Reconstruction using a simple triangle removal approach

Subhasree Methirumangalath
Indian Institute of Technology Madras

Shyam Sundar Kannan
Indian Institute of Technology Madras

Amal Dev Parakkat
Indian Institute of Technology Madras

Ramanathan Muthuganapathy*
Indian Institute of Technology Madras

ABSTRACT

Given a finite set of points $P \subseteq \mathbb{R}^3$, sampled from a surface S, surface reconstruction problem computes a model of S from P, typically in the form of a triangle mesh. The problem is ill-posed as various models can be reconstructed from a given point set. In this paper, curve reconstruction in $\mathbb{R}^2$, is initially looked at using the Delaunay triangulation (DT) of a point set. The key idea is that the edges in the DT are prioritized and the interior or exterior edges of the DT are removed as long as it has at least one adjacent triangle. Theoretically, it is shown that the reconstruction is homeomorphic to a simple closed curve. Extending this to 3D, an approach based on 'retaining solitary triangles' and 'removing triangles anywhere' has been proposed. An additional constraint based on the circumradius of a triangle has been employed. Results on public and real-world scanned data, and qualitative/quantitative comparisons with existing methods show that our approach handles diverse features, outliers and noise better or comparable with other methods.

CCS CONCEPTS

• **Computing methodologies → Shape modeling**;

KEYWORDS

Surface reconstruction, Point set, Delaunay triangulation

ACM Reference format:

Subhasree Methirumangalath, Shyam Sundar Kannan, Amal Dev Parakkat, and Ramanathan Muthuganapathy. 2017. Reconstruction using a simple triangle removal approach. In *Proceedings of SIGGRAPH Asia 2017 Technical Briefs, Bangkok, Thailand, November 27–30, 2017 (SA '17 Technical Briefs),* 4 pages.
https://doi.org/10.1145/3145749.3149447

1 INTRODUCTION

Given a finite sampling $P \subseteq \mathbb{R}^3$ of an unknown surface S, surface reconstruction problem computes a model of S from P, which is expected to match S in terms of both geometrical and topological properties [Edelsbrunner 1998]. The problem is an ill-posed one as there can be numerous models reconstructed from the same point set. The challenges of the problem are sparsity, noisiness and outliers present in the sampling. Reconstruction has applications in diverse fields such as reverse engineering, product design, computer graphics, etc [Berger et al. 2016].

1.1 Related Work

The reconstruction methods can be classified into two categories, namely implicit and explicit methods.

Implicit methods include Algebraic Point Set Surfaces (*APSS*) [Guennebaud and Gross 2007], Robust Implicit Moving Least Squares (*RIMLS*) [Öztireli et al. 2009], Screened Poisson (*SP*) [Kazhdan and Hoppe 2013] etc. The implicit methods are : (i) generally faster but require normal information, a computationally complex task. (ii) require multiple parameter tuning, a time consuming and tedious process. (iii) guarantee convergence to a local minimum, however, it might be different from the original surface and also may not pass through all the input points, leading to loss of details.

Explicit methods triangulate the points directly and normal informations are not required. They can be divided into two groups (i) Region growing (Ball Pivoting Algorithm (*BPA*) [Bernardini et al. 1999]) and (ii) Delaunay triangulation (DT)/Voronoi diagram (VD) methods (Power Crust (*PC*) [Amenta et al. 2001], Robust Cocone (*RC*) [Dey and Goswami 2004], Singular Cocone (*SC*) [Dey et al. 2012], Shape Hull (*SH*) [Peethambaran and Muthuganapathy 2015] etc.). The region growing methods are faster, but they are not robust and not easy to generalize. They degrade when two surfaces are close together or near sharp features and multiple parameters tuning is needed, a tedious task. DT/VD based algorithms do not require normal information but most require multiple parameter tuning and are slower. Only a few have handled noisy point set and outliers. For a recent survey on surface reconstruction, please refer [Berger et al. 2016].

