

**2022 NDIA MICHIGAN CHAPTER
GROUND VEHICLE SYSTEMS ENGINEERING
AND TECHNOLOGY SYMPOSIUM
AUTONOMY, ARTIFICIAL INTELLIGENCE & ROBOTICS TECHNICAL SESSION
AUGUST 16-18, 2022 - NOVI, MICHIGAN**

**PATH-FREE ESTIMATION OF NAVIGATION DISTANCE USING
OBSTACLE SHAPE STATISTICS AND DENSITY**

**Stephen J. Harnett, PhD Student^{1,2}, Sean Brennan, PhD¹, Karl Reichard, PhD^{1,2},
Jesse Pentzer, PhD², Seth Tau, PhD^{1,3}, David Gorsich, PhD³**

¹The Pennsylvania State University, University Park, PA

²The Applied Research Laboratory, The Pennsylvania State University, University Park,
PA

³The U.S. Army CCDC Ground Vehicle Systems Center, Warren, MI

ABSTRACT

In the field of ground robotics, the problems of global path planning and local obstacle avoidance are often treated separately but both are assessed in terms of a cost related to navigating through a given environment. Traversal cost is typically defined in terms of the required fuel [1], required travel time [2], and imparted mechanical wear [3] to guide route selection. Prior work [4] has shown that obstacle field complexity and navigation cost can be abstracted into quantitative dimensionless parameters. But determining the cost parameters and their relationship to field complexity requires running repeated path planning simulations [4]. This work presents a method for estimating navigation cost solely from geometric obstacle field complexity measures, namely the statistical properties of an obstacle's shape and the density of obstacles within an environment, eliminating the requirement to run a path planner in a simulation environment.

Citation: S. J. Harnett, S. Brennan, K. Reichard, J. Pentzer, S. A. Tau, D. Gorsich, "Path-free Estimation of Navigation Distance Using Obstacle Shape Statistics and Density," In *Proceedings of the Ground Vehicle Systems Engineering and Technology Symposium (GVSETS)*, NDIA, Novi, MI, Aug. 16-18, 2022.

1. INTRODUCTION

In the field of ground robotics, it is advantageous to know the cost of navigating through a given environment prior to committing to a mission. Researchers have long been interested in the

relationship between obstacle field terrain and navigation performance, leading to the development of models such as the NATO Reference Mobility Model (NRMM) for estimating mobility over known terrain [5]. Previously, determining this cost and

its relationship to obstacle field complexity required running numerous path planning simulations [4] or having experimental data [6]. This work presents a method to predict the general relationship between navigation cost and observable obstacle field parameters so that cost can be estimated for a given obstacle field without planning the path.

The key contributions of this paper are that one can estimate the traversal cost for a specific obstacle field and mission, without requiring the computational time and resources necessary to run path planning simulations. Additionally, because this high-level estimate is solely based on obstacle geometry, it can be performed prior to a path planning algorithm being designed and without knowing specific vehicle dynamics parameters.

The remainder of this paper is organized as follows: Section 2 characterizes the geometric parameters of obstacle shape and size. Section 3 determines the deflection required to navigate around an average sized obstacle derived from geometric parameters of obstacle shape and size. It also determines an estimate of the number of times the path has to deflect around an obstacle during a given mission, based on occupancy and density parameters of the obstacle field. In Section 4, this cost estimate was then evaluated on numerous simulated obstacle fields to generate data of estimated cost versus obstacle field complexity. The results and conclusions show the agreement between the estimated cost versus measured cost from path planning.

2. MAP AND PATH CHARACTERIZATION

In the remainder of this work, the term map refers to binary maps detailing only obstacles and free space. Obstacles in the field can be represented by impassable convex polytopes. These polytopes may be nearly fully tiled, forcing the planning algorithm to route around the perimeters of the obstacles. Or, the obstacles can be spaced apart by larger gaps, giving the planning algorithm more freedom to operate within as shown in figure 1.

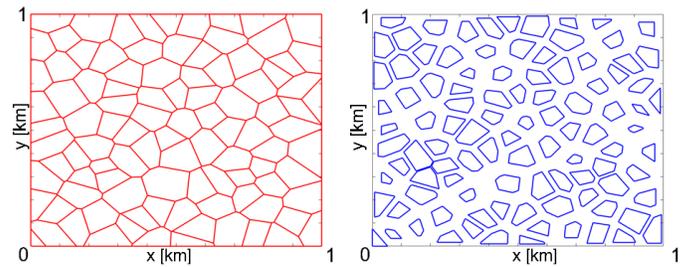


Figure 1: An example of an obstacle field that is fully tiled compared to a field generated from the same obstacles, shrunk down to produce a non-zero gap size.

Polytopes are chosen as an obstacle representation because they are a grid-free, memory-efficient representation, requiring that only their vertices are stored. Additionally, when the polytopes are convex, path planners cannot get “stuck” by routing into obstacle local minima while trying to minimize distance to the goal, as shown in figure 2 [4]. Concave obstacles can be included in this mapping convention by using a convex polytope to enclose both the concave obstacle and the free space forming the local minimum.

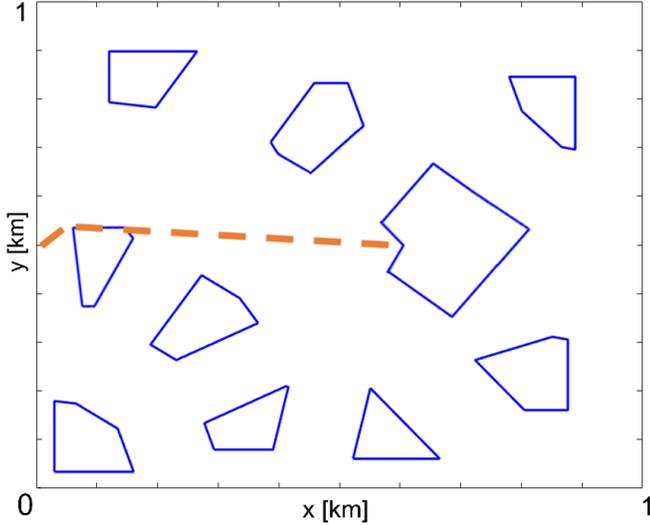


Figure 2: An example of a path, shown in orange, that could be made by a planner routing from $(0, 0.5)$ to $(1, 0.5)$ while trying to minimize distance to the goal. Note that the path routes into the local minimum of a concave obstacle.

