computational geometry algorithms library. In *Proceedings of the 17th ACM SIGSPATIAL international conference on advances in geographic information systems*. 538–539.

Zhiying Fu, Rui Xu, Shiqing Xin, Shuangmin Chen, Changhe Tu, Chenglei Yang, and Lin Lu. 2022. Easyvrmodeling: Easily create 3d models by an immersive vr system. *Proceedings of the ACM on Computer Graphics and Interactive Techniques* 5, 1 (2022), 1–14.

Mengyuan Ge, Junfeng Yao, Baorong Yang, Ningna Wang, Zhonggui Chen, and Xiaohu Guo. 2023. Point2MM: Learning medial mesh from point clouds. *Computers & Graphics* 115 (2023), 511–521.

Wim H Hesselink and Jos BTM Roerdink. 2008. Euclidean skeletons of digital image and volume data in linear time by the integer medial axis transform. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30, 12 (2008), 2204–2217.

Jianwei Hu, Gang Chen, Baorong Yang, Ningna Wang, Xiaohu Guo, and Bin Wang. 2022. IMMAT: Mesh reconstruction from single view images by medial axis transform prediction. *Computer-Aided Design* 150 (2022), 103304.

Jianwei Hu, Bin Wang, Lihui Qian, Yiling Pan, Xiaohu Guo, Lingjie Liu, and Wenping Wang. 2019. MAT-Net: Medial Axis Transform Network for 3D Object Recognition. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI'19)*. 774–781.

Jianwei Hu, Ningna Wang, Baorong Yang, Gang Chen, Xiaohu Guo, and Bin Wang. 2023. S3DS: Self-supervised Learning of 3D Skeletons from Single View Images. In *Proceedings of the 31st ACM International Conference on Multimedia*. 6948–6958.

Yixin Hu, Teseo Schneider, Bolun Wang, Denis Zorin, and Daniele Panozzo. 2020. Fast tetrahedral meshing in the wild. *ACM Transactions on Graphics (TOG)* 39, 4 (2020), 117–1.

Andrei C. Jalba, Jacek Kustra, and Alexandru C. Telea. 2013. Surface and Curve Skeletonization of Large 3D Models on the GPU. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35, 6 (2013), 1495–1508. https://doi.org/10.1109/TPAMI.2012.212

Andrei C Jalba, Andre Sobiecki, and Alexandru C Telea. 2015. An unified multiscale framework for planar, surface, and curve skeletonization. *IEEE transactions on pattern analysis and machine intelligence* 38, 1 (2015), 30–45.

Tao Ju, Qian-Yi Zhou, and Shi-Min Hu. 2007. Editing the topology of 3D models by sketching. *ACM Transactions on Graphics (TOG)* 26, 3 (2007), 42–es.

Sebastian Koch, Albert Matveev, Zhongshi Jiang, Francis Williams, Alexey Artemov, Evgeny Burnaev, Marc Alexa, Denis Zorin, and Daniele Panozzo. 2019. Abc: A big cad model dataset for geometric deep learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 9601–9611.

Jacek Kustra, Andrei Jalba, and Alexandru Telea. 2016. Computing refined skeletal features from medial point clouds. *Pattern Recognition Letters* 76 (2016), 13–21. https://doi.org/10.1016/j.patrec.2015.05.007 Special Issue on Skeletonization and its Application.

Lei Lan, Ran Luo, Marco Fratarcangeli, Weiwei Xu, Huamin Wang, Xiaohu Guo, Junfeng Yao, and Yin Yang. 2020. Medial Elastics: Efficient and Collision-Ready Deformation via Medial Axis Transform. *ACM Trans. Graph.* 39, 3, Article 20 (apr 2020).

J. Leray. 1950. *L'anneau spectral et l'anneau filtré d'homologie d'un espace localement compact et d'une application continue: (cours professés au Collège de France en 1947-1948 et 1949-1950)*. Gauthier-Villars. https://books.google.com/books?id=QYGKzwEACAAJ

Bruno Lévy and Alain Filbois. 2015. Geogram: a library for geometric algorithms. (2015).

Pan Li, Bin Wang, Feng Sun, Xiaohu Guo, Caiming Zhang, and Wenping Wang. 2015. Q-mat: Computing medial axis transform by quadratic error minimization. *ACM Transactions on Graphics (TOG)* 35, 1 (2015), 1–16.