In this paper, we present an algorithm for reconstruction based on DT of the input point set. The key difference over the existing approaches is that the removal of an edge for curve reconstruction is based on adjacency of triangles associated with the edge. This approach enables an edge to be removed from anywhere in the DT as opposed to orderly removal in sculpting methods. The approach has then been extended to surface reconstruction, where a triangle is removed from anywhere using the idea of 'solitary triangles' and a single parameter based on circumradius of a triangle in a tetrahedron. The output surface is called as Surface Reconstructed from Solitary Triangles, *SRST*.

2 CURVE RECONSTRUCTION

Consider a simple closed curve $\mathfrak{C}$ (Figure 1a) and the sampled points P (red in Figure 1b) from $\mathfrak{C}$. In most of the DT-based curve reconstruction approaches, the exterior edges of the DT (i.e., the edges

*Corresponding author, email: emry01@gmail.com

SA '17 Technical Briefs, November 27–30, 2017, Bangkok, Thailand

ACM ISBN 978-1-4503-5406-6/17/11…$15.00
https://doi.org/10.1145/3145749.3149447

that share only one triangle) are prioritized and removed successively to obtain a resultant graph. Removing only exterior edges might lead to the following scenario: even if there are interior edges which are eligible to be removed, the exterior edges cause blockages due to the edge removal rules[Boissonnat 1984]. For the DT in Figure 1b, the graph obtained after removal of a few of the exterior edges is shown in Figure 1c. Figure 1d shows an exterior edge e which is not removed and causes blockage. The reconstructed shape is as shown in Figure 1e, where the concave portion of $\mathfrak{C}$ is not captured. This motivated us to look into removing the edges of DT from anywhere, either it is an exterior or an interior edge (*removal anywhere* strategy).

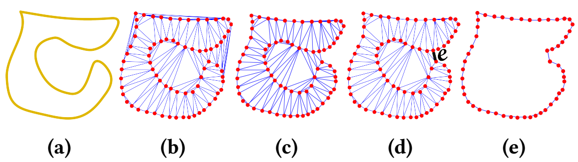

Figure 1: (a) Curve $\mathfrak{C}$ (b) DT of the sample points P (in red) (c) After removing a few of the exterior edges (d) Edge e causing blockage (e) Boundary edges which do not capture the concave portion of $\mathfrak{C}$.

It can be observed that, based on its adjacency, an edge of DT can be categorized as: (i) it is part of only one triangle (such as e_{pq} in Figure 2a) or (ii) it is shared by a maximum of two triangles (eg: e_{ij} in Figure 2a).

Definition 1. *An edge is known as a solitary edge if it is not part of any triangle.*

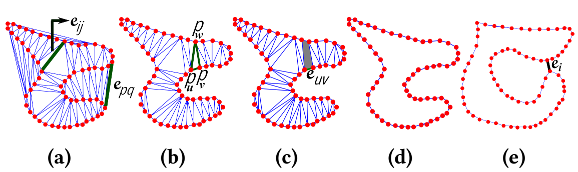

Figure 2: (a) e_{pq} part of a triangle, e_{ij} shared by two triangles (b) $\triangle_{uvw}$ (c) Solitary edge e_{uv} (not part of any triangle), obtained after removing e_{uw} from $\triangle_{uvw}$ of Figure 2b (d) Solitary edges as boundary. (e) a singular edge e_i.

Consider $\triangle_{uvw}$ in a graph (Figure 2b). Figure 2c shows the graph obtained after removing e_{uw}. It can be observed that e_{uv} is no more part of a triangle (Note that the shaded area is not a triangle) and it is an example of a solitary edge. All the other edges which are part of at least one triangle (Figure 2c) are non-solitary edges.