Individual obstacles can be described by their maximum radius, R , where radii are defined as the distance from the centroid to the vertices, as shown in figure 3 [4].

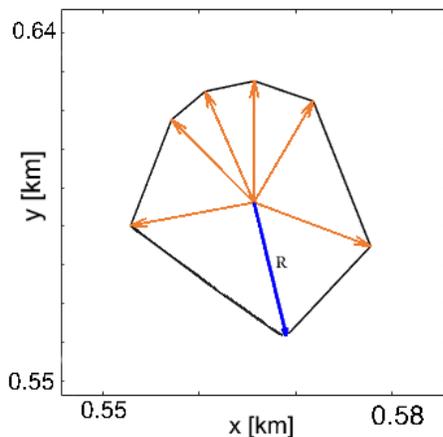


Figure 3: A polytope showing the various radii as orange vectors, with the maximum radius highlighted in blue.

The standard deviation of obstacle radius, σ_R , the obstacle density, ρ , and the occupancy ratio, $r_{A,occ}$, can be used to characterize the obstacle field holistically. Note that ρ is the number of obstacles per unit area, independent of the size of these obstacles, while the occupancy ratio is the ratio of occupied area to unoccupied area, independent of the number of obstacles comprising the occupied area.

To describe the complexity of an obstacle field, this work uses departure ratio, r_D , which is the product of average maximum radius of all polytopes in the field and the square root of area obstacle density as shown in equation (1) [4].

$$r_D = \bar{R}\sqrt{\rho} \quad (1)$$

An example of how departure ratio scales with other obstacle field parameters is shown in figure 4; this figure shows how two fields with 20 obstacles per unit area each can have very different departure ratios based on obstacle size. However, the figure also shows that a field with over double the number of obstacles can have a similar departure ratio, if it also has a smaller average obstacle size.

To describe the cost of a path, we can use the ratio of the total path length required to navigate around obstacles, L_P , to the Euclidean distance from the start to the goal, L_E , defined as the length cost ratio, r_{LC} in equation (2) [4].

$$r_{LC} = \frac{L_P}{L_E} \quad (2)$$

Previously an empirical relationship between length cost ratio and departure ratio was discovered [4]. This was obtained by running path planning simulations for each data point over obstacle field maps and measuring the length cost of the navigated paths. This relationship is shown in figure 5. This work aims to discover a relationship between obstacle field parameters and length cost without requiring repeated runs of path planning algorithms in simulation.

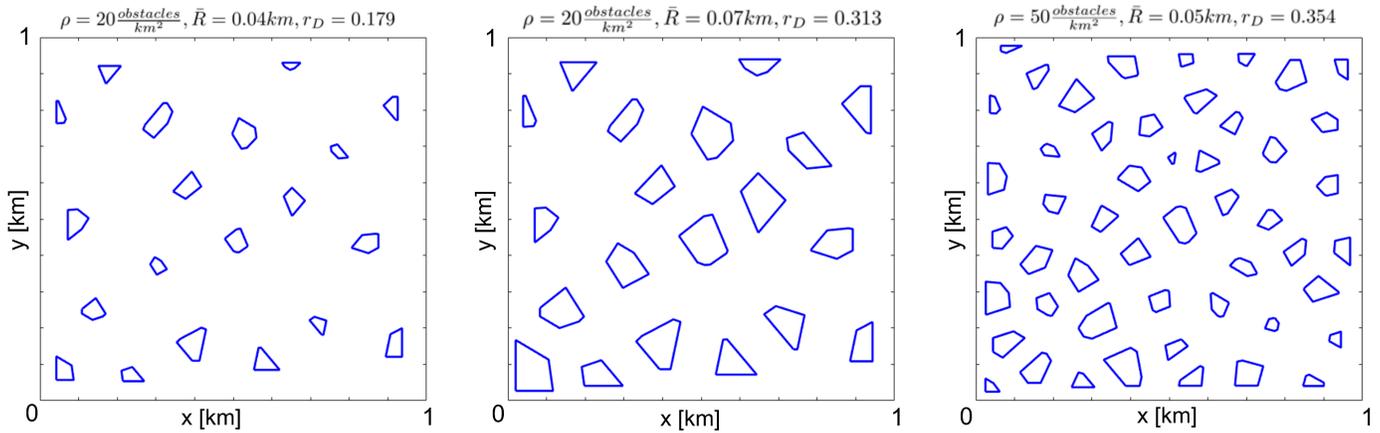


Figure 4: An example of departure ratio as it relates to the geometric parameters of obstacle density and average maximum radius. The field on the left has the same area obstacle density as the field in the center but because its average maximum radius is approximately half of that of the center field, the departure ratio is also approximately halved. Contrast this with the field on the right, which has a similar departure ratio to the field in the center, despite having over double the area obstacle density.