André Lieutier. 2004. Any open bounded subset of Rn has the same homotopy type as its medial axis. *Computer-Aided Design* 36, 11 (2004), 1029–1046.

André Lieutier and Mathijs Wintraecken. 2023. Hausdorff and gromov-hausdorff stable subsets of the medial axis. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*. 1768–1776.

Cheng Lin, Changjian Li, Yuan Liu, Nenglun Chen, Yi-King Choi, and Wenping Wang. 2021. Point2skeleton: Learning skeletal representations from point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 4277–4286.

Lu Liu, Erin W Chambers, David Letscher, and Tao Ju. 2010. A simple and robust thinning algorithm on cell complexes. In *Computer Graphics Forum*, Vol. 29. Wiley Online Library, 2253–2260.

Xiaohan Liu, Lei Ma, Jianwei Guo, and Dong-Ming Yan. 2020. Parallel computation of 3D clipped Voronoi diagrams. *IEEE Transactions on Visualization and Computer Graphics* 28, 2 (2020), 1363–1372.

Yilin Liu, Jiale Chen, Shanshan Pan, Daniel Cohen-Or, Hao Zhang, and Hui Huang. 2024. Split-and-Fit: Learning B-Reps via Structure-Aware Voronoi Partitioning. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)* 43, 4 (2024), 108:1–108:13.

Jaehwan Ma, Sang Won Bae, and Sunghee Choi. 2012. 3D medial axis point approximation using nearest neighbors and the normal field. *The Visual Computer* 28, 1 (2012), 7–19.

Balint Miklos, Joachim Giesen, and Mark Pauly. 2010. Discrete scale axis representations for 3D geometry. In *ACM SIGGRAPH 2010 papers*. 1–10.

Victor Milenkovic. 1993. Robust Construction of the Voronoi Diagram of a Polyhedron.. In *CCCG*, Vol. 93. Citeseer, 473–478.

Yuta Noma, Silvia Sellán, Nicholas Sharp, Karan Singh, and Alec Jacobson. 2024. Surface-Filling Curve Flows via Implicit Medial Axes. *ACM Transactions on Graphics (TOG)* 43, 4 (2024), 1–12.

Dmitry Petrov, Pradyumn Goyal, Vikas Thamizharasan, Vladimir Kim, Matheus Gadelha, Melinos Averkiou, Siddhartha Chaudhuri, and Evangelos Kalogerakis. 2024. GEM3D: GEnerative Medial Abstractions for 3D Shape Synthesis. In *ACM SIGGRAPH 2024 Conference Papers*. 1–11.

Nicolas Ray, Dmitry Sokolov, Sylvain Lefebvre, and Bruno Lévy. 2018. Meshless Voronoi on the GPU. *ACM Transactions on Graphics (TOG)* 37, 6 (2018), 1–12.

Martin Rumpf and Alexandru Telea. 2002. A continuous skeletonization method based on level sets. In *EPRINTS-BOOK-TITLE*. University of Groningen, Johann Bernoulli Institute for Mathematics and Computer Science.

Punam K Saha, Gunilla Borgefors, and Gabriella Sanniti di Baja. 2016. A survey on skeletonization algorithms and their applications. *Pattern recognition letters* 76 (2016), 3–12.

Evan C Sherbrooke, Nicholas M Patrikalakis, and Erik Brisson. 1996. An algorithm for the medial axis transform of 3D polyhedral solids. *IEEE transactions on visualization and computer graphics* 2, 1 (1996), 44–61.

Kaleem Siddiqi, Sylvain Bouix, Allen Tannenbaum, and Steven W Zucker. 2002. Hamilton-jacobi skeletons. *International Journal of Computer Vision* 48 (2002), 215–231.

Kaleem Siddiqi and Stephen Pizer. 2008. *Medial representations: mathematics, algorithms and applications*. Vol. 37. Springer Science & Business Media.

André Sobiecki, Andrei Jalba, and Alexandru Telea. 2014. Comparison of curve and surface skeletonization methods for voxel shapes. *Pattern Recognition Letters* 47 (2014), 147–156.

