Feature-Preserving Mesh Repair via Restricted Power Diagram

HUIBIAO WEN, Shandong University, China and University of Health and Rehabilitation Sciences, China
GUILONG HE, Shandong University, China
RUI XU, University of Hong Kong, China
SHUANGMIN CHEN, Qingdao University of Science and Technology, China
SHIQING XIN*, Shandong University, China
ZHENYU SHU, NingboTech University, China
TAKU KOMURA, University of Hong Kong, China
JIEQING FENG, State Key Laboratory of CAD & CG, Zhejiang University, China
WENPING WANG, Texas A&M University, USA
CHANGHE TU, Shandong University, China

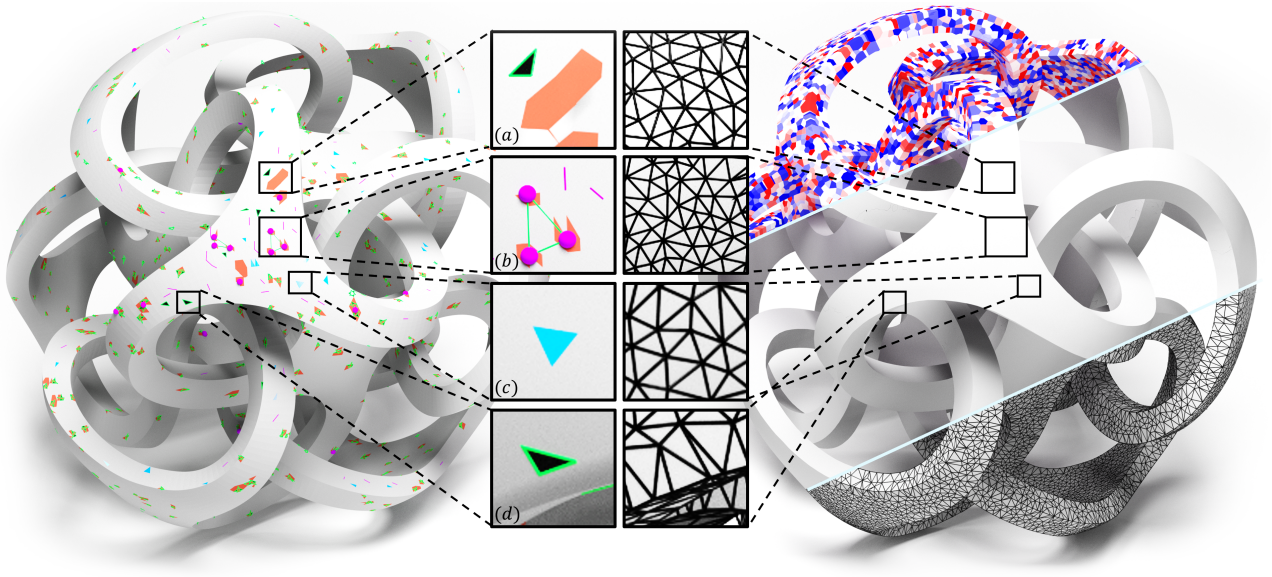


Fig. 1. Given a polygonal surface with various defects—such as self-intersections (highlighted in red, (a)), non-manifold vertices (in purple, (b)), duplicate facets (in cyan, (c)), and narrow gaps or cracks (bounded by green curves, (d))—we propose a unified method, supported by a wrap surface, to repair these imperfections. Our approach generates a watertight, manifold mesh while preserving the inherent feature lines.

*Corresponding author: Shiqing Xin.

Authors' Contact Information: Huibiao Wen, Shandong University, Qingdao, China and University of Health and Rehabilitation Sciences, Qingdao, China, ericvein@163.com; Guilong He, Shandong University, Qingdao, China, hegl2533@gmail.com; Rui Xu, University of Hong Kong, Hong Kong, China, ruixu1999@connect.hku.hk; Shuangmin Chen, Qingdao University of Science and Technology, Qingdao, China, csmqq@163.com; Shiqing Xin, Shandong University, Qingdao, China, xinshiqing@sdu.edu.cn; Zhenyu Shu, NingboTech University, Ningbo, China, shuzhenyu@nit.zju.edu.cn; Taku Komura, University of Hong Kong, Hong Kong, China, taku@cs.hku.hk; Jieqing Feng, State Key Laboratory of CAD & CG, Zhejiang University, Hangzhou, China, jqfeng@cad.zju.edu.cn; Wenping Wang, Texas A&M University, College Station, USA, wenping@tamu.edu; Changhe Tu, Shandong University, Qingdao, China, chtu@sdu.edu.cn.

Mesh repair is a critical process in 3D geometry processing aimed at correcting errors and imperfections in polygonal meshes to produce watertight, manifold, and feature-preserving meshes suitable for downstream tasks. While errors such as degeneracies, duplication, holes, and overlaps can be addressed through standard repair processes, cracks along trimmed curves require special attention and should ideally be repaired to align with sharp feature lines.

In this paper, we present a unified framework for repairing diverse mesh imperfections by leveraging a manifold wrap surface as a mediating agent. The primary role of the wrap surface is to define spatial connections between points on the original surface, thereby decoupling the challenges of edge connectivity and point relocation during repair. Throughout the process, our algorithm operates on the dual objects: the original defective mesh and

SIGGRAPH Conference Papers '25, Vancouver, BC, Canada

ACM ISBN 979-8-4007-1540-2/25/08
https://doi.org/10.1145/3721238.3730671

the manifold wrap surface. The implementation begins by extracting a set of samples from the wrap surface and projecting them onto the original surface. These projected samples are optimized by minimizing the quadratic error relative to the tangent planes of neighboring points on the original surface. Notably, samples far from feature lines remain unchanged, while samples near feature lines converge to those lines even when the input surface lacks correct mesh topology. We then assign an adaptive weight to each sample based on the squared moving distance. By introducing this weight setting, we observe that the restricted power diagram prioritizes connectivity along feature lines, thereby effectively preserving sharp features. Through extensive experiments, we demonstrate the superiority of our proposed algorithm over existing methodologies in terms of manifoldness, watertightness, topological correctness, triangle quality, and feature preservation.

CCS Concepts: • **Computing methodologies** → **Shape analysis**; **Mesh geometry models**.

Additional Key Words and Phrases: mesh repair, restricted power diagram, wrap surface, manifold and watertight

ACM Reference Format:
Huibiao Wen, Guilong He, Rui Xu, Shuangmin Chen, Shiqing Xin, Zhenyu Shu, Taku Komura, Jieqing Feng, Wenping Wang, and Changhe Tu. 2025. Feature-Preserving Mesh Repair via Restricted Power Diagram. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers (SIGGRAPH Conference Papers '25), August 10–14, 2025, Vancouver, BC, Canada.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3721238.3730671

1 Introduction

Mesh repair is a critical process in 3D geometry processing aimed at correcting errors and imperfections in polygonal meshes, which are commonly used in applications such as computer-aided design (CAD), animation, 3D printing, and medical imaging. Mesh imperfections encompass a wide range of topological, geometric, and feature-related errors that compromise the quality and usability of 3D models, as seen in datasets like Thingi10K [Zhou and Jacobson 2016] and ABC [Koch et al. 2019]. The goal of mesh repair is to produce watertight, manifold, and feature-preserving meshes suitable for downstream tasks.

The imperfections of a defective mesh are diverse, including self-intersections, non-manifold edges/vertices, duplicate facets, and narrow gaps/cracks, as shown in Fig. 1 and Fig. 2. The primary challenge of mesh repair lies in developing a unified approach to address these various defects. Traditional repair methods either focus on local defect correction or adopt global strategies to ensure watertightness and manifoldness [Attene et al. 2013]. However, local methods, while effective at preserving fine details, may introduce new artifacts during repair [Attene 2010, 2014]. Global techniques [Chu et al. 2019; Ju 2004], such as volumetric representations and graph-based algorithms, provide robust solutions but may suffer from the loss of sharp features. In recent years, hybrid methods [Huang et al. 2020] combine localized adjustments with global optimization to improve both robustness and detail preservation. Additionally, visual-guided techniques [Zheng et al. 2024] utilize visual correctness as a guiding principle, leveraging visual cues such as visibility and orientation to produce repairs that align with human perception.