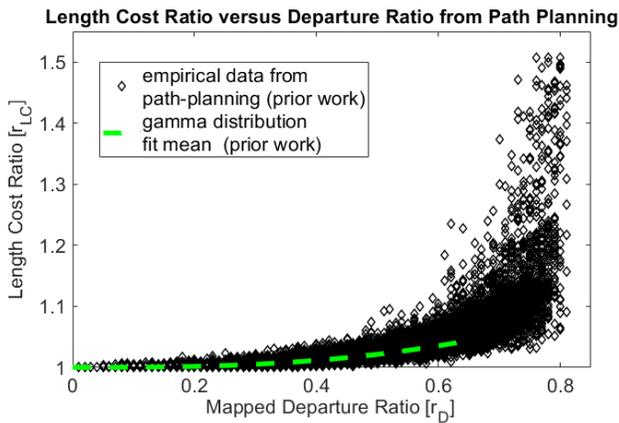


Figure 5: The previously discovered empirical relationship between obstacle field complexity and navigation length cost. A gamma distribution fit mean is shown with shape, k , and scale, θ , as functions of mapped departure ratio: $k = 0.4124r_D + 41.91r_D^2$ and $\theta = 0.0048r_D - 0.0016r_D^2$. Details of the curve fit process are beyond the scope of this paper but can be found in a prior publication [4].

Note that the mean is not shown above a mapped departure ratio of 0.65 because the range in length cost ratio increases ρ dramatically as obstacle fields become harder to navigate at higher departure ratios [4].

3. COST ESTIMATION ALGORITHM

This paper analyzes the entire navigation task by decomposing it into the discrete sub-tasks of routing around each blocking obstacle, as shown in figure 6. Therefore, this series of encounters that would occur within a path plan can be estimated as repeated encounters with a single average-sized obstacle. Thus, estimating the length cost consists of two separate estimation methods: one to estimate the divergence necessary to route around the average-sized obstacle, and one to estimate the obstacle encounter rate in a path of a given length.

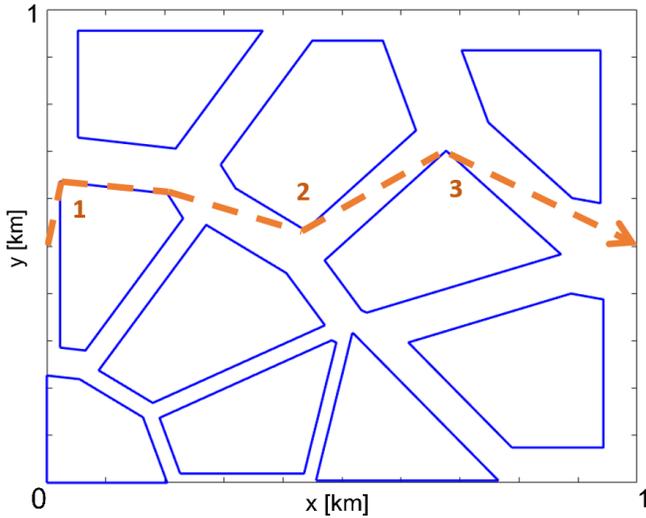


Figure 6: A path, shown in orange, from (0, 0.5) to (1, 0.5) encountering three obstacles that force a divergence from a straight line.

3.1. Average Obstacle Geometry Estimate

The first step is to determine the average polytope geometry for the desired path direction. Each vertex facing away from the travel direction on a potentially blocking obstacle can be treated as a position where the planner may have to make a choice of whether to deflect left or right around the obstacle. To determine the relevant vertices, any vertex whose interior vertex normal does not point in the direction of travel is discarded, as shown in figure 7.

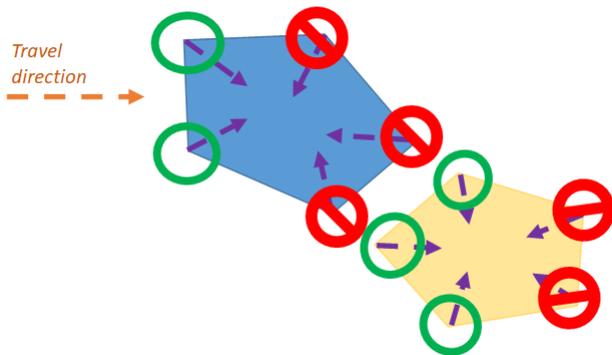


Figure 7: An illustration of which vertices are discarded based on travel direction. While it may seem counter-intuitive to discard the lower-most

vertex on the blue polytope, as it appears to obstruct travel in the indicated direction, recall from Section 2 that the obstacle map starts as a fully-tiled polytope field so decision points at vertices on the “back” of the blue polytope are analyzed when considering decision points on the “front” of the adjacent, yellow polytope.

Travel direction is accounted for rather than considering all vertices in the obstacle field as potential decision points because an obstacle field could conceivably have different average geometries for different travel directions. A physical example to illustrate this would be a desert of barchan sand dunes, which form a sawtooth wave pattern based on wind direction, resulting in a field that has very steep (and therefore potentially impassable) obstacles in one travel direction, but shallower obstacles (that could be considered non-blocking depending on vehicle capability) in the opposite direction.

Additionally, vertices whose vertex normal is not within half the vertex size, θ_{vertex} , of the travel direction are also discarded, as this implies that, while the vertex is facing away, the polytope is not obstructing the path, as shown in figure 8.

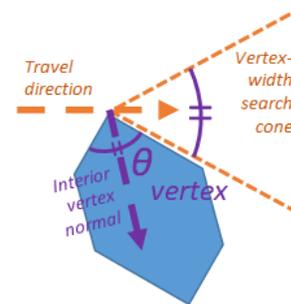


Figure 8: An illustration of a vertex that opens away from the travel direction yet would not block the travel direction and therefore is not a decision point for the path planner.

Once the relevant decision points are known for each possible encounter with a polytope obstructing the path, the navigation task can be approximated

as the choice between the larger divergence angle, θ_{big} , and the smaller divergence angle, θ_{small} , both of which would divert the path from the straight line Euclidean distance from start to goal, as shown in figure 9. Choosing to route around the smaller deflection angle can sometimes result in a longer path than choosing the larger deflection angle if S_{small} is sufficiently large and S_{big} is sufficiently small, in which case this assumption breaks down. This limitation is discussed further in Section 4.

