Visually Adaptive Geometric Navigation

Shravan Ravi[1*], Shreyas Satewar[1*], Gary Wang[1*], Xuesu Xiao[1],
Garrett Warnell[1,2], Joydeep Biswas[1], and Peter Stone[1,3]

***Abstract*— While classical navigation systems can move robots from one point to another in a geometrically collision-free manner, recent approaches to visual navigation allow robots to reason beyond geometry and consider semantic information. However, the navigation behaviors generated by visual navigation often lack provable safety guarantees. This paper presents Visually Adaptive Geometric Navigation (VAGN), which marries the two schools of navigation approaches to produce a navigation system that is able to adapt to the visual appearance of the environment while maintaining collision-free behavior. Employing a classical geometric navigation system to address geometric safety and efficiency, VAGN consults visual perception to dynamically adjust the classical planner's hyper-parameters (e.g., maximum speed, inflation radius) to enable navigational behaviors not possible with pure geometry. VAGN is implemented on a physical ground robot in a test course with rich semantic and geometric features and demonstrates superior navigation performance compared to other navigation baselines using visual and/or geometric input.**

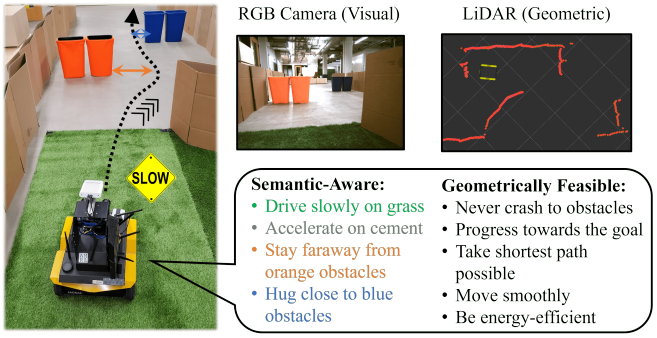

Fig. 1. Visually Adaptive Geometric Navigation (VAGN) enables semantic-aware and geometrically feasible navigation in real-world scenarios.

I. INTRODUCTION

Navigating mobile robots from one point to another in an appropriate manner has been a research topic for the robotics community for decades. Autonomous mobile robots take in perceptual input from onboard sensors (e.g., LiDAR, camera) and compute motion commands for the actuators (e.g., wheels, tracks) to move in their workspace.

Decades of research have been devoted to geometric navigation [1], [2], in which robots perceive their surroundings as *free* or *occupied* (and, sometimes, *unknown*) tessellations of the workspace and seek to find geometrically appropriate paths that are, for example, collision-free, shortest, fastest, energy efficient, or a combination thereof. Due to their reliability at these basic tasks, these geometric navigation systems have been successfully deployed autonomously in real-world settings over extended periods of time [3], [4].

Thanks to recent successes in computer vision driven by machine learning, there has been a recent surge of interest in visual navigation from within the robotics community [5]–[7]. Using visual input for navigation is attractive for many reasons. For example, RGB cameras are less expensive and therefore more commonly available on mobile robot platforms than LiDARs. Furthermore, it provides more information than pure geometry, e.g., semantics (Fig. 1). However, visual navigation systems often lack a few valuable features of their geometric counterpart: the lack of a collision-free guarantee limits these systems to be deployed only as a "proof-of-concept" in simulation or only in small-scale real-world experiments [8]; while exhibiting qualitatively correct navigation behaviors, their navigation path is usually far from optimal, e.g., wobbling on a straight path [9].

In this paper, we introduce Visually Adaptive Geometric Navigation (VAGN), a novel paradigm which marries geometric and visual navigation using a classical geometric motion planner and an online visual parameter planner. Although we intentionally frame the VAGN paradigm broadly enough to be instantiated by any visual processing and geometric planning methods, in the experiments of this paper, we implement VAGN's visual parameter planner with a Convolutional Neural Network (CNN) and its geometric motion planner with an existing sampling-based method [2]. After specifying the desired semantic-aware and geometric-feasible navigation behavior with a teleoperated human demonstration, VAGN is able to adaptively adjust the classical geometric planner's parameters with visual input (Fig. 1). VAGN is experimentally tested in an obstacle course with both rich semantic and geometric features. Comparison against a pure geometric approach and other approaches that utilize visual and/or geometric input shows that VAGN can most efficiently replicate the desired semantic-aware and geometric-feasible navigation behavior demonstrated by the human.

II. RELATED WORK

This section summarizes related work in terms of geometric and visual navigation systems.

