Graph-based Global Path Planning for an Autonomous Electric Scooter using Historical Ride Data*

Vibhu Agrawal, Srijal Shekhar Poojari, and Derek A. Paley

Abstract— **This paper presents a novel strategy for using historical ride data from ride-share electric scooters for path planning of an autonomous electric scooter. Commonly used map datasets do not consider the unique ride characteristics of an electric scooter, and planning over these maps may suggest a route that is undesirable. By weighting our planning based on what routes human riders have taken in the past, we generate a series of waypoints that would allow the scooter to autonomously navigate to a target destination using an efficient and reasonable route. Our results show that this method shows improvement over conventional map-based planners in avoiding suboptimal or undesirable routes.**

I. Introduction

Demand for electric scooters is increasingly popular as our society advances towards more efficient and sustained ways to get around. Cities, residents, and scooter distributors alike often struggle with quickly adjusting to this increased demand, often leading to inequities. This demand also leads to scooters being improperly used, which negatively affects residents and typical city operations. Additionally, it is common to see commuters having trouble finding a sufficiently charged rental scooter nearby when they need it; this process is especially cumbersome for the mobility impaired [1]. Autonomous navigation for electric scooters can solve both of these issues. By utilizing autonomous path planning, electric scooters can navigate to riders when needed and route themselves to park, charge, or be serviced.

Typically, the autonomous path planning process includes a global path, which provides long-term directions toward the goal, and a local path, which considers any structures in the immediate vicinity of the robot. Traditional methods to plan a robot's global path use graph-based search algorithms, such as A^* or Dijkstra's algorithm, over a known map [2, 3]. For outdoor robots, this global map consists of the network of roads that the robot can traverse, such as OpenStreetMap [4] or Google Maps. However, for micromobility systems like the electric scooter, these maps provide insufficient information about terrain, structures, sidewalk, or trails. For example, a scooter rider would not want to ride along a path that could have otherwise been ridden on a bike [5, 6]. Many jurisdictions disallow scooters from using pedestrian sidewalks, although sometimes trails are permitted.

Previous work has utilized similar data from ride-share bicycles, or electric scooters for urban lane planning of bicycle and scooter lanes [7–10]. However, to the best of our knowledge, this is the first work to utilize such data for autonomous path planning of a micromobility transportation modality. The technical approach is to use historical ride data from human-operated scooters to generate a quantitative method for comparing candidate routes. Substantial prior use of a particular route segment by human riders is weighted in the autonomous route planning algorithm, along with heuristic factors such as the route length and tortuosity.

The primary contribution of this paper is the application of optimized routing using historical data to generate route plans for micromobility platforms. A graph-based planning framework generates edge weights based on the frequency of prior usage to select routes that are suitable for a micromobility transportation modality. Additional heuristics include the straight-line distance from the route start to end and bounding the edge weights to avoid tortuous paths. The optimization algorithm is implemented using traditional A^* planning. Simulation results illustrate the framework based on a two-month dataset from a Veo scooter pilot at the University of Maryland College Park in October 2019 and October 2020.

The paper is organized as follows. Section II provides technical background on our methodology. Section III presents the design of the algorithm used to learn the edge weights in the constructed graph. Section IV shows experimental results generated from using this algorithm. Finally, Section V summarizes the results and ongoing and future work.

II. Background

A. Microbility Trip Data

The proposed approach requires a historical ride data set that contains GPS measurements collected along each ride. To illustrate our results, we used an anonymized data set from a campus pilot of electric scooters ridden on and around the University of Maryland College Park campus during October 2019 and October 2020. The data was collected from scooters operated by Veo over 14,468 trips. GPS coordinates of the scooter were logged every 6-8 seconds while each ride was in progress.

B. Path Planning Algorithms

The path planning algorithms discussed in this paper include Dijkstra's algorithm and the A^* algorithm. Dijkstra's algorithm is a greedy search algorithm that produces single-source shortest paths in a graph with nonnegative edge weights. It works by incrementally discovering lower cost paths to a target vertex. A^* is a path planning algorithm

*This work is supported by the Maryland Robotics Center at the University of Maryland.

