In *Proceedings of the 2020 IEEE International Symposium on Safety, Security, and Rescue Robotics* (SSRR 2016), Abu Dhabi, November 2020

Benchmarking Metric Ground Navigation

Daniel Perille[1], Abigail Truong[1], Xuesu Xiao[1], and Peter Stone[1]

***Abstract*— Metric ground navigation addresses the problem of autonomously moving a robot from one point to another in an obstacle-occupied planar environment in a collision-free manner. It is one of the most fundamental capabilities of intelligent mobile robots. This paper presents a standardized testbed with a set of environments and metrics to benchmark difficulty of different scenarios and performance of different systems of metric ground navigation. Current benchmarks focus on individual components of mobile robot navigation, such as perception and state estimation, but the navigation performance as a whole is rarely measured in a systematic and standardized fashion. As a result, navigation systems are usually tested and compared in an ad hoc manner, such as in one or two manually chosen environments. The introduced benchmark provides a general testbed for ground robot navigation in a metric world. The Benchmark for Autonomous Robot Navigation (BARN) dataset includes 300 navigation environments, which are ordered by a set of difficulty metrics. Navigation performance can be tested and compared in those environments in a systematic and objective fashion. This benchmark can be used to predict navigation difficulty of a new environment, compare navigation systems, and potentially serve as a cost function and a curriculum for planning-based and learning-based navigation systems. We have published our dataset and the source code to generate datasets for different robot footprints at `www.cs.utexas.edu/~xiao/BARN/BARN.html`.**

I. INTRODUCTION

Autonomously moving from one point to another, especially in challenging environments, is one essential capability of intelligent mobile robots. This problem of mobile robot navigation has been studied by the robotics community for decades [1]–[3]. Sophisticated navigation systems have been developed using classical control methods [3]–[5], path and motion planning [1], [2], [6], or, more recently, machine learning techniques [7]–[11].

However, despite the plethora of works in mobile robot navigation, there is no generally accepted metric by which to compare different approaches against one another, even for navigation in a simple metric world, where only geometric obstacles are considered. Although in relatively open space, navigation performance of different systems may not vary significantly, the lack of an accepted metric becomes particularly relevant in environments that are more difficult to navigate. Those environments include, for example, unstructured or confined spaces for search and rescue [12] and highly constrained spaces where agile maneuvers are required for robots with nonholonomic motion constraints [10]. Newly developed navigation systems are only tested and compared to existing ones in a limited number of ad hoc environments with unquantified difficulties.

To address the lack of a standardized method to test and compare mobile robot navigation systems, this work provides a Benchmark for Autonomous Robot Navigation (BARN) of 300 simulated test environments, which can also be easily instantiated in the physical world. We design a set of metrics to quantify navigation difficulty of these simulated environments for ground mobile robots to move in obstacle-occupied spaces without collision. We then unify this set of metrics via a learned function approximator to introduce a novel measure of an environment's difficulty level for metric ground navigation. Through an extensive amount of 3000 simulated trials using two widely used planners, Dynamic Window Approach (DWA) [2] and Elastic Bands (E-Band) [1], we benchmark the relative difficulty levels for the test environments. We also run multiple physical navigation trials to validate our model's predictive power of navigation difficulty. To summarize, the main contributions of this work are:

- A benchmark dataset (BARN) of 300 simulated test environments for metric ground navigation,
- A set of metrics and a data-driven model to combine them to quantify the challenge posed by a particular environment for mobile robot navigation.

The rest of the paper is organized as follows: Section II reviews existing benchmarks related to mobile robot navigation. Section III describes our method of constructing BARN, including the test environments, the set of difficulty metrics, and the data-driven model to combine them. Section IV provides implementation details and experiment results to validate that the proposed difficulty metrics and the learned function approximator can predict navigation difficulties using a physical mobile robot in the real world. Section V concludes the paper.

II. RELATED WORK

This section reviews existing testbeds and metrics related to mobile robot navigation.

A. Testbeds

Testbeds are designed as an apparatus to quantify performance based on a set of pre-defined metrics. The tests can be replicated when following a standardized testing procedure.

