OWLS: Open-World Language-driven Servoing for Zero-Shot Traversability

Anonymous CVPR submission

Paper ID *****

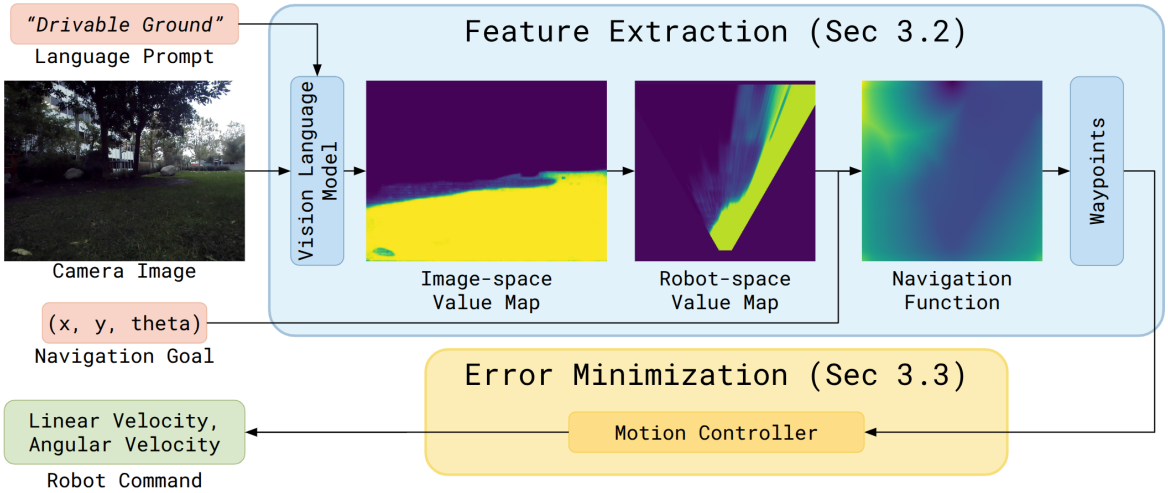

Figure 1. The system diagram of our method (Sec. 3). Our method takes in a camera image, language prompt, and a goal as input, extracts the image feature from the inputs (Sec. 3.2), and minimizes the feature error by generating robot velocity commands (Sec. 3.3).

Abstract

Traversable path selection by an autonomous robot requires reasoning over the geometry and semantics of an environment. Scene geometry provides information about occupancy to enable collision avoidance, while scene semantics provide information about physical affordances such as how much weight a surface can bear. In this work, we propose jointly optimizing the robot's trajectory over semantically rich features from vision language models (VLMs) together with scene geometry estimates in a modified approach to visual servoing. We select semantically informed navigation actions to a given goal location directly from camera-space optimization. We perform indoor and outdoor navigation experiments on a wheeled rover to select traversable paths. We find that VLM-based visual-servoing allows for lightweight semantically-informed navigation without task specific model training, access to prior knowledge of the scene, or requiring retention of a globally consistent map. These findings support viability of this methodological direction for robotic autonomy in remote environments such as space.

1. Introduction

Robots infer traversability in morphology-dependent ways. Current methods for inferring traversability are tied to particular robot platforms. For example, if a robot has legs, it will use an algorithm to mark regions traversable that may not be by wheeled platforms. Furthermore, safe navigation often necessitates hand-engineered safety constraints. For example, programming a robot to stay on sidewalks marked by detectable curbs is a manually specified heuristic of safe navigation given in advance to the system. In addition to these rules, it is still common to train closed set models to align observations to physical characteristics of the environment, such as expected traction and terrain class.

In this work, we leverage vision-language models (VLMs) to perform spatial reasoning over conceptual rules and to condition our estimation of a map regions' traversability on robot configuration. VLMs can be used to generate heatmaps over an environment to infer relation-

ships between observations and semantic concepts such as safety during interaction in the scene. Then, rules such as following "a safe path for a wheeled robot" can be physically grounded. This relationship between high level semantic rules and visual observations steps away from explicit segmentation for semantic decision making, enabling less manual system design by humans.

Our method is designed for the unique challenges of planetary exploration where low-texture regolith, GPS-denied environments, communication latency, and strict size, weight, and power (SWaP) constraints limit the applicability of many existing approaches. By adopting a mapless, ego-centric navigation strategy and controlling the robot directly from the current camera frame, we eliminate the high computational overhead of real-time SLAM and the energy cost of dense geometric sensors such as LiDAR. This framework effectively decouples semantic traversability logic from the underlying hardware. By conditioning the traversability heatmap through natural language prompts rather than hardware-embedded heuristics, we provide a reconfigurable reasoning layer. This decoupling allows a single navigation framework to be adapted across diverse morphologies from wheeled surface rovers to limbed climbing robots minimizing the costly, mission-specific software engineering typically required for new hardware configurations.

