BallMerge

High-quality Fast Surface Reconstruction via Voronoi Balls

Parakkat, Amal Dev; Ohrhallinger, Stefan; Eisemann, Elmar; Memari, Pooran

DOI
10.1111/cgf.15019

Publication date
2024

Document Version
Final published version

Published in
Computer Graphics Forum

Citation (APA)
Parakkat, A. D., Ohrhallinger, S., Eisemann, E., & Memari, P. (2024). BallMerge: High-quality Fast Surface Reconstruction via Voronoi Balls. *Computer Graphics Forum*, *43*(2), Article e15019. https://doi.org/10.1111/cgf.15019

DOI: 10.1111/cgf.15019
EUROGRAPHICS 2024 / A. Bermano and E. Kalogerakis
(Guest Editors)

COMPUTER GRAPHICS *forum*
Volume 43 (2024), Number 2

BallMerge: High-quality Fast Surface Reconstruction via Voronoi Balls

Amal Dev Parakkat[1], Stefan Ohrhallinger[2], Elmar Eisemann[3], Pooran Memari[4]

[1] LTCI - Telecom Paris, IP Paris, France, [2]TU Wien, Austria, [3] TU Delft, Netherlands, [4] CNRS, LIX - INRIA, Ecole Polytechnique, IP Paris, France

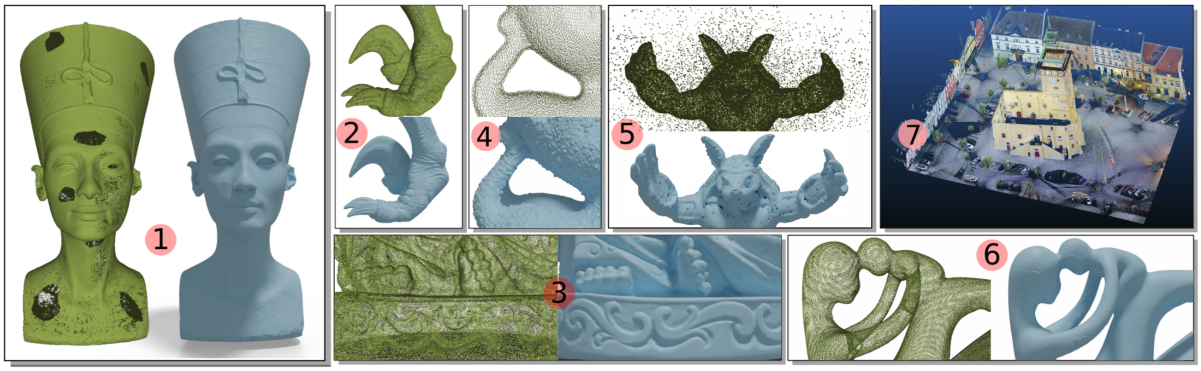


Figure 1: *Reconstruction of a variety of 3D models with different properties or artifacts, namely synthetic holes (1), sharp and subtle features (2,3), noise (4), outliers (5), genus> 0 (6) and challenging real-world scans (7). All results (except the last) used* GLOBAL BALLMERGE. *It can be used parameter-free (1,2,3,6) or manually adjusted to handle noise and outliers (4 and 5) via a single easily tunable parameter. The last example (7) uses* LOCAL BALLMERGE*, a variant carefully designed for challenging real-world scans (rendered with textures).*

Abstract

We introduce a Delaunay-based algorithm for reconstructing the underlying surface of a given set of unstructured points in 3D. The implementation is very simple, and it is designed to work in a parameter-free manner. The solution builds upon the fact that in the continuous case, a closed surface separates the set of maximal empty balls (medial balls) into an interior and exterior. Based on discrete input samples, our reconstructed surface consists of the interface between Voronoi balls, which approximate the interior and exterior medial balls. An initial set of Voronoi balls is iteratively processed, merging Voronoi-ball pairs if they fulfil an overlapping error criterion. Our complete open-source reconstruction pipeline performs up to two quick linear-time passes on the Delaunay complex to output the surface, making it an order of magnitude faster than the state of the art while being competitive in memory usage and often superior in quality. We propose two variants (local and global), which are carefully designed to target two different reconstruction scenarios for watertight surfaces from accurate or noisy samples, as well as real-world scanned data sets, exhibiting noise, outliers, and large areas of missing data. The results of the global variant are, by definition, watertight, suitable for numerical analysis and various applications (e.g., 3D printing). Compared to classical Delaunay-based reconstruction techniques, our method is highly stable and robust to noise and outliers, evidenced via various experiments, including on real-world data with challenges such as scan shadows, outliers, and noise, even without additional preprocessing.

1. Introduction

Reconstructing high-quality 3D meshes from unstructured point sets is a classic problem in Computer Graphics and Computational Geometry with applications in many domains, such as CAD, medical imaging, or visualization. Yet, the problem is ill-posed as multiple shapes can explain the same point sets - especially in the presence of outliers, noise, missing parts, or density variations.

Our contribution is to introduce an intuitive geometric criterion to reconstruct a piece-wise linear approximation of a sampled surface. Although our method does not provide topological guarantees

on the reconstruction other than watertightness, it achieves competitive quality to state-of-the-art methods in practice, as shown via various challenging examples (Fig. 1). Further, our solution is easy to implement, uses less memory, and is an order of magnitude faster than competing work, which makes it very practical. We propose two variants to address closed and open surfaces, respectively, since they pose differing challenges: **Global Ballmerge** which can be made automatic, guarantees a watertight surface and robustness regarding sampling density, outliers, and a reasonable noise level, and is thus ideally suited to reconstruct entire scans (closed surfaces that can contain holes). Moreover, it's extremely easy to implement (around 100 lines of code using the CGAL library). The second variant, called **Local Ballmerge**, reconstructs non-watertight surfaces well with the default parameter value, which is easily tunable to handle imperfections and handles even scans from various sensor types with strong noise and missing parts, so it targets open surfaces from scans of, e.g., outdoor scenes.

2. Related Work

Our algorithm supports curve and surface reconstruction. Here, we briefly review both, focusing on 3D reconstruction.

2D reconstruction targets a piece-wise linear approximation of a curve from a set of sample points. Although generalizing to 3D, the classic α-SHAPES [EKS83] assumes that the input is uniformly sampled according to maximum curvature, which can lead to dense sampling requirements. β-skeletons [KR85] employ a Delaunay Triangulation (DT) but exclude an edge $[pq]$ if the diameter of two adjacent Voronoi balls exceeds $\beta\|pq\|$. Our criterion is related but uses intersecting ratios, which are less restrictive in terms of radii differences and extend to 3D.

CRUST [ABE98] uses a new sampling model (ϵ-sampling) linking the sample quality to the local feature size [Rup93], through a parameter ϵ. The upper ϵ bound of initially 0.252 was later improved to $1/3$ [DK99]. Further, it was shown that the curve's piecewise approximation is a subset of the Delaunay triangulation. This motivated many Delaunay filtering algorithms, often targeting specific cases, e.g., open curves [DMR00] and sharp corners [DW01].

Handling sparse point sets [OMW16] was enabled via ρ-sampling. Similarly, outliers were addressed [PPT*19]. Additionally, non-feature specific reconstruction [PM16], as well as self-intersections and noise [PMM18] were covered, while optimal transport [dGCAD11] was shown to handle noise and outliers. Yet, most of these algorithms, including other classical ones [Att98], cannot be easily extended to 3D.

3D reconstruction via α-shapes [EM92] relies on the Delaunay complex (often called triangulation) of a uniformly-sampled point set P to contain a subset (subcomplex) well approximating the underlying 2-manifold of P. A faster computation is enabled via Ball-pivoting [BMR*99], where the surface is reconstructed by growing a seed by adding additional triangles. Local tangent-plane projection [Boi84], while handling local topological inconsistencies between tangent planes to correctly approximate the DT, is a popular acceleration that inspired others [GKS00; DFKM08; FR02; AGJ02; Kós01]. Attene et al. [AS00] extend the reconstruction to objects with genus> 0. The solution, like Chaine et al. [Cha03], uses the Gabriel graph but neither orders the former by a criterion, nor requires the result to be a manifold triangulation. These approaches suffer from artifacts in under-sampled regions.

CRUST [ACDL00] was the first method relating surface reconstruction to an ϵ-sampling. It filters the DT but produces slivers and fails in under-sampled regions. There are extensions to handle noisy samples and under-sampling at the cost of many unnecessary triangles, POWERCRUST [ACK01], or by removing some points [DG04], simplifications for homeomorphic reconstruction with $\epsilon < 0.06$ [ACDL00], solutions for surfaces with boundaries, COCONE [DG01], or including hole-filling strategies, TIGHTCOCONE [DG03]. Finally, involving graph cut [KSO04] improves reconstruction quality but requires post-processing to handle commonplace density-varying samples. Other methods use, e.g., the flow complex [Ede03] and WRAP [RS07], graph cuts [PB01; HK06; LPK09], witness complexes [GO08],SCALESPACE [DMSL11], compute the restricted Voronoi diagram [BL17], are based on a voxel structure [LLZ21], or compute an initial surface and sculpt from it [OMW13; WWX*22] but are not used much in practice.

