Visually Adaptive Geometric Navigation

Shravan Ravi[1*], Shreyas Satewar[1*], Gary Wang[1*], Xuesu Xiao[2,3],
Garrett Warnell[1,4], Joydeep Biswas[1], and Peter Stone[1,5]

***Abstract*— While classical autonomous navigation systems can move robots from one point to another in a collision-free manner due to geometric modeling, recent approaches to visual navigation allow robots to consider semantic information. However, most visual navigation systems do not explicitly reason about geometry, which may potentially lead to collisions. 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 purely geometric reasoning. VAGN is implemented on two different physical ground robots with different action spaces, navigation systems, and parameter sets. VAGN demonstrates superior navigation performance in both a test course with rich semantic and geometric features and a real-world deployment compared to other navigation baselines using visual and/or geometric input.**

I. INTRODUCTION

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. These geometric navigation systems have proven to be highly reliable, and have been successfully deployed without supervision in real-world settings over extended periods of time [3].

Thanks to successes in computer vision driven by machine learning, there has recently been a surge of interest in visual navigation from within the robotics community [4]–[6]. Using visual input for navigation is attractive for many reasons. Chief among these is that the extra information provided by the semantics of the environment (as opposed to only its geometry) creates the opportunity for the robot to make more complex and intelligent movement decisions about where and how to move (Fig. 1). However, visual navigation systems, especially those that rely on monocular cameras, often lack a collision-free guarantee and their navigation path is generally sub-optimal, e.g., wobbling on a straight path [7].

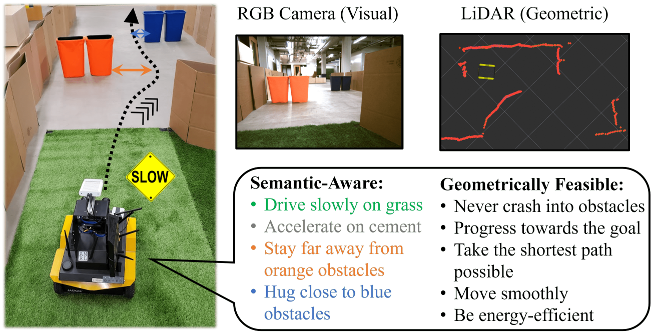

Fig. 1: Visually Adaptive Geometric Navigation (VAGN) enables semantic-aware and geometrically feasible navigation in real-world scenarios.

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 using a Convolutional Neural Network (CNN) and its geometric motion planner with existing sampling-based methods. After being shown 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 both an artificially constructed obstacle course with both rich semantic and geometric features and in natural real-world scenarios. Comparison against a pure geometric approach and other approaches that utilize visual and/or geometric input indicates that VAGN can effectively replicate the desired semantic-aware and geometric-feasible navigation behavior demonstrated by the human in both domains.

II. RELATED WORK

VAGN is a hybrid of geometric and visual navigation systems. This section reviews the literature pertaining to each of these approaches.

A. *Geometric Navigation*

Recently, geometric navigation has been improved using machine learning approaches [8]–[10]: imitation learning [11], [12] and reinforcement learning [10], [13] are used to learn end-to-end local navigation policies; Learning from Hallucination [14]–[16] 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.

[*] Equally contributing authors

[1] Department of Computer Science, University of Texas at Austin [2] Department of Computer Science, George Mason University [3] Everyday Robots [4] Army Research Laboratory [5] Sony AI {shravanr, shreyas2, gwang, xiao, warnellg, joydeepb, pstone}@cs.utexas.edu

Another line of work, Adaptive Planner Parameter Learning (APPL) [17]–[21], 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 [22], 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. Additionally, hyper-parameter tuning based navigation optimization has been largely confined to artificial courses that are specifically designed to highlight their respective effectiveness; we use VAGN to extend these controlled experiments to a real-world field deployment.

