EUROGRAPHICS 2022 / R. Chaine and M. H. Kim
(Guest Editors)

Volume 41 (2022), Number 2

Coverage Axis: Inner Point Selection for 3D Shape Skeletonization Supplementary Material

Zhiyang Dou[1], Cheng Lin[2], Rui Xu[3], Lei Yang[1], Shiqing Xin[3], Taku Komura[1], Wenping Wang[4]

[1] The University of Hong Kong
[2]Digital Content Technology Center, CROS, Tencent Games
[3]Shandong University
[4]Texas A&M University

Appendix A: Inner Points Labeling for Point Cloud with Normals.

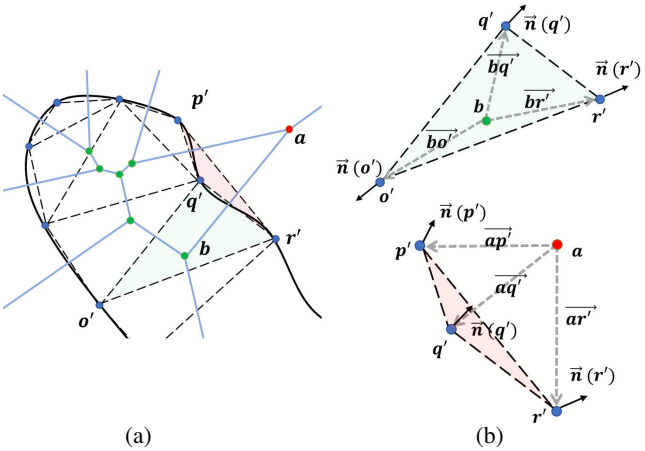


Figure 1: *Inner points labeling for an oriented point cloud in 2D. b is labelled as an inner candidate since the dot products of* $\overrightarrow{bo'} \cdot \vec{n}(o')$, $\overrightarrow{bq'} \cdot \vec{n}(q')$ *and* $\overrightarrow{br'} \cdot \vec{n}(r')$ *are all positive, while a is labeled as an outside point because* $\overrightarrow{ap'} \cdot \vec{n}(p'), \overrightarrow{aq'} \cdot \vec{n}(q')$ *and* $\overrightarrow{ar'} \cdot \vec{n}(r')$ *are negative. The Voronoi diagram and its dual graph Delaunay triangulation are denoted by blue lines and dashed lines respectively. Surface samples are in blue.*

Our point selection strategy does not rely on mesh connection; it is able to handle more generalized inputs such as polygon soups, point clouds as long as candidate inner points can be identified inside the volume. Given a point cloud input, in this paper, we generate and label candidate inner points by utilizing normal vectors of the point cloud. Recall that we first compute its Delaunay triangulation which is the dual of the Voronoi diagram w.r.t. the input point cloud. Consider a Voronoi vertex p and the vertices of its dual tetrahedron p'_0, p'_1, p'_2 and p'_3. The candidate p is considered as an inner point only if we have $\overrightarrow{pp'_i} \cdot \vec{n}(p'_i) > 0, \forall i = 0,1,2,3$. Here $\vec{n}(p'_i)$ is the input normal of p'_i and $\cdot$ is dot product. A detailed 2D example is given in Figure 1.

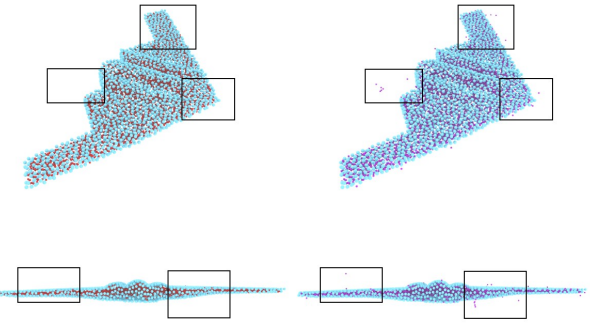
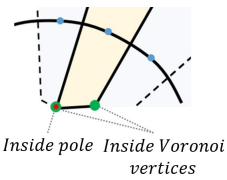