*Equally contributing authors
[1]Department of Computer Science, University of Texas at Austin [2]The Computational and Information Sciences Directorate, Army Research Laboratory [3]Sony AI {shravanr, shreyas2, gwang, xiao, warnellg, joydeepb, pstone}@cs.utexas.edu

A. Geometric Navigation

Geometric navigation is often framed as an optimization problem: given a global path from a coarse global planner (e.g., Dijkstra's [10] or A* [11] algorithm) and a geometric representation of the surroundings (e.g., obstacle costmap [12]), the navigation planner seeks to find an optimal path according to a pre-defined cost function (e.g., distance to the closet obstacle and to the goal). Sampling-based [2] and optimization-based [1] methods are two major classes of geometric planners.

Recently, machine learning approaches have also been applied to address geometric navigation [13]–[15]: imitation learning [16], [17] and reinforcement learning [15], [18] are used to learn end-to-end local navigation policies; Learning from Hallucination [19]–[21] is a more recent learning paradigm to learn navigation planners by randomly exploring in an open space and synthetically adding (or "hallucinating") virtual obstacles to make the motion plans in the open space optimal. Another line of work, Adaptive Planner Parameter Learning (APPL) [22]–[26], is combined with classical approaches to learn a parameter policy and adaptively adjust planner parameters.

While being safe (collision-free) and efficient (shortest path) in the geometric sense [27], these systems do not consider any other information than geometry, e.g., semantics. VAGN takes advantage of the safety and efficiency of geometric planners and combines them with a vision component on top which interacts with the geometric planner through dynamic hyper-parameter adjustment.

B. Visual Navigation

Visual navigation systems have emerged with the success in deep learning and computer vision. One focus of visual navigation is through end-to-end learning, i.e., from raw RGB pixels to motion commands [5], [13] to move ground [7], [28], aerial [6], and water [29] vehicles. These systems do not require extensive engineering but can still capture subtle semantic information from the training set. Other more structured ways of learning visual navigation include learning planners [8], trajectory cost [9], and semantic mapping [30], [31].

Visual navigation systems, especially end-to-end approaches that directly map from pixels to torque, lack safety guarantees when being deployed out of simulation in the real physical world. Most visual navigation systems have a discrete action space (i.e., go straight, turn left, turn right) and do not have the potential to master fine-grained motor skills [8]. Autonomous robots that have been deployed long-term in the real world without any human supervision, unfortunately, most directly use geometric input, not vision [3], [4]. Due to their lack of safety assurance, roboticists are still reluctant to deploy purely vision-based systems in the real-world for extended period of unsupervised time.

Based on the observation that visual navigation can mostly generate discrete actions, VAGN only uses visual input to extract semantic features and generate high-level behaviors, e.g., driving quickly/slowly, being cautious around certain types of obstacles, by setting appropriate hyper-parameters for an underlying geometric planner. Regarding low-level geometric behaviors, e.g., obstacle avoidance, shortest path, VAGN leaves them to the properly parameterized geometric planner to assure safety and efficiency during real-world deployment.

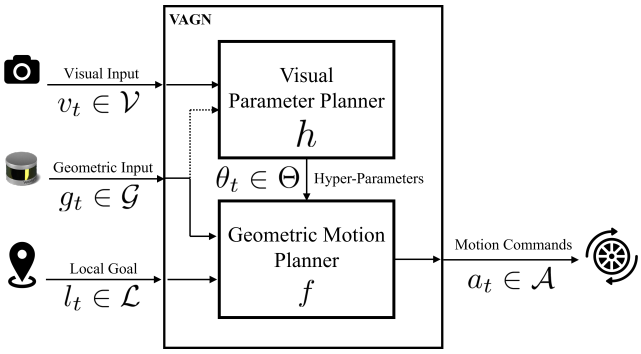

Fig. 2. VAGN Architecture.

III. APPROACH

VAGN employs a vision component which sits on top of a geometric navigation planner and makes high-level semantic decisions, which are then used to adjust the hyper-parameters of the geometric planner (Fig. 2). In this section, we describe each component of the VAGN system.

A. Geometric Motion Planner

VAGN employs a classical geometric motion planner to produce accurate, efficient, and collision-free motions that move the robot toward a goal. We describe the geometric motion planner as a function $f : \mathcal{G} \times \mathcal{L} \times \Theta \to \mathcal{A}$, where $\mathcal{G}$ is the space of geometric onboard perception (e.g., LiDAR, depth camera), $\mathcal{L}$ is all the information related with local goal, including necessary odometry and localization, Θ is the hyper-parameter space for f (e.g., maximum velocity, inflation radius), and $\mathcal{A}$ is the planner's action space (e.g., commanded linear and angular velocities). At each time step t, the geometric motion planner, parameterized by $\theta_t \in \Theta$, receives geometric input $g_t \in \mathcal{G}$ and goal-related information $l_t \in \mathcal{L}$ and produces motion command $a_t \in \mathcal{A}$.

The geometric motion planner f is responsible for generating precise, efficient, and collision-free motions that move the robot from its current location toward the goal location. While semantic-aware navigation is typically not possible for the geometric motion planner, VAGN provides this system with a level of semantic awareness through f's hyper-parameters, which are dynamically adjusted by a vision-based parameter planner.

