$\epsilon^{\star}$+: An Online Coverage Path Planning Algorithm for Energy-constrained Autonomous Vehicles

Zongyuan Shen, James P. Wilson, Shalabh Gupta

Laboratory of Intelligent Networks and Knowledge-Perception Systems

Department of Electrical & Computer Engineering, University of Connecticut, Storrs, CT, USA

IEEE/MTS Global Oceans 2020, Singapore-US Gulf Coast

Outline of the Presentation

- Introduction
- Problem Statement
- Methodology
- Simulation Validations
- Conclusions and Future Work

Introduction

- Objective: Develop an online coverage path planning algorithm for energy-constrained vehicles
- Background:

- Battery powered coverage (BPC) algorithm [1]
  - Generate difficult-to-follow and less desirable contour path
  - Produce redundant paths with longer length
- Energy-constrained coverage algorithm [2]
  - Rely on the *a priori* knowledge of the environment, thus it's performance can degrade if the knowledge is incomplete or incorrect
- $\epsilon^\star$ algorithm [3] for complete coverage
  - Use Exploratory Turing Machine (ETM) as a supervisor
  - Limitation: it didn't consider the limited energy capacity of the vehicle, i.e., while executing a coverage trajectory, the vehicle has to return to the charging station for a recharge before the battery runs out

Introduction

- Highlights of the $\epsilon^{\star}$+ Algorithm:
  - Presents online coverage path planning of unknown environments using energy-constrained vehicles
  - The remaining energy is monitored throughout the coverage process
  - Before the battery runs out, the vehicle returns to the charging station by following the computed retreat trajectory
  - After getting a full recharge, the computed advance trajectory takes the vehicle to the new coverage start point to restart the coverage process
- Benefits and Contributions of the $\epsilon^{\star}$+ Algorithm:
  - Produces easy-to-follow and user-desired back-and-forth trajectories
  - Chooses a nearby unexplored cell as the new coverage start point after each recharging to avoid longer travel to the cell from where it retreated back
  - Guarantees complete coverage with minimum overlaps
  - Works in unknown environments
  - Computationally efficient for real-time implementation

Problem Statement

- Let $A \subset \mathbb{R}^2$ be the estimated area that includes the desired area to cover.
- Tiling: The set $T = \{\tau_i \subset \mathbb{R}^2 : i = 1 \ldots |T|\}$ is called a tiling of A if its elements, called cells:
  - have mutually exclusive interiors, i.e., $\tau_i^0 \cap \tau_j^0 = \emptyset, \forall i \neq j$, where $i, j \in \{1 \ldots |T|\}$, and superscript 0 denotes the interior of a cell
  - form a cover of A, i.e., $A \subseteq \cup_{i=1}^{|T|} \tau_i$, while removal of any cell destroys the covering property
- The tiling T is partitioned into two subsets:
  - Obstacle cells (T^o): they are occupied by obstacles and detected online
  - Allowed cells (T^a): these are free of the obstacles and further classified as unexplored and explored

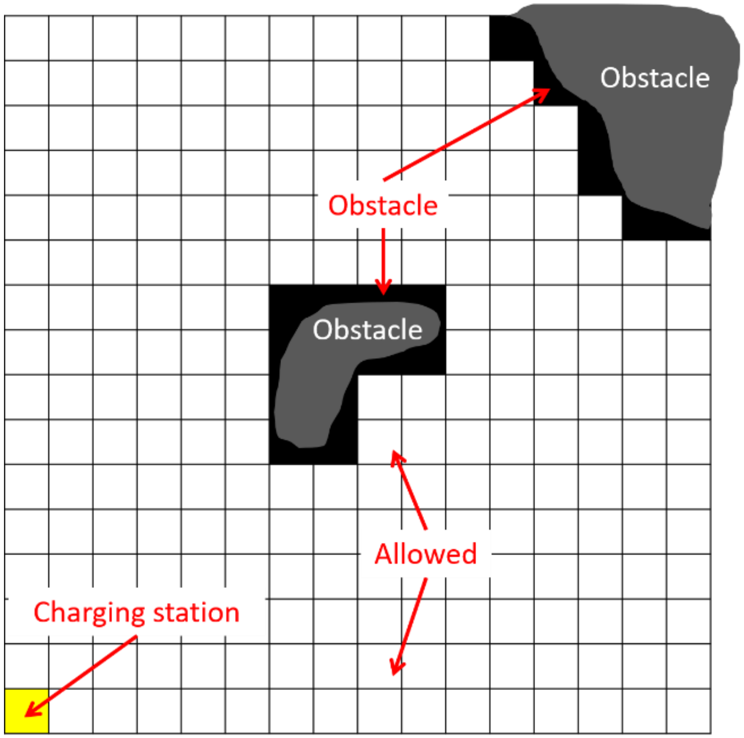


Fig. Tiling of the Search Area

Problem Statement

- Let $A(T^a)$ denote the total area of the allowed cells in $T^a \subseteq T$. Let $\tau_c \in T$ be the cell at which the charging station is located. Let $E_0 \in \mathbb{R}^+$ be the total energy that the vehicle has under full charge.
- Trajectory: A trajectory π is defined as a sequence of cells visited by the autonomous vehicle consisting of three segments: i) advance, ii) coverage, and iii) retreat.
  - The advance segment takes the vehicle from the charging station to an unexplored cell
  - The coverage segment performs coverage of unexplored cells using the $\epsilon^\star$ algorithm till the vehicle's battery depletes to the extent that allows it to go back to the charging station
  - The retreat segment brings the vehicle back to the charging station along the shorter path