Figure 2: *Inner points labeling with filtering by clustering. (a) and (b) indicate the results after applying filtering by clustering (red points) and before applying filtering by clustering (purple points). The input point cloud is in blue. We provide both normal view (first row) and side view (second row).*

Figure 3: *Inner point candidates generated from point clouds with normals.*

After the initial labeling, we further apply filtering by clustering to those labeled inner points. For each inner point, we count the number of its neighbors within 0.02 among the top 100 nearest neighbors by K-neighbor searching [TF21]. If the result is less than 20, the point is considered an outlier and is discarded. A result of the UFO model as an example is shown in Figure 2. Some labeling results are given in Figure 3.

Besides, we also conduct an experiment taking inside poles [ACK01] (the inside Voronoi vertex with the furthest distance to the seed of the cell) instead of all Voronoi vertices enclosed by the surface as candi-



dates. We find there is no significant difference in point selection results.

Appendix B: Details of Connection Establishment.

Mesh Input

Recall that for a mesh input, we take the whole subset of Voronoi structure inside the model generated by 4000 surface samples ($|C| = 4000$) as the original inner point candidates P. After selecting inner points based on set coverage, we build up the connection structure. For the Voronoi initialization style, all selected points are embedded on the Voronoi diagram. We further remove all redundant points and edges to achieve the simplification by edge collapse using quadric error metric (QEM) [FTB13] same as [LWS*15].

After that, we adopt LOP [DLRS10] algorithm to adjust the mesh tessellations as long as the increase of the local approximation error caused by the flipping operation is less than 5%. The local approximation error is measured by the Hausdorff distance from the covered surface to candidate local reconstructed surfaces (the envelope of two candidate triangulation.), a.k.a., one-sided Hausdorff distance. Note that we first unstitch each face patch based on non-manifold edges before performing LOP.

Point Cloud Input

In order to suppress over-connection, we up-sample on the point cloud with normals following [HWG*13], leading to 30000 surface samples. These points are treated as surface points in **RT** during connection establishment. The effect of point cloud up-sampling is demonstrated in Figure 4.

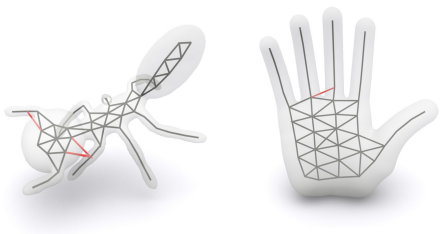


Figure 4: *Over-connection suppression by up-sampling. The over-connected structures (red edges) are suppressed after up-sampling. We visualize the original surface for illustration purposes.*

Appendix C: Parameter Analysis on Surface points and Candidate Inner Points.

We conduct comprehensive experiments on the number of surface points $|S|$ and candidate inner points $|P|$ (note it is determined by $|C|$, which is the number of samples to compute the Voronoi diagram) for the algorithm performance. Detailed results including approximation error are summarized in Table 1 and Table 2, respectively. In Table 1, we fix $|C| = 4000$ and in Table 2, we always set $|S| = 1500$. Same as the main paper, we use $|V|$ to denote the number of skeletal points, $\overrightarrow{\varepsilon}$ to denote the HD from surface to reconstruction, $\overleftarrow{\varepsilon}$ to denote the HD from reconstruction to surface,

Table 1: *Effect of different surface point numbers $|S|$ on selected points and time efficiency.*

$\lvert S\rvert$	$\lvert V\rvert$	Time (s)	$\overrightarrow{\varepsilon}$	$\overleftarrow{\varepsilon}$	$\overleftrightarrow{\varepsilon}$
500	47	0.25	2.062%	1.981%	2.062%
1500	49	0.29	1.843%	1.884%	1.884%
2500	49	0.34	1.761%	1.799%	1.799%
3500	49	0.73	2.031%	1.937%	2.031%
4500	49	0.98	1.750%	1.762%	1.762%
5500	49	1.33	1.883%	1.862%	1.883%
6500	49	2.44	1.701%	1.747%	1.747%
7500	49	3.24	1.810%	1.894%	1.894%
8500	49	4.22	1.624%	1.670%	1.670%