The goals of this paper are three-fold. First, we aim to deliver an all-in-one solution to the mesh repair problem. Second, considering that cracks are often caused by trimmed curves, they should be repaired to align with sharp feature lines rather than simply being filled. As Fig. 2 shows, VisualRepair [Chu et al. 2019] entirely overlooks the cracks, while VolumeMesher [Diazzi and Attene 2021] treats these cracks as simple holes, filling them with narrow triangles. Finally, we ensure that the repaired model maintains high triangle quality.

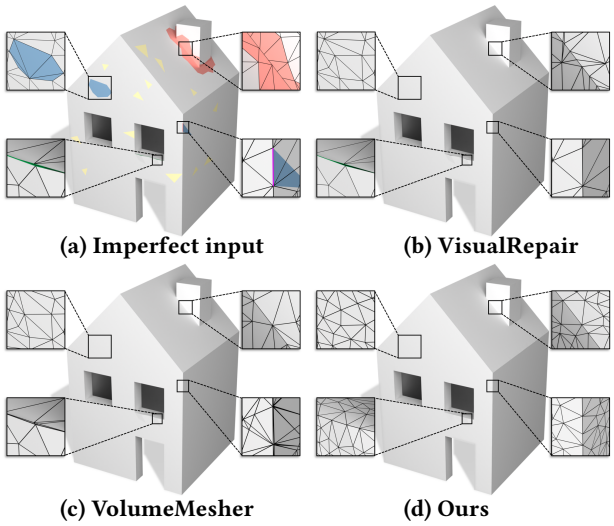


Fig. 2. (a) A polygonal surface exhibits various defects, including self-intersections (highlighted in red), duplicate facets (in cyan), cracks (bounded by green curves), unoriented facets (in yellow), and non-manifold edges (in purple). (b) VisualRepair [Chu et al. 2019] entirely overlooks the cracks. (c) VolumeMesher [Diazzi and Attene 2021] treats these cracks as simple holes, filling them with narrow triangles. (d) Our approach not only preserves geometric feature lines during mesh repair but also generates high-quality triangles.

In this paper, we present a unified framework for repairing various imperfections by leveraging a manifold wrap surface as a mediating agent. Throughout the process, our algorithm operates on the dual objects: the original defective mesh and the manifold wrap surface. To elaborate, movable points are constrained on the original mesh, but the connections between them are defined based on how they partition the wrap surface. In this way, we decouple the challenges of edge connectivity and point relocation.

Our implementation begins by extracting a representative set of samples from the wrap surface and projecting them onto the original surface. Next, following the spirit of QEM (Quadric Error Metrics) [Garland and Heckbert 1997], we fine-tune the position of each sample by minimizing its quadratic error relative to the tangent planes of neighboring points, ensuring optimal alignment. During this process, samples far from feature lines remain unchanged, while those near feature lines converge precisely to the lines—even in cases where cracks lie along trimmed curves or two walls intersect.

Finally, based on the moving distance of each sample, we compute the restricted power diagram on the wrap surface, assigning each sample a weight equal to the squared moving distance. We observe that the restricted power diagram, with this weight setting, prioritizes connectivity along feature lines, thereby effectively preserving

sharp features. As shown in Fig. 2, our algorithm excels in triangle quality and feature preservation.

In summary, our contributions are threefold:

(1) We propose a conceptually unified algorithm to repair various imperfections. By leveraging a manifold wrap surface as a mediating agent, we ensure the repaired mesh is both watertight and manifold.
(2) We encourage samples near feature lines to converge precisely onto the feature lines by minimizing the quadratic error of each sample relative to the tangent planes of neighboring points.
(3) We employ a restricted power diagram to construct the final repaired surface, introducing a specially designed weight setting to ensure alignment with feature lines.

2 Related Work

In this section, we review several representative works on repairing defective meshes. Additionally, as our approach involves the computation of power diagrams, we review algorithms for constructing restricted power diagrams (RPDs).

2.1 Mesh Repair

Mesh repair methods have been extensively studied over the past two decades. As highlighted in comprehensive surveys [Attene et al. 2013; Ju 2009], these methods can be broadly classified into two main categories: local and global approaches. Local methods aim to address specific defects by analyzing and modifying localized regions of the mesh. In contrast, global methods overcome the limitations of local approaches by treating the entire mesh within a unified framework. These techniques often leverage volumetric representations to identify and resolve inconsistencies across the entire mesh comprehensively.

Local Methods. Local methods address defects within small, localized regions of the mesh, aiming to minimize unnecessary changes and preserve the original geometry. Techniques such as sparse adaptive voxelization combined with dual contouring effectively resolve mesh self-intersections [Bischoff et al. 2005; Ju et al. 2002]. Similarly, self-intersections can be handled by cutting along intersection lines and locally stitching the mesh [Attene 2014], while a more recent approach [Guo and Fu 2024] leverages indirect offset predicates for resolving intersections. Gaps and holes are repaired through methods like surface alignment and stitching with weighted averaging [Turk and Levoy 1994], or by using unsigned distance functions combined with graph-based optimization [Hornung and Kobbelt 2006]. Another approach [Zhao et al. 2007] involves filling holes by creating new triangles, approximating normals, and repositioning vertices using Poisson solvers. To enhance mesh quality, some methods [Attene 2010] trim excessively slender triangles or apply Binary Space Partitioning (BSP) to remove non-manifold and degenerate defects. While local methods are efficient and strive to retain as much of the original geometry as possible, they may lack robustness and occasionally introduce new defects.

Global Methods. While local approaches focus on detecting and fixing specific types of defects, global methods address a broader range of interconnected problems, ensuring that the output forms manifold surfaces enclosing a 3D solid. Most global methods rely on volumetric representations to define and resolve inconsistencies across the entire mesh. Some techniques [Andújar et al. 2002; Marschner et al. 2002; Oomes et al. 1997] enable the mutual conversion between mesh and voxel representations, effectively reducing topological defects. Others integrate range images into cumulative signed distance functions to extract optimized iso-surfaces [Curless and Levoy 1996] or employ space carving methods [Furukawa et al. 2007; Portaneri et al. 2022] to generate watertight manifold meshes. Morphological operator-based methods repair meshes by converting between volumetric and polygonal representations [Nooruddin and Turk 2003], ensuring watertight manifold properties while offering fine-grained controllability [Hétroy et al. 2011]. Despite their versatility, global methods often struggle to preserve sharp feature lines and fine details, making precise mesh repair challenging in certain applications.

Hybrid Methods. Robust mesh repair methods that preserve feature lines have been proposed to address the limitations of traditional approaches. Techniques based on biharmonic fields [Argudo et al. 2015] have proven effective for handling complex topologies. ManifoldPlus [Huang et al. 2020] introduces a method that extracts meshes between occupied and empty voxels, employing projection-based optimization to obtain watertight manifold meshes. VolumeMesher [Diazzi and Attene 2021] leverages indirect geometric predicates and solves minimum graph cut problems to generate high-quality meshes with improved accuracy. ImatiSTL [Attene 2017] combines floating-point and exact arithmetic, balancing efficiency and precision when handling defective geometries. TetWild [Hu et al. 2018] distinguishes mesh interiors and exteriors using winding numbers. Vision-based methods [Chu et al. 2019] adhere to principles of global optimization and minimal local modifications, while another approach [Zheng et al. 2024] employs three ray tracing-based metrics—visibility, orientation, and openness—to guide repairs.

Although existing methods can resolve many mesh defects, they often struggle with inherent CAD model defects such as narrow gaps and self-intersections near feature lines.

2.2 Restricted Voronoi and Power Diagrams

Accurately computing restricted power diagrams (RPD) remains challenging. While 3D clipped Voronoi diagram algorithms [Meng et al. 2023; Yan et al. 2010] and industry-standard tools [Fabri and Pion 2009; Lévy and Filbois 2015; Xu et al. 2024] compute restricted Voronoi diagrams (RVD), they often exhibit long computation times. Recent GPU parallelization efforts [Basselin et al. 2021; Liu et al. 2022; Ray et al. 2018] have accelerated Voronoi/power diagram computations. Notably, Basselin et al. [2021] developed a method to compute integrals over RPD cells without explicit diagram construction. Wang et al. [2020] proposed a robust RVD computation for thin-plate models, later extended by a parallel post-processing method [Zong et al. 2023]. Yan et al. [2009] demonstrated RVD applications in isotropic remeshing. Recent work utilizes RPD for medial axis transforms preserving geometric features [Wang et al. 2024, 2022] and CAD reconstruction [Xu et al. 2022] with feature line preservation.