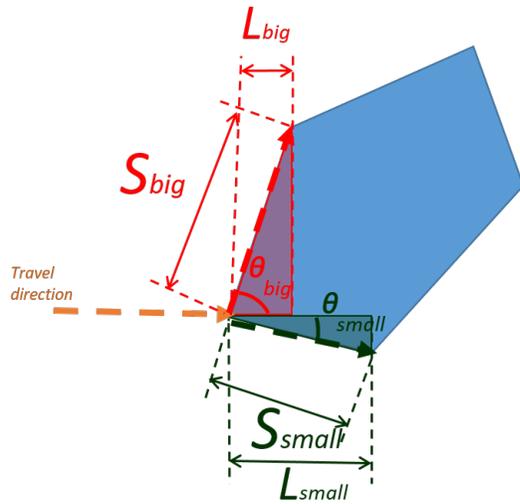


Figure 9: A schematic of a potentially encountered polytope showing the larger and smaller divergence angles, θ_{big} and θ_{small} , respectively, the associated side lengths, S_{big} and S_{small} , and the distances that either choice would progress the vehicle towards the goal, L_{big} and L_{small} .

If path planning is modeled as an assumption that the vehicle always chooses to route around the obstacle on the side with the smaller divergence angle, the vehicle has to travel a distance of the side length associated with the smaller divergence, S_{small} , to traverse towards the goal by a distance of the cosine of the divergence angle scaled by the associated side length. Thus knowing the divergence angles and their associated side lengths gives an approximate length cost ratio for this individual obstacle encounter, $r_{LC,i}$, as shown in equations (3-5).

$$\frac{L_P}{L_E} = \frac{S_{small}}{L_{small}} \quad (3)$$

$$L_{small} = \cos(\theta_{small}) \cdot S_{small} \quad (4)$$

$$\Rightarrow \frac{1}{\cos(\theta_{small})} = r_{LC,i} \quad (5)$$

For each relevant vertex, the angle from the interior vertex normal to the travel direction, $\theta_{VN,TD}$, can be used with the interior vertex angle, θ_{vertex} to calculate the chosen, smaller divergence angle and unchosen, larger divergence angle, θ_{big} and θ_{small} per equation (6). These quantities are visually described in figure 10.

$$\theta_{big}, \theta_{small} = \frac{\theta_{vertex}}{2} \pm \theta_{VN,TD} \quad (6)$$

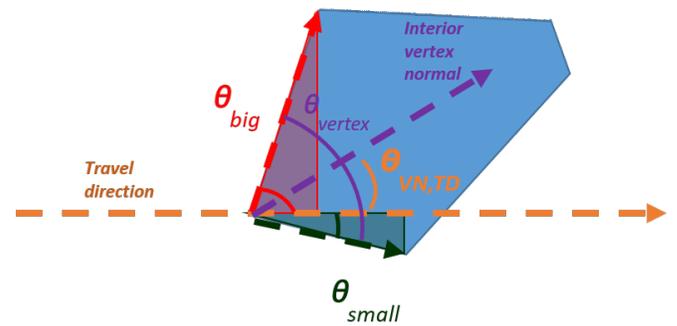


Figure 10: The angle from the interior vertex normal to the travel direction, $\theta_{VN,TD}$, which can be used with the interior vertex angle, θ_{vertex} to calculate the big and small divergence angle choices, θ_{big} and θ_{small} per equation (6)

Analyzing each polytope in this way yields a set of the chosen divergence angle and associated side length for every relevant decision that a planner could have to make in this field, for this travel direction. A histogram of the set of small divergence angles and large divergence angles is shown in figure 11. From this, the deflection around a statistically average obstacle is known, considering the length cost of routing around an individual obstacle as shown in equations (3-5).

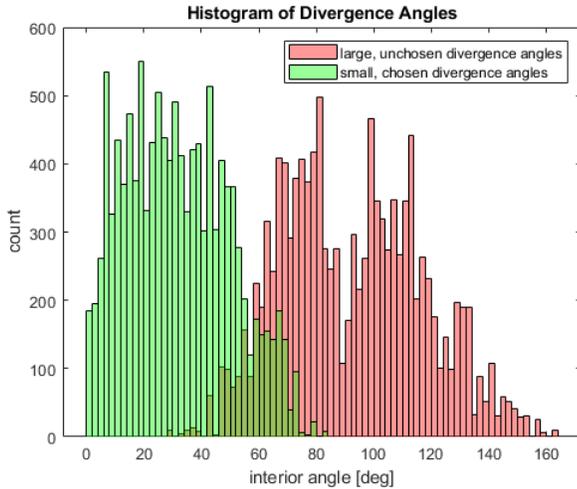


Figure 11: Histogram of the sets of the larger and smaller divergence angles.

3.2. Encountered Obstacle Count Estimate

Now that the average obstacle geometry is known, the next step is to determine the obstacle encounter rate, N_{int} . As mentioned before when defining departure ratio, the obstacle encounter rate depends on both obstacle density, ρ , and obstacle size, R , because the obstacle encounter rate increases with both increasing obstacle size and increasing obstacle density. This work describes three approaches for estimating the obstacle encounter rate. The simplest approach is to use the square root of area obstacle density, ρ , as an approximation of linear obstacle density, λ , over the course of a path of length L , but this only works well for uniform, nearly-fully tiled obstacle fields. As obstacles in the field shrink, despite the area obstacle density not changing, a straight line is less likely to encounter obstacles because the occupied space has decreased, as illustrated by figure 12.