B. Visual Parameter Planner

Since the high-dimensional visual input is prone to subtle environmental variations (e.g., lighting conditions), directly interacting with the geometric planner may lead to spurious and suboptimal motions. Therefore, VAGN does not allow the low-level geometric planner to directly interact with the

visual input, in contrast to most end-to-end visual navigation approaches [5]–[7]. The only interface from the geometric planner to the visual input is through its hyper-parameters θ_t at each time step.

We describe the visual parameter planner as a function $h : \mathcal{V} \times \Phi \to \Theta$, where $\mathcal{V}$ is the space of visual onboard perception (e.g., RGB camera), Φ is the space of h's own internal parameters (e.g., neural network weights and biases), and Θ is still f's hyper-parameter space. At each time step t, the visual parameter planner, parameterized by a constant $\phi \in \Phi$, receives visual input $v_t \in \mathcal{V}$ and produces a parameter set $\theta_t \in \Theta$ to be used by the geometric motion planner f. Note that unlike θ_t that changes at each deployment time step, ϕ is pre-determined and fixed during deployment.

The visual parameter planner h and the geometric motion planner f, interfacing via f's hyper parameter θ_t at each time step t, work together to enable semantic-aware and geometrically-feasible navigation.

C. Visual Context Predictor and Parameter Library

In general, the visual parameter planner h's fixed parameter $\phi \in \Phi$ can be determined pre-deployment using different approaches, e.g., classical or learning methods. In this paper, the visual parameter planner $h : \mathcal{V} \times \Phi \to \Theta$ is instantiated by imposing two intermediate functions, i.e., a parameterized visual context predictor $d : \mathcal{V} \times \Psi \to \mathcal{C}$ (h and d may have different parameter spaces, Φ and Ψ), and a one-to-one mapping $p : \mathcal{C} \to \Theta$. $c \in \mathcal{C}$ denotes a visual context, i.e., a visually cohesive region, and $\psi \in \Psi$ is d's parameters. We simplify the notation by writing $d_\psi : \mathcal{V} \to \mathcal{C}$, where $\psi \in \Psi$. Therefore, $\theta_t = p(d_\psi(v_t))$.

We also assume that a library $\mathcal{B}$ of f's hyper-parameters is obtainable, as a subset of Θ. Each parameter set $\theta \in \mathcal{B}$ is associated with a visual context. Such a library can be manually constructed by roboticists who are familiar with the underlying geometric motion planner f (e.g., using existing parameter tuning guides [32]), or automatically learned through teleoperated demonstration [22], corrective interventions [23], evaluative feedback [24], or reinforcement learning [25].

In this paper, VAGN automatically learns the first intermediate function, d, and the parameter library, $\mathcal{B}$ (and thus the second intermediate function, p), from a human teleoperated demonstration of desired semantic-aware and collision-free navigation behavior with manual segmentation [22]. To be specific, the teleoperated demonstration is collected as a sequence $\mathcal{D} = \{v_i^D, g_i^D, a_i^D\}_{i=1}^N$ of N steps in length, which is then segmented into K contexts with $K-1$ segmentation points, τ_1, τ_2, ..., τ_{K-1} with $\tau_0 = 1$ and $\tau_K = N+1$: $\{\mathcal{D}_k = \{v_i^D, g_i^D, a_i^D \mid \tau_{k-1} \le i < \tau_k\}\}_{k=1}^K$. For each context $\mathcal{D}_k$, an optimal parameter set θ_k^* is learned through behavior cloning so that the geometric motion planner f produces the closest actions to the demonstration:

$$\theta_k^* = \underset{\theta}{\operatorname{argmin}} \sum_{(g,a)\in\mathcal{D}_k} ||a - f(g,\theta)||_H, \tag{1}$$

where $||\cdot||_H$ indicates the weighted Euclidean norm with weights specified by a diagonal matrix H to weigh each action dimension. Eqn. 1 can be solved by any black-box optimization technique; we use CMA-ES [33]. The one-to-one mapping p is then simply $p(c_k) = \theta_k^*$.

Algorithm 1 VAGN

1: **Input:** geometric motion planner f, visual parameter planner h, instantiated as a visual context predictor d_{ψ^*} and a one-to-one mapping p (from context to parameters in library $\mathcal{B}$).
2: **for** $t = 1 : T$ **do**
3: Receive visual input v_t, geometric input g_t, local goal l_t
4: Identify visual context $c_t = d_{\psi^*}(v_t)$
5: Select planner parameter $\theta_t = p(c_t)$
6: Navigate with $f(g_t, l_t, \theta_t)$.
7: **end for**

To learn function d_ψ, VAGN takes the visual (and potentially geometric) input to form a supervised dataset $\{v_i^D, c_i\}_{i=1}^N$, where $c_i = k$ if i is in the k-th segment. It then finds the optimal parameters ψ^* by

$$\psi^* = \underset{\psi}{\operatorname{argmax}} \sum_{i=1}^{N} \log \frac{\exp\left(d_\psi(v_i^D)[c_i]\right)}{\sum_{c=1}^{K} \exp\left(d_\psi(v_i^D)[c]\right)}, \tag{2}$$

where $[c]$ denotes the output probability of class c. Since d_ψ takes in visual input, VAGN uses a CNN to determine which context k each v_t comes from during runtime.

