Combined Algorithm for Voronoi Diagram Construction in Application to Dynamic Ride Sharing

A. Butenko, J. M. Gómez
Department of Computing Science, Carl von Ossietzky University of Oldenburg
Oldenburg, Germany
E-mail: anton.butenko@uol.de
E-mail: jorge.marx.gomez@uni-oldenburg.de

Abstract—A standard Voronoi diagram decomposes a plane into cells with a common closest site. This structure is widely used in computational geometry in application to the nearest neighbor problem. Using Euclidean metric is the most straightforward solution, however, in an urban environment, it may lead to insufficient accuracy that is crucial in applications such as dynamic ride sharing. Deviations in determining the nearest meeting point are especially significant under the presence of obstacles: water reservoirs, railway tracks, highways, industrial zones, as well as hilly terrain. Here, we propose a combined approach for a city Voronoi diagram construction in a generalized metric space. A transportation network is modelled as a weighted graph, so that the route consists of a foot-walking part and the shortest path in the graph. The presented algorithm constructs a continuous Voronoi diagram for a plane using the individual Voronoi cells graph as generator objects. Evaluation for the specific city topography shows that the described algorithm provides more accurate results in comparison with the standard Voronoi diagram.

Keywords- Voronoi diagram; dynamic ride sharing.

I. INTRODUCTION

Ride sharing applications are aimed at connecting drivers and passengers in an optimal way. What this optimal way means depends a lot on the specific mobility solution philosophy and its target audience. Nevertheless, most of isthem face such optimization problem as the nearest neighbor search: identifying the point from a set of points which is the closest to a given point according to some measure. The mobility application Instaride [2] developed for the spontaneous shared trips is driven by an instant matching algorithm. It connects drivers and passengers in real time based on the user's mobile device positioning (satellite navigation data, triangulation in mobile network) [1]. In order to minimize the driver's efforts and his route detour, the finite set of preselected fixed points is used for passengers' pick-up and drop-off (named meeting points, in general). Preselection of the meeting points is determined by the environmental conditions and is based on criteria such as parking opportunity, presence of pedestrian zones and easily recognizable landmarks. Such an approach leads to the problem of finding the nearest meeting point for users based on their real-time positions.

The paper structure is the following. The Introduction explains the problem's origin. In Section 2, we describe the concept of the presented approach and introduce the terms and notation. Sections 3 and 4 describe two parts of the algorithm: discrete and continuous. In Section 5, the algorithm steps are given in detail. Section 6 presents the algorithm efficiency evaluation for the specific city topography. Section 7 concludes our work.

II. VORONOI DIAGRAM IN A GENERALIZED METRIC SPACE

One of the most effective ways to solve problems related to the nearest neighbor search is to use the Voronoi diagram. We introduce the following notation here: L_p is a metric space with the corresponding function $\rho : L \times L \to \mathbb{R}^+ \cup \{0\}$ that satisfies metric axioms. Then, $S_r(x) = \{z : \rho(x,z) = r\}$ is the metric sphere with radius $r \in \mathbb{R}^+$ and $B(x,y) = \{z : \rho(x,z) = \rho(y,z)\}$ is the bisector of x and y $(x, y, z \in L_\rho)$. It splits L_p into the half-spaces $D(x,y) = \{z : \rho(x,z) < \rho(y,z)\}$ and, lying on the other bisector side, we have $D(y,x) = \{z : \rho(y,z) < \rho(x,z)\}$. For a given finite set of seeds $S = \{s_1, ..., s_k\} \in L_\rho$, the Voronoi cell related to s_i is expressed as

$$VR(s_i, S) = \bigcap_{i \neq j} D(s_i, s_j)$$

and the Voronoi diagram of S:

$$V(S) = \bigcup_{i \neq j} \overline{VR}(s_i, S) \cap \overline{VR}(s_j, S),$$

with the horizontal line denoting closure.