1) Physical Testbeds: National Institute of Standards and Technology (NIST) has created standard testbeds for response robots, including ground, aerial, and aquatic vehicles [13]. Specialized test methods are developed to test

[1]Daniel Perille, Abigail Truong, Xuesu Xiao, and Peter Stone are with Department of Computer Science, University of Texas at Austin, Austin, TX 78712 {danny.perille, a.truong}@utexas.edu, {xiao, pstone}@cs.utexas.edu

individual robot capacity, e.g. locomotion, sensing, communication, durability. Robotarium [14] is a testbed developed to test algorithms for multi-robot research, using a fleet of miniature differential drive robots. Xiao et al [15] reviewed 20 physical testbeds for snake robots and pointed out that all of them are designed in an ad hoc manner, i.e. tailored to demonstrate the newly developed capability. They also provided suggestions on a general testbed design. The research thrust on developing robot testbeds demonstrates the robotics community's need for standardized test methods to quantify robot performance and research progress. Similar to the aforementioned testbeds but with a different purpose, the proposed testbed is developed to benchmark mobile robot navigation systems operating in a metric world.

2) Software Testbeds: Thanks to the recent progress on data-driven approaches, testbeds are frequently instantiated as datasets, e.g. ImageNet [16]. In the mobile robot navigation domain, especially on the perception and estimation side, many such software testbeds have also been created [17]–[20], where perceptual data is collected along a fixed motion trajectory. However, when arbitrary motion execution is required, such interactive testbeds become sparse. Even when motion is allowed [21], [22], the locomotion part of navigation is assumed to be trivial, i.e. the testbeds only benchmark the robot's ability to infer "where" to navigate, instead of to generate feasible and optimal motion commands for "how" to navigate. The proposed testbed focuses on the ability to autonomously generate viable motion commands in order to navigate between two fixed points in a given environment. Since only geometric obstacles are considered, it can easily be instantiated into a physical testbed.

B. Metrics

A common metric to quantify mobile robot navigation difficulty is distance from points on the path to the closest obstacle [23], [24], as the closer the robot needs to come to an obstacle, the more difficult the navigation task. Past experiences (e.g. previous failure cases) have also been utilized to quantify difficulty as a function of a single state when navigating in the ocean [25] or in city traffic [26]. Not many works considered more than one single source of difficulty: Soltani et al. [27] represented difficulty with both distance to closest obstacle and visibility of a particular location and combined their effects using manually defined weights. Robot motion risk can also be viewed as an indication of difficulty: recent risk reasoning frameworks extended the dependency of risk associated with a certain state into motion history [12], [28], and pointed out that difficulty/risk caused by, for example, turning or dragging a tether, cannot be determined by a single state alone. The multiple difficulty metrics designed in this work are inspired by the risk universe [12]. In order to determine the combined effect of all individual elements, we use a data-driven approach instead of manually assigned weights.

III. APPROACH

In this section, we describe our method of constructing BARN. The navigation environments are first generated using cellular automaton [29], for which navigational paths are planned on the robot Configuration Space (C-Space) [30]. Second, we introduce a set of metrics used to quantify navigation difficulty level. Third, a function approximator is learned in a data-driven manner to combine these difficulty metrics and determine the final difficulty level of navigating through a specific environment.

A. Navigation Environments

Navigation environments are systematically generated through the method of cellular automaton [29]. A cellular automaton was originally designed as a collection of black cells on a white grid of specified shape that evolves through a number of discrete time steps according to a set of rules based on the states of neighboring cells [29]. In this work, we use black cells to represent obstacle-occupied space and white cells to represent free space. The evolution of the black cells generates different obstacle configurations. Cellular automaton is easy to scale to any size, generates more realistic environments than random fill, and is also easily customizable due to its parameters that can be changed to generate different types of worlds. Due to the smoothing iterations, the resulting grid resembles real-world obstacles more than the initial randomly filled grid does. We use four parameters of the cellular automaton to control the generation of obstacles: *initial fill percentage*, *smoothing iterations*, *fill threshold*, and *clear threshold*. The procedure to generate navigation environments using cellular automaton is provided in Algorithm 1, with an example in Figure 1.