The entire VAGN algorithm is shown in Alg. 1.

IV. EXPERIMENTS

We implement VAGN to evaluate whether a classical geometric navigation planner whose hyper-parameters are dynamically adjusted by a vision system can exhibit desired semantic-aware and collision-free navigation at the same time. Given one single trial of teleoperated demonstration of the desired navigation behavior, VAGN's performance is compared against a purely geometric navigation method [2] and end-to-end navigation systems that take in visual and/or geometric input [16]. The experiment results show that VAGN demonstrates the geometrically safest and semantically most similar navigation behavior to the demonstration in terms of collision-free navigation success rate, Hausdorff distance, context-based obstacle clearance, and velocity difference. Note that instead of speed as one of the most important metrics for conventional navigation approaches, we focus on VAGN's resemblance to the desired navigation behavior specified by the demonstration.

A. Implementation Details

1) Robot Platform: We implement VAGN on a Clearpath Jackal, which is a small, differential-drive, wheeled unmanned ground vehicle with a top speed of 2.0m/s and a turning radius of zero. It is equipped with a Velodyne VLP-16 LiDAR to provide geometric perception and a FLIR RGB camera for visual input. The Velodyne 3D point cloud is projected into 2D laser scan for 2D navigation and the camera streams `256x320` (down-sampled from original `1024x1280`) RGB images. It runs Robot Operating System (ROS) onboard with the commonly-used `move_base` navigation stack. We apply VAGN to the ROS `move_base`

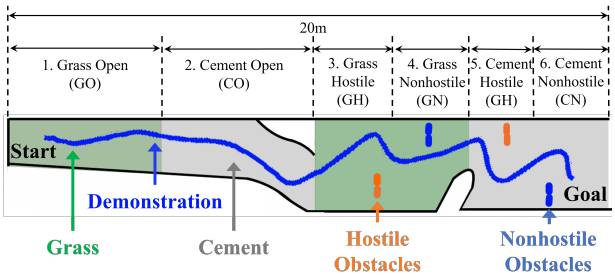

Fig. 3. Obstacle Course with Geometric and Semantic Features and the Six Visual Contexts.

stack's local planner, the Dynamic Window Approach (DWA) planner. The global planner, i.e., Dijkstra's algorithm, provides the local DWA with a coarse global path.

2) Obstacle Course: To test VAGN, we construct an obstacle course which contains both rich semantic and geometric features. As shown in Fig. 3, the black lines represent the boundaries of the course constructed by cardboard boxes. The ground of the course is divided into semantically different segments: two green segments covered by grass and two grey segments paved by cement. Apart from terrain, we also want to understand whether VAGN can enable different navigation behaviors based on obstacle appearance. To do so, we place two different types of obstacles in the environment: *hostile obstacles*, which the robot should stay especially faraway from, and *nonhostile obstacles*, which the robot simply needs to avoid colliding with at minimum clearance. We use small orange bins as hostile obstacles and small blue bins (geometrically identical to the orange bins) as nonhostile obstacles. A total of six unique contexts exist on the course: Grass Open (GO), Cement Open (CO), Grass with Hostile Obstacles (GH), Grass with Nonhostile Obstacles (GN), Cement with Hostile Obstacles (CH), and Cement with Nonhostile Obstacles (CN). Note that all these features are semantic and cannot be perceived by the geometric laser scan.

3) Human Demonstration: The authors first decide on the desired semantic-aware and collision-free navigation behavior in advance. Then, one author of the paper uses a PS4 controller to teleoperate the robot through the obstacle course to demonstrate such decided behavior (blue trajectory in Fig. 3). The human demonstrator chooses to drive cautiously and slowly on the grass and speeds up on the cement surfaces. The author keeps a large distance when facing the orange hostile obstacles, but hugs very closely to the blue nonhostile ones. During teleoperation, we also run the ROS `move_base` navigation stack in the background, whose motion commands are preempted by human demonstration. We collect the sequence of teleoperated linear and angular velocity commands, RGB images, local goals on the `move_base` global path 1m away from the robot, and all inputs to the `move_base` node, including laser scans and global navigation goal. The training set consists only 749 points. Note that with such a small training set, the learned visual context predictor for VAGN is not expected to generalize well to unseen environments. But as mentioned before, the purpose of the experiments is to demonstrate VAGN as a new navigation paradigm to enable semantic-aware and collision-free navigation, not as a generalizable machine learning algorithm.

