Parameter-Free and Improved Connectivity for Point Clouds

D. Marin[1] and S. Ohrhallinger[1] and M. Wimmer[1]

[1]Institute of Visual Computing & Human-Centered Technology, TU Wien, Austria

Abstract

Determining connectivity in unstructured point clouds is a long-standing problem that is still not addressed satisfactorily. In this poster, we propose an extension to the proximity graph introduced in [MOW22] to three-dimensional models. We use the spheres-of-influence (SIG) proximity graph restricted to the 3D Delaunay graph to compute connectivity between points. Our approach shows a better encoding of the connectivity in relation to the ground truth than the k-nearest neighborhood (kNN) for a wide range of k values, and additionally, it is parameter-free. Our result for this fundamental task offers potential for many applications relying on kNN, e.g., improvements in normal estimation, surface reconstruction, motion planning, simulations, and many more.

CCS Concepts

• ***Computing methodologies*** → ***Point-based models;***

1. Introduction

Retrieving connectivity from points without a known surface has been a long-established research problem in computer graphics. It is a necessary precondition in surface reconstruction since some type of connectivity is required in order to be able to create a triangulated surface from points. Usually, this problem is solved by computing a graph and filtering some of the edges to retain the best encoding connectivity. We propose the 3D version of SIGDT [MOW22], which is a restriction of the *spheres-of-influence* proximity graph to the 3D Delaunay graph, and name it SIGDT3D. This graph recovers the original connectivity of the surface better than the commonly used kNN graph while dropping the need for users to search for a suitable parameter that typically varies depending on the input.

2. Related Work

For determining the neighborhood of unstructured points in 3D, the kNN method is widely used, e.g., when computing the tangent plane for connectivity or normals [HDD*92]. Many subsequent surface reconstruction methods rely on kNN, up to recent deep learning methods that require it for training purposes, e.g., [EGO*20].

The *spheres-of-influence* graph (SIG) has been introduced in [Tou88] as a clustering method. Two vertices are connected in SIG if the distance between them is less or equal to the sum of the distances to their respective nearest neighbours. The rarely used SIG has recently gotten attention for its improved connectivity when reconstructing curves from unstructured points [MOW22].

3. Method

Our algorithm first computes the 3D Delaunay graph on the input points. We then filter the triangles with at least 2 SIG edges. The only information required for this computation is the distance to the nearest neighbour, and the Delaunay triangulation already contains these edges. Hence, we pass over all edges in order to save the nearest neighbour distances at the vertices in $O(NlogN)$ time for N vertices, without increasing time or space complexity to the Delaunay graph computation. Then, we apply the SIG criterion to only keep the Delaunay edges that satisfy the condition.

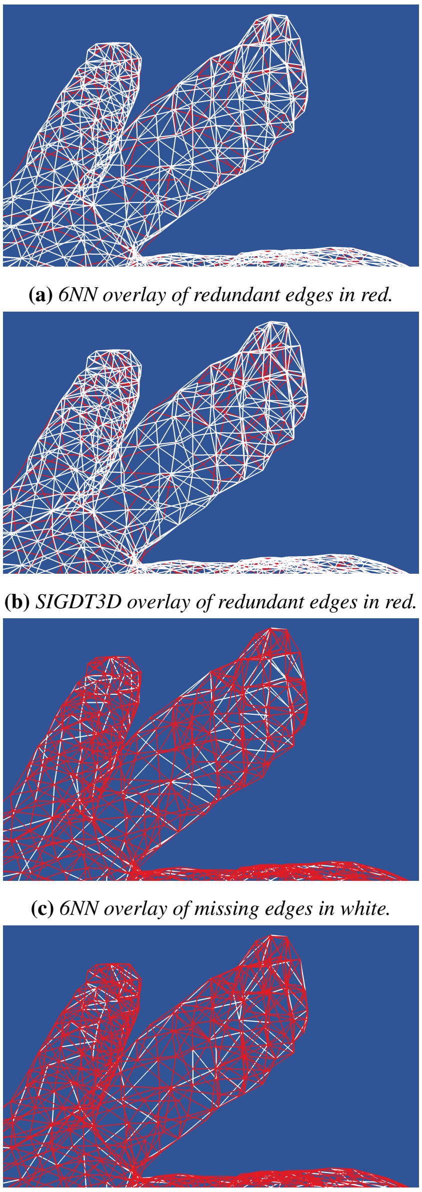
4. Results

We have tested SIGDT3D on various models at different resolutions from the Stanford Scanning Repository. We have also computed the kNN graphs for typical values used in points' connectivity, $k = 6,8,12$, and extracted the edges of the original reconstruction. In order to assess the quality of the connectivity encoding, we computed the intersection over union between the obtained and the original edge set - $IoU = (NewEdges \cap OriginalEdges)/(NewEdges \cup OriginalEdges)$. Using IoU, 1 means a perfect match while 0 means no overlap at all - Figure 1.

