

A Non-uniform Sampling Approach for Fast and Efficient Path Planning

James P. Wilson^{†*} Zongyuan Shen[†] Shalabh Gupta[†]

Abstract—In this paper, we develop a non-uniform sampling approach for fast and efficient path planning of autonomous vehicles. The approach uses a novel non-uniform partitioning scheme that divides the area into obstacle-free convex cells. The partitioning results in large cells in obstacle-free areas and small cells in obstacle-dense areas. Subsequently, the boundaries of these cells are used for sampling; thus significantly reducing the burden of uniform sampling. When compared with a standard uniform sampler, this smart sampler significantly 1) reduces the size of the sampling space while providing completeness and optimality guarantee, 2) provides sparse sampling in obstacle-free regions and dense sampling in obstacle-rich regions to facilitate faster exploration, and 3) eliminates the need for expensive collision-checking with obstacles due to the convexity of the cells. This sampling framework is incorporated into the RRT* path planner. The results show that RRT* with the non-uniform sampler gives a significantly better convergence rate and smaller memory footprint as compared to RRT* with a uniform sampler.

Index Terms—non-uniform sampling; autonomous vehicles; sampling-based algorithms; near-optimal path planning

I. INTRODUCTION

Autonomous vehicles are becoming increasingly useful and cost-effective for a variety of tasks in many scientific expeditions. Specifically, autonomous underwater vehicles (AUVs) have been extensively used for exploration [1], [2], data collection (e.g., ocean salinity and temperature), ship hull cleaning [3], underwater pipeline and cable monitoring [4], marine wildlife monitoring [5], seabed mapping [6], [7], [8], [9], [10], [11], oil spill cleaning [12], mine hunting [13], [14], and other marine research. Despite recent advances, their autonomy is limited. AUV missions might require on-demand path synthesis in unknown or dynamic environments [15], [16]. The autonomy of AUVs is determined by the proficiency in which they can replan paths as new information becomes available [17]. It is thus of practical importance for path planners to be computationally efficient and robust when constructing the cost-minimizing paths.

A review of path planning methods for AUVs is presented in [17]. In general, path planning can be divided into two categories: grid-based and sample-based. Grid-based methods, such as A* [18], discretize the configuration space and search for the optimal solution; however, the solution quality

depends on the grid resolution and suffers from the curse of dimensionality. On the other hand, sample-based methods, such as RRT* [19], [20], are becoming increasingly popular for on-demand motion planning since they can 1) find feasible solutions quickly in high-dimensional spaces and 2) approach the optimal solution as the number of samples increases. However, while these methods perform well in relatively open spaces, the uniform samplers often struggle with narrow passages since the chance of placing a sample within the passage with a collision-free connection is very small [21]. Although a feasible path is eventually found, it requires a large number of samples and computationally expensive collision checks, thus providing a slow convergence rate.

To overcome the limitations of sample-based approaches, researchers have recently focused on sampling only the *critical regions* [22], [23]. These regions are often the entrances of a passage connecting two larger areas (e.g., doorways). The idea is that by biasing a large number of samples to these regions, sample-based planners can quickly find a feasible path. Furthermore, it encourages sparse sampling in obstacle-free regions, thus facilitating faster convergence. The identification of these critical regions has mostly been achieved using deep learning models trained on supplied examples of the shortest path [24], [25], [26]. While the results are promising, these models require large and diverse datasets for training and are not guaranteed to find these regions. Furthermore, these approaches might not scale or be robust in new environments.

In this paper, we present a new path planning framework that combines the best features of grid-based and sample-based path planners for intuitively identifying, sampling and planning along the critical regions. These regions are identified using a new method inspired by renormalization group theory [27], [28], [29]. This method partitions the region which creates non-overlapping large convex cells in obstacle-free spaces, and small convex cells in obstacle-dense regions. The critical regions are the obstacle-free boundaries of these cells. Then, a feasible path is found quickly by sampling only in these regions using a sample-based path planner. In particular, the convexity of the cells joining critical regions eliminates the need for collision checking, which significantly reduces the computation time. Finally, once a feasible path is found, a local search is performed to find the (near)optimal solution.

The rest of this paper is organized as follows. Section II formulates the path planning problem for AUVs. Section III presents the smart sampling procedure and the path planning algorithm. Section IV shows the results on a simulated scenario, and Section V concludes this paper with recommendations for future work.