B. Visual Navigation

Visual navigation systems have emerged with the success of deep learning and computer vision. Here, we use the term *visual navigation* to denote specifically using vision to directly control robot motion to navigate. One focus of these system is end-to-end learning, i.e., learning a single function that directly maps raw RGB pixel input to motion commands [4], [8]. These systems do not require extensive engineering, and can capture subtle semantic information from the training set. Other, more structured ways of implementing visual navigation include learning planners [23], trajectory costs [7], and semantic mapping [24].

Unfortunately, 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. For this reason, the vast majority of the autonomous robots that have been successfully deployed long-term in the real world without any supervision utilize purely geometric navigation systems. That is, due to the lack of safety guarantees for visual navigation systems, 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 successful visual navigation systems predominantly focus on generating discrete actions [23], VAGN only uses visual input to extract semantic features and generate high-level behaviors (e.g., driving quickly/slowly or being cautious around certain types of obstacles) by setting appropriate hyper-parameters for an underlying geometric planner. At a high level, this paradigm of using visual input to trigger changes in a classical geometric planner can produce similar navigation behaviors as the Visual Teach and Repeat (VTR) paradigm introduced by Furgale et al. [25]. However, unlike VTR, VAGN does not require the complex machinery of mapping and localization, instead learning a holistic mapping directly from single images to planner parameters that can theoretically be deployed in environments beyond those in which it has previously operated.

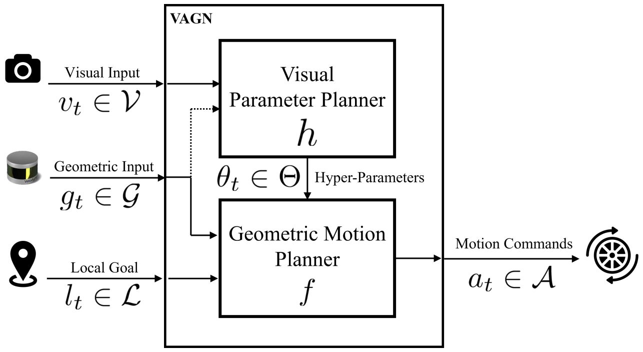

Fig. 2: VAGN Architecture. VAGN combines both visual and geometric navigation by using visual inputs to drive a parameter planner that determines high level behavior modes for a geometric motion planner.

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 relevant to planning to reach the local goal, including odometry and localization information, 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, in contrast to most end-to-end visual navigation approaches [4]–[6], VAGN does not allow the low-level geometric planner to directly interact with the visual input. 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 $h : \mathcal{V} \times \Phi \rightarrow \Theta$, 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 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 ϕ is pre-determined and fixed during deployment, whereas θ_t may change at each deployment time step.

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 $h : \mathcal{V} \times \Phi \rightarrow \Theta$ is instantiated by imposing two intermediate functions, i.e., a parameterized visual context predictor $d : \mathcal{V} \times \Psi \rightarrow \mathcal{C}$ (h and d may have different parameter spaces, Φ and Ψ), and a one-to-one mapping $p : \mathcal{C} \rightarrow \Theta$. $c \in \mathcal{C}$ denotes a visual context, i.e., a visually cohesive region, and $\psi \in \Psi$ is d's parameters. We simplify the notation by writing $d_\psi : \mathcal{V} \rightarrow \mathcal{C}$, 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 [26]), or automatically learned through teleoperated demonstration [17], corrective interventions [18], evaluative feedback [19], or reinforcement learning [20].

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 [17]. 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 $\tau_K = N+1$: $\{\mathcal{D}_k = \{v_i^D, g_i^D, a_i^D \,|\, \tau_{k-1} \leq i < \tau_k\}\}_{k=1}^K$. 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 [27]. 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 $d_{\psi^*}$ 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 estimates the optimal parameters ψ^* defined 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 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. We deployed VAGN on two robots—a Clearpath Jackal and Boston Dynamics Spot—that operate in different domains using different underlying navigation systems. For the purpose of our evaluation, the most relevant comparison point for VAGN is each robot's underlying geometric planner itself (with default hyper-parameters); if the VAGN system exhibits more of the desired behavior than the geometric planner alone, then we may conclude that classical geometric planners can indeed be endowed with semantic-aware behavior through hyper-parameter tuning by a vision system. The experiment results show that VAGN produces navigation behaviors that are both geometrically safe and also the most similar to the demonstrator 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 primarily on VAGN's ability to replicate the desired navigation behavior as specified by the demonstration.