Contributions. Our contributions are as follows:

- We present OWLS, a visual servoing method for safely navigating traversable regions in widely varying domains from a single camera. Our novel visual servoing formulation leverages VLMs to extract actionable traversability estimates from camera imagery, enabling real-time trajectory optimization without the need for active sensors or persistent mapping.
- Hardware experiments with a 4-wheel rover in indoor and outdoor scenarios, demonstrating the effectiveness of our method in widely varying visual and trajectory domains, analyzing characteristics and pitfalls of off-the-shelf VLMs in visual navigation.

2. Related Work

Navigation via Image-based Control. Visual servoing directly utilizes visual information to do closed-loop control on the pose of the robot by minimizing an error function defined on a set of visual features (2D or 3D) [5]. While more commonly used in manipulation (e.g. to move an end effector to a grasp location relative to a target object [3]), visual servoing has also been successfully applied to controlling mobile robots [8, 9, 12]. Recently, an alternative line of research has explored using learned policies to make navigation decisions from RGB images directly without building globally consistent maps. This line of work utilize imitation learning [1] or offline reinforcement learning [17] to train neural networks that directly produce future actions from a history of image observations.

Semantic Reasoning for Traversability. Mobile robots determine the traversability, or the navigation cost of regions, from any combination of proprioceptive input [11], range-based sensors [2, 16], or cameras [7, 10] by applying geometric feature extractors with handcrafted heuristics or data-driven feature extractors with associated planning costs. However, the offline dataset is expensive to manually annotate, the planning costs for various terrain types are difficult to design, and the closed set of classes are often ambiguous or application-specific, often rendering many such methods a one-off solution. In contrast, our method directly takes the heatmap generated from the vision language model as the traversability, avoiding the need to manually design the planning cost while also achieving generalizable visual understanding from internet-scale data.

VLMs for Language-grounded Navigation. Recent success of VLMs has sparked their use in autonomous robot navigation, combining the strong semantic capabilities of VLMs with explicit spatial priors of classical methods. For example, VLMNav [4] utilizes perfect depth information from simulated home environments to generate ground-truth traversable regions, procedurally sample candidate paths, and utilize VLMs to select the best one leading to the goal. Others, such as BeHAV [19] and Open-Nav [20] similarly build 2D occupancy grid maps from 3D LiDAR sensors for collision avoidance in motion planning, and uses the open-vocab segmentation masks from a VLM to specify a goal or adhere to user preference as an additional objective (e.g. "avoid grass" although grass is traversable). In contrast, our method only uses a single calibrated camera to generate the top-down traversability map from a single image, without other spatial priors or sensors, and navigate the traversable regions.

3. Method

We propose a camera-only navigation algorithm by formulating traversable region navigation as an image-based visual servoing task (Sec. 3.1). Then, we explain how we extract features from images using VLMs for use in image-based control (Sec. 3.2). Finally, we explain how we perform robotic control through minimization of our error function defined between the robot pose and our extracted image features (Sec. 3.3).

3.1. Image-based Visual Servoing

Image-based visual servoing minimizes the *error function* $f_{\text{error}} : \mathcal{F} \to \mathcal{E}$, which is a mapping from image features $\mathbf{f} \in \mathcal{F} \subseteq \mathbb{R}^f$ to errors $\mathbf{e} \in \mathcal{E} \subseteq \mathbb{R}^e$ [5], such that the motion of the robot (and in turn, the camera) changes the observed image features. The feature extractor $f_{\text{feat}} : \mathcal{I} \to \mathcal{F}$ generates image features $\mathbf{f} \in \mathcal{F} \subseteq \mathbb{R}^k$ from an RGB

image $I \in \mathcal{I} = [0,1]^{h_\mathcal{I} \times w_\mathcal{I} \times 3}$ with image height $h_\mathcal{I}$ and image width $w_\mathcal{I}$. Then, as the motion of the image features is a function of the motion of the robot, the desired robot action $\mathbf{u}^*$ can be produced from the desired change in image features $\dot{\mathbf{f}}^*$ in the error minimizing direction.