Algorithm 1 Navigation Environments Generation

```
1: Inputs: $m$, $n$, initial fill percentage, smoothing iterations, fill threshold, clear threshold,
2: Randomly fill a $m \times n$ grid of 0's with initial fill percentage of 1's
3: for iteration $k = 1$ : smoothing iterations do
4:     for cell in grid do
5:         if $|$FilledNeighbors(cell)$| \geq$ fill threshold then
6:             cell $\leftarrow 1$                      ▷ Fill cell
7:         end if
8:         if $|$FilledNeighbors(cell)$| \leq$ clear threshold then
9:             cell $\leftarrow 0$                      ▷ Empty cell
10:        end if
11:    end for
12: end for
```

Each obstacle grid is then converted into the robot's C-space based on the robot dimension. In the C-space, one free point on both the left and right edge are chosen at random to be the start and end points of the path, respectively. A flood-fill algorithm [31] is used to determine if there is an open path between the points. If no path is possible, then the space is discarded. A* algorithm [32] is then used to plan a path in the free C-space.

B. Difficulty Metrics

Upon generating an environment through cellular automaton and establishing a path therein, various metrics are calculated in the C-space along the path to quantify the difficulty of traversal.

1) Distance to Closest Obstacle: At each cell in the environment, the Distance to Closest Obstacle is defined as the distance from this cell to the nearest occupied space. This metric is averaged over all points in the path.

2) Average Visibility: A cell's Average Visibility is defined as the average of the distances to an obstacle along each ray in a 360° scan. In our discrete space, we average the visibility along eight rays (four cardinal directions and four diagonals), then average this metric over all points in the path. Figure 2 provides examples of high and low visibility.

3) Dispersion: From a given state in the environment, a 360° scan is cast up to a certain max length. The dispersion at that state is defined as the number of alternations in that scan from occupied to unoccupied space or vice versa, as shown in Figure 3.

Dispersion captures the number of potential paths out of a given location. A higher dispersion means more possible options for the navigation algorithm and therefore means the environment poses more challenges, especially for a sampling-based local planner, like DWA [2]. In our discrete space, dispersion is calculated by casting 16 rays up to a max length, and checking which are blocked or open. This metric is calculated for each point in the path and averaged over the length of the path.

4) Characteristic Dimension: The Characteristic Dimension at a given cell is defined as the visibility of the axis through the cell with the lowest visibility. In our discrete space, 8 axes (each made up of 2 rays 180° apart) are cast 22.5° apart from one another. Each axis' visibility is calculated as the sum of the distances to an obstacle along both of the rays that make it up. The Characteristic Dimension is defined as the visibility of the axis with the lowest visibility. The Characteristic Dimension captures the tightness of a space. A low Distance to Closest Obstacle could occur in a relatively open space and therefore still be quite easy to navigate. Additionally, a space might be tight along one axis yet very open along another (e.g. a long tunnel) and therefore have a high Average Visibility despite being narrow. Figure 4 captures such an instance of when Distance to Closest Obstacle and Average Visibility may fail to completely represent the difficulty of a space.

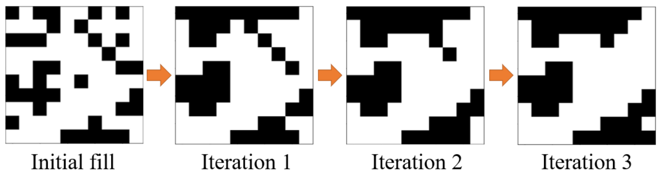

Fig. 1. Three *smoothing iterations* of a cellular automaton with an *initial fill percentage* of 0.35, *fill threshold* of 5, and *clear threshold* of 1

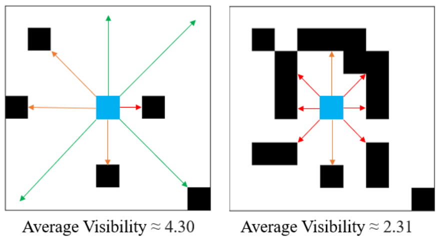

Fig. 2. An example of high and low Average Visibility

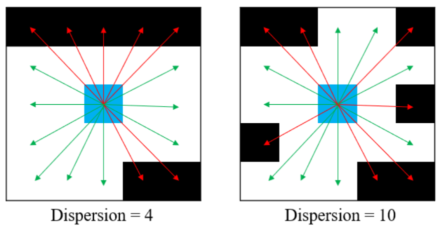

Fig. 3. Dispersion represents the alternations from occupied to unoccupied space or vice versa.