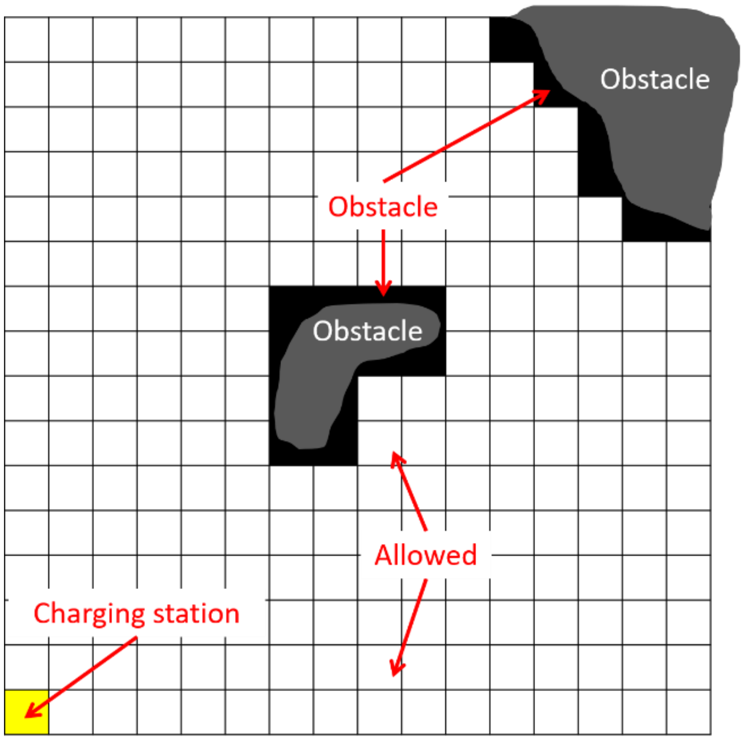


Fig. Tiling of the Search Area

- Online Energy-Constrained Coverage Problem: Let $A(\pi_n)$ denote the area covered by a trajectory π_n. Then the goal is to find an ordered set of trajectories $\Pi = \{\pi_1, \cdots, \pi_N\}$ such that:
- Each trajectory $\pi_n \in \Pi$ starts and ends at τ_c
- Each trajectory $\pi_n \in \Pi$ consumes energy $E(\pi_n) \leq E_0$
- Π forms a cover of $A(T^a)$, i.e., $A(T^a) \subseteq \cup_{n=1}^{N} A(\pi_n)$

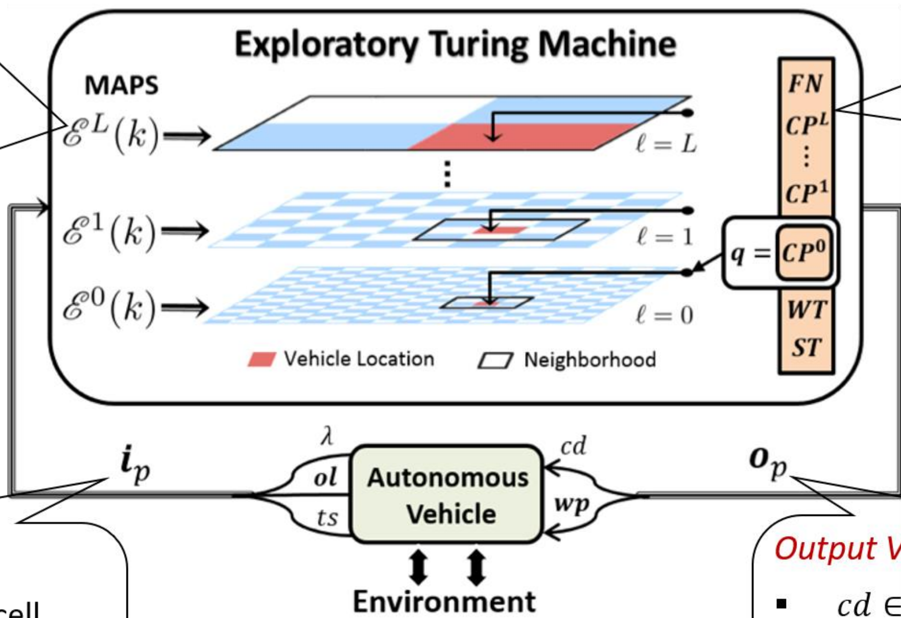
Methodology

Review of the $\epsilon^\star$ Algorithm

Exploratory Turing Machine (ETM)

Multi-scale Adaptive Potential Surfaces(MAPS)

- $\mathcal{E}^0$: time-varying potential surface at the finest level
- $\mathcal{E}^\ell, 1 \leq \ell \leq L$: time-varying potential surfaces at higher levels



Machine States:

- $ST \equiv$ Start
- $CP \equiv$ Compute
- $WT \equiv$ Waiting
- $FN \equiv$ Finished

Input Vector $\boldsymbol{i_p}$

- $\lambda \in \{1, \ldots |T|\}$: index of current cell
- $\boldsymbol{ol} \subset \{1, \ldots |T|\}$, indices of obstacle cells
- $ts \in \{cm, ic\}$, tasking status, where $cm \equiv$ *Complete*, or $ic \equiv$ *Incomplete*.