Being the most straightforward solution, a Voronoi diagram based on the Euclidean distance provides tolerable approximation in the urban environment if the points are located quite far apart within the uniform transportation network. In other cases, the results are significantly worse: for short distances, in areas with irregular topography, under the presence of obstacles, or in application to suburbs stretched along the roads forming axon-like structures. The use of other metric functions may improve the accuracy; however, another problem arises: in some cases, the bisector dimension may be more than 1 (this is true even for the Manhattan distance $\rho(x,y) = \sum(|x_1 - x_2| + |y_1 - y_2|)$).

In a number of works, the graph represents the streets network. The discrete network Voronoi diagram is then constructed while the metric used is the link between nodes (e.g. Yomono [5]). However, such models do not allow

 ISBN: 978-1-61208-962-1

shortcuts, which are often used by pedestrians to shorten the paths. Aichholzer et al. [3] consider a plane with Manhattan distance and isothetic transportation network. There are also several works that use the generalized concept of Voronoi region (*needle*) proposed by Bae and Chwa [6].

The approach presented in this work is aimed at being applicable for the non-orthogonal street structure with curvilinear street segments. At the same time, as the ride sharing is spontaneous, we strive to avoid excessive model complexity; only walking to/from meeting points is assumed. The same diagram is used by the driver and the passenger. In addition, being flexible to the possibility of using the available network bandwidth data, the model should also work with the minimum information of this kind. Thus, we believe, the task of developing an optimal method for constructing a Voronoi diagram for a similar class of problems is to find a balance between complexity, accuracy and flexibility in using available data as the latter may vary a lot in different regions.

The main idea of the approach presented below is to construct a discrete Voronoi diagram on a graph and then transform the obtained cells into the seeds or generator objects for the continuous Voronoi diagram on the plane. The latter represents the partition of the plane with a transportation network into proximity regions for the set of the given meeting points.

III. VORONOI DIAGRAM ON THE GRAPH

We consider the area of interest as domain $\Omega \subset \mathbb{R}^2$ containing the city transportation network, providing fixed routes. This network is modelled as a weighted graph $G(V,E)$, where $E=\{e_i\}$ is the set of edges, representing roads and streets. $V=\{v_k\}$ are the graph vertices, corresponding to the intersections and the deadlocks. Non-negative edge weights $w(e_i)$ determine some proximity measure between the vertices connected by the edge e_i. Depending on data availability, it can be, for e.g., edge length or edge travel time. The latter depends on the segment's capacity, inclination, or traffic. Setting $\rho_G(v_i,v_j)$ in an ordinary way as the length of the shortest path between v_i and v_j, one can consider V_{ρ_G} as a metric space. Without additional constraints, it is true for the undirected graph as ρ_G always satisfies the symmetry axiom unlike the directed graph case. Assuming the only way to move (car-driving or foot-walking), we can build a Voronoi diagram graph for $V(G,E)$ with respect to the meeting points $S=\{s_1,...,s_k\}\subset V$ (Figure 1).

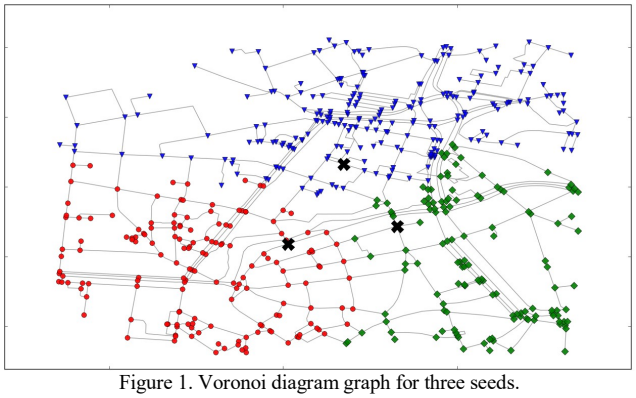

Figure 1. Voronoi diagram graph for three seeds.

The Voronoi diagram brakes up the set of vertices into the direct sum of the Voronoi cells $V=V_1\oplus...\oplus V_k$, where $V_i=VR(s_i,S)$. Let $E_i(s_i)$ be a set of edges connecting vertices within V_i. Then, $E=E_1\oplus...\oplus E_k\oplus C$, where C is the set of «border» edges whose vertices belong to the different cells.