A. Jackal Obstacle Course Navigation

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.

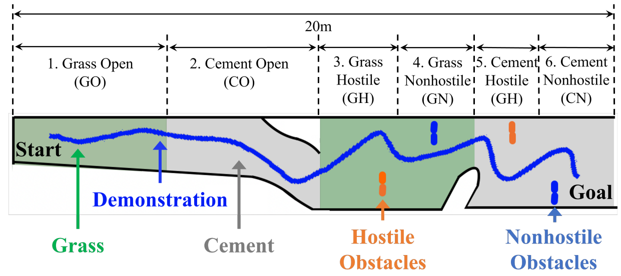

Fig. 3: Obstacle Course with Geometric and Semantic Features and the Six Visual Contexts.

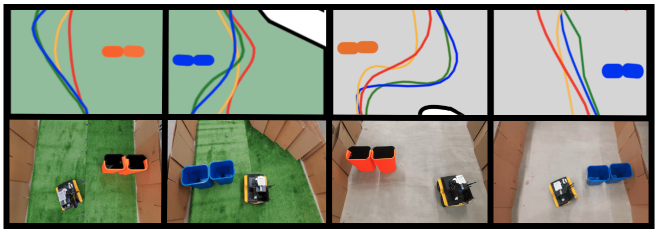

Fig. 4: Navigation around Hostile (orange) and Nonhostile (blue) Obstacles: demonstration (blue), VAGN-VG (green), VAGN-V (yellow), and DWA (red).

It runs Robot Operating System (ROS) onboard with the commonly-used `move_base` navigation stack. We apply VAGN to the ROS `move_base` stack's local planner, the Dynamic Window Approach (DWA) planner [2]. The global planner, i.e., Dijkstra's algorithm, provides the local DWA planner with a coarse global path.

As a controlled, pilot test of VAGN, we construct an obstacle course (20m) which contains both rich semantic and geometric features. As shown in Fig. 3, the black lines represent the boundaries of the course and 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 types of geometrically identical, but visually different obstacles in the environment: *hostile obstacles* (small orange bins), which the robot should stay especially far away from, and *nonhostile obstacles* (small blue bins), which the robot simply needs to avoid colliding with at minimum clearance. 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.

The authors first decide on the desired semantic-aware and collision-free navigation behavior in advance. To demonstrate the desired semantic-aware and collision-free navigation behavior, as determined by the authors in advance, one author of the paper uses a PS4 controller to teleoperate the robot through the obstacle course (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 the paradigm will only devolve to a classical geometric navigation system that continues to guarantee safety. 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.

We implement the following six methods: (1) pure geometric (DWA) [2], (2) end-to-end geometric only (E2E-G) [11], (3) end-to-end vision only (E2E-V) [5], (4) end-to-end vision and geometric (E2E-VG) [28], (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.

2) E2E-G: E2E-G is a local planner similar to the approach by Pfeiffer et al. [11]. 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. [5], 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, and the learned embedding is then concatenated with the local goal. The output is also linear and angular velocities.

4) E2E-VG: E2E-VG is similar to the work by Everett et al. [28], 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 5Hz.

Following the demonstration, we find each θ_k^* using CMA-ES [27] as our black-box optimizer. The optimization runs on a single Asus Desktop (Intel Xeon) and takes about six hours. The specific parameters learned by VAGN include MAX_VEL_X, MAX_VEL_THETA, VX_SAMPLES, VTHETA_SAMPLES, OCCDIST_SCALE, PATH_DISTANCE_BIAS, GOAL_DISTANCE_BIAS, and INFLATION_RADIUS. Although the interplay among all parameters may be intricate, many of them are intuitive. For example, VAGN recognizes that the demonstrator is careful and stays far away from the orange hostile obstacles by increasing the inflation radius and decreasing the max velocity. On the other hand, when encountering the nonhostile obstacles, VAGN is able to maintain the same speed for the specific terrain and travels closer to the object due to the decreased inflation radius.