Recent surveys cover fitting/prior-specific methods [BTS*14] and deep-learning approaches [YLL*20], which are outside the scope of this paper. POINTS2SURF [EGO*20] trains on local patches for detail and a down-sampled point cloud for global orientation simultaneously. Its quality exceeds SCREENED POISSON reconstruction [KBH06], which is widely used and computes an implicit function from a point cloud with oriented normals. The local-learning approach POINTTRINET [SO20] achieves higher geometrical but lower topological precision compared to Ball-pivoting [BMR*99]. POINT2MESH [HMGC20] uses Poisson reconstruction and iterative subdivisions to optimize the distance to the point cloud. It completes missing parts and handles scan artifacts well, but is slow and maintains the initial genus. DMNET [ZYT23] applies a graph neural network with local graph iterations for reconstruction. Our method is much simpler while resulting in competitive results and requiring no point-set preprocessing. Lastly, while our approach relates to Delaunay triangulation, we cannot provide topological guarantees on the reconstruction, but show that it is fast and performs well on many challenging cases.

3. Background and Definitions

Our method handles 2D-curve and 3D-surface reconstruction, using a unified observation on *medial balls*. To ease understanding, we focus on the 3D case but illustrate mostly 2D configurations.

Continuous surface Let C be a surface embedded in 3D. A medial ball of C is a ball whose interior contains no point of C yet is not included in any other such empty ball. The union of these ball centres is called the medial axis of C. For C being a closed surface, its medial axis is defined by balls inside C (interior medial balls) and balls outside of C (exterior medial balls). Interior and exterior balls only intersect at a point on C, and, indeed, C is the locus of intersections between interior and exterior medial balls. We will build upon this well-known characteristic observation to propose an intuitive yet robust reconstruction method in the discrete setting.

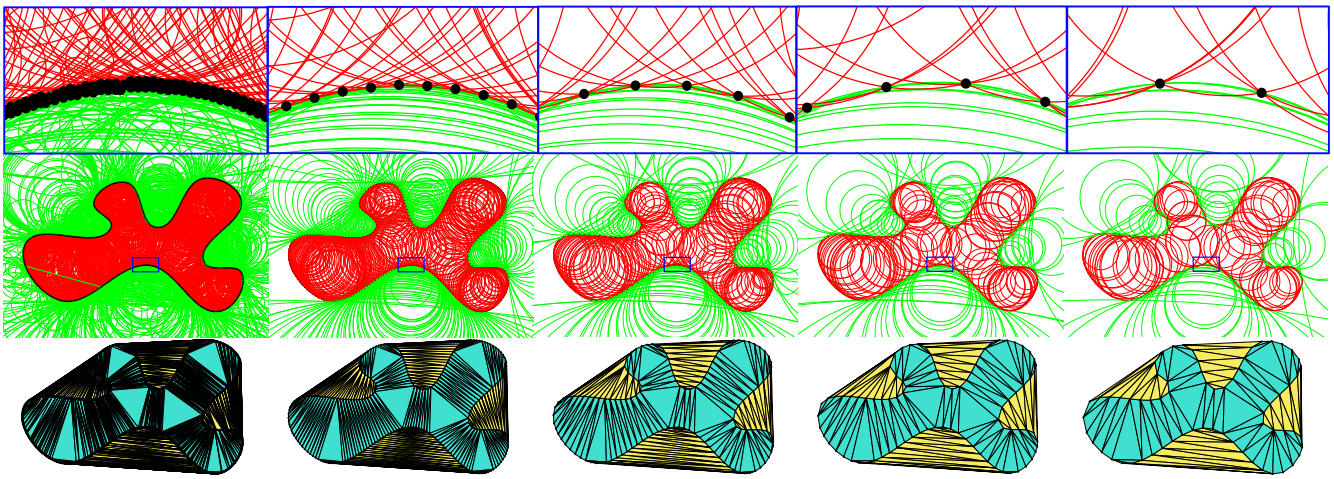


Figure 2: *Top: The interior (red) and exterior (green) medial balls of a point set with decreasing point density from l.t.r. Bottom: Corresponding δ-merged components on Delaunay triangulation.*

Sampled surface When the surface C is represented by discrete sample points P, the maximal balls empty of sample points P are referred to as Voronoi balls. These correspond to circumcircles of Delaunay triangles in 2D or circumspheres of Delaunay tetrahedra in 3D, see [Del*34] for instance. P approximates C, and the Voronoi balls (interior/exterior with respect to C) intersect along the boundary (Fig. 2) with a ratio inversely related to the sampling density. Amenta et al. show in Lemma 15 and 16 [ACK01] that inside and outside balls cannot intersect each other deeply, giving an explicit angle condition but it is valid only for a weak ϵ-sampling and based on the radius of the smaller ball. Therefore, based on the intersection between both the adjacent Voronoi balls, we design a criterion to decide for all triangles in 3D (edges in 2D) whether they are likely to belong to the approximated surface (curve), independent of any sampling condition.

4. Our Method

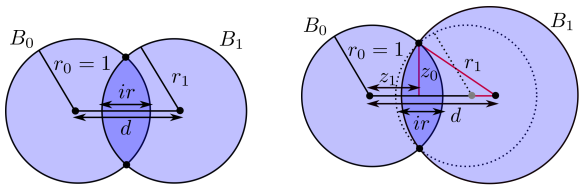
In the following, we will first provide the necessary definitions before introducing our algorithm. Our solution will be covered in its global variant (for closed surfaces) before introducing the local variant (for open surfaces).

4.1. Intersection ratio definition (2D and 3D)

We define the *intersection ratio* of two adjacent Voronoi balls B_0 and B_1 of radius r_0 and r_1 respectively, and the circumcenters' distance d as which represents the maximum ratio one ball intersects the other on the line segment between the two circumcenters (Fig. 3).

It is easy to see that the ratio ir only depends on the overlap ratio with the radius of the smaller Voronoi ball.We show that $0 \leq ir \leq 2$ (Fig. 4). Without loss of generality, let $r_0 \leq r_1$, $r_1' = r_1/r_0$ and $d' = d/r_0$. Since $r_1' \geq 1$, we have:

As the interior of B_1 is empty, it cannot contain B_0 entirely (which contains sample points on its boundary). Thus $d \geq r_1 - r_0$, which transforms into $d' \geq r_1' - 1$. Inserting into the definition of ir yields $ir \leq 2$. Further, as two adjacent Voronoi balls have to overlap, $d \leq r_1 + r_0$. This transforms into $d' \leq r_1' + 1$, and inserted into ir's definition yields $ir \geq 0$, so $ir \in [0..2]$.



(a) *Intersection ratio ir:fraction of r* **(b)** *Growing ball B_1 decreases ir*

Figure 3: *The intersection ratio definition and properties illustrated in 2D. Left: Voronoi balls B_0, B_1 share two samples as vertices of an edge. The intersection ratio ir is the sum of the radii $r_0 + r_1$ minus the distance d between the ball centers, as ratio of the smaller radius. Right: If the right ball grows, the distance d increases, and the arc's curvature between the shared samples decreases, thus reducing overlap and in turn, ir.*

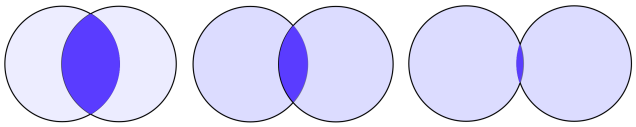


Figure 4: *Overlap examples: ir = 1 (left), 0.5 (center), 0.1 (right).*

BallMerge Intersection Criterion For a given d-dimensional point set P ($d = 2$ or 3), our algorithm first computes the Delaunay complex, denoted by DT_P. The pairs of adjacent Voronoi balls represent the circumspheres of two d-dimensional Delaunay simplices sharing a $(d-1)$-dimensional simplex. The Voronoi balls represent

Figure 5: *Automatic parameter tuning procedure: Starting from a seed tetrahedron (δ = 2), we iteratively decrease the δ threshold value until there is a sudden decrease in the vertex count of $G(\delta)$ (here, δ=1.64) and returns the $G(\delta)$ from the previous iteration (here, δ=1.65).*

the circumcircles/circumspheres of two triangles/tetrahedra sharing an edge/triangle in 2D/3D.

Our key idea is that if the intersection ratio between two adjacent Voronoi balls is small, they likely correspond to a pair of interior and exterior balls sharing a thin overlapping region along the underlying curve/surface (Fig. 2 top-right). More formally, given a threshold δ, an edge/triangle in 2D/3D is said to belong to a set $BM(\delta)$ if its two corresponding adjacent Voronoi balls have an intersection ratio less than δ. By definition, $BM(\delta)$ is a subcomplex of DT_P. It is easy to see that $BM(0)$ is empty, while $BM(\delta)$, for any $\delta \geq 2$, contains all the edges/triangles of DT_P in 2D/3D. This simple intuitive criterion forms the basis of our approach and is powerful enough to handle common artifacts in practice, such as outliers, noise, missing data, and downsampling (see Sec. 5.3).

4.2. Global algorithm

Just filtering Delaunay simplices based on the intersection-ratio thresholding does not guarantee a watertight surface. Here, we introduce a global merging procedure to ensure this property.

Merging procedure Given a threshold δ, we define an equivalence relation between Delaunay triangles/tetrahedra in 2D/3D. Two Delaunay simplices are called δ-merged either if their corresponding Voronoi balls are adjacent with an intersection ratio $> \delta$ or if there exists another Delaunay simplex, which is δ-merged with both of them. The Delaunay complex is hereby decomposed into equivalence classes of simplices (δ-*merged components* - as shown in Fig. 2). We define the δ-reconstructed shape as the largest δ-merged component of Delaunay simplices. Its boundary yields the reconstructed curve/surface.

Algorithmically, the same procedure as a classical connected component computation can be used. We start visiting δ-merged simplices from an arbitrary seed simplex (the procedure is order-independent) and tag them with the current component label until no more δ-merged simplices remain. Then we restart with any yet unvisited simplex until all have been visited, and output the simplices with the most frequent label. The reconstructed surface is then computed from the boundary of this output, as detailed below. The overall merging procedure is also detailed in Algorithm 1.