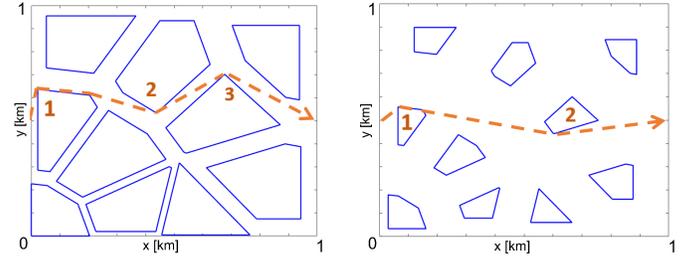


Figure 12: Two obstacle fields with the same linear obstacle density, $\sqrt{10}$, have different obstacle encounter rates depending on obstacle size.

This leads us to a method that can account for obstacle size, so an approximation can be derived in terms of linear unoccupancy ratio, $r_{L,unocc}$, which is defined in terms of unoccupied area, A_{unocc} , and total area, A_{tot} , in equation (7), distance from start to goal, L , and average gap size \bar{G} , which is described in figure 13. This estimate is derived in equations (8-9) [4].

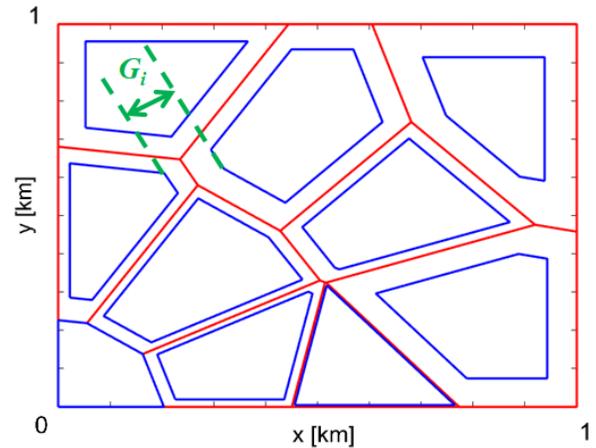


Figure 13: An obstacle field showing the gap size of an individual gap, G_i , labeled. The obstacles as they would appear if fully tiled (i.e. when gap size is zero) appear in red.

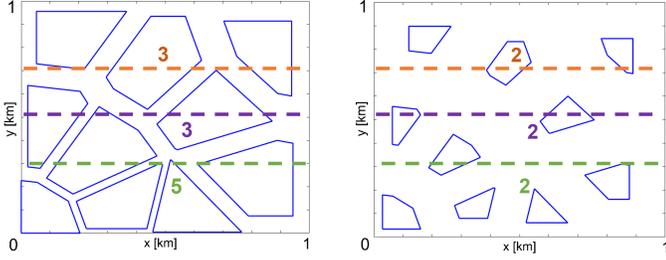


Figure 14: Two obstacle fields with the same obstacle density but different obstacle radii result in different encountered obstacle rates, as measured by ray casting, showing that ray casting does account for obstacle size. The field on the left has an average rate of $3.67 \frac{\text{obstacles}}{\text{km}}$ over the three rays shown, while the figure on the right has an average rate of $2 \frac{\text{obstacles}}{\text{km}}$.

$$r_{L,unocc} = \sqrt{\frac{A_{unocc}}{A_{tot}}} \quad (7)$$

$$\bar{G} = \frac{r_{L,unocc} \cdot L}{N_{int}} \quad (8)$$

$$\Rightarrow N_{int} = \frac{r_{L,unocc} \cdot L}{\bar{G}} \quad (9)$$

A third method, is to employ ray casting to estimate linear obstacle density by measuring how many polytopes are intersected by rays cast in the direction of travel at random locations, normalizing each count by ray length, and averaging this over the entire field. As obstacles shrink, this value decreases, as shown in figure 14.

Plotting all three of these encounter rate estimates concurrently, as shown in figure 15, shows that the estimate from occupancy ratio grows exponentially for high departure ratios, which is expected as gap size approaches zero. The estimate based on linear obstacle density is constant, as expected for obstacle fields of the same obstacle density. The estimate from ray casting is near zero for departure ratios of zero, as expected because obstacles are nearly completely shrunk at this obstacle density, and then approaches a value just above the maximum of the estimate based on linear obstacle density. The

estimate based on ray casting is expected to be the most accurate, however it requires the most map knowledge and computational resources. For this reason, the estimate from linear point density can be useful if actual polytope locations are not available but their density is. The estimate based on occupancy ratio can be useful for estimating the upper-bound of encounter rate that can occur in high departure ratio maps. Section 4 utilizes this feature to create an upper-bound length cost ratio estimate.

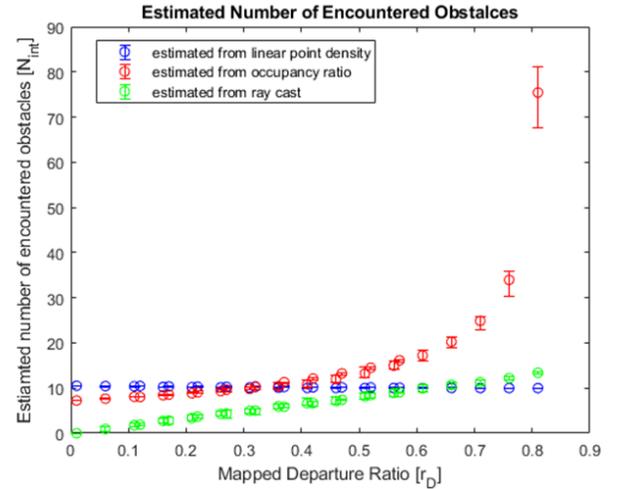


Figure 15: All three methods for estimating obstacle encounter rate, calculated for maps of varied departure ratios.