The algorithm for curve reconstruction processes for solitary edges from DT, using adjacency information of the edges. An edge e_{ij} of a triangle can be retained in DT only if e_{ij} is a solitary edge. Similarly, an edge of a triangle can be removed from DT, only if it is a non-solitary edge or a singular edge (Refer Definition 2). The resultant reconstructed simple closed curve is represented as $\mathcal{G}$. An example of $\mathcal{G}$ is as shown in Figure 2d and one can observe that $\mathcal{G}$ has only solitary edges. To the best of our knowledge, the approach of identifying solitary edges based on adjacency is a novel one, not employed in any other DT-based ones.

Assuming P is a sample obtained from an input curve under ϵ-sampling (a sufficiently dense sampling), in $DT(P)$, it has been observed that the edges on the boundary of $\mathcal{G}$ (if any, in a triangle) are shorter than the non-boundary edges. Hence, from $DT(P)$, all the edges are prioritized in the descending order of the length of the edges. Based on the priority, the edges which are non-solitary are removed either it is an interior or an exterior edge from $DT(P)$, retaining all solitary edges in $\mathcal{G}$.

After retaining all the solitary edges, $\mathcal{G}$ may contain edges between non-adjacent points (edge e_i in Figure 2e), which have to be removed to obtain the best perceived shape.

Definition 2. *A singular edge is a solitary edge between two non-adjacent points (edge e_i in Figure 2e).*

From Figure 2e, it can be observed that all the points in $\mathcal{G}$ (reconstructed shape from the points which are sampled from a simple closed curve) have degree as two, except the end points of a singular edge. It can also be noted that both the end points of a singular edge have more than two as degree. In order to compute the final reconstructed shape, singular edges are removed from $\mathcal{G}$ using this graph theoretic property. Theoretical guarantee of the curve reconstruction is given in the supplementary document.

3 SURFACE RECONSTRUCTION

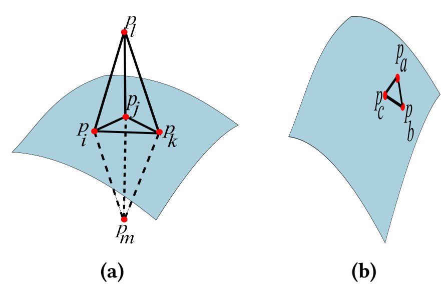

Figure 3: (a) Adjacent triangles - $\triangle_{ijk}$ & $\triangle_{ijl}$ in TET_{ijkl}, $\triangle_{ijk}$ is one of the non-solitary triangles due to existence of its adjacent triangles (b) Solitary triangle - $\triangle_{abc}$.

Definition 3. *A solitary triangle (ST) is a triangle if it is not part of any tetrahedron.*

In Figure 3a, $\triangle_{ijk}$ is non-solitary as it is part of TET_{ijkl} whereas in Figure 3b, $\triangle_{abc}$ is a solitary triangle. Let $\triangle_s$ denotes the triangle with smallest circumradius (say, r_0), on the convex hull of P.

For a reconstructed surface to be homeomorphic to a closed surface, one can observe that all the triangles have to be solitary. In Figure 4a, on a surface, only one triangle from the tetrahedron is solitary (the smallest circumradii one). In Figure 4b, two of the triangles from a tetrahedron will be on the surface (shaded in yellow in Figure 4c) if their circumradii are smaller than the other two. There can be three triangles from a tetrahedron forming part of a surface. We conjecture that ϵ-sampling can lead to such a point-set (similar to that in 2D). However, in practise, a point-set need not

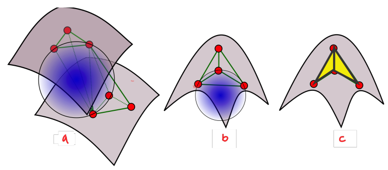

Figure 4: (a) Single solitary triangle on a surface (b) A tetrahedron from which two solitary triangles are on a smooth surface (yellow ones in (c).

confirm to such a sampling, and hence we decided to introduce a parameter ϑ. If circumradius of a triangle is within the range of $(0, \vartheta * r_0]$ (where $\vartheta > 0$), then that triangle has to be retained.