Boundary extraction & definition Let $G(\delta)$ be the set of edges (triangles) in DT_P on the boundary of the largest δ-merged component of Delaunay triangles (tetrahedra) in 2D (3D). $G(\delta)$ is by definition, a closed curve (a watertight surface) since it bounds a solid - even if it is not necessarily manifold. It is referred to as the reconstructed curve (surface) of our GLOBAL BALLMERGE method.

Algorithm 1 Merge

```
1:  procedure MERGE(P,δ)
2:      DT_P ← Delaunay_Triangulation(P)
3:      Group ← 0, Merged_Group ← ϕ
4:      for each d-Simplex X_i ∈ DT_P in R^d, d = 2,3 do
5:          if X_i.visited = false then
6:              Group ← Group+1
7:              Q ← {X_i}
8:              while Q ≠ {} do
9:                  X_j ← Q.last
10:                 Q ← Q \ X_j
11:                 Merged_Group ← Merged_Group ⋃ X_j
12:                 X_j.visited ← true
13:                 X_j.group ← Group
14:                 for each X_k ∈ DT_P \ X_j, X_k ∩ X_j ≠ {} do
15:                     if ir(X_j, X_k) ≥ δ then
16:                         Q ← Q+X_k
17:     return the largest among Merged_{1...Group}
```

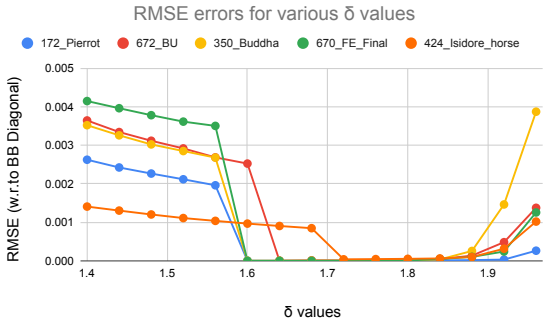


Figure 6: *RMSE for varied δ values on representative models.*

Automatic parameter converging Identifying a good value for δ turns out to be quite straightforward (and corresponds in most of our experiments to a range from 1.65 to 1.85). Starting from the maximal value 2, we iteratively decrease δ until G(δ) has a sudden vertex-count decrease. This means that a large amount of interior/exterior medial balls became δ-merged, meaning that part of the reconstructed surface collapsed. Consequently, our algorithm returns the $G(\delta)$ from the previous iteration as the optimal result. Fig. 5 shows intermediate results decrementing $G(\delta)$ by 0.01 per iteration, while Fig. 6 shows the significant vertex-count change for several examples.

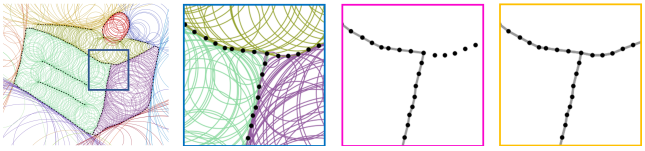


Figure 7: *L.t.r.: Global view with colored Voronoi balls, closeup (blue), reconstruction - global (pink)/local (orange).*

4.3. Local algorithm

Although our GLOBAL BALLMERGE algorithm is well suited for reconstructing watertight surfaces, many real-world scans represent non-watertight models due to extensive missing data and open surfaces. For example, a set of points representing a shape with open curves and junctions as in Fig. 7. Though our method could successfully partition the Voronoi balls based on the intersection ratio, we did not get the expected result using GLOBAL BALLMERGE. Therefore, in this local variant of the BALLMERGE algorithm, for any two adjacent d-simplices, which are not δ-mergeable, the shared $(d-1)$-simplex is retained. As expected and also observed in our experiments, outliers and open surfaces result in very large simplices. Thus, we consider elements δ-mergeable only if the longest edge of the simplex is smaller than $\frac{1}{\eta}B$ (η is a scaling factor, B is the length of the bounding box diagonal) to remove large simplices (typical for open meshes).

In practice, the threshold $\eta = 200$ was observed to perform well and can be adjusted if needed (Fig. 8 shows the effect of varying η). A value of $\delta = 1.85$ has shown to work well in general and is specified for results if adjusted. The overall procedure is given in Algorithm 2.

Algorithm 2 LOCAL BALLMERGE

```
1: procedure LOCAL BALLMERGE(P,δ,η)
2:     DT_P = Delaunay_Triangulation(P)
3:     T ← ∅
4:     for each k-Simplex X_i ∈ DT_P in R^d, d = 2,3 do
5:         for each X_j ∈ DT_P \ X_i, X_j ∩ X_i ≠ {} do
6:             if maxlength(F) < BBDiagLen/η then
7:                 if ir(X_i, X_j) ≤ δ then
8:                     T ← T + (X_i ∩ X_j)
9:     return T
```

5. Results and Discussion

Here, we show the performance of both our GLOBAL and LOCAL BALLMERGE reconstruction algorithms on different real-world datasets, and present various results with detailed analysis and quantitative as well as qualitative evaluations of challenging inputs.

5.1. Comparison 2D Reconstruction

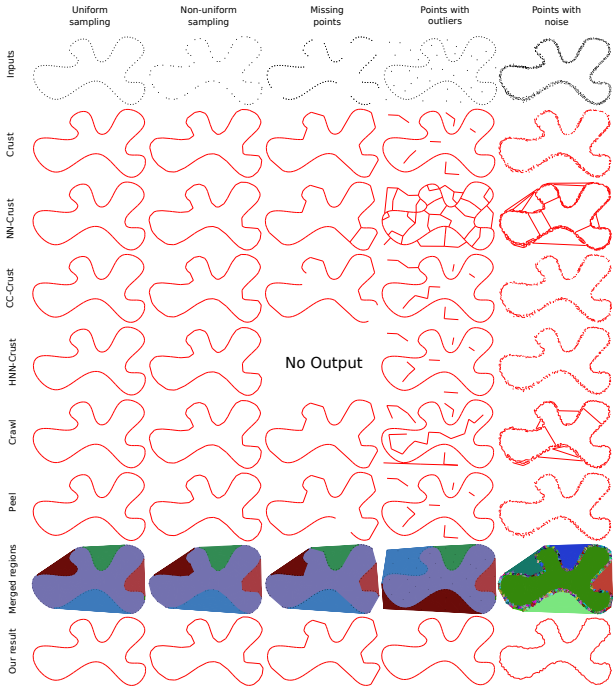


Figure 9: *2D case: Input,* CRUST, NN-CRUST, CC-CRUST, HNN-CRUST, CRAWL, PEEL, GLOBAL BALLMERGE *merged components, result applied to: Uniform/non-uniform sampling, missing data, outliers, and noise.*

We compare the global variant of our algorithm in 2D to various state-of-the-art reconstruction algorithms, using a recent benchmark [OPP*21]. Fig. 9 shows results with different input types (uniformly sampled, non-uniformly sampled, missing points, points with outliers, and noisy points) and methods (ours, CRUST [ABE98], NN-CRUST [DK99], HNN-CRUST [OMW16], CRAWL [PM16], and PEEL [PMM18]). The competing methods have all been designed to reconstruct curves from non-uniformly spaced samples. As can be seen, all algorithms performed well under dense (uniform and non-uniform) sampling. For incomplete point sets (missing parts), HNN-CRUST fails to generate edges. Adding outliers and noise causes all competing methods to fail.

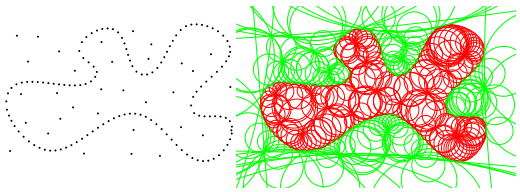


Figure 10: *2D point set with outliers and Voronoi balls.*

BALLMERGE handles outliers gracefully, as the surrounding Voronoi balls will have larger intersection ratios than those at the surface (Fig. 10). In the same way, our algorithm handles noise

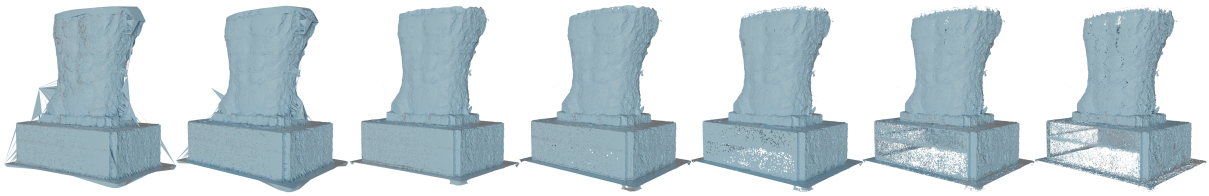


Figure 8: *Effect of* η *(*$\delta = 1.8$*); l.t.r:* $\eta = 5, 50, 100, 150, 200, 250, 300$ *(mesh EPFL Geometric Computing Laboratory).*