Output Vector $\boldsymbol{o_p}$

- $cd \in \{mv, tk, id, sp\}$: command to the vehicle, where: $mv \equiv$ Move, $tk \equiv$ Task, $id \equiv$ Idle, and $sp \equiv$ Stop.
- $\boldsymbol{wp} \subset \{1, \ldots |T|\}$, candidate set of new waypoints for the vehicle

❖ Summary of the $\epsilon^\star$ Algorithm :

- During the coverage process, the environmental information is encoded on the symbolic map, which is used to dynamically construct the Multi-scale Potential Surfaces (MAPS)
- By default, the ETM uses the lowest level of MAPS for generating the back-and-forth coverage path online
  - The navigation goal is selected as centroid of the cell that possesses the highest positive potential in the vehicle's local neighborhood
  - If multiple cells have the same highest positive potential, the vehicle selects the cell with the least travel cost to reach it
- It switches to higher levels of MAPS as needed to escape from a local extremum
  - It sequentially switches to higher levels of MAPS, until a coarse cell with highest positive potential is found in the local neighborhood
  - Within the selected coarse cell, the unexplored cell at the finest level is randomly picked as the goal

Methodology

Novel Features of the $\epsilon^{\star}$+ Algorithm

❖ Computation of the Retreat Trajectory:

- Each time the navigation waypoint is computed by the $\epsilon^{\star}$ algorithm, the retreat trajectory is computed from this waypoint to the charging station as the shortest path using visibility graph [4] and $A^{\star}$ search [5]
    - The visibility graph is extracted from the latest symbolically encoded tiling structure
    - The retreat path is obtained by performing $A^{\star}$ search on the visibility graph
- The expected energy consumption, for retreating to the charging station, is evaluated against the vehicle's remaining energy, such that it can maintain sufficient energy for returning to the charging station after covering the navigation waypoint
    - If the remaining energy is less than the expected energy consumption for covering the next navigation waypoint and then retreating to the charging station, then the vehicle returns back to the charging station

❖ Energy consumption: the energy consumption of the vehicle is modeled as proportional to the trajectory length for the advance and retreat segments, while it is twice this amount for coverage segment.

Methodology

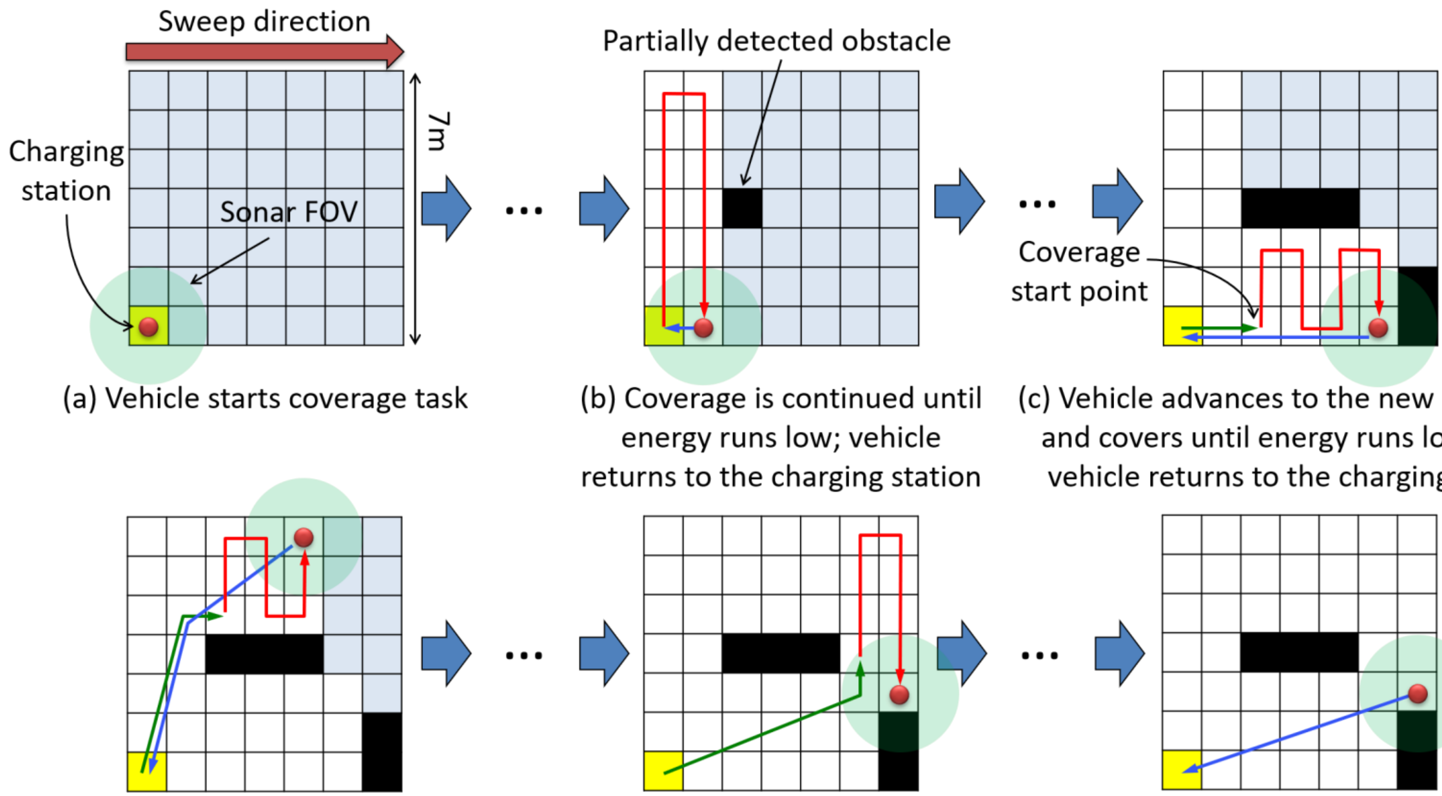
Novel Features of the $\epsilon^{\star}$+ Algorithm

❖ Computation of the Advance Trajectory:

- At the charging station, the vehicle uses the MAPS structure of the $\epsilon^{\star}$ algorithm to find a nearby unexplored point
  - The $\epsilon^{\star}$ algorithm switches to a coarse level on the MAPS until it finds an unexplored coarse cell with the highest positive potential
  - Within the coarse cell, the closest unexplored cell at the finest level is selected as the new coverage start point
- The advance trajectory from the charging station to the selected coverage start point is computed using the visibility graph and the $A^{\star}$ search

Methodology

Execution Example



(a) Vehicle starts coverage task

(b) Coverage is continued until energy runs low; vehicle returns to the charging station

(c) Vehicle advances to the new start point and covers until energy runs low again; vehicle returns to the charging station

(d) Vehicle advances to the new start point and covers until energy runs low again; vehicle returns to the charging station

(e) Vehicle advances to the new start point and covers; complete coverage is achieved

(f) Vehicle returns to the charging station and terminates the task

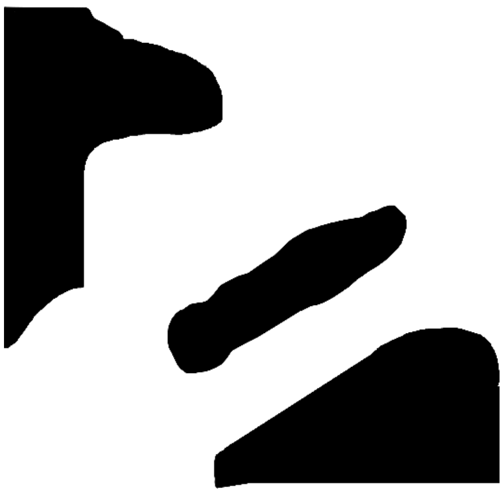
Obstacle **Explored** **Unexplored** **Advance Trajectory** **Coverage Trajectory** **Retreat Trajectory**

Simulation Validations

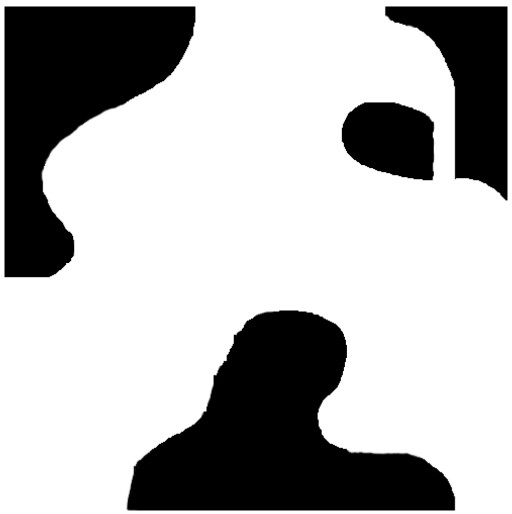
Simulation Setup

- A high-fidelity simulator called Player/Stage [6] is used for simulation
- Autonomous Vehicle: a Pioneer AT2 of dimensions $0.44\mathrm{m} \times 0.38\mathrm{m} \times 0.22\mathrm{m}$ was used with constraints:
  - Total energy capacity: 320 units
  - Top speed: $0.5\mathrm{m/s}$
  - Maximum acceleration: $0.5\mathrm{m/s}^2$
  - Top turning rate: $60\ \mathrm{degree/s}$
- Sensing systems
  - Sonar: 16-beam with detection range of 5m
- Search Area: the search area is of size $50\mathrm{m} \times 50\mathrm{m}$, which is partitioned into a 50×50 tiling consisting of cells of size $1\mathrm{m} \times 1\mathrm{m}$.