5) Tortuosity: Tortuosity, as a property of a curve being tortuous, is calculated over the entire path using the arc-chord ratio. The arc length is the length of the entire path, while the chord length is the length of a straight line between the start and end points. This metric captures bends in the path that make navigation more difficult compared to navigation along a relatively straight path.

C. Combined Difficulty Level

We first use a data-driven approach to benchmark the relative difficulty level of all the navigation environments in our dataset. Thousands of simulation trials using representative and widely-used navigation systems are conducted to reveal the actual difficulty of an environment. Second, to investigate how the difficulty metrics interact with each other and contribute to the combined difficulty measure, we learn a function approximator to map from the individual difficulty metrics to the final difficulty level of a given environment. This model can be used to predict difficulty of unseen navigation environments. Please refer to Section IV for details.

IV. EXPERIMENTS

In this section, we present implementation details of our dataset generation and physical experiments using a ground robot to validate that our benchmark model can accurately predict the difficulty level of unseen physical navigation environments.

A. Dataset Generation

We use 12 sets of cellular automaton parameters to generate the 300 navigation environments of 30 by 30 cells in our dataset, shown in Table I. The parameters are chosen to generate varied worlds with reasonably realistic configurations. Each of those parameter sets is repeated 25 times. We use a Clearpath Jackal robot's dimension (0.508m by 0.430m, corresponding to 5 by 5 cells) to inflate the obstacles and

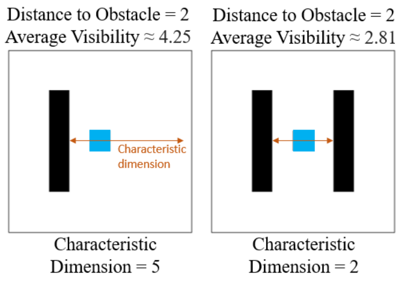
Distance to Obstacle = 2
Average Visibility ≈ 4.25
Distance to Obstacle = 2
Average Visibility ≈ 2.81
Characteristic dimension
Characteristic Dimension = 5
Characteristic Dimension = 2

Fig. 4. Characteristic Dimension captures the tightness of a state in an environment.

generate the C-spaces. We also provide the original obstacle map in the dataset so that C-spaces corresponding to different robot sizes can be generated by the users.

TABLE I
CELLULAR AUTOMATON PARAMETERS

Parameters	Values
initial fill percentage	{0.15, 0.20, 0.25, 0.30}
smoothing iterations	{2, 3, 4}
fill threshold	5
clear threshold	1
neighborhood	8
Repetitions	25

The minimum, maximum, mean, and standard deviation of the five difficulty metrics present in the dataset are shown in Table II. For a given cell, the minimum Distance to Closest Obstacle is 1 for an unoccupied space, and the maximum is 10.20. We cast eight rays to compute Average Visibility, resulting in a range from 1 to 14. The minimum dispersion is 0 (no alternations between occupied and unoccupied spaces along the 16 axes) and the maximum is 12. The minimum Characteristic Dimension is 0 (completely closed space) and the maximum is 20.00. The four metrics above are computed for and averaged over all states on a path. The minimum tortuosity of a path is 1 (straight line) and the maximum is 1.71.

TABLE II
DIFFICULTY METRIC VALUES

	Min.	Max.	Mean	Std.
Distance to Closest Obstacle	1	10.20	2.37	0.93
Average Visibility	1	14.00	4.42	1.64
Dispersion	0	12	4.35	0.89
Characteristic Dimension	0	20.00	4.05	2.66
Tortuosity	1	1.71	1.21	0.14

1) Simulation Trials: We use a simulated Clearpath Jackal, a four-wheeled, differential drive, nonholonomic ground robot, in a Robot Operating System (ROS) Gazebo simulator [33] to benchmark the relative difficulty levels of the navigation environments in our dataset.