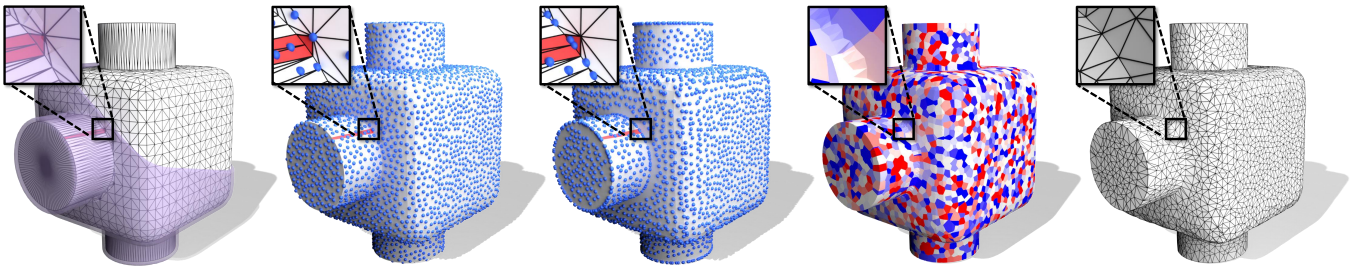


Fig. 3. Algorithm Overview. Given a polygonal surface with self-intersections (highlighted in red), our approach begins by constructing a wrap surface (in purple) as an intermediate representation (a). Next, we project N movable points, extracted from the wrap surface, onto the input surface (b). Following the principle of QEM (Quadric Error Metrics), we fine-tune the position of each sample by minimizing its quadratic error relative to the tangent planes of neighboring points. During this process, samples far from feature lines remain unchanged, while those near feature lines converge precisely to the lines (c). Subsequently, we compute the restricted power diagram (d). Finally, its dual structure, the restricted regular triangulation (e), yields the repaired surface, eliminating various imperfections while preserving distinctive feature lines.

3 Methodology

Given a defective model $\mathcal{M}$ with various imperfections, the task of this paper is to provide an all-in-one solution for repairing these defects. Specifically, the defective model may have cracks along trimmed curves or contain two walls penetrating each other, where feature lines need to be reconstructed.

Our algorithm operates on a pair of objects: the original surface $\mathcal{M}$ and a wrap surface $\mathcal{S}$, which is a manifold surface tightly wrapping the original surface. As shown in Fig. 3, the algorithm consists of the following steps:

Step 1. Compute the wrap surface.
Step 2. Sample N point from the wrap surface and project them onto the input surface.
Step 3. Following the principle of QEM (Quadric Error Metrics), fine-tune the position of each sample by minimizing its quadratic error relative to the tangent planes of neighboring points.
Step 4. Assign an adaptive weight to each optimized point and compute the restricted power diagram (RPD) on the wrap surface.
Step 5. Extract the dual structure of the RPD as the final repaired outcome.

3.1 Wrap Surface and Initial Sample Set

Alpha Wrapping [Portaneri et al. 2022] refines and sculpts a 3D Delaunay triangulation on an offset surface of the input model in a greedy manner. Its most significant advantage is the ability to produce a high-quality manifold mesh surface $\mathcal{S}$ that completely encloses the original mesh $\mathcal{M}$, even when processing defective inputs. The method includes two parameters: one parameter ϵ to control the offset distance and the other α to determine the meshing density. All models were normalized to fit within a unit bounding box in our experiments, with parameter values set to $\epsilon = 1/3000$ and $\alpha = 1/500$.

Next stage, we uniformly sample N points using blue-noise sampling [Corsini et al. 2012], $\{\mathbf{x}_i\}_{i=1}^N$, from the wrap surface and then project them onto the original mesh surface:

$$\mathbf{y}_i = \arg\min_{\mathbf{y}\in\mathcal{M}} \|\mathbf{y} - \mathbf{x}_i\|^2.$$

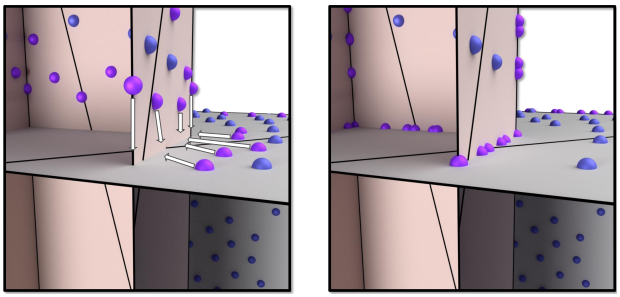


Fig. 4. Our point-shifting strategy effectively handles cases where two walls penetrate each other, pulling nearby points onto potential feature lines.

Note that $\mathbf{y}_i$ may be a vertex, an edge point, or a surface point. If $\mathbf{y}_i$ is a vertex or an edge point, there is ambiguity in defining its normal vector, which introduces difficulty in the subsequent point relocation process. To address this, we introduce a small perturbation to $\mathbf{y}_i$ while still restricting it to the original mesh surface.

As a result, we obtain a point set $\{\mathbf{y}_i\}_{i=1}^N$ lying on the surface $\mathcal{M}$, with each point located in the interior of a triangle. On the one hand, we treat these points as movable but constrain their movement to $\mathcal{M}$. On the other hand, we use the wrap surface $\mathcal{S}$ to define the connections between these points, even though they do not lie on $\mathcal{S}$.

3.2 Point Relocation

Neighboring Relationship. Since the points $\{\mathbf{y}_i\}_{i=1}^N$ reside on the defective surface $\mathcal{M}$, defining valid connections between them is challenging. Interestingly, this can be efficiently addressed with the help of the wrap surface $\mathcal{S}$.

Recall that the point set $\{\mathbf{x}_i\}_{i=1}^N$ defines the restricted Voronoi Diagram (RVD) of the wrap surface $\mathcal{S}$, which offers a simple solution: $\mathbf{y}_i$ is connected to $\mathbf{y}_j$ if and only if $\mathbf{x}_i$ is connected to $\mathbf{x}_j$.

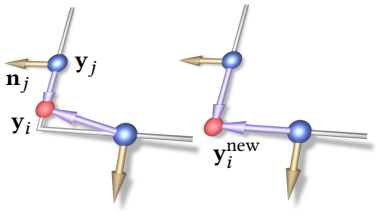
Point Displacement. Based on the above discussion, we can determine the neighboring relationships between $\{\mathbf{y}_i\}_{i=1}^N$. At this stage, we optimize each point in $\{\mathbf{y}_i\}_{i=1}^N$ independently. Specifically, the

new position $\mathbf{y}_i^{\text{new}}$ depends only on the previous locations $\{\mathbf{y}_i\}_{i=1}^N$, with no dependence between $\mathbf{y}_i^{\text{new}}$ and $\mathbf{y}_j^{\text{new}}$. The point displacement strategy for $\mathbf{y}_i$ follows the spirit of QEM (Quadric Error Metrics) [Garland and Heckbert 1997]:

$$\mathbf{y}_i^{\text{new}} = \arg\min_{\mathbf{y}} \sum_{\mathbf{y}_j \in N(i)} \Big((\mathbf{y} - \mathbf{y}_j) \cdot \mathbf{n}_j \Big)^2 + \mu \|\mathbf{y} - \mathbf{y}_i\|^2, \quad (1)$$

where $N(i)$ represents $\mathbf{y}_i$'s 1-ring neighboring points, $\mathbf{n}_i$ denotes the unit normal vector at $\mathbf{y}_i$ (corresponding to the normal of the facet of $\mathcal{M}$ containing $\mathbf{y}_i$), and μ is set to 0.01 in our experiments.

Note that in Eq. (1), the first term drives the point $\mathbf{y}$ to align with the normals of its neighboring points. When the neighboring points are coplanar, this term vanishes, so the optimal position of $\mathbf{y}$ is unchanged. Conversely, if the neighbors are not coplanar, $\mathbf{y}$ is pulled toward the local "corner" formed by their normals. In the 2D inset, for example, two neighboring points lie on different lines; the first term is minimized only when $\mathbf{y}$ moves to their intersection. The second term simultaneously penalizes large displacements, preventing overshooting.