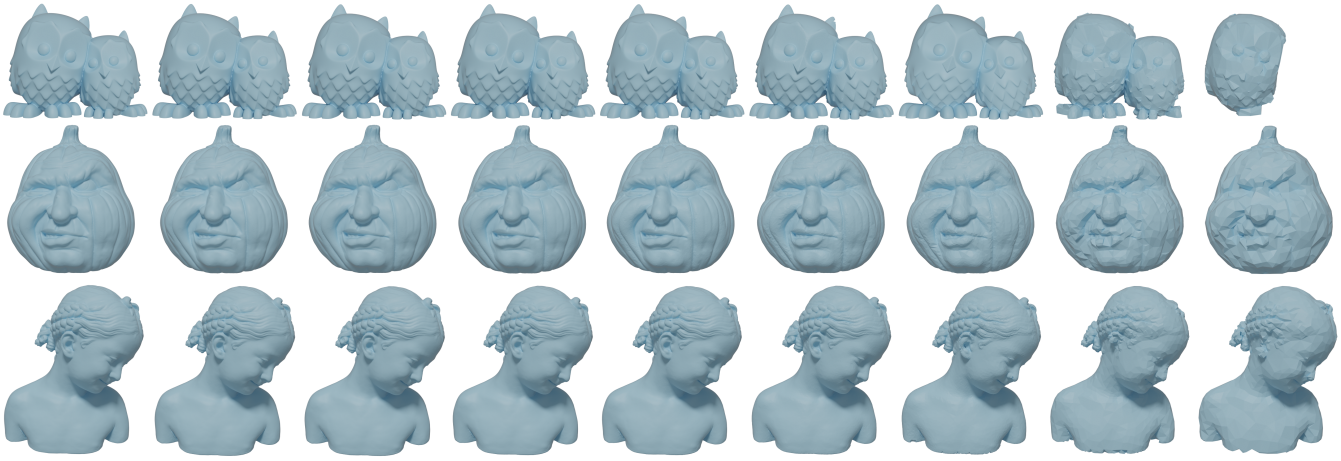


Figure 11: *Result of our algorithm on down-sampling. L.t.r.: 100%, 80%, 60%, 40%, 20%, 10%, 5%, 1%, 0.05% of points are retained using ReMesh simplification. Models taken from [HWW*22] (originally from Thingi10k [ZJ16]) and aim@shape.*

well. Even if the samples seem to shrink the shape, our solution captures the underlying object best and outputs a watertight boundary (Fig. 9).

Our algorithm produces generally equal or better results due to our robust merge criterion.

5.2. Comparison 3D reconstruction

We also compare 3D reconstruction against state-of-the-art techniques: Delaunay triangulation-based algorithms (POWERCRUST, TIGHTCOCONE, CONNECT3D, and SCALESPACE), implicit reconstruction algorithm (SCREENEDPOISSON), and recent learning-based algorithms (POINTTRINET and POINTS2SURF). SCREENED POISSON needs normal information, which we derived via the default solution in MeshLab. Most methods, as also GLOBAL BALLMERGE, enforce a watertight surface, which can be limiting for non-orientable surfaces. In contrast, SCALE SPACE and LOCAL BALLMERGE differ in this respect. For completeness, we will also compare them using a benchmark that includes diverse open surfaces (a category that our method is not designed for).

For the quantitative evaluation, a real watertight VRIP (DRAGON from aim@shape) and a real open surface scan (CAR from KSR42 dataset), as well as point sets with several artifacts (representative of real-world scans) were chosen: an object with hole and missing points due to occlusion (HAND simulated via the Blensor tool [GKUP11]), varying downsampling levels (downsampling performed with the ReMesh tool [AF06], see Fig. 11), outliers

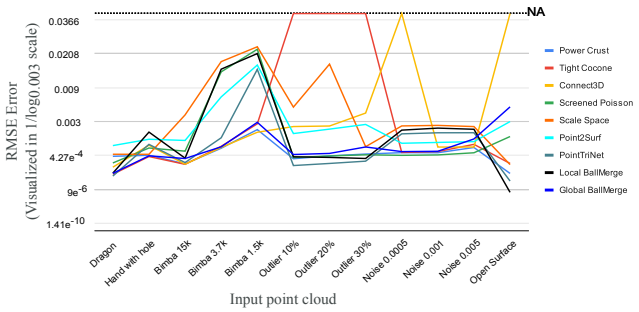


Figure 12: *RMSE (w.r.t. bounding box diagonal) for various algorithms, artifacts and levels.*

(DRAGON with 10%, 20% and 30% of outliers [RLG*20]), and noise (Blensor on FERTILITY). We report the Hausdorff distance to the ground truth (clean models). Our variants, for watertight and open surfaces, exhibit superior or on-par performance compared to other methods (Fig. 5.2). Further, our methods are not input-specific, while some algorithms failed to generate an output (due to input assumptions that did not hold).

While our method is robust to noise, especially for open scans (Fig. 13), and identifies the shape well, the smoothing property of SCREENED POISSON leads to a comparatively lower error with increasing noise. Yet, the results, in this case, show an over smoothing

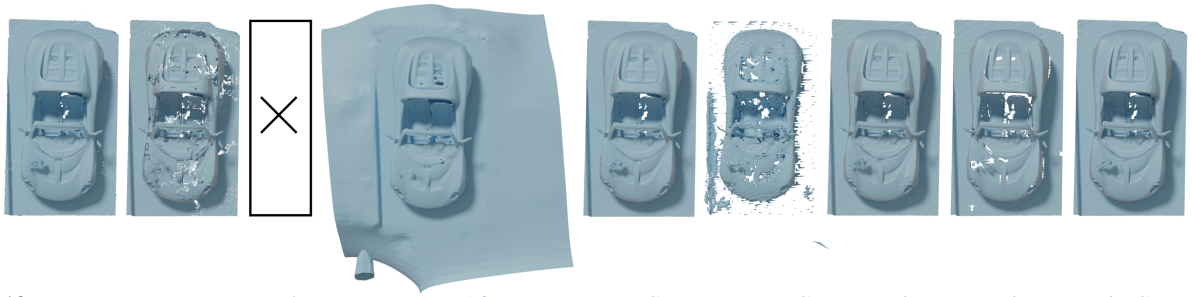
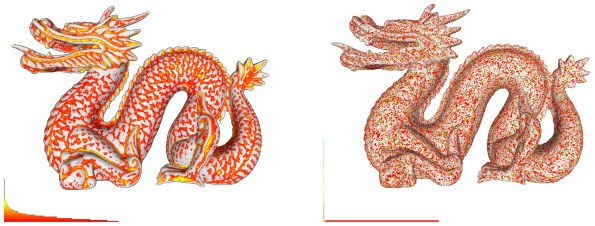


Figure 13: *Comparison on open-surface scan [CZMK16]. L.t.r.:* PowerCrust, TightCocone *(long triangles pruned),* Connect3D *(no generated result),* ScreenedPoisson, ScaleSpace, Point2Surf, PointTriNet, *Ours:* Global BallMerge *(long triangles pruned),* Local BallMerge *($\delta = 1.95$, $\eta = 200$).*



(a) Screened Poisson *RMSE* **(b)** BallMerge *RMSE*

Figure 14: *Hausdorff distance to reference at sampled points (error: green to red).* Screened Poisson *handles smooth regions well, but affects sharp features (max=0.007, mean=0.00015, RMSE=0.000239), better handled by our method (max=0.01, mean=0.000012, RMSE=0.000152).*

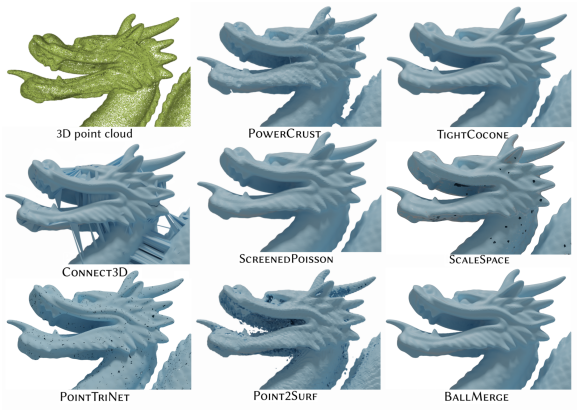


Figure 15: *Details of* Dragon *versus state of the art.*

of sharp features (Fig. 14). Our algorithm has a much more constant error distribution and is more robust to outliers than the other methods. It also handles incomplete scans, where Screened Poisson performs poorly. Figs. 15 and 16 show more examples of our generally superior results. One may note extra erroneous (PowerCrust and Connect3D) or missing triangles (Point2Surf and PointTriNet) in the results of the competing methods.

Fig. 17 and Fig. 18 show runtime and memory-usage comparisons. Our parametric version takes comparatively little time (magnitudes faster than most) and memory, even with increasing input size, showing that it is lightweight (Table 1).

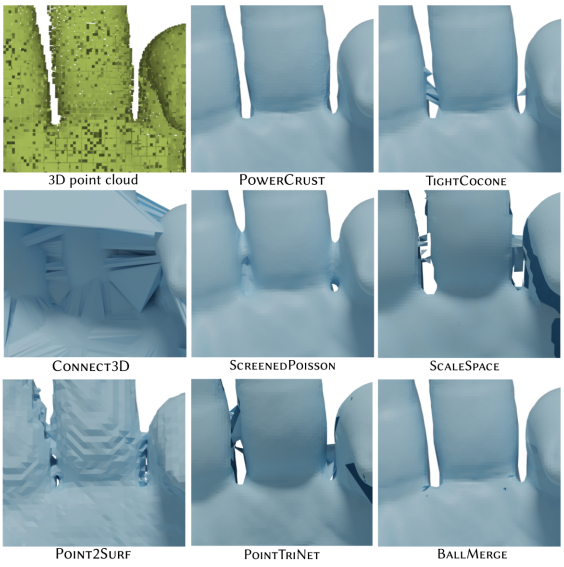


Figure 16: *Zoomed-in details of visual comparison with the state-of-the-art for a synthetically scanned object (using Blensor) with occlusions and holes. We used $\delta = 1.87$ for* Global BallMerge *and default parameters for all other methods.*

Finally, Fig. 19 shows our comparison with classical Delaunay-based algorithms - PowerCrust [ACK01], BPA [BMR*99], Cocone [DG03], TightCocone [DG03], RobustCocone [DG04], SingularCocone [DW13], and AdvancingFront [CD04]. Please note that we used the default parameter settings for all the algorithms. As it can be seen, our results are better or on par with other methods.