Definition 4. *A triangle is not-retainable, if it is non-solitary and its circumradius does not lie in the range of* $(0, \vartheta * r_0]$. *Hanging triangle (akin to singular edge in 2D) is a triangle which has at least one unshared edge.*

The algorithm for Surface Reconstruction is as follows: From $DT(P)$, the triangles are processed in the descending order of the circumradius. If it is a retainable triangle, it is added to *SRST*. On the other hand, a triangle is removed from *DT* if it is not-retainable.

Algorithm 1 SURFACE_RECONSTRUCTION(P)

```
Input point set, P
Output surface, SRST
Construct 3D Delaunay triangulation DT
SRST = ∅
Compute r_0
Construct a priority queue PQ with triangular faces in descending order of the circumradius
while PQ ≠ ∅ do
    △_ijk = POP(PQ)
    if NOT_RETAINABLE(△_ijk, DT, r_0) then
        Remove △_ijk from DT
    else
        if SRST ⋃ △_ijk forms a tetrahedron TET_ijkl then
            Remove triangle with largest circumradius of TET_ijkl from SRST
            SRST = SRST ⋃ △_ijk
        end if
    end if
end while
Remove hanging triangles (using the adjacency information) from SRST
return SRST
```

Algorithm 1 presents the pseudo code of the proposed surface reconstruction algorithm. The function $NOT\_RETAINABLE$ checks whether $\triangle_{ijk}$ is shared with any of its six (at most) adjacent triangles of two (at most) neighbouring tetrahedra and whether the circumradius of $\triangle_{ijk}$ is within the range of $(0, \vartheta * r_0]$.

4 RESULTS AND DISCUSSION

Figure 5 shows the results (implemented using CGAL 4.6) for publicly available data and (Results for real-world scanned data and for large data (close to five million) are shown in the supplementary document). For each of the results, the number of points and ϑ are shown in the bottom. Qualitatively (Figure 6), we compared our approach with the following - APSS, RIMLS, SP, BPA, PC, RC, SC, and SH. The algorithm is able to capture sharp features and also works for multiple genus objects, comparable or better than other algorithms for outlier and down sampled ones. For noisy models (created using ReMesh 2.1) extra triangles are present in our result (overall, it has still captured the essence of the output models). For a real data with noise, our algorithm has performed quite well. BPA, RIMLS and APSS results have been obtained using Meshlab's plugin (with 'Projection - Max iterations' set to zero for RIMLS and APSS for noise and outliers).

Figure 5: *SRST* for AIM@SHAPE data set with number of points and ϑ. Detailed features, genus, sharp features and concavities are captured.

Quantitatively, the RMS error for Hausdorff distance computed on reconstruction on input point sets, point sets with noise and that with outliers shows that our simple approach shows a better or equal performance (Figure 7). Table 1 shows that *SRST* has less running time (for benchmark models [Berger et al. 2013]) than *SH*, *PC*, *SC* and *RC* (parameter tuning took lesser time than others).

Table 1: Running time with number of points

Models	# Points	Running Time (seconds)								
		BPA	*SP*	*RIMLS*	*APSS*	*SRST*	*SH*	*PC*	*SC*	*RC*
Anchor	30644	0.77	1.52	3.74	2.7	2.97	6.98	12.17	15.5	17.7
Daratech	60319	0.79	2.82	4	4.5	6.76	8.63	18.3	23.24	39.6
Quasimoto	90716	0.99	2.87	8.71	8.89	10.62	14.24	34.5	46.91	61.61
Gargoyle	119746	3.72	4.3	15.82	19.14	14.49	16.46	42.75	63.01	78.19
Dancing Children	241016	4.72	4.17	21.7	26.5	34.88	35.05	87.76	195.93	218.57

Conclusions: Based on the insight of the 'solitary edge' for curve reconstruction, we proposed a 'removal anywhere' approach for surface reconstruction using solitary triangles. The proposed approach is capable of detecting different features such as sharp corners, multiple genus and concavities, noise and outlier without preprocessing. We performed an extensive comparative study using publicly available data and real scanned data, with the existing methods and demonstrated that our approach performs in a comparable way in many aspects. The limitation of the algorithm is that it is a parametric one, requiring a trial and error approach to determine it.