Shibo Song and Ningna Wang. 2023. blender-mat-addon. https://github.com/songshibo/blender-mat-addon.

Andrea Tagliasacchi, Thomas Delame, Michela Spagnuolo, Nina Amenta, and Alexandru Telea. 2016. 3d skeletons: A state-of-the-art report. In *Computer Graphics Forum*, Vol. 35. Wiley Online Library, 573–597.

Ningna Wang, Bin Wang, Wenping Wang, and Xiaohu Guo. 2022. Computing Medial Axis Transform with Feature Preservation via Restricted Power Diagram. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia 2022)* 41, 6 (2022).

Zimeng Wang, Zhiyang Dou, Rui Xu, Cheng Lin, Yuan Liu, Xiaoxiao Long, Shiqing Xin, Lingjie Liu, Taku Komura, Xiaoming Yuan, et al. 2024. Coverage Axis++: Efficient Inner Point Selection for 3D Shape Skeletonization. *arXiv preprint arXiv:2401.12946* (2024).

Rui Xu, Zhiyang Dou, Ningna Wang, Shiqing Xin, Shuangmin Chen, Mingyan Jiang, Xiaohu Guo, Wenping Wang, and Changhe Tu. 2023. Globally consistent normal orientation for point clouds by regularizing the winding-number field. *ACM Transactions on Graphics (TOG)* 42, 4 (2023), 1–15.

Rui Xu, Longdu Liu, Ningna Wang, Shuangmin Chen, Shiqing Xin, Xiaohu Guo, Zichun Zhong, Taku Komura, Wenping Wang, and Changhe Tu. 2024. CWF: Consolidating Weak Features in High-quality Mesh Simplification. *ACM Transactions on Graphics (TOG)* 43, 4 (2024). https://doi.org/10.1145/3658159

Rui Xu, Zixiong Wang, Zhiyang Dou, Chen Zong, Shiqing Xin, Mingyan Jiang, Tao Ju, and Changhe Tu. 2022. RFEPS: Reconstructing feature-line equipped polygonal surface. *ACM Transactions on Graphics (TOG)* 41, 6 (2022), 1–15.

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: 6th International Conference, GMP 2010, Castro Urdiales, Spain, June 16-18, 2010. Proceedings 6*. Springer, 269–282.

Yajie Yan, David Letscher, and Tao Ju. 2018. Voxel cores: Efficient, robust, and provably good approximation of 3d medial axes. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 1–13.

Yajie Yan, Kyle Sykes, Erin Chambers, David Letscher, and Tao Ju. 2016. Erosion thickness on medial axes of 3D shapes. *ACM Transactions on Graphics (TOG)* 35, 4 (2016), 1–12.

Baorong Yang, Junfeng Yao, and Xiaohu Guo. 2018. DMAT: Deformable medial axis transform for animated mesh approximation. In *Computer Graphics Forum*, Vol. 37. Wiley Online Library, 301–311.

Baorong Yang, Junfeng Yao, Bin Wang, Jianwei Hu, Yiling Pan, Tianxiang Pan, Wenping Wang, and Xiaohu Guo. 2020. P2MAT-NET: Learning medial axis transform from sparse point clouds. *Computer Aided Geometric Design* 80 (2020), 101874.