Scenario 1



Scenario 2



Simulation Validations

Simulation Results

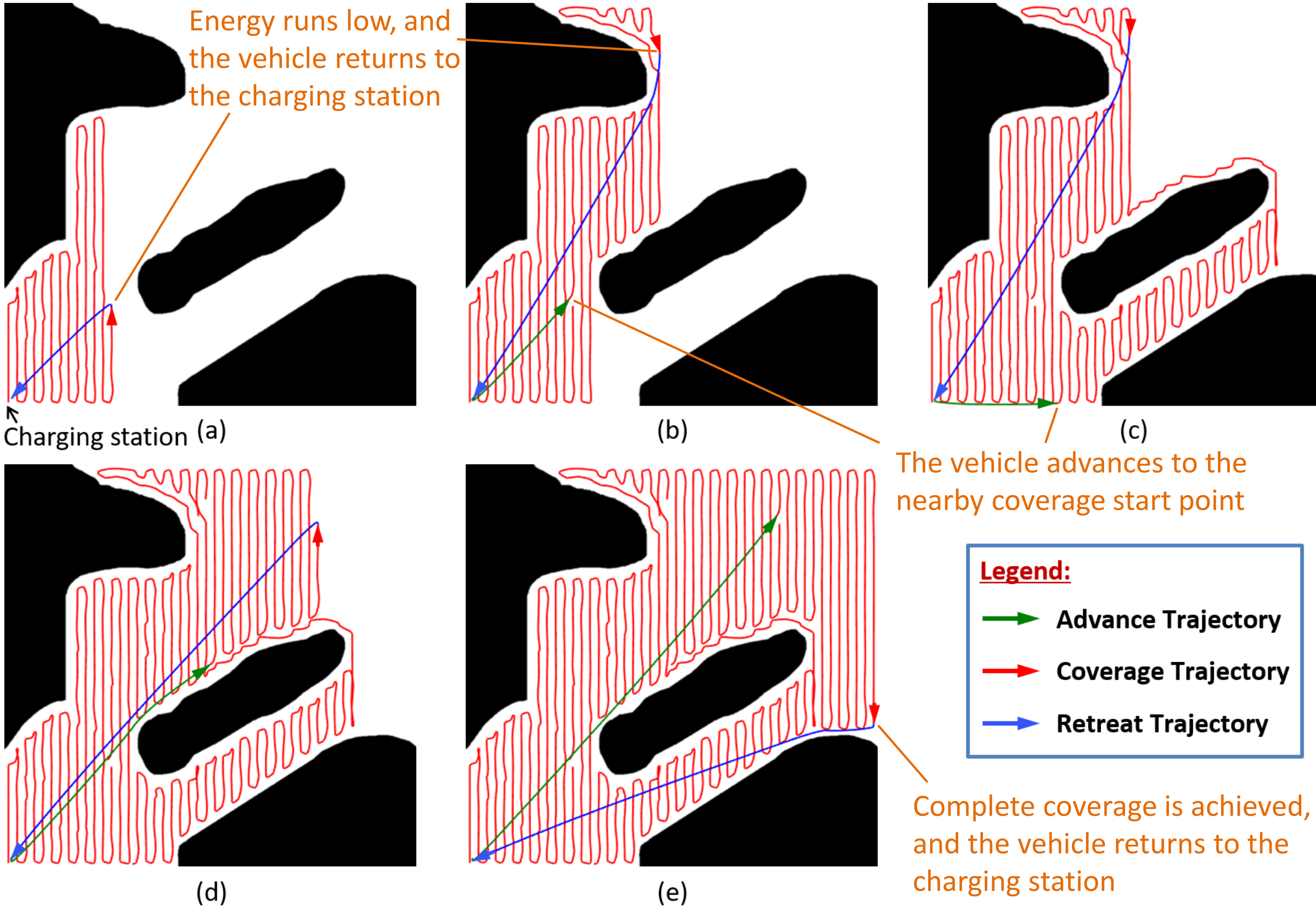


Fig. The vehicle trajectory showing complete coverage of the search area of scenario 1

Simulation Validations

Simulation Results

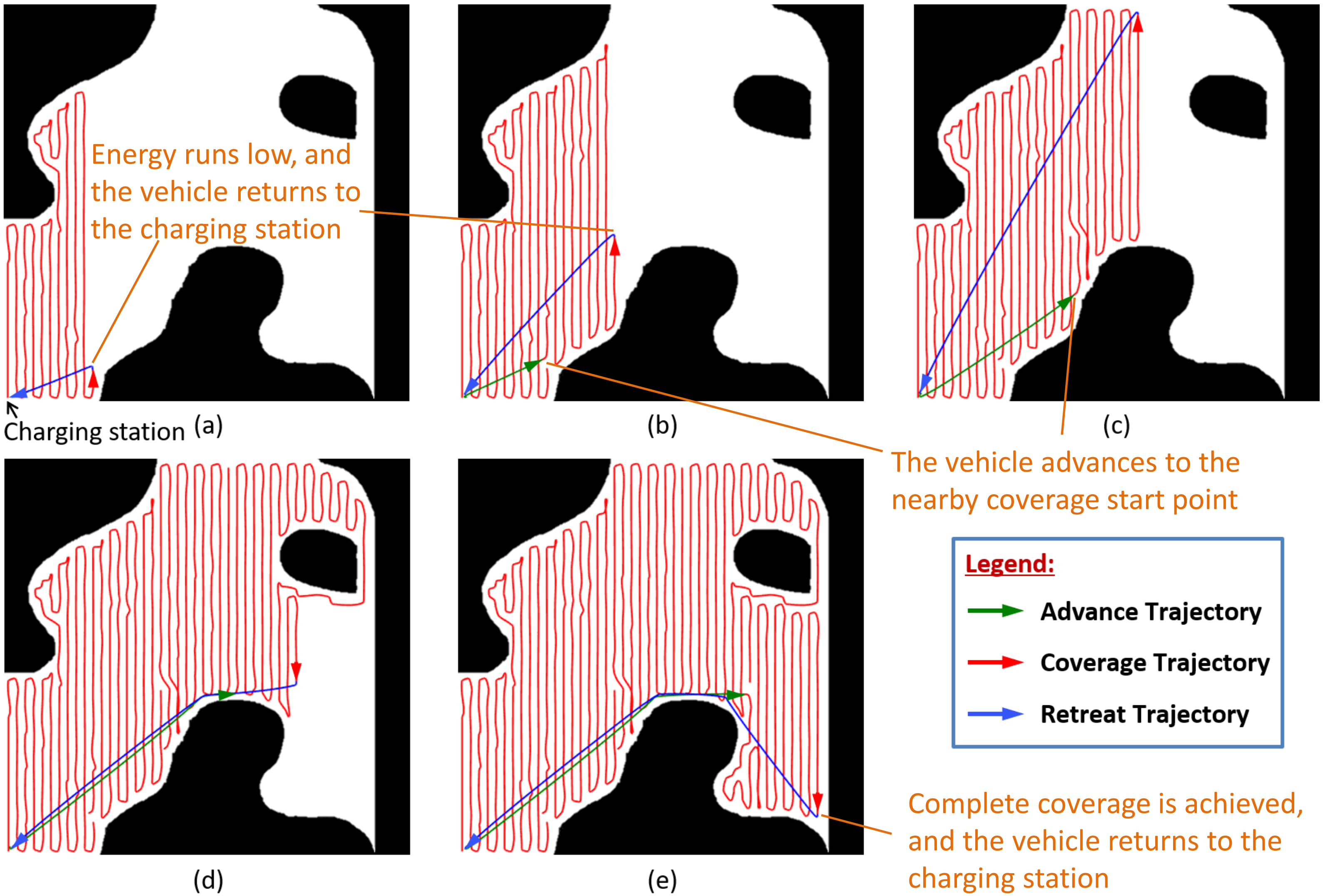


Fig. The vehicle trajectory showing complete coverage of the search area of scenario 2