We justify the effectiveness of this strategy as follows. First, even if a triangle in $\mathcal{M}$ is flipped, the quadratic error remains unaffected, so the updated position $\mathbf{y}_i^{\text{new}}$ is still well defined. Second, although the same optimization is applied to every sample in $\{\mathbf{y}_i\}_{i=1}^N$, its influence is location dependent: points far from feature lines remain almost stationary, whereas points near feature lines are naturally attracted toward them. In Fig. 4, for instance, two walls intersect and the original mesh completely loses the correct topology around the intersection curve. Instead of explicitly detecting this curve, the point-shifting procedure automatically drags nearby samples onto it. Moreover, any redundant interior geometry is ignored, because no sample point on the wrap surface $\mathcal{S}$ can be projected onto such spurious elements.

To summarize, the main operation in this step is solving a quadratic optimization problem, equivalent to solving a linear system, to compute the displacement of each point. This process is applied indiscriminately to each point in $\{\mathbf{y}_i\}_{i=1}^N$ only once. At the end of this process, each point $\mathbf{y}_i$ obtains a displacement:

$$(\delta\mathbf{y})_i = \mathbf{y}_i^{\text{new}} - \mathbf{y}_i.$$

3.3 Surface Extraction with RPD

The final stage of the algorithm involves extracting a manifold and watertight repaired surface using the optimized points $\mathbf{y}_i^{\text{new}}$. For this task, the restricted Voronoi diagram (RVD) [Edelsbrunner and Shah 1994; Yan et al. 2009] is a commonly used tool in mesh generation. However, the RVD treats each site with equal importance, and as a result, the resulting triangulation may not align well with potential feature lines if the edge points are not sufficiently dense. In contrast, the restricted Power diagram (RPD) [Basselin et al. 2021] allows for a more flexible weight setting by assigning greater influence to edge points.

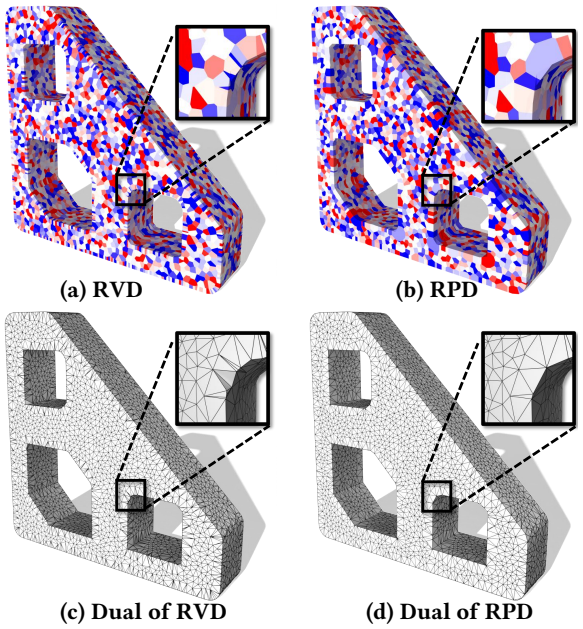


Fig. 5. RPD exhibits superior feature line preservation compared to RVD.

In our approach, only points situated near potential features exhibit significant displacement, while all other points remain nearly unchanged. Consequently, we assign the squared distance $\|(\delta\mathbf{y})_i\|^2$ to the site $\mathbf{y}_i^{\text{new}}$ during the RPD calculation. Its dual structure yields the final repaired triangle mesh. As shown in Fig. 5, the weighting strategy effectively preserves the feature-line alignment property.

We explain the weight setting as follows. First, it can be observed that when $(\delta\mathbf{y})_i$ does not vanish, it is very likely to reside near a feature line, which implies that we must increase the influence of $\mathbf{y}_i^{\text{new}}$. Second, consider two sites $\mathbf{y}_1$ and $\mathbf{y}_2$, which are projected to nearly the same position on the feature line, i.e., $\mathbf{y}_1^{\text{new}}$ and $\mathbf{y}_2^{\text{new}}$ are close to each other. In this case,

$$\left| \|(\delta\mathbf{y})_1\|^2 - \|(\delta\mathbf{y})_2\|^2 \right| \gg \|\mathbf{y}_1^{\text{new}} - \mathbf{y}_2^{\text{new}}\|^2,$$

making one of them a hidden site based on the principle of the power diagram. Furthermore, this approach works for both convex and concave feature lines.

At first glance, it may seem feasible to extract a sufficiently dense set of samples from the defective mesh and then perform surface reconstruction to obtain a repaired outcome. However, this does not work in practice. Existing surface reconstruction methods fall into two main categories: implicit approaches, which infer a scalar field whose zero level set approximates the geometry, and explicit approaches [Fu et al. 2024; Salman et al. 2010; Ye et al. 2024; Zhao et al. 2023], which directly infer meshing topology from the point set. Since mesh repair requires fully respecting sample positions, implicit approaches are ruled out. As for explicit methods, even with extra points inserted along feature lines, the reconstructed mesh may still fail to recover the correct topology, because the

Table 1. The statistics of imperfections for the defective meshes corresponding to the repaired models in Fig. 6.

Model ID	# OB	# NMV	# NME	# SI	# DF
1	8	1	1	6	0
2	3	0	0	35	7
3	2274	0	0	1	16
4	0	5	2	11	0
5	1	0	2	18	0
6	43260	0	0	91	0
7	14586	10	2	54	9
8	24	0	0	0	0
9	8	1	1	5	0
10	3	3	1	21	2

reconstruction process operates independently of the original mesh structure. Moreover, if the defective mesh contains self-intersections, applying explicit reconstruction would require detecting all self-intersections beforehand—a computationally expensive process.

To overcome these challenges, we leverage a wrap surface. On one hand, the wrap surface is computed with reference to the defective mesh, allowing approximate recovery of the correct geometry, albeit with slight compromises in precision and feature preservation. The RPD of the wrap surface ensures the output mesh is manifold, with vertices either exactly sampled from the original surface or newly predicted along feature lines. On the other hand, Eq. (1) naturally attracts points near self-intersection lines to their correct positions.

4 Experiments

4.1 Experimental Setting

We implemented our algorithm in C++ on a desktop computer equipped with an Intel Core i9-13900K CPU and 64 GB of RAM. The implementation utilizes Eigen [Guennebaud et al. 2010] for linear algebra routines and Libigl [Jacobson et al. 2018] for fundamental geometry processing tasks. To optimize the energy in Eq. (1), we employed Ceres-Solver [Agarwal et al. 2023], while the computation of the RPD followed the approach outlined in [Xiao et al. 2023].

4.2 Validation

To evaluate the robustness of our algorithm, we conducted experiments on man-made datasets that inherently contain various defects, such as ABC [Koch et al. 2019] and Thingi10K [Zhou and Jacobson 2016]. To further assess the feature-preservation capability of our method, we manually introduced additional defects, detailed in Section 1, to selected models from these datasets. For the evaluation, we selected 2K defective meshes. Our method demonstrated strong robustness, with only 17 failure cases: 5 due to wrap surface failure (e.g., Model #93366 from Thingi10K), and 12 due to insufficient point sampling, which led to missing or mixed feature lines. Notably, the latter can be mitigated by increasing the sampling density. In addition, we selected 10 representative repaired models for visualization, as shown in Fig. 6. Each corresponding imperfect mesh exhibits multiple types of defects, including open boundaries (OB), non-manifold vertices and edges (NMV/NME), self-intersections (SI), and duplicate facets (DF), as detailed in Table 1.

4.3 Comparison Methods

To demonstrate the superiority of our mesh repair method in preserving feature lines, we compare our algorithm against the following state-of-the-art methods[1]: TetWild (TW) [Hu et al. 2018] and VolumeMesher (VM) [Diazzi and Attene 2021] use BSP-based gap closure but are highly sensitive to input orientations, producing inconsistent results when faced with misorientations or nested structures. VisualRepair (VisR) [Chu et al. 2019] utilizes visual guidance to segment surfaces, which can lead to gaps and non-watertight patches. MeshFix (MF) [Attene 2010] addresses local inconsistencies but lacks the capability to preserve the global shape. Biharmonic Repair (BR) [Argudo et al. 2015] approximates SDFs using bi-harmonic fields, but fails to maintain sharp features. ManifoldPlus (MP) [Huang et al. 2020] and Alpha Wrapping (AW) [Portaneri et al. 2022] construct watertight enclosures via displacement but compromise feature integrity, with MP introducing face duplication and self-intersections.