3.3. Length Cost Ratio Estimate

Now that the average obstacle geometry and the estimated number of encounters, N_{int} , are both known, the estimate can be developed for the length cost required to route around N_{int} -obstacles of average-size. The estimated obstacle encounter rate, N_{int} , informs the predicted spacing of obstacles along the path, assuming obstacles are evenly spaced along the distance from the start to the goal, L_E . The estimate for deflection per obstacle, R_{small} , is derived from the chosen side length for each analyzed polytope vertex, S_{small} , and the smaller, chosen divergence angle associated with that side length, θ_{small} , as shown in equation (10).

$$R_{small} = S_{small} \cdot \sin(\theta_{small}) \quad (10)$$

The average of the estimated deflection per obstacle is then calculated for the entire obstacle field, called \bar{R}_{small} , and used with the estimated obstacle encounter rate to produce an estimate for total path length, and therefore length cost, as shown in equation (11).

$$r_{LC} = \frac{L_P}{L_E} \approx \frac{N_{int} \cdot (\bar{R}_{small}^2 + (\frac{L_E}{N_{int}})^2)^{\frac{1}{2}}}{L_E} \quad (11)$$

This is visually described in figure 16.

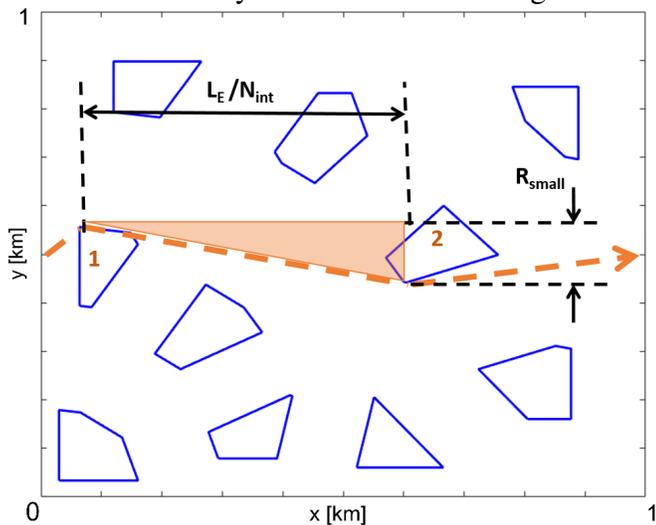


Figure 16: The estimated obstacle encounter rate, N_{int} , informs the predicted spacing of obstacles along the path while the estimate for deflection per obstacle, R_{small} , informs the predicted deflection perpendicular to the path at each encountered obstacle.

4. COMPARISON TO PATH PLANNING SIMULATION DATA

To test the navigation cost estimation method, simulated obstacle fields were created by forming a Voronoi diagram about points from the Halton set, as shown in figure 17 [4]. The Voronoi diagram consists of line segments placed such that the two closest points from the Halton set are equidistant [7]. The Halton set is a pseudo-random point set that ensures more even spacing than a truly random data set [8]. The boundaries to form convex polytopes are then shrunk inwards, towards the Halton points from the Voronoi boundaries, until the polytopes have the desired radii [4]. Maps were generated with 100 obstacles in a 1-kilometer-by-1-kilometer square area, obstacle radii were set between 1 and 81 meters, and standard deviation of obstacle radii was set between 0 and 0.32 meters. The cost estimation algorithm was evaluated on each of these maps and compared to the cost from measuring the performance of a path planning algorithm, bounded A* in this case [4], over maps with the same parameters.

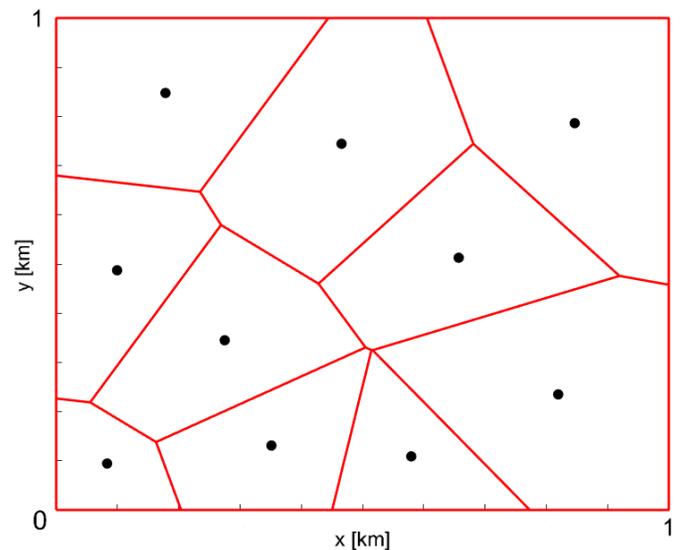


Figure 17: An example of a Voronoi diagram, shown in red, formed around Halton set points, shown in black.

In comparing the estimated cost data to the measured cost data, the estimate proves to be an accurate proxy for path planning. Plotting predicted costs over simulated cost data in figure 18 shows that the cost with the estimated obstacle count from occupancy ratio follows the higher standard deviation trends from path planning, forming an upper-bound on the estimate, while the cost estimated with the obstacle count from ray casts follows the trends of the gamma distribution fit mean of the path planning data. This data is also shown against a gamma distribution fit to the empirical data in figure 19.

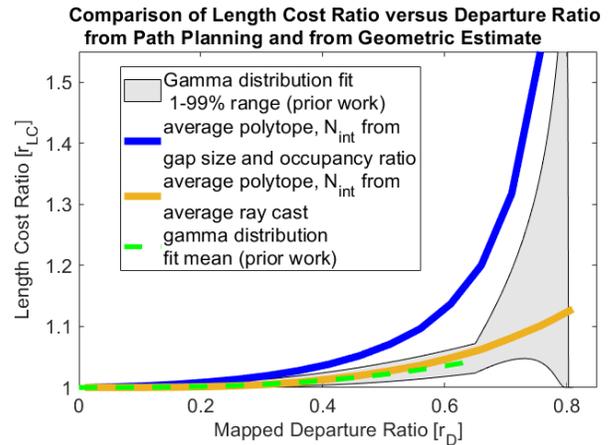


Figure 19: The same predicted length cost ratio curves as shown in figure 18, shown against the gamma distribution fit data range and gamma distribution fit mean. Note that the unusual shape of the gamma distribution fit range after a mapped departure ratio of 0.65 occurs because the range in length cost ratio increases dramatically as obstacle fields become harder to navigate at higher departure ratios and the fit begins to fail [4].