In our case, the robot's workspace is planar such that the robot is represented by its 2D pose $p_r = (x_r, y_r, \theta_r) \in SE(2)$ and commanded by linear and angular velocity commands $u_r = (v_r, w_r)$. Our image feature is a selection of k discrete waypoints along a trajectory so that our error function measures the distance between the robot pose and the desired trajectory. Then, we optimize robot commands by minimizing error values such that

$$\min_{u_r} f_{\text{error}}(p_r, f_{\text{feat}}(I)). \tag{1}$$

Intuitively, the robot command changes the robot pose in the direction where the error value decreases, bringing it closer to the desired trajectory at each time step.

3.2. Feature Extraction

We construct an egocentric map from each camera view of semantic scores from a pretrained VLM. Then, we optimize for a traversable trajectory using this egocentric semantic value map together with a navigation function defined over an occupancy map to avoid collisions. This feature extraction method can be explicitly defined as

$$f_{\text{feat}} : \mathcal{I} \times \mathcal{P} \times \mathbb{R}^2 \to \mathcal{F} \tag{2}$$

which takes in an image $I \in \mathcal{I}$, language prompt $P \in \mathcal{P}$, and a 2D goal position $(x_g, y_g) \in \mathbb{R}^2$.While any goal from a global navigation planner can be used, we focus on demonstrating the local navigation capabilities of our method by choosing an arbitrarily far point in front of the robot.

Robot-space Value Map. Open-vocab segmentation models such as Grounded SAM 2 [13, 14] produce a pixel-wise probability heatmap before thresholding it to get a binary mask. Since traversability is dependent on the morphology of the robot (wheeled, legged, etc.), we prompt these segmentation models with a natural language query aligned with the specific robot platform (for example, "drivable ground" in the case of our wheeled rover). Then, we directly take the language-aligned raw probability heatmap to use as our traversability heatmap. Formally, the vision language model (VLM) $f_{\text{VLM}} : \mathcal{I} \times \mathcal{P} \to \mathcal{H}$ produces an image-space heatmap $H \in \mathcal{H} = [0,1]^{h_\mathcal{I} \times w_\mathcal{I}}$. We project this heatmap to a top-down viewpoint in order to construct a robot-space value map of traversability scores from the VLM.

Smooth Navigation Function. A navigation function $N : \mathbb{R}^2 \to \mathbb{R}$ maps every position $p = (x, y)$ to a scalar such that $N(p) = 0$ for any goal p, $N(p) = \infty$ for any unreachable p, and a local operator producing any intermediate values. However, as we have a smooth traversability map with continuous probabilities $\in [0,1]$ for every position instead of a binary $\{0,1\}$ obstacle map, we treat these probabilities as the speed in which the robot can move through that location. This is a natural generalization of traditional navigation functions, as a speed of 0 represents a solid obstacle that cannot be traversed and a speed of 1 represents a fully traversable region that does not hinder movement. Once we have the speed at each grid location, we can use wave propagation algorithms such as Fast Marching Method [15] or Fast Iterative Method [6] to produce the time it takes from any location to a goal, which is our smooth navigation function.

Traversable Trajectory Optimization. We use gradient descent to extract a trajectory from the combined results of our robot-space value map and smooth navigation function. We start our optimization procedure at the robot's position on the map and repeatedly take the step in a fixed radius that minimizes the time to goal location (provided by the navigation function). This generates a discrete set of waypoints starting at the robot and ending at the goal. To summarize, the final image feature is a list of top-down-space positions with respect to the robot's current coordinate frame $\mathbf{f} = ((0,0), (x_1, y_1), (x_2, y_2), ...)$, optimized with gradient descent over the navigation function taking into account geometric information and the value map of semantic information.

3.3. Error Minimization

First, we remove all waypoints too close to the robot as defined by a tuned threshold. Then, we take the first waypoint (x_i, y_i) with respect to the current robot coordinate frame and compute the error as the linear and angular distance to that waypoint from the current robot position. These distances are computed in an egocentric map which is updated on a longer time interval than the trajectory is recomputed, so these two error terms are defined as

$$e_{\text{linear}}(x_r, y_r, x_i, y_i) = \sqrt{(x_r - x_i)^2 + (y_r - y_i)^2} \tag{3}$$

$$e_{\text{angular}}(x_r, y_r, x_i, y_i) = \text{arctan2}(y_r - y_i, x_r - x_i) \tag{4}$$

respectively. We then produce linear and angular velocity actions via proportional control

$$(v^*, w^*) = (K_v e_{\text{linear}}(...), K_w e_{\text{angular}}(...))$$

where v^* is the velocity command and w^* is the angular velocity command to the robot.