4.4 Comparisons on Imperfect Meshes

We evaluated the quality of the repaired mesh using several metrics, including OB, NMV/NME, SI, normal consistency (NC), and the triangle quality (TQ). To quantify the difference between the repaired mesh and the original imperfect mesh, we employed four indicators: Chamfer Distance (CD), F-score (F1), Normal Consistency (NC), and Hausdorff Distance (HD). Furthermore, for CAD models, we utilized Edge Chamfer Distance (ECD) and Edge F-score (EF1) proposed by NMC [Chen and Zhang 2021] to assess the sharpness of the repaired mesh and its ability to preserve sharp features.

Qualitative comparisons are presented in Fig. 7, with corresponding quantitative metrics summarized in Table 2 and computational time provided in Table 3. In the first two rows of Fig. 7, defects such as non-manifold vertices (highlighted in purple), narrow gaps (bounded by green curves), and self-intersections (in red) near feature lines remain unrepaired by methods like TW, MP, and VM. Although other methods manage to address these imperfections, they fail to preserve the original features accurately. Additionally, some models in the Thingi10K dataset are fragmented into multiple discrete triangles, a defect that neither MF nor BR can process effectively. In cases where the imperfect models contain isolated small components, only TW and our method successfully handle these challenges, as shown in the third row of Fig. 7.

To assess the robustness of our methodology, 20% of the triangles within a triangle mesh were randomly selected and deliberately fragmented into several smaller discrete triangles. Subsequently, these fragments were subjected to rotations around their normal vectors at the barycenters by a random angle, intentionally inducing various defects such as self-intersections, folded faces, and gaps, as depicted in Fig. 8(a). In Fig. 8, we conduct a comparative analysis of our approach with alternative methods capable of generating output meshes for visualization. TW was unable to produce the expected

[1]While [Zheng et al. 2024] reports the ability to handle such models, no open-source code has been provided for verification.

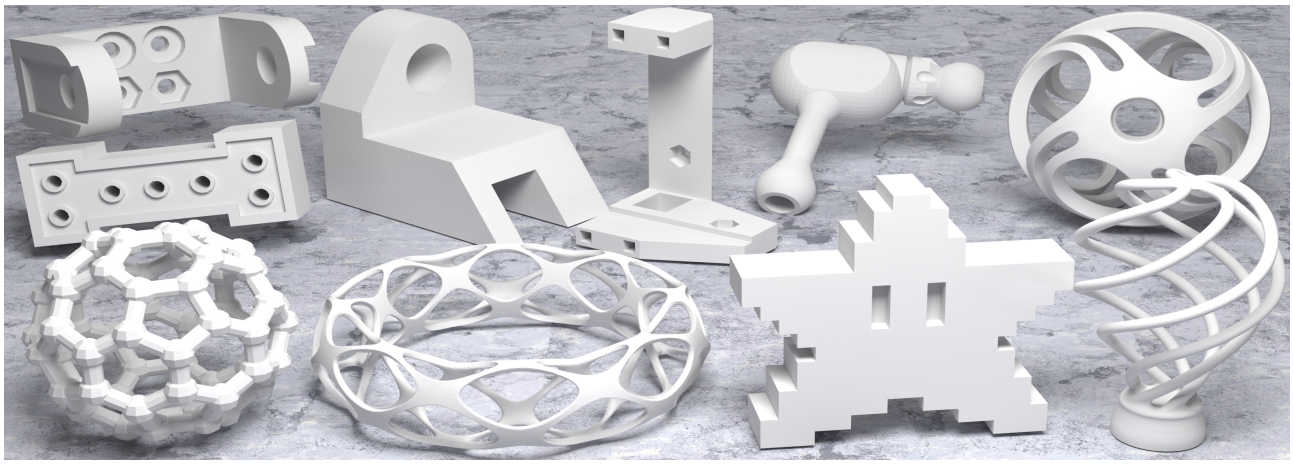


Fig. 6. A gallery of repair results by our algorithm.

Table 2. Statistics of comparisons for the models in Fig. 7. In each column, the **best** scores are emphasized in bold with under lining, while the **second best** scores are highlighted only in bold.

Methods	# OB ↓	# NMV ↓	# NME ↓	# SI ↓	TQ ↑	NC ↑	CD ($\times 10^4$) ↓	F1 ↑	HD ($\times 10^2$) ↓	ECD ($\times 10^2$) ↓	EF1 ↑
MF	**0**	**0**	**0**	**0**	0.382	0.914	43.635	0.863	19.712	2.598	0.308
BR	**1**	**0**	**0**	111	0.654	0.979	0.209	0.921	1.906	**0.745**	0.432
TW	**0**	7	**18**	5	0.716	0.897	4.649	0.814	7.074	2.995	0.231
VisR	20	**1**	**0**	**2**	0.414	0.976	0.176	**0.949**	**1.004**	**0.819**	**0.519**
MP	**0**	**0**	**0**	426	0.613	0.982	**0.161**	0.850	1.313	1.573	0.366
VM	**0**	69	219	59	0.388	**0.989**	0.194	0.913	4.591	2.167	0.389
AW	**0**	**0**	**0**	**0**	**0.765**	0.976	0.177	0.890	2.024	3.298	0.297
Ours	**0**	**0**	**0**	**0**	**0.778**	**0.993**	**0.172**	**0.927**	**1.229**	0.930	**0.485**

Table 3. Comparative execution time (seconds) of the evaluated methods across models in Fig. 7.

MF	**BR**	**TW**	**VisR**	**MP**	**VM**	**AW**	**Ours**
0.047	8.270	14.048	3.208	3.676	0.102	26.892	39.141
0.035	4.375	18.669	2.562	0.906	0.094	15.648	26.608
0.896	139.678	78.236	46.926	8.812	4.547	42.544	54.409

meshes, whereas MP successfully generated meshes but introduced duplicate triangles, clearly observable as black regions in the figure. AW generated high-quality triangulations; however, it was unable to maintain sharp geometric features.

Furthermore, these methodologies were evaluated on an imperfect mesh characterized by a missing feature line and surrounding open boundaries, as illustrated in Fig. 9. Among all the methods, ours was uniquely capable of recovering the sharp feature.

4.5 Ablation Study

Wrap surface vs. Poisson surface. In the first stage of our algorithm, we utilize a manifold wrap surface as a mediating agent for the imperfect mesh. However, the Poisson surface, as proposed in [Xu et al. 2022], is another commonly used proxy. To evaluate the differences, we implemented the Poisson surface as an alternative proxy and conducted comparisons with the wrap surface while keeping all other experimental settings consistent. Since the Poisson surface requires consistent normals as input, we first reorient all faces coherently to ensure fairness in comparison. Subsequently, 10K points were sampled on the mesh surface to reconstruct the Poisson surface. However, due to the Poisson surface being constructed based on the entire set of sampled points, including those from internal structures, it may result in the formation of holes or self-intersections, as illustrated in Fig. 10.

Influence of Parameters. The wrap distance ϵ governs the proximity between the wrap surface and the initial mesh, and it determines the granularity of detail captured. Within our algorithm, both Voronoi and power diagrams are derived based on the wrap surface, thereby its level of detail has a direct impact on the accuracy and feature retention in the repaired surface. As ϵ increases, there may be a reduction in the inherent details of the original mesh in the repaired configuration, as depicted in Fig. 11. Consequently, we assign a lower default value of $1/3000$ to ϵ.