IV. PLANAR VORONOI DIAGRAM

Graph G can be considered not only as a topological structure, but its vertices and edges determine geometrical objects: points and lines within Ω. Hence, each edges subset E_m determines the lines set $E'_m\in\mathbb{R}^2$. Let us consider $E'=\{E'_m\}$, $m=1..k$ as seeds for a planar Voronoi diagram in Ω with the Euclidean distance. Such metric function choice is based on the following assumption. The motion between transportation lines is carried out along a straight line in the direction to the nearest network segment. Practically, the construction of $V(E')$ is based on the search for the metric spheres intersection for each pair (E'_m,E'_n). With the sphere radius variation, the points of these intersections form bisectors $B(E'_m,E'_n)$ and corresponding half-planes $D(E'_m,E'_n)$. Thus, each Voronoi cell $VR(E'_m,E)$ can be computed as an intersection of all half-planes containing E'_m. This process is described in detail in the next section.

V. COMBINED ALGORITHM

The computational algorithm can be performed with the following steps:

1. The city transportation network representation as a graph $G(V,E)$ is obtained from the OpenStreetMap project geodata [6].
2. The set of meeting points $S=\{s_1,...,s_k\}$ is added to the graph vertices V;
3. Construct the Voronoi diagram $V(G)$ graph. If the graph order lets it, brute-force can be used: $\forall v\in V$ find the distance on the graph $\rho_G(v,s_i)\ \forall i$ using the Dijkstra algorithm. If s_j satisfies $\rho_G(v,s_j)=\min_i \rho_G(v,s_i)$ then $v\in VR(s_j)$. Using a more optimal algorithm with the trees can improve the computational performance, if necessary.
4. From the constructed Voronoi cells from the previous step, V_m we get E_m — the subsets of edges that belong to the cell.

 ISBN: 978-1-61208-962-1

5. Each set E_m is transformed into E'_m — the set of the planar lines with their coordinates.
6. Construct the Voronoi diagram graph $V(E')$ using $\{E'_m\}$ as seeds and the Euclidean distance ρ_2 as metric functions:
   6.1. $\forall(E'_m, E'_n)$ in order to find the bisector:
   $$B(E'_m, E'_n) = \bigcup_{r\in(0;\infty)} S_r(E'_m) \cap S_r(E'_m)$$
   interpolate points obtained from this formula for a finite number of radii $r_{k+1} = r_k + \Delta r, \quad r \in [r_{\min}; r_{\max}]$. Here, $r_{\min} = \frac{1}{2}\rho_2(E'_m, E'_n)$ and $r_{\max}$ is the minimum radius that satisfies the condition $S_r(E'_m) \cap S_r(E'_n) \notin \Omega$: the spheres intersect outside of the domain. In the calculations below, a Δr of 10 meters is used;
   6.2. Determine the corresponding half-plane $D(E'_m, E'_n)$ that contains E'_m;
   6.3. Repeat steps 6.1-6.2 for all $m \neq n$. The intersections of the obtained half-planes form the Voronoi cell for the seed E'_m: $VR(E'_m) = \bigcap_{m\neq n} B(E'_m, E'_n) \cap \Omega$;
7. Making a reverse substitution $E'_m \to E_m \to s_m$, gets $VR(s_m, S)$ as the cells of the combined continuous Voronoi diagram based on ρ_2 and ρ_G.

VI. EVALUATION

The accuracy of the two types of Voronoi diagrams was compared for the central part of Oldenburg, Germany. Without claiming to be a full-fledged test, such comparison illustrates the potential prospects of the above-presented approach. For a given area with 1 km radius and 15 meeting points, we construct the standard Voronoi diagram based on the Euclidean Distance (DE) and the combined diagram in a way described above (DC) (see Figure 2).

Comparing the two corresponding types of Voronoi cells C_1 and C_2, their area-weighted average difference can be calculated as:

$$\Delta S = \frac{1}{S(\Omega)} \sum_i \frac{S(C_1^i \triangle C_2^i)}{S(C_1^i \cup C_2^i)} \cdot (S(C_1^i) + S(C_2^i)) \cdot 0.5$$

For the considered example, $\Delta S \approx 0.31$. Also, for 1000 random locations uniformly distributed within the domain, we determine the nearest meeting point in three ways: a) from DE; b) from DC; c) by computing the routes to all the meeting points with the Openrouteservice engine (ORS) [4] and detecting the meeting point corresponding to the minimum route length. The results are the following:

- The nearest meeting points obtained from DE and DC are equal - 792 locations.
- otherwise - 208 locations.