The authors are affiliated with the Maryland Robotics Center, University of Maryland, College Park, MD 20742, USA. dpaley@umd.edu

Manuscript 2204 submitted to 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Received March 1, 2023.

that is an extension of Dijkstra's algorithm. It incorporates a heuristic estimate in order to reduce the number of explored states by incorporating the cost to get to the goal from a given state. Whereas Dijkstra's algorithm sorts vertices to explore by $C^*(x')$, where C^* is the optimal cost to reach vertex x' by traversing edges on the graph, A^* sorts vertices to explore by $C^*(x')+\hat{G}^*(x')$ where $\hat{G}^*$ is the heuristic cost to reach the goal from vertex x' [11]. A^* is guaranteed to find the optimal route from the start to the goal vertex as long as $\hat{G}^*(x')$ is an underestimate [12]. A commonly used heuristic is the straight-line distance from some vertex to the goal vertex. This heuristic is admissible because this length ignores obstacles that an agent moving along the path will inevitably encounter, causing it to be an underestimate [11].

C. Hexagonal Grid System

To form a spatial histogram of our ride database, we used Uber H3, a Hexagonal Hierarchical Spatial Index [13]. Uber H3 was used in order to cluster scooter GPS coordinates into cells on a map. This step is necessary to get meaningful frequency estimates along historical routes with appropriate resolution. H3 is the index system Uber uses to optimize ride pricing in different localities by analyzing geographic information localized to different hexes. Uber H3 works by dividing the surface of the Earth into hexagons, providing regions to analyze of varying sizes depending on the sizes of the hexagons used [13]. The advantage of using a hexagonal grid system rather than a conventional rectangular grid is that hexagons avoid many of the distortions that come with rectangular grid cells. Due to the curvature of the surface of the Earth, one must account for the slight differences in length between the two horizontal sides of the rectangle, which can make route planning more difficult. Hexagons have the useful property of having a center point that is equidistant to each corner, even when distorted over the surface of a spherical object. Another advantage of using H3 is the hierarchical organization, which allows us to adjust the resolution or size of the grid cells depending on how many GPS points are available within a given locality. If the data is more spread apart, it would necessitate using larger hexagons to localize in order to generate meaningful frequency estimates.

III. Usage-based Edge-Weighting Algorithm

This section describes the edge-weighting algorithm for path planning, starting from the construction of a graph that describes the route network.

A. Graph Construction

The first step is to process and clean the historical ride data, filtering out erroneous data points that do not fit the geographic locality of the rest of the data. This processing involves localizing GPS coordinates within their respective hexes in the H3 system. It is important to choose a resolution that does not separate the coordinates too much in order to attain route frequencies that are useful for the path-planning algorithm. We construct a graph using these hexes, with vertices in the graph representing the hexes and undirected edges representing instances of two hexes appearing consecutively in a given trip. The choice to use undirected edges is to maximize the frequency discount for edges in the graph, to allow the algorithm to better choose frequently traveled edges. This choice was made with the assumption that most routes will encounter the same obstacles when traveling in one direction or the other, and thus the optimal ground truth path would be similar in either direction. By making the graph undirected, the granularity of the generated waypoints is maximized.