[†]Dept. of Electrical and Computer Engineering, University of Connecticut, Storrs, CT 06269, USA.

* Corresponding Author (email id: james.wilson@uconn.edu)

©2021 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

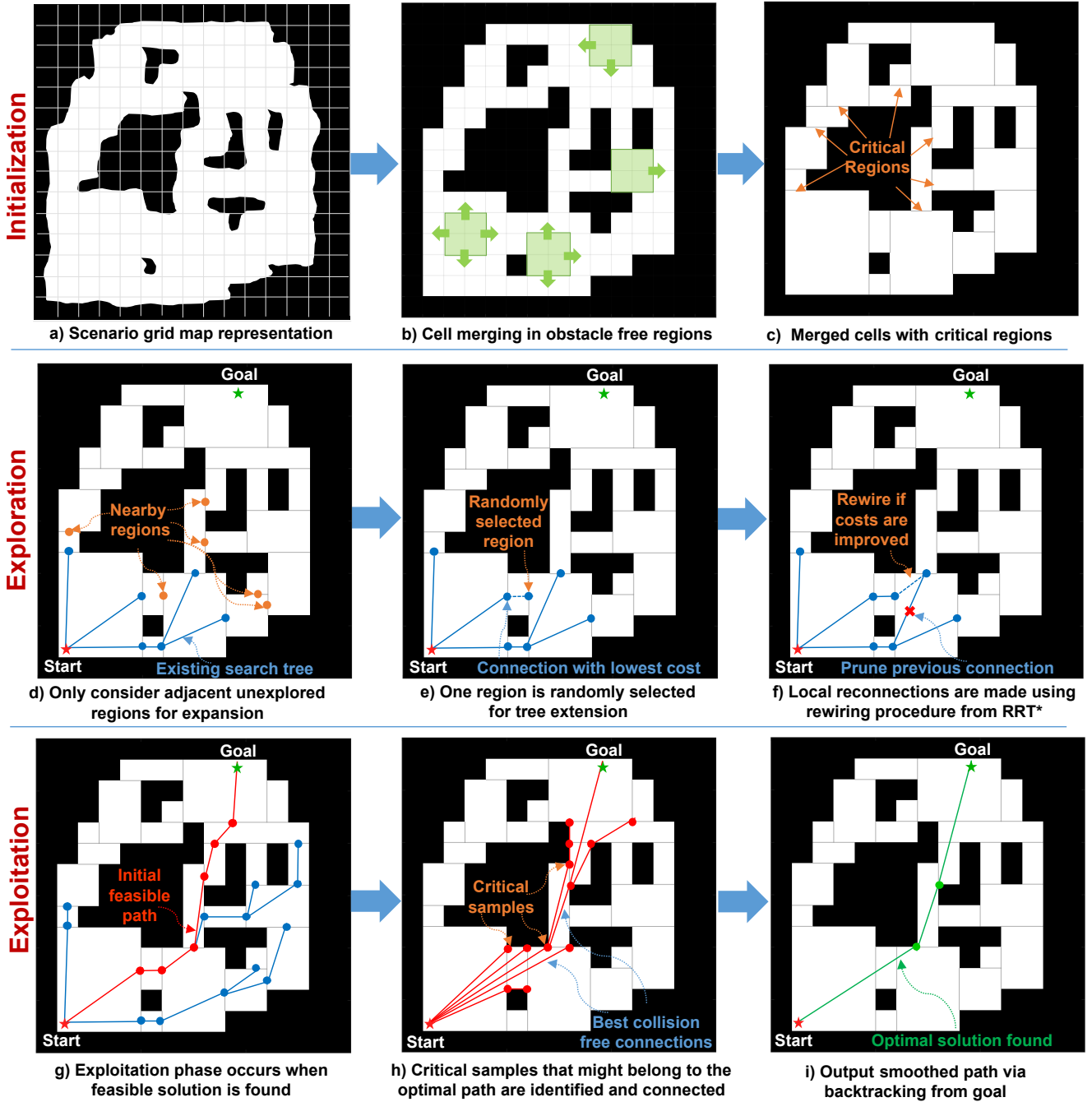


Fig. 1: Non-uniform sampling for fast and efficient RRT*-based path planning.