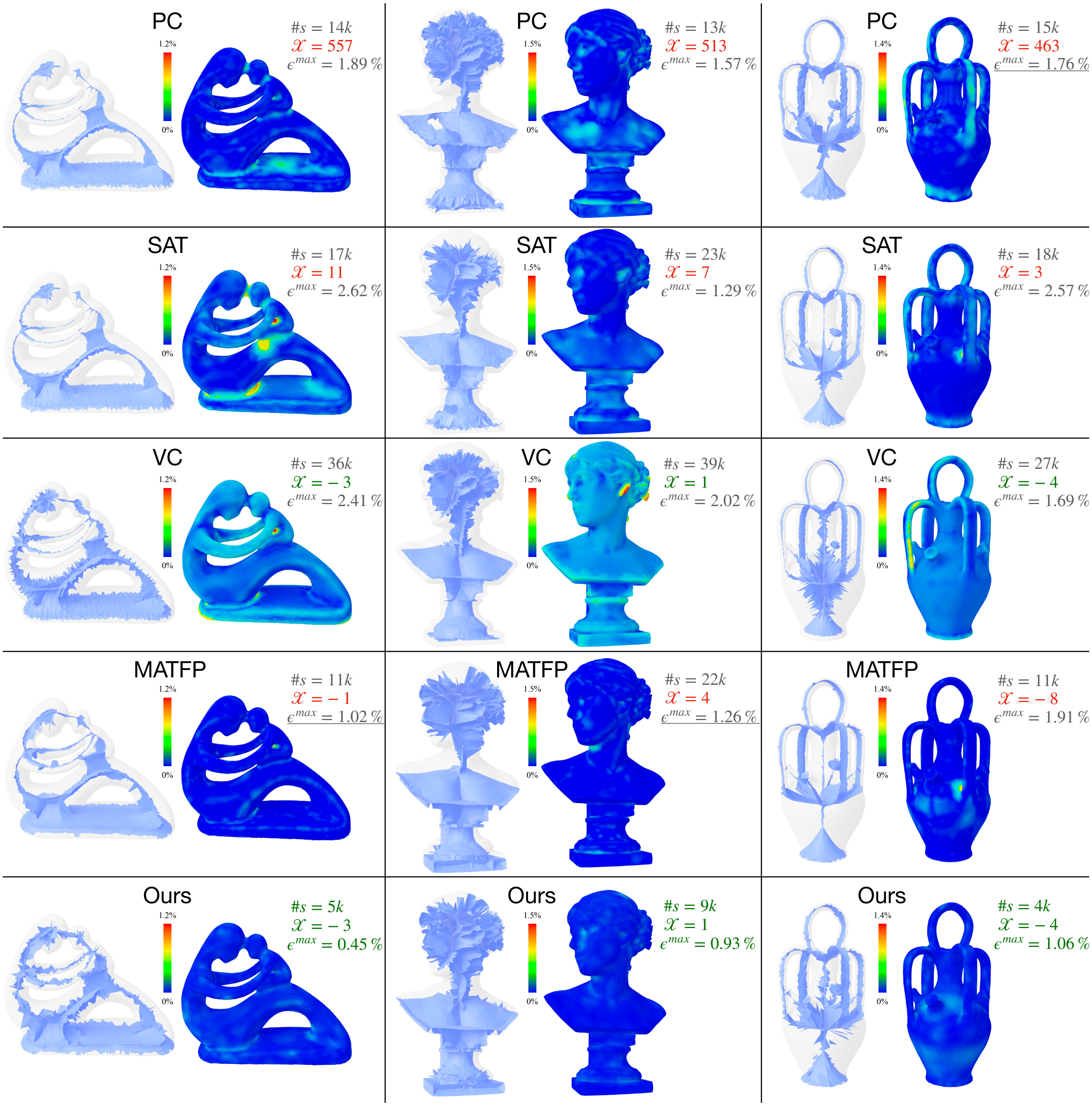


Fig. 21. Comparison with PC [Amenta et al. 2001], SAT [Miklos et al. 2010], VC [Yan et al. 2018] and MATFP [Wang et al. 2022] on three organic models. We show the generated medial mesh and the color-coded distribution of two-sided Hausdorff errors between the input surface and the reconstructed surface. Here, s is the number of generated medial spheres. And $\mathcal{X}$ is the Euler characteristic of the computed medial mesh, where a wrong number is shown in red. ϵ^{max} is the maximum of two-sided Hausdorff error, as the best result shown in green and the second best shown with underline.