5.3. Reconstruction Robustness

We illustrate the robustness of our method via multiple experiments. Besides discussing particular data types, we also report performance on recent reconstruction benchmarks.

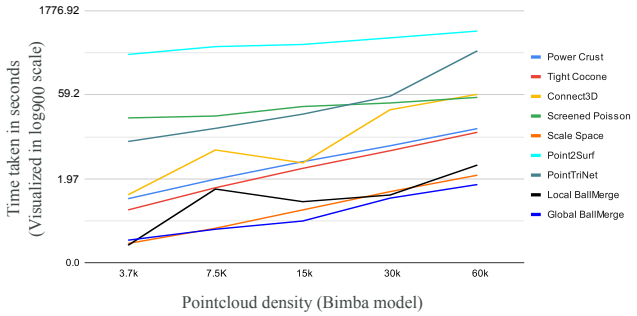


Figure 17: *Runtime on models with varying density.*

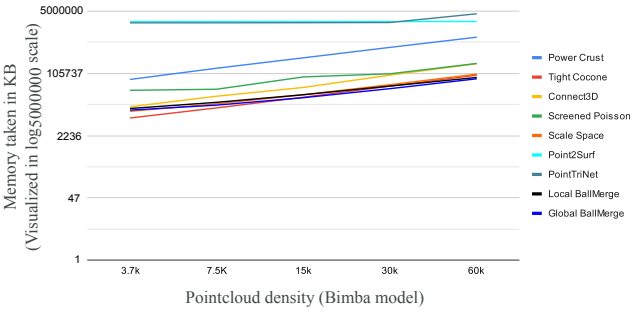


Figure 18: *Memory usage on models with varying density.*

Fig. 20 gathers automatically-generated results obtained by the parameter-free GLOBAL BALLMERGE for several point clouds. Our method successfully captured various challenging features, guarantees watertightness by definition, and can address downsampling artifacts very well (Fig. 11).

Noise and outliers are well handled by our BALLMERGE criterion (which is shared by both variants), even if the methods themselves are not specially tuned for this purpose.

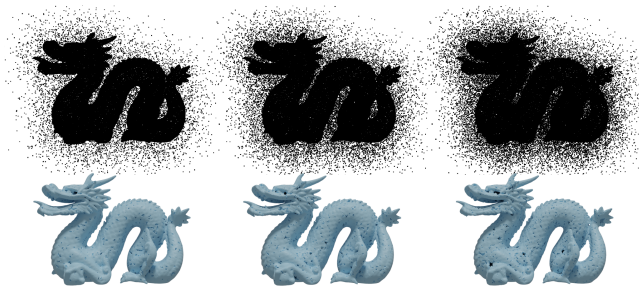


Figure 21: *Results of* GLOBAL BALLMERGE *with from l.t.r.: 10%, 20%, and 30% outliers -* $\delta = 1.75, 1.7, 1.68$ *respectively (RMS error w.r.t bbox diagonal: 0.000567, 0.000554, and 0.000793).*

Point clouds with outliers are common due to scanning artifacts. As explained, our method successfully withstands a high level of outliers (Fig. 21, data from [RLG*20]). It produces a low reconstruction error, increasing only slowly with more outliers.

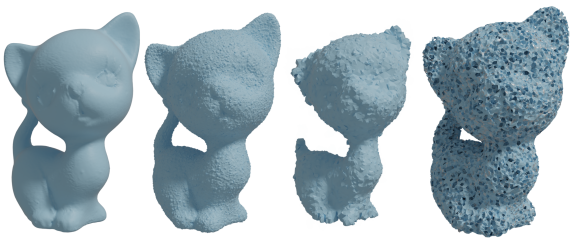


Figure 22: *Effect of synthetic noise on ReMesh [AF06]. L.t.r:* GLOBAL BALLMERGE *for a noise amplitude (ReMesh, related to BBall radius) of 0, 100, and 1000 and* LOCAL BALLMERGE *(not watertight,* $\delta = 1.8$, $\eta = 50$*) for 1000.*

Noisy point clouds Noise is another common artifact of 3D scanning, e.g., due to sensor limitations. Fig. 22 shows that our algorithm robustly identifies the shapes from very noisy point clouds, produced with the ReMesh tool (noise amplitudes of 0, 100, and 1000 of the bounding ball radius). The band around the boundary contains relatively small Voronoi balls compared to the interior/exterior balls. It therefore acts as an isolation barrier between both sets (Fig. 23). However, for high noise levels, the reconstructed shape shrinks slightly when using the GLOBAL method, while the LOCAL variant is much more robust.

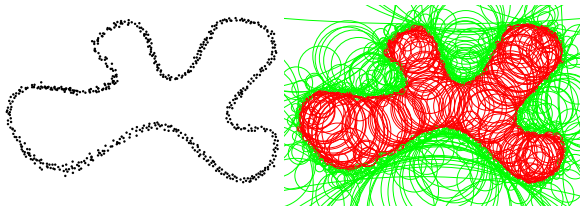


Figure 23: *2D case: noisy point set with Voronoi balls.*

Open Surfaces To illustrate the practical usefulness of BallMerge with respect to other related works, we evaluated our local algorithm on a benchmark [HWW*22] that contains many open surfaces. Results shown in Fig. 24 compared to corresponding benchmarks provided in Table 5 of [HWW*22] confirm that our algorithms generally perform as good as the state-of-the-art techniques while being an order of magnitude faster. In short, the Normal Consistency Score (NCS) measures normal correctness, Chamfer Distance (CD)/F-score is an average/worst surface error metric, and Neural Feature Similarity (NFS) compares two shapes in the deep feature. As visible in Fig. 24, our LOCAL BALLMERGE shows mostly on-par or superior performance compared to related methods. Especially for clean models, our method outperforms the others.

We also successfully generate watertight meshes from the real-world point-cloud scans from aim@shape (Fig. 20). However, scans representing an unoriented manifold (no triangulation can close its boundaries), GLOBAL BALLMERGE will fail because it relies on orienting the Delaunay triangles. Fig. 25 shows an example

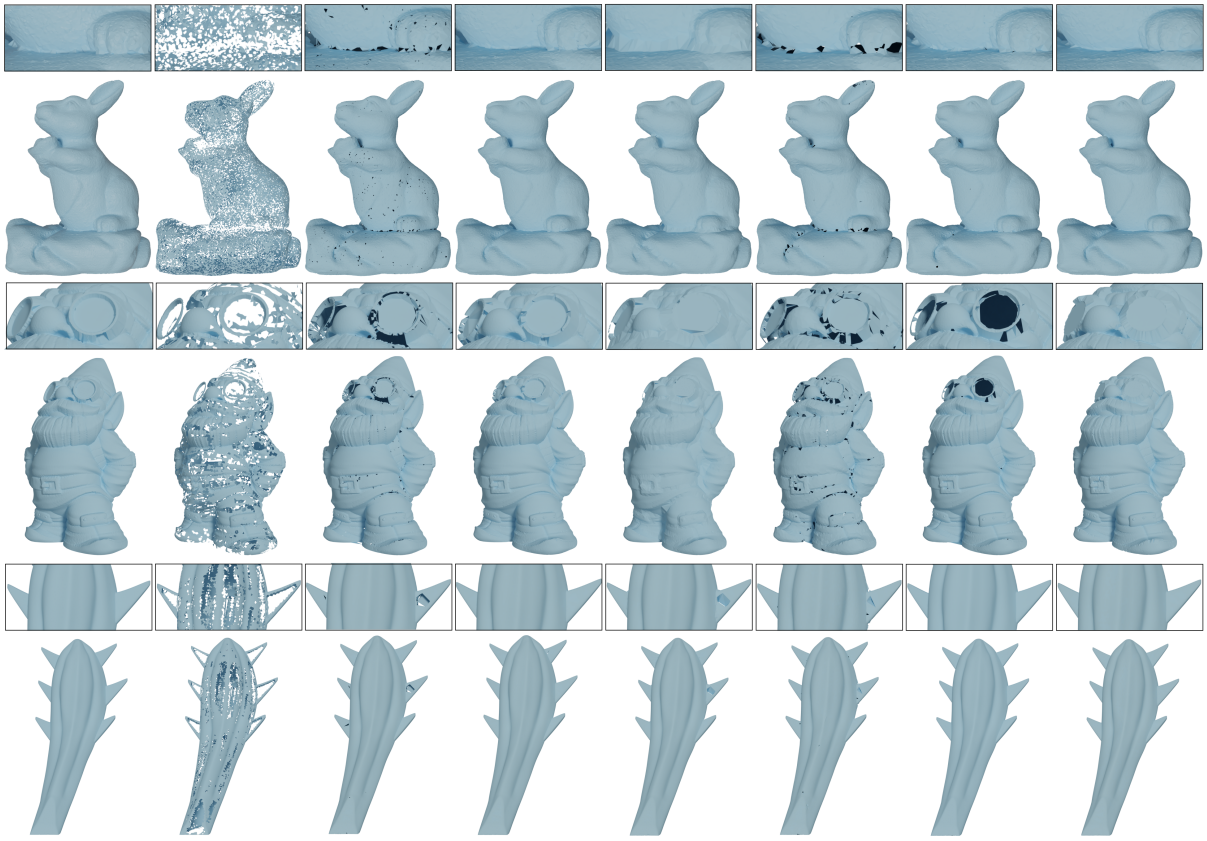


Figure 19: *Comparison with classical Delaunay-based reconstruction techniques. L.t.r.: reconstruction results using* PowerCrust, BPA, Cocone, TightCocone, RobustCocone, SingularCocone, AdvancingFront, Global BallMerge.

where even long-triangle removal from the Global BallMerge result is unsatisfactory because tetrahedra are merged through large holes on both exterior and interior (thus not capturing the boundary triangles). For outdoor scans, this is common, as the scanner captures the ground, which is not a watertight model. It is addressed by Local BallMerge, and Fig. 26 shows its results on various 3D scans. Similarly, Fig. 27 shows that our efficient reconstruction can produce good meshes even from large datasets.

