International Cartographic Association
Association Cartographique Internationale

Exploring Spatial Voronoi Diagrams for Autonomous Vehicle Operation in 3D Road Networks

Ganesh P Kumar [a]*, Md. Nahid Pervez [a], Quinn Wu [a], Sharnam Shah [a]

[a] *Ford Greenfield Labs, Palo Alto, CA 94304, USA. {gkumar29, npervez2, qwu44, sshah89}@ford.com*

* Corresponding author

Keywords: Autonomous Vehicle (AV), 3D mapping, localization, motion planning, Voronoi diagram

Abstract:

Voronoi diagrams are convex subdivisions of Euclidean space and have been extensively studied in robotic mapping, localization, and motion planning problems for their efficiency. However, current applications have implemented them largely for solving 2D (planar) versions of these problems, leaving the general 3D (spatial) case unexplored. In this abstract, we present 3D problem scenarios for an *Autonomous Ground Vehicle (AV)*, that navigates on a 3D *road network.* The novelty of this paper arises chiefly from attempting to address the challenges inherent in this generalization.

Voronoi Diagrams*:*

Given: (1) a metric space $\mathbb{S}$ equipped with metric $\boldsymbol{l}$**,** and (2) a finite set of point *sites* $P \coloneqq \{p_1, p_2, \dots, p_n\}$ within $\mathbb{S}$. We define the *Voronoi cell* R_i as the set of points closer to p_i than to any other site, in the metric $\boldsymbol{l}$ (Devadoss et al. 2011), and the *Voronoi diagram* of the sites as the collection $\{(p_i, R_i): 1 \leq i \leq n\}$. For us, the space $\mathbb{S}$ is the Euclidean plane or space i.e., $\mathbb{R}^2$ or $\mathbb{R}^3$ equipped with the Euclidean metric L^2, for which each Voronoi cell R_i becomes a convex polyhedron. In autonomous robotics, Voronoi diagrams are used in *robot mapping, localization, and motion planning.*

Autonomous Driving Problem Scenarios*:*

For this paper, the robot of interest is an *Autonomous Vehicle (AV)*, such as a car, that is required to navigate on a *road network.* A road network is a graph consisting of multiple nodes - each denoting a road junction or dead-end - interconnected by edges formed by the roads themselves. Each road (edge) comprises one or more *lanes.* Geometrically, each node is a point defined using its northing and easting in the UTM reference frame; each lane is specified by its left and right boundaries, with adjacent lanes sharing a boundary. Each boundary in turn is specified by a polyline (sequence of line segments) in UTM. The AV's position in the UTM frame is input to it periodically through its GNSS (i.e., GPS). The road network has *static obstacles* of known polyhedral shape, placed on it at known positions. The road network itself is acquired from sensor data (camera, lidar, or imaging radar) captured by a fleet of manually operated vehicles (ground vehicles / drones) navigating an environment.

We now pose the problem of investigating Spatial (3D) Voronoi diagrams, and evaluating their performance against alternatives, for the following scenarios: (1) *Mapping:* Generate a *composite map* from two distinct voxel (i.e., 3D) point clouds (taken, e.g., by imaging radar/ lidar) from a data collection vehicle placed in two different (yet unknown) poses in an environment. Here, a composite map contains data from both point clouds stitched together cohesively, accounting for the difference in the vehicle poses used to generate them. (2) *Localization:* Given a 3D road network consisting of nodes and lane boundaries specified as UTM points augmented with an altitude; representing, e.g., mountain roads or a set of intersecting under- and over-passes. Given the AV GNSS position, *localize* the AV by finding the closest line segment to it on its left and right lane boundary polylines. Assume no knowledge of its prior positions. (3) *Motion Planning:* Several polyhedral obstacles (with prescribed facets), together with start and goal AV positions, are prescribed on the 3D road network above. Determine an obstacle-free path of the AV from the start to the goal (if one exists).