B. VAGN and Baseline Methods

We implement the following six methods: (1) pure geometric (DWA) [2], (2) end-to-end geometric only (E2E-G) [16], (3) end-to-end vision only (E2E-V) [6], (4) end-to-end vision and geometric (E2E-VG) [34], (5) VAGN with visual context (VAGN-V), and (6) VAGN with visual and geometric context (VAGN-VG).

1) DWA: DWA is the default local planner in `move_base`. Based on a geometric costmap around the robot built by 2D laser scans, DWA samples physically feasible linear and angular velocities and rolls out these commands using a forward kinodynamics model. It then evaluates these candidate trajectories using a cost function based on distance to the closest obstacle, to the local path, and to the local goal. DWA uses its default planner parameters (first row in Tab. I).

2) E2E-G: E2E-G is a local planner similar to the approach by Pfeiffer et al. [16]. It takes in the 897-dimensional laser scan, concatenates it with the 2D local goal, and feeds them into a four-layer neural network with `[128, 128, 64, 2]` neurons. The output is directly linear and angular velocities.

3) E2E-V: E2E-V is similar to the work by Giusti et al. [6], but adds the local goal in addition to the RGB image as input of the neural network. The RGB image is consumed by four convolution and maxpooling layers (left portion of Fig. 4), and the learned embedding is then concatenated with the local goal (not shown in Fig. 4). The output is also linear and angular velocities.

4) E2E-VG: E2E-VG is similar to the work by Everett et al. [34], but not with a focus on social navigation. It concatenates the CNN output from RGB image with laser scan and local goal and uses a four-layer neural network with `[128, 128, 64, 2]` neurons to produce motion commands.

5) VAGN-V: Our VAGN-V implementation takes in RGB image in the high-level context predictor and then dynamically adjusts the local DWA planner's hyper parameters. All CNNs utilize the same architecture as the previous methods.

6) VAGN-VG: Our VAGN-VG implementation takes in both RGB image and laser scan in the context predictor. Its performance is tested against VAGN-V to see if adding extra geometric information in the context prediction can improve performance. Both visual context predictors run at around 5Hz. The full VAGN-VG visual context predictor architecture is shown in Fig. 4.

C. Parameter Learning

Following the demonstration, we find each θ_k^* using CMA-ES [33] as our black-box optimizer. The optimization runs on a single Asus Desktop (Intel Xeon) and takes about six hours. The specific parameters learned

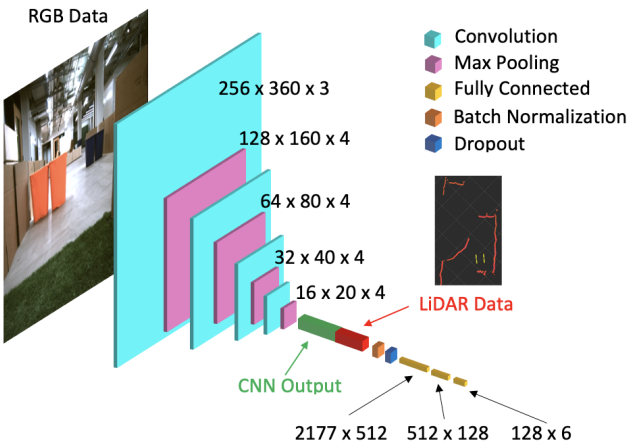

Fig. 4. VAGN Visual Context Predictor (same convolution and maxpooling layers are used for visual input in other methods).

by VAGN include MAX_VEL_X (v), MAX_VEL_THETA (w), VX_SAMPLES (s), VTHETA_SAMPLES (t), OCCDIST_SCALE (o), PATH_DISTANCE_BIAS (p), GOAL_DISTANCE_BIAS (g), and INFLATION_RADIUS (i). The default DWA parameters and the learned parameters for the six contexts are shown in the first row and last six rows in Tab. I.

TABLE I
DEFAULT DWA AND VAGN PARAMETERS

	v	w	s	t	o	p	g	i
DEF.	0.50	1.57	6	20	0.10	0.75	1.00	0.30
1 GO	1.22	0.71	16	31	0.71	0.19	0.99	0.18
2 CO	1.86	1.00	18	27	0.51	0.62	0.44	0.07
3 GH	1.34	0.97	7	45	0.22	0.65	0.17	0.32
4 GN	1.42	2.19	9	41	0.84	0.30	0.99	0.10
5 CH	1.50	1.66	11	40	0.77	0.75	0.97	0.44
6 CN	1.86	0.78	20	56	0.82	0.40	0.92	0.17

Although the interplay among all parameters may be intricate, many of them are intuitive. For example, VAGN finds that the demonstrator limits the max velocity on grass while accelerates on cement (v). Additionally, VAGN recognizes that the demonstrator is careful and stays faraway from the orange hostile obstacles by increasing the inflation radius (i) and decreasing the max velocity (v). On the other hand, when encountering the nonhostile obstacles, VAGN is able to maintain the same speed (v) for the specific terrain and travels closer to the object due to the decreased inflation radius (i).