For latter 208 locations, we determine the nearest meeting point with ORS:

- the meeting points obtained from ORS and DC are equal - 165 locations.
- the meeting points obtained from ORS and DE are equal - 41 locations.
- otherwise – 2 locations.

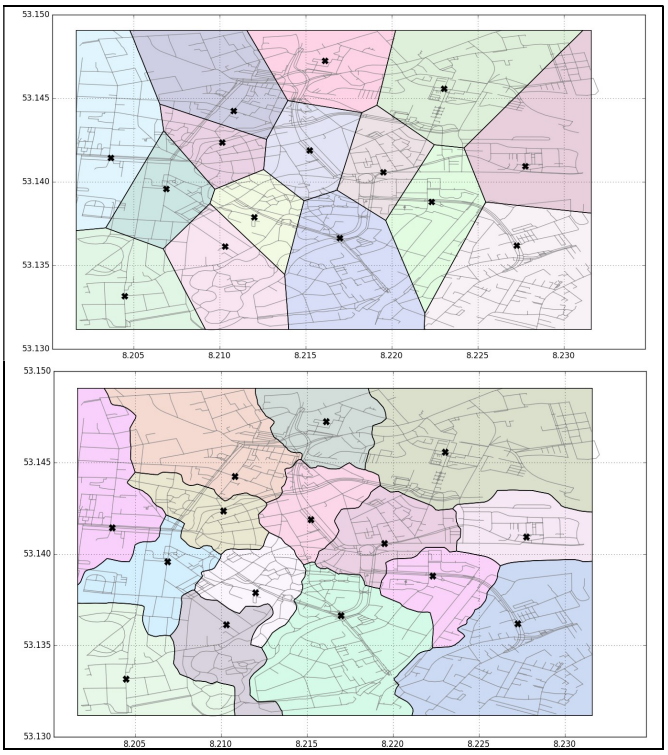

Figure 2. Voronoi diagram for Oldenburg city centre: a) DE; b) DC.

VII. CONCLUSION

There are several steps to be performed next in the context of this work. First, a full algorithm evaluation must be performed in a large area using a number of different criteria. Second, its computational complexity must be evaluated and, probably, reduced. Third, the potential of using additional bandwidth data must be analyzed.

Nevertheless, at this stage, one can conclude that, despite the number of simplifications, the described algorithm provides more accurate results in comparison with a standard Voronoi diagram. At the same time, complex topography features processing requires further study since they are probably the main reason for the remaining imprecision. These include multi-level road crossings, tunnels, elongated geometric objects and natural obstacles.

REFERENCES

[1] M. Eilers et al., “An instant matching algorithm in the context of ride-hailing applications, using isochrones and social scoring”, In:, (Hrsg.), Informatic 2021. Gesellschaft für Informatik, Bonn, pp. 103-114, 2021, doi: 10.18420/informatik2021-007.

[2] Instaride, https://instaride.webflow.io, retrieved: May 2022.

[3] O. Aichholzer, F. Aurenhammer, and B. Palop, “Quickest Paths, Straight Skeletons, and the City Voronoi Diagram”, Discrete and Computational Geometry, 31. pp. 17-35, 2004, doi:10.1007/s00454-003-2947-0.

[4] Openrouteservice, https://www.openstreetmap.org, retrieved: May 2022.

[5] H. Yomono, “The Voronoi diagram on a network”, Technical report, Nippon Systems Co, Tokyo, 1991.

[6] S. W. Bae and K.-Y. Chwa, “Voronoi Diagrams with a Transportation Network on the Euclidean Plane,” Int. J. Comput. Geometry and Appl., vol. 16, pp. 101-112, 2004, doi:10.1007/978-3-540-30551-4_11.

 ISBN: 978-1-61208-962-1