II. PROBLEM FORMULATION

Let $\mathcal{A} \subset \mathbb{R}^2$ be the 2D search area composed of free space \mathcal{A}_f and obstacle space \mathcal{A}_o , where $\mathcal{A} = \mathcal{A}_f \cup \mathcal{A}_o$, $\mathcal{A}_f \cap \mathcal{A}_o = \emptyset$, and \mathcal{A}_f is a connected space. The position of the AUV is denoted as $\mathbf{p} = (x, y) \in \mathcal{A}_f$. Define Γ as the set of all collision-free paths from the start position \mathbf{p}_{start} to the goal position \mathbf{p}_{goal} . For every feasible path $\gamma \in \Gamma$, its length is:

$$J(\gamma) = \int_{\gamma} ds \quad (1)$$

The objective is to find the path γ^* s.t. $J(\gamma^*) \leq J(\gamma), \forall \gamma \in \Gamma$.

III. SMART SAMPLING APPROACH

The basic idea is outlined in Figure 1 and is described in three phases: initialization, exploration, and exploitation. The critical regions for non-uniform sampling are found during the initialization phase. First, the known environment map is represented as a high-resolution uniform grid. Then, this map is converted into a coarse non-uniform grid by merging large groups of obstacle-free cells together. The boundaries of these merged cells form the *critical regions* for sampling. Next, in the exploration phase, a modified RRT* algorithm efficiently explores the space by only sampling these critical regions. To facilitate faster exploration, a region is considered *explored*

when it is first sampled, and only unexplored adjacent regions are considered in the next iteration. Connections between adjacent regions are guaranteed to be collision-free since the adjoining grid cell is a convex obstacle-free space. Finally, once a feasible solution is found, the (locally) optimal path is quickly obtained during the exploitation phase. From visibility graph theory, the corner points around obstacles are known to be a part of the optimal solution [30]. As such, the end points of the critical regions along the feasible path are identified. Then, a graph with the shortest paths to each of these points is created, and the (locally) optimal solution is obtained.

A. Initialization

The critical regions for non-uniform sampling are identified in the initialization phase. In order to find these regions, the search space \mathcal{A} is partitioned into $N \times N$ uniform grid \mathcal{C} with pairwise disjoint cell interiors, i.e., $\mathcal{C} = \{c(i, j) \subset \mathbb{R}^2 : i, j = 1, \dots, N\}$, such that $c^o(i, j) \cap c^o(k, l) = \emptyset, \forall i \neq k$ and $j \neq l$, and $\bigcup_{i,j} c(i, j) = \mathcal{A}$, where $c(i, j)$ is a cell located at position (i, j) of the grid and o denotes the interior of a cell. A cell $c(i, j)$ is denoted as an obstacle cell $c_o(i, j)$ if it is partially or fully occupied by any obstacle; otherwise, $c(i, j)$ is denoted as a free cell $c_f(i, j)$. Figure 1-a shows an example of partitioning an obstacle-rich scenario.

Once the obstacle and free cells are obtained from the grid partitioning, the free cells are expanded and merged together if they belong to the same convex obstacle-free group, as shown in Figure 1-b. The merging procedure is as follows. To create merged cell group C_m , where $m = 1, 2, \dots, M$ and M is the total number of groups (determined once all free cells are merged), select any free cell $c_f(i, j)$, that has not been merged. Check all of its immediate eight neighboring cells $c(i-1, j-1), c(i-1, j), c(i-1, j+1), \dots, c(i+1, j+1)$, and merge the cells into C_m that create the largest obstacle-free rectangular group of unmerged cells including $c_f(i, j)$. Keep expanding along the boundaries of cell group C_m in this manner until no more cells can be merged in any direction. Then, pick any unmerged free cell $c_f(i, j)$ and create the next cell group C_{m+1} in the same manner. Repeat until all free cells belong to a cell group.

Once the cell groups C_1, \dots, C_M are created, the critical regions for sampling are identified. Note that the cell groups: 1) have disjoint interiors, i.e., $C_m^o \cap C_n^o = \emptyset$, 2) include all free cells, i.e., $\bigcup_m C_m = \bigcup_{i,j} c_f(i, j)$, and 3) are convex since they are rectangular. The sampling regions are defined as the (obstacle-free) boundaries (i.e., lines) between all pairs of neighboring cell groups. We denote any such boundary line between neighboring cells C_m and C_n as r_{mn} . Note that r_{mn} and r_{nm} are identical. Figure 1-c shows the critical regions.