D. Results

We conduct ten trials each method in the obstacle course (Fig. 3). We use `amcl` localization to localize the robot trajectory and record the velocity profiles. Due to extremely small training set (749 data points), all three end-to-end approaches are not able to finish traversing the entire obstacle course without any collisions on any trial. The robot exhibits certain signs of obstacle avoidance behaviors (e.g., moving slightly toward left when getting close to an obstacle on the right), but it is not able to completely avoid all obstacles and successfully reach the other side of the course. However, both visual E2E-V and E2E-VG learn to speed up on cement and slow down on grass, which shows that vision is suitable to generate high-level semantics-based navigation behavior, rather than low-level precise motor skills such as obstacle avoidance. All ten trials of DWA, VAGN-V, and VAGN-VG successfully traverse the course without any collision (first row in Tab. II).

TABLE II
SUCCESS RATE (SR), HAUSDORFF DISTANCE (HD), AND VELOCITY DIFFERENCE (VD)

	DWA	E2E-G	E2E-V	E2E-VG	VAGN-V	**VAGN-VG**
SR	100%	0%	0%	0%	100%	**100%**
HD	0.9m	N/A	N/A	N/A	0.6m	**0.3m**
VD	0.9m/s	N/A	N/A	N/A	0.6m/s	**0.4m/s**

The second row of Tab. II shows the Hausdorff distance of each method with respect to the human demonstration. Averaged over ten trials each method, VAGN-VG achieves the smallest average Hausdorff distance, while the trajectory executed by DWA is the most distinct from the demonstration.

Fig. 5 shows close-ups of the overhead view of the robot trajectory in the vicinity of hostile and nonhostile obstacles. The blue trajectory denotes the human demonstration, while the green one is VAGN-VG, yellow one VAGN-V, and red one default DWA. Around hostile obstacles, VAGN-VG and VAGN-V learn to increase obstacle inflation radius and stay away from the obstacles, but around nonhostile obstacles, inflation radius is decreased and the robot hugs close to the obstacles. Note whether or not an obstacle is hostile is not observable by LiDAR alone. Since both hostile and nonhostile obstacles have the same geometric shape, DWA simply treats them as the same. To be specific, Tab. III shows the average minimum distance to the two hostile (H) and nonhostile (N) obstacles on grass (G) and on cement (C).

TABLE III
AVERAGE MINIMUM DISTANCE TO OBSTACLES

	3 GH	4 GN	5 CH	6 CN
Demonstration	0.70m	0.36m	0.83m	0.23m
VAGN-VG	**0.64m**	**0.35m**	**0.72m**	**0.29m**
VAGN-V	0.62m	0.45m	0.64m	0.37m
DWA	0.46m	0.52m	0.54m	0.49m

In terms of velocity similarity, Fig. 6 shows the linear velocity profiles of different approaches in comparison to the human demonstration. The x-axis is the distance to the start point in meters (since the obstacle course is one directional), and the vertical axis is linear velocity. At beginning, VAGN-V and VAGN-VG learn to drive slowly on grass and speed up on cement, just as the human demonstration. Note that the grass and cement are both a perfectly traversable plane in a geometric sense, and the color and texture of grass