We choose two different widely used navigation planners, DWA [2] and E-Band [1] to navigate Jackal. DWA is a representative sampling-based motion planner. Given a global path produced by the A* algorithm [32], DWA generates samples of linear and angular velocities and evaluates the score of each sample based on closeness to the obstacle, to the global path, and progress toward the local goal. The action sample with the best score is executed to move the robot. The randomness in the sampling process leads to non-deterministic behavior in the same environment. E-Band is a representative optimization-based motion planner, which optimizes an initial trajectory. It deforms the trajectory using virtual bubbles along it, which are subject to repulsive force from the obstacles. The optimized trajectory acts like an elastic band. The default navigation planner from the robot manufacturer, Clearpath Robotics, is the DWA planner. We use the default planner parameters from the manufacturer[1] for DWA, and the default parameters from the E-Band designer[2] for the E-Band planner. We set E-Band's maximum allowable linear and angular velocities (0.5m/s and 1.57rad/s) to match with those of DWA for a fair comparison.

For each one of the 300 environments in our dataset, a pre-built map is provided to the planner, and we run five trials each for DWA and E-Band, resulting in a total number of 3000 trials (Figure 5). The final difficulty measure is the traversal time averaged over the ten trials and normalized by path length. We also compute the variance of the traversal time. A 30-second penalty is introduced to trials where the robot fails to reach the goal, e.g. getting stuck. The DWA, E-Band, and combined (averaged) with predicted navigation performance is shown in Figure 6. From left to right, the 300 environments are ordered from easy to difficult. High difficulty level is correlated with high variance. Operating with a map, DWA and E-Band achieve an average normalized traversal time of 3.30 ± 0.50s/m and 3.24 ± 0.38s/m, respectively. As a sampling-based planner, DWA results in higher standard deviation than the optimization-based E-Band, and is more sensitive to the increased difficulty level.

2) Function Approximator: To approximate the combined effect of all five difficulty metrics, we use a simple neural network consisting of two fully connected layers with 64 neurons each. Distance to Closest Obstacle, Average Visibility, Dispersion, and Characteristic Dimension are computed for and averaged over all the states on a path. Tortuosity is computed for the entire path. All these metrics are normalized based on their mean and standard deviation (Table II). The label of the neural network output is the average traversal time normalized by path length, computed from the 3000 simulation trials. Our function approximator can achieve a 0.10 prediction loss on normalized traversal time, which corresponds to, for example, 0.50 seconds error while traversing a 5m long path.

B. Physical Experiments

To validate our benchmarks in the real world, we also conduct physical trials in unseen navigation environments. We use a physical Jackal with DWA and E-Band planners

[1]https://github.com/jackal/jackal/tree/melodic-devel/jackal_navigation/params
[2]http://wiki.ros.org/eband_local_planner

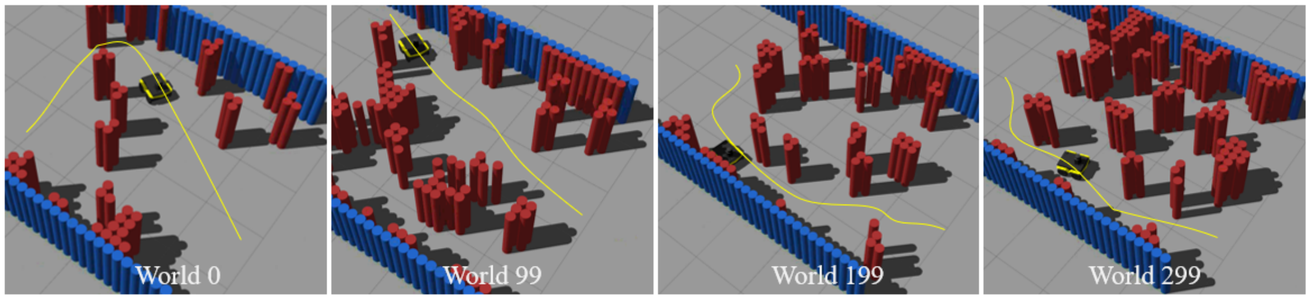

Fig. 5. Four Example Environments in Gazebo Simulation (ordered by ascending relative difficulty level)