B. Exploration

During the exploration phase, the search tree is grown using a modified RRT* path planner until a feasible solution to the goal is found. The RRT* algorithm consists of six main functions: *sampling*, *distance*, *nearest neighbor*, *nearby vertices*, *collision check*, and *local steering*. In order to utilize the critical regions for smart sampling, only the *sampling* and

nearby vertices are updated. Additionally, we create one new function called *nearby regions*, which identifies all regions close to the search tree. All other functions are the same. For brevity's sake, we refer the reader to [19] for more details on RRT*. It should be noted that the *distance* function used in this paper is the Euclidean distance since the objective is to find the shortest path.

1) *Nearby Regions*: Given an existing search tree $T = (V, E)$, where V is the set of nodes denoting the samples and E is the set of connections between the samples, the set of nearby regions is:

$$\begin{aligned} \text{NearbyRegions}(T) = \{r_{mn} : \\ v \in r_{mj} \vee v \in r_{nj} \vee v \in C_m^o \vee v \in C_n^o, \\ \forall v \in V, v \notin r_{mn}, j \neq m, n\} \end{aligned} \quad (2)$$

Note that $v \in C_m^o$ or C_n^o if C_m^o or C_n^o contain the start or the goal. In other words, the nearby regions to search tree T are the regions that have not yet been sampled but share a cell group with an already sampled region. An example of the nearby regions for a given search tree are shown in Figure 1-d.

2) *Sampling*: Given the set $\text{NearbyRegions}(T)$, a nearby region is randomly selected for tree expansion according to a discrete uniform distribution, i.e., if there are N_{nr} nearby regions, the probability of selecting any region is $1/N_{nr}$. Then, a new node v_{new} is sampled in the center of this region. It is then connected to its nearest neighbor in search tree T in the same manner as RRT*. An example is shown in Figure 1-e. Note that this sampling paradigm limits the number of samples per region to one in order to facilitate faster exploration.

3) *Nearby Vertices*: Given a search tree $T = (V, E)$ and sample $\mathbf{p} = (x, y) \in \mathbb{R}^2$ lying on some region r_{mn} , the set of nearby vertices $\{v_{near}\}$ consists of vertices that either lie 1) on regions that adjoin either cell group C_m or C_n with some other cell group C_i , or 2) within the interiors of C_m^o or C_n^o :

$$\begin{aligned} \text{NearbyVertices}(T, \mathbf{p}) = \{v_{near} \in V : \\ v_{near} \in r_{mi} \vee v_{near} \in r_{ni} \vee \\ v_{near} \in C_m^o \vee v_{near} \in C_n^o, \\ \forall i = 1, \dots, M, i \neq n, m, \mathbf{p} \in r_{mn}\} \end{aligned} \quad (3)$$

If instead point \mathbf{p} lies inside the interior of C_m^o and not on any region r_{mj} , the nearby vertices are:

$$\begin{aligned} \text{NearbyVertices}(T, \mathbf{p}) = \{v_{near} \in V : \\ v_{near} \in C_m^o \vee v_{near} \in r_{mj}, \\ \forall j = 1, \dots, M\} \end{aligned} \quad (4)$$

Note that the only points \mathbf{p} that lie in the interior of the cell groups are \mathbf{p}_{start} and \mathbf{p}_{goal} . The function NearbyVertices is used to find the set of nearest neighbors for connecting a newly sampled node to the tree and performing a local rewiring of the tree as needed in the same manner as RRT*. An example is shown in Figure 1-f.

C. Exploitation

Once a feasible path is found, as shown in Figure 1-g, a search begins for the (locally) optimal solution within the homotopy class of this path. From visibility graph theory [30], the corner points of obstacles create the sufficient set of points that could belong to the shortest path. As such, new