To quantitatively determine the efficacy of VAGN, we conduct ten trials of each method in the obstacle course (Fig. 3). We use `amcl` localization to localize the robot trajectory and record the velocity profiles. None of the three end-to-end approaches is 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. I).

TABLE I: Success Rate (SR) and Hausdorff Distance (HD)

	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**

The second row of Tab. I 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 different from the demonstration.

Fig. 4 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 inflation radius and stay away from the orange bins, but around nonhostile obstacles, inflation radius is decreased and the robot hugs close to the blue bins. Since both hostile and nonhostile obstacles have the same geometric shape, DWA simply treats them as the same. Tab. II further shows the average minimum distance to the two hostile (H) and nonhostile (N) obstacles on grass (G) and on cement (C) for the successful navigation systems.

TABLE II: 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

TABLE III: Velocity Difference (m/s)

	1 G	2 C	3 GH	4 GN	5 CH	6 CN
VAGN-VG	**0.09**	**0.14**	**0.16**	**0.11**	**0.17**	**0.28**
VAGN-V	0.12	0.18	0.13	0.16	0.23	0.33
DWA	0.34	1.06	0.32	0.14	0.51	0.45

In terms of velocity similarity, Tab. III shows the average velocity difference in each context relative to the human demonstration. 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 an equally traversable plane in a geometric sense, and the color and texture of grass and cement are not perceivable by LiDAR. Therefore, DWA navigates at roughly constant speed and completely ignores the ground type.

B. Spot Field Deployment

From the controlled environment with the Jackal, we have demonstrated that VAGN-VG is both safe and the most accurate at emulating a human demonstration. To this end, we deploy VAGN-VG on another robot with a different underlying navigation system in a real-world environment to show the adaptability of the VAGN navigation paradigm. In particular, we use a BostonDynamics Spot, which is a mid-sized quadruped robot with a top speed of 1.6m/s and a turning radius of zero. It is equipped with a Velodyne VLP-16 LiDAR to provide geometric perception and an Azure Kinect DK Depth RGB camera for visual input that follows a similar down sampling protocol as Jackal. The Spot runs GraphNav, a different sampling-based, purely geometric navigation stack. Similar to the experiments with Jackal, we utilize teleoperation using a PS4 controller to provide a human demonstration. The training procedure for VAGN in this experiment is identical to that defined above on the Jackal. We deploy VAGN on Spot on a path that spans 200 meters and is comprised of many naturally unique contexts. We select four key points along the trajectory for close inspection: office (O), hallway (H), wheelchair ramp (R), and sidewalk (S). The whole course along with the human demonstration and pertinent contexts is shown in Fig. 5.

We deploy each method (GraphNav and VAGN-VG) and use ENML [29] to localize the robot trajectory and record the velocity profiles. GraphNav is unable to traverse the entire course—namely the office, ramp, and sidewalk contexts—and has to be manually reset by human teleoperation to