For almost all 3D models, SIGDT3D had a higher (better) IoU value than all of the kNN graphs. This value decreases significantly when k is increased, due to the number of extra edges that are not contained in the ground truth surface. Detailed results are presented in Table 1. Furthermore, Table 2 also highlights the fact that SIGDT3D managed to capture proximity well with a low amount of edges, having an average degree of about 6 over multiple models.

However, our method proves to be slower due to the need to

DOI: 10.2312/egp.20231023



(a) *6NN overlay of redundant edges in red.*

(b) *SIGDT3D overlay of redundant edges in red.*

(c) *6NN overlay of missing edges in white.*

(d) *SIGDT3D overlay of missing edges in white.*

Figure 1: *Visual results of IoU for a sparsely sampled ears region of bunny_zipper_res3: the original edges are in white, while the reconstructed ones are in red. We overlay them to show the number of redundant/missing edges compared to the original.*

compute the Delaunay triangulation. We have tested our method against kNN, with $k = 6, 8, 12$, on the models provided in the Stanford repository, and our method is on average 3 times slower than the kNN neighbourhood. Both implementations took advantage of CGAL's parallel of Delaunay triangulation and the nearest neighbourhood search respectively. However, the Delaunay triangulation accounts, on average, for 60% of the computation, which indicates the possibility of improvement in our code, but clearly not above the performance of kNN.

Model	6-*IoU*	8-*IoU*	12-*IoU*	SIGDT-*IoU*
bun_zipper_res2	0.69	0.65	0.47	**0.72**
bun_zipper_res3	0.68	0.64	0.46	**0.7**
bun_zipper_res4	**0.66**	0.61	0.46	0.65
dragon_vrip_res2	0.67	0.62	0.46	**0.68**
dragon_vrip_res3	0.68	0.64	0.47	**0.7**
dragon_vrip_res4	0.67	0.63	0.47	**0.7**
happy_vrip_res2	**0.66**	0.61	0.45	**0.66**
happy_vrip_res3	0.68	0.64	0.47	**0.69**
happy_vrip_res4	**0.68**	0.64	0.47	**0.68**

Table 1: *Intersection over union (IoU) computed on various models in kNN with $k = 6, 8, 12$ and in SIGDT3D.*

Model	#Vertices	SIGDT avg. deg.
bun_zipper_res2	8171	6.66
bun_zipper_res3	1889	6.84
bun_zipper_res4	453	6.88
dragon_vrip_res2	100250	5.84
dragon_vrip_res3	22998	6.48
dragon_vrip_res4	5205	6.73
happy_vrip_res2	144647	5.78
happy_vrip_res3	32328	6.52
happy_vrip_res4	7108	6.89

Table 2: *The average degree in SIGDT3D, computed over various models with different numbers of input vertices.*

5. Conclusion and Future Work

We propose an extension of the SIGDT proximity graph to 3D. This graph shows significantly better proximity encoding, while keeping a low number of edges and not requiring a parameter. Thus, it offers two important advantages over kNN. The current results also encourage more experiments with the many other applications of kNN in improvements in normal estimation, surface reconstruction, motion planning, learning from 3D data, simulations, and many more.

References

[EGO*20] Erler P., Guerrero P., Ohrhallinger S., Mitra N. J., Wimmer M.: Points2surf learning implicit surfaces from point clouds. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V* (2020), Springer, pp. 108–124. 1

[HDD*92] Hoppe H., DeRose T., Duchamp T., McDonald J., Stuetzle W.: Surface reconstruction from unorganized points. In *Proceedings of the 19th annual conference on computer graphics and interactive techniques* (1992), pp. 71–78. 1

[MOW22] Marin D., Ohrhallinger S., Wimmer M.: Sigdt: 2d curve reconstruction. *Computer Graphics Forum 41*, 7 (Oct. 2022), 25–36. doi:10.1111/cgf.14654. 1

[Tou88] Toussaint G. T.: A graph-theoretical primal sketch. In *Machine Intelligence and Pattern Recognition*, vol. 6. Elsevier, 1988, pp. 229–260. 1