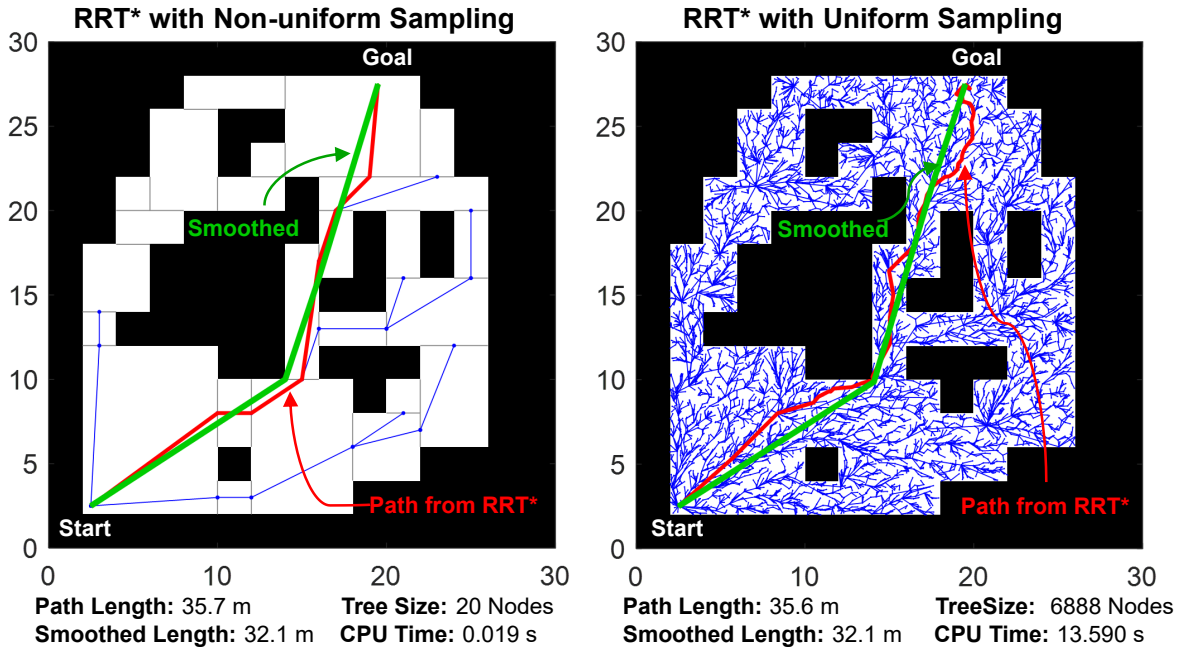


Fig. 2: Performance comparison of RRT* with the non-uniform sampler (left) and RRT* with the uniform sampler (right).

samples are made at the end points of each region r_{mn} that the feasible solution passes through. This set of samples contains at least all of the corner points along the feasible solution. Then, a new search tree is created using only these samples with the shortest collision-free connections made to the start. An example is shown in Figure 1-h. Note that in this step, collision-checking is necessary since connections between two samples can traverse several cell groups. Finally, the (locally) optimal path is obtained from this search tree by backtracking from the goal to the start, as shown in Figure 1-i.

IV. RESULTS AND DISCUSSION

In this section, the performance of RRT* with the non-uniform sampler is compared against that of the standard RRT* with a uniform sampler. The scenario considered is a $30m \times 30m$ map populated with several obstacles, as shown in Figure 2. For non-uniform sampling, the scenario is uniformly partitioned into cells of size $2m \times 2m$, resulting in a grid map of 15×15 cells. For the RRT* planner with the uniform sampler, the maximum connection distance between samples is set to be $5m$. The simulation was carried out in MATLAB using the RRT* path planner in the Navigation Toolbox on a Windows 10 machine with an Intel Core-i7 7700 CPU and 32GB of RAM.

Figure 2 shows the results generated by RRT* using both the non-uniform sampler (left) and the uniform sampler (right). Each plot shows: 1) the search tree in blue, 2) the best feasible path found from the RRT* path planner in red, and 3) the smoothed path in green. The benefits of the non-uniform sampler are evident since the identification of the critical regions has enabled the RRT* planner to quickly and efficiently cover the search space. The feasible path found goes directly to the goal through the narrow corridors between the obstacles in the center of the map, providing a smoothed path length of $32.1m$. Moreover, this path is found with only

20 samples in the search tree and a total computation time of 0.019 seconds. This low computation time is achieved since 1) the non-uniform sampler greatly reduces the size of the sampling space, and 2) no collision-checking is required during the exploration phase.

On the other hand, RRT* with the uniform sampler struggles in the obstacle-rich center of the map and thus needs a significantly longer computation time of 13.59 seconds (about 16,000 iterations) and a much larger tree size of 6888 nodes to find a path of the same solution quality of path length of $32.1m$ as compared to RRT* with the non-uniform sampler. Overall, the proposed non-uniform sampling approach shows significant promise over the uniform sampling, providing an intuitive way to identify these critical regions and efficiently find the shortest path.