4. Experiments

To measure the effectiveness of OWLS in practice, we evaluate our method in two practical scenarios – 1. an indoor hallway and 2. an offroad field with a mix of grass, dirt, bushes, trees, and pavement. First, we describe the implementation details of our method and experimental setup

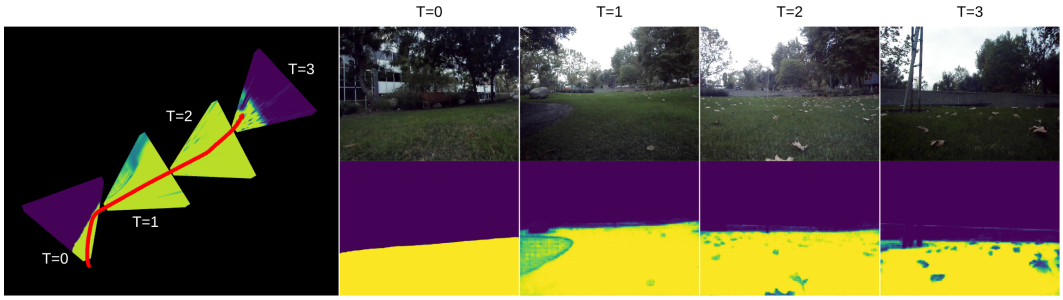

Figure 2. Successful navigation of the rover in an offroad field. 4 timesteps are chosen for visualization purposes, with the robot trajectory and topdown traversability on the left, camera views in the top row, and camera-view traversability in the bottom row.

(Sec. 4.1). Second, we describe our evaluation metrics and compare the performance over our environments.

4.1. Implementation

We use a 4-wheel rover for both scenarios, equipped with a front-facing RGB pinhole camera of 640 x 480 resolution. Our method is implemented and run on a laptop equipped with an NVIDIA RTX 4090 GPU, which receives wheel angle readings at 100Hz and image observations at 10Hz, generates features (trajectories) at 1Hz (25ms runtime, limited for stability), and publishes closed-loop wheel velocity commands at 100Hz. We compare 3 different heatmap generators in our pipeline: Grounded SAM 2 [13], DINOv3txt [18] with patch embeddings, and DINOv3txt with language embeddings.

4.2. Results

The **success rate** is the task success rate over 5 randomized trials, where a trial is considered successful if the model traverses the environment from one end to the other without colliding with any obstacles. This measures how consistently our method navigates a traversable area until it runs out of such area without further guidance from a global navigation module.

Figure 3. Navigation success rate, where an episode is considered successful if the model traverses the environment from one end to the other.

Our experiments show that Grounded SAM 2 (gsam2) performed best in all of our environments. Due to its strong spatial consistency, it perfectly avoided all static obstacles in both indoor and outdoor environments, achieving a 100% success rate (Fig. 3). It also avoided dynamic obstacles such as humans and moving boxes to some degree in indoor environments by correctly swerving away from them 80% of the time. However, due to the visual variations in outdoor environments and under-segmentation of dynamic obstacles, it achieved a much lower performance of 40% when operating with dynamic obstacles in outdoor scenes. While DINOv3 with patch sometimes successfully traversed the area, it was much less reactive than Grounded SAM 2 due to the heatmap being much more smoothly distributed. This limited DINOv3 with patch to only avoid obstacles that could be avoided through smaller steering actions, as it easily collided with obstacles that were directly in front of the robot. Finally, DINOv3 with language could not traverse any environments at all, heavily favoring turning left consistently without showing any reaction to changes in the image.

5. Conclusion

We propose OWLS, a novel approach for mobile robots to identify traversable regions and safely navigate them using a single front-facing camera. Our method uses VLMs to generate actionable traversability values directly from camera images, which benefits from generalizable visual understanding trained on internet-scale data. By carefully formulating our method as visual servoing, we avoid the overhead and complexity of having classical mapping and motion planning frameworks reliant on explicit geometric models / multimodal sensing, and directly optimize trajectories and actions from vision alone. Hardware experiments with a 4-wheel rover in indoor and outdoor scenarios demonstrate the viability of our method in widely varying visual and trajectory domains, while ablation studies of VLMs provide insights into the characteristics and pitfalls of using off-the-shelf VLMs for visual navigation. We plan to explore cross-platform generalization and reasoning as future work.