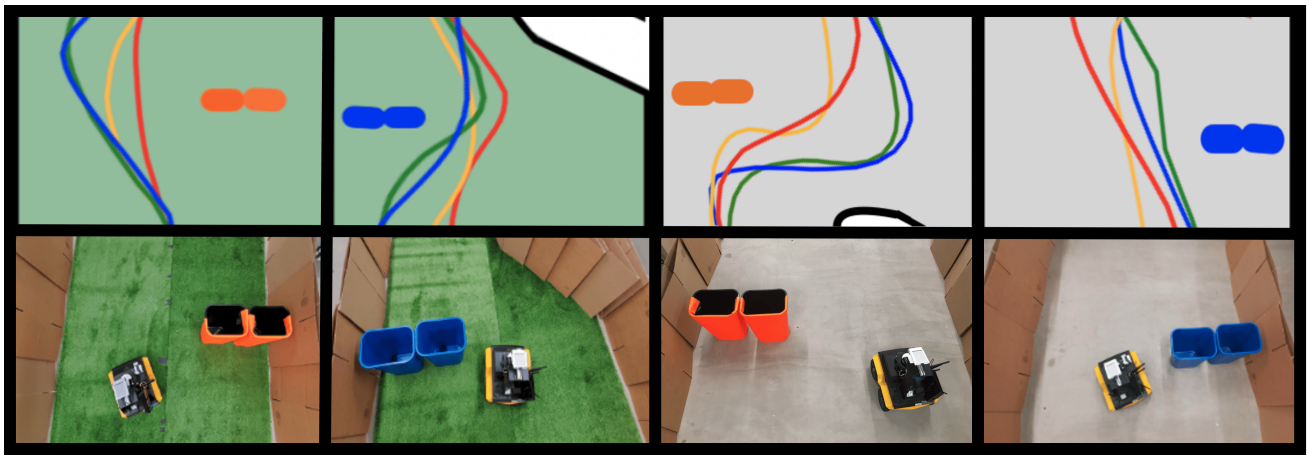

Fig. 5. Navigation around Hostile (orage) and Nonhostile (blue) Obstacles: demonstration (blue), VAGN-VG (green), VAGN-V (yellow), and DWA (red).

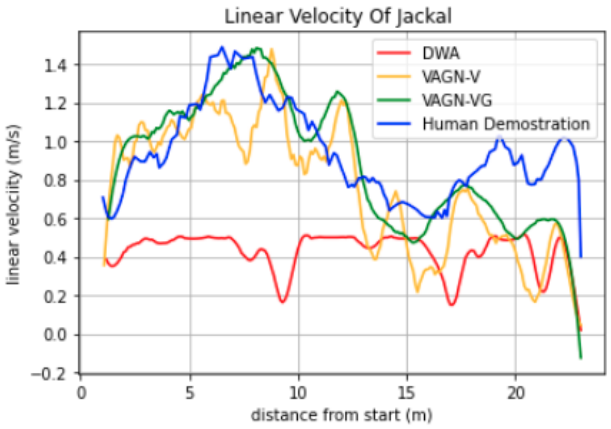

Fig. 6. Linear Velocity Profiles: human demonstration (blue), VAGN-VG (green), VAGN-V (yellow), and DWA (red).

and cement are not perceivable by LiDAR. Therefore, DWA navigates at roughly constant speed and completely ignores the ground type. We also present the average Velocity Difference (VD) with respect to the human demonstration in the third row of Tab. II (absolute velocity difference between two closest points on two trajectories). Again, VAGN-VG shows the smallest velocity difference, while DWA is the most different.

To summarize the experiment results, it is difficult for all end-to-end approaches (E2E-G, E2E-V, E2E-VG) to learn successful low-level precise obstacle-avoidance behaviors using such a small training set. Therefore they fail in all ten trials due to collision with obstacles. But they exhibit signs of high-level navigation behaviors such as accelerating and decelerating on cement and grass. Thanks to the classical local planner, DWA, VAGN-V, and VAGN-VG can successfully perform low-level obstacle avoidance. DWA does not consider semantics at all, and therefore produces navigation behaviors most different from human demonstration. VAGN-VG utilizes both vision and geometry in context prediction and achieves slightly better performance compared to VAGN-V's vision only context predictor. We conclude that our VAGN-VG is the best among all alternatives tested in our experiments to enable semantic-aware and collision-free navigation at the same time.

V. CONCLUSIONS

This paper presents Visually Adaptive Geometric Navigation (VAGN), a novel paradigm that marries geometric and visual navigation using a classical geometric motion planner and an online visual parameter planner. VAGN employs a visual component which sits on top of a geometric planner and produces high-level semantic decisions (e.g., increase/decrease speed, be aggressive/conservative around obstacles). These high-level decisions interface with the low-level geometric planner via planner hyper-parameters. The visual context predictor dynamically adjusts planner parameters in response to different semantics in the environment while the geometric planner produces safe, accurate, and efficient local motions. VAGN is tested on an autonomous ground robot and our experiment results show that VAGN can enable similar semantic-aware and collision-free navigation behaviors as specified by a human demonstration, compared to other baselines which fail either at collision-avoidance or considering semantics. One future research direction is to automate visual context segmentation. Another direction is to learn VAGN through less demanding human interaction modalities than teleoperated demonstration, e.g., evaluative feedback.

ACKNOWLEDGMENTS

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), ARL, 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.

[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] J. Biswas and M. Veloso, "The 1,000-km challenge: Insights and quantitative and qualitative results," *IEEE Intelligent Systems*, vol. 31, no. 3, pp. 86–96, 2016.

[4] P. Khandelwal, S. Zhang, J. Sinapov, M. Leonetti, J. Thomason, F. Yang, I. Gori, M. Svetlik, P. Khante, V. Lifschitz *et al.*, "Bwibots: A platform for bridging the gap between ai and human–robot interaction research," *The International Journal of Robotics Research*, vol. 36, no. 5-7, pp. 635–659, 2017.