B. Edge Weighting

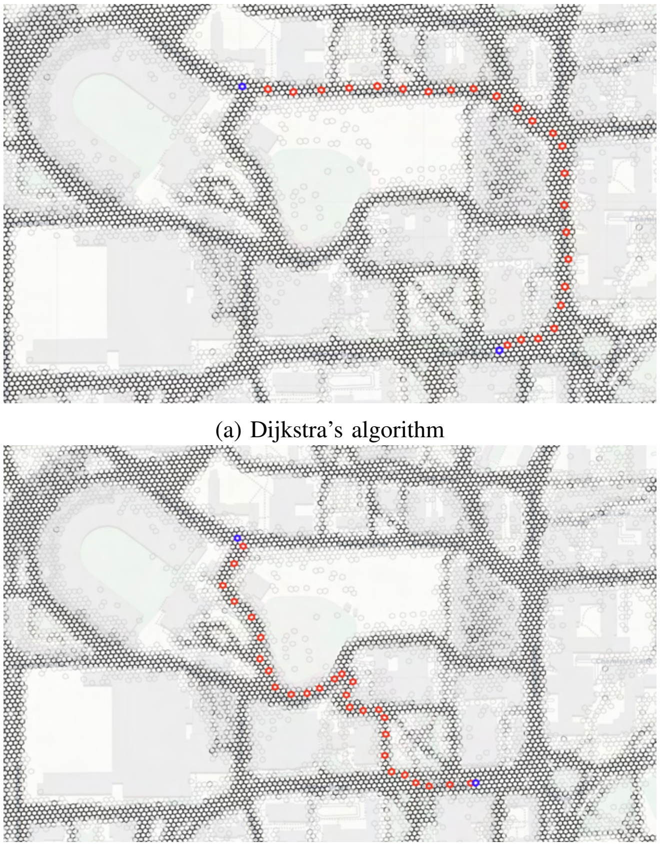
Edge weights are initially calculated as the distance between the center of two consecutive hexes in a trip. A discount factor is incorporated into the edge weighting in order to encode which edges were traveled more frequently in the ride data set. This discount factor is some number less than 1 that is multiplied by the existing edge weight whenever an edge is seen more than once in the dataset, thereby decreasing the weight of that edge. Then, when the path-planning algorithm is run, the algorithm will favor more commonly traveled edges because of lower weights. Generated waypoints fall on routes that human riders have traveled before. Thus, the algorithm not only favors routes that take it from the start vertex to the goal vertex, but also routes that avoid obstacles and other hindrances because it is likely a large number of human riders avoided the same obstacles, and this pattern would appear in the data. The discount factor is chosen after considering the auto-generated routes to determine the quality of paths produced with various discount factors. A lower discount factor allows a path-planning algorithm to more heavily favor frequently traveled routes, but it also causes edge weights to approach zero faster. This means that short, high-frequency edges can dominate other edges that might be more practical for the scooter to reach its destination. This can produce tortuous paths that are suboptimal. A higher discount factor avoids this pitfall but cannot take as much advantage of higher frequency edges. A useful discount factor must strike a balance among these considerations.

It is also necessary to bound the edge weights. Due to the asymptotic nature of an exponential function, the weight of a higher frequency edge will approach zero as it occurs more and more times in the dataset. This can cause the edge to be overly favored in a path-finding algorithm and can result in large bends in the generated paths as sequences of short high-frequency edges are chosen over more direct, lower-frequency edges. Thus weights must be bounded to some number that allows frequent edges to be favored but not so much that they dominate more practical edges in a generated route.

C. Path-planning algorithms

The A^* algorithm is selected here over Dijkstra's algorithm because there is an admissible heuristic, namely the

Manuscript 2204 submitted to 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Received March 1, 2023.



(a) Dijkstra's algorithm

(b) A* algorithm

Fig. 1: Comparison of usage-weighted paths produced by Dijkstra's algorithm and A^* on the Uber H3 grid. In this example, the Dijkstra's path (724 m) is longer than the A* path (632 m). The hex gray-scale intensity depicts the frequency of prior usage by human riders.

straight-line distance from a given vertex to the goal vertex. This algorithm generates waypoints and routes that are closer to an optimal straight line to the goal, as shown in Fig. 1. An additional benefit of using A^* is this choice allows the path-planning algorithm to avoid short high-frequency edges that do not contribute as much towards reaching the final destination. These vertices, although they might have had a lower edge weight, have a higher heuristic cost because they cause movement away from the goal vertex. In addition, it is necessary to incorporate a discount factor into this heuristic to ensure the algorithm does not overly prefer long edges that join some vertex and another vertex much closer to the goal vertex. We tuned the algorithm to introduce more granularity into the generated paths and prefer shorter edges that generate more waypoints suitable for autonomous navigation.