V. CONCLUSIONS AND FUTURE WORK

In this paper, we developed a novel non-uniform sampler that intuitively identifies the critical regions for fast and efficient path planning. This is achieved by developing a new partitioning method that creates large convex cells in obstacle-free spaces and smaller convex cells in obstacle-dense regions. Then, we illustrate how to sample and search for a path along these regions in the RRT* framework. Specifically, non-uniform sampling speeds up the search significantly by dramatically reducing the size of the sampling space and systematically connecting samples in a way that does not require expensive collision checking during exploration. Compared to RRT* with a uniform sampler, this approach provides the same solution quality in significantly less computation time and memory footprint. Compared to other methods that identify critical regions using deep learning, this approach guarantees complete coverage of the configuration space without the necessity of uniform sampling.

Future work will include in-depth theoretical analysis of the non-uniform sampler with proofs for completeness and optimality. The partitioning scheme will be updated to search for the critical regions incrementally instead of requiring the entire map be partitioned a-priori. Informed subsets [31] will also be incorporated into this non-uniform sampling framework to provide an even faster convergence rate. Finally, this work will be extended to provide these nonuniform sampling benefits to multi-speed non-holonomic vehicles where both travel time and collision risk [32] are considered in the cost [33], [20].

REFERENCES

- [1] J. Song and S. Gupta, “ ϵ^* : An online coverage path planning algorithm,” *IEEE Transactions on Robotics*, vol. 34, no. 2, pp. 526–533, 2018.
- [2] J. Song and S. Gupta, “Care: Cooperative autonomy for resilience and efficiency of robot teams for complete coverage of unknown environments under robot failures,” *Autonomous Robots*, vol. 44, pp. 647–671, 2020.
- [3] A. A. F. Nassiraei, T. Sonoda, and K. Ishii, “Development of ship hull cleaning underwater robot,” in *Fifth International Conference on Emerging Trends in Engineering and Technology*, pp. 157–162, Nov 2012.
- [4] A. Bagnitsky, A. Inzartsev, A. Pavin, S. Melman, and M. Morozov, “Side scan sonar using for underwater cables & pipelines tracking by means of AUV,” in *IEEE Symposium on Underwater Technology and Workshop on Scientific Use of Submarine Cables and Related Technologies*, pp. 1–10, Apr 2011.
- [5] I. Robbins, G. Kirkpatrick, S. Blackwell, J. Hillier, C. Knight, and M. Moline, “Improved monitoring of HABs using autonomous underwater vehicles (AUV),” *Harmful Algae*, vol. 5, no. 6, pp. 749 – 761, 2006.
- [6] N. Palomeras, N. Hurtós, M. Carreras, and P. Ridao, “Autonomous mapping of underwater 3-D structures: From view planning to execution,” *IEEE Robotics and Automation Letters (RA-L)*, vol. 3, no. 3, pp. 1965–1971, 2018.
- [7] Z. Shen, J. Song, K. Mittal, and S. Gupta, “An autonomous integrated system for 3-D underwater terrain map reconstruction,” in *OCEANS 2016 - Monterey*, pp. 1–6, 2016.
- [8] Z. Shen, J. Song, K. Mittal, and S. Gupta, “Autonomous 3-D mapping and safe-path planning for underwater terrain reconstruction using multi-level coverage trees,” in *OCEANS 2017 - Anchorage*, pp. 1–6, 2017.
- [9] Z. Shen, J. Song, K. Mittal, and S. Gupta, “CT-CPP: 3D coverage path planning for unknown terrain reconstruction using coverage trees,” *arXiv preprint arXiv:2010.09231*, 2020.
- [10] Z. Shen, J. P. Wilson, and S. Gupta, “An online coverage path planning algorithm for curvature-constrained AUVs,” in *OCEANS 2019 - SEATTLE*, pp. 1–5, 2019.
- [11] Z. Shen, J. P. Wilson, and S. Gupta, “ ϵ^*+ : An online coverage path planning algorithm for energy-constrained autonomous vehicles,” in *Global OCEANS 2020: Singapore – U.S. Gulf Coast*, pp. 1–6, 2020.