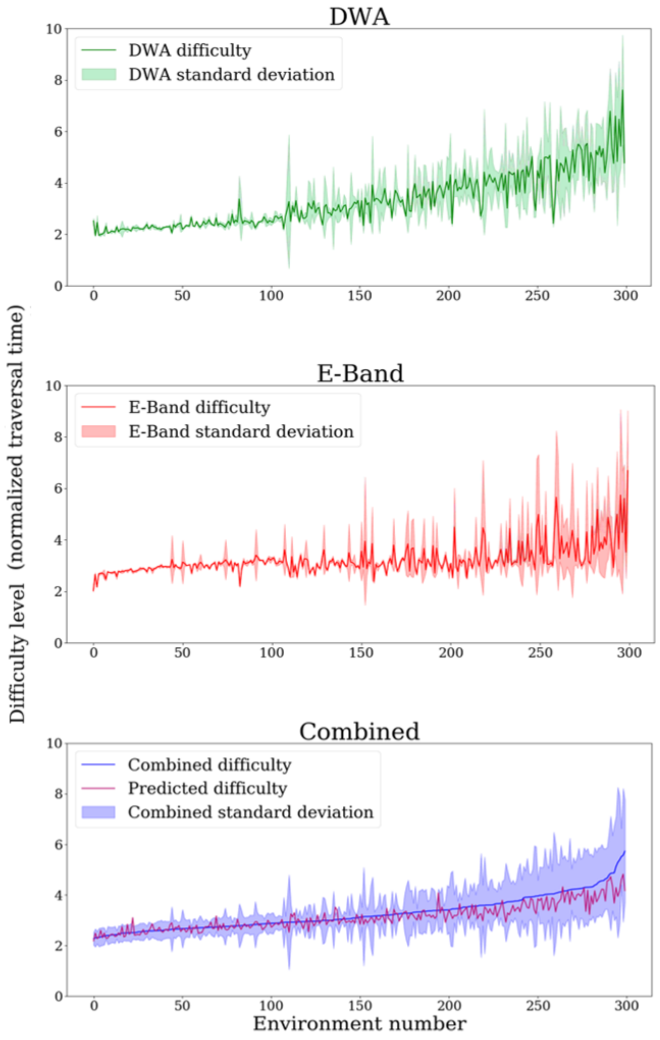

Fig. 6. Environment Difficulty Benchmarked by Navigation Performance of DWA and E-Band

parameterized in the same way as in the simulation trials. Five new navigation environments are created using cellular automaton and instantiated in the real-world with cardboard boxes representing obstacles (Figure 7). We run five trials with each planner in each environment, resulting in a total of 50 physical trials. Unlike the simulated trials, the planners do not have access to a pre-built map. As shown in Figure 8, higher predicted difficulty corresponds to longer normalized traversal time (The fitted blue line has a slope of 0.96 and almost zero intercept). In physical environments with low difficulty, DWA performs better than E-Band without a pre-built map. However, the steeper slope of the green line (1.17) than that of the red line (0.74) indicates that DWA is more sensitive to increased difficulty level than E-Band is, which is a similar trend we observe in simulation.

V. CONCLUSIONS

We present a dataset[3] of 300 simulated navigation environments and five difficulty metrics along with a data-driven model to quantify the difficulty measure of a particular environment for mobile robot navigation. We benchmark the relative difficulty level using 3000 simulated navigation trials with two widely used navigation planners, DWA and E-Band, which are representative of sampling-based and optimization-based planners, respectively. Our model can predict the difficulty of unseen navigation environments based on the five difficulty metrics, i.e. Distance to Closest Obstacle, Average Visibility, Dispersion, Characteristic Dimension, and Tortuosity. 50 physical experiment trials demonstrate that the difficulty level predicted by our model corresponds to real-world performance in unseen environments. As a general testbed, metric ground navigation performance of different systems can be tested and compared with each other in a systematic and objective fashion. The difficulty metrics and the learned function approximator can be used as a new cost function and a curriculum for planning-based and learning-based navigation systems.