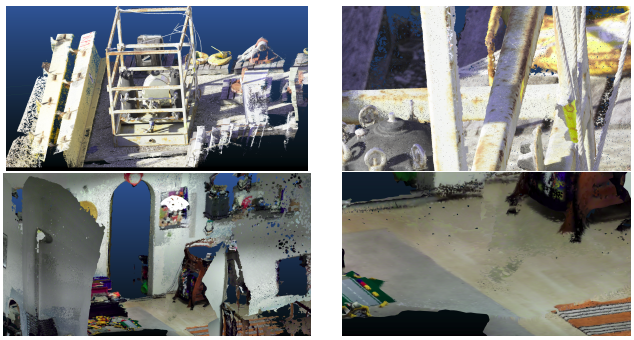


Figure 27: *Top: Reconstructed* Module *(photogrammetry, 58M points) plus zoom. Bottom: Reconstructed* Livingroom *(RGB-D camera, 1M points) plus zoom (both* $\delta = 1.85, \eta = 200$*).*

Input (# of points)	Time (in seconds)			#nm (#Faces)
	DT	Merge	Write	
Dragon (437k)	1.3169	0.5299	0.5922	79 (437k)
Hand (84k)	0.3196	0.9704	0.1034	3 (168k)
Outlier 10% (140k)	0.4497	0.146	0.8865	516 (257k)
Outlier 20% (140k)	0.3975	0.1599	0.9645	693 (236k)
Outlier 30% (140k)	0.4195	0.1675	0.9856	796 (213k)
Noise0.0005 (131k)	0.4094	0.1474	0.2234	0 (261k)
Noise0.001 (131k)	0.4321	0.1434	0.2587	0 (260k)
Noise0.005 (131k)	0.394	0.1688	0.1814	31 (164k)
Open Surface^{+} (1259k)	3.43	*	4.195	-
Livingroom^{+} (1152k)	2.95	*	5.37	-
Retz^{+} (49784k)	164.10	*	191.62	-
Module^{+} (58318k)	188.86	*	423.45	-

Table 1: *Runtime. We used* Local BallMerge *on scans (marked with* $^{+}$*), else* Global BallMerge. *Intersection ratios are computed, while the file is written, where the writing time depends on the triangle count. (#nm) counts the non-manifold vertices. Large datasets (marked with +), were measured on a similar machine with more memory.*

5.4. Runtime Performance

Our algorithm is lightweight and requires only a little computation in terms of input size. It performs two quick linear-time passes on

Figure 20: *Automatically-generated Watertight triangulations via* GLOBAL BALLMERGE *for point clouds (up to 3609k points) from aim@shape and Stanford ([TL94],[CL96]) with challenging features (e.g., corners, ridges, and close surface sheets).*

the Delaunay complex (global), or just a single pass (local). Hence, it is dominated by its $O(NlogN)$ construction time expected for surface samples, although we observed almost linear time performance in practice. Table 1 shows the run time of our algorithm (written in C++ using CGAL libraries [PY20; Yvi20; JPT20a; JPT20b]) on an 8-core 3.70GHz PC with 32GB memory. We evaluated the Delaunay computation, Ball Merging (global variant), and writing of the largest component to a file. Merging takes significantly less time than the other steps, even for large datasets, showing almost linear behavior in relation to the number of points. Notably, our algorithm runs a magnitude faster than the next fastest competitor (see Fig. 17).

5.5. Limitations

GLOBAL BALLMERGE handles sparse point clouds well, is robust to outliers and noise (outperforms most competitors for practical cases), and it targets watertight models, not unorientable surfaces. The latter is handled by LOCAL BALLMERGE. Yet, our approach is not well-suited for the reconstruction of artificial models with sharp ridges (such as FANDISK), very close surface sheets (an extremely down-sampled BUNNY lacks ears), or extremely high outliers, a limitation shared with most related reconstruction methods (Fig. 28). An interesting future extension could be the use of *protecting balls* along sharp features [CDR10]. Also, unlike LOCAL BALLMERGE, which can reconstruct multiple components (as shown in Fig.29), GLOBAL BALLMERGE can only reconstruct a single watertight surface. Finally, though our GLOBAL BALLMERGE algorithm guarantees a watertight surface, a few faces might not be manifold (Table 1), which can be easily corrected by adopting an inflating step [OM13].

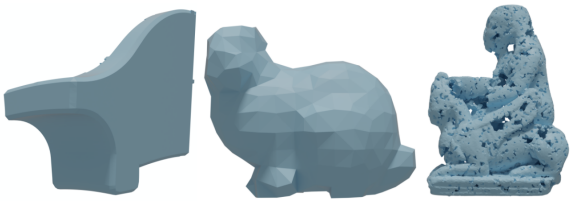


Figure 28: *Limit cases (l.t.r.): Sharp edges on artificial object, under-sampled regions (bunny ears) not reconstructed (down-sampled to 1%), and extreme outlier levels (40%).*

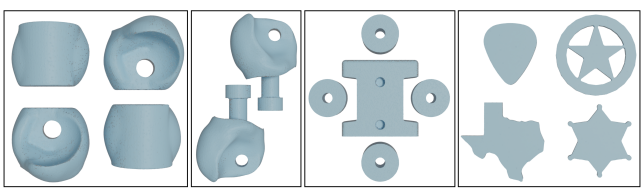


Figure 29: *Multi-component* LOCAL BALLMERGE *results* ($\delta = 1.85$, $\eta = 100$)*, Models from [HWW*22] (originally [ZJ16]).*

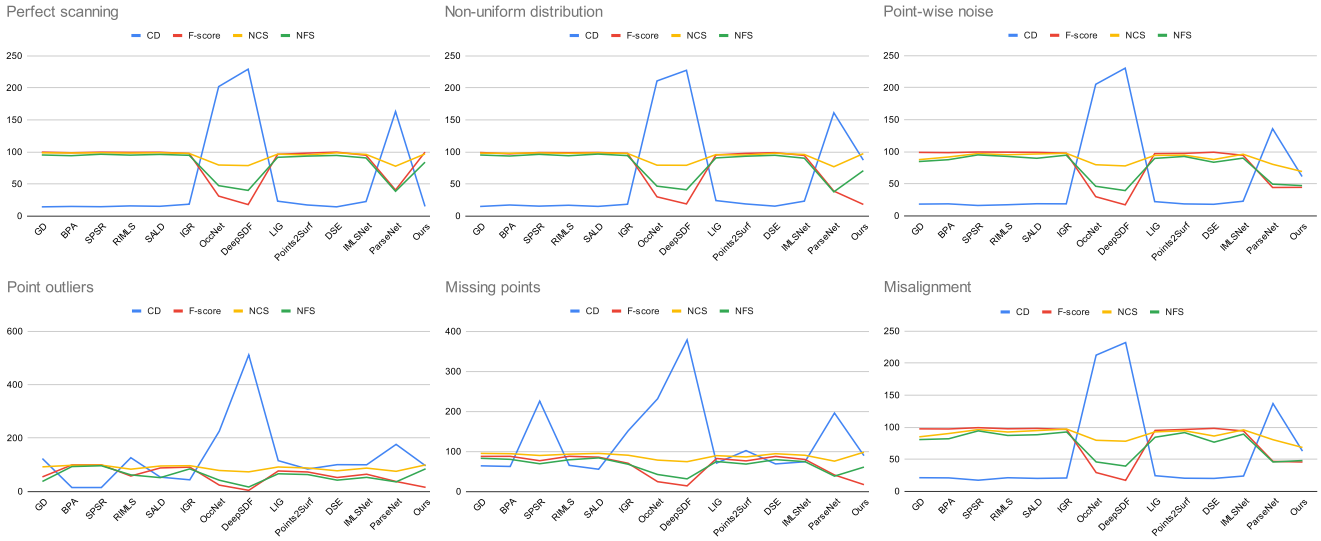


Figure 24: *Evaluation of* LOCAL BALLMERGE *on benchmark [HWW*22], where we expect to have minimum CD and maximum F-score, NCS and NFS. As can be seen, our results on perfect scans outperform other methods and give comparable results in the remaining cases. Please note that we used a fixed* η *for each class for pruning unnecessary triangles, which sometimes resulted in redundant triangles, altering the scores from user-adjusted parameters for* η *and* δ.