To define the edge weight, let $h(u, g)$ be the straight-line distance from vertex u to an ultimate goal vertex g, d the distance from u to v, w_e the discount factor on the edge weights used to indicate the frequency with which each edge was traveled, f the number of times that edge is seen in the dataset, w_h the weight for the heuristic, and l the lower bound for the edge weights. The cost $C(u, v)$ to move from vertex u to vertex v is

$$C(u, v) = \max((w_e)^f \cdot d(u, v) + w_h \cdot h(u, g), l).$$

IV. RESULTS

Fig. 2 illustrates the frequency of prior usage by human riders of electric scooters on the University of Maryland campus. This is useful in depicting how human riders travel around the campus and where, consequently, an autonomous scooter might travel as well. This graph was constructed using a resolution of 13 for the hexes, which gives each cell an edge length of 3.55 m. Through experimentation, this resolution was found to be optimal due to the amount of data present in each hex at that resolution, as well as the ability of the scooter to maneuver around obstacles within such distances.

Fig. 2: The density of micromobility travel around the University of Maryland campus. Some of the most frequently traveled edges are shown, with brighter edges representing more frequently travelled paths and the black hexes representing all recorded GPS locations localized to hexes.

The proposed edge-weighting approach often finds routes that are more practical than those offered by existing methods. As an example, Fig. 3 shows two selected points on campus that are on several frequently traveled routes. The best option available to emulate scooter travel on a platform like Google Maps is through the bicycle modality. However, Google Maps fails to avoid obstacles that might hinder a human scooter rider or autonomous scooter, such as staircases. Note that in the following figures, blue hexes indicate the start and end of each route while red hexes represent waypoints generated along that route. Hexes that are colored differently from the aforementioned colors are not part of the route but indicate the degree of the vertex at that hex, with darker hexes representing higher degree vertices and vice versa.

Manuscript 2204 submitted to 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Received March 1, 2023.

(a) Proposed route

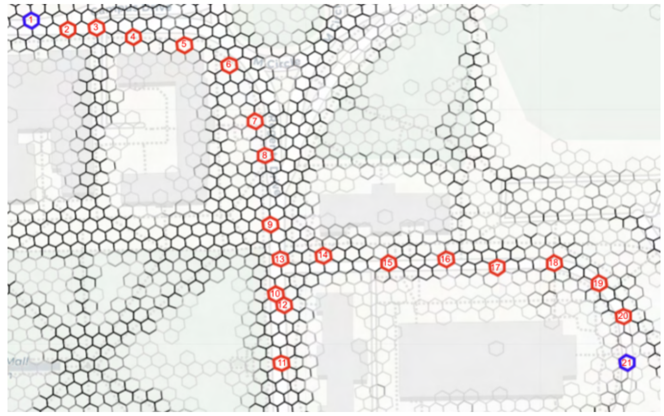
(b) Google Maps route

(c) Stairs on Google Maps route

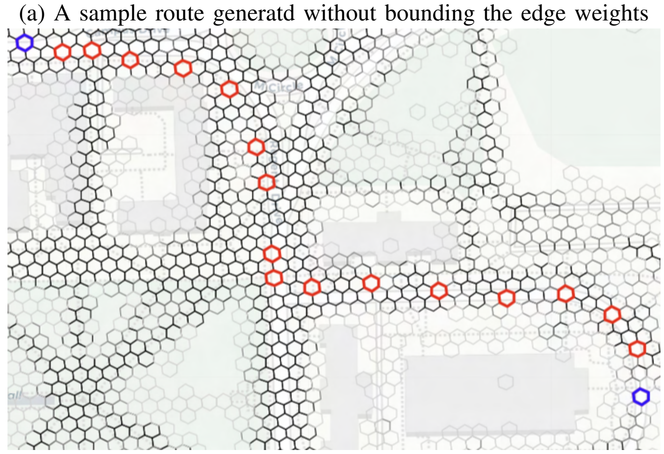
Fig. 3: This figure shows the Google Maps route [14] generated between two points, illustrating the obstacles one could run into using current available algorithms

Figs. 4 and 5 illustrate the effect of the various parameters used by the algorithm. Through experimentation, it was found that a discount factor of 0.9 was useful in generating practical paths for the autonomous scooter. This was the lowest weight at which observed results were not overly convoluted in that they did not contain the types of hairpin u-turns demonstrated in Fig. 4(a).