References

[1] Felipe Codevilla, Matthias Müller, Antonio López, Vladlen Koltun, and Alexey Dosovitskiy. End-to-end driving via conditional imitation learning. In *International Conference on Robotics and Automation (ICRA)*, 2018. 2

[2] Anushri Dixit, Ali-akbar Agha-mohammadi, David D Fan, Joel W Burdick, Yuki Kubo, and Kyohei Otsu. Step: Stochastic traversability evaluation and planning for risk-aware off-road navigation. *Robotics: Science and Systems (RSS)*, 2021. 2

[3] J.T. Feddema and O.R. Mitchell. Vision-guided servoing with feature-based trajectory generation (for robots). In *Transactions on Robotics and Automation (T-RO)*, 1989. 2

[4] Dylan Goetting, Himanshu Gaurav Singh, and Antonio Loquercio. End-to-end navigation with vision-language models: Transforming spatial reasoning into question-answering. In *International Conference on Neuro-symoblic Systems (NeuS)*, 2025. 2

[5] S. Hutchinson, G.D. Hager, and P.I. Corke. A tutorial on visual servo control. In *Transactions on Robotics and Automation (T-RO)*, 1996. 2

[6] Won-Ki Jeong and Ross T Whitaker. A fast iterative method for eikonal equations. In *SIAM Journal on Scientific Computing*, 2008. 3

[7] Anqiao Li, Chenyu Yang, Jonas Frey, Joonho Lee, Cesar Cadena, and Marco Hutter. Seeing through the grass: Semantic pointcloud filter for support surface learning. *Robotics and Automation Letters (RA-L)*, 2023. 2

[8] Yi Ma, J. Kosecka, and S. Sastry. Vision guided navigation for a nonholonomic mobile robot. In *Conference on Decision and Control (CDC)*, 1997. 2

[9] M. Maya-Mendez, P. Morin, and C. Samson. Control of a nonholonomic mobile robot via sensor-based target tracking and pose estimation. In *International Conference on Intelligent Robots and Systems (IROS)*, 2006. 2

[10] Xiangyun Meng, Nathan Hatch, Alexander Lambert, Anqi Li, Nolan Wagener, Matthew Schmittle, JoonHo Lee, Wentao Yuan, Zoey Chen, Samuel Deng, et al. Terrainnet: Visual modeling of complex terrain for high-speed, off-road navigation. *Robotics: Science and Systems (RSS)*, 2023. 2

[11] Sebastian Otte, Christian Weiss, Tobias Scherer, and Andreas Zell. Recurrent neural networks for fast and robust vibration-based ground classification on mobile robots. In *International Conference on Robotics and Automation (ICRA)*, 2016. 2

[12] R. Pissard-Gibollet and P. Rives. Applying visual servoing techniques to control a mobile hand-eye system. In *International Conference on Robotics and Automation (ICRA)*, 1995. 2

[13] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. Sam 2: Segment anything in images and videos. In *ArXiv Preprint*, 2024. 3, 4

[14] Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, Zhaoyang Zeng, Hao Zhang, Feng Li, Jie Yang, Hongyang Li, Qing Jiang, and Lei Zhang. Grounded sam: Assembling open-world models for diverse visual tasks. In *ArXiv Preprint*, 2024. 3

[15] James A Sethian. A fast marching level set method for monotonically advancing fronts. In *Proceedings of the National Academy of Sciences*, 1996. 3

[16] Amirreza Shaban, Xiangyun Meng, JoonHo Lee, Byron Boots, and Dieter Fox. Semantic terrain classification for off-road autonomous driving. In *Conference on Robot Learning (CoRL)*, 2022. 2

[17] Dhruv Shah, Arjun Bhorkar, Hrishit Leen, Ilya Kostrikov, Nicholas Rhinehart, and Sergey Levine. Offline reinforcement learning for visual navigation. In *Conference on Robot Learning*, 2022. 2

[18] Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. DINOv3. In *ArXiv Preprint*, 2024. 4

[19] Kasun Weerakoon, Mohamed Elnoor, Gershom Seneviratne, Vignesh Rajagopal, Senthil Hariharan Arul, Jing Liang, Mohamed Khalid M Jaffar, and Dinesh Manocha. Behav: Behavioral rule guided autonomy using vlms for robot navigation in outdoor scenes. In *International Conference on Robotics and Automation (ICRA)*, 2025. 2

[20] Mingfeng Yuan, Letian Wang, and Steven L. Waslander. Opennav: Open-world navigation with multimodal large language models. In *International Conference on Intelligent Robots and Systems (IROS)*, 2025. 2