Table 2: *Effect of different numbers of inner point candidates $|P|$ on selected points and time efficiency. C denotes surface samples used for generating candidate inner points.*

$\lvert C\rvert$	$\lvert P\rvert$	$\lvert V\rvert$	Time (s)	$\overrightarrow{\varepsilon}$	$\overleftarrow{\varepsilon}$	$\overleftrightarrow{\varepsilon}$
1000	2964	48	0.08	1.982%	2.196%	2.196%
2000	6894	49	0.18	1.843%	1.811%	1.843%
3000	10776	49	0.28	1.905%	1.952%	1.952%
4000	13432	49	0.31	1.855%	1.894%	1.894%
5000	16749	49	0.49	1.674%	1.744%	1.744%
6000	22301	49	0.82	1.675%	1.735%	1.735%
7000	26161	49	1.23	1.765%	1.807%	1.807%
8000	29859	49	1.43	1.754%	1.916%	1.916%
9000	33651	49	1.65	1.936%	1.948%	1.948%

and $\overleftrightarrow{\varepsilon}$ to represent two-sided HD. We find both $|S|$ and $|P|$ do not show a large influence on the selected point number as well as the approximation error. However, as we mentioned in the main paper, the two factors, especially the number of surface samples $|S|$, have more effect on the time efficiency.

Appendix D: Comparison with Q-MAT for Highly Decimated MAT Computation.

Table 3: *Evaluations on approximation accuracy for highly decimated medial surfaces between Q-MAT and Coverage Axis. We adopt the FEMUR model for evaluation.*

Offset δ_r	$\lvert V\rvert$	Q-MAT			Coverage Axis		
		$\overrightarrow{\varepsilon}$	$\overleftarrow{\varepsilon}$	$\overleftrightarrow{\varepsilon}$	$\overrightarrow{\varepsilon}$	$\overleftarrow{\varepsilon}$	$\overleftrightarrow{\varepsilon}$
0.01	47	1.015%	1.114%	1.114%	0.947%	1.104%	**1.104**%
0.02	24	1.545%	1.281%	**1.545**%	1.535%	1.593%	1.593%
0.05	13	1.779%	1.783%	**1.783**%	2.576%	2.534%	2.576%
0.1	7	2.679%	2.480%	**2.679**%	6.086%	4.218%	6.086%
0.2	3	7.565%	4.099%	**7.565**%	12.434%	4.271%	12.434%
0.5	2	29.855%	5.743%	**29.855**%	32.993%	2.919%	32.993%

The quantitative comparison of surface reconstruction by highly decimated medial surfaces between Q-MAT and Coverage Axis is shown in Table 3.

References

[ACK01] AMENTA N., CHOI S., KOLLURI R. K.: The power crust. In *Proceedings of the sixth ACM symposium on Solid modeling and applications* (2001), pp. 249–266. 1

[DLRS10] DE LOERA J., RAMBAU J., SANTOS F.: *Triangulations: Structures for algorithms and applications*, vol. 25. Springer Science & Business Media, 2010. 2

[FTB13] FARAJ N., THIERY J.-M., BOUBEKEUR T.: Progressive medial axis filtration. In *SIGGRAPH Asia 2013 Technical Briefs*. 2013, pp. 1–4. 2

[HWG*13] HUANG H., WU S., GONG M., COHEN-OR D., ASCHER U., ZHANG H.: Edge-aware point set resampling. *ACM transactions on graphics (TOG) 32*, 1 (2013), 1–12. 2

[LWS*15] LI P., WANG B., SUN F., GUO X., ZHANG C., WANG W.: Q-mat: Computing medial axis transform by quadratic error minimization. *ACM Transactions on Graphics (TOG) 35*, 1 (2015), 1–16. 2

[TF21] TANGELDER H., FABRI A.: dD spatial searching. In *CGAL User and Reference Manual*, 5.3 ed. CGAL Editorial Board, 2021. 1