Conclusions & Future Work

Conclusions

- An online algorithm for complete coverage of unknown environments using energy-constrained vehicle is proposed
- The vehicle is able to autonomously navigate in an unknown environment while avoiding the obstacles and maintaining the sufficient energy for retreating to the charging station as needed
- The performance of proposed method is validated on two complex scenarios in Player/Stage

Future Work

- Extend the algorithm to the problem to consider kinematic constraints, multi-agent systems, and risk
- Variable-speed and acceleration constraints may be considered to make the vehicle motion more realistic
- A sample-based approach for coverage path planning will be investigated to enable faster and more adaptive waypoint selection for real-time decision in dynamic environments

References

[1] I. Shnaps and E. Rimon, “Online coverage of planar environments by a battery powered autonomous mobile robot,” IEEE Transactions on Automation Science and Engineering, vol. 13, no. 2, pp. 425–436, 2016.

[2] M. Wei and V. Isler, “Coverage path planning under the energy constraint,” in 2018 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2018, pp. 368–373.

[3] J. Song and S. Gupta, “$\epsilon^{\star}$: An online coverage path planning algorithm,” IEEE Trans. Robot., vol. 34, pp. 526–533, 2018.

[4] T. Lozano-Perez and M. A.Wesley, “An algorithm for planning collisionfree paths among polyhedral obstacles,” Communications of the ACM, vol. 22, no. 10, pp. 560–570, 1979.

[5] P. E. Hart, N. J. Nilsson, and B. Raphael, “A formal basis for the heuristic determination of minimum cost paths,” IEEE transactions on Systems Science and Cybernetics, vol. 4, no. 2, pp. 100–107, 1968.

[6] B. Gerkey, R. T. Vaughan, and A. Howard, “The player/stage project: Tools for multi-robot and distributed sensor systems,” in Proceedings of the 11th international conference on advanced robotics, vol. 1, 2003, pp. 317–323.


Q&A

Methodology

Review of the $\epsilon^\star$ Algorithm

Dynamically Constructed Multi-scale Potential Surfaces (MAPS)

❖ Level 0 of MAPS

- Environmental Information Encoding:

The environmental information is encoded on each cell based on its physical state:

  - U = Unexplored
  - E = Explored
  - O = Obstacle

- Construction of Potential Surface: for each cell τ_{α^0} with state $S_{\alpha^0}(k)$ at time k:

$$\mathcal{E}_{\alpha^0}(k) = \begin{cases} -1, & \text{if } S_{\alpha^0}(k) = O \\ 0, & \text{if } S_{\alpha^0}(k) = E \\ B_{\alpha^0}, & \text{if } S_{\alpha^0}(k) = U \end{cases}$$

where $B = \{B_{\alpha^0} \in \{1, \dots, B_{\max}\}, \alpha^0 = 1, \dots |T^0|\}$ is a time-invariant exogenous potential field. It is designed *offline* to have plateaus of equipotential surfaces along each column of the tiling.

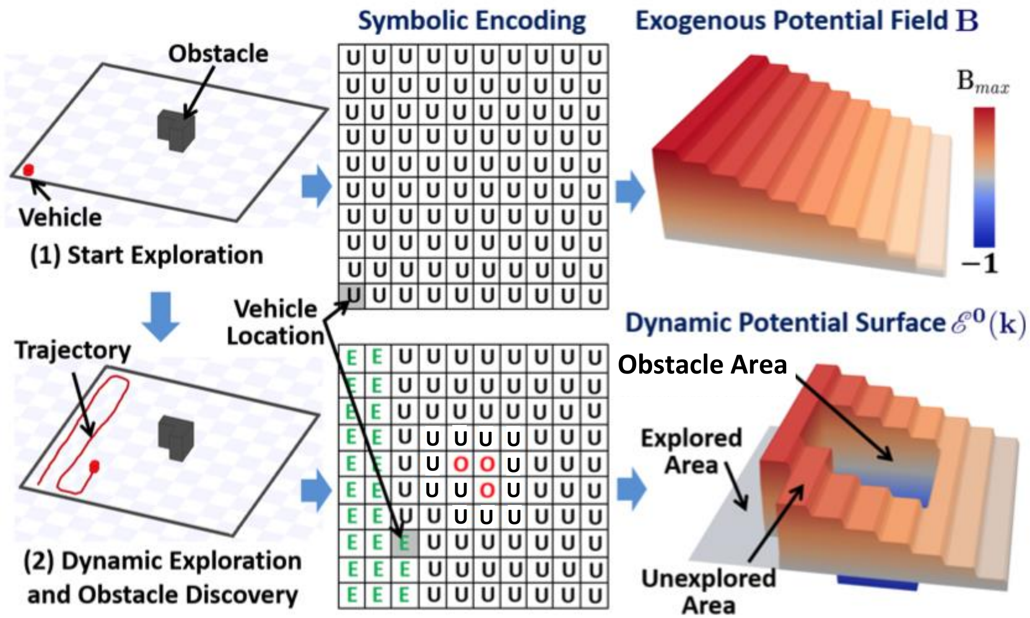


Fig. An example of constructing the potential surface

Dynamically Constructed Multi-scale Potential Surfaces (MAPS)

Note: Higher levels of MAPS are used to prevent the local extrema problem.