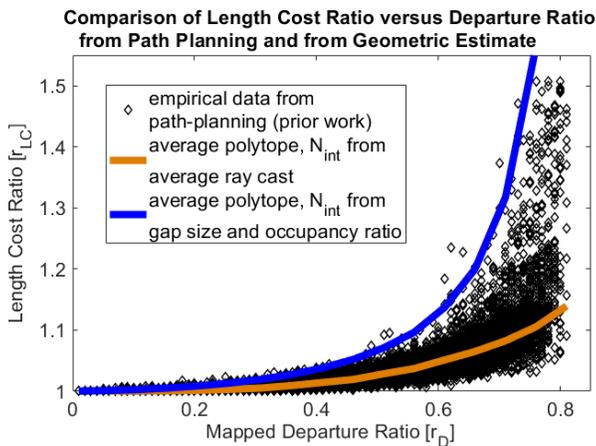


Figure 18: The predicted length cost curves as calculated with two different encountered obstacle rate methods plotted against a scatter of length cost data empirically observed from path planning simulations. The same predicted length cost ratio curves shown against a gamma distribution fit mean are shown in figure 19.

The cost estimate also proves to be more time efficient than path planning simulation. For a 4000-obstacle field, time profiling shows 29.4s wall time for the estimate entire call stack, with 0.7s self-time spent in the prediction algorithm code. Path planning with bounded A* through an obstacle field of the same obstacle density, average radius, and radius standard deviation takes 203.8s wall time, with 22.9s self-time spent in bounded A*. This means the prediction code is 7-times faster by wall time, and the algorithm is 33-times faster than bounded A* by self-time. Additionally, there are approximately 100-times as many points in the empirical scatter as were used in each theoretical curve. Accounting for the additional simulations required for the path planning approach gives an improvement of 700-times by wall time and

3200-times by self-time. The total wall-time required to generate path planning data for a single departure ratio is 101,885s (1 day, 4 hr, 18 min, 5 s) while the total wall-time required to generate navigation cost estimate data for a single departure ratio is 147 s (2 min, 27 s). This data is summarized in Table 1.

Table 1: Table comparing the time efficiency of running path planning simulations versus the geometric length cost prediction algorithm. Time profiling was performed using an 8 core, 16 thread, 3.8 GHz CPU with 48 GB of RAM.

	Bounded A*	Geometric Cost Estimation
Total wall time per trial [s]	203.8	29.4
Self-time in algorithm per trial [s]	22.9	0.7
Number of trials per departure ratio	500	5
Total time per departure ratio [s]	101,885	147

As mentioned in Section 3.1, one caveat with the cost estimation algorithm is that choosing the smaller angle does not necessarily lead to the shortest path. This is because the divergence angle selection algorithm cannot look ahead like a path planning algorithm to see when a larger divergence from the Euclidean distance would yield a shorter path, rather, it always selects the smallest divergence from the Euclidean distance, even when this yields a poor scenario later. An example of this edge case is shown in figure 20. For this edge case to occur, S_{small} must be sufficiently large and S_{big} must be sufficiently small. For the scope of this work, because the Halton set was used to generate obstacle locations, ensuring more even spacing than random points [8],

it can be assumed that obstacle aspect ratios are relatively square and therefore $S_{small} \approx S_{big}$ in many cases. Future work could modify this algorithm to take obstacle side length into account and experiment with maps generated from less uniformly spaced point sets.

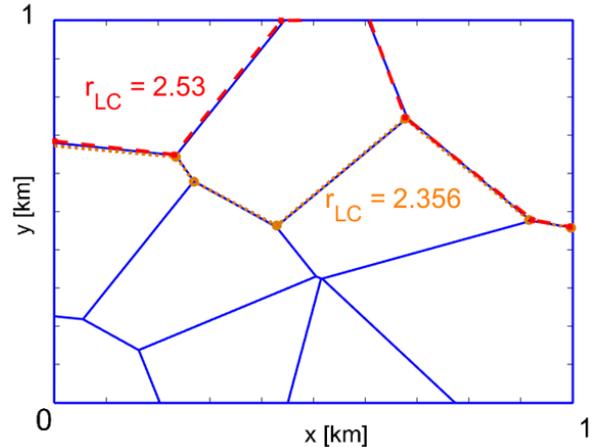


Figure 20: An example of the edge case where selecting the smaller divergence angle does not lead to a shorter path. The upper path around the center polytope, shown in red, deflects by a smaller angle but ultimately leads to a longer path while the larger deflection, shown in orange, would have resulted in a shorter overall path.

Another potential issue with this cost estimation algorithm is that it assumes there is no cumulative error in the direction normal to the travel direction caused by modeling path planning as always choosing to route around obstacles on the sides with small divergence angles. In other words, there is an assumption that, if the vehicle always routes around the small side of an obstacle, the vehicle will not always a turn to one side. The obstacle fields used in testing this algorithm started as fully tessellated maps of polytopes. Polytope radii were then shrunk towards their centroids to produce gaps between obstacles. Therefore the geometry of adjacent obstacles is related, i.e. one obstacle having a small divergence angle on the right side of the