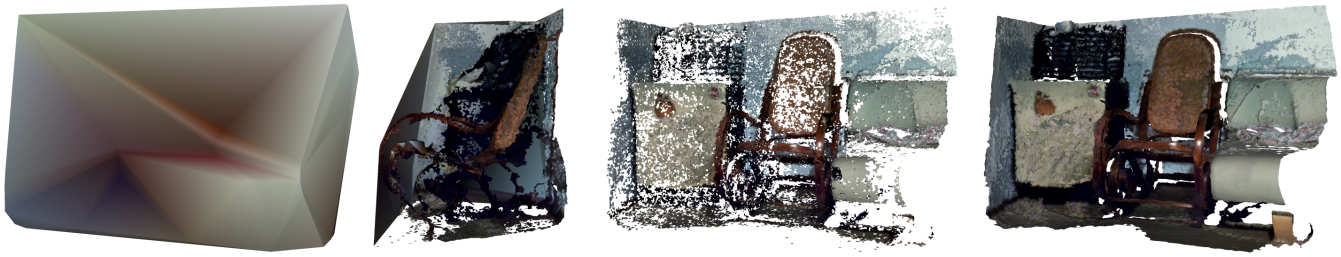


Figure 25: LIVINGROOM *detail, L.t.r: Output of* GLOBAL BALLMERGE, *it's Cut view, Result of* GLOBAL BALLMERGE *after removing long triangles, (d) Result of* LOCAL BALLMERGE ($\delta = 1.85, \eta = 200$).

Figure 26: *Meshes generated via* LOCAL BALLMERGE *(with default* $\delta = 1.85$ *and* $\eta = 200$*) of real-world point clouds (up to 1378k points) from the KSR42_dataset [BL20] and the Statue Model Repository (©EPFL Geometric Computing Laboratory).*

6. Conclusion & Future Work

BALLMERGE is a simple and intuitive point-cloud reconstruction algorithm with two variants. It handles real-world outdoor scans well and outperforms existing reconstruction methods in many cases, as shown in various examples. Our simple criterion makes it a magnitude faster than competing methods, while using less memory and delivering competitive reconstruction quality. Experiments show that it works well on under-sampled point clouds and those infected with noise and outliers, which is essential for real-world scans. Moreover, GLOBAL BALLMERGE guarantees watertight results, which is crucial for many applications. As the DT construction dominates runtime, parallel GPU or multi-CPU execution could accelerate our method further. The fact that BALLMERGE fills holes robustly and with fair triangulations also makes it attractive for mesh repairs, such as hole-filling or transforming triangle soups into watertight models. Finally, given the high quality of our reconstruction, linking the BALLMERGE criterion and sampling-condition-based topological guarantees is interesting future work.

7. Acknowledgements

MODULE is courtesy of Weiss AG. The scan of RETZ townplace is courtesy of Riegl company. Nefertiti is courtesy of Cosmo Wenman (from Berlin's Neues Museum made available in Thingiverse under cc license). The other 3D models are taken from public repositories like aim@shape, Thingi10k, EPFL statue repository, KSR42, [CZMK16], [HWW*22], [RLG*20], and Stanford repository. We thank Derek Liu for giving us permission to use his Blender toolbox. This work has been partially funded by the Austrian Science Fund (FWF) project no. P32418-N31, the Wiener Wissenschafts-, Forschungs- und Technologiefonds (WWTF) project ICT19-009 and by the NWO Vernieuwingsimpuls' VIDI grant NextView.

References

[ABE98] AMENTA, NINA, BERN, MARSHALL, and EPPSTEIN, DAVID. "The crust and the β-skeleton: Combinatorial curve reconstruction". *Graphical models and image processing* 60.2 (1998), 125–135 2, 5.

[ACDL00] AMENTA, N., CHOI, S., DEY, T. K., and LEEKHA, N. "A simple algorithm for homeomorphic surface reconstruction". *SCG '00: Proc. 16th ann. SCG*. New York, NY, USA: ACM, 2000, 213–222 2.

[ACK01] AMENTA, N., CHOI, S., and KOLLURI, R.K. "The power crust, unions of balls, and the medial axis transform". *Computational Geometry* 19 (2001), 127–153(27) 2, 3, 7.

[AF06] ATTENE, MARCO and FALCIDIENO, BIANCA. "Remesh: An interactive environment to edit and repair triangle meshes". *IEEE International Conference on Shape Modeling and Applications 2006 (SMI'06)*. IEEE. 2006, 41–41 6, 8.

[AGJ02] ADAMY, UDO, GIESEN, JOACHIM, and JOHN, MATTHIAS. "Surface reconstruction using umbrella filters". *Comput. Geom. Theory Appl.* 21.1 (2002), 63–86. ISSN: 0925-7721 2.

[AS00] ATTENE, MARCO and SPAGNUOLO, MICHELA. "Automatic surface reconstruction from point sets in space". *CG Forum* 19.3 (2000), 457–465 2.

[Att98] ATTALI, DOMINIQUE. "r-Regular shape reconstruction from unorganized points". *Computational Geometry* 10.4 (1998), 239–247 2.

[BL17] BOLTCHEVA, DOBRINA and LÉVY, BRUNO. "Surface reconstruction by computing restricted voronoi cells in parallel". *Computer-Aided Design* 90 (2017), 123–134 2.

[BL20] BAUCHET, JEAN-PHILIPPE and LAFARGE, FLORENT. "Kinetic shape reconstruction". *ACM Transactions on Graphics (TOG)* 39.5 (2020), 1–14 11.

[BMR*99] BERNARDINI, FAUSTO, MITTLEMAN, JOSHUA, RUSHMEIER, HOLLY, et al. "The Ball-Pivoting Algorithm for Surface Reconstruction". *IEEE TVCG* 5.4 (1999), 349–359 2, 7.

[Boi84] BOISSONNAT, JEAN-DANIEL. "Geometric structures for three-dimensional shape representation". *ACM TOG* 3.4 (1984), 266–286 2.

[BTS*14] BERGER, MATTHEW, TAGLIASACCHI, ANDREA, SEVERSKY, LEE, et al. "State of the art in surface reconstruction from point clouds". *Eurographics 2014-State of the Art Reports*. Vol. 1. 1. 2014, 161–185 2.

[CD04] COHEN-STEINER, DAVID and DA, FRANK. "A greedy Delaunay-based surface reconstruction algorithm". *Vis. Comput.* 20.1 (2004) 7.

[CDR10] CHENG, SIU-WING, DEY, TAMAL K, and RAMOS, EDGAR A. "Delaunay refinement for piecewise smooth complexes". *Discrete & Computational Geometry* 43.1 (2010), 121–166 10.

[Cha03] CHAINE, RAPHAËLLE. "A geometric convection approach of 3-D reconstruction". *SGP '03: Proc. of the 2003 Eurographics/ACM SIGGRAPH symposium on Geometry processing*. 2003, 218–229 2.

[CL96] CURLESS, BRIAN and LEVOY, MARC. "A volumetric method for building complex models from range images". *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*. 1996, 303–312 10.

[CZMK16] CHOI, SUNGJOON, ZHOU, QIAN-YI, MILLER, STEPHEN, and KOLTUN, VLADLEN. "A Large Dataset of Object Scans". *arXiv:1602.02481* (2016) 7, 12.

[Del*34] DELAUNAY, BORIS et al. "Sur la sphere vide". *Izv. Akad. Nauk SSSR, Otdelenie Matematicheskii i Estestvennyka Nauk* 7.793-800 (1934), 1–2 3.

[DFKM08] DUMITRIU, DANIEL, FUNKE, STEFAN, KUTZ, MARTIN, and MILOSAVLJEVIC, NIKOLA. "How much Geometry it takes to Reconstruct a 2-Manifold in R^3". *Proc. 10th Workshop on Alg. Eng. and Exp., ALENEX 2008, San Francisco, USA*. ACM-SIAM. 2008, 65–74 2.

[DG01] DEY, TAMAL K. and GIESEN, JOACHIM. "Detecting undersampling in surface reconstruction". *SCG '01: Proceedings of the seventeenth annual symposium on Computational geometry*. 2001 2.

[DG03] DEY, TAMAL K. and GOSWAMI, SAMRAT. "Tight cocone: a water-tight surface reconstructor". *SM '03: Proceedings of the eighth ACM symposium on Solid modeling and applications*. ACM, 2003 2, 7.

[DG04] DEY, TAMAL K. and GOSWAMI, SAMRAT. "Provable surface reconstruction from noisy samples". *Proceedings of the 20th annual symposium on Computational geometry*. SCG '04. ACM, 2004 2, 7.

[dGCAD11] DE GOES, FERNANDO, COHEN-STEINER, DAVID, ALLIEZ, PIERRE, and DESBRUN, MATHIEU. "An Optimal Transport Approach to Robust Reconstruction and Simplification of 2D Shapes". *Computer Graphics Forum* 30.5 (2011), 1593–1602 2.

[DK99] DEY, TAMAL K and KUMAR, PIYUSH. "A Simple Provable Algorithm for Curve Reconstruction." *SODA*. Vol. 99. 1999 2, 5.

[DMR00] DEY, TAMAL K., MEHLHORN, KURT, and RAMOS, EDGAR A. "Curve reconstruction: Connecting dots with good reason". *Computational Geometry* 15.4 (2000), 229–244. ISSN: 0925-7721 2.

[DMSL11] DIGNE, JULIE, MOREL, JEAN-MICHEL, SOUZANI, CHARYAR-MEHDI, and LARTIGUE, CLAIRE. "Scale space meshing of raw data point sets". *Computer Graphics Forum*. Vol. 30. 6. Wiley Online Library. 2011, 1630–1642 2.

[DW01] DEY, TAMAL K. and WENGER, REPHAEL. "Reconstructing curves with sharp corners". *Computational Geometry* 19.2 (2001). Combinatorial Curves and Surfaces, 89–99. ISSN: 0925-7721 2.

[DW13] DEY, TAMAL K and WANG, LEI. "Voronoi-based feature curves extraction for sampled singular surfaces". *Computers & Graphics* 37.6 (2013), 659–668 7.

[Ede03] EDELSBRUNNER, H. "Surface Reconstruction by Wrapping Finite Sets in Space". *Discrete and Computational Geometry—The Goodman-Pollack Festschrift*. Springer-Verlag, 2003, 379–404 2.