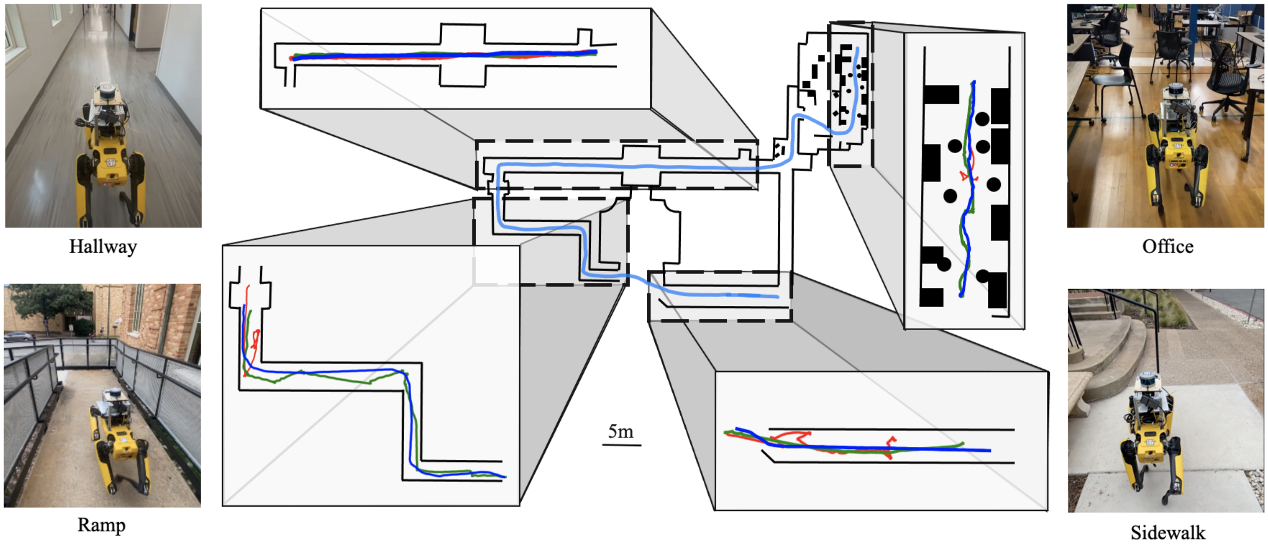

Fig. 5: Real-World Navigation Deployment of Spot: human demonstration (blue), VAGN-VG (green), GraphNav (red). Note GraphNav (red) gets stuck and fails in three of the four contexts due to the default parameters not allowing progress in constrained environments.

TABLE IV: Hausdorff Distance (m) Velocity Difference (m/s)

HD	1 O	2 H	3 R	4 S
GraphNav	5.09	4.34	2.03	1.45
VAGN-VG	**0.65**	**2.56**	**0.73**	**0.34**
VD	1 O	2 H	3 R	4 S
GraphNav	0.22	1.06	0.44	1.17
VAGN-VG	**0.15**	**0.14**	**0.27**	**0.23**

continue navigation; however, VAGN-VG is able to traverse the entire course successfully. Fig. 5 shows the overall trajectory of the human demonstration and the trajectories of each system in the key points of the course. The blue trajectory denotes the human demonstration, while the green is VAGN-VG, and red default GraphNav. VAGN-VG is able to stay in the middle of the hallway, and plan successful paths in very narrow corridors. Hausdorff Distance (HD) in Tab. IV between both GraphNav and VAGN-VG with respect to the demonstration shows that VAGN-VG plans a path that is more similar to human demonstration than GraphNav. Additionally, Tab. IV also shows the average velocity difference for each context relative to the human demonstration. VAGN-VG is able to correctly speed up in safer contexts such as the hallway and the sidewalk, but also slows down in more confined and cluttered contexts such as the ramp and office; GraphNav navigates at roughly constant speed and ignores the semantic information of the contexts.

C. Experiment Summary

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. However, they do exhibit signs of high-level navigation behaviors such as accelerating and decelerating on cement and grass. VAGN-V and VAGN-VG, on the other hand, learn these high-level behavior and *can* successfully perform low-level obstacle avoidance thanks to the classical local planners that they employ. Both underlying navigation systems alone do not consider semantics at all, and therefore produce 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 two autonomous ground robots 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 investigate how the visual context predictor can generalize to unseen scenarios, e.g., a visually different obstacle course with similar orange and blue obstacles.

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] 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.

[5] 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.

[6] 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.

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

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] 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.

[25] P. Furgale and T. D. Barfoot, "Visual teach and repeat for long-range rover autonomy," *Journal of Field Robotics*, vol. 27, no. 5, pp. 534–560, 2010.

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

[27] 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.

[28] 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.

[29] J. Biswas and M. Veloso, "Episodic non-markov localization: Reasoning about short-term and long-term features," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 3969–3974.