❖ Levels $\ell = 1,2,\ldots L$ of MAPS

- Potential Surfaces $\mathcal{E}^\ell$, $\ell = 1,\ldots L$, are constructed by assigning τ_{α^ℓ} the *average* potential generated by all the *unexplored* cells within τ_{α^ℓ}, such that

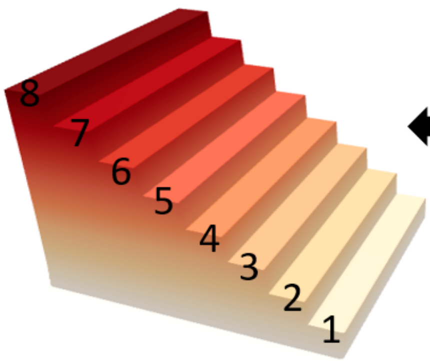
$$\boxed{\mathcal{E}_{\alpha^\ell}(k) = p^U_{\alpha^\ell}(k) \cdot \bar{B}_{\alpha^\ell}}$$

where $\bar{B}_{\alpha^\ell}$ is the mean exogenous potential of τ_{α^ℓ}, and $p^U_{\alpha^\ell}(k)$ is the probability of *unexplored* cells in τ_{α^ℓ}.

Methodology

Review of the $\epsilon^\star$ Algorithm

Updates of MAPS at Level 1



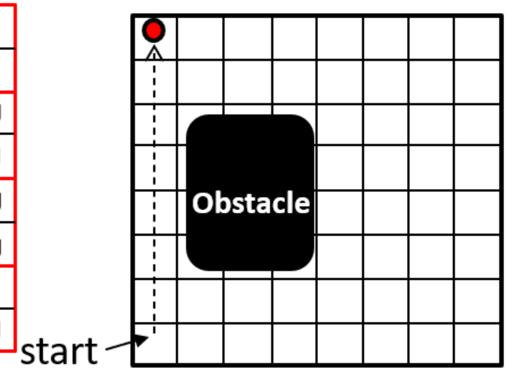
Potentials B at Level 0

8	7	6	5	4	3	2	1
8	7	6	5	4	3	2	1
8	7	6	5	4	3	2	1
8	7	6	5	4	3	2	1
8	7	6	5	4	3	2	1
8	7	6	5	4	3	2	1
8	7	6	5	4	3	2	1
8	7	6	5	4	3	2	1

Symbolic encodings at Level 0

E	U	U	U	U	U	U	U
E	U	U	U	U	U	U	U
E	O	O	O	U	U	U	U
E	O	O	O	U	U	U	U
E	O	O	O	U	U	U	U
E	O	O	O	U	U	U	U
E	U	U	U	U	U	U	U
E	U	U	U	U	U	U	U