[5] F. Bonin-Font, A. Ortiz, and G. Oliver, "Visual navigation for mobile robots: A survey," *Journal of intelligent and robotic systems*, vol. 53, no. 3, pp. 263–296, 2008.

[6] 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.

[7] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang *et al.*, "End to end learning for self-driving cars," *arXiv preprint arXiv:1604.07316*, 2016.

[8] S. Gupta, J. Davidson, S. Levine, R. Sukthankar, and J. Malik, "Cognitive mapping and planning for visual navigation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2616–2625.

[9] G. Kahn, P. Abbeel, and S. Levine, "Badgr: An autonomous self-supervised learning-based navigation system," *arXiv preprint arXiv:2002.05700*, 2020.

[10] E. W. Dijkstra *et al.*, "A note on two problems in connexion with graphs," *Numerische mathematik*, vol. 1, no. 1, pp. 269–271, 1959.

[11] 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.

[12] L. Jaillet, J. Cortés, and T. Siméon, "Sampling-based path planning on configuration-space costmaps," *IEEE Transactions on Robotics*, vol. 26, no. 4, pp. 635–646, 2010.

[13] X. Xiao, B. Liu, G. Warnell, and P. Stone, "Motion control for mobile robot navigation using machine learning: a survey," *arXiv preprint arXiv:2011.13112*, 2020.

[14] B. Liu, X. Xiao, and P. Stone, "A lifelong learning approach to mobile robot navigation," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1090–1096, 2021.

[15] Z. Xu, X. Xiao, G. Warnell, A. Nair, and P. Stone, "Machine learning methods for local motion planning: A study of end-to-end vs. parameter learning," in *2021 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*. IEEE, 2021.

[16] 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 *IEEE International Conference on Robotics and Automation*. IEEE, 2017.

[17] X. Xiao, J. Biswas, and P. Stone, "Learning inverse kinodynamics for accurate high-speed off-road navigation on unstructured terrain," *IEEE Robotics and Automation Letters*, 2021.

[18] A. Faust, K. Oslund, O. Ramirez, A. Francis, L. Tapia, M. Fiser, and J. Davidson, "Prm-rl: Long-range robotic navigation tasks by combining reinforcement learning and sampling-based planning," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 5113–5120.

[19] X. Xiao, B. Liu, G. Warnell, and P. Stone, "Toward agile maneuvers in highly constrained spaces: Learning from hallucination," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1503–1510, 2021.

[20] X. Xiao, B. Liu, and P. Stone, "Agile robot navigation through hallucinated learning and sober deployment," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021.

[21] Z. Wang, X. Xiao, A. J. Nettekoven, K. Umasankar, A. Singh, S. Bommakanti, U. Topcu, and P. Stone, "From agile ground to aerial navigation: Learning from learned hallucination," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021.

[22] 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.

[23] Z. Wang, X. Xiao, B. Liu, G. Warnell, and P. Stone, "Appli: Adaptive planner parameter learning from interventions," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021.

[24] Z. Wang, X. Xiao, G. Warnell, and P. Stone, "Apple: Adaptive planner parameter learning from evaluative feedback," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7744–7749, 2021.

[25] Z. Xu, G. Dhamankar, A. Nair, X. Xiao, G. Warnell, B. Liu, Z. Wang, and P. Stone, "Applr: Adaptive planner parameter learning from reinforcement," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021.

[26] X. Xiao, Z. Wang, Z. Xu, B. Liu, G. Warnell, G. Dhamankar, A. Nair, and P. Stone, "Appl: Adaptive planner parameter learning," *arXiv preprint arXiv:2105.07620*, 2021.

[27] D. Perille, A. Truong, X. Xiao, and P. Stone, "Benchmarking metric ground navigation," in *2020 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*. IEEE, 2020, pp. 116–121.

[28] H. Karnan, G. Warnell, X. Xiao, and P. Stone, "Voila: Visual-observation-only imitation learning for autonomous navigation," *arXiv preprint arXiv:2105.09371*, 2021.

[29] 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.

[30] D. Maturana, P.-W. Chou, M. Uenoyama, and S. Scherer, "Real-time semantic mapping for autonomous off-road navigation," in *Field and Service Robotics*. Springer, 2018, pp. 335–350.

[31] M. Wigness, J. G. Rogers, and L. E. Navarro-Serment, "Robot navigation from human demonstration: Learning control behaviors," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1150–1157.

[32] K. Zheng, "Ros navigation tuning guide," in *Robot Operating System (ROS)*. Springer, 2021, pp. 197–226.

[33] N. Hansen, S. D. Müller, and P. Koumoutsakos, "Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (cma-es)," *Evolutionary computation*, vol. 11, no. 1, pp. 1–18, 2003.

[34] M. Everett, Y. F. Chen, and J. P. How, "Motion planning among dynamic, decision-making agents with deep reinforcement learning," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 3052–3059.