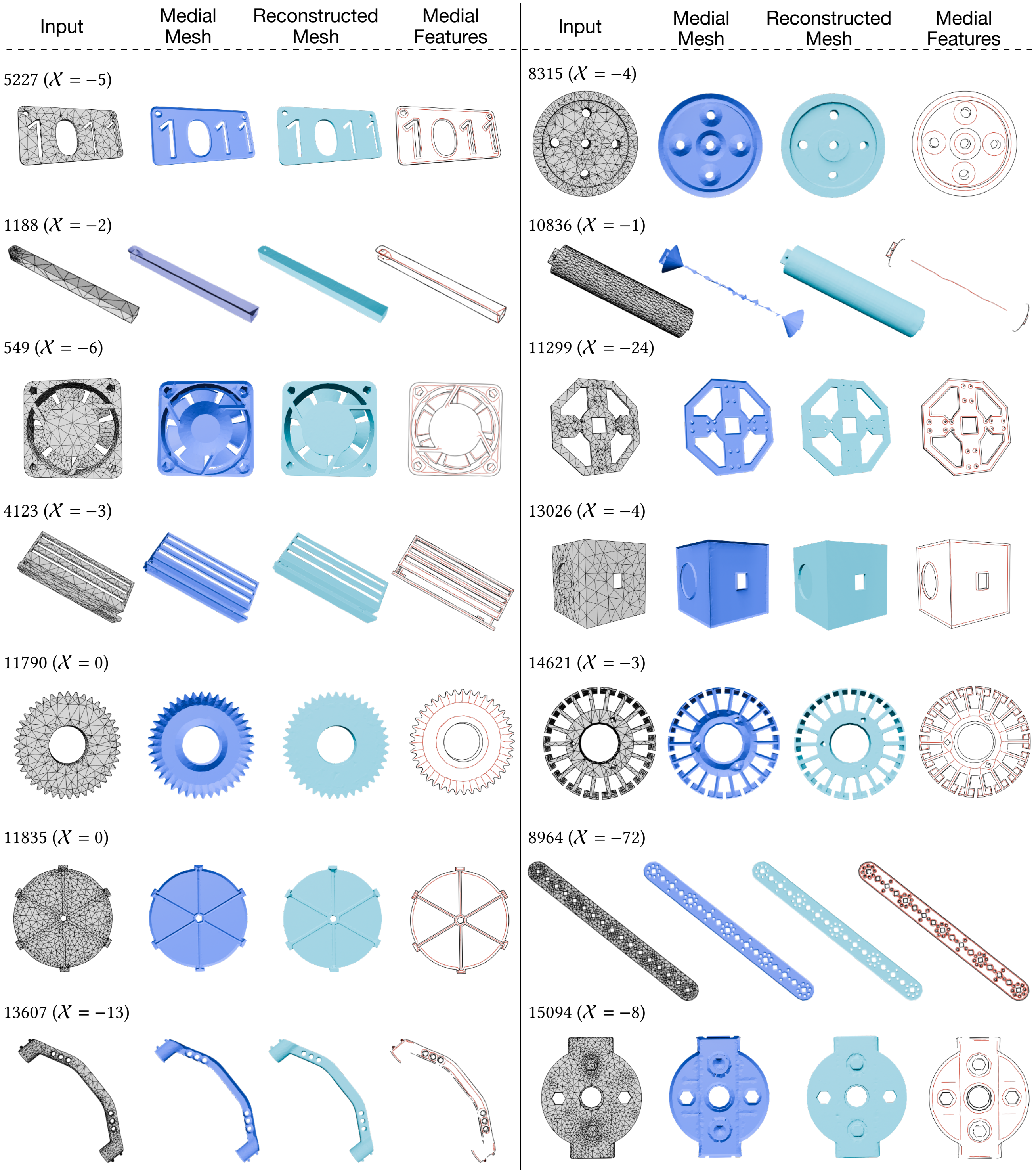

Input
Medial Mesh
Reconstructed Mesh
Medial Features
Input
Medial Mesh
Reconstructed Mesh
Medial Features
5227 ($\mathcal{X} = -5$)
8315 ($\mathcal{X} = -4$)
1188 ($\mathcal{X} = -2$)
10836 ($\mathcal{X} = -1$)
549 ($\mathcal{X} = -6$)
11299 ($\mathcal{X} = -24$)
4123 ($\mathcal{X} = -3$)
13026 ($\mathcal{X} = -4$)
11790 ($\mathcal{X} = 0$)
14621 ($\mathcal{X} = -3$)
11835 ($\mathcal{X} = 0$)
8964 ($\mathcal{X} = -72$)
13607 ($\mathcal{X} = -13$)
15094 ($\mathcal{X} = -8$)

Fig. 22. Visualization of models shown in Table 2, with their model ID and Euler characteristic $\mathcal{X}$. From left to right are the input tetrahedral meshes, the generated medial meshes, the surfaces reconstructed from our medial meshes, and the extracted medial features. For the medial features, the black curves are the external features and the red curves are the internal features.