Within our algorithm, the parameter μ in Eq. (1) is pivotal in regulating point movements towards potential feature lines, thereby ensuring geometric integrity throughout the repair process. A lower

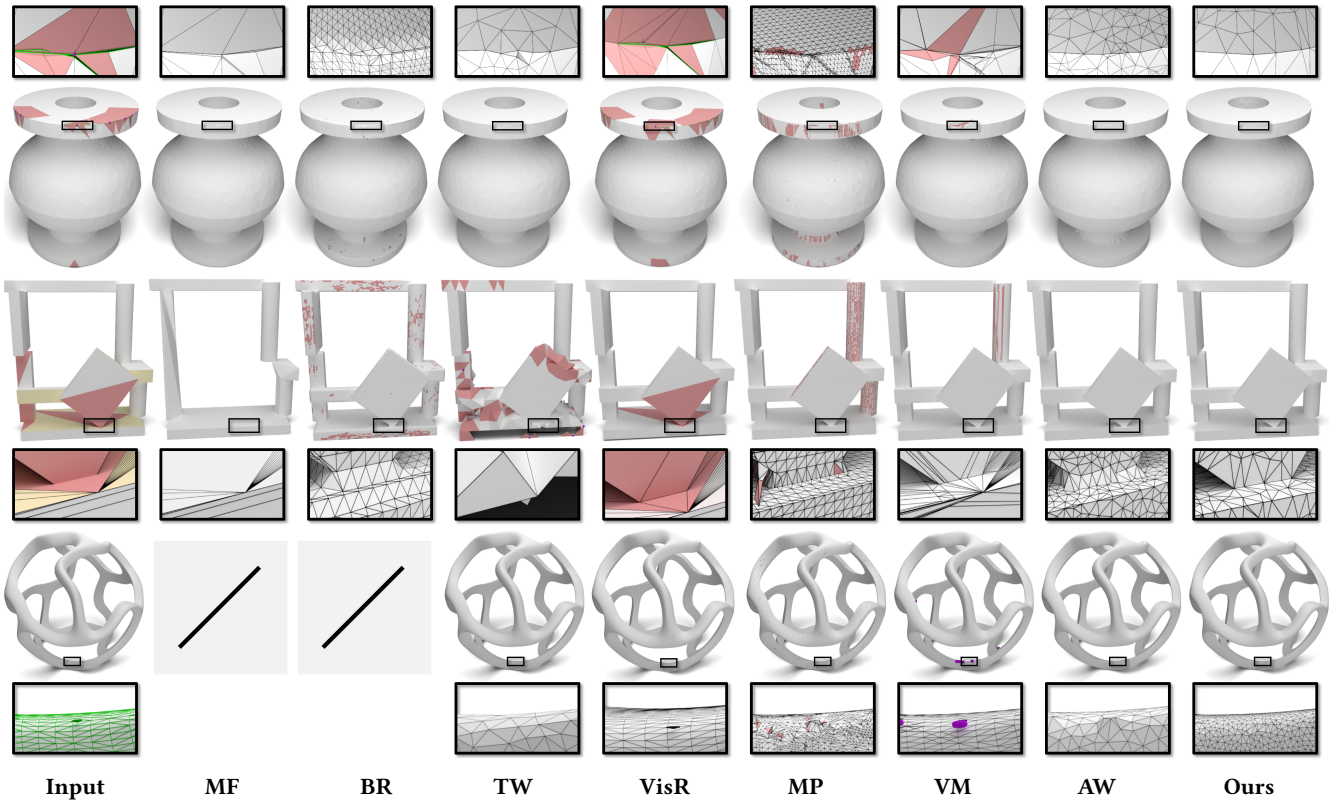


Fig. 7. Comparison with state-of-the-art methods on CAD and organic model with multiple defects. Our method effectively repairs the mesh while preserving geometric features.

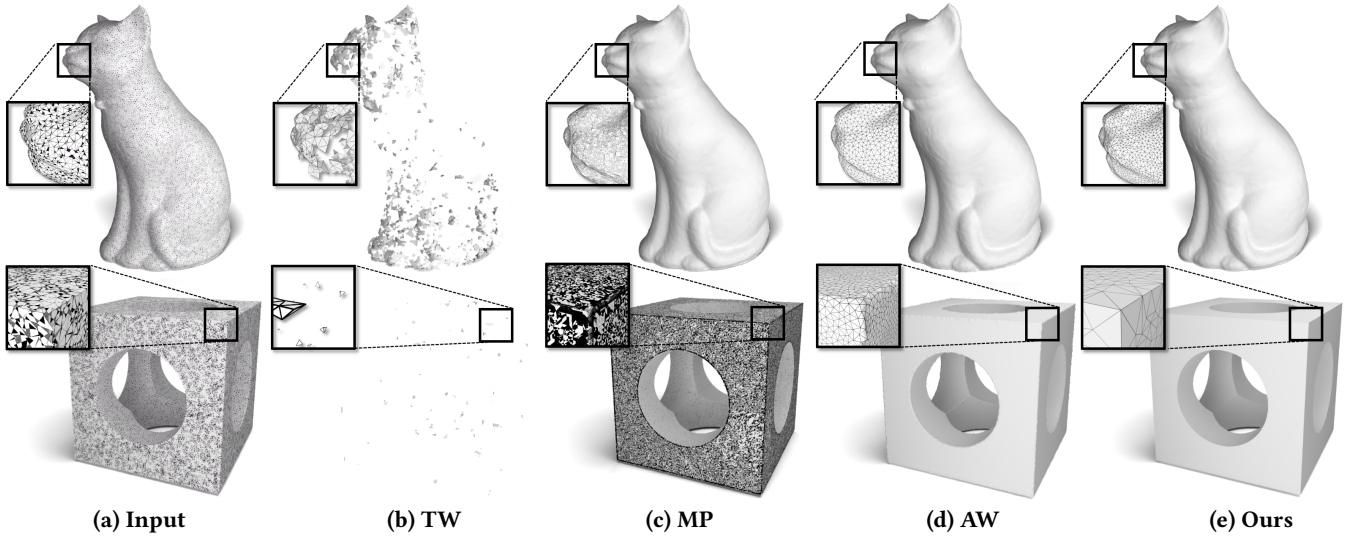


Fig. 8. Comparison with contemporary techniques on organic and CAD models consisting of disordered triangles, which result in multiple defects, demonstrates the efficacy of our method in mesh repair.

value of μ could cause considerable drifting of points from the imperfect mesh, leading to a misalignment with the original geometry. In contrast, a higher value could excessively constrain point adjustments, thus failing to adequately maintain sharp features and

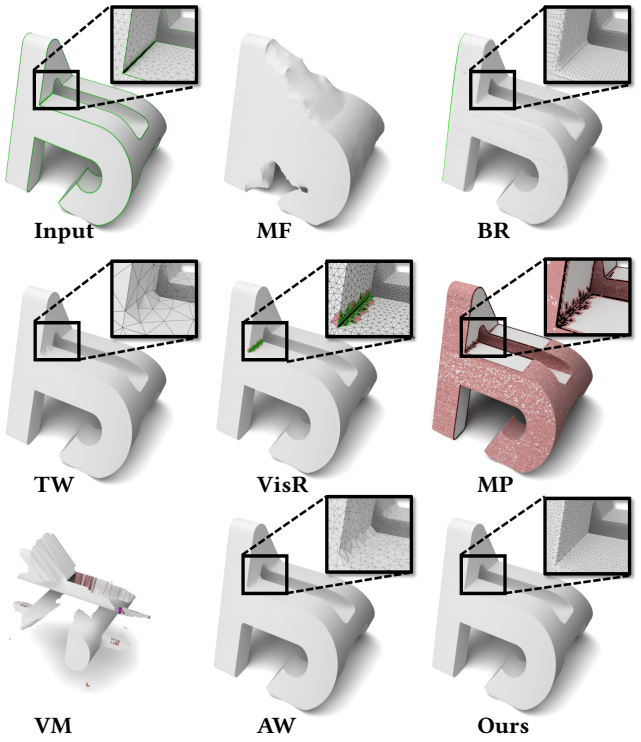


Fig. 9. Comparison with state-of-the-art methods on an imperfect mesh with a missing feature line shows that our method effectively recovers the feature line and fills the gap.

intricate details. This equilibrium is illustrated in Fig. 12, which elucidates the trade-offs arising from varying values of μ.

5 Limitations

Nonetheless, it is essential to recognize that our methodology does have certain limitations:

(1) As depicted in Fig. 13, the present implementation exhibits inefficiencies, particularly when the number of sample points increases.
(2) The proposed RPD strategy, while simple and effective in many cases, may fail when the number of sampled points is too small. In such situations, the dual structure of the RPD struggles to accurately capture feature lines, especially when two feature lines are in close proximity.
(3) Our method is currently limited in its ability to handle meshes with large holes or near-zero volumes. Addressing such cases remains an open challenge for future work.

6 Conclusion

In this paper, we propose a unified methodology for repairing defective polygonal meshes while preserving geometric features. Our algorithm operates on a pair of objects—the original surface and a wrap surface—thereby decoupling the challenges of edge connectivity repair and point relocation during the restoration process.