Fig. 4 shows the usefulness of using a lower bound for the edge weights. Without the bound, the algorithm generates a path with a large diversion in the middle to target several high frequency edges rather than a more direct edge from vertex 9 to vertex 13 in Fig. 4(a). Through experimentation,



(a) A sample route generatd without bounding the edge weights



(b) The same route with bounded edge weights

Fig. 4: Comparison of route plans with and without bounded edge weights. Bounding the edge weights avoid tortuous paths

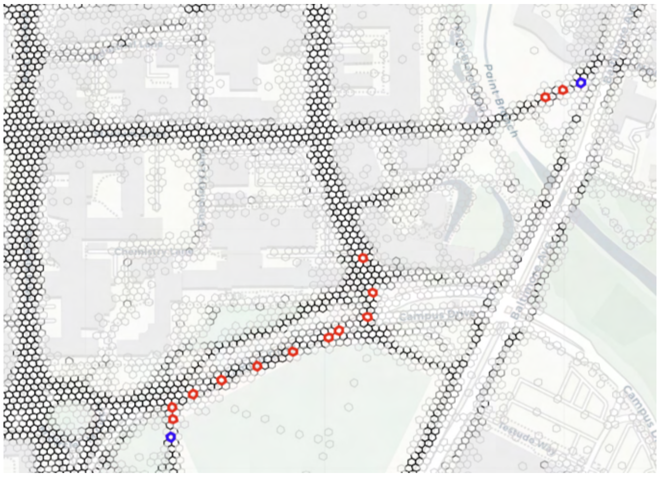
a useful lower bound was $\sqrt{3}a - \epsilon$ where a represents the length of one side of a hexagon, meaning $\sqrt{3}a$ is the distance from the center of one hexagon to the center of a hexagon bordering it. This is the closest two generated waypoints can be, and some small ϵ was chosen in order to allow edges between two adjacent hexagons that have been seen more than once in the dataset to decrease slightly. So the edge that has been seen more than once is then favored over another less traversed edge.

Fig. 5 demonstrates the need for weighting the heuristic. Through experimental results, a useful weight was 0.75. A lower weight produces paths more similar to Dijkstra's algorithm, whereas higher weights produce more dispersed waypoints that would not be practical for an autonomous scooter to use to navigate.

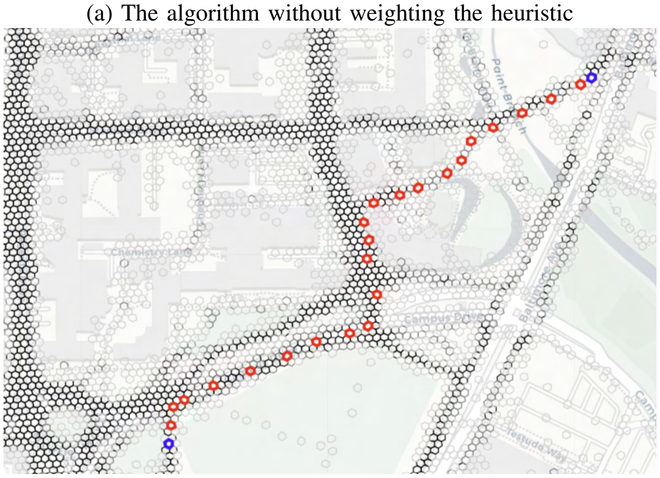
V. Conclusion

This paper introduces an algorithm to learn edge weights in a graph constructed from historical data of micromobility trips. GPS coordinates from historical data are localized using a hexagonal grid system to estimate the frequencey of usage of each route segment. A parameterized approach to edge-weighted planning is added to conventional planning

Manuscript 2204 submitted to 2023 IEEE/RSJ International Conference
on Intelligent Robots and Systems (IROS). Received March 1, 2023.