- [12] J. Song, S. Gupta, J. Hare, and S. Zhou, “Adaptive cleaning of oil spills by autonomous vehicles under partial information,” in *OCEANS 2013 - San Diego*, pp. 1–5, 2013.
- [13] S. Sariel, T. Balch, and N. Erdogan, “Naval mine countermeasure missions,” *IEEE Robotics Automation Magazine*, vol. 15, pp. 45–52, Mar 2008.
- [14] K. Mukherjee, S. Gupta, A. Ray, and S. Phoha, “Symbolic analysis of sonar data for underwater target detection,” *IEEE Journal of Oceanic Engineering*, vol. 36, no. 2, pp. 219–230, 2011.
- [15] B. Garau, M. Bonet, A. Alvarez, S. Ruiz, and A. Pascual, “Path planning for autonomous underwater vehicles in realistic oceanic current fields: Application to gliders in the Western Mediterranean Sea,” *Journal of Maritime Research*, vol. 6, no. 2, pp. 5–21, 2009.
- [16] K. Mittal, J. Song, S. Gupta, and T. A. Wettergren, “Rapid path planning for Dubins vehicles under environmental currents,” *Robotics and Autonomous Systems*, vol. 134, p. 103646, 2020.
- [17] Z. Zeng, L. Lian, K. Sammut, F. He, Y. Tang, and A. Lammas, “A survey on path planning for persistent autonomy of autonomous underwater vehicles,” *Ocean Engineering*, vol. 110, pp. 303–313, 2015.
- [18] S. M. LaValle, *Planning algorithms*. Cambridge university press, 2006.
- [19] S. Karaman, M. R. Walter, A. Perez, E. Frazzoli, and S. Teller, “Anytime motion planning using the RRT*,” in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1478–1483, 2011.
- [20] J. P. Wilson, Z. Shen, S. Gupta, and T. A. Wettergren, “T*-Lite: A fast time-risk optimal motion planning algorithm for multi-speed autonomous vehicles,” in *Global OCEANS 2020: Singapore – U.S. Gulf Coast*, pp. 1–6, 2020.
- [21] Z. Sun, D. Hsu, T. Jiang, H. Kurniawati, and J. Reif, “Narrow passage sampling for probabilistic roadmap planning,” *IEEE Transactions on Robotics*, vol. 21, no. 6, pp. 1105–1115, 2005.
- [22] B. Ichter, E. Schmerling, T.-W. E. Lee, and A. Faust, “Learned critical probabilistic roadmaps for robotic motion planning,” in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9535–9541, 2020.
- [23] D. Molina, K. Kumar, and S. Srivastava, “Learn and Link: Learning critical regions for efficient planning,” in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 10605–10611, 2020.
- [24] J. Wang, W. Chi, C. Li, C. Wang, and M. Q.-H. Meng, “Neural RRT*: Learning-based optimal path planning,” *IEEE Transactions on Automation Science and Engineering*, vol. 17, no. 4, pp. 1748–1758, 2020.
- [25] A. H. Qureshi, Y. Miao, A. Simeonov, and M. C. Yip, “Motion Planning Networks: Bridging the gap between learning-based and classical motion planners,” *IEEE Transactions on Robotics*, vol. 37, no. 1, pp. 48–66, 2021.
- [26] R. Kumar, A. Mandalika, S. Choudhury, and S. Srinivasa, “LEGO: Leveraging experience in roadmap generation for sampling-based planning,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1488–1495, 2019.
- [27] N. Goldenfeld, *Lectures on phase transitions and the renormalization group*. CRC Press, 2018.
- [28] K. G. Wilson, “Renormalization group and critical phenomena. I. Renormalization group and the Kadanoff scaling picture,” *Physical review B*, vol. 4, no. 9, p. 3174, 1971.
- [29] K. Mittal and S. Gupta, “Minimum-time motion-planning of AUVs under spatially varying ocean currents,” in *OCEANS 2019 - SEATTLE*, pp. 1–5, 2019.
- [30] M. M. Zafar, M. L. Anjum, and W. Hussain, “LTA*: Local tangent based A* for optimal path planning,” *Autonomous Robots*, vol. 45, no. 2, pp. 209–227, 2021.
- [31] J. D. Gammell, T. D. Barfoot, and S. S. Srinivasa, “Informed sampling for asymptotically optimal path planning,” *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 966–984, 2018.
- [32] J. Song, S. Gupta, and T. A. Wettergren, “T*: Time-optimal risk-aware motion planning for curvature-constrained vehicles,” *IEEE Robotics and Automation Letters*, vol. 4, no. 1, pp. 33–40, 2019.
- [33] J. P. Wilson, K. Mittal, and S. Gupta, “Novel motion models for time-optimal risk-aware motion planning for variable-speed AUVs,” in *OCEANS 2019 - SEATTLE*, pp. 1–5, 2019.