ACKNOWLEDGMENT

This work has taken place in the Learning Agents Research Group (LARG) at UT Austin. LARG research is supported in part by NSF (CPS-1739964, IIS-1724157, NRI-1925082), ONR (N00014-18-2243), FLI (RFP2-000), ARO (W911NF-19-2-0333), DARPA, Lockheed Martin, GM, and Bosch. Peter Stone serves as the Executive Director of Sony AI America and receives financial compensation for this work. The terms of this arrangement have been reviewed and approved by the University of Texas at Austin in accordance with its policy on objectivity in research.

REFERENCES

[1] S. Quinlan and O. Khatib, "Elastic bands: Connecting path planning and control," in *[1993] Proceedings IEEE International Conference on Robotics and Automation.* IEEE, 1993, pp. 802–807.

[3]`www.cs.utexas.edu/~attruong/metrics_dataset.html`

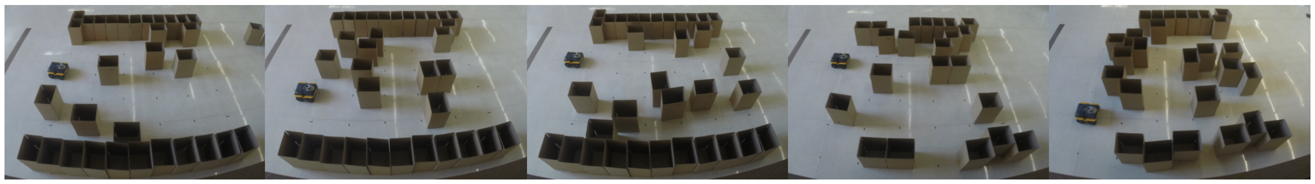

Fig. 7. For each of the five physical navigation environments, five DWA and five E-Band trials are conducted.

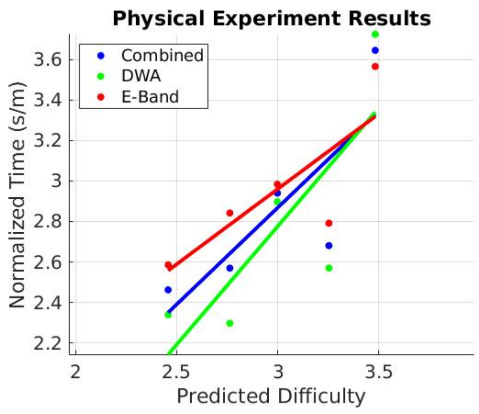

Fig. 8. Real-World Navigation Performance vs. Predicted Difficulty

[2] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robotics & Automation Magazine*, vol. 4, no. 1, pp. 23–33, 1997.

[3] R. M. Knotts, I. R. Nourbakhsh, and R. C. Morris, "Navigates: A benchmark for indoor navigation," in *Robotics 98*, 1998, pp. 36–42.

[4] X. Xiao, J. Dufek, T. Woodbury, and R. Murphy, "Uav assisted usv visual navigation for marine mass casualty incident response," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 6105–6110.

[5] X. Xiao, E. Cappo, W. Zhen, J. Dai, K. Sun, C. Gong, M. J. Travers, and H. Choset, "Locomotive reduction for snake robots," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 3735–3740.

[6] X. Xiao, J. Dufek, M. Suhail, and R. Murphy, "Motion planning for a uav with a straight or kinked tether," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 8486–8492.

[7] A. Giusti, J. Guzzi, D. C. Cireşan, F.-L. He, J. P. Rodríguez, F. Fontana, M. Faessler, C. Forster, J. Schmidhuber, G. Di Caro *et al.*, "A machine learning approach to visual perception of forest trails for mobile robots," *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 661–667, 2015.

[8] M. Pfeiffer, M. Schaeuble, J. Nieto, R. Siegwart, and C. Cadena, "From perception to decision: A data-driven approach to end-to-end motion planning for autonomous ground robots," in *2017 ieee international conference on robotics and automation (icra)*. IEEE, 2017, pp. 1527–1533.