(a) The algorithm without weighting the heuristic



(b) The algorithm with a weighted heuristic

Fig. 5: Comparison of route plans with and without weighting the heuristic. Weighting the heuristic appropriately generates waypoints more suitable for route following.

algorithms to generate routes that are practically useful for an autonomous scooter to travel. A discount factor is necessary to encode information about the frequency with which an edge was traveled, and a weighted heuristic produces routes that are straighter while still avoiding obstacles. Including a lower bound on the edge weight avoids overly winding paths. This approach, in conjunction with other onboard systems, would help a scooter navigate autonomously in environments where there is access to large quantities of historical data. This approach is useful for global path planning for autonomous scooters, as well as for manually operated micromobility systems.

References

[1] R. L. Sanders, M. Branion-Calles, and T. A. Nelson, "To scoot or not to scoot: Findings from a recent survey about the benefits and barriers of using e-scooters for riders and non-riders," *Transportation Research Part A: Policy and Practice*, vol. 139, pp. 217–227, 2020, ISSN: 0965-8564. DOI: https://doi.org/10.1016/j.tra.2020.07.009.

[2] J. Giesbrecht, "Global path planning for unmanned ground vehicles," Defence Research and Development Suffield (ALBERTA), Tech. Rep., 2004.

[3] P. Marin-Plaza, A. Hussein, D. Martin, and A. d. l. Escalera, "Global and local path planning study in a ros-based research platform for autonomous vehicles," *Journal of Advanced Transportation*, vol. 2018, pp. 1–10, 2018.

[4] OpenStreetMap contributors, *Planet dump retrieved from https://planet.osm.org*, https://www.openstreetmap.org, 2017.

[5] R. Zhu, X. Zhang, D. Kondor, P. Santi, and C. Ratti, "Understanding spatio-temporal heterogeneity of bike-sharing and scooter-sharing mobility," *Computers, Environment and Urban Systems*, vol. 81, p. 101 483, 2020, ISSN: 0198-9715. DOI: https://doi.org/10.1016/j.compenvurbsys.2020.101483.

[6] B. Laa and U. Leth, "Survey of e-scooter users in vienna: Who they are and how they ride," *Journal of Transport Geography*, vol. 89, p. 102 874, 2020, ISSN: 0966-6923. DOI: https://doi.org/10.1016/j.jtrangeo.2020.102874.

[7] P. Folco, L. Gauvin, M. Tizzoni, and M. Szell, "Data-driven micromobility network planning for demand and safety," *Environment and Planning B: Urban Analytics and City Science*, p. 23 998 083 221 135 611, 2022.

[8] F. Chiariotti, C. Pielli, A. Zanella, and M. Zorzi, "A dynamic approach to rebalancing bike-sharing systems," *Sensors*, vol. 18, no. 2, p. 512, 2018.

[9] S. Liu, A. Siddiq, and J. Zhang, "Planning bike lanes with data: Ridership, congestion, and path selection," *Congestion, and Path Selection (March 12, 2022)*, 2022.

[10] J. L. Meyer *et al.*, "Bike big data: How gps route data collected from smartphones can benefit bicycle planning," 2013.

[11] S. M. LaValle, *Planning Algorithms*. Cambridge University Press, 2006.

[12] P. E. Hart, N. J. Nilsson, and B. Raphael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," *IEEE Transactions of Systems Science and Cybernetics*, vol. SCC-4, no. 2, pp. 100–107, 1968.

[13] U. Inc., *H3: Uber's Hexagonal Hierarchical Spatial Index*, Accessed 28 February 2023, 2018. [Online]. Available: https://www.uber.com/blog/h3/.

[14] G. Inc., *Google Maps*, Accessed 28 February 2023, 2023. [Online]. Available: https://www.google.com/maps/dir/38.987568,-76.941569/38.990620,+-76.945453/@38.9890857,-76.9457193,692m/data=!3m2!1e3!4b1!4m7!4m6!1m0!1m3!2m2!1d-76.945453!2d38.99062!3e1.

Manuscript 2204 submitted to 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Received March 1, 2023.