Vehicle trajectory



MAPS update on Level 1

Average potential $\bar{B}$

7.5	5.5	3.5	1.5
7.5	5.5	3.5	1.5
7.5	5.5	3.5	1.5
7.5	5.5	3.5	1.5

×

Update probabilities $p^U_{\alpha^\ell}$

0.5	1	1	1
0	0	1	1
0	0	1	1
0.5	1	1	1

⇒

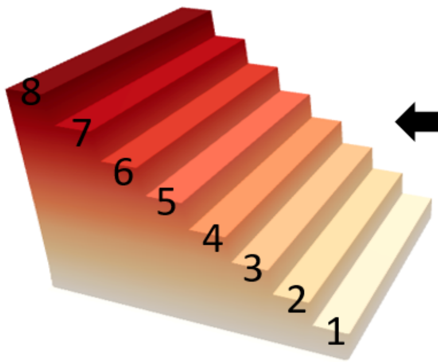
MAPS update

3.75	5.5	3.5	1.5
0	0	3.5	1.5
0	0	3.5	1.5
3.75	5.5	3.5	1.5

Methodology

Review of the $\epsilon^\star$ Algorithm

Updates of MAPS at Level 2



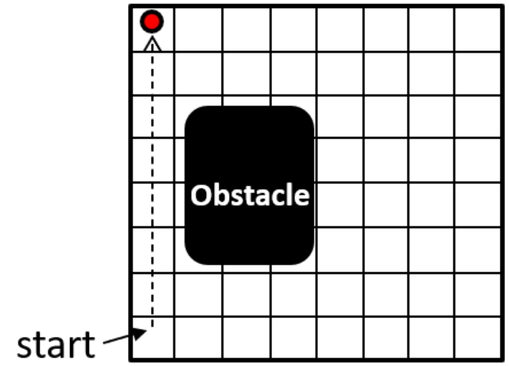
Potentials B at Level 0

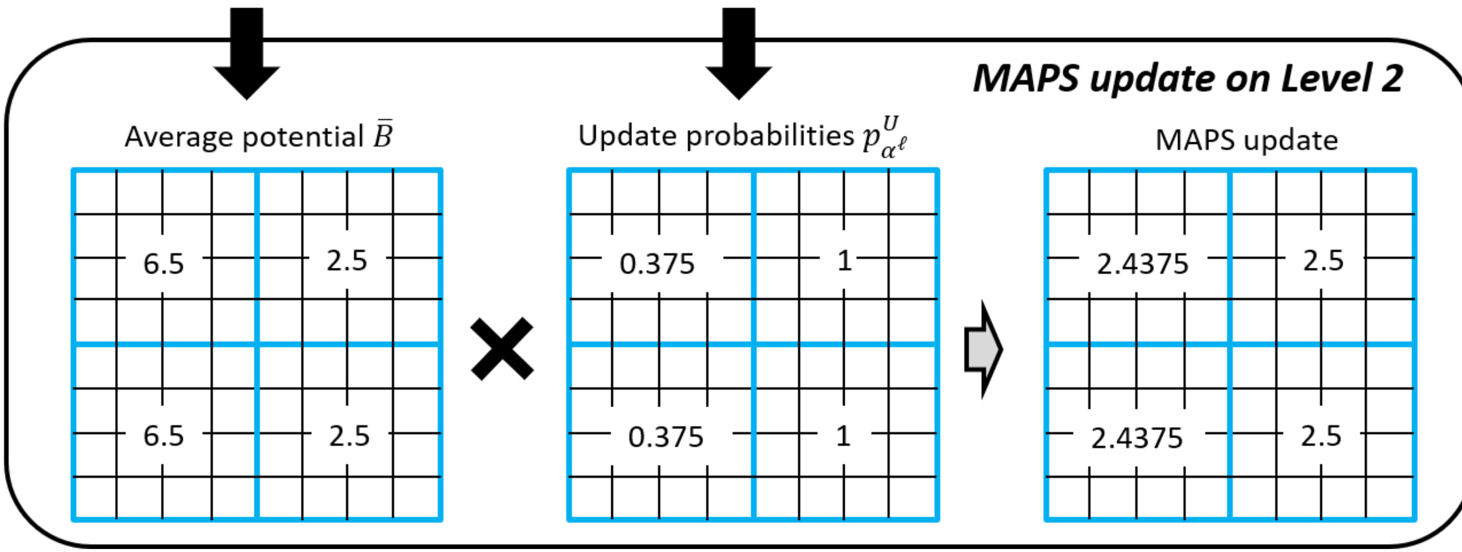
8	7	6	5	4	3	2	1
8	7	6	5	4	3	2	1
8	7	6	5	4	3	2	1
8	7	6	5	4	3	2	1
8	7	6	5	4	3	2	1
8	7	6	5	4	3	2	1
8	7	6	5	4	3	2	1
8	7	6	5	4	3	2	1

Symbolic encodings at Level 0

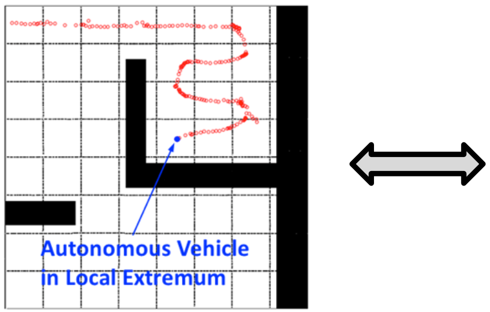
E	U	U	U	U	U	U	U
E	U	U	U	U	U	U	U
E	O	O	O	U	U	U	U
E	O	O	O	U	U	U	U
E	O	O	O	U	U	U	U
E	O	O	O	U	U	U	U
E	U	U	U	U	U	U	U
E	U	U	U	U	U	U	U

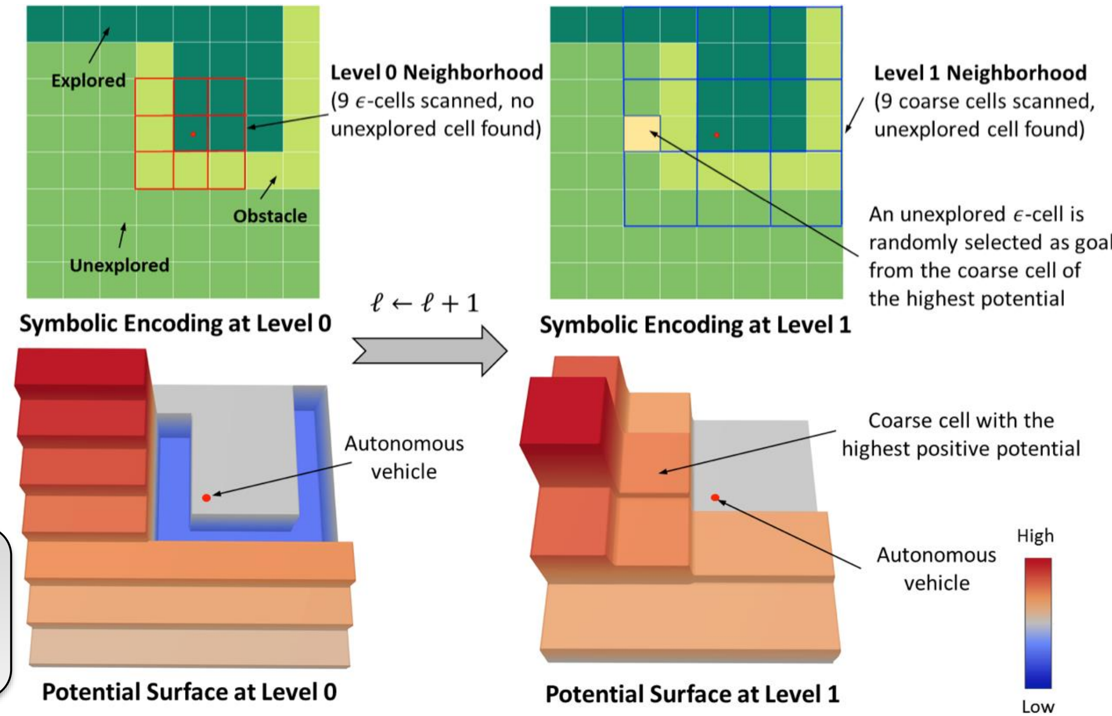
Vehicle trajectory





An Example of using MAPS to Prevent the Local Extrema Situation





Local Extrema: no unexplored cells are available in the local neighborhood on Level 0.

Low Complexity: time complexity of $O(|N^0| + L \cdot |N^{\ell}|)$ to compute waypoints, where N^{ℓ} is the local neighborhood on Level ℓ of MAPS, $\ell = 0,1, \dots, L$.