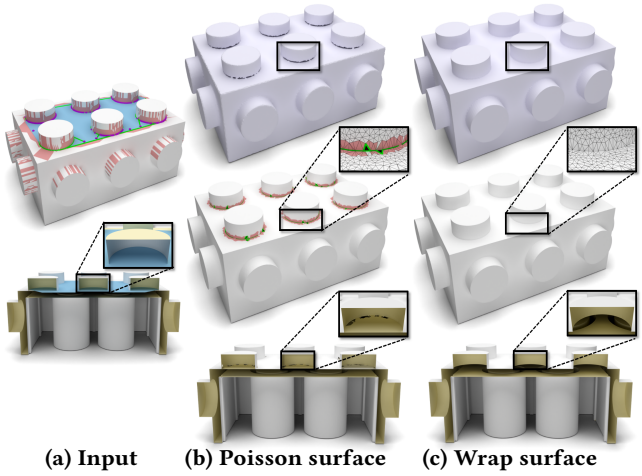


Fig. 10. Ablation study on proxy surfaces: The input imperfect mesh exhibits various defects, including self-intersections, duplicate facets, open boundaries, and non-manifold vertices. While our method effectively repairs all these issues, the Poisson surface-based approach fails to address self-intersections and open boundaries. In the last two columns, the top row shows the proxy surfaces, the middle row showcases the repaired meshes, and the bottom row provides the slice viewer of the repaired meshes for detailed comparison.

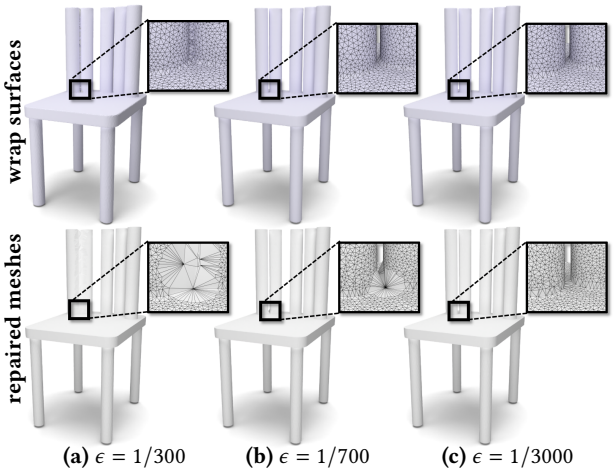


Fig. 11. Ablation study on the parameter ϵ: A larger value may eliminate the intrinsic details of the original mesh.

We employ a QEM-inspired technique to define a geometry-aware point-shifting strategy. Points distant from feature lines or corners remain stationary, while those near these features shift toward potential feature lines or corners. By using the squared displacement length as the weight in the final power diagram, our method consistently generates a high-quality triangle mesh as the repaired result.

Future works include three key directions: (1) enhancing runtime performance through parallel processing, (2) improving the

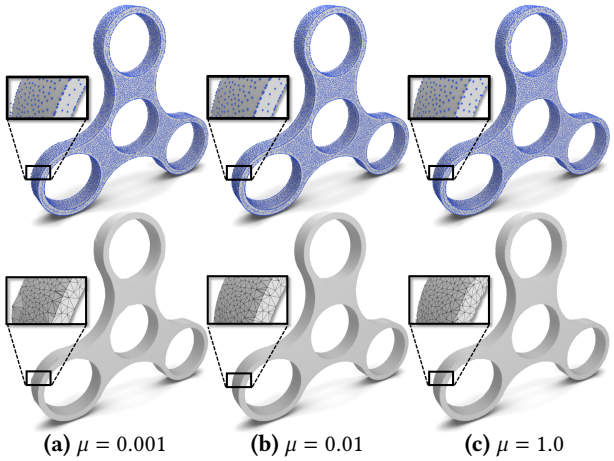


Fig. 12. Ablation study on the parameter μ: A smaller value may cause the points to drift away from the imperfect mesh, whereas a larger value might fail to preserve sharp features.

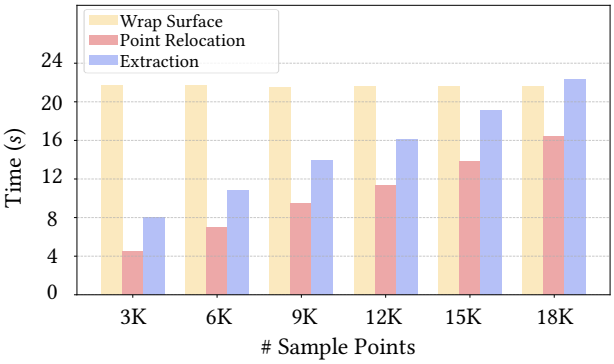


Fig. 13. The time consumption for each stage of our algorithm applied to an imperfect model with varying numbers of sample points.

RPD strategy to address sparse sampling scenarios, and (3) conducting comprehensive testing on highly complex defective models to further strengthen robustness.

Acknowledgments

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions. This work was supported by the National Key R&D Program of China (2022YFB3303200), the National Natural Science Foundation of China (U23A20312, 62272277, 62272408, 62172356), and the Zhejiang Provincial Natural Science Foundation of China (LZ25F020012). It was also partially funded by the Research Grants Council of Hong Kong (Ref: 17210222), and by the Innovation and Technology Commission of the HKSAR Government under the ITSP-Platform grant (Ref: ITS/335/23FP) and the InnoHK initiative (TransGP project). Part of the research was conducted in the JC STEM Lab of Robotics for Soft Materials, funded by The Hong Kong Jockey Club Charities Trust.

References

Sameer Agarwal, Keir Mierle, and The Ceres Solver Team. 2023. *Ceres Solver.* https://github.com/ceres-solver/ceres-solver

Carlos Andújar, Pere Brunet, and Dolors Ayala. 2002. Topology-reducing surface simplification using a discrete solid representation. *ACM Trans. Graph.* 21, 2 (April 2002), 88–105.

Oscar Argudo, Pere Brunet, Antoni Chica, and Àlvar Vinacua. 2015. Biharmonic fields and mesh completion. *Graphical Models* 82 (2015), 137–148.

Marco Attene. 2010. A lightweight approach to repairing digitized polygon meshes. *The visual computer* 26 (2010), 1393–1406.

Marco Attene. 2014. Direct repair of self-intersecting meshes. *Graphical Models* 76, 6 (2014), 658–668.

Marco Attene. 2017. ImatiSTL-fast and reliable mesh processing with a hybrid kernel. *Transactions on Computational Science XXIX* (2017), 86–96.

Marco Attene, Marcel Campen, and Leif Kobbelt. 2013. Polygon mesh repairing: An application perspective. *ACM Comput. Surv.* 45, 2, Article 15 (March 2013), 33 pages.

J. Basselin, L. Alonso, N. Ray, D. Sokolov, S. Lefebvre, and B. Lévy. 2021. Restricted Power Diagrams on the GPU. *Computer Graphics Forum* 40, 2 (2021), 1–12.

Stephan Bischoff, Darko Pavic, and Leif Kobbelt. 2005. Automatic restoration of polygon models. *ACM Trans. Graph.* 24, 4 (Oct. 2005), 1332–1352.

Zhiqin Chen and Hao Zhang. 2021. Neural marching cubes. *ACM Trans. Graph.* 40, 6, Article 251 (Dec. 2021), 15 pages.

Lei Chu, Hao Pan, Yang Liu, and Wenping Wang. 2019. Repairing man-made meshes via visual driven global optimization with minimum intrusion. *ACM Trans. Graph.* 38, 6, Article 158 (Nov. 2019), 18 pages.

Massimiliano Corsini, Paolo Cignoni, and Roberto Scopigno. 2012. Efficient and Flexible Sampling with Blue Noise Properties of Triangular Meshes. *IEEE Transactions on Visualization and Computer Graphics* 18, 6 (2012), 914–924.

Brian Curless and Marc Levoy. 1996. A volumetric method for building complex models from range images. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '96).* Association for Computing Machinery, New York, NY, USA, 303–312.

Lorenzo Diazzi and Marco Attene. 2021. Convex polyhedral meshing for robust solid modeling. *ACM Trans. Graph.* 40, 6, Article 259 (Dec. 2021), 16 pages.