[EGO*20] ERLER, PHILIPP, GUERRERO, PAUL, OHRHALLINGER, STEFAN, et al. "Points2Surf Learning Implicit Surfaces from Point Clouds". *European Conference on Computer Vision*. Springer. 2020, 108–124 2.

[EKS83] EDELSBRUNNER, H., KIRKPATRICK, D., and SEIDEL, R. "On the shape of a set of points in the plane". *IEEE Transactions on Information Theory* 29.4 (1983), 551–559 2.

[EM92] EDELSBRUNNER, HERBERT and MÜCKE, ERNST P. "Three-dimensional alpha shapes". *VVS '92: Proc. 1992 workshop on Volume vis.* Boston, Massachusetts, United States: ACM, 1992, 75–82 2.

[FR02] FUNKE, STEFAN and RAMOS, EDGAR A. "Smooth-surface reconstruction in near-linear time". *SODA '02: Proc. 13th ann. ACM-SIAM SODA*. San Francisco, California: SIAM, 2002, 781–790 2.

[GKS00] GOPI, M., KRISHNAN, S., and SILVA, C.T. "Surface Reconstruction based on Lower Dimensional Localized Delaunay Triangulation". *Computer Graphics Forum* 19.3 (2000), 467–478 2.

[GKUP11] GSCHWANDTNER, MICHAEL, KWITT, ROLAND, UHL, ANDREAS, and PREE, WOLFGANG. "BlenSor: Blender sensor simulation toolbox". *Advances in Visual Computing: 7th International Symposium, ISVC 2011, Las Vegas, NV, USA, September 26-28, 2011. Proceedings, Part II 7*. Springer. 2011, 199–208 6.

[GO08] GUIBAS, LEONIDAS and OUDOT, STEVE. "Reconstruction Using Witness Complexes". *Discrete & Computational Geometry* 40 (3 2008), 325–356. ISSN: 0179-5376 2.

[HK06] HORNUNG, ALEXANDER and KOBBELT, LEIF. "Robust reconstruction of watertight 3 d models from non-uniformly sampled point clouds without normal information". *Symposium on geometry processing*. 2006, 41–50 2.

[HMGC20] HANOCKA, RANA, METZER, GAL, GIRYES, RAJA, and COHEN-OR, DANIEL. "Point2Mesh: a self-prior for deformable meshes". *arXiv preprint arXiv:2005.11084* (2020) 2.

[HWW*22] HUANG, ZHANGJIN, WEN, YUXIN, WANG, ZIHAO, et al. "Surface reconstruction from point clouds: A survey and a benchmark". *arXiv preprint arXiv:2205.02413* (2022) 6, 8, 10–12.

[JPT20a] JAMIN, CLÉMENT, PION, SYLVAIN, and TEILLAUD, MONIQUE. "3D Triangulation Data Structure". *CGAL User and Reference Manual*. 5.2. CGAL Editorial Board, 2020. URL: https://doc.cgal.org/5.2/Manual/packages.html#PkgTDS3 10.

[JPT20b] JAMIN, CLÉMENT, PION, SYLVAIN, and TEILLAUD, MONIQUE. "3D Triangulations". *CGAL User and Reference Manual*. 5.2. CGAL Editorial Board, 2020. URL: https://doc.cgal.org/5.2/Manual/packages.html#PkgTriangulation3 10.

[KBH06] KAZHDAN, MICHAEL, BOLITHO, MATTHEW, and HOPPE, HUGUES. "Poisson surface reconstruction". *Proc. 4th Eurographics symp. on Geometry processing*. SGP '06. 2006 2.

[Kós01] KÓS, GÉZA. "An Algorithm to Triangulate Surfaces in 3D Using Unorganised Point Clouds". *Geometric Modelling*. London, UK: Springer-Verlag, 2001, 219–232. ISBN: 3-211-83603-9 2.

[KR85] KIRKPATRICK, DAVID G and RADKE, JOHN D. "A framework for computational morphology". *Machine Intelligence and Pattern Recognition*. Vol. 2. Elsevier, 1985, 217–248 2.

[KSO04] KOLLURI, RAVIKRISHNA, SHEWCHUK, JONATHAN RICHARD, and O'BRIEN, JAMES F. "Spectral surface reconstruction from noisy point clouds". *Proc. SGP 2004*. 2004, 11–21 2.

[LLZ21] LV, CHENLEI, LIN, WEISI, and ZHAO, BAOQUAN. "Voxel structure-based mesh reconstruction from a 3D point cloud". *IEEE Transactions on Multimedia* 24 (2021), 1815–1829 2.

[LPK09] LABATUT, P., PONS, J.-P., and KERIVEN, R. "Robust and Efficient Surface Reconstruction From Range Data". *Comp. Graphics Forum* 28.8 (2009), 2275–2290. ISSN: 1467-8659 2.

[OM13] OHRHALLINGER, STEFAN and MUDUR, SUDHIR. "An efficient algorithm for determining an aesthetic shape connecting unorganized 2d points". *Computer Graphics Forum*. Vol. 32. 8. Wiley Online Library. 2013, 72–88 10.

[OMW13] OHRHALLINGER, STEFAN, MUDUR, SUDHIR, and WIMMER, MICHAEL. "Minimizing edge length to connect sparsely sampled unstructured point sets". *Computers & graphics* 37.6 (2013), 645–658 2.

[OMW16] OHRHALLINGER, S., MITCHELL, S. A., and WIMMER, M. "Curve Reconstruction with Many Fewer Samples". *Proceedings of the Symposium on Geometry Processing*. SGP '16. 2016, 167–176 2, 5.

[OPP*21] OHRHALLINGER, STEFAN, PEETHAMBARAN, JIJU, PARAKKAT, AMAL D, et al. "2D Points Curve Reconstruction Survey and Benchmark". *Computer Graphics Forum*. Vol. 40. 2. 2021 5.

[PB01] PETITJEAN, SYLVAIN and BOYER, EDMOND. "Regular and Non-Regular Point Sets: Properties and Reconstruction". *Comput. Geom. Theory Appl.* 19 (2001), 101–126 2.

[PM16] PARAKKAT, AMAL DEV and MUTHUGANAPATHY, RAMANATHAN. "Crawl through Neighbors: A Simple Curve Reconstruction Algorithm". *Computer Graphics Forum* (2016) 2, 5.

[PMM18] PARAKKAT, AMAL DEV, METHIRUMANGALATH, SUBHASREE, and MUTHUGANAPATHY, RAMANATHAN. "Peeling the longest: A simple generalized curve reconstruction algorithm". *Computers & Graphics* 74 (2018), 191–201. ISSN: 0097-8493 2, 5.

[PPT*19] PEETHAMBARAN, J., PARAKKAT, A.D., TAGLIASACCHI, A., et al. "Incremental Labelling of Voronoi Vertices for Shape Reconstruction". *Computer Graphics Forum* 38.1 (2019), 521–536 2.

[PY20] PION, SYLVAIN and YVINEC, MARIETTE. "2D Triangulation Data Structure". *CGAL User and Reference Manual*. 5.2. CGAL Editorial Board, 2020. URL: https://doc.cgal.org/5.2/Manual/packages.html#PkgTDS2 10.

[RLG*20] RAKOTOSAONA, MARIE-JULIE, LA BARBERA, VITTORIO, GUERRERO, PAUL, et al. "PointCleanNet: Learning to Denoise and Remove Outliers from Dense Point Clouds". *Computer Graphics Forum* 39.1 (2020), 185–203 6, 8, 12.

[RS07] RAMOS, EDGAR A. and SADRI, BARDIA. "Geometric and topological guarantees for the WRAP reconstruction algorithm". *Proceedings of the 18th annual ACM-SIAM symposium on Discrete algorithms*. SODA '07. Philadelphia, PA, USA, 2007, 1086–1095 2.

[Rup93] RUPPERT, JIM. "A new and simple algorithm for quality 2-dimensional mesh generation". *Proc. 4th ann. ACM-SIAM SODA*. SODA '93. Philadelphia, PA, USA: SIAM, 1993, 83–92 2.

[SO20] SHARP, NICHOLAS and OVSJANIKOV, MAKS. "Pointtrinet: Learned triangulation of 3d point sets". *European Conference on Computer Vision*. Springer. 2020, 762–778 2.

[TL94] TURK, GREG and LEVOY, MARC. "Zippered polygon meshes from range images". *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*. 1994, 311–318 10.

[WWX*22] WANG, PENGFEI, WANG, ZIXIONG, XIN, SHIQING, et al. "Restricted delaunay triangulation for explicit surface reconstruction". *ACM Transactions on Graphics* 41.5 (2022), 1–20 2.

[YLL*20] YOU, CHENG CHUN, LIM, SENG POH, LIM, SENG CHEE, et al. "A Survey on Surface Reconstruction Techniques for Structured and Unstructured Data". *2020 IEEE Conference on Open Systems*. 2020 2.

[Yvi20] YVINEC, MARIETTE. "2D Triangulation". *CGAL User and Reference Manual*. 5.2. CGAL Editorial Board, 2020. URL: https://doc.cgal.org/5.2/Manual/packages.html#PkgTriangulation2 10.

[ZJ16] ZHOU, QINGNAN and JACOBSON, ALEC. "Thingi10k: A dataset of 10,000 3d-printing models". *arXiv preprint arXiv:1605.04797* (2016) 6, 10.

[ZYT23] ZHANG, CHEN, YUAN, GANZHANGQIN, and TAO, WENBING. "DMNet: Delaunay Meshing Network for 3D Shape Representation". *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE Computer Society. 2023, 14372–14382 2.