[9] X. Xiao, B. Liu, G. Warnell, J. Fink, and P. Stone, "Appld: Adaptive planner parameter learning from demonstration," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 4541–4547, 2020.

[10] X. Xiao, B. Liu, G. Warnell, and P. Stone, "Toward agile maneuvers in highly constrained spaces: Learning from hallucination," *arXiv preprint arXiv:2007.14479*, 2020.

[11] B. Liu, X. Xiao, and P. Stone, "Lifelong navigation," *arXiv preprint arXiv:2007.14486*, 2020.

[12] X. Xiao, "Risk-aware path and motion planning for a tethered aerial visual assistant in unstructured or confined environments," *arXiv preprint arXiv:2007.09595*, 2020.

[13] NIST. Standard test methods for response robots. [Online]. Available: https://www.nist.gov/el/intelligent-systems-division-73500/standard-test-methods-response-robots

[14] D. Pickem, P. Glotfelter, L. Wang, M. Mote, A. Ames, E. Feron, and M. Egerstedt, "The robotarium: A remotely accessible swarm robotics research testbed," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 1699–1706.

[15] X. Xiao and R. Murphy, "A review on snake robot testbeds in granular and restricted maneuverability spaces," *Robotics and Autonomous Systems*, vol. 110, pp. 160–172, 2018.

[16] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[17] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of rgb-d slam systems," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 573–580.

[18] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.

[19] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, "1 year, 1000 km: The oxford robotcar dataset," *The International Journal of Robotics Research*, vol. 36, no. 1, pp. 3–15, 2017.

[20] W. Wang, D. Zhu, X. Wang, Y. Hu, Y. Qiu, C. Wang, Y. Hu, A. Kapoor, and S. Scherer, "Tartanair: A dataset to push the limits of visual slam," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020.

[21] E. Kolve, R. Mottaghi, W. Han, E. VanderBilt, L. Weihs, A. Herrasti, D. Gordon, Y. Zhu, A. Gupta, and A. Farhadi, "Ai2-thor: An interactive 3d environment for visual ai," *arXiv preprint arXiv:1712.05474*, 2017.

[22] M. Savva, A. Kadian, O. Maksymets, Y. Zhao, E. Wijmans, B. Jain, J. Straub, J. Liu, V. Koltun, J. Malik *et al.*, "Habitat: A platform for embodied ai research," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 9339–9347.

[23] L. De Filippis, G. Guglieri, and F. Quagliotti, "A minimum risk approach for path planning of uavs," *Journal of Intelligent & Robotic Systems*, vol. 61, no. 1-4, pp. 203–219, 2011.

[24] S. Feyzabadi and S. Carpin, "Risk-aware path planning using hirerachical constrained markov decision processes," in *Automation Science and Engineering (CASE), 2014 IEEE International Conference on*. IEEE, 2014, pp. 297–303.

[25] A. A. Pereira, J. Binney, G. A. Hollinger, and G. S. Sukhatme, "Risk-aware path planning for autonomous underwater vehicles using predictive ocean models," *Journal of Field Robotics*, vol. 30, no. 5, pp. 741–762, 2013.

[26] J. Krumm and E. Horvitz, "Risk-aware planning: Methods and case study for safer driving routes." in *AAAI*, 2017, pp. 4708–4714.

[27] A. Soltani and T. Fernando, "A fuzzy based multi-objective path planning of construction sites," *Automation in construction*, vol. 13, no. 6, pp. 717–734, 2004.

[28] X. Xiao, J. Dufek, and R. R. Murphy, "Robot risk-awareness by formal risk reasoning and planning," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2856–2863, 2020.

[29] S. Wolfram, "Statistical mechanics of cellular automata," *Reviews of modern physics*, vol. 55, no. 3, p. 601, 1983.

[30] J.-C. Latombe, *Robot motion planning*. Springer Science & Business Media, 2012, vol. 124.

[31] S. Torbert, *Applied computer science*. Springer, 2016.

[32] P. Hart, N. 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. [Online]. Available: https://doi.org/10.1109/tssc.1968.300136

[33] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*, vol. 3. IEEE, 2004, pp. 2149–2154.