Herbert Edelsbrunner and Nimish R. Shah. 1994. Triangulating topological spaces. In *Proceedings of the Tenth Annual Symposium on Computational Geometry* (Stony Brook, New York, USA) *(SCG '94).* Association for Computing Machinery, New York, NY, USA, 285–292.

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.

Rao Fu, Kai Hormann, and Pierre Alliez. 2024. LFS-Aware Surface Reconstruction From Unoriented 3D Point Clouds. *IEEE Transactions on Multimedia* 26 (2024), 11415–11427.

Ryo Furukawa, Tomoya Itano, Akihiko Morisaka, and Hiroshi Kawasaki. 2007. Improved space carving method for merging and interpolating multiple range images using information of light sources of active stereo. In *Proceedings of the 8th Asian Conference on Computer Vision - Volume Part II* (Tokyo, Japan) *(ACCV'07).* Springer-Verlag, Berlin, Heidelberg, 206–216.

Michael Garland and Paul S. Heckbert. 1997. Surface Simplification Using Quadric Error Metrics. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '97).* ACM Press/Addison-Wesley Publishing Co., USA, 209–216.

Gaël Guennebaud, Benoît Jacob, et al. 2010. Eigen: A C++ Template Library for Linear Algebra: Matrices, Vectors, Numerical Solvers, and Related Algorithms. http://eigen.tuxfamily.org.

Jia-Peng Guo and Xiao-Ming Fu. 2024. Exact and Efficient Intersection Resolution for Mesh Arrangements. *ACM Trans. Graph.* 43, 6, Article 165 (Nov. 2024), 14 pages.

Alexander Hornung and Leif Kobbelt. 2006. Robust reconstruction of watertight 3D models from non-uniformly sampled point clouds without normal information. In *Proceedings of the Fourth Eurographics Symposium on Geometry Processing* (Cagliari, Sardinia, Italy) *(SGP '06).* Eurographics Association, Goslar, DEU, 41–50.

Yixin Hu, Qingnan Zhou, Xifeng Gao, Alec Jacobson, Denis Zorin, and Daniele Panozzo. 2018. Tetrahedral Meshing in the Wild. *ACM Trans. Graph.* 37, 4, Article 60 (July 2018), 14 pages.

Jingwei Huang, Yichao Zhou, and Leonidas Guibas. 2020. ManifoldPlus: A Robust and Scalable Watertight Manifold Surface Generation Method for Triangle Soups. *arXiv preprint arXiv:2005.11621* (2020).

Franck Hétroy, Stéphanie Rey, Carlos Andújar, Pere Brunet, and Àlvar Vinacua. 2011. Mesh repair with user-friendly topology control. *Computer-Aided Design* 43, 1 (2011), 101–113.

Alec Jacobson, Daniele Panozzo, et al. 2018. Libigl: A Simple C++ Geometry Processing Library. https://libigl.github.io/.

Tao Ju. 2004. Robust repair of polygonal models. *ACM Trans. Graph.* 23, 3 (Aug. 2004), 888–895.

Tao Ju. 2009. Fixing geometric errors on polygonal models: a survey. *Journal of Computer Science and Technology* 24, 1 (2009), 19–29.

Tao Ju, Frank Losasso, Scott Schaefer, and Joe Warren. 2002. Dual contouring of hermite data. *ACM Trans. Graph.* 21, 3 (July 2002), 339–346.

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 *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

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

Xiaohan Liu, Lei Ma, Jianwei Guo, and Dong-Ming Yan. 2022. Parallel Computation of 3D Clipped Voronoi Diagrams. *IEEE Transactions on Visualization and Computer Graphics* 28, 2 (Feb. 2022), 1363–1372.

S.R. Marschner, Matt Garr, and Marc Levoy. 2002. Filling Holes In Complex Surfaces Using Volumetric Diffusion. 428 – 441.

Wenlong Meng, Pengbo Bo, Xiaodong Zhang, Jixiang Hong, Shiqing Xin, and Changhe Tu. 2023. An efficient algorithm for approximate Voronoi diagram construction on triangulated surfaces. *Computational Visual Media* 9, 3 (01 Sep 2023), 443–459.

Fakir S. Nooruddin and Greg Turk. 2003. Simplification and Repair of Polygonal Models Using Volumetric Techniques. *IEEE Transactions on Visualization and Computer Graphics* 9, 2 (April 2003), 191–205.

Stijn Oomes, Peter Snoeren, and Tjeerd Dijkstra. 1997. 3D Shape Representation: Transforming Polygons into Voxels. In *Proceedings of the First International Conference on Scale-Space Theory in Computer Vision (SCALE-SPACE '97)*. Springer-Verlag, Berlin, Heidelberg, 349–352.

Cédric Portaneri, Mael Rouxel-Labbé, Michael Hemmer, David Cohen-Steiner, and Pierre Alliez. 2022. Alpha Wrapping with an Offset. *ACM Transactions on Graphics* 41, 4 (June 2022), 1–22. https://inria.hal.science/hal-03688637

Nicolas Ray, Dmitry Sokolov, Sylvain Lefebvre, and Bruno Lévy. 2018. Meshless voronoi on the GPU. *ACM Trans. Graph.* 37, 6, Article 265 (Dec. 2018), 12 pages.

Nader Salman, Mariette Yvinec, and Quentin Merigot. 2010. Feature Preserving Mesh Generation from 3D Point Clouds. *Computer Graphics Forum* 29, 5 (2010), 1623–1632.

Greg Turk and Marc Levoy. 1994. Zippered polygon meshes from range images. In *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '94)*. Association for Computing Machinery, New York, NY, USA, 311–318.

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 (Nov. 2024), 16 pages.

Ningna Wang, Bin Wang, Wenping Wang, and Xiaohu Guo. 2022. Computing Medial Axis Transform with Feature Preservation via Restricted Power Diagram. *ACM Trans. Graph.* 41, 6, Article 188 (Nov. 2022), 18 pages.

Pengfei Wang, Shiqing Xin, Changhe Tu, Dongming Yan, Yuanfeng Zhou, and Caiming Zhang. 2020. Robustly computing restricted Voronoi diagrams (RVD) on thin-plate models. *Computer Aided Geometric Design* 79 (2020), 101848.

Yanyang Xiao, Juan Cao, Shaoping Xu, and Zhonggui Chen. 2023. Meshless Power Diagrams. *Computers & Graphics* 114 (2023), 247–256.

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).

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 Trans. Graph.* 41, 6, Article 228 (Nov. 2022), 15 pages.

Dong-Ming Yan, Wenping Wang, Bruno Levy, and Yang Liu. 2010. Efficient Computation of 3D Clipped Voronoi Diagram. *Adv. Geometric Modeling Processing* 6130, 269–282.

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 (2009), 1445–1454.

Yuanyan Ye, Yubo Wang, Juan Cao, and Zhonggui Chen. 2024. Watertight surface reconstruction method for CAD models based on optimal transport. *Computational Visual Media* 10, 5 (01 Oct 2024), 859–872.

Tong Zhao, Laurent Busé, David Cohen-Steiner, Tamy Boubekeur, Jean-Marc Thiery, and Pierre Alliez. 2023. Variational Shape Reconstruction via Quadric Error Metrics. In *ACM SIGGRAPH 2023 Conference Proceedings* (Los Angeles, CA, USA) *(SIGGRAPH '23)*. Association for Computing Machinery, New York, NY, USA, Article 45, 10 pages.

Wei Zhao, Shuming Gao, and Hongwei Lin. 2007. A robust hole-filling algorithm for triangular mesh. *Vis. Comput.* 23, 12 (Nov. 2007), 987–997.

Zhongtian Zheng, Xifeng Gao, Zherong Pan, Wei Li, Peng-Shuai Wang, Guoping Wang, and Kui Wu. 2024. Visual-Preserving Mesh Repair. *IEEE Transactions on Visualization and Computer Graphics* 30, 9 (2024), 6586–6597.

Qingnan Zhou and Alec Jacobson. 2016. Thingi10K: A Dataset of 10,000 3D-Printing Models. *arXiv preprint arXiv:1605.04797* (2016).

Chen Zong, Pengfei Wang, Dong-Ming Yan, Shuangmin Chen, Shiqing Xin, Changhe Tu, and Qiang Hu. 2023. Parallel Post-processing of Restricted Voronoi Diagram on Thin Sheet Models. *Computer-Aided Design* 159 (2023), 103511.