Challenges of Spatial Voronoi Diagrams:

The above problem scenarios are well studied in theory and practice for the planar case, using 2D Voronoi diagrams - e.g. Voronoi-based mosaicing in planar mapping (Laraqui et al. 2017), localization (Suri 2002, Cheng et al. 2017), and Voronoi-generated *visibility graph* in motion planning (LaValle 2006, Binder 2018).

Abstracts of the International Cartographic Association, 5, 2022.
European Cartographic Conference – EuroCarto 2022, 19–21 September 2022, TU Wien, Vienna, Austria.
https://doi.org/10.5194/ica-abs-5-90-2022

In contrast, Spatial Voronoi diagrams are theoretically and practically challenging. Our localization problem scenario entails the computation of Voronoi diagrams based on the AV's proximity to n spatial line segments, the runtime complexity of whose construction is conjectured to be (the prohibitive) $O(n^3)$ in the worst case (Devadoss et al. 2011). Many spatial Voronoi algorithms fail to handle degenerate inputs (Ledoux 2007). To our knowledge, there exists only one well-known open-source implementation of spatial diagrams (Rycroft 2009), and none for 3D point location and visibility graphs. Finally, we may need to generate a synthetic road data set with requisite degeneracies and edge cases to test an implementation.

Current and Future Work:

Partial results: We generated Voronoi Diagrams with several randomly generated point sets using Voro++ (Rycroft 2009). From Table 1, its time and memory consumption are sufficiently small for us to extend it for our purpose.

Number of points (n)	**Resource Consumption**	
	Latency (microseconds)	**Memory** (KiB)
1000	76	6392
10000	1520	7212
100,000	12601	13292
1,000,000	130,389	86996

Table 1: Resource consumption of Voro++ diagram generation

We are currently in the early stages of designing and implementing the spatial point location and visibility graph algorithms and extending (Laraqui et al. 2017) for 3D mapping, with Voro++.

Copyright Transfer

References

Binder, B., 2018. TUW Voronoi Graph. Available: http://wiki.ros.org/tuw_voronoi_graph

Boots, B., Kokichi, S., Chiu, S.N., and Okabe, A., 2009. Spatial tessellations: concepts and applications of Voronoi diagrams. John Wiley & Sons, 2009.

Cheng, S-W., and Lau, M-K., 2017. Adaptive point location in planar convex subdivisions. *International Journal of Computational Geometry & Applications* 27, no. 01n02 (2017): 3-12.

De Berg, M.T., Van Kreveld, M., Overmars, M., and Schwarzkopf, O., 2008. Computational geometry: algorithms and *applications*. Springer Science & Business Media, 2008.

Devadoss, S. L., and O'Rourke, J., 2011. Discrete and computational geometry. Princeton University Press, 2011.

Fedorenko, R., 2016. ROS Voronoi Planner. Available: http://wiki.ros.org/voronoi_planner.

Laraqui, A., Baataoui, A., Saaidi A., Jarrar, A., Masrar, M., and Satori, K., 2017. "Image mosaicing using voronoi diagram." *Multimedia Tools and Applications* 76, no. 6 (2017): 8803-8829.

LaValle, S. M., 2006. *Planning algorithms*. Cambridge university press, 2006.

Ledoux, H., 2007. "Computing the 3d voronoi diagram robustly: An easy explanation." In *4th International Symposium on Voronoi Diagrams in Science and Engineering (ISVD 2007)*, pp. 117-129. IEEE, 2007.

Lidar News, 2021. Why does Cadillac Super Cruise work? Available: https://blog.lidarnews.com/cadillac-super-cruise-works/

Rycroft, C.H., 2009. Voro++: A three-dimensional Voronoi cell library in C++, *Chaos* 19, 04111, 2009. Available: https://github.com/chr1shr/voro

Schewchuk, J., 2019. Nearest Neighbor Algorithms: Voronoi diagrams and k-d Trees. Available: https://people.eecs.berkeley.edu/~jrs/189s19/lec/25.pdf

Suri, S., 2002. Point Location. Available: https://sites.cs.ucsb.edu/~suri/cs235/Location.pdf

https://doi.org/10.5194/ica-abs-5-90-2022