REFERENCES

Nina Amenta, Sunghee Choi, and Ravi Krishna Kolluri. 2001. The Power Crust. In *Proceedings of the Sixth ACM Symposium on Solid Modeling and Applications (SMA '01)*. ACM, New York, NY, USA, 249–266.

Matthew Berger, Joshua A. Levine, Luis Gustavo Nonato, Gabriel Taubin, and ClÃąudio T. Silva. 2013. A benchmark for surface reconstruction. *ACM Trans. Graph.* 32, 2 (2013), 20.

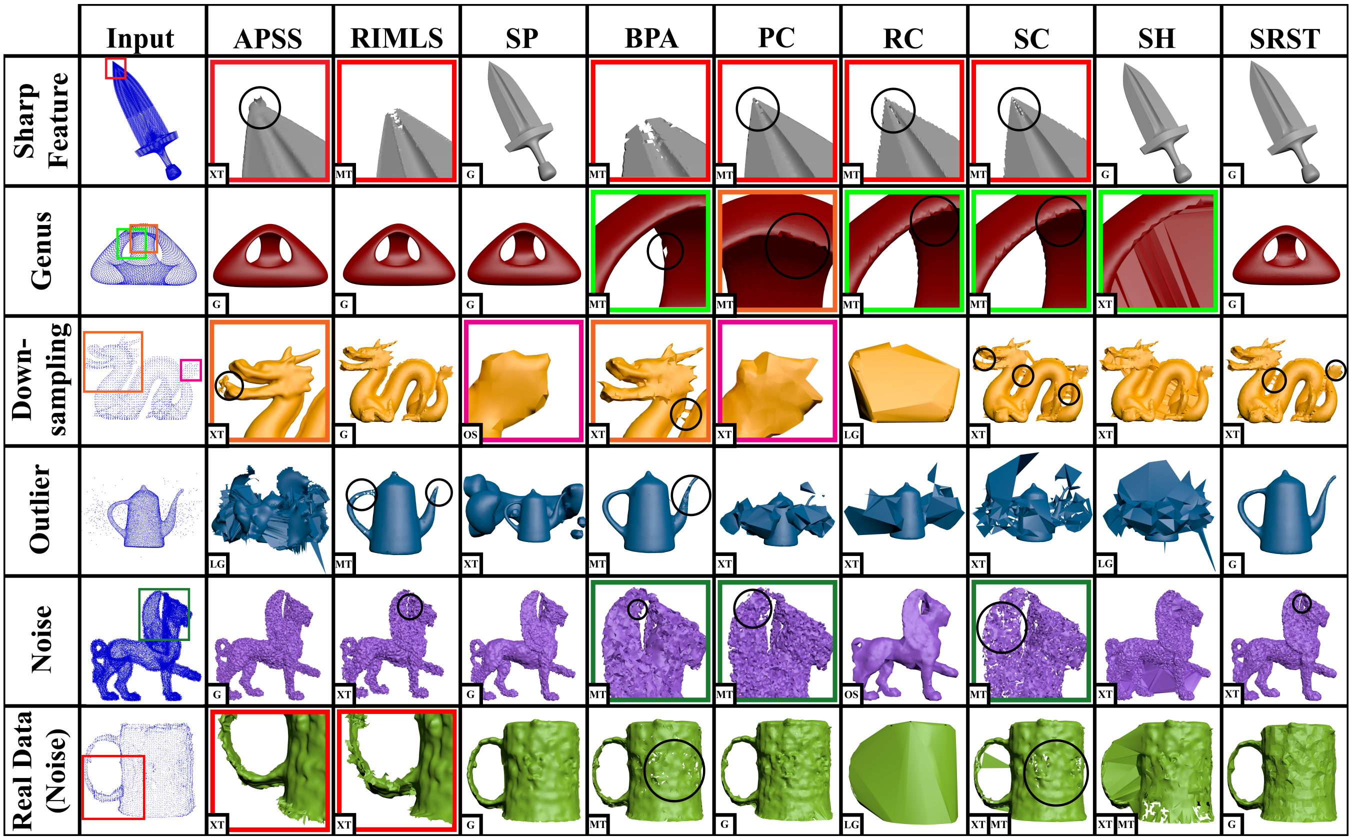

Figure 6: Comparative results with other approaches. Characterising the output; XT - Extra triangles, MT - Missing triangles, LG - Loss of geometry, OS - Over smoothing, G - Good reconstruction. Some figures have been exploded locally.

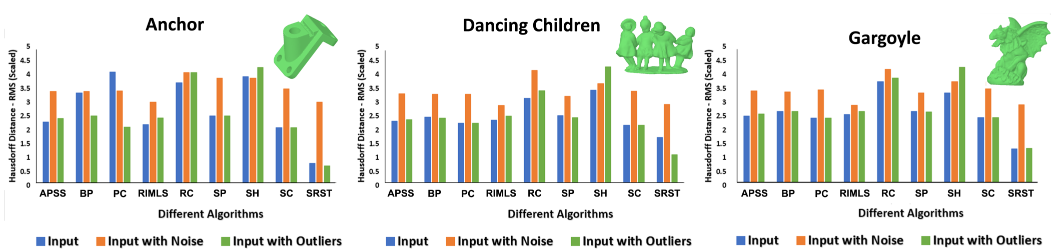

Figure 7: Using benchmark models: Bar charts of Hausdorff distance (RMS error) between the original and results of different algorithms for input point set, noisy point set and point sets with outliers. *SRST* is also shown as inset.

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

Fausto Bernardini, Joshua Mittleman, Holly Rushmeier, Claudio Silva, and Gabriel Taubin. 1999. The Ball-Pivoting Algorithm for Surface Reconstruction. *IEEE Transactions on Visualization and Computer Graphics* 5 (1999), 349–359.

Jean-Daniel Boissonnat. 1984. Geometric Structures for Three-dimensional Shape Representation. *ACM Trans. Graph.* 3, 4 (Oct. 1984), 266–286.