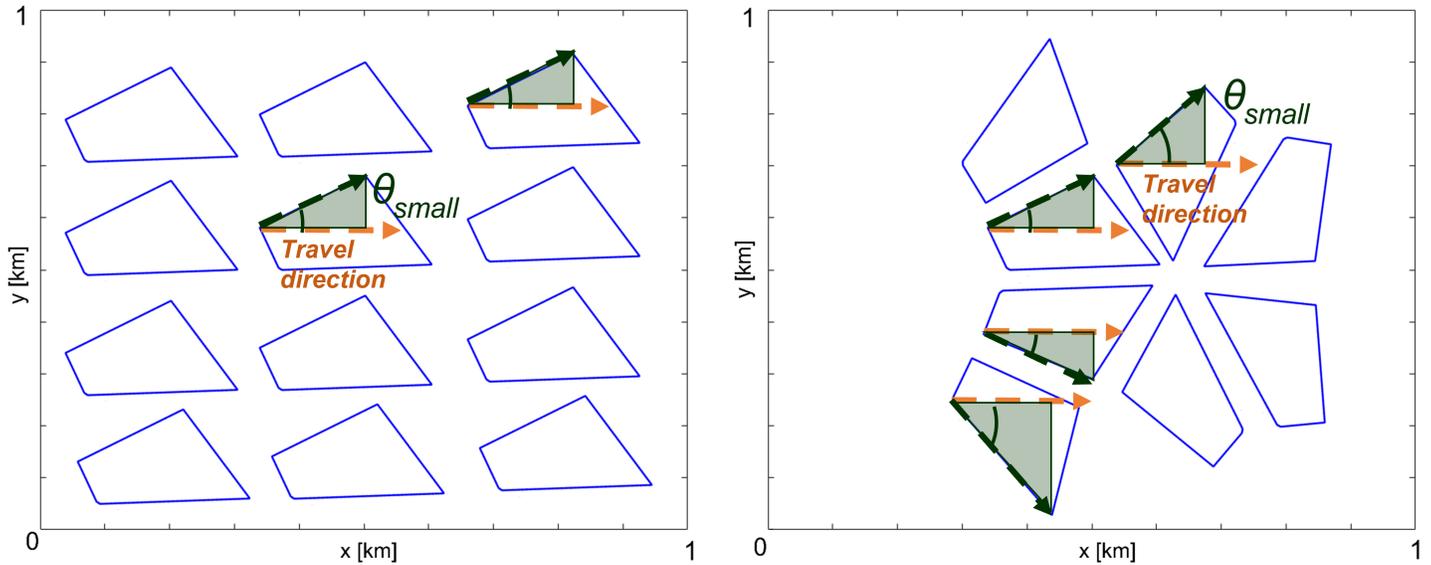


Figure 21: On the left is an example an obstacle field that is not generated from a tessellation and therefore has a bias in the relative positioning of small divergence angles with respect to the travel direction. Here, all small divergence angles are shown to be to the left of the travel direction. On the right is an example of an obstacle field generated from a tessellation which therefore has small divergence angles appearing to the left and right of the travel direction.

travel direction implies the geometry of the adjacent obstacle would have a small divergence angle on the left side of the travel direction. If an obstacle field was generated from a method that does not start from a fully tiled field, as shown in figure 21, it is possible that the planner model would accumulate a lateral error while routing towards the goal longitudinally in the travel direction as obstacles may consistently have shallow angles on the same side of the travel direction.

5. CONCLUSIONS AND FUTURE WORK

From looking solely at map geometry, without any path planning simulation, the length cost of navigating through the obstacle field can be estimated, both in the most likely case and in a theoretical worst case. While data from the path planning simulations could be used to estimate the length cost for maps of similar complexity, i.e. maps at the same departure ratio, the length cost estimation

algorithm can be used to predict length cost for a specific map, not just a geometrically similar map. Additionally, because this cost estimation is done without the planner in the loop, this can be obtained prior to designing and implementing a path planner, requiring only a map of obstacle field geometry.

There are several areas for future work to explore. As shown above in figure 20, there is an edge case that this algorithm is susceptible to that leads to selecting sub-optimal paths when large divergence leads to a shorter overall path length. Mitigations for this could be developed such that the model path planner's decisions more closely resemble the decisions of an actual path planner by including a consideration of obstacle side length, as well as divergence angle, in the cost estimation algorithm. The other edge case, described in figure 21 could also be explored by repeating this analysis on obstacle fields that are not generated from tessellations. Additionally, real-world environments

could be analyzed and abstracted into the binary convex polytope maps used in this work. The path planner and cost prediction algorithm could then be run on these maps and compared to the results from using geometric, simulated maps created from the Halton set to see if this cost estimate is relevant for real-world maps.

[8] J. Halton and G. Smith, “Radical-Inverse Quasi-Random Point Sequence,” *Communications of the ACM*, vol. 7, no. 12, pp. 701–702, 1964.

6. REFERENCES

- [1] J. Pentzer, K. Reichard and S. Brennan, “Energy-based path planning for skid-steer vehicles operating in areas with mixed surface types,” *2016 American Control Conference (ACC)*, pp. 2110-2115, 2016.
- [2] M. Lepetič, G. Klančar, I. Škrjanc, D. Matko, and B. Potočnik, “Time optimal path planning considering acceleration limits,” *Robotics and Autonomous Systems*, vol. 45, pp.199-210, 2003.
- [3] K. Reichard, “Integrating self-health awareness in autonomous systems”, *Robotics and Autonomous Systems*, vol. 49, pp. 105-112, 2004.
- [4] S. A. Tau, “The Effects of Path-Planning Uncertainty on Intelligent Vehicle Performance” PhD Thesis, The Pennsylvania State University, University Park, PA, 2021.
- [5] M. McCullough, J. Jayakumar, J. Dasch, and D. Gorsich, “Developing the Next Generation NATO Reference Mobility Model,” *2016 NDIA Ground Vehicle Systems Engineering and Technology Symposium*, 2016.
- [6] Y. Zhu, Y. Wu and B. Li, “Trajectory Improves Data Delivery in Urban Vehicular Networks,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 4, pp. 1089-1100, 2014.
- [7] M. de Berg, O. Cheong, M. van Kreveld, and M. Overmars, *Computational Geometry*, 3rd ed., Berlin: Springer, 2008.