T. K. Dey, X. Ge, Q. Que, I. Safa, L. Wang, and Y. Wang. 2012. Feature-Preserving Reconstruction of Singular Surfaces. *Comput. Graph. Forum* 31, 5 (Aug. 2012), 1787–1796. https://doi.org/10.1111/j.1467-8659.2012.03183.x

Tamal K. Dey and Samrat Goswami. 2004. Provable Surface Reconstruction from Noisy Samples. In *Proceedings of the Twentieth Annual Symposium on Computational Geometry (SCG '04)*. ACM, New York, NY, USA, 330–339.

Herbert Edelsbrunner. 1998. Shape Reconstruction with Delaunay Complex.. In *LATIN (Lecture Notes in Computer Science)*, Claudio L. Lucchesi and Arnaldo V. Moura (Eds.), Vol. 1380. Springer, 119–132.

Gaël Guennebaud and Markus Gross. 2007. Algebraic Point Set Surfaces. *ACM Trans. Graph.* 26, 3, Article 23 (July 2007).

Michael Kazhdan and Hugues Hoppe. 2013. Screened Poisson Surface Reconstruction. *ACM Trans. Graph.* 32, 3, Article 29 (July 2013), 13 pages.

Cengiz Öztireli, Gaël Guennebaud, and Markus Gross. 2009. Feature Preserving Point Set Surfaces based on Non-Linear Kernel Regression. In *Proceedings of Eurographics 2009*. 493–501.

Jiju Peethambaran and Ramanathan Muthuganapathy. 2015. Reconstruction of water-tight surfaces through Delaunay sculpting. *Computer-Aided Design (Proc. of Solid and Physical Modeling 2014)* 58 (2015), 62 – 72.